Escuela Politécnica Superior

Red de comunicaciones para una entidad con 2 centros de cálculo y 200 sedes

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Javier Gombao Fernández-Calvillo

Tutor/es:
Luis Miguel Crespo Martínez

Septiembre 2019

Universitat d'Alacant
Universidad de Alicante

RESUMEN

La rápida y continua expansión de Internet ha supuesto un profundo cambio en el modo de comunicación y acceso a la información digital. Esta situación ha influido, también, en las redes de comunicaciones empleadas por todo tipo de entidades (empresas, universidades, bancos, administraciones públicas…) con el surgimiento de tecnologías que facilitan el uso de Internet como medio de transmisión para establecer enlaces de datos entre sedes remotas y sus centros de procesamiento de datos. Concretamente, este trabajo se focaliza en la creación de una red privada virtual dinámica multipunto, mediante el programa de simulación gráfico de red GNS3, utilizando, para ello, una topología completamente mallada con el uso de interfaces multiacceso en combinación con técnicas de tunelizado, encaminamiento y encriptación. Además, la red emplea protocolos y tecnologías adicionales que proporcionan alta disponibilidad, políticas de calidad de servicio eficientes para permitir el reparto de latencias y caudales según el tipo de aplicación, Multicast IP, componentes de seguridad; así como herramientas para la gestión y monitorización de red.

Palabras clave: *DMVPN, IPSEC, QoS, Multicast, EIGRP, NHRP, HSRP, Túnel IP, Cisco, Red de comunicaciones redundada, GNS3, mGRE, NBMA, Algoritmo DUAL, IGMP, PIM-SM, IKEv2*

ABSTRACT

The rapid and continuous expansion of the Internet has meant a deeply change in our way of communication and access to digital data. This situation has also influenced on the computer networks used by all types of companies, universities and colleges, banks, public administrations... with the emergence of technologies that facilitate the use of the Internet as a transmission tool to establish data links between remote sites and their data centres. Specifically, this work focuses on the creation of a Dynamic Multipoint Virtual Private Network with the GNS3 network graphical simulation program and using a full mesh topology with multiaccess interfaces in combination with techniques such as *tunneling*, routing and encryption. Furthermore, the network uses additional protocols and technologies which provide high availability, efficient quality of service policies in order to allow the distribution of latencies and application data flows, IP Multicast, security elements and tools for network management and monitoring.

Keywords: *DMVPN, IPSEC, QoS, Multicast, EIGRP, NHRP, HSRP, IP Tunnel, Cisco, Network redundancy, GNS3, mGRE, NBMA, DUAL algorithm, IGMP, PIM-SM, IKEv2*

JUSTIFICACIÓN Y OBJETIVOS

El desarrollo de Internet ha constituido una verdadera revolución en la sociedad moderna y es una pieza fundamental en nuestro día a día. Son muchas las organizaciones que están empleando esta red pública para el establecimiento de enlaces de comunicaciones entre oficinas y sus sedes centrales, usando tecnologías Cisco para instalar, configurar y administrar los dispositivos que conforman la infraestructura de red (*switches*, *routers*, *firewalls*…) debido a su gran fiabilidad y robustez en su sistema operativo Cisco iOS.

Hoy en día, existen programas de simulación que permiten la virtualización de redes de comunicaciones y por consiguiente, la creación y simulación de escenarios de redes sin la necesidad física de los equipos implicados ya sea hardware de interconexión de red físico, enlace o red (router, switch y hub); así como nubes o servidores en el nivel de aplicación. En este proyecto se usará GNS3 para generar y configurar una red virtual de comunicaciones DMVPN redundada, aplicable a una entidad con un centro principal de procesos de datos, uno secundario y 200 sedes remotas, mediante imágenes de routers virtuales que simulen el comportamiento de encaminadores reales. La **topología de red** que implementaremos será la siguiente:

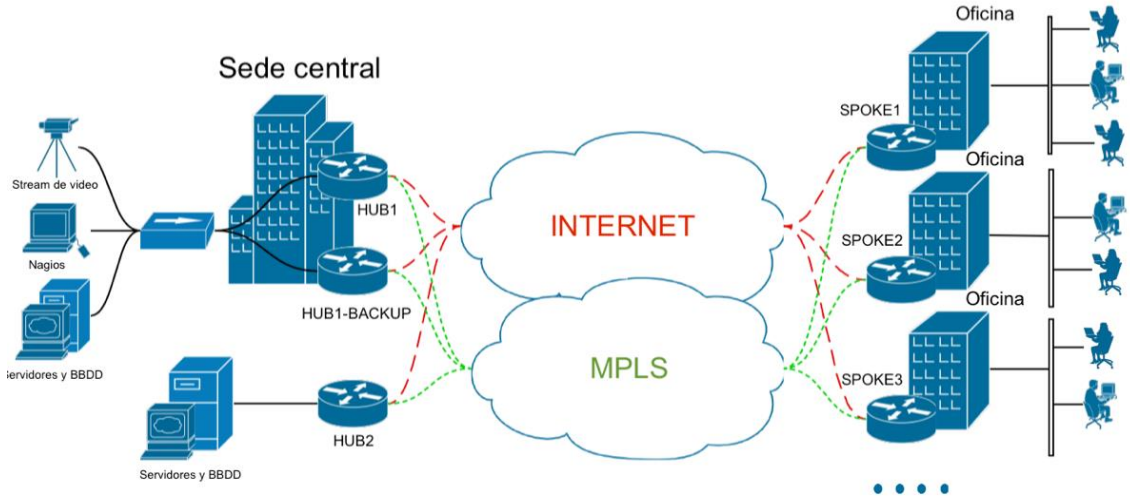


Figura 1 - Topología de red inicial[1]

[1] Fuente: elaboración propia

La ilustración anterior refleja el diagrama de red a desplegar. La topología utilizada será de tipo *Full-Mesh*, de modo que aunque todas las sedes se encuentren conectadas a los dos servidores y bases de datos, se podrán establecer enlaces de comunicaciones entre ellas dinámicamente y bajo demanda. Además, se utilizarán dos WAN de diferentes operadores: MPLS e Internet. Finalmente, la red deberá utilizar tecnologías que ofrezcan alta disponibilidad, calidad de servicio IP, enrutamiento de tráfico Multicast, seguridad en la transmisión de datos entre sedes y herramientas de monitorización y automatización.

AGRADECIMIENTOS

Quisiera dedicar esta sección a todas aquellas personas que me han ayudado a lo largo de todos estos años, comenzando por mi madre, mi padre y mi hermana por el apoyo incondicional diario y a los valores que me han inculcado. A mi abuela, por estar siempre pendiente de todo. A mi familia materna y paterna por todos sus cuidados.

A mi tutor de Trabajo Fin de Grado, Luis Miguel, por su tiempo, paciencia, comprensión y haberme orientado de un modo excepcional en el proceso de desarrollo de este proyecto.

A la Universidad de Alicante y a la Escuela Politécnica Superior por darme la oportunidad de haber cursado una doble titulación en el grado de Ingeniería Informática y un Bachelor of Science (Honours) in Network Management & Cloud Infrastructure en Athlone Institute of Technology (República de Irlanda).

A todos los docentes, de los cuales he sido alumno, desde la educación infantil hasta la Universidad, por compartir sus conocimientos y a guiarme en mi proceso de aprendizaje.

A mis amigos por comprender mis ausencias durante todo este tiempo.

Y finalmente, a los lectores.

CITAS

"No nos atrevemos a muchas cosas porque son difíciles, pero son difíciles porque no nos atrevemos a hacerlas."

- Séneca

"Si he logrado ver más lejos ha sido porque he subido a hombros de gigantes."

- Isaac Newton

"*Para mí, escribir es simplemente pensar con mis dedos.*"

- Isaac Asimov

CONTENIDOS

RESUMEN **1**
ABSTRACT **2**
JUSTIFICACIÓN Y OBJETIVOS **3**
AGRADECIMIENTOS **5**
CITAS **6**
ÍNDICE DE FIGURAS **10**
ÍNDICE DE TABLAS **12**
ÍNDICE DE ABREVIATURAS **13**
1. INTRODUCCIÓN **16**
2. MARCO TEÓRICO Y ESTADO DEL ARTE **19**
2.1 Tecnologías de operadores de acceso 19
2.1.1 Requerimientos de conectividad 19
2.1.2 Tecnologías de acceso 20
2.1.3 Indicadores para evaluar la conectividad 21
2.2 Redes DMVPN 22
2.2.1 Técnicas de tunelizado o "Tunneling" 23
2.2.2 Protocolo GRE 23
2.2.2.1 GRE punto a punto 25
2.2.2.2 GRE multipunto 26
2.2.3 Protocolo NHRP 26
2.2.4 Protocolo EIGRP 29
2.2.5 Protocolo HSRP 33
2.2.6 Calidad de servicio 34
2.2.6.1 Clasificación y marcado 35
2.2.6.2 Priorización del tráfico 36
2.2.6.3 Algoritmo de cubo de testigos 37
2.2.7 Multicast IP 38
2.2.8 Seguridad 42
2.2.8.1 Protocolo IPSec 42
2.2.8.1.1 Protocolo AH 43
2.2.8.1.2 Protocolo ESP 45
2.2.8.1.3 Modos de funcionamiento de IPSec 46
2.2.8.2 Protocolo de intercambio de claves IKE 48
2.2.8.2.1 Fase 1 49
2.2.8.2.2 Fase 2 50
2.2.8.3 Zona desmilitarizada 51
2.2.8.4 IDS 53
2.2.8.5 IPS 54
2.2.8.6 SIEM 54
2.2.8.7 Cisco Umbrella 56
2.2.8.8 Herramientas de detección de vulnerabilidades 57
2.2.8.9 Nessus 58
2.2.8.10 Acunetix web vulnerability scanner 58
2.2.9 Herramientas de monitorización 59
2.2.9.1 Nagios 60
2.2.9.2 Ntop 61
2.2.10 Netflow 61

3. OBJETIVOS 63

3.1 Objetivo general 63
3.2 Objetivos específicos 63
3.2.1 Alta disponibilidad 63
3.2.2 Seguridad 63
3.2.3 Calidad de servicio 64
3.2.4 Gestión de red 64
3.3 Topología de red en GNS3 64

4. METODOLOGÍA 66

5. DESPLIEGUE DEL PILOTO 68

5.1 Cálculo del ancho de banda para HUBs y SPOKEs 68
5.2 Modelo de router cisco para cada HUB y SPOKE 72
5.3 Bloque 1: Routing 77
5.3.1 Técnicas de tunelizado 77
5.3.1.1 Configuración 78
5.3.1.2 Fragmentación IP 84
5.3.1.3 Validación de conectividad HUB-SPOKE 86
5.3.1.4 Validación de conectividad SPOKE-SPOKE 93
5.3.1.5 Comprobaciones de Fragmentación IP 95
5.3.2 HSRP 97
5.3.2.1 Configuración 98
5.3.2.2 Validación 98
5.3.3 Conectividad a Internet 103
5.3.3.1 Configuración 103
5.3.3.2 Validación 105
5.3.4 IP secundaria en el centro de respaldo 107
5.3.5 Centralización de syslogs en los CPDs 109
5.3.6 Routing final 110
5.3.6.1 Verificación de la conectividad 110
5.3.6.2 Pruebas de basculación 112
5.3.6.2.1 Comunicación de SPOKE a HUB 113
5.3.6.2.2 Comunicación de HUB a SPOKE 117
5.4 Bloque 2: Calidad de servicio 118
5.4.1 Configuración 118
5.4.1.1 Definición de clases de tráfico 119
5.4.1.2 Definición de políticas 121
5.4.1.3 Aplicación de políticas a la interfaz 123
5.4.2 Validación 124
5.5 Bloque 3: Multicast IP 130
5.5.1 Configuración 131
5.5.2 Validación 131
5.6 Bloque 4: IPSec 135
5.6.1 Configuración 135
5.6.2 Validación 137
5.7 Bloque 5: Nagios 141
5.7.1 Configuración 142
5.7.2 Validación 143

CONCLUSIONES Y LÍNEAS FUTURAS 150

REFERENCIAS 153

ANEXO 155

Configuración de los routers 155
HUB1 155

HUB1-BACKUP *159*
HUB2 ..... *163*
SPOKE1 ..... *168*
SPOKE2 ..... *172*
SPOKE3 ..... *177*
R1 ..... *181*
R2 ..... *183*
COMANDOS CISCO IOS PARA COMPROBAR LA CONECTIVIDAD 185
HUB1 ..... *185*
HUB1-BACKUP *193*
HUB2 ..... *197*
SPOKE1 ..... *201*
SPOKE2 ..... *209*
SPOKE3 ..... *218*
CONFIGURACIÓN DE NAGIOS CORE 227
Servicio PING *227*
Máquina Nagios *228*
HUB1 ..... *228*
HUB1-backup *229*
HUB2 ..... *230*
SPOKE1 ..... *231*
SPOKE2 ..... *231*
SPOKE3 ..... *232*

ÍNDICE DE FIGURAS

FIGURA 1 - TOPOLOGÍA DE RED INICIAL 3
FIGURA 2 – ESQUEMA CONCEPTUAL DMVPN A UNA ENTIDAD. 22
FIGURA 3 - TÉCNICA DE TUNELIZADO 23
FIGURA 4 - PROTOCOLO GRE 24
FIGURA 5 - GRE PUNTO A PUNTO 25
FIGURA 6 - GRE MULTIPUNTO 26
FIGURA 7 - PROTOCOLO NHRP 27
FIGURA 8 - ADYACENCIAS EIGRP 29
FIGURA 9 - MÉTRICA EIGRP 30
FIGURA 10 - CÁLCULO FEASIBLE DISTANCE EIGRP 32
FIGURA 11 - SELECCIÓN SUCCESOR EIGRP 32
FIGURA 12 - PROTOCOLO HSRP 33
FIGURA 13 - MECANISMOS QOS 35
FIGURA 14 - CABECERA IP Y CAMPO TOS 35
FIGURA 15 - MÉTODO POLICING AND SHAPING 37
FIGURA 16 - TOKEN BUCKET 37
FIGURA 17 - SOURCE TREE 41
FIGURA 18 - SHARED TREES 41
FIGURA 19 - FUNCIONAMIENTO PROTOCOLO AH 44
FIGURA 20 - FUNCIONAMIENTO PROTOCOLO ESP 45
FIGURA 21 - MODO TRANSPORTE IPSEC 46
FIGURA 22 – APLICACIÓN DEL MODO TRANSPORTE IPSEC 46
FIGURA 23 - MODO TÚNEL IPSEC 47
FIGURA 24 - APLICACIÓN DEL MODO TÚNEL IPSEC 47
FIGURA 25 - FUNCIONAMIENTO IKEV1 50
FIGURA 26 - SERVIDOR UBICADO FUERA DE LA RED 51
FIGURA 27 - SERVIDOR UBICADO DENTRO DE LA RED 52
FIGURA 28 - DMZ 53
FIGURA 29 - APPLIANCE APLIENVAULT 55
FIGURA 30 - CISCO UMBRELLA 57
FIGURA 31 - GUI NESSUS 58
FIGURA 32 - GUI ACUNETIX 59
FIGURA 33 - MODELOS DE HERRAMIENTAS DE MONITORIZACIÓN 60
FIGURA 34 - FUNCIONAMIENTO NETFLOW 62
FIGURA 35 - REPORTING. NETFLOW ANALYZER 62
FIGURA 36 - TOPOLOGÍA DE RED EN GNS3 64
FIGURA 37 – HARDWARE. ISR ROUTERS 4000 73
FIGURA 38 – HARDWARE. LICENCIAS ROUTERS ISR 4000 73
FIGURA 39 – HARDWARE. BENCHMARK ROUTER ISR4221 Y ISR4321 74
FIGURA 40 – HARDWARE. BENCHMARK ROUTER ISR4431 75
FIGURA 41 – HARDWARE. BENCHMARK ROUTER ISR4451 75
FIGURA 42 – ROUTING. INTERFACES FÍSICAS EN EL PILOTO 79
FIGURA 43 – ROUTING. DIAGRAMA DE RED SIMPLIFICADO USANDO MPLS Y TUNNEL10 81
FIGURA 44 – ROUTING. PILOTO GN3 UTILIZANDO LA MPLS Y TUNNEL10 81
FIGURA 45 – MTU Y MSS 85
FIGURA 46 - FUNCIONAMIENTO DEL PROTOCOLO TCP 86
FIGURA 47 – ROUTING. CONECTIVIDAD HUB-SPOKE 87
FIGURA 48 – ROUTING. WIRESHARK. PAQUETES “HELLO” EIGRP 89

FIGURA 49 – ROUTING. WIRESHARK. VALOR "HOLD TIME" EN PAQUETES HELLO EIGRP 89
FIGURA 50 – ROUTING. CREACIÓN DE TÚNELES DINÁMICOS O DMVPN 93
FIGURA 51 – ROUTING. WIRESHARK. NHRP REGISTRATION REPLY Y NHRP REQUEST 95
FIGURA 52 – ROUTING. WIRESHARK. BIT DON'T FRAGMENT ACTIVO. FRAGMENTACIÓN IP 96
FIGURA 53 – ROUTING. MTU OF NEXT HOP 97
FIGURA 54 – ROUTING. HSRP EN EL PILOTO 97
FIGURA 55 – ROUTING. VALIDACIÓN HSRP 98
FIGURA 56 – ROUTING. WIRESHARK. ENVÍO DE PAQUETES HELLO DEL GRUPO HSRP 100
FIGURA 57 – ROUTING. CAÍDA DEL HUB1 101
FIGURA 58 – ROUTING. WIRESHARK. ACTUACIÓN DEL PROTOCOLO HSRP I 101
FIGURA 59 – ROUTING. WIRESHARK. ACTUACIÓN DEL PROTOCOLO HSRP II 102
FIGURA 60 – ROUTING. CONECTIVIDAD A INTERNET 106
FIGURA 61 – ROUTING. CONECTIVIDAD CON HUB2 107
FIGURA 62 – ROUTING. RUTA SEGUIDA PARA ALCANZAR EL CENTRO DE RESPALDO (IP SECUNDARIA) 108
FIGURA 63 – ROUTING. ENVÍO DE SYSLOGS AL CPD PRINCIPAL 110
FIGURA 64 – ROTING. TÚNELES Y ENLACES PRINCIPALES Y SECUNDARIOS ACTIVOS 113
FIGURA 65 – QOS. PASOS DE QOS CON MCLI 118
FIGURA 66 – QOS. APLICACIÓN DE POLÍTICAS A LAS INTERFACES 123
FIGURA 67 – QOS. ENVÍO Y RECEPCIÓN DE TRÁFICO ISÓCRONO 125
FIGURA 68 – QOS. ENVÍO Y RECEPCIÓN DE TRÁFICO HTTPS 125
FIGURA 69 – QOS. ENVÍO Y RECEPCIÓN DE FLUJOS ORACLE 125
FIGURA 70 – QOS. ENVÍO Y RECEPCIÓN DE TRÁFICO FTP 126
FIGURA 71 – QOS. NTOP. HOSTS QUE ESTÁN ENVÍANDO TRÁFICO 126
FIGURA 72 – QOS. NTOP. FLUJOS ENVIADOS POR LOS TERMINALES 127
FIGURA 73 – QOS. NTOP. FLUJOS ACTIVOS EN LA RED 127
FIGURA 74 – MULTICAST. SERVIDOR Y CLIENTES 130
FIGURA 75 – MULTICAST. ENVÍO Y RECEPCIÓN DEL STREAM 132
FIGURA 76 – MULTICAST. WIRESHARK. TRÁFICO UDP 133
FIGURA 77 – MULTICAST. WIRESHARK. MENSAJE IGMP LEAVE 133
FIGURA 78 – MULTICAST. WIRESHARK. MENSAJE MEMBERSHIP REPORT 134
FIGURA 79 – IPSEC. INTERCAMBIO DE MENSAJES EN IKEV2 137
FIGURA 80 – IPSEC. WIRESHARK. MENSAJES IKE_SA_INIT IKEV2 138
FIGURA 81 – IPSEC. WIRESHARK. MENSAJE IKE_SA_INIT (KEY EXCHANGE & NONCE) 139
FIGURA 82 – IPSEC. WIRESHARK. MENSAJE IKE_AUTH IKEV2 139
FIGURA 83 – IPSEC. WIRESHARK. CABECERA ESP 140
FIGURA 84 – NAGIOS. PANTALLA INICIAL 141
FIGURA 85 – NAGIOS. SERVICIOS 144
FIGURA 86 - NAGIOS. INFORMACIÓN DETALLADA DE UN HOST 145
FIGURA 87 - NAGIOS. TRENDS 146
FIGURA 88 - NAGIOS. AVAILABILITY 147
FIGURA 89 - NAGIOS. ALERTS 147
FIGURA 90 - NAGIOS. SIMULACIÓN DE CORTE DE LUZ EN UNA OFICINA 148
FIGURA 91 - NAGIOS. CAMBIO DE ESTADO 148
FIGURA 92 - NAGIOS. CAMBIO DE ESTADO II 149

ÍNDICE DE TABLAS

TABLA 1 - MEDIOS DE INTERCONEXIÓN 19
TABLA 2 - TECNOLOGÍAS DE ACCESO A INTERNET 21
TABLA 3 - INDICADORES DE LA EVALUACIÓN DE LA CONECTIVIDAD 22
TABLA 4 - MÉTRICAS EIGRP 31
TABLA 5 - PRIORIZACIÓN DEL TRÁFICO 36
TABLA 6 - VERSIONES IGMP 40
TABLA 7 - REQUERIMIENTOS DE UN CANAL SEGURO 43
TABLA 8 - POLÍTICAS APLICABLES A UNA DMZ 53
TABLA 9 – CLASIFICACIÓN DE OFICINAS 68
TABLA 10 - ANCHO DE BANDA POR OFICINA 69
TABLA 11 - ANCHO DE BANDA POR OFICINAS 70
TABLA 12 - ANCHO DE BANDA DE OFICINAS, SEDE CENTRAL Y CENTRO DE RESPALDO 71
TABLA 13 - ANCHO DE BANDA Y MODELO DE ROUTER ISR ADECUADO 76
TABLA 14 – ROUTING. FASES DMVPN 77
TABLA 15 – ROUTING. INTERFACES FÍSICAS DE LOS ROUTERS DE LA SIMULACIÓN 78
TABLA 16 – ROUTING. TÚNELES PRINCIPALES Y SECUNDARIOS 80
TABLA 17 – ROUTING. CONFIGURACIÓN TUNEL PRINCIPAL EN HUB1 82
TABLA 18 – ROUTING. CONFIGURACIÓN TUNEL PRINCIPAL EN SPOKE1 83
TABLA 19 – ROUTING. CONFIGURACIÓN DE HSRP 98
TABLA 20 – ROUTING. CONFIGURACIÓN DE NAT POR SOBRECARGA EN LAS INTERFACES 104
TABLA 21 – ROUTING. PRUEBAS DE BASCULACIÓN. CAÍDA DE ENLACES 114
TABLA 22 – ROUTING. PRUEBAS DE BASCULACIÓN. RECUPERACIÓN DE ENLACES 115
TABLA 23 – ROUTING. CAÍDA DE TÚNELES 115
TABLA 24 – ROUTING. PRUEBAS DE BASCULACIÓN. RECUPERACIÓN DE TÚNELES 116
TABLA 25 – ROUTING. PRUEBAS DE BASCULACIÓN. CAÍDA DE ROUTERS 116
TABLA 26 – ROUTING. PRUEBAS DE BASCULACIÓN. RECUPERACIÓN DE ROUTERS 117
TABLA 27 – ROUNTING. PRUEBAS DE BASCULACIÓN. CAÍDA Y RECUPERACIÓN DE ENLACES 117
TABLA 28 – ROTING. PRUEBAS DE BASCULACIÓN. CAÍDA Y RECUPERACIÓN DE TÚNELES 117
TABLA 29 - QOS. TIPOS DE TRÁFICO 118
TABLA 30 - CONFIGURACIÓN PIM-SM 131

ÍNDICE DE ABREVIATURAS

Término	Definición
3DES	Triple Data Encryption Standard
ACL	Access Control List
AES	Advanced Encryption Standard
AH	Authentication Header
AMP	Advanced Malware Protection
BER	Bit Error Rate
CLI	Command Line Interface
CPD	Center Process Data
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DMVPN	Dynamic Multipoint Virtual Private Network
DMZ	De-Militarized Zone
DNS	Domain Name Service
DUAL	Diffusing Update Algorithm
EIGRP	Enhanced Interior Gateway Routing Protocol
ESP	Encapsulating Security Protocol
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
HSRP	Hot Standby Router Protocol
IANA	Internet Assigned Number Authority
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IKEv2	Internet Key Exchange version 2
IOS	Internetwork Operating System

IP	Internet Protocol
IPmc	IP Multicast
IPS	Intrusion Prevention System
IPSec	IP Security
ISAKMP	Internet Security Association and Key Management Protocol
ISP	Internet Service Provider
LLQ	Low Latency Queuing
MD5	Message-Digest Algorithm 5
mGRE	Multiple Point-to-Point Generic Routing Encapsulation
MPLS	Multi-Protocol Label Switching
MTU	Maximum Transmission Unit
MSS	Maximum Segment Size
NAT	Network Address Translation
NBMA	Non-Broadcast multiple access
NHRP	Next Hop Resolution Protocol
NHS	Next-Hop Server
NIDS	Network Intrusion Detection Systems
NSM	Network Security Monitoring
NTOP	Network Top
OSPF	Open Shortest Path First
PAT	Port Address Translation
PC	Personal Computer
PIM – DM	PIM Dense Mode
PIM – SM	PIM Sparse Mode
QoS	Quality of Service
RIP	Routing Information Protocol
RTO	Retransmission Time Out
RTP	Real Time Protocol
RPF	Reverse Parh Forwarding
SA	Security Association

SHA-1	Secure Hash Algorithm One
SEM	Security Event Management
SIM	Security Information Management
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SMB	Server Message Block
SNMP	Simple Network Management Protocol
OS	Operating System
SRTT	Smooth Round Trip Time
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
URL	Uniform Resource Udentifier
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network

1. INTRODUCCIÓN

El surgimiento de las primeras redes de computadores fechan a principios de los años 60 del pasado siglo con la creación de las primeras redes de datos y el concepto de conmutación de paquetes. Este término hace referencia a la división de la información en paquetes con el objetivo de que pueda ser transportada por la red hasta su destino final, donde es recompuesta para formar la información original. A finales de los años 80, surgió el concepto de redes de área local (LAN) debido a la alta demanda de diversas empresas para conectar varios departamentos. En los 90, las redes se extendieron a una mayor escala dando lugar a lo que se conoce como a las comunicaciones de datos de área extensa (WAN). Desde entonces, las distintas organizaciones se vieron inmersas en todo tipo de necesidades donde Internet era la solución.

Actualmente, en el ámbito de las redes de comunicaciones, Internet está siendo empleado como medio de transmisión para intercambiar información, realizar transacciones y muchas más actividades, ya sea dentro de la misma entidad o en organizaciones diferentes. En una primera instancia, surgió la denominada Red Privada Virtual (VPN), tratándose de una topología de red que permite la extensión segura de la red LAN sobre una red pública no controlada como Internet. Las soluciones VPNs de capa 3 o nivel de red se aplican como estándar en una topología estrella utilizando los términos HUB o concentrador (en el centro de la red) y SPOKES o clientes (rodeando al HUB). Con la finalidad de poder establecer enlaces de comunicación que permitan conectar los elementos de la red con direccionamiento diferente y en puntos dispersos, se recurre a técnicas de tunelizado o "*tuneling*", consistente en encapsular un paquete IP privado dentro de un paquete público para que la información circule sobre Internet y una vez alcanzada la red privada del destino, la cabera IP pública sea eliminada y el paquete se encamine con las direcciones privadas.

En una solución **VPN**, al disponer de una **topología en estrella**, sería necesario configurar en el HUB tantos túneles como sedes remotas o SPOKES existan. Además, cada vez que se desee añadir un nuevo SPOKE, habría que intervenir en el HUB para configurar el túnel correspondiente. Esta forma de comunicación en contextos pequeños podría ser una solución viable, pero en entornos más grandes, podría generar problemas de administración ya que se presentarían dos inconvenientes: la solución no aporta escalabilidad (al incrementar la cantidad de SPOKES y enlaces redundantes, sería necesario configurar una gran cantidad de túneles en el HUB) y sería necesario generar una nueva subred IP por cada nueva conexión VPN entre los pares de enlaces correspondientes.

Ante esta situación, la compañía de telecomunicaciones Cisco Systems presentó una técnica que mejoraría el desempeño en la transferencia de información en las redes privadas virtuales sobre Internet: las conexiones multipunto a nivel de *WAN*, entre las que destaca la tecnología **DMVPN**, cuya principal característica es el establecimiento de enlaces dinámicos SPOKE-SPOKE que permiten la creación de una **red en malla**.

El propósito y la finalidad de este proyecto será la creación de una red de comunicaciones virtual DMVPN utilizando el programa de simulación GNS3. Nos focalizaremos en la configuración de los PCs, las máquinas virtuales conectadas a la red y los *routers*, cuyas imágenes virtuales simularán el comportamiento de la versión del modelo de router Cisco 7200 Series, puesto que ofrece una amplia gama de opciones y un buen rendimiento.

El resto del documento estará estructurado de la siguiente manera. En la próxima sección se detallará el marco teórico o el estado del arte y en él, los elementos conceptuales que sirven de base para la investigación. En el tercer punto se establecerán los objetivos generales y los específicos y también se reflejará la topología de red implementada en GNS3. La metodología que se ha llevado a cabo para realizar el proyecto quedará reflejada en el cuarto punto. En la quinta sección desarrollaremos el despliegue del Piloto, que a su vez está dividido en 7 bloques: cálculo del ancho de banda para HUBs y SPOKES, selección del modelo de router Cisco para cada HUB y

SPOKE, Routing, Calidad de Servicio, Multicast IP, IPSec y Nagios. Después, se detallarán las conclusiones en donde se remarcarán los objetivos y resultados conseguidos; así como las líneas futuras del. El documento finaliza con una bibliografía de los materiales consultados para la elaboración de la memoria y un Anexo donde quedarán reflejadas las configuraciones de cada *router*, verificaciones del buen funcionamiento de la simulación y los archivos de comunicación de la herramienta de monitorización empleada (Nagios).

2. MARCO TEÓRICO Y ESTADO DEL ARTE

En esta sección se detallarán los términos que servirán de base para el entendimiento y la correcta elaboración del proyecto propuesto.

2.1 TECNOLOGÍAS DE OPERADORES DE ACCESO

2.1.1 REQUERIMIENTOS DE CONECTIVIDAD

Los requerimientos de conectividad de cualquier organización compuesta por múltiples sedes suelen ser de dos tipos diferentes:

- **LAN to LAN**: conexión de dos o más redes locales separadas de manera geográfica.
- **HOST to LAN**: conexión de uno o varios usuarios itinerantes a una red LAN.

Normalmente, se emplean tres medios de interconexión para los dos requerimientos de conexión citados:

Medio de interconexión	Tecnologías	Características
Líneas dedicadas	TDM(PDS/SDH/SONET)	⇒ Información transmitida byte a byte. ⇒ Retardos de propagación muy bajos. ⇒ Se usan en transmisiones LAN to LAN.
Enlaces dedicados con redes de conmutación por circuitos virtuales	Frame Relay X25 MPLS Radioenlaces (LDMS, Wimax…)	⇒ Uso restringuido en LAN to LAN.
Redes privadas virtuales	Túnel	⇒ LAN to LAN ⇒ HOST to LAN.

Tabla 1 - Medios de interconexión[2]

[2] Fuente: elaboración propia

2.1.2 TECNOLOGÍAS DE ACCESO

Las tecnologías de acceso a Internet disponibles y más comunes actualmente son:

Tecnología	Características
ADSL	⇒ Utilizado para zonas residenciales. ⇒ Caudal asimétrico: la velocidad de transmisión de descarga es superior a la de envío (hasta 24Mbps) ⇒ Caudal compartido con otros usuarios de la red del operador.
SHDSL	⇒ Utiliza el bucle de abonado igual que ADSL. ⇒ Utiliza una codificación TC-PAM que permite caudales simétricos entre 2 y 4Mbps dependiendo del número de pares telefónicos empleados (generalmente 2 ó 4). ⇒ Se ofrecen mejoras de servicio.
HFC	⇒ Combinación de líneas de fibra óptica con cables coaxiales. ⇒ Proporción de mayor ancho de banda que ADSL y SHDSL. ⇒ Cobertura no tan extendida.
3G/4G	⇒ Accesos proporcionados por operadores de tecnología móvil ⇒ Utilizados en conexiones HOST to LAN.
LMDS/WIMAX	⇒ Establecimiento de radioenlaces con anchos de banda superiores a los 100 Mbps ⇒ LMDS requiere visibilidad directa entre antenas, WiMAX no. ⇒ WiMAX es capaz de adaptarse a las condiciones variables del medio utilizando mecanismos de control de potencia emitida, modulación adaptativa y selección automática de frecuencia.
FTTH	⇒ Empleo de cables de fibra óptica y sistemas de distribución ópticos para el despliegue de servicios residenciales con un ancho de banda elevado.

Tecnología	Características
100BASE-LX/LX 10/EX/ZX	⇒ Utilizados por operadores de servicio para productos empresariales. ⇒ Transportes de protocolos Ethernet con 802.1q mediante VLANs. ⇒ Ejemplos: SIG-ADI de ONO y accesos MACROLAN de Telefónica. ⇒ Velocidades de hasta 1Gbps.

Tabla 2 - Tecnologías de acceso a Internet[3]

2.1.3 INDICADORES PARA EVALUAR LA CONECTIVIDAD

Los parámetros básicos para la evaluación de la conectividad, son los siguientes:

Indicador	Características
Velocidad de transmisión	⇒ Número de bits de información transmitidos por unidad de tiempo.
Cadencia eficaz	⇒ Velocidad que percibe el usuario y difiere la velocidad que ofrece el operador.
MTU	⇒ Tamaño del datagrama IP que puede llegar a transmitir un medio físico (Ethernet, ATM…). Normalmente su valor es de 1500 bytes.
Latencia	⇒ Tiempo transcurrido desde que se introduce un paquete en la red hasta que llega a su destino. ⇒ Su valor depende del número de saltos que ha de realizar el paquete para atravesar la red y el tiempo de conmutación de los nodos de la red.
BER	⇒ La tasa de error se define como la cantidad de bits incorrectos divido entre el número de bits recibidos en forma porcentual. ⇒ Su valor depende del tipo de medio empleado, siendo el cable de fibra óptica el mejor y los radioenlaces los que presentan peores resultados. ⇒ Si este indicador es demasiado elevado, significa pérdida y repetición de paquetes por los niveles superiores (TCP, generalmente), provocando disminución de velocidad.

[3] Fuente: elaboración propia

Indicador	Características
SLA	⇒ Se trata del compromiso del proveedor en cuanto a tiempos de reparación de incidencias, disponibilidad del servicio y tiempos de respuesta se refiere.

Tabla 3 - Indicadores de la evaluación de la conectividad[4]

2.2 REDES DMVPN

Una red DMVPN consiste en el establecimiento de circuitos virtuales sobre red es públicas con la posibilidad de crear topologías en malla utilizando interfaces multiacceso (NBMA). La siguiente figura representa, a grandes rasgos, un ejemplo real de aplicación para una entidad con 3 oficinas:

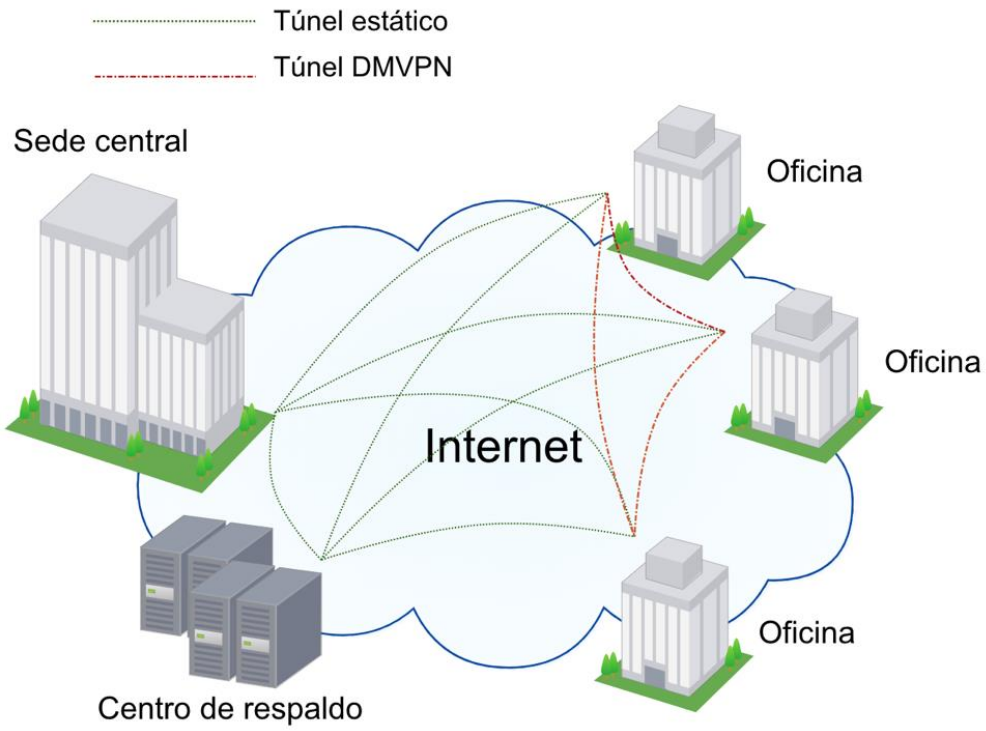


Figura 2 – Esquema conceptual DMVPN a una entidad.[5]

Una solución DMVPN consta a su vez de una combinación de tecnologías de tunelizado, encaminamiento y encriptación.

[4] Fuente: elaboración propia
[5] Fuente: elaboración propia

2.2.1 TÉCNICAS DE TUNELIZADO O "TUNNELING"

Es una técnica consistente en encapsular un paquete IP "pasajero" sobre otro "portador" para poder utilizar redes públicas y establecer sobre ellas enlaces que permitan conectar otras redes con direccionamiento diferente:

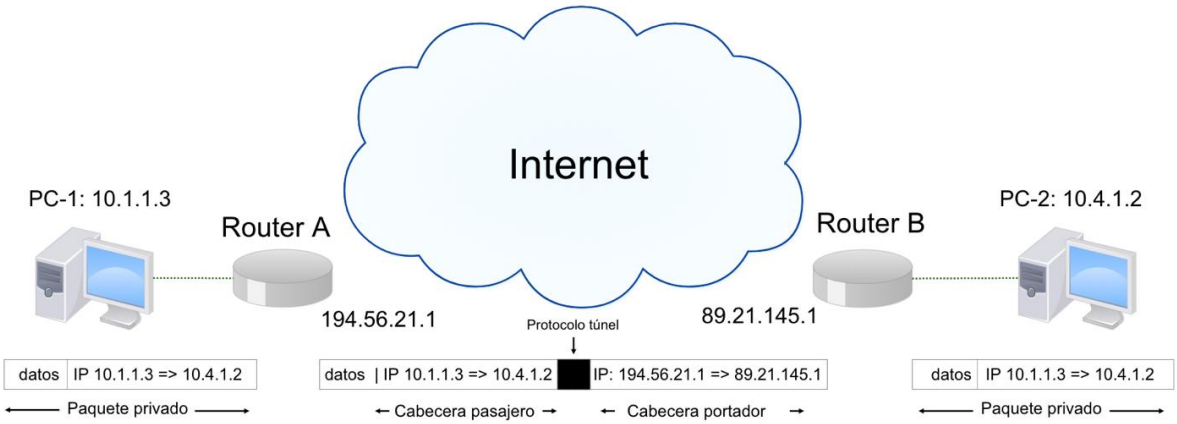


Figura 3 - Técnica de tunelizado[6]

En la imagen, el Router A encapsula el paquete privado con cabeceras IP públicas que pueden ser enrutadas sobre Internet. Posteriormente, el Router B elimina las cabeceras IP públicas del paquete recibido y devuelve el paquete original a la red privada.

Esta técnica proporciona bastantes ventajas, pero no está exenta de dificultades, tales como el aumento de información redundante, ya que se añade una cabecera IP nueva más otra correspondiente al protocolo de túnel y la fragmentación IP.

2.2.2 PROTOCOLO GRE

Se trata de un protocolo de *tunneling* desarrollado por Cisco y definido en las RFC 1701, RFC 2784 y RFC 2890 para el transporte de protocolos de nivel de red o nivel 3, denominados carga o *payload* sobre otro protocolo del mismo nivel. Este

[6] Fuente: elaboración propia

resultado es conseguido encapsulando la carga mediante una cabecera GRE y usando un paquete IP como portador.

Si el protocolo pasajero es de una trama de nivel 2 (PPP, Ethernet…) se dice que el túnel es de nivel 2, mientras que si el paquete pasajero es de nivel 3 (IPX, IP…) se denomina túnel de nivel 3.

La aplicación más típica es la conexión a una LAN privada de equipos remotos a través de una red pública como Internet. Este resultado se ve reflejado por una figura muy similar del apartado previo:

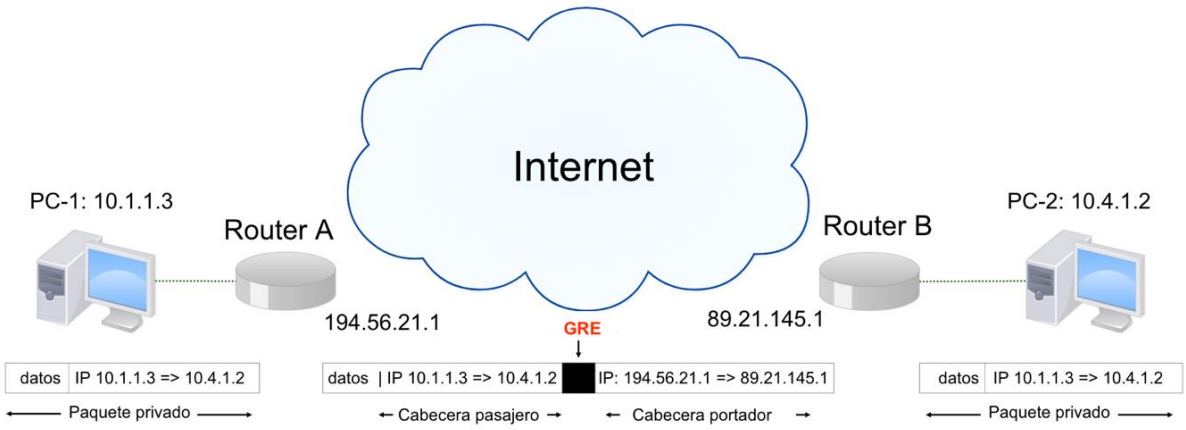


Figura 4 - Protocolo GRE[7]

Al encapsular la cabecera completa del *payload* o el paquete IP pasajero, se mantienen las direcciones, permitiendo el transporte de direcciones Broadcast (direcciones de difusión) o Multicast (direcciones de grupo). Este factor facilita la puesta en escena de protocolos de enrutamiento dinámicos que permitirán el establecimiento dinámico de enlaces redundantes de la VPN, proporcionando características de alta disponibilidad.

Otras características importantes de este protocolo son la posibilidad de enlazar redes multiprotocolo sobre una única red, posibilidad de interconectar subredes con direccionamiento IP discontinuo, posibilidad de formar redes privadas virtuales sobre redes WAN y permiten resolver incidencias de protocolos con un número de saltos

[7] Fuente: elaboración propia

limitados. No obstante, GRE no proporciona seguridad en las transmisiones de datos pero esta carencia será subsanada mediante la aplicación del protocolo IPSec.

Dentro del sistema operativo de red Cisco IOS, la configuración de GRE se realiza utilizando lo que se denominan interfaces virtuales, utilizando la secuencia de comandos '`interface tunnel x`' donde x es el identificador que permite identificar los túneles. Con esto, se puede configurar GRE con sus dos modos de funcionamiento: Punto a Punto y Punto a Multipunto.

2.2.2.1 GRE PUNTO A PUNTO

En este modelo se necesita configurar una interfaz túnel por cada conexión. En una topología en estrella, por tanto, se ha de configurar en el Router concentrador (HUB), tantos interfaces túnel como sedes remotas (SPOKEs) existan.

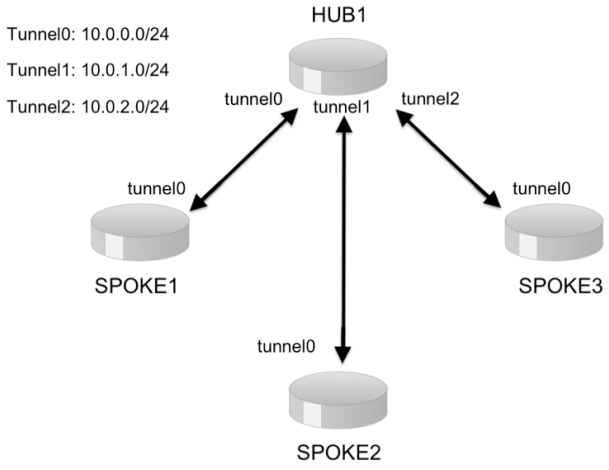


Figura 5 - GRE punto a punto[8]

Esta variante de funcionamiento GRE puede ser aplicable para entidades de pequeño tamaño, pero en circunstancias de aumento del número de SPOKEs y enlaces redundantes, existirían problemas de escalabilidad ya que habría que intervenir en el HUB para configurar las interfaces túnel y crear nuevas subredes IP por cada nueva conexión VPN entre el par de interfaces correspondientes, pudiendo ser un proceso tedioso.

[8] Fuente: elaboración propia

2.2.2.2 GRE MULTIPUNTO

En este caso, sólo se requiere crear una interfaz túnel en el router 'HUB' independientemente de la cantidad de routers SPOKE con los que se establecen los túneles GRE, resolviendo el problema de escalabilidad del modelo anterior.

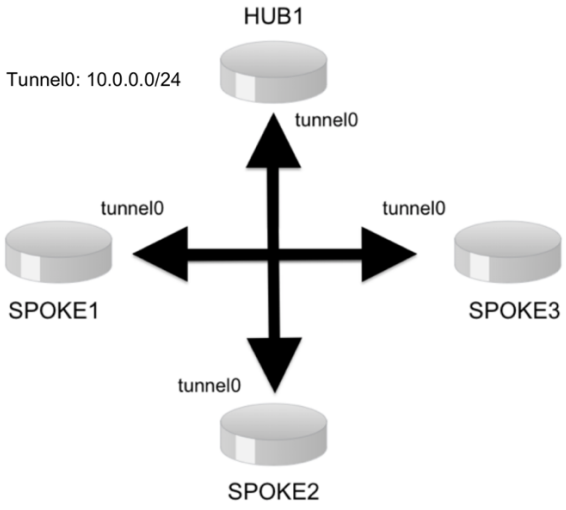


Figura 6 - GRE multipunto[9]

Otra diferencia importante con el modelo GRE punto a punto es que todas las direcciones IP de las interfaces se encuentran en una única subred, construyendo, de este modo, una red tipo NBMA: red con características de difusión, pero con conexiones punto a punto (el HUB podría enviar mensajes que lleguen a todos los SPOKEs al mismo tiempo), facilitando la aplicación de protocolos de enrutamiento y simplificación de la gestión de las direcciones IP. A todo esto, mGRE requiere de la colaboración de otro protocolo denominado NHRP, explicado en el siguiente apartado.

2.2.3 PROTOCOLO NHRP

En las redes tipo NBMA, es necesario la utilización de un protocolo que permita diferenciar los flujos procedentes de las distintas conexiones remotas y que confluyen en la misma interfaz. Esto se consigue mediante el empleo de un protocolo, muy similar ARP o a Frame Relay Inverse-ARP, denominado NHRP (definido en la RFC

[9] Fuente: elaboración propia

2332), cuyo principal cometido es resolver el problema de conocer a través de que dirección IP pública o dirección NBMA se puede alcanzar una determinada red privada. A continuación, explicaremos, brevemente, el funcionamiento de NHRP teniendo presente la siguiente figura:

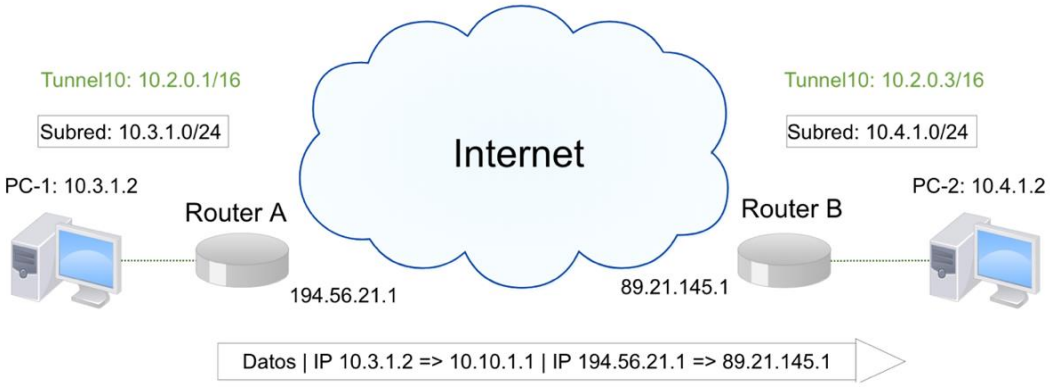


Figura 7 - Protocolo NHRP[10]

El Router B registra junto a el Router A su dirección IP pública o NBMA (89.21.145.1). Debemos tener en cuenta que el Router B ya posee en su configuración la dirección 194.56.21.1 asociada al Router A y por eso este proceso se efectúa en primer lugar.

El Router A construye una tabla con las direcciones privadas de las interfaces tunel con su dirección IP pública asociada:

Tabla NHRP Router A
10.2.0.3 → 89.21.145.1/32

De esta manera, el Router A puede incluir una entrada en su tabla de rutas que permita alcanzar la subred privada 10.4.1.0/24 utilizando su interfaz túnel '*Tunnel10*' con la dirección de próximo salto 10.2.0.3.

Tabla de rutas Router A
10.4.1.0/24 => Tunnel10 por 10.2.0.3

Este mecanismo permite la configuración de la interfaz túnel del HUB sea simple, sin necesidad de conocer las direcciones IP NBMA del resto de SPOKES,

[10] Fuente: elaboración propia

garantizando escalabilidad de red dado que para conectar un nuevo SPOKE se únicamente se configurarían las direcciones IP locales, la dirección IP del HUB y otras variables comunes.

Además, como el HUB aprende las direcciones NBMA de los SPOKES a través de NHRP, éstas pueden ser dinámicas, reduciendo, de esta forma, los costes de conexión ya que no sería un requisito indispensable disponer de una dirección IP estática en el SPOKE.

Por otro lado, para establecer conexiones dinámicas entre SPOKES, un SPOKE preguntará al HUB por la dirección NBMA del SPOKE vecino y seguidamente, se podrá iniciar la negociación de un túnel a través de dicha dirección.

En la configuración del NHRP se puede incluir un temporizador (*holdtime*), que establece el tiempo de almacenamiento de la información de la tabla NHRP en una memoria de tipo 'caché'. Tras la expiración del temporizador (por defecto de 2 horas), se fuerza a que la información sea aprendida nuevamente.

La tabla NHRP puede contener tanto entradas estáticas como dinámicas: en los HUB las entradas suelen ser añadidas dinámicamente (empleando peticiones de resolución y de registro) mientras que en los SPOKES se configuran entradas NHRP estáticas que apuntan al HUB.

Para poder participar en un proceso de petición o registro, todos los routers han de tener un identificador, detallando de esta manera que se pertenece al mismo dominio NHRP. En un mismo router pueden haber diversos dominios NHRP con sus respectivas IP públicas.

Cada SPOKE ha de ser configurado con la dirección NBMA del HUB para indicar de esta manera el servidor de próximo salto o NHS. De esta forma, se envía al HUB una solicitud de registro que contiene su dirección NBMA y la dirección privada de la interfaz túnel. Posteriormente, el HUB genera una entrada en su tabla NHRP y devuelve una respuesta de registro. Después, el SPOKE considerará al HUB como un servidor válido NHS y lo empleará como base para conocer las direcciones de otros SPOKES de la red.

2.2.4 PROTOCOLO EIGRP

Para mantener siempre la mejor conectividad de cada router con sus vecinos y aportar características de alta disponibilidad en la red de comunicaciones, se utilizan protocolos de encaminamiento dinámicos encargados de gestionar las tablas de rutas.

En una red DMVPN pueden existir varios caminos para llegar a una red privada en concreto, ya que es posible crear redes en estrella o parcialmente malladas. Esta situación conlleva la puesta en escena de un protocolo de enrutamiento dinámico, que seleccione el mejor camino existente según la topología existente.

En esta ocasión nos focalizaremos en la descripción de EIGRP, pues se trata de protocolo de encaminamiento dinámico híbrido, que trata de obtener las mejores prestaciones de los protocolos de encaminamiento de vector distancia, como RIP, y de estado de enlace, como OSPF. Además, provee una convergencia rápida, libre de bucles y *classles* (permite la transmisión de la máscara IP junto con la dirección de red asociada).

En la siguiente imagen podemos observar un diagrama básico de las adyacencias EIGRP. El proceso de descubrimiento y monitorización de routers adyacentes se efectúa mediante el envío periódico de paquetes tipo "Hello", utilizando la dirección Multicast 224.0.0.10 con el número de protocolo IP 88. Para asegurar la entrega de paquetes de manera ordenada y libre de errores se emplea el protocolo fiable RTP.

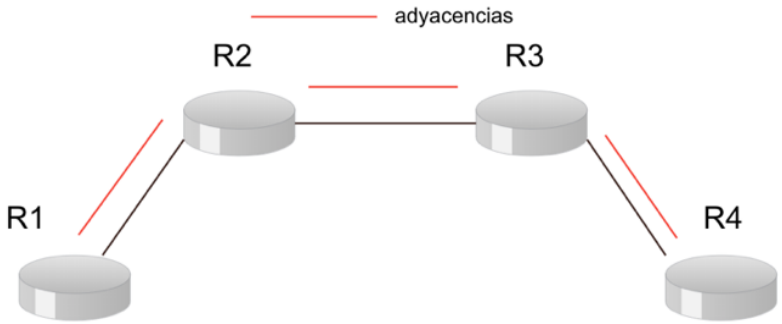


Figura 8 - Adyacencias EIGRP[11]

[11] Fuente: elaboración propia

Las adyacencias son almacenadas en la **tabla de vecinos**. Se emplea también una **tabla de topología**, que contiene todas las rutas a un destino en concreto y una **tabla de rutas** encargada de guardar la mejor ruta a un destino (“*Sucessor”)* y un sucesor posible que se activaría cuando el sucesor falle por cualquier motivo (“*Feasible Successor*”).

El funcionamiento de EIGRP se sustenta en el algoritmo DUAL, que se caracteriza por la compartición de cálculo de rutas, enviando únicamente actualizaciones a aquellos vecinos que los necesiten en caso de un cambio de topología. En contextos normales, circulan pequeños paquetes de reconocimiento; minimizando de esta manera el ancho de banda y consumo excesivo de recursos.

La métrica EIGRP es conocida como *Feasible Distance*, que consiste en una combinación de dos valores: la *Local Distance* (valor para obtener el router vecino) y la *Reported Distance* o *Advertised Distance* (valor obtenido desde el enrutador vecino al destino):

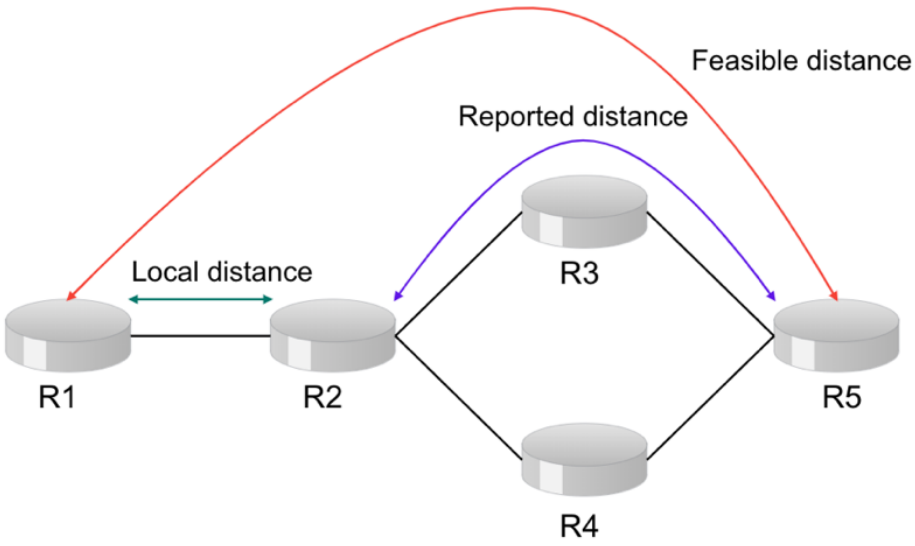


Figura 9 - Métrica EIGRP[12]

La *Feasible Distance* está compuesta por las siguientes métricas o valores K:

[12] Fuente: elaboración propia

Métrica	Características
K1. Ancho de banda de la interfaz	⇒ Medida de datos y recursos de comunicación disponible o consumida (medición en Kbps).
K2. Carga de la interfaz	⇒ Nivel de ocupación en el último intervalo.
K3. Retardo	⇒ Cantidad de tiempo que tarda una interfaz en codificarla y enviarla.
K4. Fiabilidad	⇒ Número de veces que la interfaz pierde disponibilidad. ⇒ Puede variar entre 0 y 255.
K5. MTU	⇒ Máximo tamaño del paquete. ⇒ No forma parte del cálculo empleado para la métrica.

Tabla 4 - Métricas EIGRP[13]

La expresión que permite obtener el valor de la métrica es:

$$metric = \left[\left(k_1 \cdot \frac{10^7}{BW_{min}} + \frac{k_2 \cdot BW_{min}}{256 - load} + k_3 \cdot \sum delays\right) \cdot \frac{k_5}{k_4 + reliability}\right] \cdot 256$$

Dado que por defecto el ancho de banda y retardo, K1 y K3 respectivamente, tienen valor 1 y el resto de indicadores valor 0 (no se emplean), la expresión simplificada es:

$$metric = \left(\frac{10^7}{BW_{min}} + \sum delays\right) \cdot 256$$

En la siguiente imagen se muestra el proceso de obtención de la *Feasible Distance* para alcanzar un destino en concreto (R5), suponiendo que cada valor expuesto en la imagen ha sido obtenido mediante la formula simplificada de la métrica EIGRP:

[13] Fuente: elaboración propia

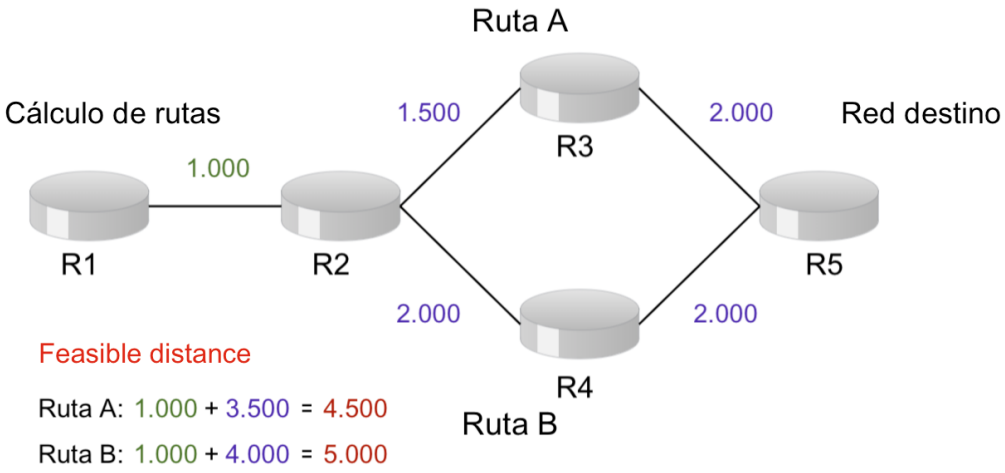


Figura 10 - Cálculo Feasible distance EIGRP[14]

Estos cálculos permitirán conocer cuál es el mejor camino a un destino (*Succesor*). Siguiendo este ejemplo, el mejor camino es R1 – R2 – R3 – R5 (Ruta A):

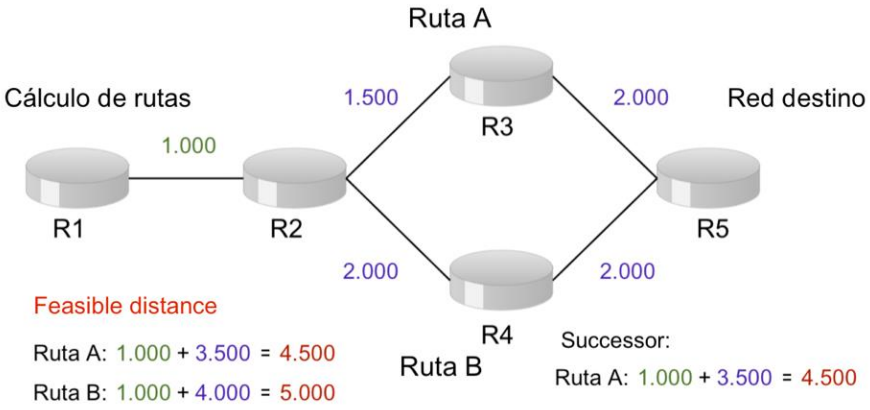


Figura 11 - Selección Succesor EIGRP[15]

El *Feasible Successor*, Ruta B en este caso, corresponde con la segunda mejor ruta existente en la topología y actúa como *backup* para proporcionar propiedades de convergencia rápida en caso de que el *Successor* falle. La *Reported Distance* para un destino concreto es comparada con la *Feasible Distance* para ese mismo destino. Si la distancia reportada es mayor que la *Feasible Distance*, la ruta no se introduce en la tabla de topología como un sucesor factible, evitando, de este modo, la aparición de bucles de encaminamiento. Ahora bien, si la *Reported Distance* es menor que la

[14] Fuente: elaboración propia
[15] Fuente: elaboración propia

Feasible Distance, el camino es considerado como un *Feasible Successor* y se introduce en la tabla de topología. Continuando con el ejemplo expuesto:

$$Successor\ route\ A: 1{,}000 + 3{,}500 = 4{,}500$$

$$Potential\ Feasible\ Successor\ (FS)\ route\ B: 1{,}000 + 4{,}000 = 5{,}000$$

$$FS\ reported\ distance < Succesor\ feasible\ distance \rightarrow 4.000 < 4{,}500$$

$$Ruta\ B\ considerado\ como\ Feasible\ Successor \rightarrow Introducción\ en\ tabla\ de\ topología$$

2.2.5 PROTOCOLO HSRP

HSRP es un protocolo propietario de Cisco y publicado posteriormente en la RFC 2281, que permite la compartición de una o varias direcciones MAC e IP entre un grupo de routers.

En la configuración del HSRP, disponemos de dos tipos de routers: el activo y el pasivo o Standby. El Router activo es aquel que responde a las peticiones realizadas a la dirección IP virtual. En caso de que se diese la situación de que el Router activo falle, el router pasivo perteneciente al mismo grupo HSRP, adquiere el rol de router activo y atiende a las peticiones enviadas a la dirección IP virtual, proporcionando un nivel de alta disponibilidad a los terminales que utilicen como puerta de enlace la dirección IP y MAC virtual:

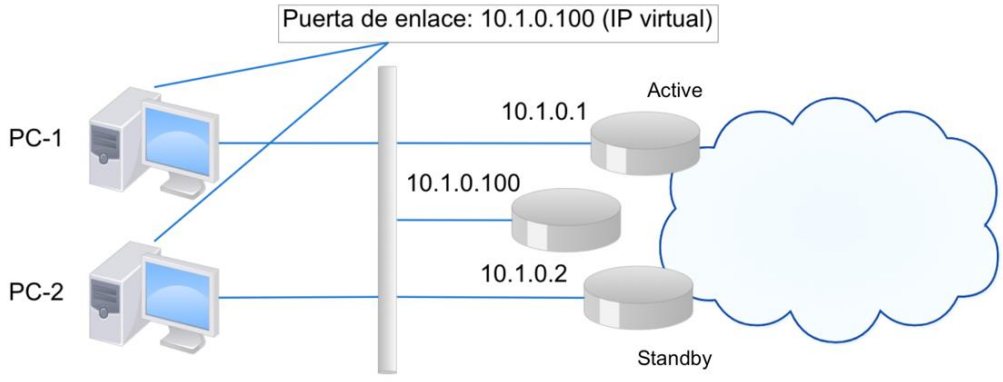


Figura 12 - Protocolo HSRP[16]

[16] Fuente: elaboración propia

Los participantes en el grupo HSRP envían y reciben paquetes "*Hello*" al puerto UDP 1985 con dirección IP *multicast* 224.0.0.2 para detectar fallos y asignar los modos de activo y pasivo.

2.2.6 CALIDAD DE SERVICIO

La calidad de servicio (QoS) se puede definir de dos formas diferentes en función del rol desempeñado de un usuario en la red. Para un usuario normal, QoS significa que las aplicaciones de los dispositivos se están ejecutando adecuadamente, es decir, no existen pérdidas en transferencias de voz, los vídeos son fluidos y de alta calidad, la red tiene tiempos de respuesta aceptables…. Para un administrador de red, significa el aprovechamiento de la utilización del ancho de banda mediante el uso de técnicas que permitan adaptar los recursos disponibles a la demanda de las diferentes aplicaciones. En otras palabras, el envío de emails requerirá un gran ancho de banda sin importar la latencia; pero una aplicación interactiva, como por ejemplo, VozIP necesitará poco ancho de banda, pero una baja latencia.

La aplicación de técnicas de calidad tiene dos ventajas principales. Primeramente, es posible permitir que los flujos más sensibles al retardo sean empleados antes que el resto de flujos. En segundo lugar, provee una mayor fiabilidad en la red gracias al control de la cantidad de ancho de banda que se puede usar para una aplicación en concreto. Por tanto, se requieren dos bloques diferentes para una correcta aplicación de políticas de calidad de servicio: el etiquetado o **marcado de paquetes** y un **sistema de priorización de colas** para que se marquen los paquetes a la entrada de la red y, después, priorizarlos en las interfaces de salida.

Un punto importante a considerar es la aplicación de técnicas de QoS en redes donde se aplican protocolos de cifrado IPSec ya que, al estar los paquetes de datos cifrados, se ha de recurrir a estrategias que puedan heredar en la cabecera del paquete IP portador algunas características que permitan la trazabilidad del pasajero, sin que la seguridad se vea comprometida.

La gestión de calidad de servicio se aplica de modo unidireccional, y por tanto, su diseño, se debe efectuar en los dos sentidos de la transmisión por separado. En la

siguiente figura los *routers* clasifican el tráfico procedente de la LAN privada para su posterior procesamiento a la salida de la interfaz WAN, lugar donde se podría generar una congestión significativa.

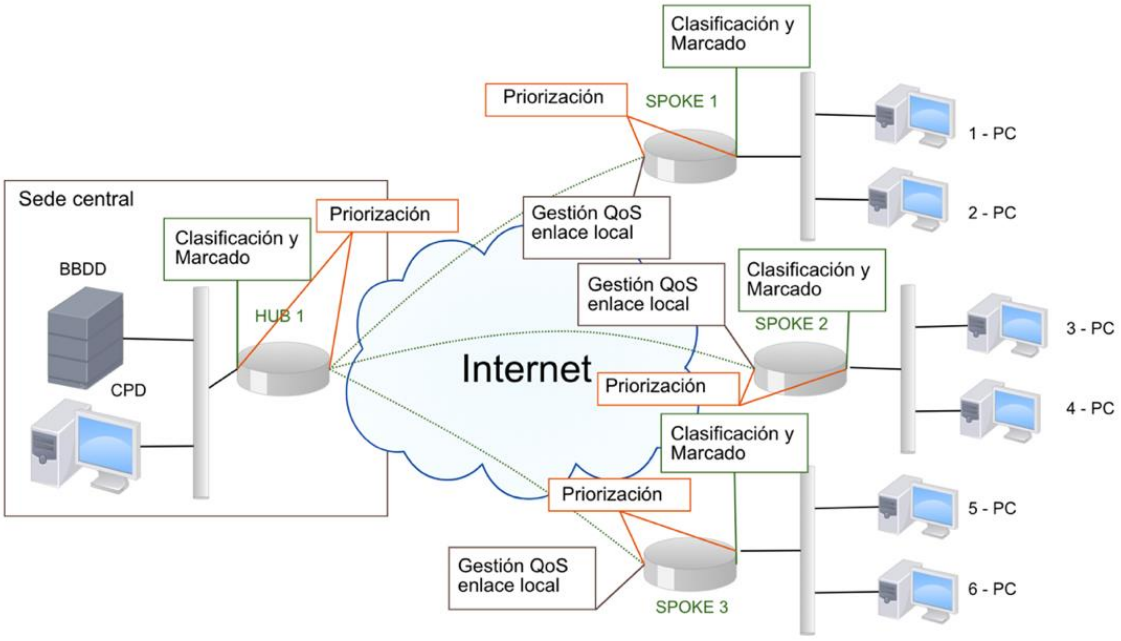


Figura 13 - Mecanismos QoS[17]

2.2.6.1 CLASIFICACIÓN Y MARCADO

Se lleva a cabo utilizando los bits del campo TOS de la cabera IP:

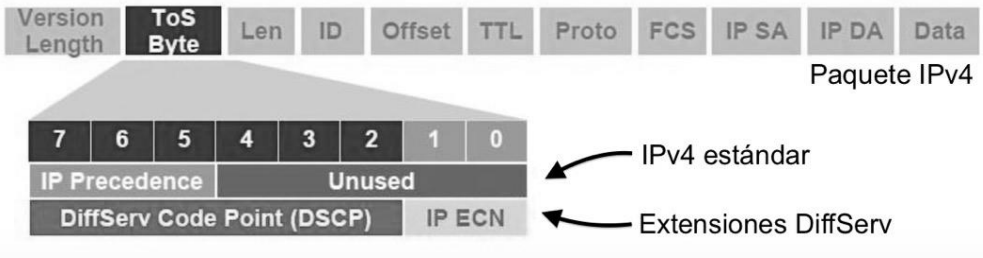


Figura 14 - Cabecera IP y campo TOS[18]

Dependiendo de los niveles de calidad de servicio requeridos, pueden ser utilizados únicamente los bits de Precedencia o los bits del DSCP.

[17] Fuente: elaboración propia
[18] Fuente: *QoS Strategies and Smart Media Techniques for Collaboration Deployments. Cisco.*

2.2.6.2 PRIORIZACIÓN DEL TRÁFICO

Existen tres funcionalidades distintas para actuar sobre el trafico saliente de una interfaz de un *router* Cisco IOS.

Métrica	Características
LLQ	⇒ Limita el número de paquetes en espera de ser transmitidos, reduciendo la latencia.
Traffic Shaping	⇒ Consiste en el retardo los paquetes de una transmisión TCP, de manera que el RTT se ajuste al valor deseado para que el emisor envíe sus paquetes con más lentitud. ⇒ Se ha de disponer de colas y suficiente memoria para poder guardar de manera temporal los paquetes retardados. ⇒ Produce una forma de tráfico constante sin pérdidas de paquetes.
Traffic Policing	⇒ Consiste en la eliminación de paquetes de una conexión TCP con el objeto de provocar una retransmisión de paquetes al expirar el tamaño de ventana, frenando la transmisión y reduciendo la velocidad. ⇒ No se requiere la necesidad de disponer de colas y memoria para el almacenamiento temporal de los paquets retardados. ⇒ Forma de tráfico no constante y posible pérdidas de paquetes.

Tabla 5 - Priorización del tráfico[19]

En definitiva, podemos constatar que el "*Traffic Policing*" es adecuado para el tráfico donde existen garantías de calidad de servicio, es decir, a flujos sensibles al retardo como VoIP y a servicios de datos de operadores y. Por otra parte, el '*Traffic Shaping*" es idóneo para aquel tráfico donde la pérdida de datos no es tolerable como el tráfico de datos TCP (reducción en el rendimiento de la comunicación).

[19]Fuente: elaboración propia

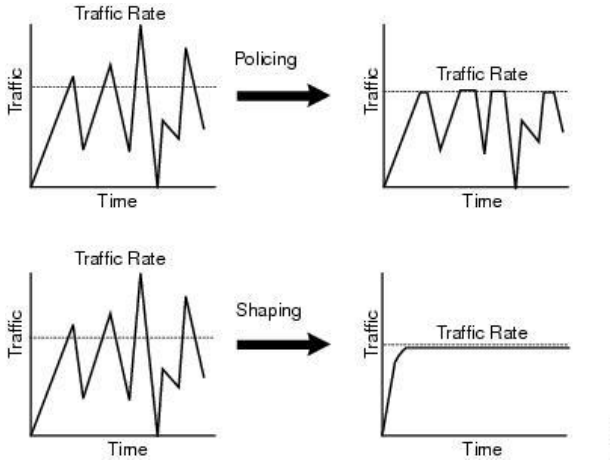


Figura 15 - Método Policing and Shaping[20]

2.2.6.3 ALGORITMO DE CUBO DE TESTIGOS

Tanto el método "*Shaping*" como el "*Policing*" se basan en el algoritmo Token Bucket o cubo de testigos, cuya misión principal es verificar que las transmisiones de datos se ajusten a los límites definidos en el ancho de banda y ráfagas (medida de la irregularidad o variaciones en el flujo de tráfico). En un router, se aplica en la interfaz correspondiente y consiste en un Buffer de memoria denominado "cubo", el cual recibe a una velocidad constante configurada por el usuario, una serie de testigos o Tokens. La ilustración representa el esquema de funcionamiento de este mecanismo:

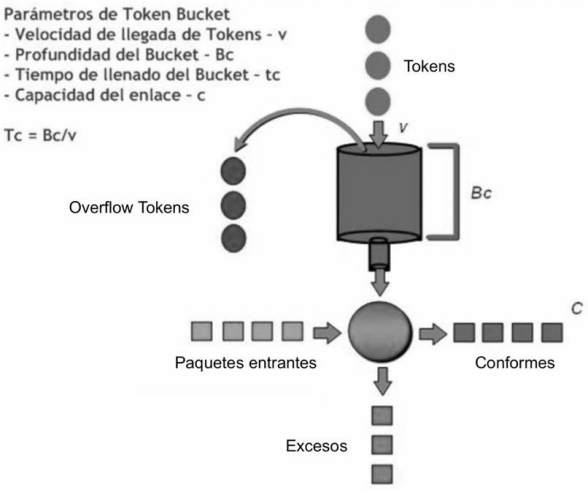


Figura 16 - Token Bucket[21]

[20] Fuente: materiales de la asignatura de Interconexión de Redes. José Ángel Berná
[21] Fuente: materiales Redes NBMA. Luis Miguel Crespo

Ante este escenario, se pueden aplicar las siguientes reglas:

1. Cuando el cubo está lleno, los Tokens que llegan después son descartados.
2. Cuando una ráfaga o paquete de datos llega al sistema, se comprueba si en el cubo existen suficientes Tokens para que éste pueda ser transmitido.
3. Si existen suficientes tokens, los paquetes son enviados suprimiendo los Tokens correspondientes del cubo.
4. Si no existen suficientes tokens, pueden ocurrir las siguientes circunstancias:
    a. Los paquetes se retardan en un Buffer temporal (Shaping).
    b. Los paquetes son eliminados (Policing).
    c. Los paquetes se marcan con un nuevo valor del campo TOS y se envían.

Según la acción que se lleve a cabo en caso de no existir suficientes Tokens para transmitir paquetes de datos, se adoptará una implementación de tipo '*Policing' o* '*Shaping'*, dependiendo de si dichos paquetes se retardan o suprimen respectivamente.

2.2.7 MULTICAST IP

Dentro de una red IPv4 se pueden distinguir tres tipos de tráfico: *Unicast* (envío de información desde un único emisor a un único receptor con independencia de la topología y ubicación), *Broadcast* (envío de una única trama a un número concreto de máquinas dentro del mismo segmento de red o subred) y *Multicast* (el tráfico es entregado a un conjunto de máquinas que han sido configuradas previamente como miembros de un grupo *Multicast* con posibilidad de localización en varias subredes dispersas).

La *multidifusión IP* o *multicast IP* permite optimizar el ancho de banda y reduce el caudal requerido para entregar un único flujo de datos (*stream*) a una gran cantidad de receptores, que pueden estar localizados en varias subredes dispersas. Esta tecnología es ampliamente utilizada en el campo de la transmisión de vídeo y audio, distribución masiva de software, sincronización horaria…

Cualquier máquina o dispositivo puede adherirse como abandonar uno o varios grupos de *Multicast* en cualquier instante, sin limitación en el número y ubicación de los participantes. Los grupos *Multicast* se identifican por la dirección IP de clase D, cuyo rango es 224.0.0.0 – 239.255.255.255. Utilizando este rango, pueden darse dos situaciones para el envío y recepción de tramas Multicast:

1. **Transmisor y el receptor se encuentra en el mismo segmento de red**. El emisor dirige el paquete IP hacia la dirección del grupo Multicast. Seguidamente, la interfaz de red realiza la asociación entre la dirección IP de clase D y la correspondiente Ethernet IEEE-802 de Multicast, procediendo a su envío. Los receptores interesados en capturar esta trama, notifican a su nivel IP su intención de recibir datagramas dirigidos a este grupo.
2. **Los receptores se encuentran en un segmento de red diferente al emisor**. En este caso, se requiere la intervención de los *routers* que interconectan dichos segmentos para el empleo de un protocolo dinámico de encaminamiento, de forma que se permita la construcción de árboles de distribución Multicast para encaminar el tráfico a través de la red y un protocolo de anexión que permita que el router conozca los grupos existentes y los dispositivos asociados a cada una de las interfaces de red configuradas.

IGMP es el protocolo de comunicaciones utilizado por los hosts (para expresar su interés en recibir paquetes enviados a un grupo multicast IP determinado) y los routers (para gestionar los grupos). Particularmente, existen diversas versiones de IGMP:

Versión	Características
IGMPv1	⇒ RFC 1112. Primera versión convertida en estándar (1989) ⇒ Dos mensajes: la afiliación de consulta "*membership query*" y membresía de respuesta "*membership reply*". ⇒ La consulta de miembros la realizan los routers usando la dirección de grupo 224.0.0.1 y las respuestas las realizan las estaciones a la dirección de grupo que desean unirse. ⇒ Se espera que todos los routers de Multidifusión envíen consultas.

	⇒ No hay mensajes de abandono del grupo (mecanismo *time-out* en los routers para descubrir los host que ya no están interesados en ser miembros).
IGMPv2	⇒ Consulta de miembros de manera general (envío a la dirección IP 224.0.0.1) y especifica de grupo (envío a la dirección de un grupo de multidifusión concreto). ⇒ Mensajes de abandono de grupo. ⇒ Se estipula que sólo el router con la dirección IP más baja en el segmento de red será el querier y podrá proceder al envío de las consultas (el resto de routers estarán a la escucha de respuestas).
IGMPv3	Permite al host especificar las fuentes de tráfico Multicast de las que recibir datos y bloquear tráfico de otras fuentes.

Tabla 6 - Versiones IGMP[22]

Aquellos routers que estén dotados de capacidades *Multicast* pueden generar árboles de distribución que permitan controlar la ruta del tráfico *Multicast* través de la red con la finalidad de que sea entregado a todos los receptores. Para que el router pueda enviar correctamente el tráfico a través del árbol, se emplea el 'reverse path forwarding' o RPF, que consiste en usar las tablas de rutas Unicast para determinar los Routers Multicast vecinos tanto aguas arriba como abajo. Después de proceder con este paso, se reenvían los paquetes Multicast si son recibidos en la interfaz más próxima a la fuente. Se pueden diferenciar dos tipos básicos de árboles de distribución:

Source Trees: crea un árbol de distribución por cada fuente o emisor de Multicast. Presenta la ventaja de que utiliza la ruta óptima a través de la red. No obstante, al existir su raíz en la fuente y en las ramas que forman el árbol, implica un mayor consumo de recursos, concretamente en las tablas de encaminamiento de Multicast.

[22] Fuente: elaboración propia

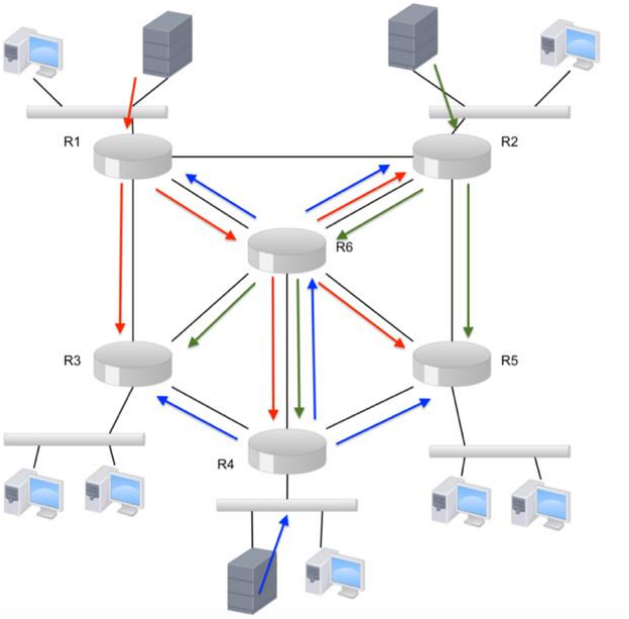


Figura 17 - Source Tree[23]

Shared Trees. A diferencia del modelo anterior, presenta una raíz común (redes-vous point o RP) y que se encuentra localizada en un punto de la red. Este modelo presenta la ventaja de que tiene un consumo de recursos menor pero los caminos entre la fuente y los receptores podrían no ser las rutas más cortas u óptimas.

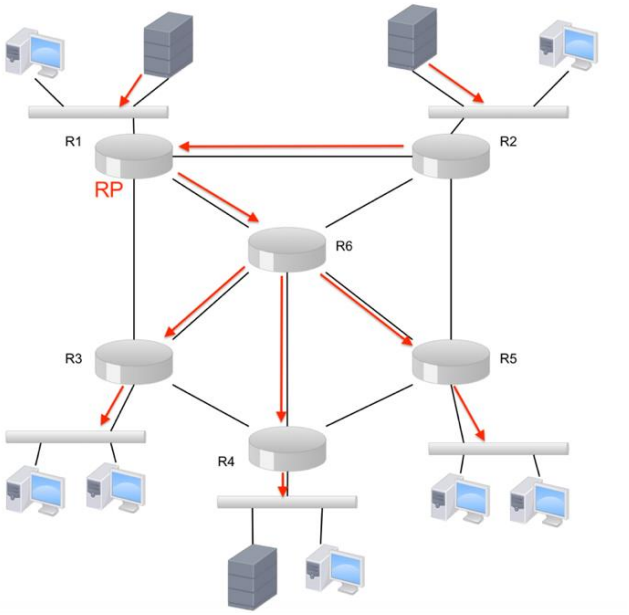


Figura 18 - Shared Trees[24]

[23] Fuente: elaboración propia
[24] Fuente: elaboración propia

Para la generar la estructura de árbol de distribución se emplea el protocolo de Multicast Independiente (PIM), que hace uso de las rutas mantenidas por protocolos de encaminamiento Unicast (estáticos y dinámicos) y utiliza la tabla de enrutamiento Unicast para llevar a cabo la función de comprobación RPF. Este protocolo tiene diferentes formatos. Los más empleados son:

1. **PIM Dense Mode (PIM-DM)**: es un protocolo adecuado en situaciones donde existan muchos receptores activos en cada segmento de la topología red dado. Genera árboles de tráfico inundando de tráfico *Multicast* hacia todos los rincones de la red. Solo admite arboles de tipo "*source trees*".
2. **PIM Sparse Mode (PIM-SM)**: distribuye información sobre las fuentes activas mediante el envío de paquetes de datos sobre el árbol compartido. Las fuentes son registradas mediante el punto de encuentro o RP y después los datos se transmiten por el árbol compartido hacia los receptores. Admite árboles de tipo "shared trees". Este será el mecanismo que utilizaremos en la simulación.

2.2.8 SEGURIDAD

La red DMVPN que se implementará, estará provista de encriptación en las transmisiones de datos mediante IPSEC. Además, se describirán otros elementos de seguridad como DMZ, tecnologías IDS/IPS, SIEM, Cisco Umbrella y herramientas de detección de vulnerabilidades informáticas, ya que son mecanismos ampliamente extendidos en la actualidad y pueden ser aplicables a líneas futuras de nuestro proyecto.

2.2.8.1 PROTOCOLO IPSEC

Dado que para establecer enlaces de comunicaciones entre puntos distantes se emplea Internet como medio de transmisión, con todos los riesgos de seguridad que ello implica, se recurre al empleo del protocolo IPSec, esto es, un estándar de origen militar que proporciona servicios de seguridad a la capa IP y a todos los protocolos

superiores basados en IP (TCP y UDP entre otros). Sus objetivos principales objetivos son satisfacer los tres requerimientos de un canal seguro:

Requerimiento	Descripción
Confidencialidad	La información puede ser vista únicamente por los interlocutores (técnicas de cifrado).
Integridad	Propiedad que garantiza que los datos transmitidos permanezcan inalterados excepto cuando sean modificados por personal autorizado y esta modificación sea registrada, asegurando su precisión y confiabilidad.
Autenticación y no repudio	El receptor tiene la garantía de que fue el emisor y no otro distinto quien envió los datos.

Tabla 7 - Requerimientos de un canal seguro[25]

Además del cumplimiento de los factores anteriores, este protocolo puede ser combinado con tecnologías de clave pública (RSA), certificados digitales X509v3, algoritmos de cifrado (DES, 3DES...) y algoritmos de hash (MD5, SHA-1). Además, es posible emplear otros algoritmos que se consideren más seguros o adecuados para un entorno en particular. Así pues, dentro de IPSec se distinguen los siguientes componentes:

⇒ Los protocolos AH y ESP, que proporcionan mecanismos de seguridad para proteger el tráfico IP.

⇒ Un protocolo de gestión de claves Internet Key Exchange (IKE) que permite a dos nodos negociar las claves y todos los parámetros necesarios para establecer una conexión AH o ESP.

2.2.8.1.1 PROTOCOLO AH

Se emplea para garantizar la integridad y la autenticación de los datagramas IP. Es decir, proporciona un medio al receptor de los paquetes IP para autenticar el origen de los datos y para verificar que dichos datos no han sido alterados en tránsito.

[25] Fuente: elaboración propia

No obstante, no proporciona ninguna garantía de confidencialidad ya que los datos transmitidos podrían ser vistos por terceros.

AH consiste en una cabecera que se introduce entre la cabecera IP estándar (IPv4 o IPv6) y los datos transportados, que pueden ser mensajes TCP, UDP o ICMP e incluso el datagrama IP completo. IANA le ha asignado el número decimal 51 y ello significa que el campo *Protocolo* de la cabecera IP contiene el valor 51, en lugar de los valores 6 ó 17, que se asocian a TCP y UDP respectivamente.

Su funcionamiento se basa en un código de autenticación de mensajes denominado HMAC, que permite aplicar una función hash a la combinación de unos datos de entrada y una clave, generando como salida una cadena de caracteres denominada extracto, similar a una huella personal asociada a los datos y a la persona que lo ha generado. Precisamente, su seguridad reside en que el cálculo del extracto (MAC) es imposible sin saber la clave y que dicha clave la únicamente la conocen el emisor y el receptor. El siguiente diagrama representa el modo de funcionamiento de dicho protocolo:

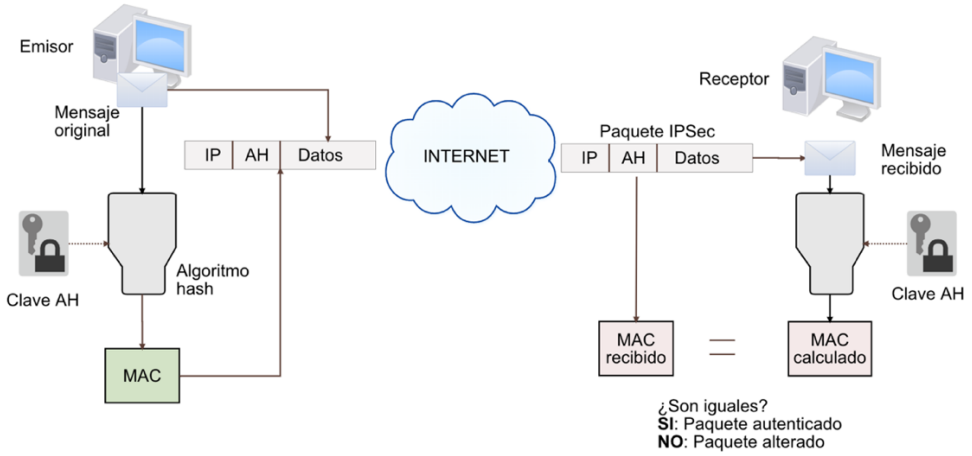


Figura 19 - Funcionamiento Protocolo AH[26]

Observamos que el emisor calcula un extracto del mensaje original, el cual se copia en uno de los campos de la cabecera AH. El paquete generado se envía a través de la red, repitiéndose en el extremo receptor el cálculo del extracto y después, se

[26] Fuente: elaboración propia

compara con el paquete recibido. Si son iguales, el receptor tiene la garantía de que el paquete no ha sido alterado durante su transmisión y que procede del origen esperado.

2.2.8.1.2 PROTOCOLO ESP

El objetivo primordial del protocolo ESP es proporcionar confidencialidad y para ello, se especifica el modo de cifrar los datos que se desean enviar y cómo este contenido cifrado se incluye en un datagrama IP. De manera adicional, puede ofrecer servicios de integridad y autenticación del origen de datos incorporando un mecanismo similar al de AH.

El formato de cabecera de ESP consta de una cabecera y una cola que rodean los datos transportados, pudiendo ser cualquier protocolo IP (TCP, UDP, ICMP o un paquete IP completo). IANA le ha designado el número decimal 50. La función de cifrado dentro del protocolo ESP es desempeñada por un algoritmo de cifrado de clave simétrica. La siguiente ilustración representa el esquema básico de este protocolo:

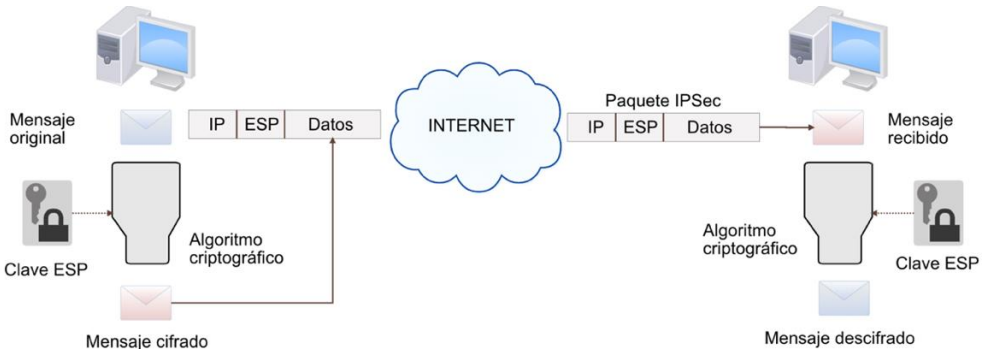


Figura 20 - Funcionamiento Protocolo ESP[27]

Como podemos apreciar, el emisor toma el mensaje original, procede a su cifrado utilizando una clave concreta y lo incluye en el paquete IP, junto con la cabecera ESP. Durante el tránsito hacia su destino, si el paquete es interceptado por un tercero, tan solo obtendría un conjunto de bits ininteligibles. Una vez llegado el paquete a su destino, se aplica, de nuevo, el algoritmo de cifrado con la misma clave, recuperando de esta manera los datos originales. Por tanto, la seguridad de este

[27] Fuente: elaboración propia

protocolo reside en la robustez del algoritmo de cifrado, así como en que la clave ESP sea conocida únicamente por el emisor y el receptor.

2.2.8.1.3 MODOS DE FUNCIONAMIENTO DE IPSEC

Según el tipo de conectividad requerida (Host to LAN o LAN to LAN), IPSec puede trabajar en dos modos distintos: transporte o túnel.

El **modo transporte** inserta la cabecera AH o ESP entre la cabecera del paquete original y el protocolo que transporta (TCP, UDP...). Las direcciones IP de los dispositivos a proteger son visibles en el paquete cifrado que se va a transmitir.

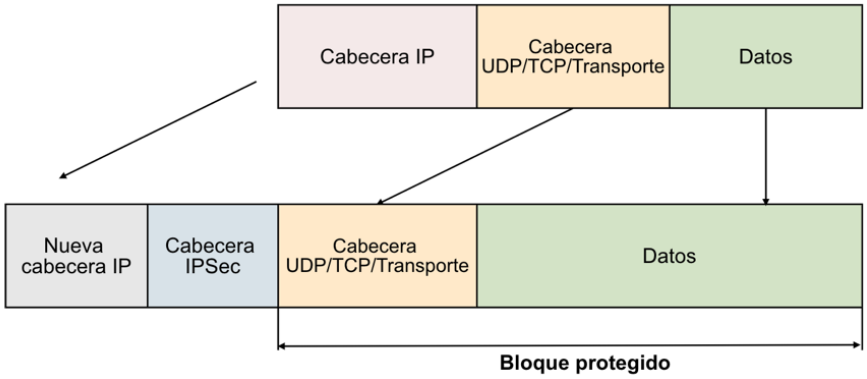


Figura 21 - Modo transporte IPSec[28]

El siguiente escenario representa un ejemplo de aplicación del modo transporte en dos hosts que entienden IPSec. La información que se protege es únicamente el protocolo TCP o UDP; así como los datos de la aplicación:

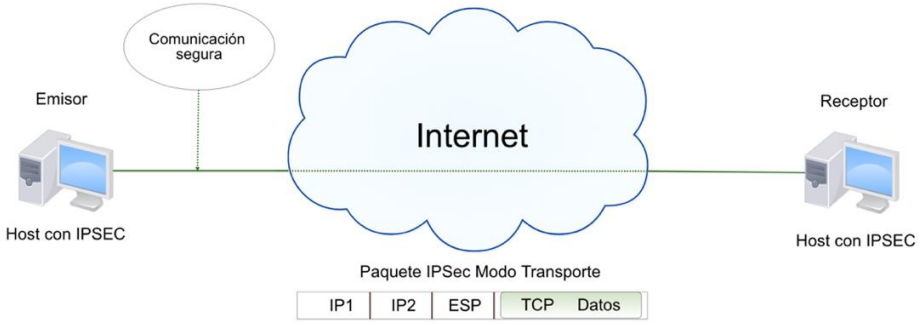


Figura 22 – Aplicación del modo transporte IPSec[29]

[28] Fuente: materiales Redes NBMA. Luis Miguel Crespo
[29] Fuente: elaboración propia

El **modo túnel** encapsula y protege un paquete IP completo incluyendo sus cabeceras. Los *routers* emplean una de sus direcciones IP encaminables para generar una nueva cabecera IP y que el paquete pueda ser enviado. Al añadir una cabecera IP, hay un aumento de información redundante de 20 bytes. Este modo puede ser empleado por tanto por AH como por ESP.

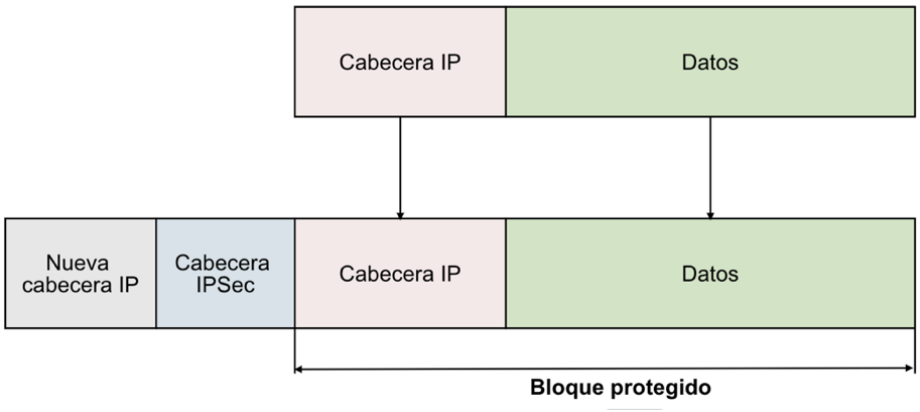


Figura 23 - Modo túnel IPSec[30]

En el siguiente diagrama se puede apreciar la aplicabilidad del modo túnel a dos redes distantes que usan para conectarse dos routers que implementan IPSec en su configuración. Sin embargo, ambos PCs envían y reciben tráfico en claro. Este esquema presenta la ventaja principal de que las funciones de seguridad se centralizan en los routers, facilitando las labores de administración y ofrecen una comunicación segura y transparente.

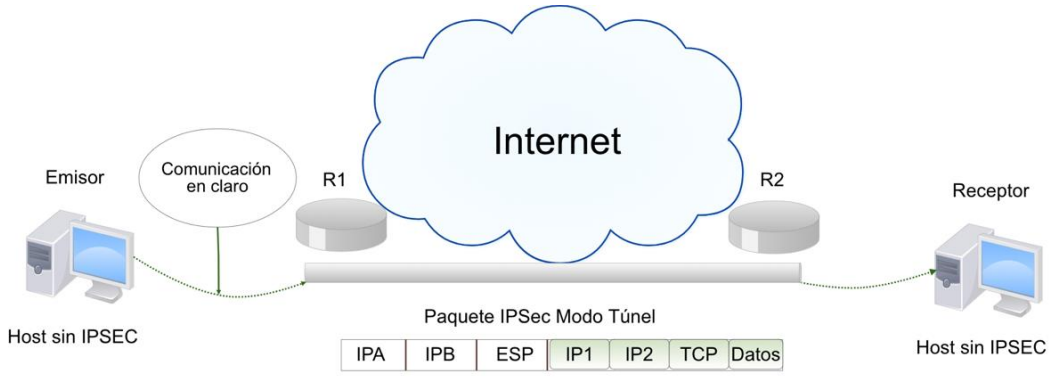


Figura 24 - Aplicación del modo túnel IPSec[31]

[30] Fuente: materiales Redes NBMA. Luis Miguel Crespo
[31] Fuente: elaboración propia

2.2.8.2 PROTOCOLO DE INTERCAMBIO DE CLAVES IKE

Una asociación de seguridad es un canal de comunicación unidireccional que conecta a dos nodos, a través del cual fluyen los datagramas protegidos mediante los mecanismos criptográficos acordados previamente. Este acuerdo incluye el nivel de fortaleza y el tipo de cifrado utilizado para la protección de los datos. La SA también incluye el método y la dureza de la autenticación de los datos; así como el mecanismo utilizado para crear nuevas claves para esa protección de datos. Al identificar únicamente un canal unidireccional, una conexión IPSec se compone de dos SAs, uno por cada sentido de la comunicación.

En una comunicación IPSec, se requiere que ambos nodos estén de acuerdo tanto en los algoritmos criptográficos como en los parámetros de control. Dicha operación se puede realizar de manera manual o utilizando un protocolo de control que tenga la labor de negociar automáticamente los parámetros requeridos (longitud de las claves, los métodos hash, servicios anti repetición…) y realizar cambios periódicos en las claves utilizadas para que el canal de comunicación no sea susceptible de sufrir ataques de descifrado por fuerza bruta. En esta línea, el IETF ha definido el protocolo **IKE** para implementar una funcionalidad de gestión automática de claves y establecimiento de las SAs.

El protocolo IKE es fruto de la integración de otros dos protocolos complementarios: ISAKMP y *Oakley*. Brevemente descrito, ISAKMP define de manera general el protocolo de comunicación y sintaxis de los mensajes que se utilizan en IKE; mientras que *Oakley* especifica la lógica de cómo se realiza de forma segura el intercambio de una clave entre dos partes que sin conocimiento previo. Por tanto, el objetivo de IKE consiste en establecer una conexión cifrada y autenticada entre dos entidades, a través del cual se negocian los parámetros necesarios para establecer una asociación de seguridad IPSec. Esta negociación se lleva a cabo en dos fases:

2.2.8.2.1 FASE 1

Es la negociación inicial de una SA bidireccional de ISAKMP entre dos extremos, también denominada modo principal o '*main mode*'.

En esta fase, ambos nodos de la aplicación establecen un canal seguro y autenticado. El canal seguro se logra mediante la aplicación de un algoritmo de cifrado simétrico y un algoritmo HMAC. Las claves requeridas se derivan de una clave maestra que se obtiene mediante un algoritmo de intercambio de claves *Diffie-Hellman*.

La negociación de claves *Diffie-Hellman*, se caracteriza por ser un mecanismo de cifrado de clave pública que permite que dos interlocutores obtengan una clave compartida sólo conocida por ellos, sobre un medio inseguro. Con dicho algoritmo, cada extremo genera una clave pública y su correspondiente clave privada (RSA). Con RSA, aquello que se cifra con la clave pública sólo puede ser descifrado con la privada y viceversa. Las claves públicas de ambos extremos son intercambiadas, así como la elección de un número primo y una base. Luego, cada extremo aplica el algoritmo y ambos obtienen la misma clave. Dicha clave ahora compartida, será la que se usará para el cifrado.

Por otro lado, para existen dos maneras distintas para garantizar la identidad de los nodos. El primer método se basa en el conocimiento de un secreto compartido (cadena de caracteres que sólo conocen los dos extremos que quiere establecer una comunicación IPSec). Usando funciones hash, cada extremo demuestra a su opuesto que conoce el secreto sin revelar su valor; de esta forma, los dos nodos se autentican mutúamente. Para que haya más seguridad, se ha de configurar un secreto diferente para cada par de nodos, por lo que el número de nodos aumentaría rápidamente cuando la cantidad de nodos se viese incrementado. Por este mismo motivo, en entornos en los que se desea interconectar diversos nodos IPSec, la gestión de claves resulta un tanto complicada. En esta situación, se recomienda la autenticación basada en certificados digitales.

El segundo método consta del uso de certificados digitales, que permiten distribuir de manera segura la clave pública de cada nodo, de forma que éste pueda testear su

identidad mediante la posesión de la clave privada y ciertas operaciones de criptografía pública. El empleo de certificados digitales requerirá un elemento más en la arquitectura: la infraestructura de clave pública o PKI.

2.2.8.2.2 FASE 2

En esta fase, denominada también modo rápido o "*quick mode*", las SA de IPSEC son negociadas mediante el proceso IKE utilizando la SA bidireccional ISAKMP previamente generada.

Concretamente, se negocian las características de la conexión ESP o AH; así como otros parámetros requeridos. El equipo inició la comunicación ofrecerá todas las posibles opciones que tenga configuradas en su política de seguridad y con la prioridad que se hayan configurado. El receptor aceptará la primera que coincida con los parámetros de seguridad que tenga definidos. Además, ambos nodos se informan del tráfico que van a intercambiarse a través de dicha conexión.

El siguiente diagrama muestra el funcionamiento de IKE y el modo de obtención de una clave de sesión, empleada para proteger a las conexiones ESP o AH:

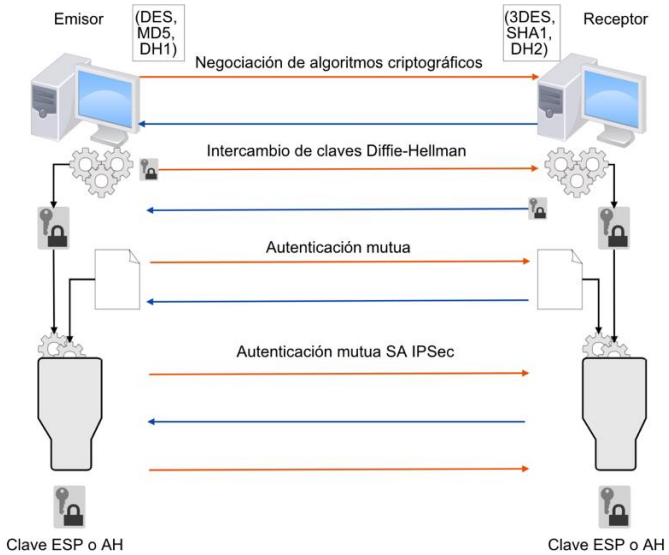


Figura 25 - Funcionamiento IKEv1[32]

[32] Fuente: elaboración propia

2.2.8.3 ZONA DESMILITARIZADA

Existen tres formas de instalación y mantenimiento de servidores locales de cualquier organización que pueden ser accesibles desde Internet. En primer lugar, muchas compañías optan por tener servidores instalados fuera de la red mediante la asignación de una dirección IP pública directa, tal y como se muestra en la siguiente ilustración:

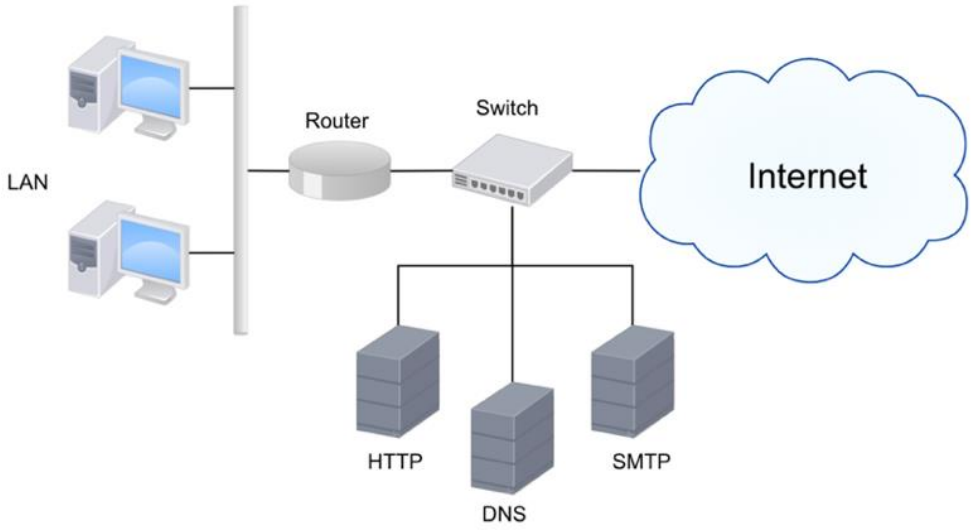


Figura 26 - Servidor ubicado fuera de la red[33]

Las ventajas que presenta este modelo son su sencilla configuración, ya que, dentro del rango disponible, la dirección IP correspondiente se configura en el *router* que hará NAT en toda la red LAN y el resto de las direcciones IP disponibles se asignarían a los servidores. Además, requiere implementar un cortafuegos físico y, en términos de seguridad, si un servidor sufre un ataque, la red LAN no estará comprometida. No obstante, presenta el inconveniente que se ha requerir un firewall de software para cada servidor (*iptables* en Linux) y, por tanto, se requeriría la consumición de muchos recursos. Además, este modelo no aporta escalabilidad y presenta una administración dificultosa.

En vista de las desventajas citadas, se podría mantener la seguridad centralizada en un *firewall* físico, pero para ello se requeriría desplazar los servidores

[33] Fuente: elaboración propia

dentro de la LAN. En este caso, la idea es aportar una protección a los servidores mediante un dispositivo centralizado (*router* o firewalls con funciones NAT):

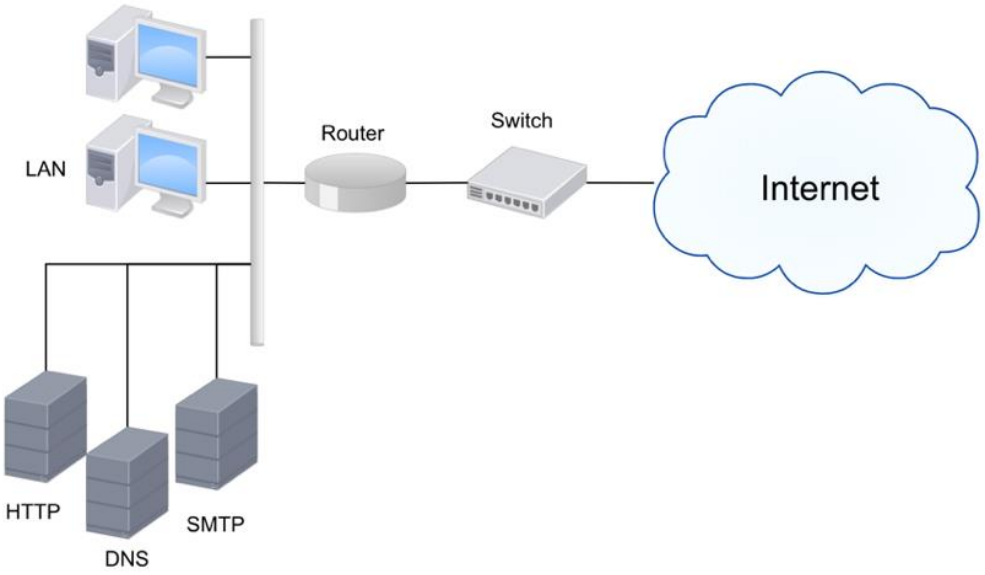


Figura 27 - Servidor ubicado dentro de la red[34]

Este modelo aporta facilidad de administración de seguridad y movilidad de las direcciones IPs públicas; así como el acceso directo por parte de los usuarios a los diferentes servicios establecidos. No obstante, si existe alguna intrusión a algún servidor, la LAN interna sería también vulnerable.

Una solución al conjunto de desventajas anteriores, sería la configuración de una zona desmilitarizada o DMZ, que se caracteriza por ser una subred independiente, separada de la red LAN y de Internet (conocida también como "*outside*"). Al establecer una DMZ en la red se puede configurar el firewall para crear reglas específicas de seguridad con listas de acceso y NAT que permitan el tráfico procedente de Internet únicamente a la DMZ. El NAT estático estará asociado entre las direcciones IPS públicas y las IPS asignadas a cada servidor de la DMZ. Así, si una máquina resulta ser comprometida en la DMZ, no tendría acceso a la red LAN corporativa:

[34] Fuente: elaboración propia

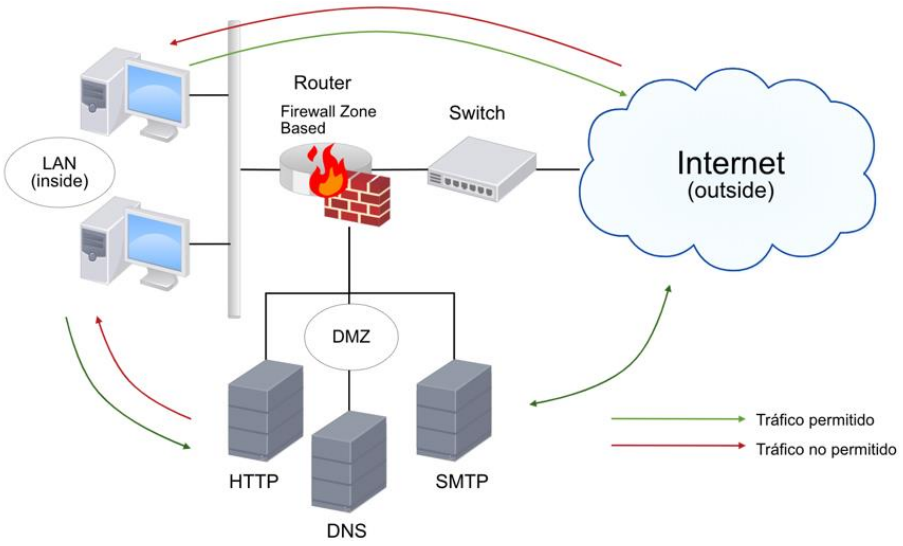


Figura 28 - DMZ[35]

Para llevar a cabo tal cometido, es necesario crear las reglas de tráfico adecuadas y definir los perfiles de seguridad entre las zonas *outside*, LAN (o *inside*) y DMZ en el Firewall. La siguiente imagen refleja las políticas básicas aplicables:

Origen	Destino	Política
Outside	DMZ	Permitido
Outside	Inside	Denegado
DMZ	Outside	Permitido
DMZ	Inside	Denegado
Inside	Outside	Permitido
Inside	DMZ	Permitido

Tabla 8 - Políticas aplicables a una DMZ[36]

2.2.8.4 IDS

Se trata de un proceso de detección y monitorización de eventos que suceden en una red, que realiza dos tareas fundamentales. En primer lugar, realiza acciones de **prevención**, que se llevan a cabo empleando herramientas que escuchan el tráfico en la red o en un ordenador, denominados sensores, e identifican los ataques aplicando reglas, reconocimiento de patrones o técnicas inteligentes. Por otra parte, adopta

[35] Fuente: elaboración propia
[36] Fuente: elaboración propia

labores de **reacción**, es decir, trata de detectar patrones de intrusión en las trazas de los servicios de red o en el comportamiento del sistema.

Para detectar intrusiones en el sistema, los IDS emplean tres tipos de datos: configuraciones actuales de los sistemas, un histórico de eventos y procesos activos del sistema o reglas.

2.2.8.5 IPS

Mientras el IDS se limita a detectar y notificar la intrusión al administrador del sistema, y éste se encarga de recibir y responder las alertas; el IPS detecta la intrusión y la detiene de algún modo ya predefinido, comprobando ciertos comportamientos en la red previamente configurados como anómalos. Un inconveniente de los IPS deriva de la reacción proactiva ante las intrusiones, ya que puede provocar efectos inesperados cuando éste reacciona ante un falso positivo, lo que podría llevar a un aislamiento de la máquina en cuestión o a denegaciones de servicio.

Los IPS basan sus decisiones teniendo en cuenta los encabezamos como en el contenido de datos del paquete. Otras características de los IPS son la capacidad de reacción automática ante incidentes, aplicación de filtros, bloqueo automático frente a ataques efectuados en tiempo real, disminución de falsas alarmas de ataques en la red, protección de sistemas no parcheados y optimización en el rendimiento del tráfico de la red.

Entre las herramientas IDS/IPS que se encuentran ampliamente utilizadas en el panorama actual, podríamos mencionar a `Snort` y `Suricata`.

2.2.8.6 SIEM

Los diferentes elementos que conforman cualquier red de comunicaciones generan eventos. En el contexto de la informática, un evento es un registro temporal que se genera cuando ha sucedido algo (*logs*) y nos pueden aportar información bastante significante sobre lo que está ocurriendo en la red: errores producidos en los

propios dispositivos, cambios de configuraciones, eventos generados por los servidores web cuando un cliente accede a algún servidor web… En organizaciones de gran tamaño, se puede generar gran cantidad de información y su análisis y seguimiento puede resultar bastante tedioso.

Una solución SIEM implica la combinación de dos tecnologías: el SEM (se ocupa de la monitorización en tiempo real, correlación de eventos, notificaciones y vistas de la consola) y el SIM (proporciona almacenamiento a largo plazo, análisis y comunicación de datos de los eventos detectados). Por tanto, el término SIEM describe múltiples capacidades como la recopilación, análisis, presentación de información de la red y los dispositivos de seguridad. Además, es capaz de procesar los datos obtenidos para estandarizarlos, llevar a cabo un análisis de dichos datos, generar alertas cuando detecta una actividad anómala, informes e incluso pueden ofrecer capacidades de bloqueo de actividades maliciosas.

La instalación de esta herramienta supone una profunda planificación previa, identificación de los requisitos y tener un buen conocimiento de cómo está estructurada la red para identificar los activos más críticos del sistema y estudiar su protección. Existen varios motivos que pueden conducir a una entidad la inclusión de un SIEM: cumplimiento de una normativa (ISO 27.001 o Esquema Nacional de Seguridad), haber sido víctima de alguna violación de seguridad e incluso disponer de un cierto nivel de seguridad antes de firmar un contrato.

En la actualidad, existen diversos tipos de compañías que ofrecen tecnologías SIEM. Podemos hacer mención al dispositivo USM *Appliance* de AlienVault:

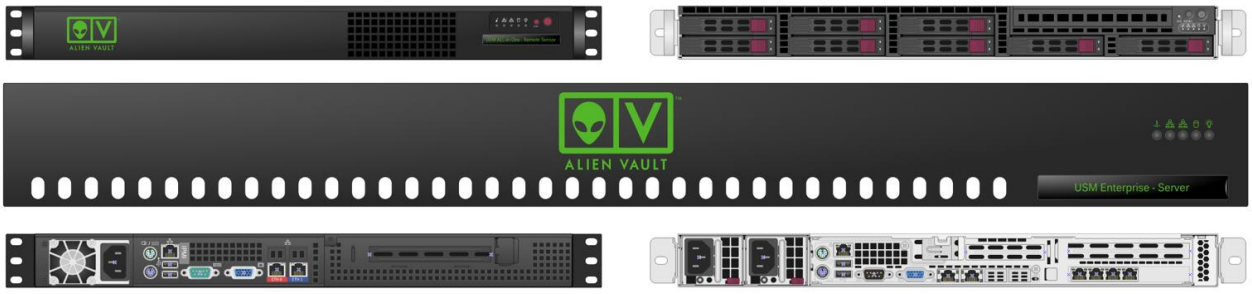


Figura 29 - Appliance AplienVault[37]

[37] Fuente: Sitio Web AlienVault: https://www.alienvault.com

2.2.8.7 CISCO UMBRELLA

Durante diversos años (y hasta la fecha), las redes de comunicaciones han seguido el esquema de la seguridad perimetral, compuesta por una arquitectura y elementos de red que proveen de seguridad al perímetro de una red interna frente a otra que generalmente es Internet.

El perímetro permite definir qué componentes se encuentran dentro de la red corporativa, qué está fuera de la misma y cuáles son las políticas de acceso desde y hacia otro sector. Sin embargo, este concepto en la actualidad es tan sencillo como obsoleto. Es un hecho que en la actualidad hay más usuarios móviles y ordenadores portátiles con información confidencial de la entidad, que pueden acceder a Internet desde otras redes, lo que se significa que los usuarios no necesariamente han de estar conectados a la red corporativa para trabajar. Cuando un usuario se encuentra fuera del perímetro de red, es más vulnerable y, además, la organización no tendría visibilidad ni protección y podría darse la situación de infección por *malware*.

Cisco Umbrella es una tecnología hospedada en la nube de pago que ofrece una línea adicional de protección frente a las amenazas de Internet, sin importar donde se ubique el usuario, utilizando la resolución de nombres de dominio OpenDNS. Cuando la plataforma recibe una solicitud DNS por parte de un usuario, utiliza inteligencia para determinar si es segura. Las que se consideran peligrosas son dirigidas a un proxy de Umbrella, que usa la reputación web de Cisco Talos (plataforma que mantiene una red de detección de amenazas mediante técnicas de detección y prevención) e información de terceros para determinar si dicha URL es peligrosa.

Además, el proxy también inspecciona los archivos que se intentan descargar desde esos sitios usando motores antivirus así como el módulo de seguridad Cisco AMP, que se encarga de detectar y bloquear los ataques ofreciendo una protección proactiva que minimice las amenazas. A partir del resultado de la inspección, se realiza o se bloquea la conexión al usuario.

Para aplicar la seguridad que ofrece Cisco Umbrella en todos los dispositivos que albergan información privada de la entidad, habría que incluir el servidor DNS de Umbrella en los archivos de configuración adecuado según el SO.

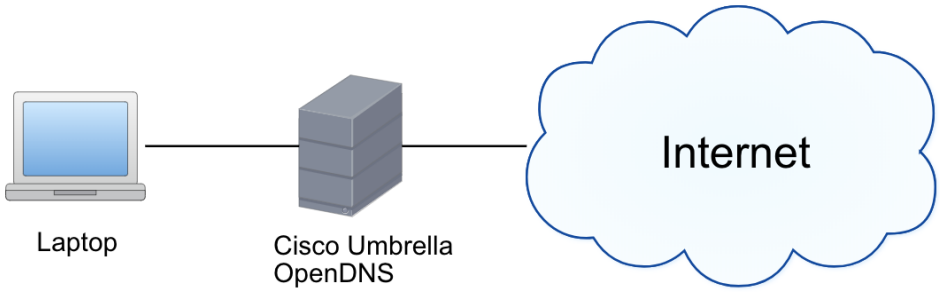


Figura 30 - Cisco Umbrella[38]

La herramienta ofrece, también, una interfaz gráfica de usuario, donde determinados empleados de la organización podrán consultar la cantidad de bloqueos y censar la navegación por Internet de los empleados.

2.2.8.8 HERRAMIENTAS DE DETECCIÓN DE VULNERABILIDADES

La mayoría de las organizaciones ofrecen servicios web expuestos en Internet con una funcionalidad determinada dependiendo del sector. Los programadores, suelen emplear metodologías ágiles y entorno de pruebas para el correcto desarrollo de las webs o aplicaciones para su posterior puesta en producción. Ahora bien, la exposición de los servicios a Internet tiene sus riesgos ya que, normalmente, existen vulnerabilidades software que un atacante podría aprovechar para su explotación y de ahí comprometer otras máquinas que se encuentren en el mismo segmento de red. Por ello, disponer de programas que faciliten el hallazgo de brechas de seguridad en los servicios será crucial para que el sistema informático funcione según lo esperado y no haya ningún host o servicio comprometido. En esta sección se detallaran dos de las herramientas de escaneo de vulnerabilidades más populares que existen en la actualidad: Nessus y Acunetix Web Scanner.

[38] Fuente: elaboración propia

2.2.8.9 NESSUS

Nessus es una herramienta de seguridad creada por la compañía Tenable® y permite definir un listado de direcciones IPs, direcciones web e incluso direcciones IP de las interfaces de los routers y switches para detectar brechas de seguridad.

En operación normal, Nessus realiza un escaneo de puertos para detectar cuáles están abiertos y después utilizar varios *exploits* para atacarlo. Un *exploit* es una secuencia de comandos que pretende aprovechar vulnerabilidades de seguridad para conseguir un comportamiento no deseado en el servicio atacado.

Una vez finalizados los escaneos a las direcciones web especificadas, los resultados son mostrados en una interfaz web, indicando el problema detectado, la forma de mitigar el problema y otros sitios webs para la consulta de más detalles:

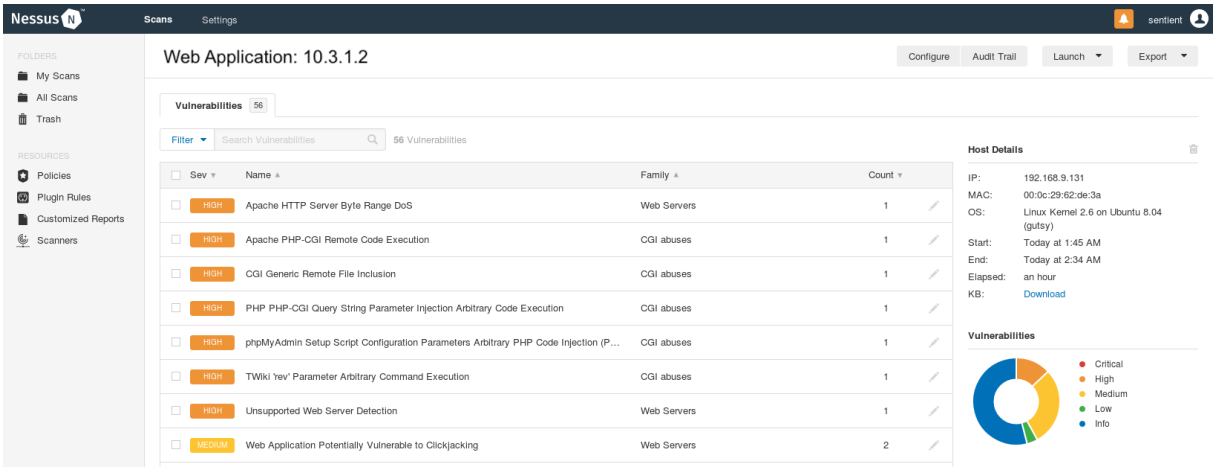


Figura 31 - GUI Nessus[39]

2.2.8.10 ACUNETIX WEB VULNERABILITY SCANNER

Se trata de una herramienta que es capaz de auditar sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en Internet a nivel de programación del sitio web y en la configuración del servidor. Del mismo modo que Nessus, los escáneres son configurables a través de una interfaz web y proporciona una descripción detallada de las vulnerabilidades encontradas, su criticidad, así los pasos necesarios para su correcta mitigación:

[39] Fuente: sitio web Nessus: https://es-la.tenable.com

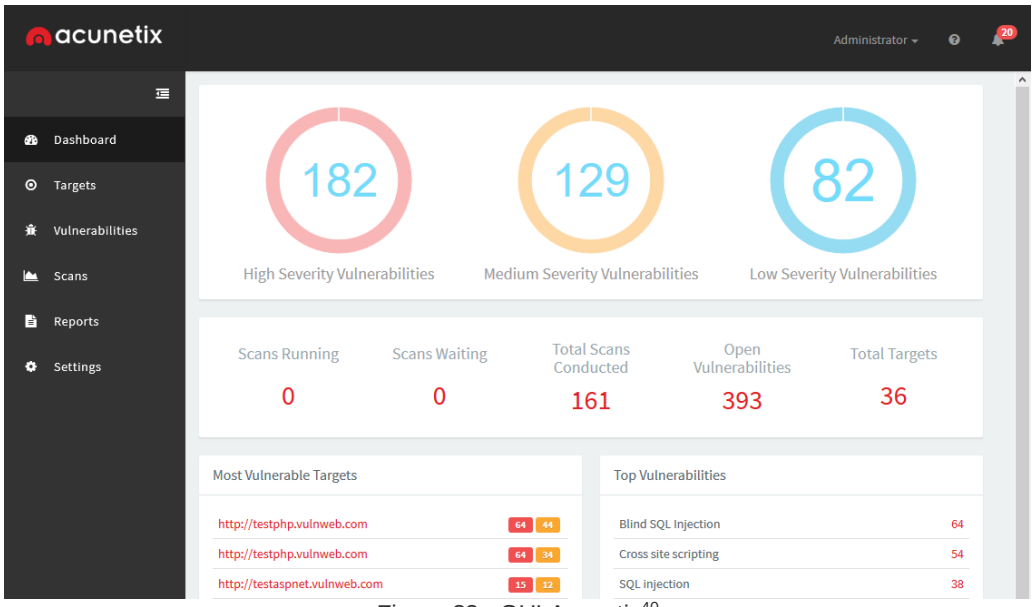


Figura 32 - GUI Acunetix[40]

2.2.9 HERRAMIENTAS DE MONITORIZACIÓN

Las organizaciones se apoyan cada vez más en aplicaciones, y en sus correspondientes infraestructuras TIC para obtener más eficiencia en sus operaciones y poder satisfacer las necesidades del cliente. Esto implica la necesidad de controlar el desarrollo de cualquier acción o suceso que afecte a las redes, servidores y aplicaciones para poder conocer que todo está funcionando según lo esperado y actuar en consecuencia. Es en este punto donde entra en juego las herramientas de monitorización, clasificadas en función según su modelo de distribución:

Modelo de distribución	Características
SaaS	⇒ El cliente tiene acceso vía Internet al software de monitorización, alojado en servidores de terceros, mediante una suscripción al servicio. ⇒ Pago de pequeñas cantidades para operaciones mediante suscripciones. ⇒ **Ejemplo**: LogicMonitor.

[40] Fuente: sitio web Acunetix https://www.acunetix.com

Código abierto	⇒ Gratis y funcionalidades de personalización. Sin embargo, carecen de servicio profesional y aquellas entidades que lo deseen instalar deberán estar capacitadas para su asistencia. ⇒ **Ejemplo**: *Nagios* y *ntop*.
Propietario	⇒ Dispone de asistencia posventa y ofrece formación de calidad. ⇒ Precio de adquisición elevado ya que incluye soporte, correcciones de errores, actualizaciones y documentación según la versión empleada. ⇒ **Ejemplo**: Paessler.

Figura 33 - Modelos de herramientas de monitorización[41]

En este documento profundizaremos en Nagios y Ntop, ya que serán las herramientas de monitorización configuradas en la simulación de la red de comunicaciones.

2.2.9.1 NAGIOS

Nagios es una aplicación cliente-servidor que permite monitorizar el funcionamiento de una serie de equipos de la red. Se encarga, pues, de obtener datos de los hosts y sus servicios, indicando si están activos o no. Esta monitorización se puede efectuar directamente (chequeando los puertos); o bien mediante la comunicación con una aplicación existente en equipos remotos, del que puede obtener mucha más información del mismo (procesos, memoria, carga, etc.) utilizando un software denominado *'nrpe'* en cada equipo.

Para mostrar los resultados de la monitorización, Nagios utiliza una aplicación web sobre Apache, que es capaz de acceder a los datos obtenidos, desde las que se puede ver el panorama actual de los sistemas monitorizados (también de los *logs* en los que se guarda cualquier cambio de estado en los equipos). Los equipos recogen los datos mediante unos scripts (escritos en *shellscript*, *perl* o *C*) llamados *'plugins'*. Además, Nagios presenta funcionalidades adicionales para liberar a los administradores de chequeos periódicos de servicios críticos, monitorización

[41] Fuente: elaboración propia

distribuida, generación informes detallados, alertas y ejecución de acciones ante determinadas circunstancias que ocurren en la red (envío de correos electrónicos, *SMS* y alertas sonoras).

Es la solución idónea para organizaciones dispuestas a invertir tiempo y esfuerzo en desarrollar una solución de monitorización interna con un elevado grado de personalización.

2.2.9.2 NTOP

Permite monitorizar en tiempo real los usuarios y aplicaciones que están consumiendo recursos de red en un instante concreto y ver configuraciones erróneas en algún equipo en concreto. Además, podemos supervisar la red y crear gráficos HTML. Una vez instalado accederemos a los datos a través del navegador web, por el puerto 3000.

2.2.10 NETFLOW

Es un protocolo de monitorización de tráfico de red desarrollado por Cisco. A diferencia de la monitorización de redes activas (`traceroute` o `ping`) donde se inyecta tráfico adicional en la red para obtener mediciones; Netflow agrega los paquetes de información en flujos, que son exportados para su almacenamiento y análisis.

Un flujo o Flow es un conjunto de paquetes que cumplen unas características comunes: dirección IP origen, dirección IP destino, puerto origen, puerto destino, tipo de protocolo IP, interfaz del router o switch y tipo de servicio IP.

El funcionamiento de Netflow se basa en las siguientes fases:

1. **Captura de paquetes**: Los paquetes que circulan por los routers y switches donde está activado Netflow, son observados y capturados.
2. **Medición y exportación del flujo**: Los paquetes se introducen en Flujos que son registrados en una base de datos denominada Flow Cache. Cunado el fjujo ha acabado, se exporta un registro a los colectores Netflow, que se encargan de su almacenamiento y procesamiento.

3. **Recopilado de información**: Los datos son guardados y pre-procesados (compresión de datos, agregación, generación de resúmenes, filtrado…)
4. **Supervisión de los datos**: Detección de anomalías, búsqueda de datos, informes y alertas, correlación y agregación…

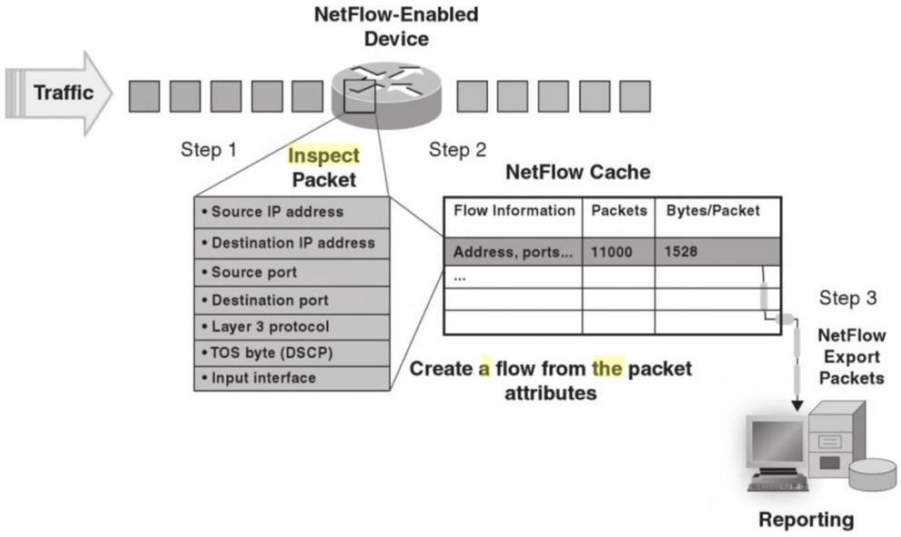


Figura 34 - Funcionamiento Netflow[42]

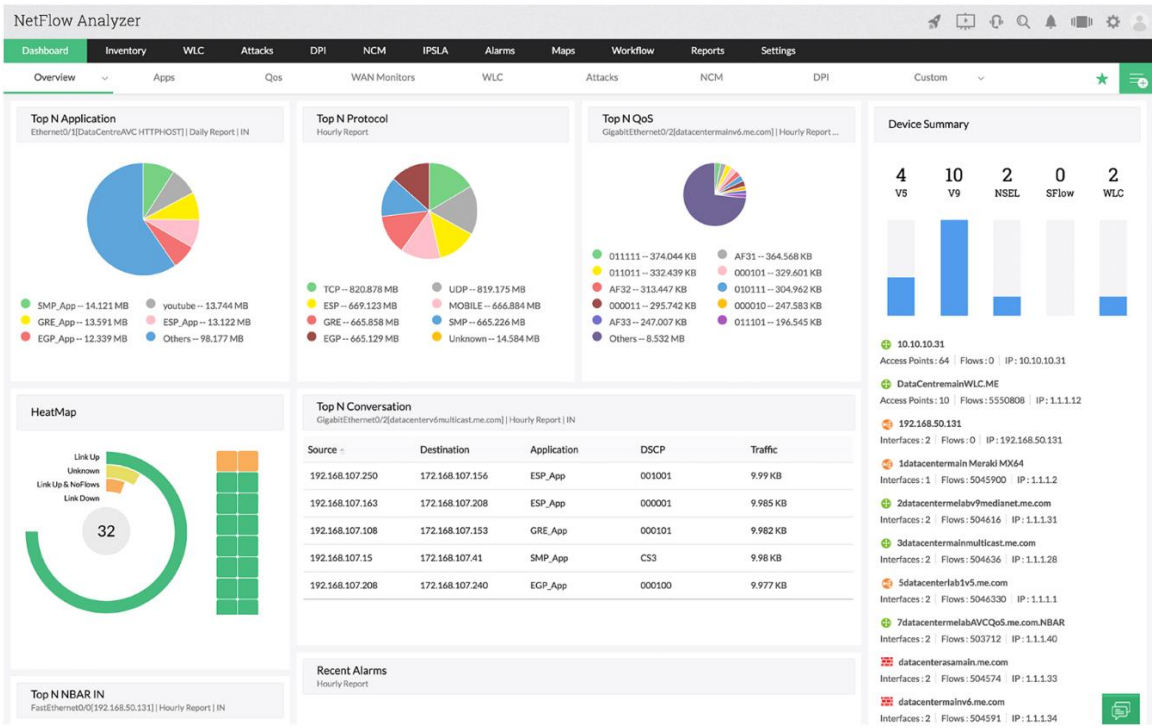


Figura 35 - Reporting. NetFlow Analyzer[43]

[42] Fuente: Joseph, V., & Chapman, B. (2009). *Deploying QoS for Cisco IP and Next Generation*
[43] Fuente: sitio web ManageEngine NetFlow Analyzer: https://www.manageengine.com/products/netflow/

3. OBJETIVOS

3.1 OBJETIVO GENERAL

El objetivo general del presente trabajo es la creación de una red de comunicaciones virtual aplicable a una entidad con 2 centros de cálculo y 200 sedes, empleando la tecnología DMVPN, junto con componentes que proporcionen alta disponibilidad, seguridad, calidad de servicio y gestión de red. Para ello, se utilizará el programa de simulación GNS3, con la imagen virtual de router de la serie 7200, concretamente, la **`c7200-adventerprisek9-mz.152-4.S7.bin`**.

3.2 OBJETIVOS ESPECÍFICOS

3.2.1 ALTA DISPONIBILIDAD

La red utilizará 2 NBMA con tecnologías WAN de diferentes operadores, junto con un protocolo de routing de convergencia rápida (EIGRP). Además, el centro de respaldo dispondrá de dos niveles de direccionamiento IP: uno para la sincronización y otro similar al centro de respaldo pero de menor coste. Finalmente, la red deberá ser capaz de encaminar tráfico *Multicast* utilizando la tecnología PIM sparse-mode para la transmisión de sesiones de vídeo.

3.2.2 SEGURIDAD

Debido a la naturaleza de Internet en cuanto a seguridad, se requerirá una capa adicional que proporcione confidencialidad, autenticidad y no-repudio mediante la aplicación de IPSec con IKEv2 como tecnología de cifrado y mGRE con NHRP para la tunelización (túnel IPSec). Asimismo, los algoritmos empleados en la red, cumplirán con las recomendaciones del centro criptológico nacional (CCN-CERT).

3.2.3 CALIDAD DE SERVICIO

La red tendrá políticas de calidad de servicio eficiente que permitirán el reparto de latencias y caudales entre las diferentes aplicaciones que la utilizan: flujos RTP para las llamadas de voz entre los usuarios, transmisiones de video para sesiones de formación, flujos de Oracle entre usuarios y servidores ubicados en el CPD, flujos HTTP/HTTPS entre usuarios y servidores ubicados en el CPD y flujos masivos restantes: correo, transferencia de archivos SMB, FTP…

3.2.4 GESTIÓN DE RED

Por último, se dispondrán de herramientas de monitorización como Nagios Core para controlar en todo momento el estado de la red y obtener tanto informes como reportes de disponibilidad.

3.3 TOPOLOGÍA DE RED EN GNS3

La siguiente imagen refleja el Piloto de red que implementado en GNS3:

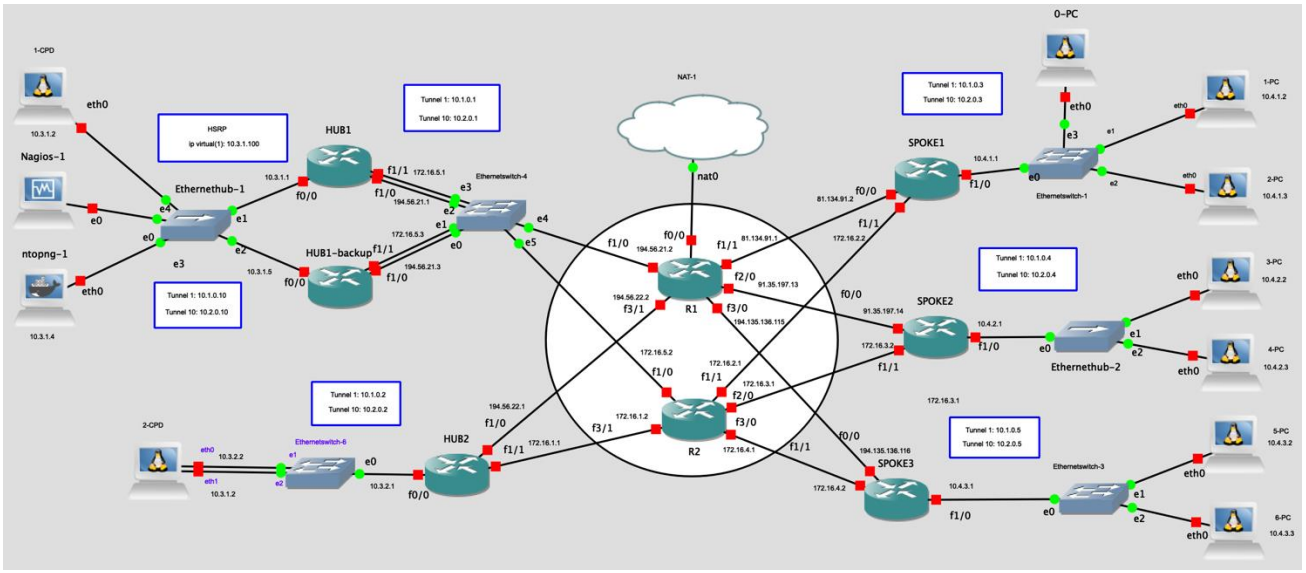


Figura 36 - Topología de red en GNS3[44]

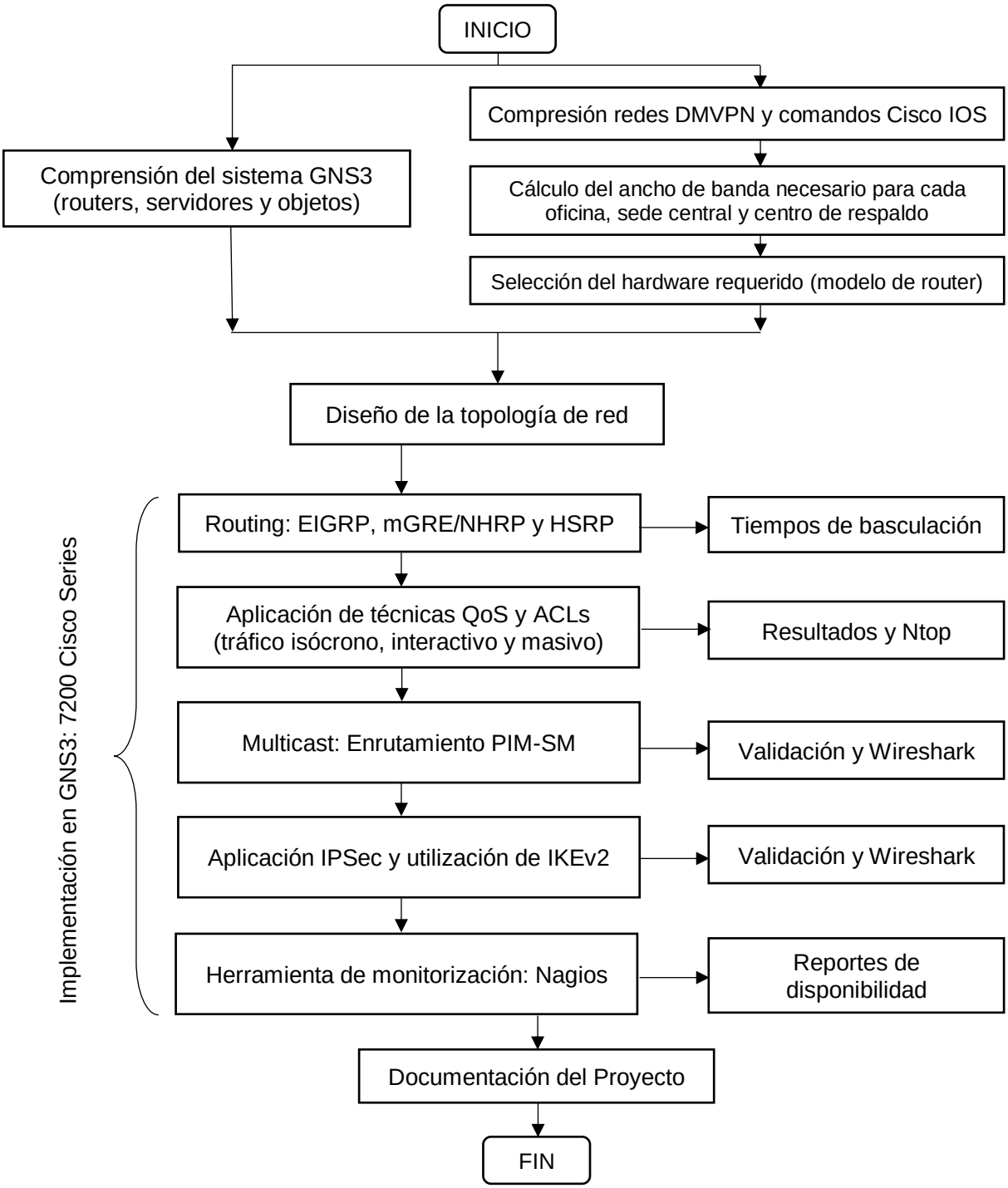
[44] Fuente: elaboración propia

Concretamente, dispondremos de una sede central en donde se ubicarán los servidores y las bases de datos, la herramienta de monitorización Nagios y un generador de stream de vídeo. Adicionalmente, se utilizarán dos routers (uno principal y otro de respaldo).

Por otro lado se utilizará un segundo HUB donde se alojarán las copias de bases de datos y servidores de la sede central, en caso de que ésta se quede sin disponibilidad. Finalmente, las oficinas estarán divididas en tres tipos diferentes, dependiendo del número de operarios existentes en gran tamaño (SPOKE A), tamaño mediano (SPOKE B) y pequeño tamaño (SPOKE C). Teniendo en cuenta esta situación, se hará un estudio previo del hardware adecuado (modelo de router Cisco) para cada tipo de oficina en función del ancho de banda consumido por los usuarios.

4. METODOLOGÍA

La metodología se basa en el desarrollo y seguimiento de un diagrama de bloques general para su posterior desarrollo de las partes implicadas:



Como podemos apreciar en el diagrama anterior, en primer lugar, realizaremos un estudio previo de las redes DMVPN y todos los protocolos y tecnologías involucradas. Así mismo, efectuaremos un estudio del ancho de banda requerido para cada tipo de oficina en función de la cantidad de usuarios, el tráfico promedio que podría consumir un trabajador y un factor de simultaneidad basado en pruebas empíricas y, a partir de estos datos, procederemos con la elección de un router físico (aquéllos que utilizan la versión de Cisco XE) para cada hub y oficina, para aproximar nuestro proyecto a las necesidades de una entidad real.

Paralelamente, se efectuará la comprensión e instalación de GNS3 en el ordenador; así como la configuración del componente *Dynamips*, que es un emulador de routers Cisco que permite ejecutar series de routers como 1700, 2600, 3600, 3700 y 7200, entre muchos otros. En este proyecto, escogeremos la imagen virtual 7200 por motivos de rendimiento y consumo de CPU.

Hechos estos pasos, diseñaremos la topología de red, de la cual nos basaremos para aplicar en GNS3 los conceptos teóricos estudiados previamente. Dividiremos la solución en cinco partes o módulos: Routing, QoS, Multicast, IPSec y Nagios. Tras la finalización de cada módulo, se validará su funcionamiento: tiempos de basculación o conmutación para comprobar la alta disponibilidad de la red, se utilizarán algunas herramientas integradas en GNS3, como el analizador de protocolos en red Wireshark o `tcpdump` para analizar los mensajes generados, se ejecutarán scripts o programas para la generación de tráfico u otras acciones y se crearán reportes de disponibilidad de la red mediante Nagios Core.

Una vez comprobada la validez de los distintos módulos que componen la solución, se procederá a la documentación del proyecto con los apartados oportunos y a la exposición del mismo.

5. DESPLIEGUE DEL PILOTO

El cuerpo del trabajo está dividido del siguiente modo: el cálculo de ancho de banda requerido para cada sede y oficina, el modelo de router Cisco necesario en función de los cálculos previos, el desarrollo y validación de los diferentes bloques que componen la solución (Routing, Calidad de Servicio, Multicast IP, IPSec y Nagios).

5.1 CÁLCULO DEL ANCHO DE BANDA PARA HUBS Y SPOKES

Antes de proceder con la implementación de la red de comunicaciones en GNS3, debemos tener en consideración el ancho de banda necesario para cada sede. Los valores obtenidos en este apartado serán utilizados en el *Bloque 2: Calidad de Servicio (QoS)* y nos ayudarán a escoger qué plataforma Cisco es la más adecuada para su aplicación a una entidad con necesidades de interconexión reales.

En este proyecto se han clasificado las oficinas en tres tipos distintos en función del número de trabajadores:

Tipo de oficina	Tamaño	Número de trabajadores
Tipo A	Grande	50
Tipo B	Mediano	20
Tipo C	Pequeño	8

Tabla 9 – Clasificación de oficinas[45]

A pesar de que en la simulación se hayan incluido únicamente tres oficinas o *spokes* por motivos de rendimiento y consumo de la CPU, la red tendrá 200 sedes remotas atendiendo a los criterios de clasificación indicados en la tabla anterior.

Por otro lado, suponemos que el ancho de banda consumido por cada trabajador, en términos generales, es de 704 kbps, obtenido de manera y teniendo en cuenta tres tipos de tráfico diferenciados entre sí: el tráfico isócrono, interactivo y masivo. Por lo que hace al primer tipo de tráfico, se ha tenido presente dos tipos de CÓDEC: G.711

[45] Fuente: elaboración propia

y G.729. Un CÓDEC es un dispositivo hardware capaz de codificar o decodificar una señal o flujo de datos digitales, utilizados ampliamente en videoconferencias o telefonía. El codificador G.711 proporciona un flujo de datos de 32 kbps; mientras que el codificador G.729 ofrece 64 kbps. Además, existe un factor de simultaneidad (fs), obtenido, también, empíricamente, cuyo valor es de 0,45, que representará el número de usuarios que pueden estar simultáneamente activos. Dicho esto, el ancho de banda que necesitará una oficina en particular se obtendrá utilizando la siguiente expresión:

$$\frac{\boldsymbol{ancho\ de\ banda}}{\boldsymbol{oficina}} = nº\ usuarios \cdot \frac{ancho\ de\ banda}{usuario} \cdot fs$$

Aplicando la fórmula a nuestro caso práctico, obtendríamos los siguientes valores:

$$Tipo\ A \rightarrow \frac{ancho\ de\ banda}{oficina_{tipo\ A}} = 50 \cdot 704 \cdot 0{,}45 = \ 15.840\ \text{kbps}$$

$$Tipo\ B \rightarrow \frac{ancho\ de\ banda}{oficina_{tipo\ B}} = 20 \cdot 704 \cdot 0{,}45 = \ 6.336\ \text{kbps}$$

$$Tipo\ C \rightarrow \frac{ancho\ de\ banda}{oficina_{tipo\ B}} = 8 \cdot 704 \cdot 0{,}45 = \ 2.534{,}4\ \text{kbps}$$

Representado estos resultados en forma de tabla obtendríamos lo siguiente:

Oficina	Nº usuarios	$\frac{ancho\ de\ banda}{usuario}\ (kbps)$	Factor simultaneidad	$\frac{ancho\ de\ banda}{oficina}\ (kbps)$
Tipo 1	50	704	0,45	**15.840**
Tipo 2	20	704	0,45	**6.336**
Tipo 3	8	704	0,45	**2.534,4**

Tabla 10 - Ancho de banda por oficina[46]

Llegados a este punto, si suponemos que existen 20 oficinas del tipo A, 60 oficinas del tipo B y 120 oficinas del tipo C (200 sedes en total), obtendríamos el cálculo de ancho de banda total del conjunto de oficinas de cada tipo:

[46] Fuente: elaboración propia

$$\frac{\boldsymbol{ancho\ de\ banda}}{\boldsymbol{oficinas}} = \frac{ancho\ banda}{oficina} \cdot n^{\underline{o}}\ oficinas$$

Si se aplica esta ecuación a nuestra situación:

$$Tipo\ A\ (20\ oficinas) \rightarrow \frac{ancho\ de\ banda}{oficinas_{tipo\ A}} = 15.840 \cdot 20 = \ 316800\ Kbps \rightarrow \ 316{,}8\ Mbps$$

$$Tipo\ B\ (60\ oficinas) \rightarrow \frac{ancho\ de\ banda}{oficinas_{tipo\ B}} = 6.336 \cdot 60 = \ 380160\ Kbps \rightarrow 380{,}16\ Mbps$$

$$Tipo\ C\ (120\ oficinas) \rightarrow \frac{ancho\ de\ banda}{oficinas_{tipo\ C}} = 2.534{,}4 \cdot 120 = \ 304128\ Kbps \rightarrow 304{,}128\ Mbps$$

Oficina	$\frac{ancho\ de\ banda}{oficina}\ (kbps)$	Nº de oficinas	$\frac{ancho\ de\ banda}{oficinas}(Mbps)$
Tipo 1	**15.840**	20	316,8
Tipo 2	**6.336**	60	380,16
Tipo 3	**2.534,4**	120	304,128
		200	**1001,088**

Tabla 11 - Ancho de banda por oficinas[47]

Entonces, el ancho de banda necesario para los dos HUBS de la sede central será la suma cálculos reflejados en la tabla anterior más el ancho de banda requerido de una oficina grande, ya que la sede central dispondrá, también, de 50 usuarios:

$$\frac{ancho\ de\ banda}{HUB} = \frac{ancho\ de\ banda}{oficina_{tipo\ A}} + \frac{ancho\ de\ banda}{oficinas_{tipo\ A}} + \frac{ancho\ de\ banda}{oficinas_{tipo\ B}} + \frac{ancho\ de\ banda}{oficinas_{tipoC}}$$

$$\frac{ancho\ de\ banda}{HUB} = 15{,}84 + 316{,}8 + 380{,}16 + 304{,}128 = 1016{,}93\ Mbps$$

Para el centro de respaldo o HUB2, suponemos que existe una degradación en el router del 34 % y que no hay usuarios, por lo que el cálculo de ancho de banda se efectuaría de la manera siguiente:

$$\frac{ancho\ de\ banda}{HUB} = \left(\frac{ancho\ de\ banda}{oficinas_{tipo\ A}} + \frac{ancho\ de\ banda}{oficinas_{tipo\ B}} + \frac{ancho\ de\ banda}{oficinas_{tipoC}}\right) \cdot degradación$$

[47] Fuente: elaboración propia

$$\frac{ancho\ de\ banda}{HUB} = (316{,}8 + 380{,}16 + 304{,}12) \cdot (1 - 0{,}34) = \ 660{,}72 Mbps$$

En la siguiente tabla se refleja el ancho de banda que se necesitaría para cada oficina, la sede central y el centro de respaldo, tomando como valores los cálculos obtenidos previamente:

	Oficina	Nº usuarios	Ancho de banda requerido (Mbps)
SPOKES	Tipo A	50	**15,84**
	Tipo B	20	**6,34**
	Tipo C	8	**2,53**
HUB1	Sede central	50	**1016,93**
HUB1-backup			**1016,93**
HUB2	Centro de respaldo		**660,72**

Tabla 12 - Ancho de banda de oficinas, sede central y centro de respaldo[48]

[48] Fuente: elaboración propia

5.2 MODELO DE ROUTER CISCO PARA CADA HUB Y SPOKE

En la simulación, la imagen virtual del modelo de router Cisco utilizada corresponde a la serie 7200. En esta sección, discutiremos qué modelo de *router* físico es el más apropiado para utilizar en cada oficina o sede, teniendo en consideración los cálculos de ancho de banda de la tabla anterior.

Dado que en nuestro caso práctico disponemos de una organización con numerosas oficinas remotas, un centro de respaldo y una sede central, que deberán responder a la alta demanda del tráfico de red (usuarios móviles, servicios en la nube, aplicaciones multimedia,), aprovechamiento del espacio en los racks, presupuestos fijos para el hardware, consumo de electricidad y refrigeración, etc; escogeremos la nueva arquitectura de routers de servicios integrados de Cisco: **ISR Cisco serie 4000**.

Esta línea de dispositivos incluye varias características importantes que los convierten en la elección perfecta para las sucursales modernas: precio por rendimiento (adaptación al aumento de ancho de banda usando únicamente un dispositivo sin necesidad de otros componentes de seguridad y optimización), rendimiento según la demanda (posibilidad de incrementar el ancho de banda sin necesidad de adquirir un nuevo dispositivo), servicios según la demanda (existen contenedores de servicios que permiten ejecutar máquinas virtuales en los routers) y servicios ampliables (integridad con servidores Cisco UCS gestionados independientemente al router, pero que utilizan la fuente de alimentación y chasis del mismo router).

Por otro lado, la serie 4000 de Cisco emplea un nuevo sistema operativo basado en Linux: Cisco iOS XE Software, que cuenta con el mismo diseño de la interfaz de usuario del sistema operativo Cisco iOS usados en modelos de routers anteriores. La principal característica de este sistema es la posibilidad de funcionamiento en entornos de CPU multiprocesador con una lógica de virtualización en base a contenedores de servicios, que ofrecen recursos de equipos virtualizados dedicados incluyendo CPU, almacenamiento en disco y memoria para cada servicio empleado.

Consecuentemente, permite responder eficientemente a los requerimientos de mayor seguridad y alta disponibilidad de las redes actuales.

La siguiente tabla refleja los modelos de routers Cisco de la serie 4000 y aquéllos que utilizan el software Cisco iOS XE. El eje Y representa el ancho de banda que el router es capaz de ofrecer y el eje X indica el tipo o tamaño de entidad recomendada:

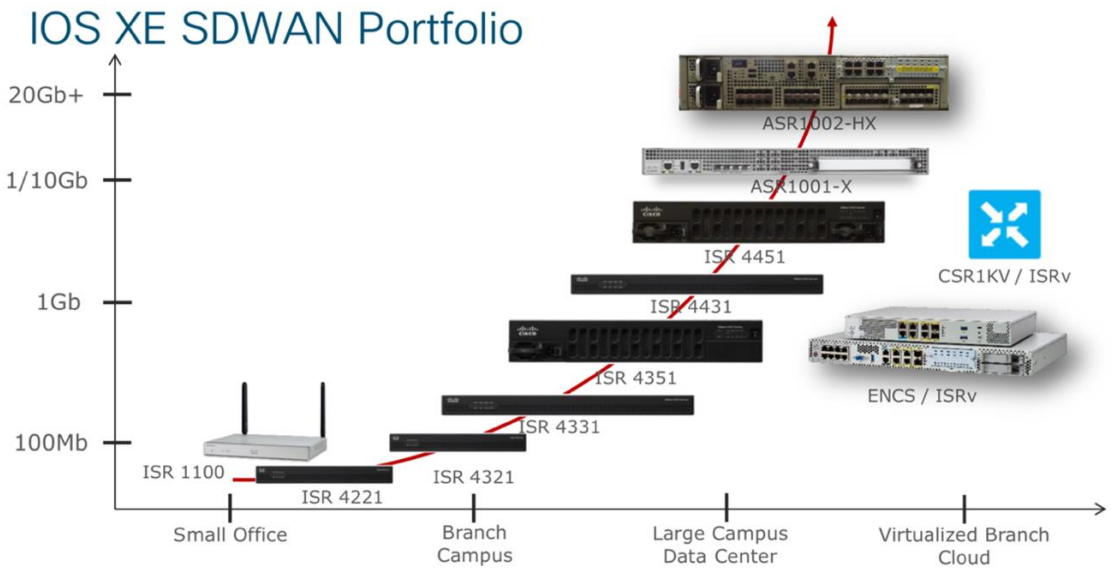


Figura 37 – Hardware. ISR Routers 4000[49]

El ancho de banda soportado por cada router y la capacidad de su ampliación según el tipo de licencia, queda reflejado en esta tabla:

	4421	4321	4331	4351	4431	4451
Rendimiento con licencia normal	35 Mbps	50 Mbps	100 Mbs	200 Mbps	500 Mbps	1 Gbps
Rendimiento con licencia "Performance license"	75 Mbps	100 Mbps	300 Mbps	400 Mbps	1 Gbps	2 Gbps
Rendimiento con licencia "*Boost license*"	1.2 Gbps	2 + Gbps	2 + Gbps	2 + Gbps	4 + Gbps	4 + Gbps

Figura 38 – Hardware. Licencias Routers ISR 4000[50]

[49] Fuente: ISR Cisco Integrated Services Router: Architectural Overview and Use Cases
[50] Fuente: elaboración propia

Todos los modelos de encaminadores expuestos previamente son compatibles con herramientas de prevención de intrusiones, protocolos de enrutamiento Multicast, IPSec, EIGRP, NHRP, entre otros. No obstante, antes de seleccionar el modelo de router, es importante tener en cuenta estudios de rendimiento o *benchmarks* para poder tener una panorámica inicial del comportamiento y del estado de los routers una vez hayamos aplicado las configuraciones pertinentes y estén en funcionamiento.

En este caso, nos basaremos en un *benchmark* realizado por la compañía *Miercom* de los modelos *4221*, *4321*, *4431*, *4451*. A priori, nos focalizaremos en los modelos 4221 para pequeñas y medianas oficinas, 4321 para oficinas grandes, 4451 para la sede central y 4431 para el centro de respaldo, ya que parecen ser los que más se ajustan a las necesidades de ancho de banda de cada sede.

Las siguientes gráficas, muestran los resultados de los tests de rendimiento de los routers seleccionados. Se debe tener presente que cada función aplicada en el router (QoS, IPSec, Firewall, NAT..) consumen recursos y siempre es conveniente dimensionar por exceso y tener en consideración la posibilidad de expansión en cuanto a sedes remotas e incremento en el número de trabajadores en cada oficina.

ISR4221 y ISR4321:

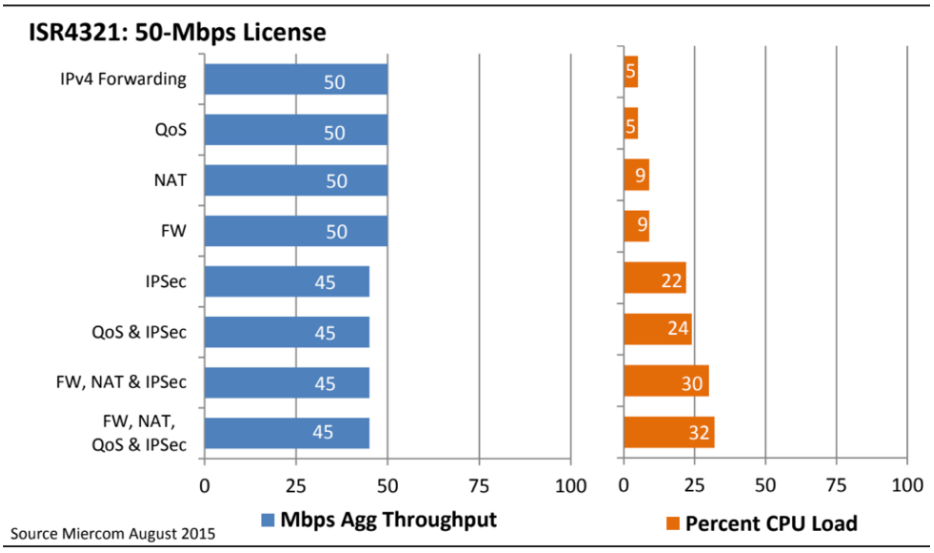


Figura 39 – Hardware. Benchmark Router ISR4221 y ISR4321[51]

[51] Fuente: Performance Analysis: Cisco ISR 4000 Family. Models 4321, 4331, 4351, 4431 & 4451. Miercom

ISR4431:

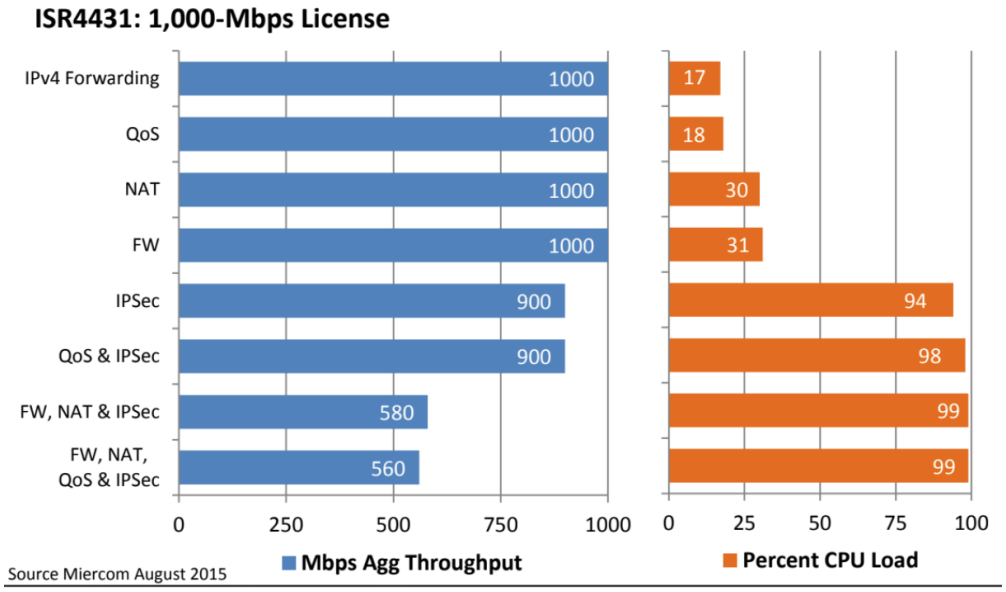


Figura 40 – Hardware. Benchmark Router ISR4431[52]

ISR4451:

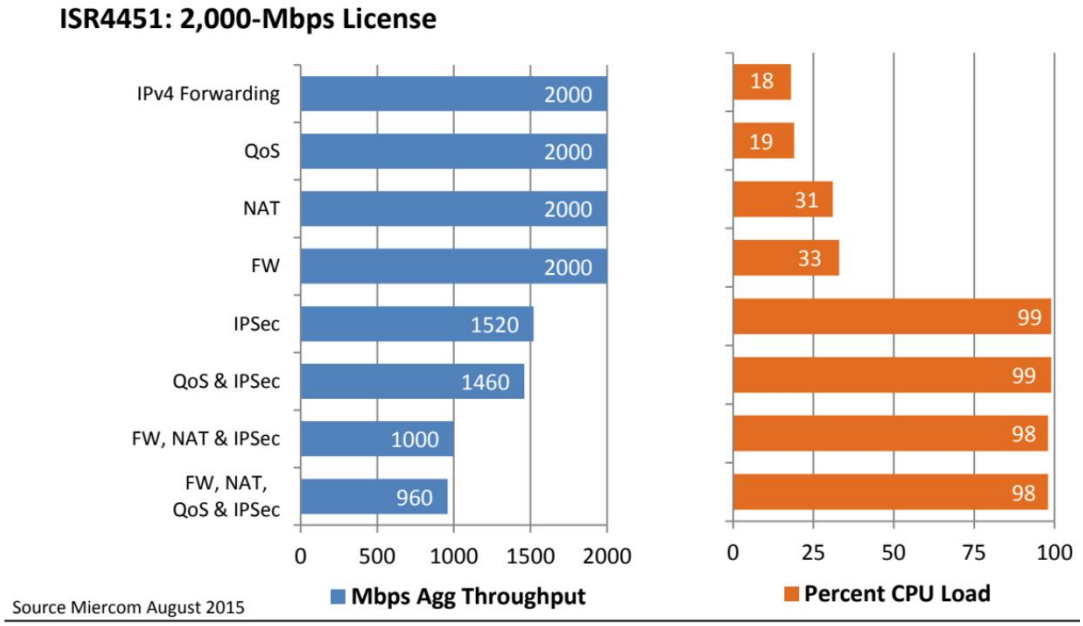


Figura 41 – Hardware. Benchmark Router ISR4451[53]

[52] Fuente: Performance Analysis: Cisco ISR 4000 Family. Models 4321, 4331, 4351, 4431 & 4451. Miercom
[53] Fuente: Performance Analysis: Cisco ISR 4000 Family. Models 4321, 4331, 4351, 4431 & 4451. Miercom

Es recomendable que el consumo de CPU de cada router no sobrepase el 75-80%, ya que su alto consumo podría repercutir negativamente en funcionamiento del encaminador, provocando pérdidas de paquetes y problemas de congestionamiento en la red. En nuestro caso, utilizaremos los valores de ancho de banda asociados a la configuración de QoS & IPSec para obtener el consumo de CPU de cada router:

$$Tipo\ A \rightarrow 15{,}84\ \cancel{Mbps} \cdot \frac{24\ \%\ carga\ CPU}{45\ \cancel{Mbps}} = 8{,}45\ \%\ carga\ CPU$$

$$Tipo\ B \rightarrow 6{,}34\ \cancel{Mbps} \cdot \frac{24\ \%\ carga\ CPU}{45\ \cancel{Mbps}} = 3{,}38\ \%\ carga\ CPU$$

$$Tipo\ C \rightarrow 2{,}53\ \cancel{Mbps} \cdot \frac{24\ \%\ carga\ CPU}{45\ \cancel{Mbps}} = 1{,}35\ \%\ carga\ CPU$$

$$Sede\ central \rightarrow 1016{,}93\ \cancel{Mbps} \cdot \frac{99\ \%\ carga\ CPU}{1460\ \cancel{Mbps}} = 68{,}96\ \%\ carga\ CPU$$

$$Centro\ de\ respaldo \rightarrow 660{,}72\ \cancel{kbps} \cdot \frac{99\ \%\ carga\ CPU}{1460\ \cancel{kbps}} = 71{,}94\ \%\ carga\ CPU$$

Mostramos los cálculos junto con el modelo de *router* adecuado para cada oficina y la carga de la CPU en función del ancho de banda necesitado:

	Oficina	Ancho de banda requerido (Mbps)	Modelo de router seleccionado	Carga de CPU (%)
SPOKES	Tipo A	**15,84**	ISR4321: 50-Mbps	8,45
	Tipo B	**6,34**	ISR 4221: 35-Mbps	3,38
	Tipo C	**2,53**	ISR 4221: 35-Mbps	1,35
HUB1	Sede central	**1016,93**	ISR4451: 2.000-Mbps	68,96
HUB1-backup		**1016,93**	ISR4451: 2,000-Mbps	68,96
HUB2	Centro de respaldo	**660,72**	ISR4431: 1,000-Mbps	71,94

Tabla 13 - Ancho de banda y modelo de Router ISR adecuado[54]

[54] Fuente: elaboración propia

5.3 BLOQUE 1: ROUTING

En términos generales, existen tres fases DMVPN:

Fase	Características
1	Conectividad HUB-SPOKE. Para la comunicación SPOKE-SPOKE, el tráfico siempre circulará, en primer lugar, por el HUB en la Fase 1.
2	Permite establecer comunicaciones túnel bajo demanda SPOKE-SPOKE con la restricción de que los SPOKES han de recibir rutas especificas para todas las subredes de SPOKES remotos.
3	Mejora de las capacidades de comunicación de SPOKE-SPOKE eliminando la restricción anterior mediante el empleo de mensajes de indicación de tráfico procedentes del HUB a los SPOKES que informan de la existencia de un mejor camino para alcanzar un determinado objetivo en la red. Esta funcionalidad se consigue configurando los comandos `ip nhrp redirect` en el HUB e `ip nhrp shortcut` en los SPOKES.

Tabla 14 – Routing. Fases DMVPN[55]

Por motivos de compatibilidad y rendimiento con la imagen de virtual de encaminador seleccionado en la simulación GNS3, nos enfocaremos en las técnicas de tunelizado necesarias para implementar la 2ª fase DMVPN, para que la sede central sea capaz de establecer enlaces de comunicación con las oficinas y el centro de respaldo; así como la comunicación entre oficinas.

5.3.1 TÉCNICAS DE TUNELIZADO

Seguidamente nos focalizaremos qué secuencia de comandos han de aplicarse a cada router para establecer túneles IP (sin IPSec) utilizando dos WAN de distintos operadores (Internet y MPLS). Finalmente, se verificará el funcionamiento de la conectividad con comandos Cisco iOS, Wireshark y pruebas de basculación para medir los tiempos de recuperación de la red ante una caída de túnel, enlace o router y dirigir el tráfico hacia una ruta alternativa, comprobando la alta disponibilidad de la red.

[55] Fuente: elaboración propia

5.3.1.1 CONFIGURACIÓN

En primer lugar, asignaremos las interfaces físicas (Fast Ethernet) a cada router:

	Red	Dirección IP	Máscara	Anotaciones
SPOKES	A	f0/0: 81.134.91.2	255.255.255.0	*Salida a Internet*
		f1/0: 10.4.1.1	255.255.255.0	*Puerta de enlace*
		f1/1: 172.16.2.2	255.255.255.0	*Salida a MPLS*
	B	f0/0: 91.35.197.14	255.255.255.0	*Salida a Internet*
		f1/0: 10.4.2.1	255.255.255.0	*Puerta de enlace*
		f1/1: 172.16.3.2	255.255.255.0	*Salida a MPLS*
	C	f0/0: 194.135.136.116	255.255.255.0	*Salida a Internet*
		f1/0: 10.4.3.1	255.255.255.0	*Puerta de enlace*
		f1/1: 172.16.4.2	255.255.255.0	*Salida a MPLS*
HUB1	Sede central	f0/0: 10.3.1.1	255.255.255.0	
		f1/0: 194.56.21.1	255.255.255.0	*Salida a Internet*
		f1/1: 172.16.5.1	255.255.255.0	*Salida a MPLS*
HUB1-backup		f0/0: 10.3.1.5	255.255.255.0	
		f1/0: 194.56.21.3	255.255.255.0	*Salida a Internet*
		f1/1: 172.16.5.3	255.255.255.0	*Salida a MPLS*
HSRP		10.3.1.100	255.255.255.0	*IP virtual*
HUB2	Centro de respaldo	f0/0: 10.3.2.1	255.255.255.0	
		f1/0: 194.56.22.1	255.255.255.0	*Salida a Internet*
		f1/1: 172.16.1.1	255.255.255.0	*Salida a MPLS*
R1	WAN1 Internet	f0/0: DHCP		*Wifi*
		f1/0: 194.56.21.2	255.255.255.0	
		f1/1: 81.134.91.1	255.255.255.0	
		f2/0: 91.35.197.13	255.255.255.0	*Internet*
		f3/0: 194.135.136.115	255.255.255.0	
		f3/1: 194.56.22.2	255.255.255.0	
R2	WAN2 MPLS	f1/0: 172.16.5.2	255.255.255.0	
		f1/1: 172.16.2.1	255.255.255.0	
		f2/0: 172.16.3.1	255.255.255.0	*MPLS*
		f3/0: 172.16.4.1	255.255.255.0	
		f3/1: 172.16.1.2	255.255.255.0	

Tabla 15 – Routing. Interfaces físicas de los Routers de la simulación[56]

[56] Fuente: elaboración propia

Las dos WAN están representadas por el R1 (Internet) y R2 (MPLS). La siguiente ilustración representa el montaje del Piloto. En ella, podemos observar qué direcciones IP están asociadas a cada interfaz del router:

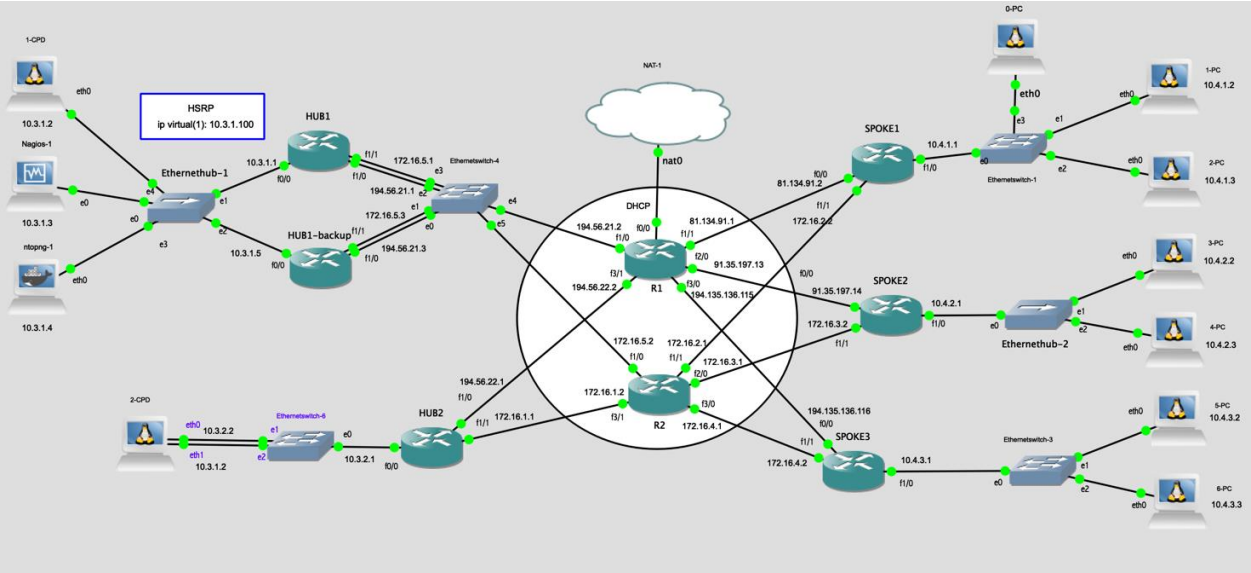


Figura 42 – Routing. Interfaces físicas en el Piloto[57]

A continuación, implementaremos las técnicas de tunelizado. Como se ha comentado en la sección de GRE Multipunto del "*Marco Teórico y Estado del Arte*", para poder resolver el problema de escalabilidad que presenta el modo punto a punto de GRE, se requiere recurrir a mGRE o GRE multipunto. Dentro del sistema operativo de red Cisco iOS XE, la configuración de mGRE se realiza definiendo interfaces virtuales, siguiendo la nomenclatura "**`interface tunnel x`**" donde x es un identificador que permite diferenciar un túnel de otro. Del mismo modo que las interfaces físicas, las interfaces virtuales requieren una dirección IP con su correspondiente máscara de subred, así como parámetros comunes a cualquier interfaz como velocidad de transmisión, política de colas, el tamaño máximo de trama…

El protocolo GRE tiene un campo que permite identificar un flujo GRE dentro de la misma interfaz, de manera que es posible hacer confluir los flujos de diferentes SPOKES en una única interfaz túnel del HUB, utilizando para ello el protocolo NHRP.

[57] Fuente: elaboración propia

En la siguiente tabla quedan reflejadas las direcciones IP y la máscara de red de los túneles de cada router de la topología (R1 y R2 no requieren la configuración de ningún túnel ya que simulan las WAN). Observaremos que existen dos túneles por cada router: el **Tunnel10 será el principal** y la comunicación será establecida empleando la red **MPLS** (R2), mientras que el **Tunnel1 será el secundario** y la transmisión de datos se establecerá por **Internet** (R1). Para establecer qué túnel es el principal y el secundario, estableceremos valores distintos del comando **`delay`** en la configuración del túnel.

	Tipo	Red LAN	IP Túnel	IP pública
SPOKES	A	10.4.1.0/24	Tunnel1: 10.1.0.3/24	81.134.91.0/24
			Tunnel10: 10.2.0.3/24	172.16.2.0/24
	B	10.4.2.0/24	Tunnel1: 10.1.0.4/24	91.35.197.0/24
			Tunnel10: 10.2.0.4/24	172.16.3.0/24
	C	10.4.3.0/24	Tunnel1: 10.1.0.5/24	194.135.136.0/24
			Tunnel10: 10.2.0.5/24	172.16.4.0/24
HUB1	Sede central	10.3.1.0/24	Tunnel1: 10.1.0.1/24	194.56.21.0/24
			Tunnel10: 10.2.0.1/24	172.16.5.0/24
HUB1-backup			Tunnel1: 10.1.0.10/24	194.56.21.0/24
			Tunnel10: 10.2.0.10/24	172.16.5.0/24
HUB2	Centro de respaldo	10.3.2.0/24	Tunnel1: 10.2.0.1/24	194.56.22.0/24
			Tunnel10: 10.2.0.2/24	172.16.1.0/24

Tabla 16 – Routing. Túneles principales y secundarios[58]

[58] Fuente: elaboración propia

Para clarificar en mayor grado la explicación y exposición de la configuración de las técnicas de tunelizado de la fase 2 DMVPN, únicamente nos focalizaremos en la configuración del túnel principal (Tunnel10) en HUB1 y los tres SPOKES:

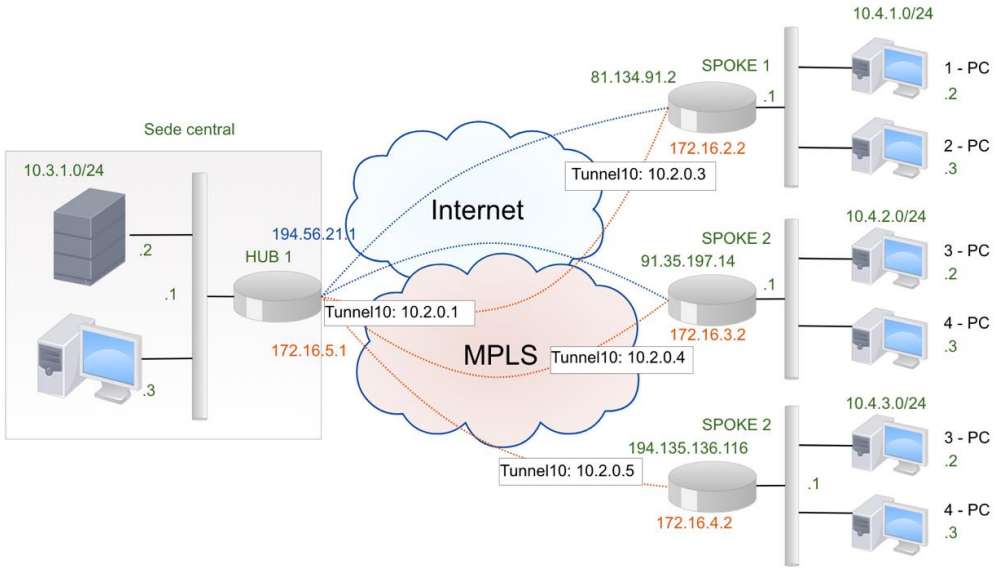


Figura 43 – Routing. Diagrama de red simplificado usando MPLS y Tunnel10[59]

Esta situación puede ser simulada en GNS3 si activados los dispositivos correspondientes (HUB1, R2, SPOKE1, SPOKE2 y SPOKE3) y ciñéndonos únicamente en el Túnel principal. La línea en verde del siguiente diagrama en GNS3 representa esta situación:

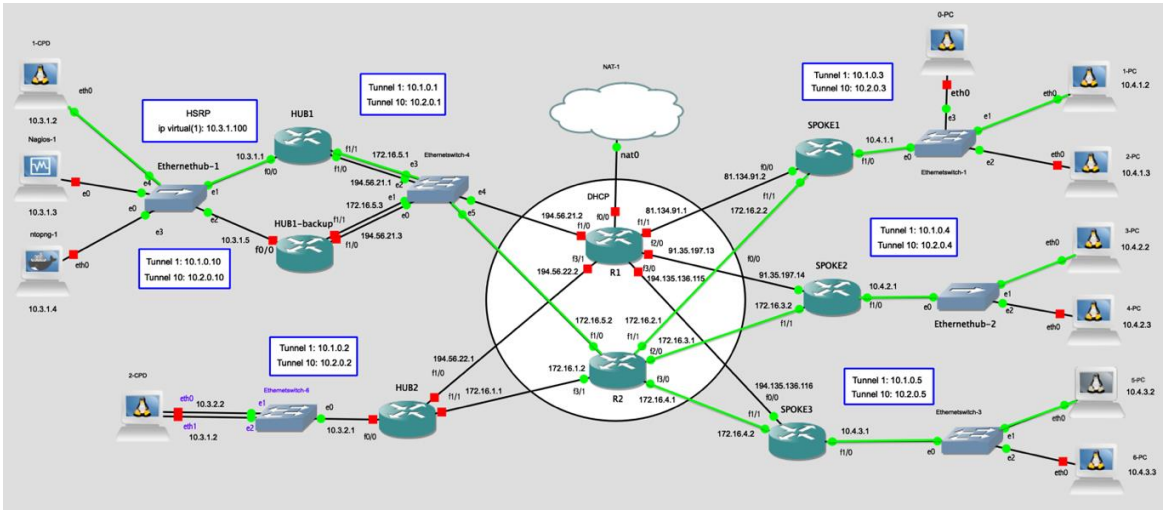


Figura 44 – Routing. Piloto GN3 utilizando la MPLS y Tunnel10[60]

[59] Fuente: elaboración propia
[60] Fuente: elaboración propia

La secuencia de comandos que se han aplicado en el HUB1 son:

Comando	Detalles
`interface Tunnel10`	Configuración de una interfaz Túnel.
`bandwidth 10000`	Ancho de banda en Kbps.
`ip address 10.2.0.1 255.255.255.0`	Dirección IP y máscara de subred del túnel.
`no ip redirects`	Evita que el router envíe mensajes ICMP redirect al resto de nodos.
`no ip next-hop-self eigrp 100`	Permite la comunicación dinámica y directa de túneles SPOKE-SPOKE en EIGRP.
`no ip split-horizon eigrp 100`	Desactiva el horizonte dividido o Split Horizon en la interfaz túnel mGRE. De lo contrario, EIGRP no devolvería información sobre una ruta a la dirección desde la que ha llegado.
`ip mtu 1400`	Habilita el tamaño máximo de la trama.
`ip nhrp authentication test`	Configuración de un *string* de autenticación.
`ip nhrp map multicast dynamic`	Habilita el reenvío de tráfico Multicast a través del túnel a los SPOKES dinámicos.
`ip nhrp network-id 200000`	Identificador que ha de coincidir con todos los nodos que quieran usar este túnel mGRE.
`ip nhrp holdtime 15`	Modifica los segundos que las direcciones NHRP NBMA se anuncian como válidas en respuestas autorizadas NHRP.
`ip nhrp registration timeout 3`	Configura el intervalo de envió de peticiones de registro NHRP.
`ip tcp adjust-mss 1360`	Configura el valor de MSS de los paquetes TCP que circulan a través del router.
`delay 200`	Modifica la métrica del enrutamiento EIGRP para las rutas aprendidas sobre la interfaz túnel.
`keepalive 3 2`	Activa el envío de mensajes para comprobar el estado de las conexiones. En este caso, se enviarán mensajes keepalive cada 3 segundos y 4 reintentos.
`tunnel source 172.16.5.1`	Dirección IP origen de la interfaz túnel.
`tunnel mode gre multipoint`	Habilita el modo mGRE como modo de encapsulación.
`tunnel key 200000`	Identificador para la interfaz túnel.

Tabla 17 – Routing. Configuración Tunel principal en HUB1[61]

[61] Fuente: elaboración propia

Del mismo modo, indicamos la configuración del túnel principal del SPOKE1. La configuración en el resto de SPOKES es muy similar, únicamente sería necesario modificar la dirección IP de la interfaz túnel y su dirección IP de origen:

Comando	Detalles
interface Tunnel10	
bandwidth 10000	
ip address 10.2.0.3 255.255.255.0	
no ip redirects	
ip mtu 1400	
no ip next-hop-self eigrp 100	
no ip split-horizon eigrp 100	
ip nhrp authentication test	
ip nhrp map multicast dynamic	
ip nhrp map 10.2.0.1 172.16.5.1	Configura estáticamente la asignación de direcciones IP a NBMA de los destinos IP conectados a la red NBMA.
ip nhrp map multicast 172.16.5.1	Permite el uso de un protocolo de enrutamiento dinámico entre el SPOKE y el HUB y permite el envío de paquetes Multicast al HUB.
ip nhrp network-id 200000	
ip nhrp holdtime 300	
ip nhrp nhs 10.2.0.1	Indica el servidor del siguiente salto. En este caso es el túnel principal del HUB.
ip nhrp registration timeout 3	
ip tcp adjust-mss 1360	
delay 200	
if-state nhrp	Permite controlar el estado de la interfaz virtual según si los túneles DMVPN conectados a la interfaz están activos o no.
keepalive 3 2	
tunnel source 172.16.2.2	
tunnel mode gre multipoint	
tunnel key 200000	

Tabla 18 – Routing. Configuración tunel principal en SPOKE1[62]

[62] Fuente: elaboración propia

En vista de la configuración anterior, podemos constatar que **únicamente se requiere configurar una interfaz túnel en el HUB**, sea cual sea la cantidad de SPOKES. El límite dependerá de la plataforma hardware del propio HUB. Por otro lado, todas las direcciones IP de la interfaces túnel se encuentran en una única subred (10.2.0.0/24, en este caso), construyendo así una red con características de difusión, pero con conexiones punto a punto (red NBMA). Es decir, el HUB puede enviar un mensaje que llegue a todos los SPOKES a la vez. Consecuentemente, esta característica permite simplificar la gestión de direcciones IP y facilita la puesta en marcha de protocolos de enrutamiento.

También es importante mencionar los **comandos NHRP** existentes en el HUB y SPOKE, dado que permite al HUB conocer a qué dirección IP pública ha de enviar un paquete tunelizado con una determinada dirección privada.

Por otro lado, para permitir la existencia de **varios flujos mGRE en el mismo router**, y por tanto, poner en servicio varias NBMA de manera simultánea, se emplea el comando **`tunnel key 200000`**, que activa el uso del campo 'Key' en los paquetes GRE. Ahora bien, tras la aplicación de esta funcionalidad, se produce un incremento de 4 bytes en la cabecera mGRE.

5.3.1.2 FRAGMENTACIÓN IP

Un factor significativo en las configuraciones expuestas previamente es la consideración de la fragmentación IP.

El protocolo IP tiene la peculiaridad de poder utilizar diferentes protocolos de enlace para formar, de esta forma, grandes redes como Internet. Debido a que cada protocolo de enlace dispone de su propio MTU, la fragmentación IP permite partir el paquete original empleando el campo "*Offset*" y los bits "*More Fragment*" de la cabecera IP para adaptar el paquete al tamaño del protocolo de enlace. No obstante, este mecanismo presenta dos contrapartidas:

1. El receptor se encarga de re-ensamblar los fragmentos y en caso de que le falte alguno, deberá de esperar un tiempo y mantener en memoria al resto de fragmentos.

2. No será posible construir el datagrama original en caso de pérdida de algún fragmento.

Además, la fragmentación produce efectos nocivos en la eficiencia de la red de comunicaciones. A pesar de que en la transmisión de un datagrama muy grande o el mismo en varios trozos no produce muchas diferencias en cuanto a tiempo de transmisión de paquetes, influye negativamente en el tiempo de encaminamiento de esos paquetes. Precisamente, hoy en día la fragmentación se elude en la medida de lo posible para prevenir retardos en el encaminamiento, desbordamientos en las capacidades de los routers (tanto en memoria como en procesador) y problemas de congestionamiento.

En nuestro caso práctico, el MTU de las interfaces túnel principales y secundarias son más pequeñas ya que incluiremos las cabeceras GRE e IPSEC. Además el encargado de re-ensamblar los paquetes será el otro extremo de la conexión IPSec. Debido a que los volúmenes de información pueden ser elevados, existe la posibilidad de que receptor acabe en condición de error y el enlace se quede sin servicio.

Para hacer frente a estos problemas, hemos aplicado los comandos **`ip mtu 1400`** en cada interfaz túnel y **`ip tcp adjust-mss 1360`** también en cada interfaz túnel y en las interfaces correspondientes a las puertas de enlace de las LAN de cada red:

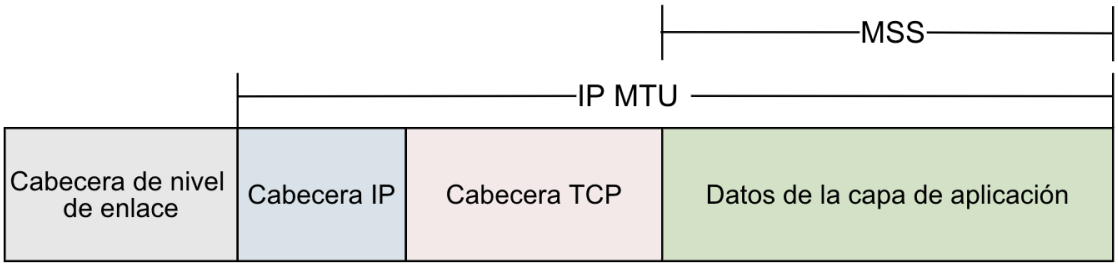


Figura 45 – MTU y MSS[63]

El valor introducido en el primer comando (1400), que nos permite ajustar el MTU en bytes de los paquetes enviados a una interfaz, ha sido calculado mediante un balance de cabeceras y contrastado con capturas de red. Mediante el otro comando,

[63] Fuente: materiales de la asignatura de Interconexión de Redes. José Ángel Berná

indicamos al router que intercepte los establecimientos de conexiones TCP y fuerce a negociar un MSS (MTU-40) menor, 1360 en este caso.

Este valor configurado de MSS es informado por cada extremo de la comunicación en el establecimiento de la conexión TCP, insertándolo en la cabecera TCP (campo opciones) de los paquetes SYN.

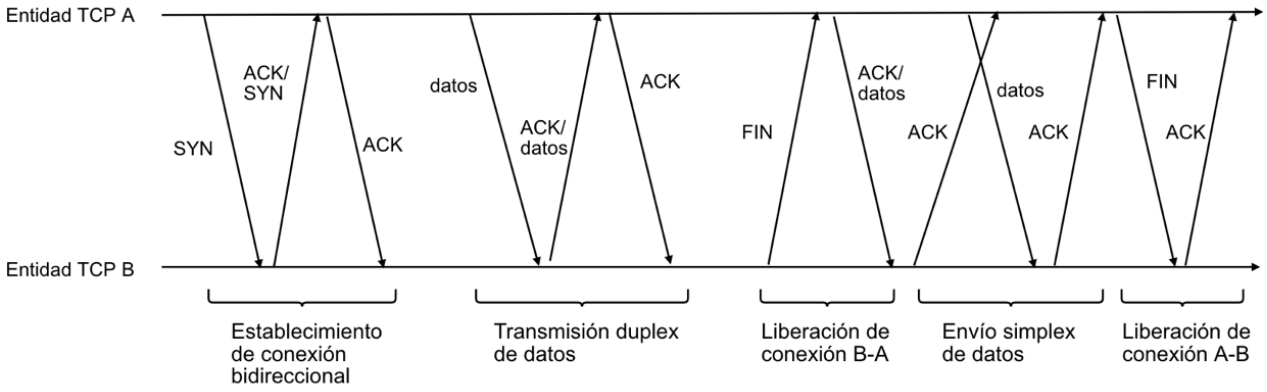


Figura 46 - Funcionamiento del protocolo TCP[64]

No obstante, cabe la posibilidad de que los datos sean fragmentados en su camino al destino al intercambiar el valor de MSS entre los dos extremos. Por ello, la norma RFC 1191 fue introducida, precisamente, para asegurar el rendimiento de las conexiones TCP, previniendo que sean fragmentados los datagramas IP que contienen paquetes de una conexión TCP. En el punto 5.3.1.5, comprobaremos si esta normativa se encuentra bien configurada en nuestro Piloto de red.

5.3.1.3 VALIDACIÓN DE CONECTIVIDAD HUB-SPOKE

Comenzaremos las pruebas de validación comprobando la conexión HUB-SPOKES. Los enlaces y dispositivos marcados en verde están en funcionamiento:

[64] Fuente: materiales de la asignatura de Interconexión de Redes. José Ángel Berná

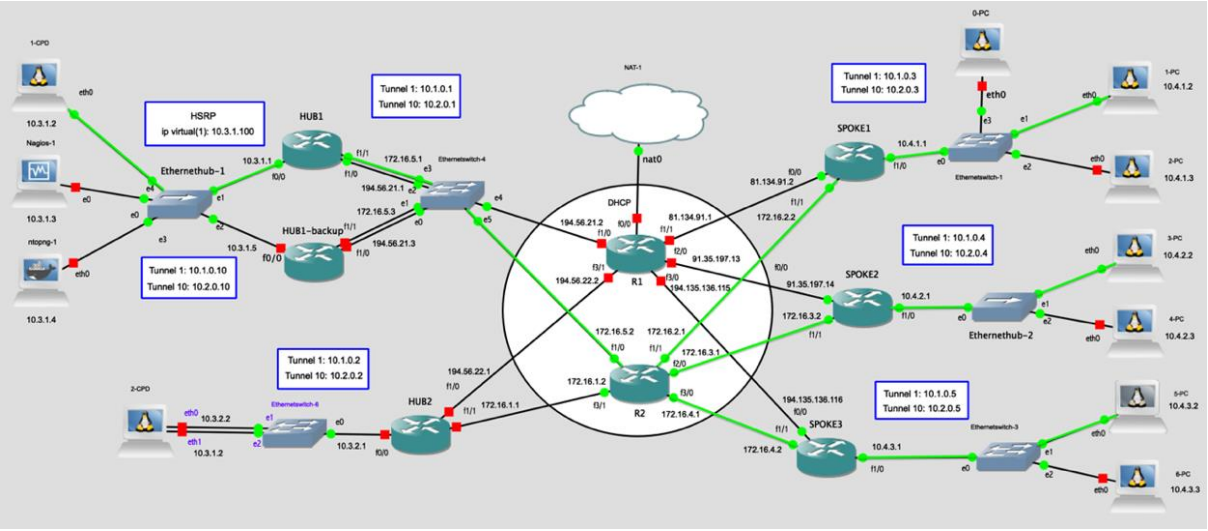


Figura 47 – Routing. Conectividad HUB-SPOKE[65]

Para verificar que la red está funcionando según lo esperado, utilizaremos una serie de comandos que ofrece el sistema operativo Cisco iOS tanto en los HUBs como en los SPOKES. En primer lugar comprobaremos la conectividad, por ejemplo del SPOKE1 al túnel primario del HUB1 mediante `ping`:

```
SPOKE1#ping 10.2.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/30/60 ms
SPOKE1#
```

El comando **`show dmvpn`** puede ser utilizado para mostrar información de sesión de la DMVPN. Tras ejecutarlo, observaremos que existe una etiqueta denominada "*Attrb*", que tiene el valor de 'S', esto es, que el SPOKE tiene una asociación estática con el HUB (lo mismo ocurriría con el resto de SPOKEs):

```
SPOKE1#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel10, IPv4 NHRP Details
Type:Spoke, NHRP Peers:3,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 172.16.5.1            10.2.0.1    UP 00:15:17     S
```

[65] Fuente: elaboración propia

Ahora bien, si ejecutamos el mismo comando en el HUB, apreciamos que existen las tres direcciones NBMA y unos túneles correspondientes al SPOKE1, SPOKE2 y SPOKE3 que el HUB ha aprendido dinámicamente mediante el protocolo NHRP (letra 'D' en el parámetro Attrb):

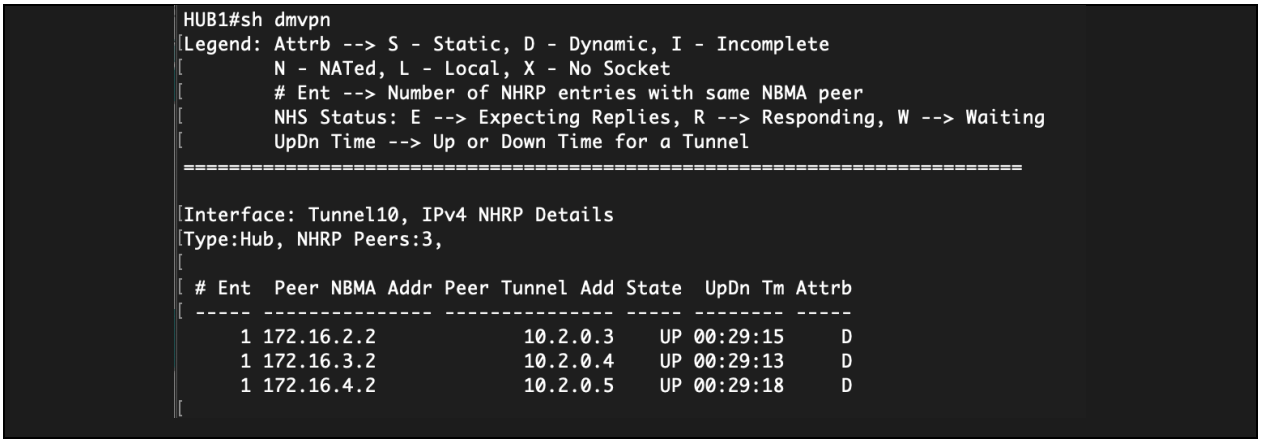


Tabla NHRP:

Por otra parte, si se emplea el comando **`show ip nhrp`**, podemos observar las direcciones privadas de las interfaces túnel con su dirección IP pública o NBMA asociada (tabla NHRP). Como el HUB aprende las direcciones NBMA de los SPOKES a través de NHRP, éstas pueden ser dinámicas (en la captura de pantalla el atributo 'Type' tiene el valor 'dynamic'), reduciendo así los costes de conexión:

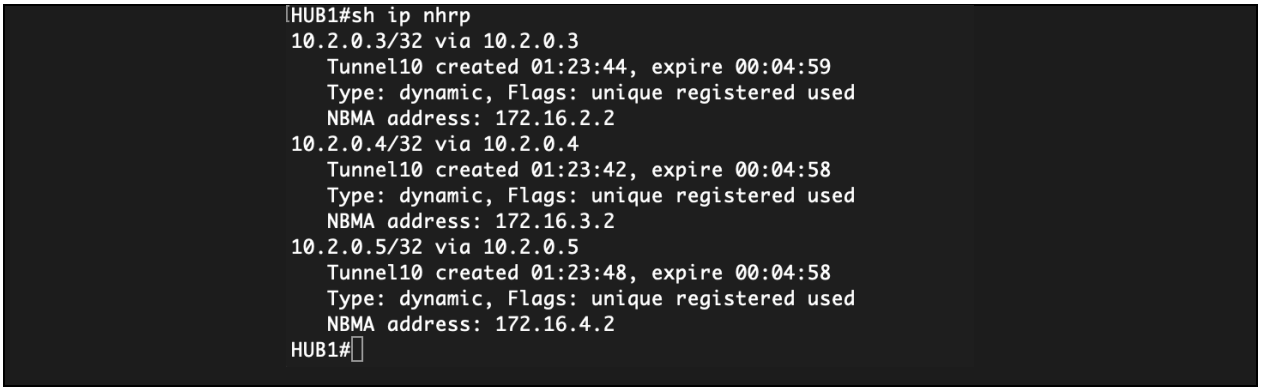


Tabla de vecinos:

El proceso de descubrimiento de vecinos se lleva a cabo mediante el envío periódico de paquetes tipo *Hello,* utilizando la dirección de Multicast 224.0.0.10 y

asegurando su entrega de manera ordenada y libre de errores utilizando el protocolo RTP:

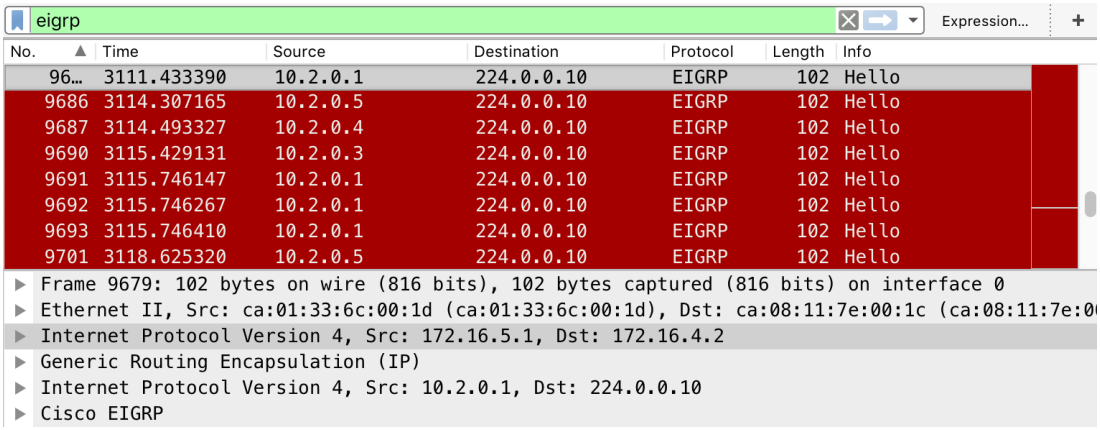


Figura 48 – Routing. Wireshark. Paquetes "Hello" EIGRP[66]

Estos paquetes *Hello* contienen un parámetro denominado "*Hold Time*" (por defecto 15 segundos), que indica el valor a partir del cual el router considera a su sucesor como inalcanzable y comienza a utilizar el sucesor factible seleccionado utilizando las métricas K1 y K3 (ancho de banda y retardo) respectivamente:

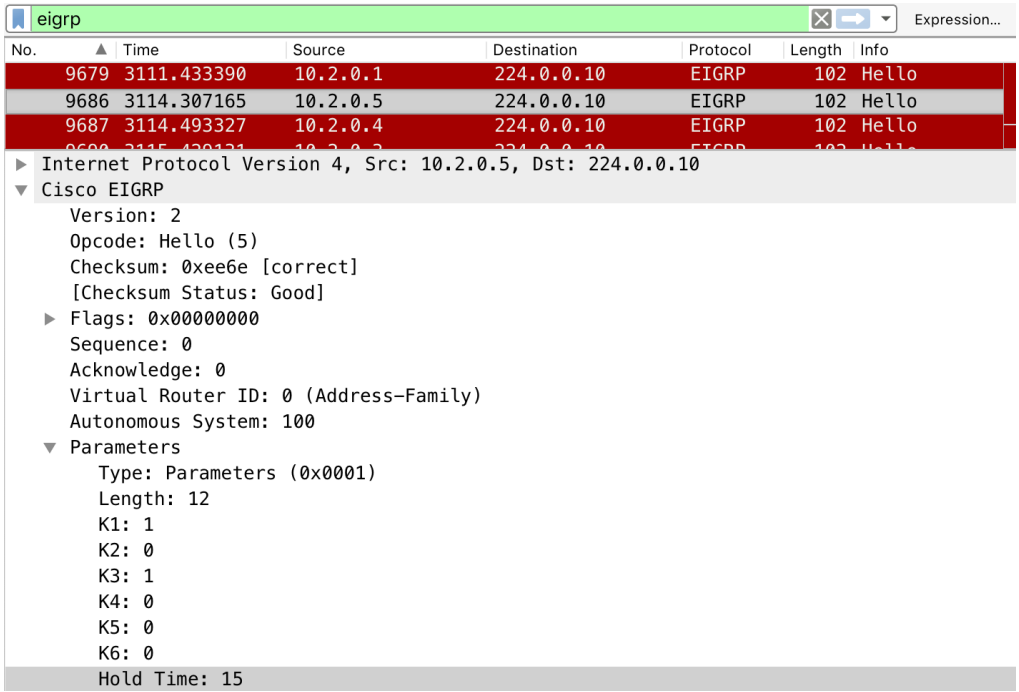


Figura 49 – Routing. Wireshark. Valor "Hold Time" en paquetes Hello EIGRP[67]

[66] Fuente: elaboración propia
[67] Fuente: elaboración propia

Por otra parte, podemos utilizar el comando **`show ip eigrp neighbors (sh ip eigrp nei)`** para comprobar la tabla de vecinos. Los valores contenidos en dicha tabla son de gran utilidad, ya que muestran información sobre el estado de los enlaces. La siguiente imagen muestra los resultados obtenidos al aplicar dicho comando al HUB1:

```
HUB1#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
2   10.2.0.3                Tu10                     14 00:00:15   54   324  0  4
1   10.2.0.5                Tu10                     14 00:00:15  149   894  0  3
0   10.2.0.4                Tu10                     13 00:00:19  161   966  0  5
HUB1#
```

Observamos los siguientes atributos:

- ⇒ **`H`**: orden en el cual los vecinos fueron descubiertos: el 0 indica qué vecino se ha descubierto en primer lugar y el 1 indica el último.
- ⇒ **`Address`**: dirección IP del vecino.
- ⇒ **`Interface`**: interfaz en la cual se recibió el paquete Hello.
- ⇒ **`Hold (sec)`**: cantidad de segundos para que el router descarte al vecino antes de recibir un paquete Hello.
- ⇒ **`Uptime`**: tiempo que lleva el vecino activo.
- ⇒ **`SRTT`**: promedio en milisegundos entre la transmisión de un paquete a un vecino y la recepción de la confirmación.
- ⇒ **`RTO`**: Si falla un envío por Multicast, se envía un paquete Unicast al router correspondiente. El RTO es, por tanto, el tiempo que se tarda en recibir la respuesta.
- ⇒ **`Q Cnt`**: número de paquetes en las colas de espera.
- ⇒ **`Seq Num`**: número de secuencia del último paquete EIGRP recibido.

Tabla de rutas:

Por otro lado, si ejecutamos el comando **`sh ip route`**, podemos observar las rutas EIGRP establecidas hacia los terminales de cada SPOKE (la letra 'D' que aparece en la imagen indica que la ruta ha sido aprendida por EIGRP):

```
HUB1#sh ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 194.56.21.2 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
D        10.4.1.0/24 [90/309760] via 10.2.0.3, 00:24:16, Tunnel10
D        10.4.2.0/24 [90/309760] via 10.2.0.4, 00:24:16, Tunnel10
D        10.4.3.0/24 [90/309760] via 10.2.0.5, 00:24:16, Tunnel10
```

Tabla de topología:

Por otra parte, el comando **`show ip eigrp topology (sh ip eigrp topo)`** nos permite visualizar la tabla de topología del router. La siguiente ilustración muestra la tabla de topología del HUB1:

```
HUB1#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(194.56.21.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.4.2.0/24, 1 successors, FD is 309760
        via 10.2.0.4 (309760/28160), Tunnel10
P 10.4.1.0/24, 1 successors, FD is 309760
        via 10.2.0.3 (309760/28160), Tunnel10
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 10.4.3.0/24, 1 successors, FD is 309760
        via 10.2.0.5 (309760/28160), Tunnel10
P 10.3.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

En la imagen se puede observar:

⇒ Distancia Factible o '*Feasible Distance*' (cuadrado en verde): 309760

⇒ Los succesores.

⇒ Distancia de cada posible sucesor a la red destino o *'Reported Distance'* (cuadrado azul): 28160

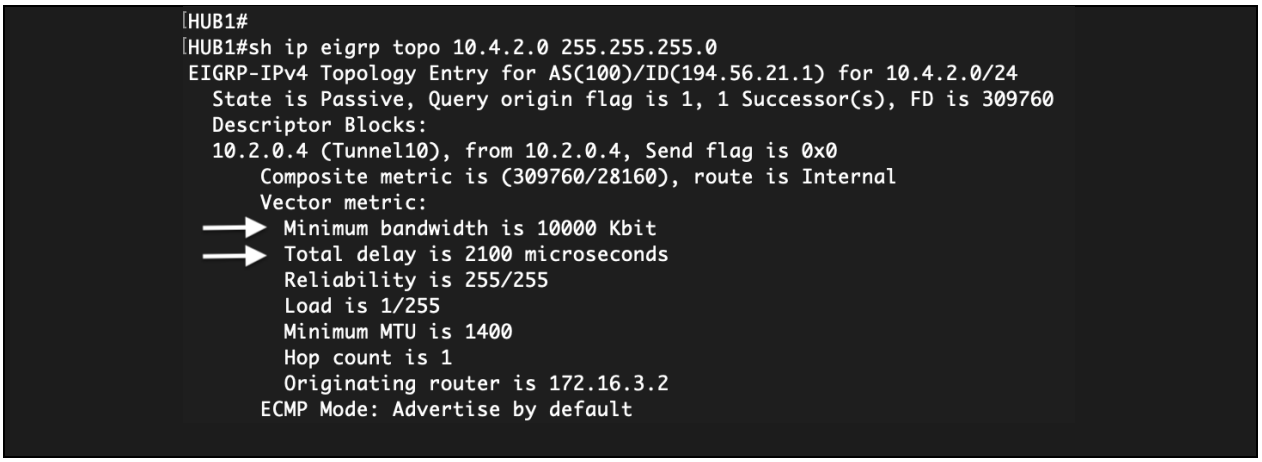
⇒ La distancia calculada desde cada sucesor posible (cuadrado en rojo): 309760

⇒ La interfaz donde se descubrió cada sucesor factible. En este caso, Tunnel10.

El valor **309.760** correspondiente a la distancia calculada o *Composite Metric* para alcanzar la red privada 10.4.2.0/24 desde el HUB1, ha sido obtenido aplicando la formula simplificada de la métrica EIGRP:

$$metric = \left(\frac{10^7}{BW_{min}} + \sum delays\right) \cdot 256$$

Una forma de obtener los valores de ancho de banda y retardo del HUB1, sería la introducción del comando `sh ip eigrp topo 10.4.1.1 255.255.255.0` en la CLI:



$$BW_{min} = 10.000\ Kbit$$

El valor del Delay mostrado por el comando previo ha de ser divido entre 10 para realizar los cálculos de la métrica para realizar la conversión a decenas de microsegundos:

$$Delay\ value = \frac{Cumulative\ value}{10} = \frac{2100}{10} = 210$$

$$Composite\ Metric = \left(\frac{10^7}{10.000} + 210\right) \cdot 256 = 309.760$$

$$Composite\ Metric = Feasible\ distance = \mathbf{309.760}$$

Para este ejemplo en concreto, observamos que sólo existe un único sucesor para alcanzar una determinada ruta. Esto significa que si, por ejemplo, el túnel 10.2.0.3 falla, no se podrá alcanzar a la red 10.4.1.0/24 (red LAN de la oficina tipo A) desde el HUB y el resto de SPOKEs. Por este mismo motivo, que la red dispondrá de dos WAN de operadores diferentes para adoptar propiedades de alta disponibilidad y, por tanto,

que hayan rutas secundarias de mayor coste para que en caso de fallo de túneles, enlaces o routers, se produzca basculaciones a enlaces alternativo teniendo presente las métricas EIGRP. Para conseguir tal fin, se han configurado los túneles principales y secundarios según queda reflejado en el punto "Configuración de los routers" del Anexo y tras ello, se han realizado pruebas de basculación, que estarán expuestas más tarde.

5.3.1.4 VALIDACIÓN DE CONECTIVIDAD SPOKE-SPOKE

Una comunicación HUB-SPOKE se realiza mediante túneles estáticos. No obstante, la comunicación SPOKE-SPOKE se lleva a cabo mediante túneles DMVPN, es decir, túneles dinámicos bajo demanda (sólo se crean cuando realmente se necesitan). El siguiente diagrama representa la situación que se abordará en esta sección. Seguimos utilizando únicamente el túnel principal y los routers HUB1, SPOKE1, SPOKE2, SPOKE3 y R2 (MPLS):

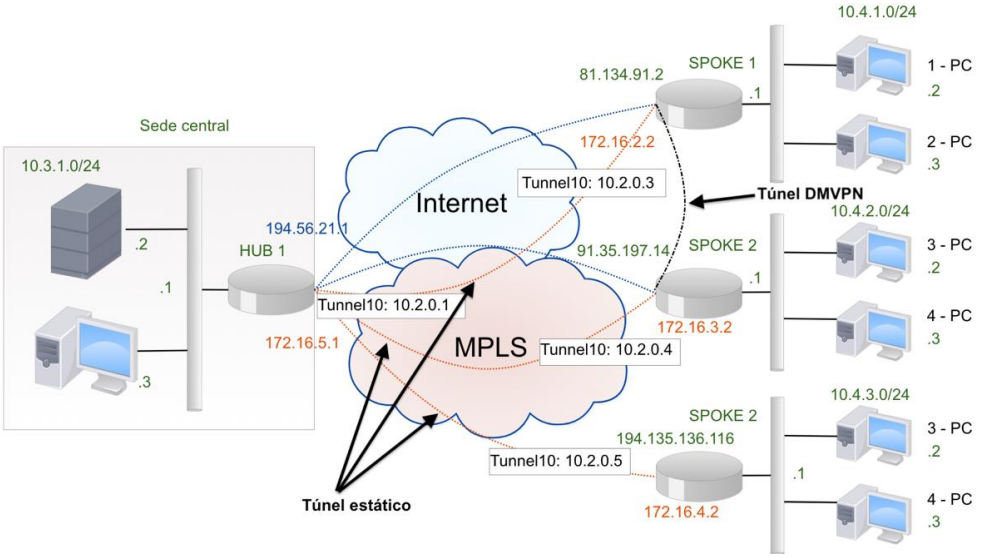


Figura 50 – Routing. Creación de túneles dinámicos o DMVPN[68]

Para comprobar la generación de dicho túnel dinámico, realizaremos una petición ICMP procedente del SPOKE1 al túnel principal del SPOKE2, utilizando el comando `trace`:

[68] Fuente: elaboración propia

```
SPOKE1#trace 10.2.0.4
Type escape sequence to abort.
Tracing the route to 10.2.0.4
VRF info: (vrf in name/id, vrf out name/id)
  1 10.2.0.1 84 msec
    10.2.0.4 144 msec 44 msec
SPOKE1#trace 10.2.0.4
Type escape sequence to abort.
Tracing the route to 10.2.0.4
VRF info: (vrf in name/id, vrf out name/id)
  1 10.2.0.4 92 msec 40 msec 32 msec
```

En la imagen previa, observamos que el tráfico circula, en primer lugar, por el HUB en la Fase 1 (10.2.0.1) hasta llegar a su destino (10.2.0.4). Tras ello, se generará un enlace dinámico entre el SPOKE1 y el SPOKE2 (letra 'D' de la etiqueta *Attrb*):

```
SPOKE1#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel1, IPv4 NHRP Details
Type:Spoke, NHRP Peers:3,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 172.16.5.1            10.2.0.1    UP 00:11:53     S
     1 172.16.3.2            10.2.0.4    UP 00:00:49     D
```

Al ser un comunicación creada bajo demanda, en la tabla NHRP existe un tiempo de expiración, en este caso es de 04:35 minutos, que indica el tiempo restante que la comunicación SPOKE-SPOKE permanecerá activa:

```
SPOKE1#sh ip nhrp
10.2.0.1/32 via 10.2.0.1
   Tunnel10 created 00:17:31, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.1
10.2.0.4/32 via 10.2.0.4
   Tunnel10 created 00:00:25, expire 00:04:35
   Type: dynamic, Flags: router used
   NBMA address: 172.16.3.2
```

En estas dos casuísticas han ocurrido procesos de petición y registro de los routers pertenecientes a la misma red NHRP (encaminadores cuyo identificador NHRP es el mismo). Los SPOKES han sido configurados con la dirección NBMA del HUB como servidor de próximo salto (NHS) para poder ser registrados. En la captura

de pantalla mostrada, se observan solicitudes de registro (*NHRP Registration Request)* por parte de los SPOKES al HUB, que contienen su dirección NBMA y la dirección privada de la interfaz túnel). Después, el HUB genera una entrada en su tabla NHRP y devuelve una respuesta de registro (*NHRP Registration Reply*):

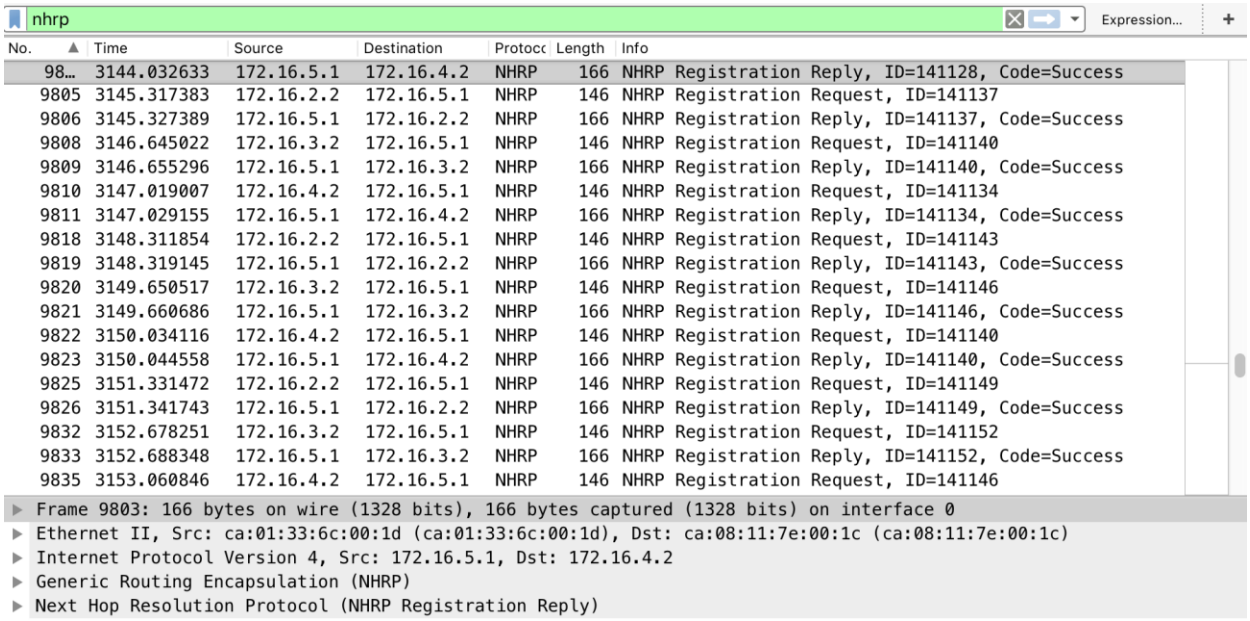


Figura 51 – Routing. Wireshark. NHRP Registration Reply y NHRP Request[69]

A partir de este momento, el SPOKE considera al HUB como servidor válido y lo utilizará para averiguar las direcciones de otros SPOKES.

5.3.1.5 COMPROBACIONES DE FRAGMENTACIÓN IP

El encaminamiento ágil de los routers es un aspecto crucial para el buen rendimiento de las comunicaciones IP. Para ello, es necesario evitar la fragmentación IP. En este apartado, comprobaremos si la normativa RFC 1191 está presente en nuestra simulación y analizaremos el tráfico de red. Para ello, ejecutaremos una solicitud ping, estableciendo un tamaño de paquete de 1472 bytes, desde el 1-CPD (sede central) hasta 1-PC (oficina tipo A):

[69] Fuente: elaboración propia

```
1-CPD console is now available... Press RETURN to get started.
root@1-CPD:~# ping 10.4.1.2 -s 1472
PING 10.4.1.2 (10.4.1.2) 1472(1500) bytes of data.
From 10.3.1.1 icmp_seq=1 Frag needed and DF set (mtu = 1400)
1480 bytes from 10.4.1.2: icmp_seq=3 ttl=62 time=83.1 ms
1480 bytes from 10.4.1.2: icmp_seq=4 ttl=62 time=59.6 ms
1480 bytes from 10.4.1.2: icmp_seq=5 ttl=62 time=73.8 ms
1480 bytes from 10.4.1.2: icmp_seq=6 ttl=62 time=42.1 ms
1480 bytes from 10.4.1.2: icmp_seq=7 ttl=62 time=63.0 ms
```

De dicha solicitud, observamos la siguiente sentencia: "`From 10.3.1.1 icmp_seq=1 Frag needed and DF set (mtu = 1400)`". También, visualizaremos los siguientes mensajes en Wireshark en la LAN interna de la sede central (10.3.1.0/24):

```
No.   Time         Source      Destination  Protocol  Length  Info
  398 135.930788   10.3.1.2    10.4.1.2     ICMP        1514  Echo (ping) request  id=0x0051, seq=1/256, ttl=64 (no response found!)
  399 135.944013   10.3.1.1    10.3.1.2     ICMP          70  Destination unreachable (Fragmentation needed)
▶ Frame 398: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
▶ Ethernet II, Src: 06:29:d2:f2:87:41 (06:29:d2:f2:87:41), Dst: All-HSRP-routers_01 (00:00:0c:07:ac:01)
▼ Internet Protocol Version 4, Src: 10.3.1.2, Dst: 10.4.1.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xe45d (58461)
  ▼ Flags: 0x02 (Don't Fragment)
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x3ab9 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.3.1.2
    Destination: 10.4.1.2
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▶ Internet Control Message Protocol
```

Figura 52 – Routing. Wireshark. Bit Don't Fragment activo. Fragmentación IP.[70]

Cuando el terminal 1-CPD envía el paquete TCP, que tiene que ser fragmentado, al atravesar una red intermedia, el router que tiene que llevar a cabo la fragmentación no puede hacerlo ya que se encuentra activo el bit "*Don't Fragment*" (nº 398). Acto seguido, dicho router envía al origen del paquete TCP (1-CPD) el mensaje "*ICMP Destination Unreachable*" (nº 399), que es el correspondiente al código "*Fragmentation needed and the Bit Don't Fragment was set*". En la cabecera ICMP de este mensaje se indica cuál es el valor del MTU de la red que necesita fragmentar el paquete TCP, indicándolo en el campo "*MTU of Next Hop*":

[70] Fuente: elaboración propia

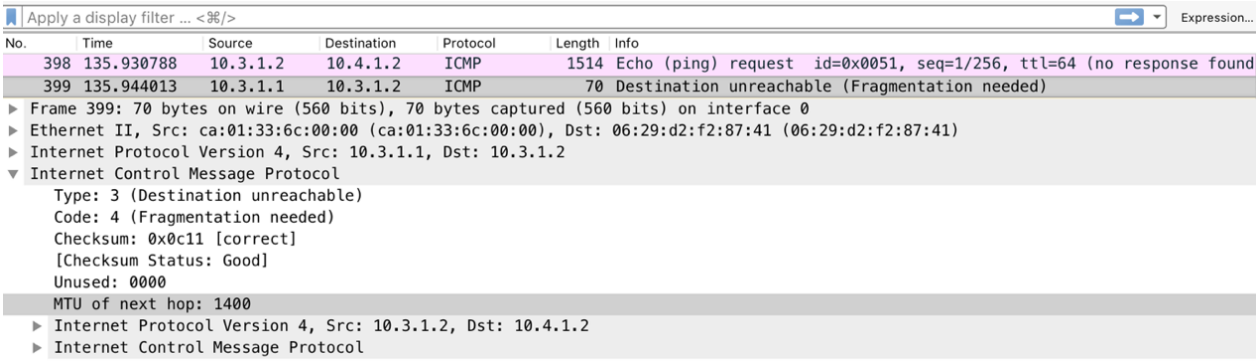


Figura 53 – Routing. MTU of Next Hop[71]

Con este valor de MTU intermedio, la estación origen determina el nuevo valor de MSS de la conexión y reenvía el paquete TCP que no ha alcanzado al destino. Por tanto, mediante este procedimiento, podremos conseguir ajustar dinámicamente la cantidad de datos introducida en cada paquete TCP para prevenir que los paquetes sean fragmentados.

5.3.2 HSRP

Crearemos la dirección IP virtual del grupo HSRP 10.3.1.100 para que, en primera instancia, el tráfico que circule hacia esta dirección sea encaminado por el router HSRP activo (HUB1). El router HUB1-backup será configurado para que éste tenga menor prioridad, permaneciendo en reposo, y será activado cuando el protocolo detecte que el router activo falle.

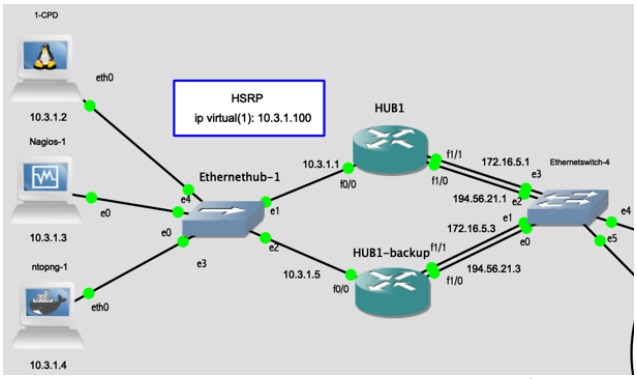


Figura 54 – Routing. HSRP en el Piloto[72]

[71] Fuente: elaboración propia
[72] Fuente: elaboración propia

5.3.2.1 CONFIGURACIÓN

Aplicaremos estos comandos en las interfaces LAN de los routers HUB1 y HUB1-backup:

HUB1	HUB1-backup
interface FastEthernet0/0	interface FastEthernet0/0
ip address 10.3.1.1 255.255.255.0	ip address 10.3.1.3 255.255.255.0
standby 1 ip 10.3.1.100 (1)	ip nat inside
standby 1 priority 110 (2)	**standby 1 ip 10.3.1.100 #1**
standby 1 preempt (3)	**standby 1 preempt #2**
duplex full	duplex full

Tabla 19 – Routing. Configuración de HSRP[73]

Mediante los tres comandos aplicados, (1) estamos definiendo la IP virtual del grupo de interfaces 1, (2) identificamos la prioridad del router (el valor predeterminado es 100) y (3) obligamos al encaminador de mayor prioridad a convertirse en el activo, en lugar del router activo en ese momento.

5.3.2.2 VALIDACIÓN

Dispondremos de un escenario en donde estarán activos el HUB1, SPOKE1, SPOKE2, SPOKE3 y HUB1-backup; así como un terminal activo de la sede central y las oficinas (1-CPD y 1-PC). Para esta comprobación en concreto, utilizaremos la WAN MPLS (R2), y por tanto el túnel principal (Tunnel10):

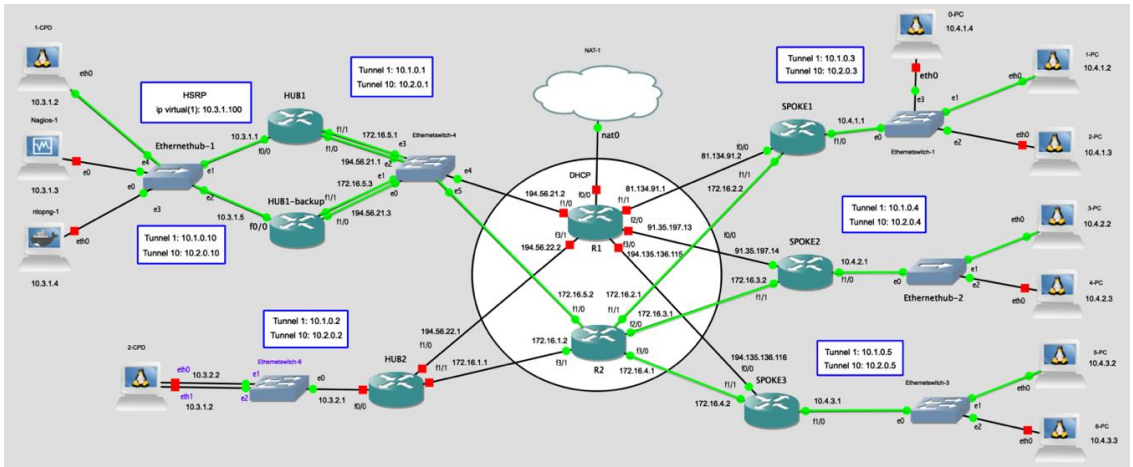
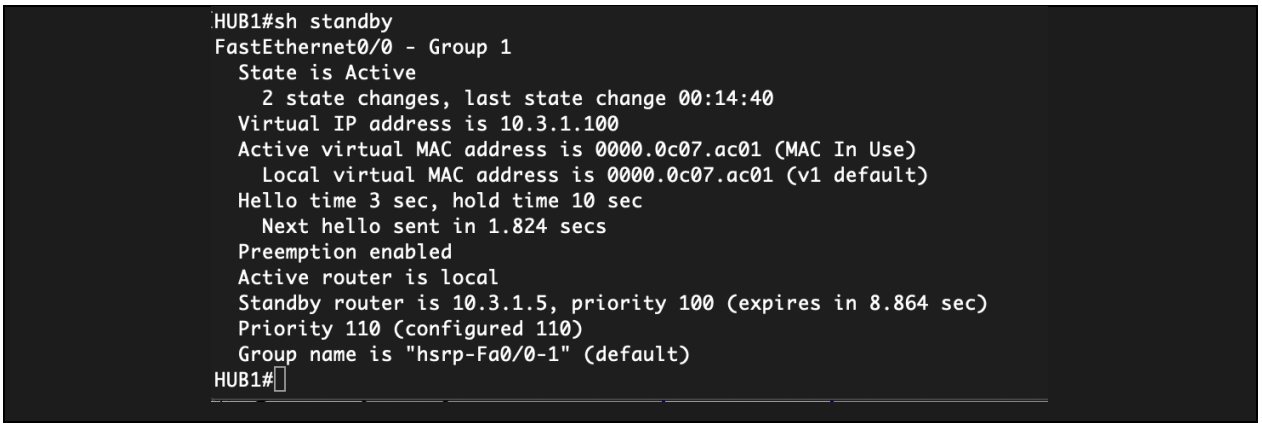


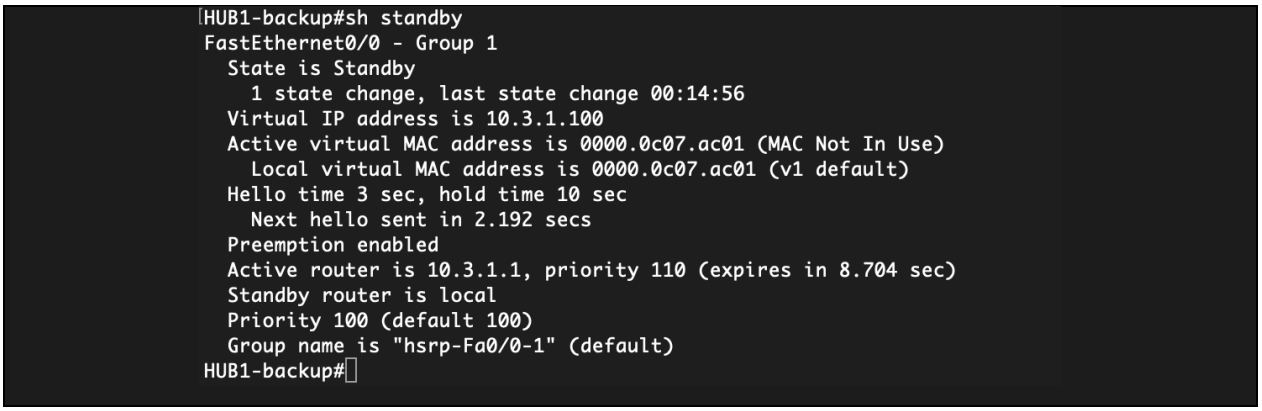
Figura 55 – Routing. Validación HSRP[74]

[73] Fuente: elaboración propia
[74] Fuente: elaboración propia

Utilizaremos el comando `show standby` para verificar que el HUB1 es el router principal debido a que su prioridad es mayor que el router secundario de la sede central (`priority 110`):



Del mismo modo, teclearemos el mismo comando en el HUB1-backup para observar que el router se encuentra en estado pasivo o Standby:



De las dos capturas de pantalla previas, observamos que para cada grupo HSRP, se han reservado una única dirección IP y MAC. Concretamente, la dirección MAC empleada por un grupo HSRP está normalizada al valor **00:00:0C:07:AC:XX** donde XX representa el número del grupo HSRP, que en este caso corresponde al valor **00:00:0C:07:AC:01.**

Al analizar el tráfico en la LAN interna del HUB1 y HUB1-backup (10.3.1.0/24), comprobamos que los participantes del grupo HSRP envían y reciben paquetes de tipo *Hello* al puerto UDP 1985 con dirección IP Multicast 224.0.0.2.

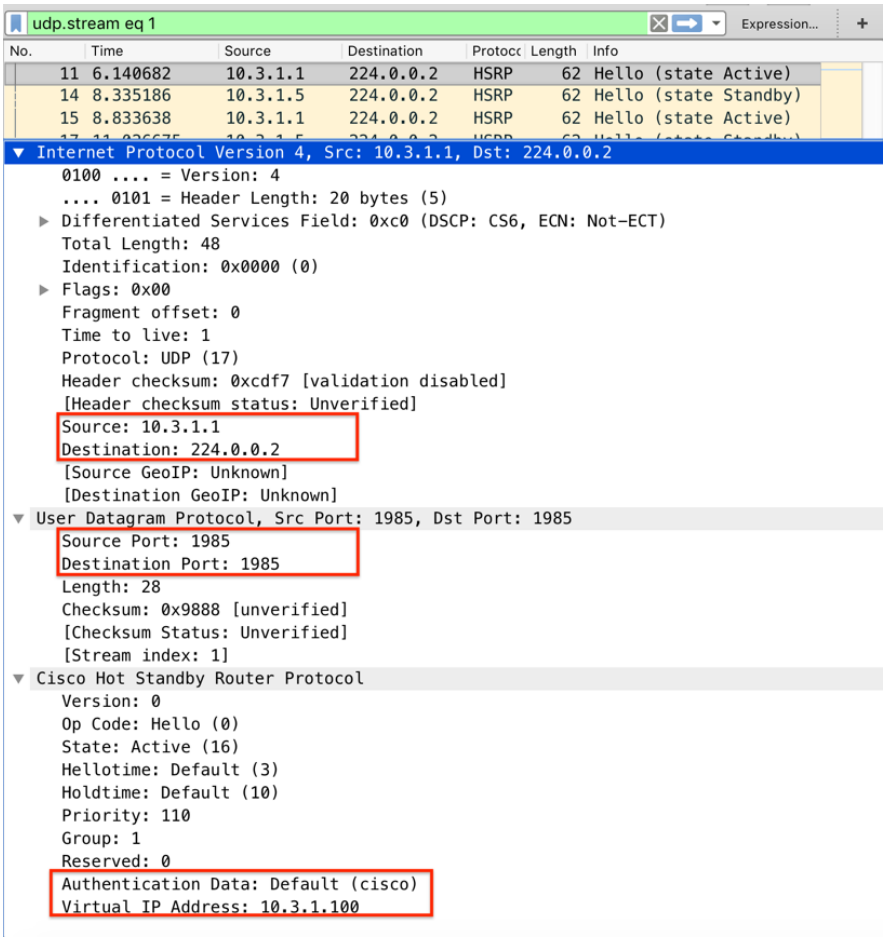


Figura 56 – Routing. Wireshark. Envío de paquetes Hello del grupo HSRP[75]

El estado **Active** significa que el router se encuentra procesando paquetes dirigidos a la dirección MAC virtual y envía periódicamente mensajes HELLO. Por otra parte, el estado **Standby** indica que el router HUB1-backup es candidato a convertirse en router activo y envía periódicamente mensajes HELLO.

Seguidamente simularemos que el ordenador 1-PC (10.4.1.2), perteneciente a la oficina del tipo A (SPOKE1), está comprobando la conectividad con el CPD de la oficina central (10.3.1.2) utilizando el comando `traceroute`. Evidentemente, se ha asignado la dirección IP virtual 10.3.1.100 como puerta de enlace de la LAN a todos los equipos ubicados en la sede central y se han configurado los túneles principales y secundarios, 10.2.0.10 y 10.1.0.10 respectivamente, en el router HUB1-backup (ver

[75] Fuente: elaboración propia

configuración HUB1-backup en el Anexo). Por tanto si no hay ninguna caída en el router principal, se realizarán los siguientes saltos:

```
root@1-PC:~# traceroute 10.3.1.2
traceroute to 10.3.1.2 (10.3.1.2), 30 hops max, 60 byte packets
 1  10.4.1.1 (10.4.1.1)  72.458 ms  72.872 ms  72.870 ms
 2  10.2.0.1 (10.2.0.1)  114.464 ms  124.486 ms  124.492 ms
 3  10.3.1.2 (10.3.1.2)  124.489 ms  124.550 ms  134.540 ms
root@1-PC:~#
```

Imaginemos ahora que se produce una caída en el HUB1, teniendo el siguiente escenario:

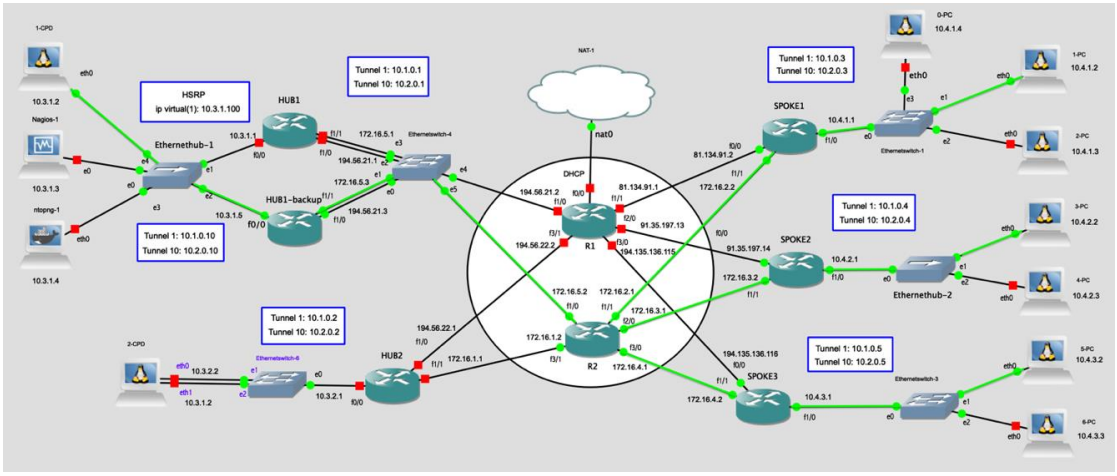


Figura 57 – Routing. Caída del HUB1[76]

El protocolo HSRP detecta esta situación (mensaje nº 106 *Advertise*) y el router HUB1-backup cambia a estado activo (mensaje nº 107 *Hello State Active*):

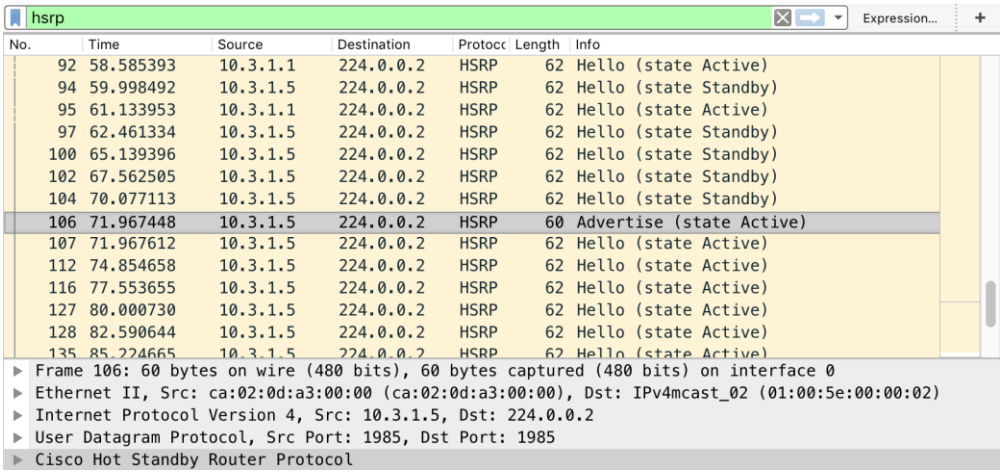


Figura 58 – Routing. Wireshark. Actuación del protocolo HSRP I[77]

[76] Fuente: elaboración propia
[77] Fuente: elaboración propia

Si se ejecuta nuevamente el comando `traceroute` en 1-PC; observamos que la ruta de saltos ahora circula por el túnel principal del HUB1-backup:

```
root@1-PC:~# traceroute 10.3.1.2
traceroute to 10.3.1.2 (10.3.1.2), 30 hops max, 60 byte packets
 1  10.4.1.1 (10.4.1.1)  16.147 ms  17.056 ms  17.054 ms
 2  10.2.0.10 (10.2.0.10)  66.616 ms  129.198 ms  129.193 ms
 3  10.3.1.2 (10.3.1.2)  129.155 ms  129.168 ms  129.164 ms
root@1-PC:~#
```

En caso de recuperación del router primario de la sede central, nuevamente, el protocolo actuará (mensaje nº 146) y seleccionará el HUB1 como *Active*:

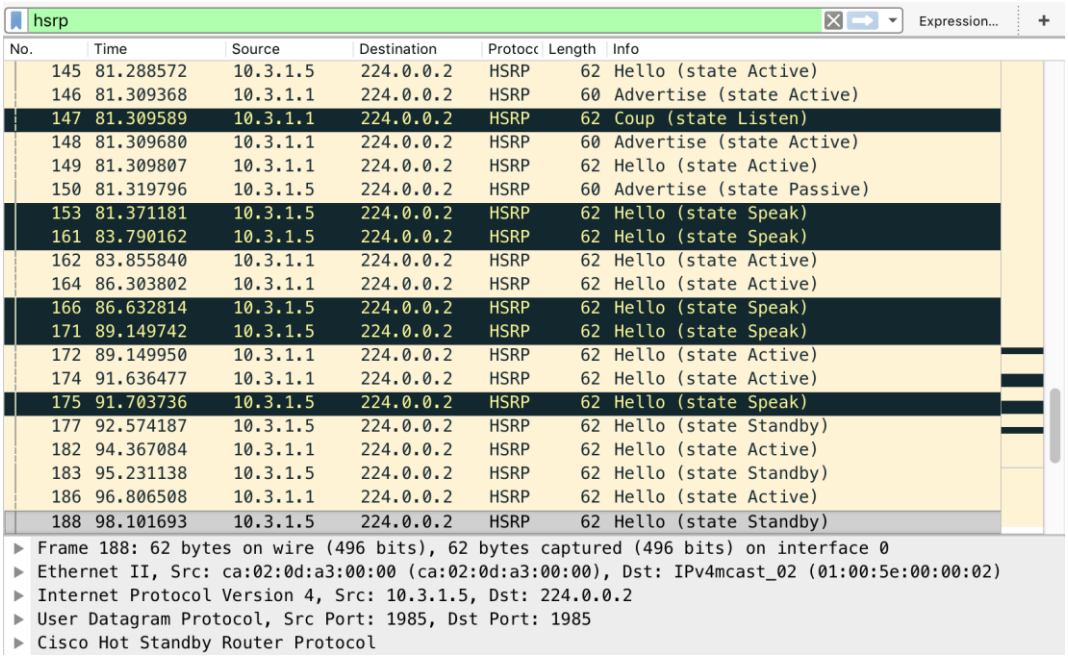


Figura 59 – Routing. Wireshark. Actuación del protocolo HSRP II[78]

Se observan otro tipo de mensajes como "*Coup (state **Listen**)*", enviado por el *router* de mayor prioridad del grupo HSRP que indica dicho router conoce la dirección IP virtual, pero ni es router Activo ni Standby en ese momento, ya que se encuentra a la escucha de mensajes Hello de esos routers. Se aprecia también el estado **Speak**, para el envío periódico de mensajes Hello y participación activa en la elección del router Activo y Standby.

Si volvemos a ejecutar un `traceroute` en el 1-PC, observaremos que el tráfico, vuelve a circular, nuevamente, por el túnel principal del HUB1:

[78] Fuente: elaboración propia

```
root@1-PC:~# traceroute 10.3.1.2
traceroute to 10.3.1.2 (10.3.1.2), 30 hops max, 60 byte packets
 1  10.4.1.1 (10.4.1.1)  12.334 ms  13.914 ms  13.911 ms
 2  10.2.0.1 (10.2.0.1)  54.552 ms  54.561 ms  54.393 ms
 3  10.3.1.2 (10.3.1.2)  43.739 ms  54.389 ms  54.386 ms
root@1-PC:~#
```

Existen otros dos estados en los que un router que pertenece a un grupo HSRP se puede encontrar: Initial y Learn. El estado **Initial**, indica que HSRP no se encuentra funcionando en el router. Dicho estado puede surgir cuando un interfaz arranca por primer vez o por cambios en las configuraciones. El estado **Learn**, que indica que el router no ha determinado la dirección IP virtual y aún no ha recibido mensajes tipo Hello proveniente de un Router activo.

5.3.3 CONECTIVIDAD A INTERNET

5.3.3.1 CONFIGURACIÓN

Internet será simulado utilizando el router R1. Previamente, se deberá haber configurado los túneles secundarios (10.1.0.0/24) en los HUBs y SPOKES. Después, configuraremos NAT en todos los routers, excepto en R2 (MPLS). La idea básica del funcionamiento de NAT consiste en la traducción de direcciones IP privadas de una red local en una dirección IP pública, posibilitando de esta manera, el envío de paquetes a Internet y del mismo modo también traducir dicha dirección IP pública a la dirección IP privada del equipo que envió el paquete.

Es posible emplear otros métodos simultáneamente con NAT, como PAT, que consiste en asignar una única dirección pública a todos los equipos de una misma red privada. A la traducción de direcciones NAT, que emplea también PAT, se denomina NAT por sobrecarga. Para proceder con su configuración en cada red privada de la simulación, primeramente definimos las interfaces que se conectan a cada red local (puerta de enlace) y las salidas, que son las que conectan con Internet:

	Red	Dirección IP	Máscara	**NAT**	Anotaciones
SPOKES	A	f0/0: 81.134.91.2	255.255.255.0	***outside***	*Salida a Internet*
		f1/0: 10.4.1.1	255.255.255.0	***inside***	*Puerta de enlace*
	B	f0/0: 91.35.197.14	255.255.255.0	***outside***	*Salida a Internet*
		f1/0: 10.4.2.1	255.255.255.0	***inside***	*Puerta de enlace*
	C	f0/0: 194.135.136.116	255.255.255.0	***outside***	*Salida a Internet*
		f1/0: 10.4.3.1	255.255.255.0	***inside***	*Puerta de enlace*
HUB1	Sede central	f0/0: 10.3.1.1	255.255.255.0	***inside***	
		f1/0: 194.56.21.1	255.255.255.0	***outside***	*Salida a Internet*
HUB1-backup		f0/0: 10.3.1.5	255.255.255.0	***inside***	
		f1/0: 194.56.21.3	255.255.255.0	***outside***	*Salida a Internet*
HUB2	Centro de respaldo	f0/0: 10.3.2.1	255.255.255.0	***inside***	
		f1/0: 194.56.22.1	255.255.255.0	***outside***	*Salida a Internet*
R1	WAN1 Internet	f0/0: DHCP		***outside***	*Wifi*
		f1/0: 194.56.21.2	255.255.255.0	***inside***	
		f1/1: 81.134.91.1	255.255.255.0	***inside***	
		f2/0: 91.35.197.13	255.255.255.0	***inside***	*Internet*
		f3/0: 194.135.136.115	255.255.255.0	***inside***	
		f3/1: 194.56.22.2	255.255.255.0	***inside***	

Tabla 20 – Routing. Configuración de NAT por sobrecarga en las interfaces[79]

Después, definimos una lista de acceso con las direcciones IPs privadas que queremos que se traduzcan. En este ejemplo, nos centralizaremos en la configuración de NAT estático utilizando el SPOKE A (ver configuración NAT en la configuración de los Routers del presente documento).

```
SPOKEA# access-list 1 permit 10.4.1.0 0.0.0.255
```

Después, definiremos el NAT en sí, indicando las direcciones IPs a traducir (lista de IPs que se han creado) y a continuación, indicamos la IP pública a traducir. Finalmente, escribimos la palabra **`overload`** al final de la línea:

```
SPOKEA# ip nat inside source list 1 interface FastEthernet0/0 overload
```

[79] Fuente: elaboración propia

De manera similar y cambiando la dirección IP de la puerta de enlace y la interfaz de salida a Internet, se procede a la configuración de PAT con el resto de routers. Para finalizar se ha incluido un componente GNS3, denominado NAT-1 (representado por una nube), para integrar la conexión WiFi del ordenador con el montaje del Piloto en GNS3. La interfaz `f0/0` está conectada a esta "nube" utilizando DHCP para la asignación dinámica de una dirección IP. Además, si deseamos asignar el servidor DNS 8.8.8.8 en cada router (excepto en R2), teclearemos los comandos:

```
ip domain lookup
ip name-server 8.8.8.8
```

Finalmente, para que los terminales tengan conexión a internet se deberá insertar el mismo servidor DNS (`nameserver`) en el archivo de configuración `/etc/resolv.conf`. Se muestra la asignación de la dirección IP estática del 1-PC correspondiente a la LAN de la oficina tipo A (SPOKE 1). La configuración del resto de terminales será idéntica excepto la dirección IP (Address) y la puerta de enlace (Gateway), dependiendo de la red dónde se ubique:

```
# Static config for eth0
auto eth0
iface eth0 inet static
       address 10.4.1.2
       netmask 255.255.255.0
       gateway 10.4.1.1
       up echo nameserver 8.8.8.8 > /etc/resolv.conf
```

5.3.3.2 VALIDACIÓN

Los enlaces de comunicación marcados en verde de la imagen mostrada a continuación, reflejan la conexión a Internet de cada red privada:

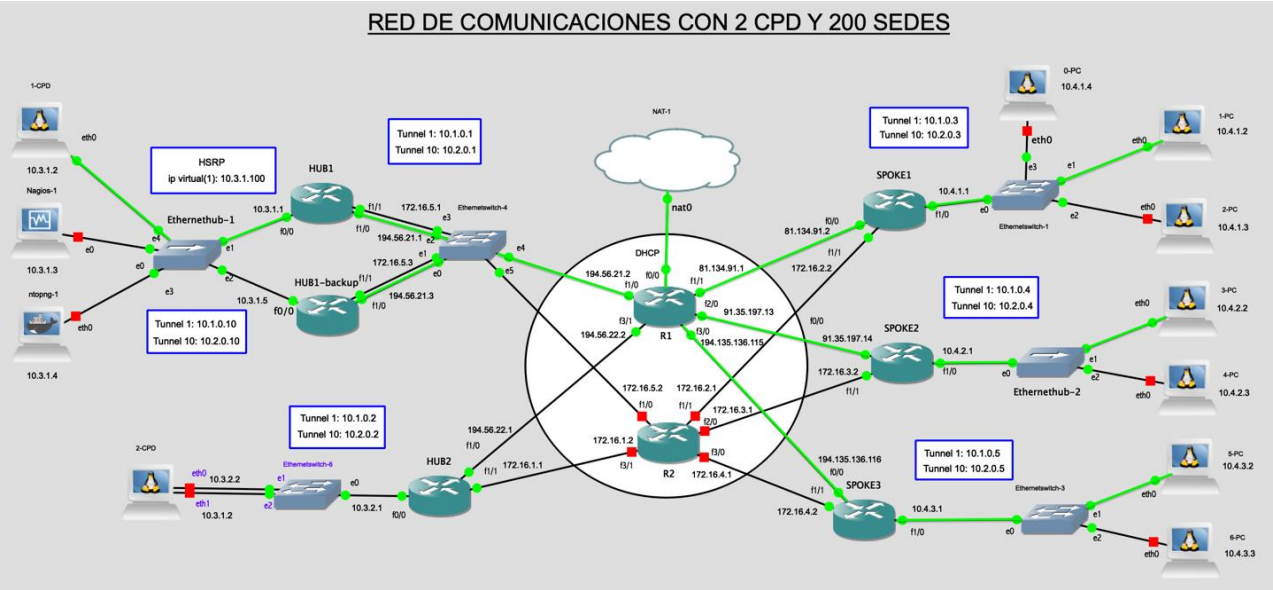


Figura 60 – Routing. Conectividad a Internet[80]

Una manera de comprobar la conectividad con Internet es realizando una solicitud ping, por ejemplo, al Google Public DNS desde 1-PC (SPOKE1):

```
root@1-PC:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=125 time=59.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=125 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=125 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=125 time=65.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=125 time=56.0 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 43.916/53.729/65.065/8.481 ms
```

Incluso podemos probar la conectividad a otros dominios, como www.cisco.com:

```
root@1-PC:~# ping www.cisco.com
PING e2867.dsca.akamaiedge.net (23.37.160.19) 56(84) bytes of data.
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=1 ttl=125 time=52.5 ms
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=2 ttl=125 time=48.1 ms
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=3 ttl=125 time=45.6 ms
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=4 ttl=125 time=70.8 ms
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=5 ttl=125 time=36.7 ms
64 bytes from a23-37-160-19.deploy.static.akamaitechnologies.com (23.37.160.19): icmp_seq=6 ttl=125 time=50.0 ms
^C
--- e2867.dsca.akamaiedge.net ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 36.735/50.681/70.813/10.289 ms
```

[80] Fuente: elaboración propia

5.3.4 IP SECUNDARIA EN EL CENTRO DE RESPALDO

Imaginemos la situación donde el router del centro de respaldo (HUB2), los dos routers de la sede central, los SPOKES y la WAN MPLS se encuentran activos:

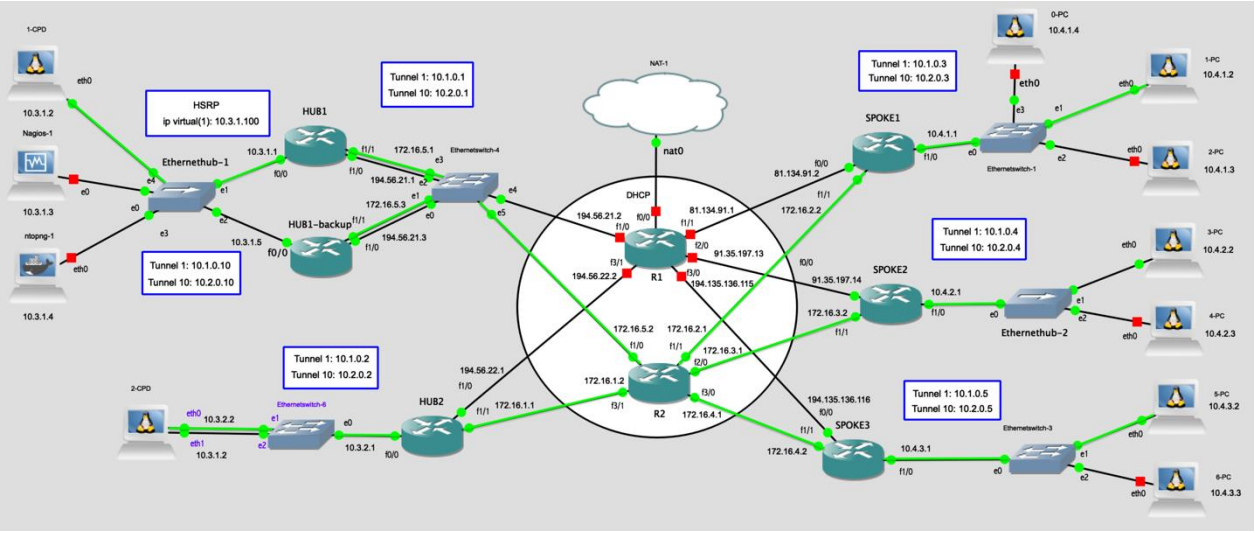


Figura 61 – Routing. Conectividad con HUB2[81]

Como ya hemos comentado en las secciones previas, todo el tráfico de red procedente de cualquier oficina hasta el CPD de la sede central y viceversa utilizará el Túnel principal (10.2.0.0/24). En caso de que el HUB1 falle, el protocolo HSRP actuará y el tráfico circulará por el túnel principal (10.2.0.10) del router HUB1-backup en lugar del túnel principal configurado en el HUB1. Ahora bien, podría darse la situación de que ambos routers fallen y que los terminales de las sedes distantes (SPOKEs) estén realizando peticiones al CPD principal. En este caso, las oficinas ya no tendrían conectividad con dicho CPD y los trabajadores no podrían seguir operando.

Para evitar esta situación y proporcionar mayores presentaciones en cuanto a alta disponibilidad, el CPD de respaldo (2-CPD) del HUB2 tiene asignado dos direcciones IPs: principal y secundaria. La interfaz `eth0` contiene la dirección IP 10.3.2.2; mientras que la interfaz de red `eth1` está asociada a la dirección IP 10.3.1.2. Por tanto, mediante la inclusión de esta dirección IP secundaria en el CPD de

[81] Fuente: elaboración propia

respaldo, el tráfico de red procedente de las oficinas al CPD principal (1-CPD), bascularía al CPD secundario sin que exista ninguna parada de servicio, utilizando el túnel principal del HUB2 (10.2.0.2). A continuación se muestra la configuración de las direcciones IP estáticas del CPD de respaldo:

```
auto eth0
iface eth0 inet static
        address 10.3.2.2
        netmask 255.255.255.0
        broadcast 10.3.2.255
        gateway 10.3.2.1
        up echo nameserver 8.8.8.8 > /etc/resolv.conf
auto eth1
iface eth1 inet static
        address 10.3.1.2
        netmask 255.255.255.0
        broadcast 10.3.1.255
```

En la siguiente captura de pantalla podemos apreciar que se han ejecutado tres `traceroute`. En la primera ejecución los dos routers de la sede central y el router del centro de respaldo estaban activos. En la segunda ejecución se ha parado el router principal de la sede central. En la tercera y última ejecución, los dos routers de la sede central están fuera de servicio, operando únicamente el HUB2 del centro de respaldo:

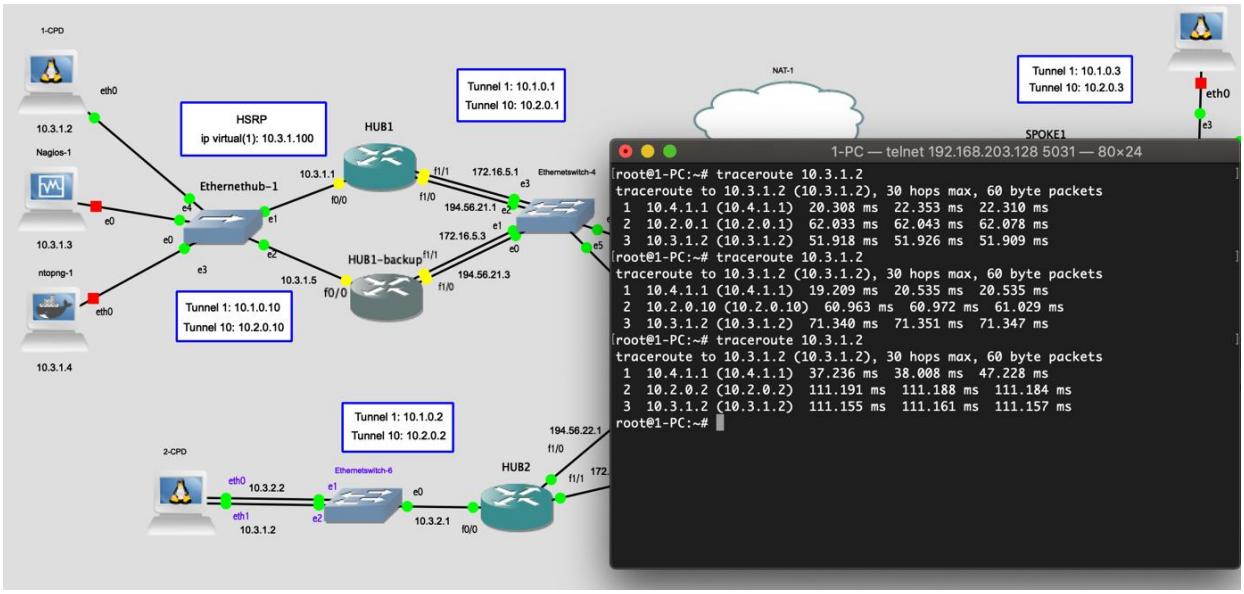


Figura 62 – Routing. Ruta seguida para alcanzar el centro de respaldo (IP secundaria)[82]

[82] Fuente: elaboración propia

5.3.5 CENTRALIZACIÓN DE SYSLOGS EN LOS CPDS

Por otro lado, la red cuenta también con una **centralización de *syslogs***. Los syslogs se caracterizan por ser registros de eventos enviados por un router o un servidor externo a un equipo en concreto para tener un censo de eventos del sistema; o bien llevar a cabo actividades tales como la correlación de eventos y detectar cambios de configuraciones, recepción de errores de hardware o software… En particular, los mensajes de *syslogs* generados en la topología de red propuesta informarán sobre alertas de cambios de estados (notificaciones) y condiciones de errores en el sistema (caída de un túnel, interfaz, router…) Para ello, se configurarán los enrutadores para enviar mensajes de *syslog* a la máquina 1-CPD (10.3.1.2) y a 2-CPD (10.3.2.2). A continuación se reflejan los comandos introducidos en el router SPOKE1 (esta configuración se aplica, de igual modo al resto de routers):

```
logging facility local0
 logging host 10.3.1.2
 logging host 10.3.2.2
```

Los mensajes de `syslogs` utilizan el protocolo UDP al puerto 514. Por tanto, para comprobar su recepción en el CPD situado en la red LAN de la sede central, ejecutaremos la siguiente sentencia `tcpdump`:

```
tcpdump -nXi eth0 udp port 514
```

Donde:

- ⇒ `n`: indica que no intente resolver nombres a las direcciones IP.
- ⇒ `X`: muestra el contenido del paquete.
- ⇒ `-i <interfaz>`: especifica la interfaz sobre la que se realizará la captura.
- ⇒ `udp port 514`: selecciona los paquetes UDP con el puerto 1234.

La captura de pantalla mostrada a continuación muestra que el terminal 1-CPD está recibiendo todos los *syslogs* generados por los routers. En este caso, se ha entrado al modo configuración de los SPOKES tecleando el comando configure

terminal (`conf t`) y finalizando la sesión (`end`). Esta acción genera un mensaje, que es enviada al CPD principal, en este caso:

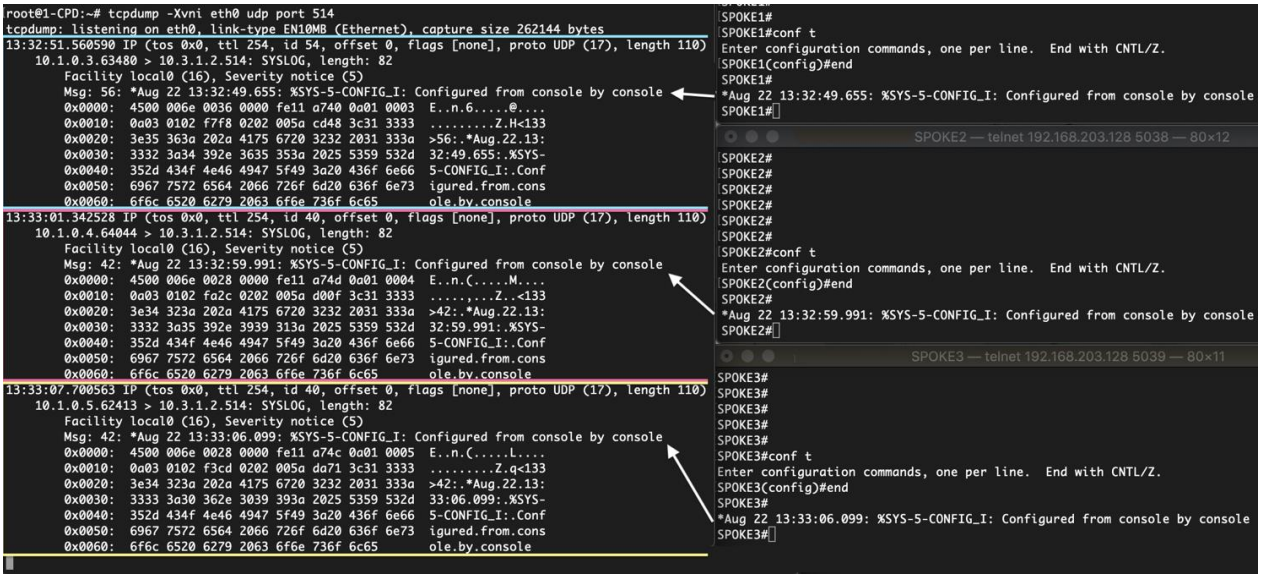


Figura 63 – Routing. Envío de syslogs al CPD principal[83]

La cantidad y variedad de información centralizada en los dos CPDs puede ser abundante si tenemos en cuenta que cualquier mensaje procedente de los routers es enviada a los CPDs: caídas y levantamiento de túneles y enlaces, mensajes de configuración, alertas...

5.3.6 ROUTING FINAL

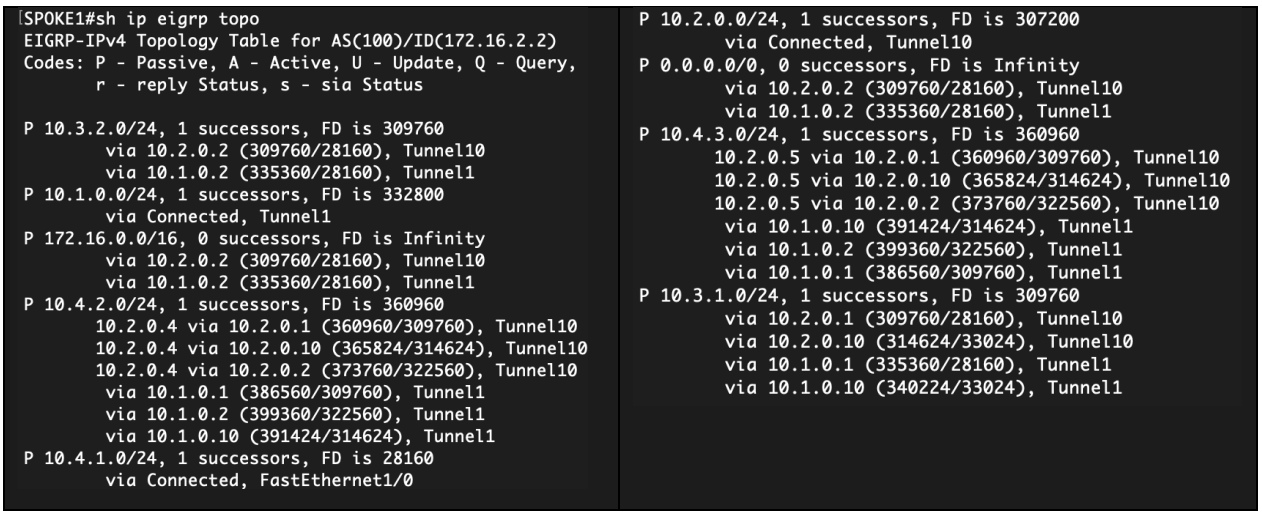
En el Anexo se encuentran todas configuraciones de cada HUB y SPOKE presente en la simulación y que una vez introducidas en la simulación obtendríamos el *Routing* final.

5.3.6.1 VERIFICACIÓN DE LA CONECTIVIDAD

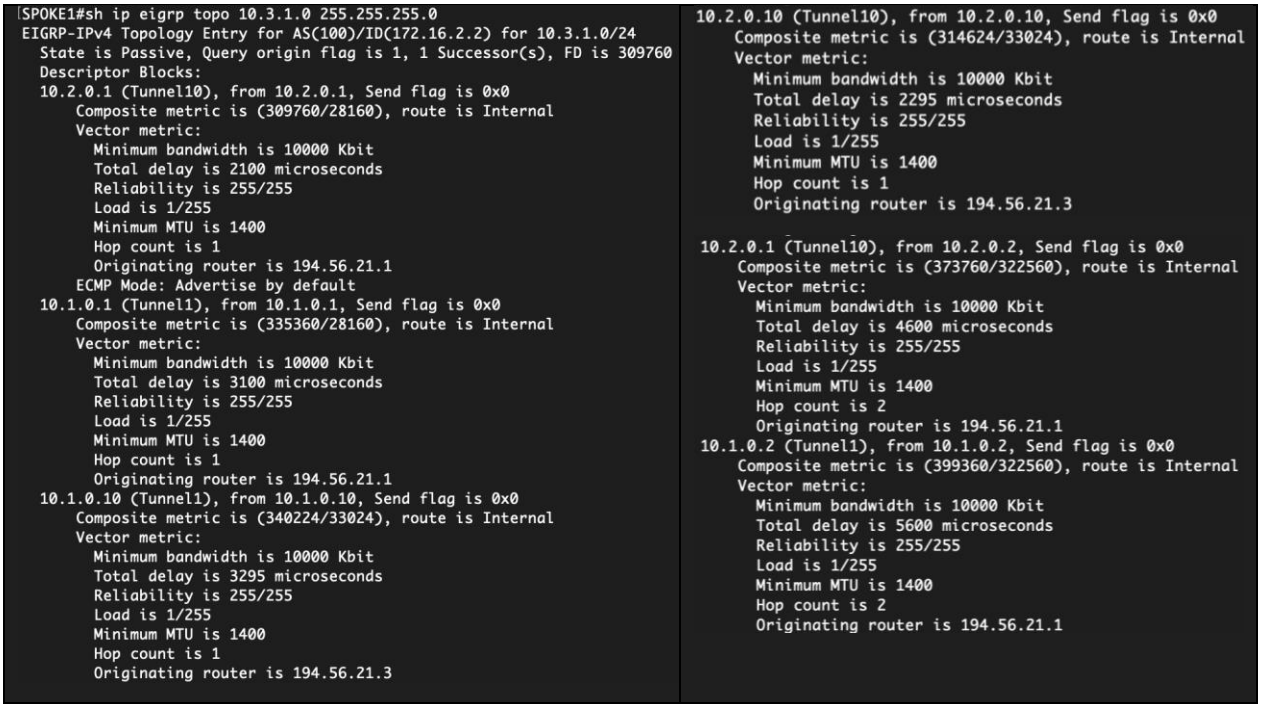
En condiciones normales, el tráfico circulará por la MPLS utilizando el Tunnel10, dejando a la otra WAN como secundaria, y que en principio se utilizaría para la navegación por Internet de los usuarios y actualizaciones automáticas del SO o aplicaciones. En caso de fallo de algún enlace o túnel intervendría el protocolo

[83] Fuente: elaboración propia

EIGRP, que haría uso de la tabla de topología correspondiente, para establecer una ruta alternativa y llegar al destino propuesto. La siguiente imagen hace referencia a la tabla de topología del SPOKE1:



Apreciamos que para llegar a red 10.3.1.0/24 (sede central) existen 4 caminos: los dos túneles principales (10.2.0.1 y 10.2.0.10), los dos túneles secundarios (10.1.0.1 y 10.1.0.10) de los routers HUB1 y HUB1-backup y el túnel principal. También, observamos, que tienen 4 valores de *Composite Metric* diferentes debido a los valores de ancho de banda y retardo:



$$metric = \left(\frac{10^7}{BW_{min}} + \sum delays\right) \cdot 256$$

$$metric\ (10.2.0.1) = \left(\frac{10^7}{10.000} + 210\right) \cdot 256 = \mathbf{309.760}$$

$$metric\ (10.2.0.10) = \left(\frac{10^7}{10.000} + 229\right) \cdot 256 = \mathbf{314.624}$$

$$metric\ (10.1.0.1) = \left(\frac{10^7}{10.000} + 310\right) \cdot 256 = \mathbf{335.360}$$

$$metric\ (10.1.0.10) = \left(\frac{10^7}{10.000} + 329\right) \cdot 256 = \mathbf{340.224}$$

La *Feasible Distance* será el menor valor de las 4 *Composite Metric* obtenidas:

$$Feasible\ Distance = 309.760$$

Por tanto, el sucesor correspondiente será la ruta 10.2.0.1. En caso de caída, el protocolo EIGRP (y teniendo también en consideración el algoritmo DUAL) basculará a la siguiente mejor ruta, que en este caso, es el túnel principal del HUB1-backup. Así pues, para contemplar los casos más importantes de caída y recuperación de los diferentes elementos de la red realizaremos pruebas de basculación, cronometrando el tiempo que la red tarda en bascular a un camino alternativo y ofrecer servicios de alta disponibilidad.

5.3.6.2 PRUEBAS DE BASCULACIÓN

Partiremos de la situación inicial en la que todos los routers de la simulación se encuentran activos y en funcionamiento. Para esta sección, debemos tener presente las tablas de topología de cada router, se muestran en la sección "Comandos Cisco iOS para comprobar la conectividad" del Anexo ya que nos indicarán el camino de menor coste para una ruta determinada. Observaremos que, en todos los casos, el Tunnel10 posee métricas de menor coste y por tanto, la comunicación se establecerá, en primer

lugar, por dicho túnel. Las mediciones se han dividido en dos casos: comunicación de SPOKEs a HUBs y la comunicación de HUB a SPOKE, teniendo en consideración las posibles caídas de enlaces, túneles y routers.

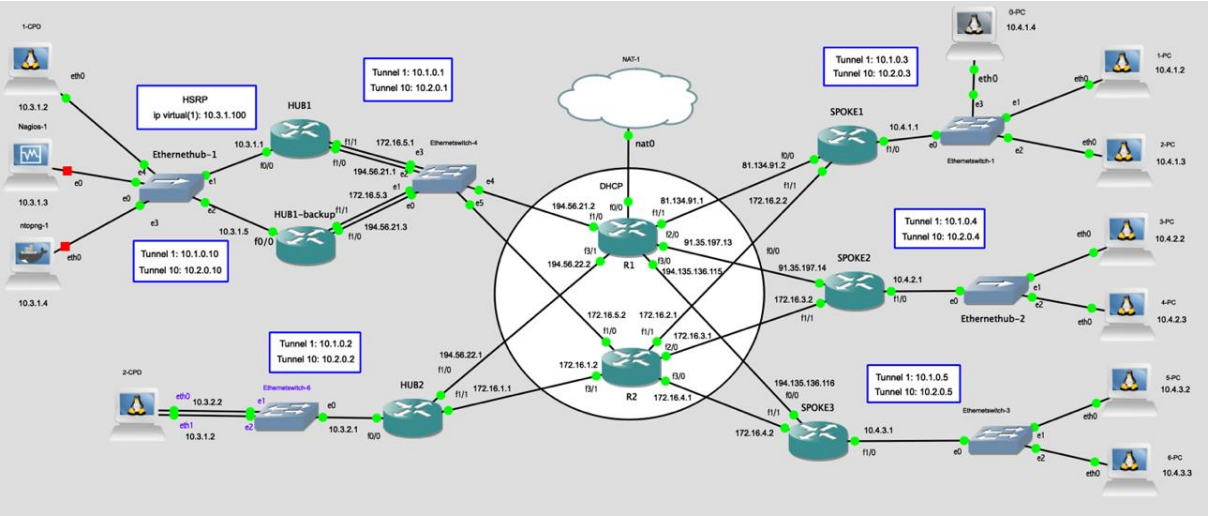


Figura 64 – Roting. Túneles y enlaces principales y secundarios activos[84]

Túnel principal: 10.2.0.0/24 *Enlace principal*: MPLS (R2)
Túnel secundario: 10.2.0.0/24 *Enlace secundario*: Internet (R1)

5.3.6.2.1 COMUNICACIÓN DE SPOKE A HUB

1. Caída de enlaces.	
1.1 Caída enlace principal HUB1.	
Resultado *Protocolo que actúa* *Tiempo*	*Basculación a túnel principal HUB1-backup.* *EIGRP / software.* *14 segundos.*
1.2 Enlace principal HUB1 caído. Se cae enlace principal HUB1-backup.	
Resultado *Protocolo que actúa* *Tiempo*	*Basculación a túnel secundario HUB1.* *EIGRP / software.* *13 segundos.*
1.3 Enlace principal HUB1 y HUB1 backup caído. Se cae enlace secundario HUB1.	
Resultado *Protocolo que actúa* *Tiempo*	*Basculación a túnel secundario HUB1-backup.* *EIGRP / software.* *12 segundos.*

[84] Fuente: elaboración propia

1.4 Enlace principal y secundario del HUB1 caído. Enlace principal HUB1 caído. Se cae enlace secundario HUB1-backup.	
Resultado	*Basculación a túnel principal HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*10 segundos.*
1.5 Enlace principal y secundario caído en HUB1 y HUB1-backup. Se cae enlace principal HUB2.	
Resultado	*Basculación a túnel secundario HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*14 segundos.*

Tabla 21 – Routing. Pruebas de basculación. Caída de enlaces[85]

2. Recuperación de enlaces.	
2.1 Enlace principal HUB1 caído. Se recupera enlace principal HUB1.	
Resultado	*Basculación de túnel principal de HUB1-backup a túnel principal HUB1.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*17 segundos.*
2.2 Enlace principal HUB1 y HUB1-backup caído. Se recupera enlace principal HUB1-backup.	
Resultado	*Basculación de túnel secundario de HUB1 a túnel principal HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*19 segundos.*
2.3 Enlace principal HUB1 y HUB1 backup caído. Enlace secundario HUB1 caído. Se recupera enlace secundario HUB1.	
Resultado	*Basculación de túnel secundario HUB1-backup a túnel secundario HUB1.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*19 segundos.*
2.4 Enlace principal y secundario del HUB1 y HUB1-backup caído. Se recupera enlace secundario HUB1-backup.	
Resultado	*Basculación de túnel principal HUB2 a túnel secundario HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*18 segundos.*

[85] Fuente: elaboración propia

2.5 Enlace principal y secundario caído en HUB1 y HUB1-backup. Enlace principal HUB2 caído. Se recupera enlace principal HUB2.	
Resultado	*Basculación de túnel secundario HUB2 a túnel principal HUB2*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*19 segundos.*

Tabla 22 – Routing. Pruebas de basculación. Recuperación de enlaces[86]

3. Caída de túneles.	
3.1 Caída túnel principal HUB1.	
Resultado	*Basculación a túnel principal HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*14 segundos.*
3.2 Túnel principal HUB1 caído. Se cae túnel principal HUB1-BACKUP.	
Resultado	*Basculación a túnel secundario HUB1.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*13 segundos.*
3.3 Túnel principal HUB1 y HUB1-backup caído. Se cae túnel secundario HUB1.	
Resultado	*Basculación a túnel secundario HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*12 segundos.*
3.4 Túnel principal y secundario HUB1 caído. Túnel principal HUB1-backup caído. Se cae túnel secundario HUB1-backup.	
Resultado	*Basculación a túnel principal HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*10 segundos.*
3.5 Túnel principal y secundarios caídos en HUB1 y HUB1 backup. Se cae túnel principal HUB2.	
Resultado	*Basculación a túnel secundario HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*15 segundos.*

Tabla 23 – Routing. Caída de túneles[87]

[86] Fuente: elaboración propia

[87] Fuente: elaboración propia

4. Recuperación de túneles.	
4.1 Túnel principal HUB1 caído. Se recupera túnel principal.	
Resultado	*Basculación de túnel principal HUB1-backup a túnel principal HUB1.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*13 segundos.*
4.2 Túnel principal HUB1 y HUB1-backup caído. Se recupera túnel principal HUB1-backup.	
Resultado	*Basculación de túnel secundario HUB1 a túnel principal HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*12 segundos.*
4.3 Túnel principal y secundario HUB1 caído. Túnel principal HUB1-backup caído. Se recupera túnel principal HUB1-backup.	
Resultado	*Basculación de túnel secundario HUB1-backup a túnel principal HUB1-backup.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*10 segundos.*
4.4 Túnel principal y secundarios caídos en HUB1 y HUB1 backup. Túnel principal HUB2. Se recupera túnel principal HUB2.	
Resultado	*Basculación de túnel secundario HUB2 a túnel principal HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*14 segundos.*

Tabla 24 – Routing. Pruebas de basculación. Recuperación de túneles[88]

5. Caída de routers.	
5.1 Caída router HUB1.	
Resultado	*Basculación a túnel principal HUB1-backup.*
Protocolo que actúa	*HSRP / hardware.*
Tiempo	*14 segundos.*
5.2 Router HUB1 caído. Se cae router HUB1-backup.	
Resultado	*Basculación a túnel principal del HUB2.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*18 segundos.*

Tabla 25 – Routing. Pruebas de basculación. Caída de Routers[89]

[88] Fuente: elaboración propia

[89] Fuente: elaboración propia

6. Recuperación de routers.	
6.1 Router HUB1 caído. HUB1-backup activo. Recuperación router HUB1.	
Resultado	*Basculación de túnel principal HUB1-backup a túnel principal HUB1.*
Protocolo que actúa	*HSRP / hardware.*
Tiempo	*14 segundos.*
6.2 Router HUB1 caído y HUB1-caído. Recuperación HUB1-backup.	
Resultado	*Basculación de túnel principal HUB2 a túnel principal HUB1-backup.*
Protocolo que actúa	*EIGRP/software.*
Tiempo	*19 segundos.*

Tabla 26 – Routing. Pruebas de basculación. Recuperación de Routers[90]

5.3.6.2.2 COMUNICACIÓN DE HUB A SPOKE

1. Caída y recuperación de enlaces.	
1.1 Caída enlace principal del SPOKE.	
Resultado	*Basculación a túnel secundario del SPOKE.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*14 segundos.*
1.2 *Recuperación del enlace principal del SPOKE.*	
Resultado	*Basculación a túnel principal.*
Protocolo que actúa	*EIGRP/software.*
Tiempo	*9 segundos.*

Tabla 27 – Rounting. Pruebas de basculación. Caída y recuperación de enlaces[91]

2. Caída y recuperación de túneles.	
2.1 Caída del túnel principal del SPOKE.	
Resultado	*Basculación a túnel secundario del SPOKE.*
Protocolo que actúa	*EIGRP / software.*
Tiempo	*10 segundos.*
2.2 *Recuperación del túnel principal del SPOKE.*	
Resultado	*Basculación a túnel principal del SPOKE.*
Protocolo que actúa	*EIGRP/software.*
Tiempo	*12 segundos.*

Tabla 28 – Roting. Pruebas de basculación. Caída y recuperación de túneles[92]

[90] Fuente: elaboración propia
[91] Fuente: elaboración propia
[92] Fuente: elaboración propia

5.4 BLOQUE 2: CALIDAD DE SERVICIO

En esta sección se aplicarán y se verificarán las políticas de QoS para el reparto de latencias y caudales en función del tipo de tráfico:

Tipo de tráfico	Tipo de flujo	Puertos	Protocolos
Isócrono	VoIP y transmisiones de vídeo	[16000, 32000]	UDP
Interactivo	Flujos Oracle Flujos HTTP Flujos HTTPS	[1520, 1560] 80 443	TCP
Masivo	Correo, transferencias de archivos SMB, FTP…	Resto de puertos	TCP

Tabla 29 - QoS. Tipos de tráfico[93]

5.4.1 CONFIGURACIÓN

El sistema operativo Cisco IOS permite la creación de políticas de QoS basadas en clases mediante un Modular QoS CLI, cuya estructura permite crear políticas y anexarlas a las interfaces. En términos generales, los pasos a seguir para llevar a cabo un mecanismo de QoS con MCLI son:

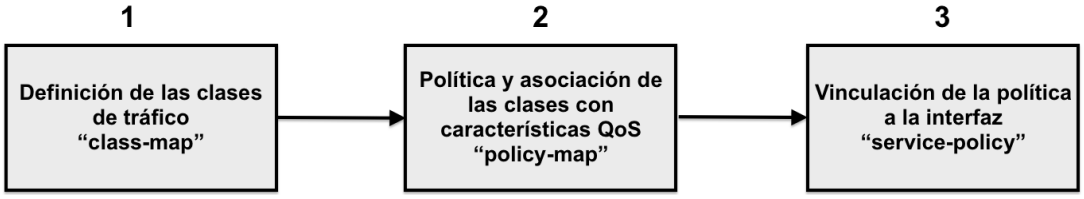


Figura 65 – QoS. Pasos de QoS con MCLI[94]

Para explicar más en detalle cada una de estas tres fases implicadas, a continuación se mostrará la configuración QoS del SPOKE1, siendo muy similar la configuración del resto de encaminadores (ver configuración de Routers en el Anexo).

[93] Fuente: elaboración propia
[94] Fuente: elaboración propia

5.4.1.1 DEFINICIÓN DE CLASES DE TRÁFICO

En primer lugar, definimos las clases mediante el comando `class-map`. Esta clase contiene tres elementos principales: un nombre, comandos `match` e instrucciones para la evaluación de comandos en caso de que existan más de un comando `match` (`match-all` y `match-policy`).

Mediante el comando `match` podemos especificar distintos criterios de clasificación de paquetes, que posteriormente serán comprobados para determinar si cumplen unos requisitos definidos, en cuyo caso se consideran pertenecientes a una clase en concreto y se aplicarán las especificaciones QoS establecidas en la política. Las siguientes clases agrupan un tipo de tráfico definido en la lista de acceso 101 y 102. El comando '`match-all`' indica que un paquete pertenecerá a la clase si cumple todas las cláusulas match.

```
class-map match-all RealTime
match access-group 101
class-map match-all Interactive
match access-group 102
```

Además, también se ha tenido en consideración el valor del campo DSCP de la cabecera IP, indicando de esta manera que el tráfico tiene un determinado valor en el campo TOS:

```
class-map match-all pInteractive
 match dscp af31
class-map match-all pRealTime
 match dscp ef
```

La lista de acceso 101 agrupa tráfico UDP con un rango de puertos que van desde 16000 hasta 32000 (Voz IP). En cambio, la lista de acceso 102 agrupa tráfico TCP con un rango de puertos 1520 y 1560 (flujos Oracle) y para los puertos 80 y 443.

```
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
```

```
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
```

Por otro lado, las siguientes listas clasifican el tráfico utilizando como criterio las direcciones MAC de los dos ISPs existentes de la simulación (MPLS e Internet). Observamos, que en este caso, se ha aplicado la sentencia `match-any`, indicando que se debe cumplir al menos uno de los comandos `match`:

```
class-map match-any ISP1.destination
 match destination-address mac CA08.117E.001C
class-map match-any ISP2.destination
 match destination-address mac CA03.1122.001C
class-map match-any ISP1.source
 match source-address mac CA08.117E.001C
class-map match-any ISP2.source
 match source-address mac CA03.1122.001C
```

Para conocer la dirección MAC de los dos ISP desde el router SPOKE1, usaremos el comando `show arp`:

```
SPOKE1#sh arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.4.1.1               -   ca04.3e35.001c  ARPA   FastEthernet1/0
Internet  81.134.91.1            0   ca03.1122.001d  ARPA   FastEthernet0/0
Internet  81.134.91.2            -   ca04.3e35.0000  ARPA   FastEthernet0/0
Internet  172.16.2.1             0   ca08.117e.001d  ARPA   FastEthernet1/1
Internet  172.16.2.2             -   ca04.3e35.001d  ARPA   FastEthernet1/1
SPOKE1#
```

Por último, las dos listas siguientes serán utilizadas para diferenciar la procedencia del tráfico: `qos1` hará referencia a todo aquél tráfico procedente de la MPLS y `qos2` se asociará con el tráfico procedente de Internet.

```
class-map match-any qos1
 match qos-group 1
class-map match-any qos2
 match qos-group 2
```

Por defecto, existe una clase denominada `class-default`, dónde se asociarán aquellos paquetes que no cumplan con ninguna clase definida.

5.4.1.2 DEFINICIÓN DE POLÍTICAS

La política define las acciones que se deberán adoptar con las clases de tráfico que se han definido previamente. Para crear una política de tráfico se utiliza el comando `policy-map`, cuyo objetivo es configurar características de QoS (shaper, policier…) así como asignación y reserva de caudales, marcado, priorización y eliminación de paquetes. Las políticas contienen, al menos, tres componentes: el nombre, una clase de tráfico y políticas QoS. Además, se permiten políticas anidadas, de manera que una política defina un comportamiento global que puede contener otras políticas con acciones más granulares.

En los HUBs y en los SPOKES se han definido 4 políticas distintas: `MARKING_DOWN` y `MARKING_UP` para el marcado de paquetes recibidos y enviados; así como `GLOBAL_DOWN` y `GLOBAL_UP` para definir un ancho de banda estricto de los paquetes entrantes y entregados respectivamente, utilizando los valores calculados en el punto *Cálculo de ancho de banda para HUBs y SPOKEs* del presente documento.

En la política `GLOBAL_DOWN` se utiliza la clase `qos1` y `qos2` y se define un ancho de banda estricto de 1016,93 Mbps. Después, dentro de estas clases, se ha creado otra denominada `child`, definida en la clase `MARKING_DOWN`, con reservas de caudal porcentuales.

En cuanto a la política `GLOBAL_UP` se especifica que cualquier paquete que pertenezca a `ISP1.destination` o `ISP2.destination`, tenga el mismo ancho de banda y la asociación de dicha política a la clase `pchild`.

```
policy-map GLOBAL_DOWN
 class qos1
  shape average 15840000
   service-policy child
 class qos2
  shape average 15840000
   service-policy child
```

```
policy-map GLOBAL_UP
 class ISP1.destination
  shape average 15840000
   service-policy pchild
 class ISP2.destination
  shape average 15840000
   service-policy pchild
```

La configuración de la política MARKING_DOWN se muestra a continuación. Observaremos que dentro del policy-map, se han empleado los comandos priority (tráfico isócrono o *realtime*) y bandwidth (tráfico interactivo y el resto de tráfico sin clasificar):

```
policy-map MARKING_DOWN
 class ISP1.source
  set qos-group 1
 class ISP2.source
  set qos-group 2
policy-map pchild
 class pRealTime
  priority percent 30
 class pInteractive
  bandwidth percent 20
 class class-default
  bandwidth percent 10
```

```
policy-map child
 class RealTime
  priority percent 30
  set dscp ef
 class Interactive
  bandwidth percent 20
  set dscp af31
 class class-default
  bandwidth percent 10
  set dscp default
```

El comando bandwidth permite efectuar reservas de ancho de banda a diferentes clases de tráfico. Si se produce la situación de que varias clases están reguladas mediante la aplicación de este comando y una de ellas no está generando tráfico de red, el resto de clases pueden utilizar el ancho de banda sobrante en la misma proporción que tienen configurada.

El comando priority ofrece una baja latencia a las clases que lo están utilizando, mediante el empleo de LLQ. No permite que la clase supere el caudal prefijado, aunque sí que permite que otro comando bandwidth haga uso del ancho de banda sobrante. Si una clase configurada con priority excede el ancho de banda configurado, los paquetes sobrantes serán suprimidos. Es decir, si la clase *Realtime* posee una reserva del 30 % el ancho de banda, la clase Interactive posee un 20 % y la class-default posee 10%, el ancho de banda sobrante se repartirá en la misma proporción entre las dos últimas clases, esto es:

$$Interactive \rightarrow \frac{20}{60} \cdot 40\%$$

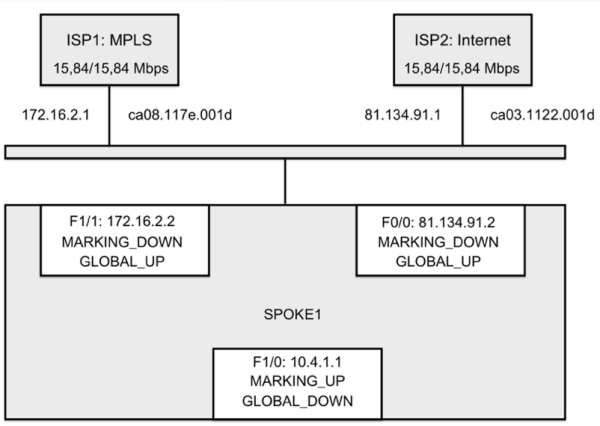
$$Class - Default \rightarrow \frac{10}{60} \cdot 40\%$$

Por último, en la política `MARKING_UP`, se ha configurado dos clases que, dependiendo del tipo de tráfico saliente (tráfico isócrono o interactivo), se asignará el atributo `dscp` correspondiente:

```
Policy-map MARKING_UP
 class RealTime
  set dscp ef
 class Interactive
  set dscp af31
```

5.4.1.3 APLICACIÓN DE POLÍTICAS A LA INTERFAZ

Por último, aplicaremos las políticas a las interfaces utilizando las cláusulas input/output según corresponda:




```
interface FastEthernet0/0
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP

interface FastEthernet1/0
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN

interface FastEthernet1/1
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
```

Figura 66 – QoS. Aplicación de políticas a las interfaces[95]

[95] Fuente: elaboración propia

5.4.2 VALIDACIÓN

Seguidamente, realizaremos simulaciones de generación de flujos de datos que permitan la comprobación del correcto funcionamiento de las políticas QoS configuradas. En este aspecto, existe un gran número de tecnologías (bing, ttcp…), de las cuales se utilizará `iperf`, ya que ofrece un amplio rango de funcionalidades y se encuentra instalada por defecto en los terminales `ipterm` (máquinas virtuales Linux de GNS3). Además, esta herramienta se utilizará en conjunción con `ntop`, que nos permitirá visualizar gráficamente los distintos flujos de red.

En primer lugar, debemos de establecer un servidor `iperf` (parámetro `-s`) y el puerto de escucha (parámetro `-p`), por ejemplo, en la máquina 10.4.1.2 (SPOKE1). Como se han definido tres tipos de tráfico distintos, ejecutaremos en el servidor tres sentencias `iperf` para que la máquina sea capaz de responder a tres tipos de flujos diferentes:

Servidor → Máquina 10.4.1.2

```
Iperf3 -s -p 16000 # VozIP (tráfico isócrono)
Iperf3 -s -p 443 # Tráfico HTTPS (tráfico interactivo)
Iperf3 -s -p 1521 # Flujos Oracle (tráfico interactivo)
Iperf3 -s -p 21 # Flujos FTP (tráfico masivo)
```

A continuación, creamos clientes `iperf` en otros terminales de la simulación para que se conecten al servidor y envíen tráfico al puerto especificado:

Clientes

```
iperf3 -c 10.4.1.2 -u -p 16000 # VozIP (UDP) → Máquina 10.4.2.2 (SPOKE2)
iperf3 -c 10.4.1.2 -p 443 # Tráfico HTTPS (TCP) → Máquina 10.3.1.2 (HUB1)
iperf3 -c 10.4.1.2 -p 1521 # Flujos Oracle (TCP) → Máquina 10.4.3.3 (SPOKE3)
iperf3 -c 10.4.1.2 -p 21 # Flujos FTP (TCP) → Máquina 10.4.3.2 (SPOKE3)
```

Ejecutamos los comandos anteriores en cada máquina correspondiente. Las siguientes capturas de pantalla muestran el envío y recepción de los flujos de datos.

TRÁFICO ISÓCRONO

Envío de tráfico isócrono desde 10.4.2.2 a 10.4.1.2, utilizando el puerto 16000 (UDP):

Cliente	Servidor

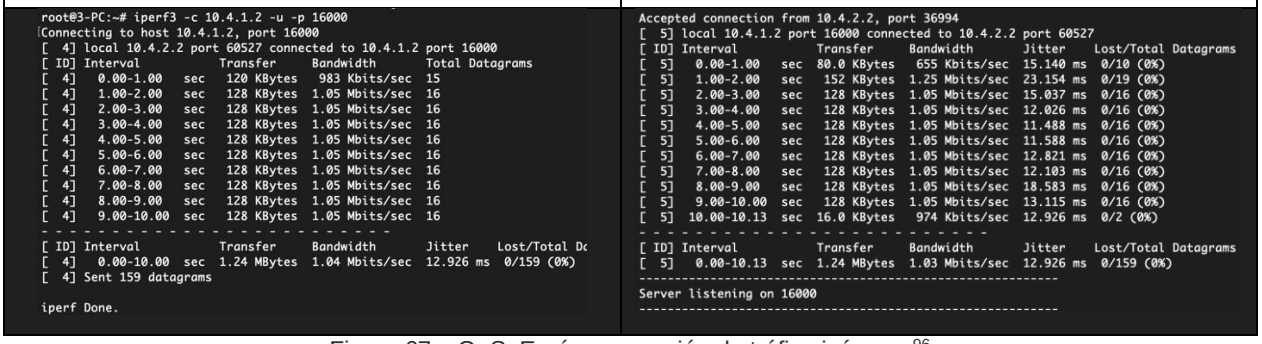


Figura 67 – QoS. Envío y recepción de tráfico isócrono[96]

TRÁFICO INTERACTIVO:

Envío de tráfico isócrono desde 10.3.1.2 y 10.4.3.3, a 10.4.1.2, utilizando los puertos 443 y 1521 (TCP), respectivamente:

Cliente	Servidor

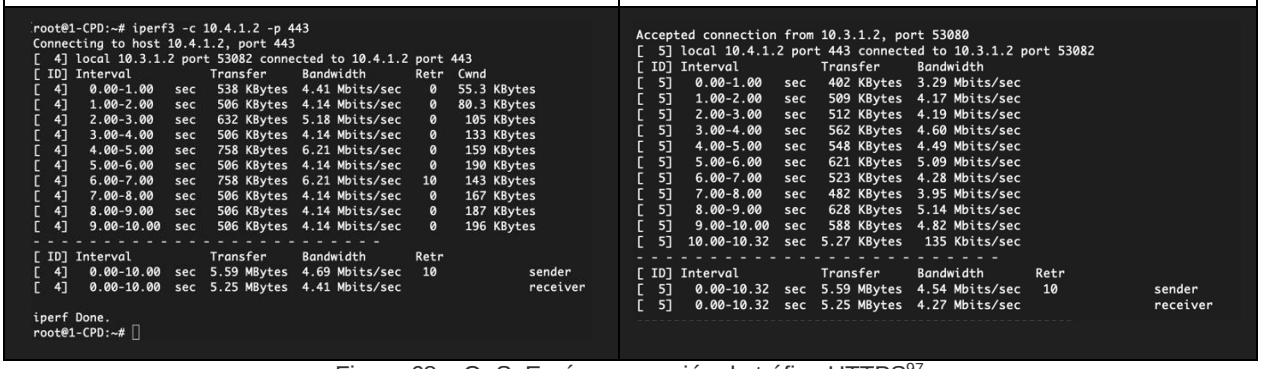


Figura 68 – QoS. Envío y recepción de tráfico HTTPS[97]

Cliente	Servidor

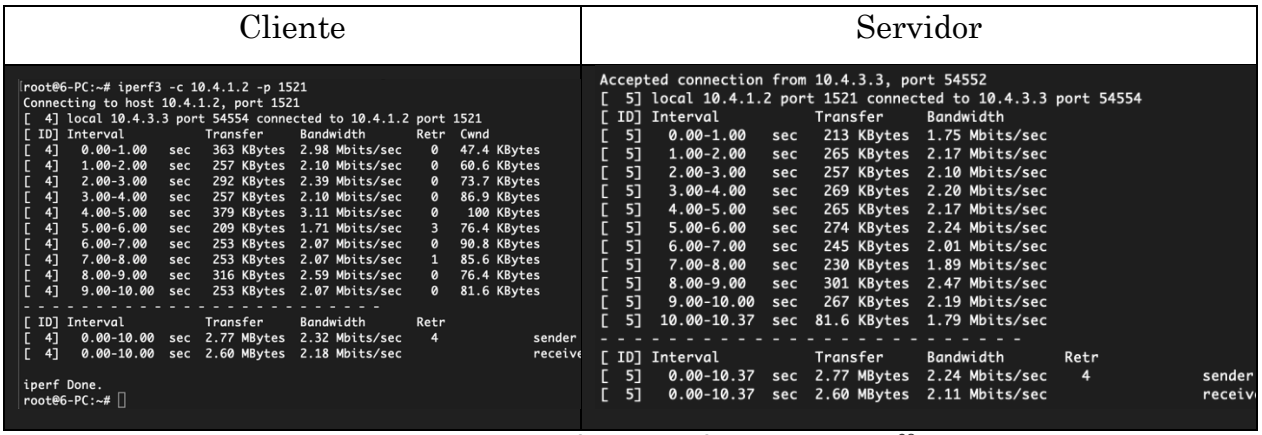


Figura 69 – QoS. Envío y recepción de flujos Oracle[98]

[96] Fuente: elaboración propia

[97] Fuente: elaboración propia

[98] Fuente: elaboración propia

TRÁFICO MASIVO:

Envío de tráfico masivo desde 10.4.2.2 a 10.4.1.2, utilizando los puertos 443 y 1521 (TCP).

Cliente	Servidor

```
Connecting to host 10.4.1.2, port 21
[  4] local 10.4.3.2 port 58408 connected to 10.4.1.2 port 21
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec   120 KBytes   983 Kbits/sec  15
[  4]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  16
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
[  4]   0.00-10.00  sec  1.24 MBytes  1.04 Mbits/sec  37.468 ms  0/159 (0%)
[  4] Sent 159 datagrams
```

```
Accepted connection from 10.4.3.2, port 42566
[  5] local 10.4.1.2 port 21 connected to 10.4.3.2 port 58408
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   104 KBytes   852 Kbits/sec  20.159 ms  0/13 (0%)
[  5]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  44.659 ms  0/16 (0%)
[  5]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  42.586 ms  0/16 (0%)
[  5]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  44.334 ms  0/16 (0%)
[  5]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  38.908 ms  0/16 (0%)
[  5]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  38.533 ms  0/16 (0%)
[  5]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  37.161 ms  0/16 (0%)
[  5]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  37.246 ms  0/16 (0%)
[  5]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  36.404 ms  0/16 (0%)
[  5]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  38.793 ms  0/16 (0%)
[  5]  10.00-10.16  sec  16.0 KBytes   819 Kbits/sec  37.468 ms  0/2 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-10.16  sec  1.24 MBytes  1.03 Mbits/sec  37.468 ms  0/159 (0%)
```

Figura 70 – QoS. Envío y recepción de tráfico FTP[99]

Una manera de visualizar y controlar los máquinas o aplicaciones que están consumiendo recursos de red en un instante concreto de manera gráfica es utilizando herramientas de monitorización como NetFlow o ntop. A continuación, mostraremos algunas imágenes capturadas durante la simulación.

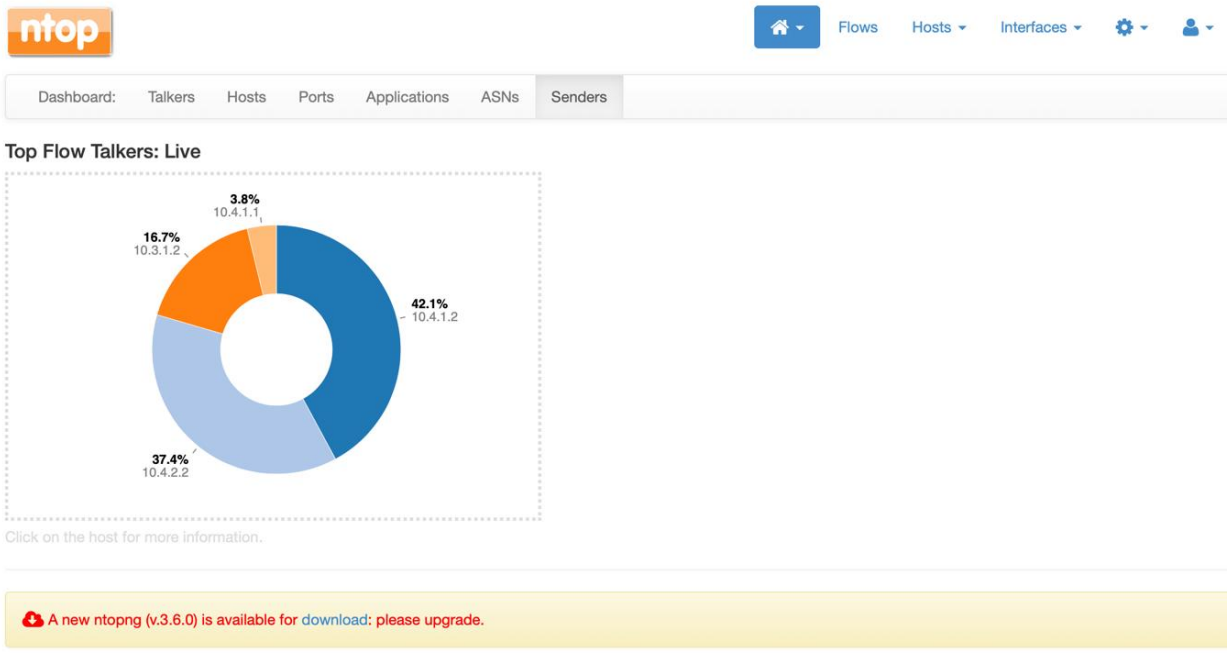


Figura 71 – QoS. ntop. Hosts que están envíando tráfico[100]

[99] Fuente: elaboración propia

[100] Fuente: elaboración propia a partir de ntop

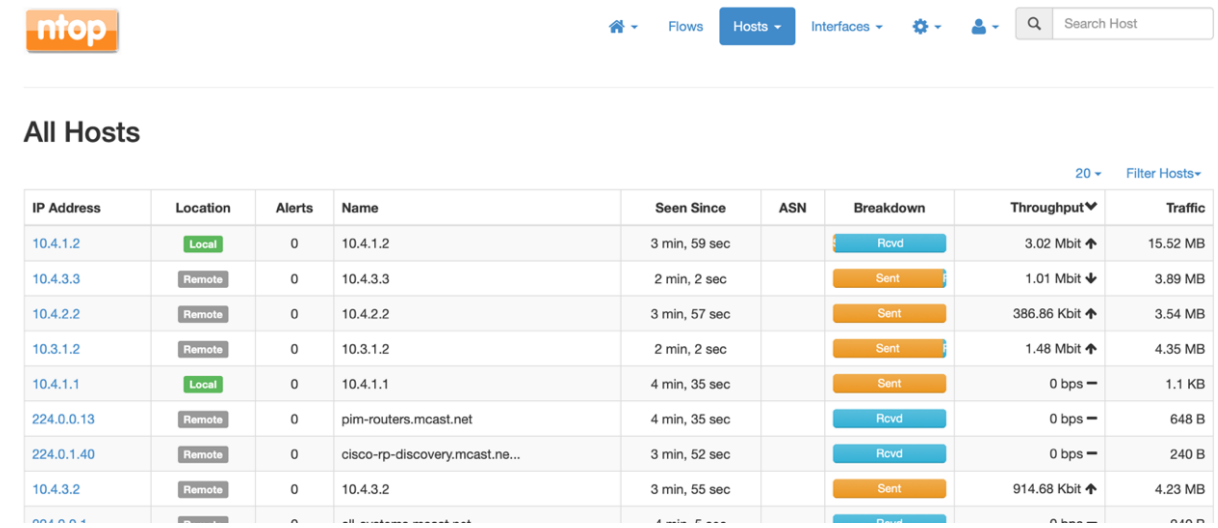


Figura 72 – QoS. ntop. Flujos enviados por los terminales[101]

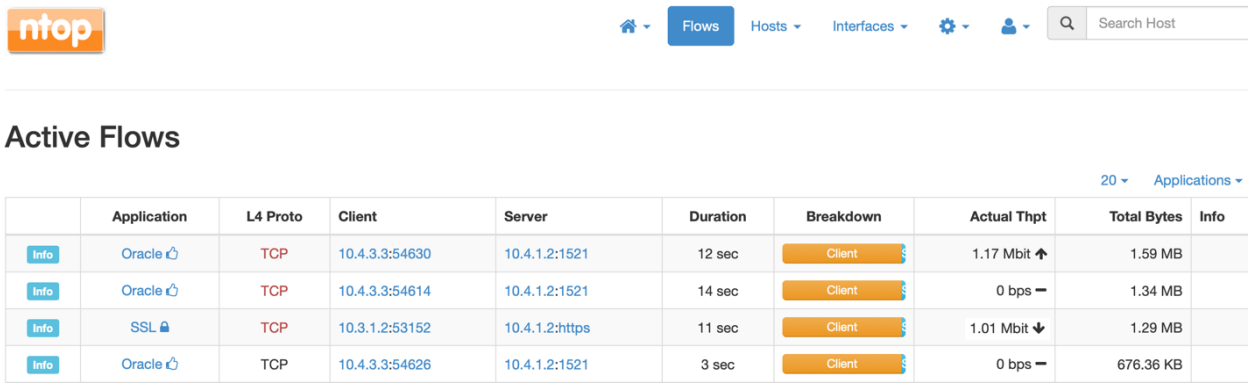


Figura 73 – QoS. ntop. Flujos activos en la red[102]

Después, si utilizamos la sentencia Cisco iOS `sh policy-map interface [nre_interfaz]`, podremos verificar que cada tipo de tráfico se ha introducido en las clases configuradas. Por ejemplo, si sabemos que desde la oficina tipo C (SPOKE3) se ha generado tráfico isócrono e interactivo utilizando la WAN principal (MPLS), las políticas que habrán actuado en el router de dicha oficina serán aquéllas aplicadas a la interfaz LAN `f1/0` y a la interfaz de salida a la MPLS, esto es, `f1/1`. En la siguiente imagen, observamos que la política `MARKING_UP` de la interfaz `f1/0` ha marcado el tráfico (*matches*):

[101] Fuente: elaboración propia a partir de ntop
[102] Fuente: elaboración propia a partir de ntop

```
SPOKE3#sh policy-map int f1/0
 FastEthernet1/0

  Service-policy input: MARKING_UP

    Class-map: RealTime (match-all)
      795 packets, 1095510 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 101
      QoS Set
        dscp ef
          Packets marked 795

    Class-map: Interactive (match-all)
      2122 packets, 2969867 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 102
      QoS Set
        dscp af31
          Packets marked 2122

    Class-map: class-default (match-any)
      177 packets, 242221 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Después, la política `GLOBAL_UP` actúa en la interfaz de salida a la WAN principal, con el tráfico clasificado:

```
Service-policy output: GLOBAL_UP

  Class-map: ISP1.destination (match-any)
    9012 packets, 5539768 bytes
    5 minute offered rate 2000 bps, drop rate 0000 bps
    Match: destination-address mac CA08.117E.0054
      9012 packets, 5539768 bytes
      5 minute rate 2000 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/4/0
    (pkts output/bytes output) 9008/5533792
    shape (average) cir 2530000, bc 10120, be 10120
    target shape rate 2530000

    Service-policy : pchild

      queue stats for all priority classes:
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 1431/1219530

      Class-map: pRealTime (match-all)
        1431 packets, 1219530 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match:  dscp ef (46)
        Priority: 30% (759 kbps), burst bytes 18950, b/w exceed drops: 0

      Class-map: pInteractive (match-all)
        2122 packets, 3139420 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match:  dscp af31 (26)
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/4/0
        (pkts output/bytes output) 2118/3133444
        bandwidth 20% (506 kbps)

      Class-map: class-default (match-any)
        5459 packets, 1180818 bytes
        5 minute offered rate 2000 bps, drop rate 0000 bps
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 5459/1180818
        bandwidth 10% (253 kbps)
```

En el SPOKE1, que es donde se encuentra el servidor iperf configurado previamente, habrán actuado las políticas `MARKING_DOWN` en la interfaz f1/1:

```
SPOKE1#sh policy-map int f1/1
 FastEthernet1/1

  Service-policy input: MARKING_DOWN

    Class-map: ISP1.source (match-any)
      15431 packets, 13552182 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: source-address mac CA08.117E.001D
        15431 packets, 13552182 bytes
        5 minute rate 2000 bps
      QoS Set
        qos-group 1
          Packets marked 15431
```

En la interfaz LAN del mismo SPOKE, la política `GLOBAL_DOWN` habrá intervenido y clasificado el tráfico para su recepción:

```
Service-policy output: GLOBAL_DOWN

  Class-map: qos1 (match-any)
    9967 packets, 11760465 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps

Match: qos-group 1
  9967 packets, 11760465 bytes
  5 minute rate 0 bps
Queueing
queue limit 64 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 9967/11760465
shape (average) cir 15840000, bc 63360, be 63360
target shape rate 15840000

Service-policy : child

  queue stats for all priority classes:
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 3340/2703364

  Class-map: RealTime (match-all)
    3340 packets, 2703364 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: access-group 101
    Priority: 30% (4752 kbps), burst bytes 118800, b/w exceed drops: 0

    QoS Set
      dscp ef
        Packets marked 3340

  Class-map: Interactive (match-all)
    6433 packets, 9030951 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: access-group 102
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 6433/9030951
    bandwidth 20% (3168 kbps)
    QoS Set
      dscp af31
        Packets marked 6433

  Class-map: class-default (match-any)
    194 packets, 26150 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 194/26150
    bandwidth 10% (1584 kbps)
    QoS Set
      dscp default
        Packets marked 194
```

Una anotación importante son los *matches* producidos en la class-default debido a otro tipo de mensajes generados en la red durante la realización de la validación: tráfico FTP, solicitudes ping, mensajes EIGRP, mensajes PIM (explicados en la siguiente sección)…

5.5 BLOQUE 3: MULTICAST IP

En este apartado generaremos una fuente de difusión de datos en 1-CPD simulando un *stream* de vídeo, que se propagará al resto de la red. Este flujo deberá ser recibido por las máquinas 1-PC (SPOKE1), 3-PC (SPOKE2) y 5-PC (SPOKE3). Particularmente, se utilizará el protocolo de encaminamiento *Multicast PIM* en su modalidad *Sparse Mode* (PIM-SM), lo que requerirá un punto de encuentro en el HUB1:

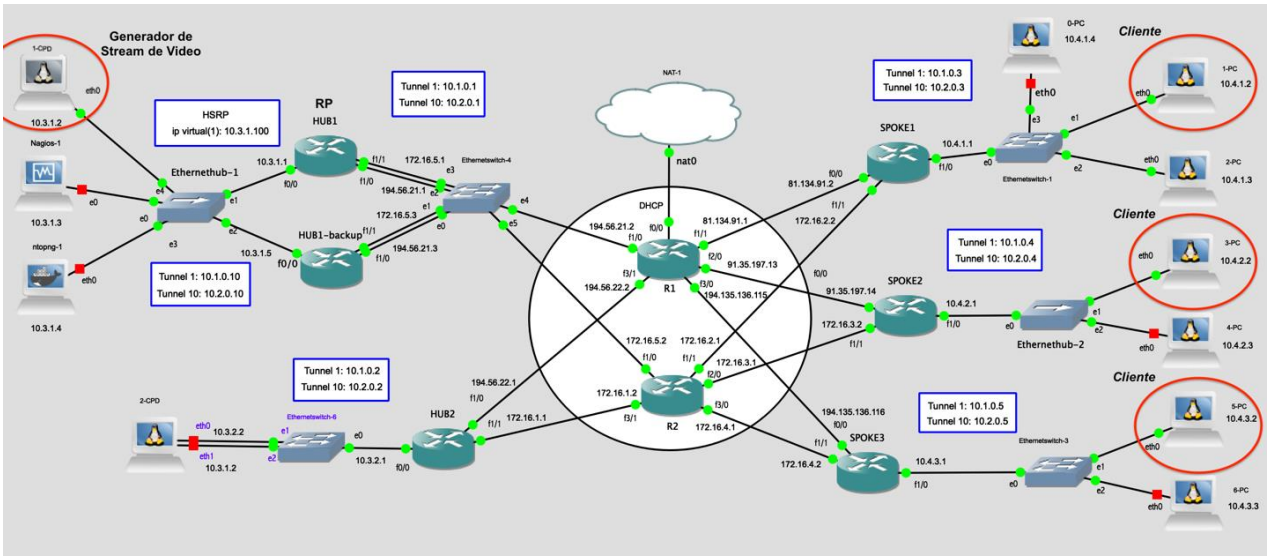


Figura 74 – Multicast. Servidor y Clientes[103]

Detallaremos, por tanto, la configuración necesaria, tanto en los PCs como en los routers, poder transmitir Multicast IP. Tras ello, validaremos la transmisión, utilizando un programa auxiliar escrito en lenguaje de programación C denominado "*multicast.c*" y comprobaremos el mecanismo utilizado por IGMP para la anexión de grupos Multicast mediante el analizador de red `tcpdump`. Para acabar, verificaremos el diseño mediante comandos Cisco IOS.

[103] Fuente: elaboración propia

5.5.1 CONFIGURACIÓN

Los comandos adecuados para configurar PIM-SM en los routers implicados son:

HUB1 (RP)	SPOKEs
(1) ip multicast-routing	ip multicast-routing
(2) ip pim rp-address 10.2.0.1	ip pim rp-address 10.2.0.1
# Tunnel 10 (túnel principal)	# Tunnel 10 (túnel principal)
(3) ip pim sparse-mode	ip pim sparse-mode
(4) ip pim nbma-mode	# Interfaz LAN interna
# Interfaz LAN interna	ip pim sparse-mode
ip pim sparse-mode	

Tabla 30 - Configuración PIM-SM[104]

La línea (1) indica a un router que ha de encaminar tráfico Multicast y tiene que ser configurado en todos los encaminadores a excepción de aquél que simula Internet (R1). Después, indicaremos la ubicación del RP (2). Esta asignación implica que para cada grupo de Multicast, se creará un árbol de distribución con origine en el Router elegido como RP, HUB1 en este caso. En los SPOKES también introduciremos dicho comando. Después, añadiremos la línea (3) a cada interfaz involucrada directamente en la transmisión de tráfico Multicast, es decir, a la interfaz LAN interna y a las interfaces Túnel. Finalmente, introduciremos el comando (4) en el HUB1 para indicar a dicho router que deberá procesar la interfaz especificada como concentrador multipunto para Multicast (en redes NBMA).

La configuración Linux se resume en la inclusión de una ruta para indicar al Kernel del SO que la interfaz (eth0) encamine tráfico Multicast:

```
route add -net 224.0.0.0/4 dev eth0
```

5.5.2 VALIDACIÓN

En primer lugar, usaremos el comando `netstat -rn` para comprobar la consistencia de la tabla de rutas en cada máquina virtual:

[104] Fuente: elaboración propia

```
root@1-CPD:~# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.3.1.100      0.0.0.0         UG        0 0          0 eth0
10.3.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 eth0
root@1-CPD:~#
```

En este caso, ejecutaremos en cada cliente (1-PC, 3-PC, 5-PC), el receptor de Multicast para el grupo 225.1.1.1 y al puerto UDP 1234 de manera que el router local informe a su vecino utilizando IGMP de la presencia de oyentes en el grupo Multicast especificado:

```
root@1-PC:~# ./multicast
usage: multicast IP_ADDRESS UDP_PORT <server|client> [TTL]
root@1-PC:~# ./multicast 225.1.1.1 1234 client
```

Después, enviaremos un flujo *Multicast* al grupo 225.1.1.1 utilizando el puerto UDP 1234, simulando de esta manera un *stream* de vídeo.

```
root@1-CPD:~# ./multicast 225.1.1.1 1234 server 4
sending message: time is Fri Aug 23 17:49:49 2019
sending message: time is Fri Aug 23 17:49:54 2019
sending message: time is Fri Aug 23 17:49:59 2019
sending message: time is Fri Aug 23 17:50:04 2019
```

En la siguiente imagen podemos comprobar que los tres clientes (1-PC, 3-PC y 5-PC) están recibiendo el stream *Multicast* en el mismo instante de tiempo:

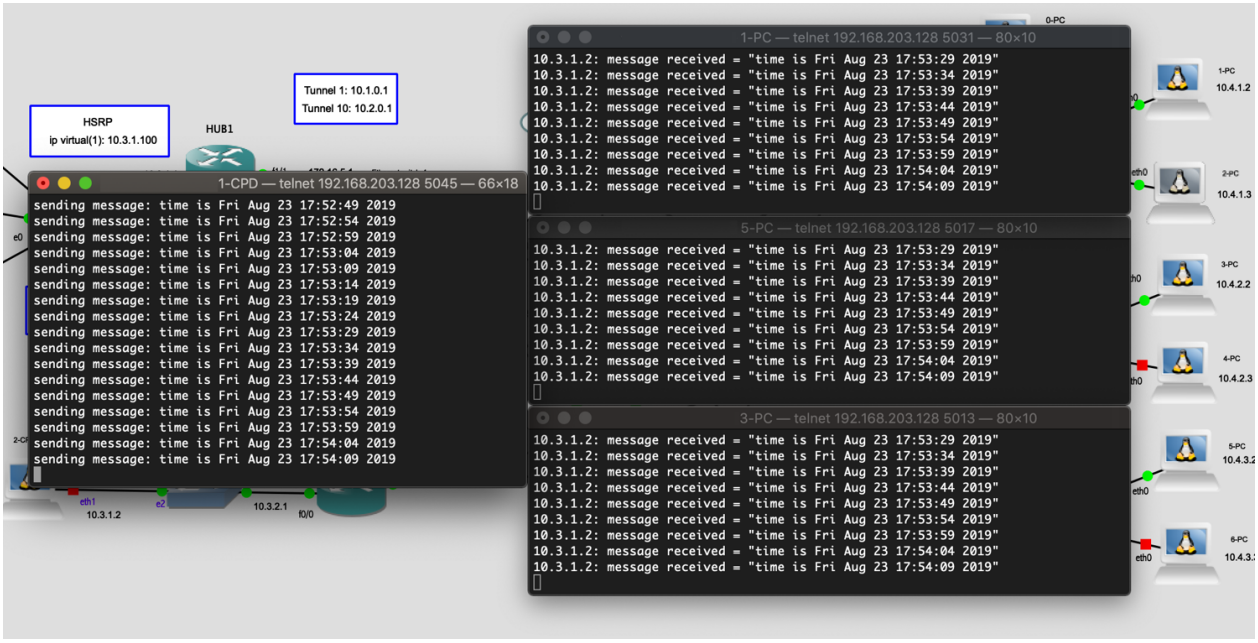
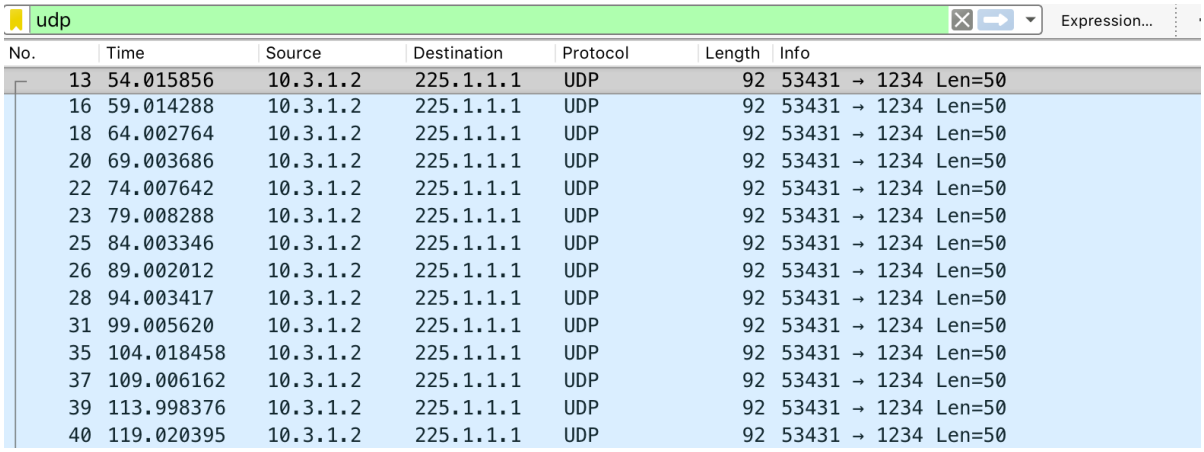


Figura 75 – Multicast. Envío y recepción del stream[105]

[105] Fuente: elaboración propia

Seguidamente, usaremos Wireshark en la LAN interna del SPOKE1 para comprobar que el flujo Multicast del stream de vídeo (1-CPD, en este caso) dirigido al grupo 225.1.1.1 está llegando a dicha red:



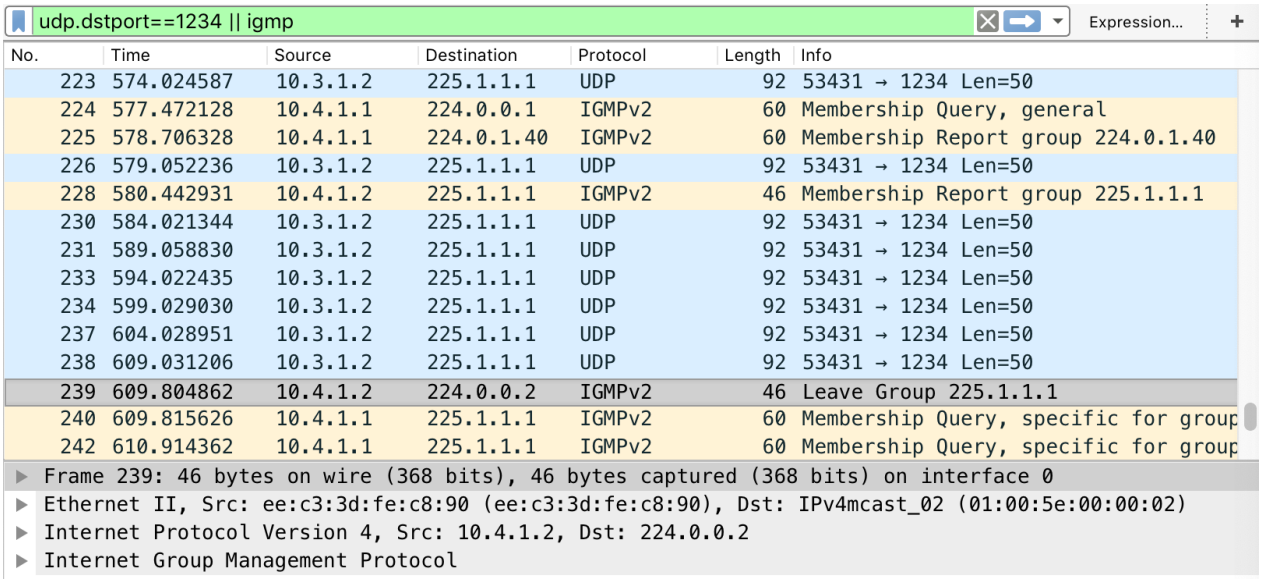
udp

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
13	54.015856	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
16	59.014288	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
18	64.002764	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
20	69.003686	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
22	74.007642	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
23	79.008288	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
25	84.003346	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
26	89.002012	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
28	94.003417	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
31	99.005620	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
35	104.018458	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
37	109.006162	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
39	113.998376	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
40	119.020395	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50

Figura 76 – Multicast. Wireshark. Tráfico UDP[106]

Ahora abortaremos el programa de recepción de tráfico Multicast en el terminal 1-PC, mediante la combinación de teclas `Ctrl + C`. Tras realizar esta acción, observaremos que dicho equipo envía un mensaje "*IGMP Leave*", deteniéndose la recepción de flujo:



udp.dstport==1234 || igmp

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
223	574.024587	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
224	577.472128	10.4.1.1	224.0.0.1	IGMPv2	60	Membership Query, general
225	578.706328	10.4.1.1	224.0.1.40	IGMPv2	60	Membership Report group 224.0.1.40
226	579.052236	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
228	580.442931	10.4.1.2	225.1.1.1	IGMPv2	46	Membership Report group 225.1.1.1
230	584.021344	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
231	589.058830	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
233	594.022435	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
234	599.029030	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
237	604.028951	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
238	609.031206	10.3.1.2	225.1.1.1	UDP	92	53431 → 1234 Len=50
239	609.804862	10.4.1.2	224.0.0.2	IGMPv2	46	Leave Group 225.1.1.1
240	609.815626	10.4.1.1	225.1.1.1	IGMPv2	60	Membership Query, specific for group
242	610.914362	10.4.1.1	225.1.1.1	IGMPv2	60	Membership Query, specific for group

▶ Frame 239: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▶ Ethernet II, Src: ee:c3:3d:fe:c8:90 (ee:c3:3d:fe:c8:90), Dst: IPv4mcast_02 (01:00:5e:00:00:02)
▶ Internet Protocol Version 4, Src: 10.4.1.2, Dst: 224.0.0.2
▶ Internet Group Management Protocol

Figura 77 – Multicast. Wireshark. Mensaje IGMP Leave[107]

[106] Fuente: elaboración propia
[107] Fuente: elaboración propia

Finalmente, volvemos a iniciar el receptor de Multicast, ejecutando de nuevo `./multicast 125.1.1.1 1234 client` en 1-PC y apreciamos los resultados en Wireshark:

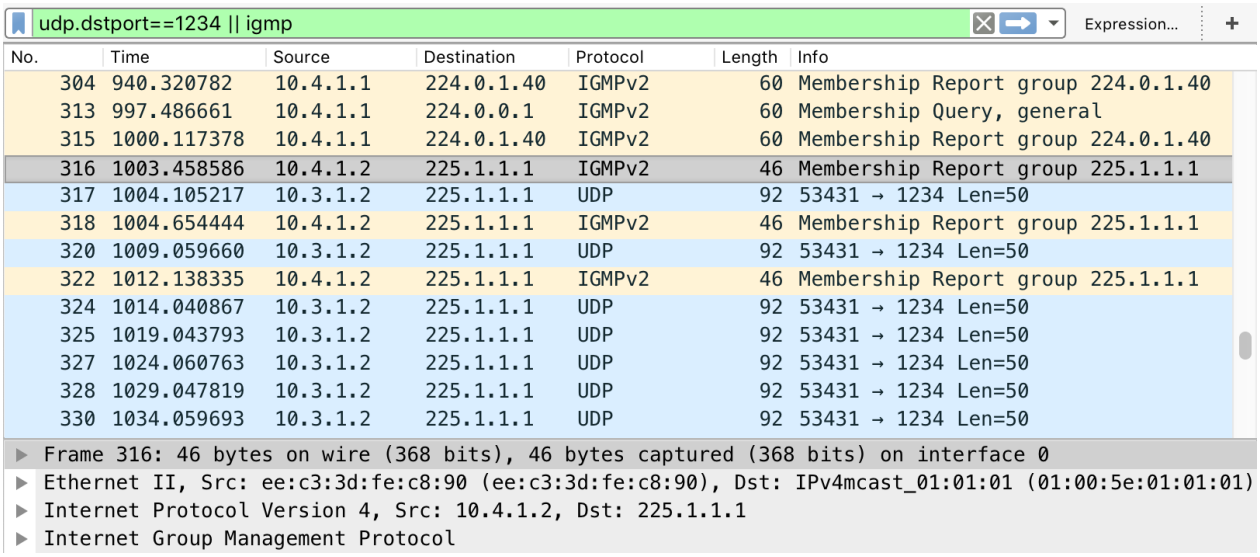


Figura 78 – Multicast. Wireshark. Mensaje Membership Report [108]

Observamos, que existen mensajes de anexión a otro grupo diferente Multicast de 225.1.1.1 (*Membership Report group 225.1.1.1*), un mensaje de anexión al grupo 225.1.1.1 (*Membership Report group 255.1.1.1*) después de haber ejecutado el programa de recepción de tráfico Multicast y el inicio de las nuevas transmisiones Multicast.

Otra forma de comprobar y verificar el funcionamiento de la simulación es utilizando comandos que proporcionen información sobre las tablas Multicast y grupos, tabla de routers y vecinos PIM; así como los RP configurados. Estos datos se encuentran en el Anexo.

[108] Fuente: elaboración propia

5.6 BLOQUE 4: IPSEC

En este punto se detallará la configuración y validación del protocolo IPSec en la simulación para que la transmisión de datos salientes de las dos interfaces de salida a las WAN sean seguras. Particularmente, se utilizará la versión 2 de IKE, desarrollado por IETF, a fin de mejorar la función de efectuar la autenticación de *peer* o socio y el intercambio de claves dinámico. Además, simplifica los flujos del intercambio de claves e introduce medidas para reparar vulnerabilidades y ambigüedades de la primera versión del protocolo.

5.6.1 CONFIGURACIÓN

En IKEv1, un *transform set* representa un conjunto de parámetros de seguridad, protocolos y algoritmos. Durante la negociación del SA, los nodos de comunicación se ponen de acuerdo para utilizar un particular *transform set* para proteger el flujo de datos. En cambio, en IKEv2, se utiliza una propuesta o *proposal*, que se caracteriza por ser un conjunto de *transforms* utilizadas en la negociación de la SA de IKEv2 como parte del intercambio `IKE_SA_INIT`, que visualizaremos y explicaremos más en detalle utilizando pantallazos de Wireshark. Un *proposal* se considera que es completa cuando posee un algoritmo de cifrado, un algoritmo de integridad y un grupo Diffie-Hellman. En este caso, el nombre del *proposal* asignado es `prop-1`, especifica el algoritmo de encriptación `AES-CBC 256 – 256-bit AES CBC`, el algoritmo hash `SHA-2 Family 256-bit` y el grupo 15, es decir, `3072-bit DH`:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-256
 integrity sha256
 group 15
```

Para utilizar un IKEv2 *proposal* en las negociaciones IPSec, deben estar presentes en la definición de las políticas IKEv2:

```
crypto ikev2 policy site-policy
 proposal prop-1
```

Si no se hubiese especificado la propuesta `prop-1` a una política, se utilizaría una propuesta establecida por defecto en el router (*default*), que en nuestro caso hubiese podido ser cualquiera de las siguientes combinaciones:

```
HUB1#sh crypto ikev2 proposal
 IKEv2 proposal: default
      Encryption : AES-CBC-256 AES-CBC-192 AES-CBC-128
      Integrity  : SHA512 SHA384 SHA256 SHA96 MD596
      PRF        : SHA512 SHA384 SHA256 SHA1 MD5
      DH Group   : DH_GROUP_1536_MODP/Group 5 DH_GROUP_1024_MODP/Group 2
 IKEv2 proposal: prop-1
      Encryption : AES-CBC-256
      Integrity  : SHA256
      PRF        : SHA256
      DH Group   : DH_GROUP_3072_MODP/Group 15
HUB1#
```

Después definimos la *pre-shared key* para el HUB y el SPOKEs del siguiente modo:

```
crypto ikev2 keyring cisco-ikev2-keyring
 peer dmvpn-node
  description symmetric pre-shared key for the hub/spoke
  address 0.0.0.0 0.0.0.0
  pre-shared-key TFG_2019_DMVPN
```

Utilizando la *pre-shared key* definida anteriormente, creamos un perfil para la conexión IPSec:

```
crypto ikev2 profile cisco-ikev2-profile
 match fvrf any
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local cisco-ikev2-keyring
```

Seguidamente, se selecciona un algoritmo de encriptación ESP con 128-bit AES (esp-aes) para el *ESP Encryption Transform* y ESP con SHA (variante de HMAC) para el *ESP Authentication Transform* y se establece el modo transporte:

```
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
 mode transport
```

Utilizamos los dos perfiles de transform set creados previamente en la siguiente sentencia:

```
crypto ipsec profile cisco-ipsec-ikev2
 set transform-set cisco-ts
 set ikev2-profile cisco-ikev2-profile
```

Y finalmente, asignamos el `crypto ipsec` creado a las interfaces túnel (aplicable a todos los routers de la DMVPN):

```
interface Tunnel10
  tunnel protection ipsec profile cisco-ipsec-ikev2
```

5.6.2 VALIDACIÓN

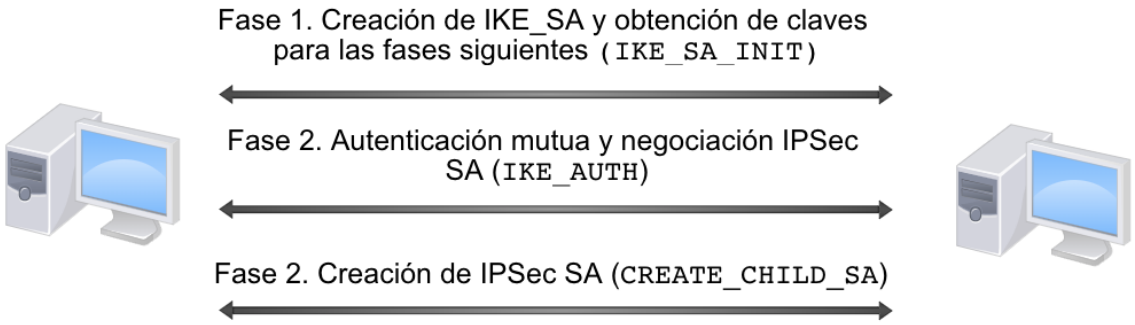


Figura 79 – IPSec. Intercambio de mensajes en IKEv2[109]

Después de haber aplicado la protección IPSec a los túneles (principales y secundarios), verificaremos su funcionamiento utilizando Wireshark en las interfaces de salida de las WAN. IKEv2 (al igual que IKEv1) opera en dos fases. La primera fase de IKEv2, comparable a la primera fase de IKE (intercambio no seguro), es `IKE_SA`, cuyos atributos son definidos en la política de intercambio de claves, consta del par de mensajes `IKE_SA_INIT`.

Estos mensajes constituyen el intercambio inicial en el cual los pares establecen un canal seguro y por tanto, se negocia los parámetros de seguridad de `IKE SA`, los números aleatorios temporales o *nonces* y los valores de Diffie-Hellman:

[109] Fuente: elaboración propia

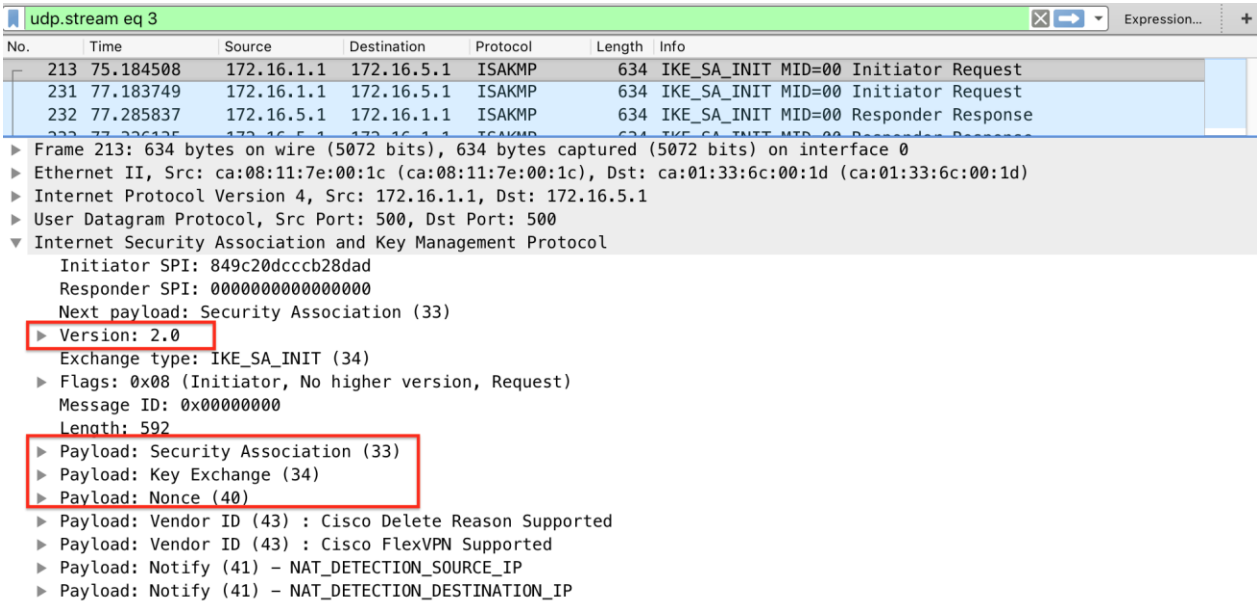
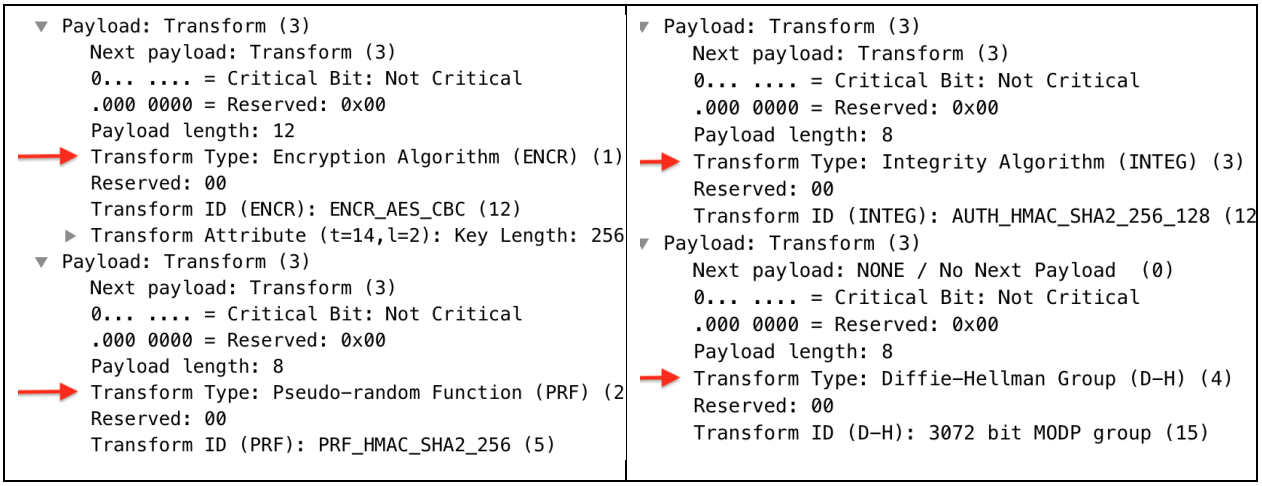


Figura 80 – IPSec. Wireshark. Mensajes IKE_SA_INIT IKEv2[110]

Si hacemos extendemos la sección de `Internet Security Association and Management Control > Payload Security Association > Payload: Proposal (2) #1`, podremos observar la carga útil que se emplea, esencialmente, para negociar propuestas de IKE.



Key Exchange y las cargas útiles (payload) *nonce* se usan para intercambiar materiales de clave:

[110] Fuente: elaboración propia

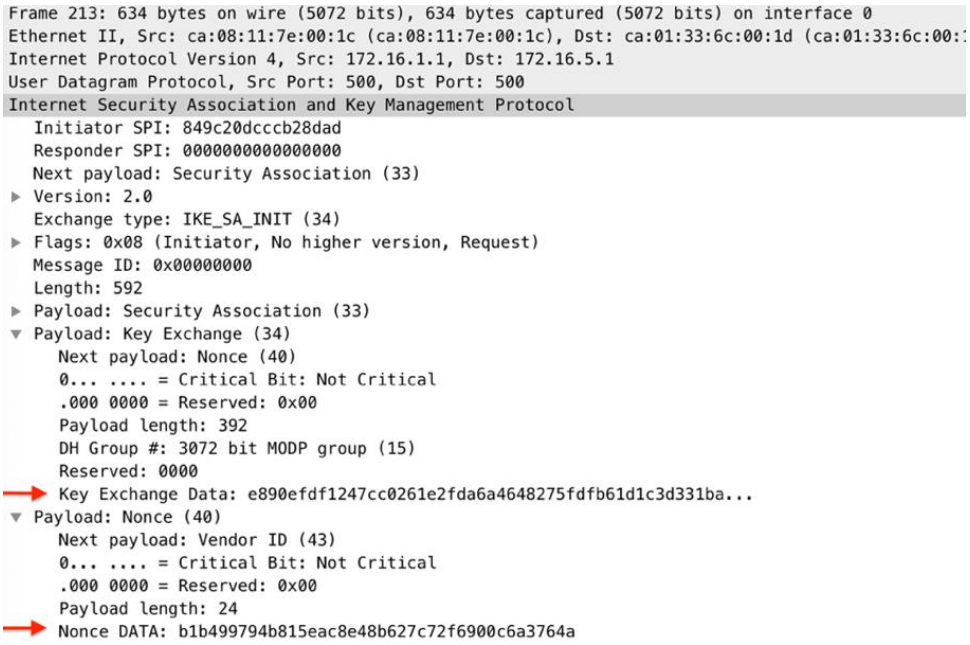


Figura 81 – IPSec. Wireshark. Mensaje IKE_SA_INIT (key exchange & nonce)[111]

La segunda fase de IKEv2, comparable a la Fase 2 de IKEv1 (intercambio seguro) es `CHILD_SA`, cuyos atributos se definen en la política de datos. El primer `CHILD_SA` constituye el par de mensajes `IKE_AUTH`, que se encarga de la autenticación mutua y la negociación de los algoritmos usados en IPSec SA:

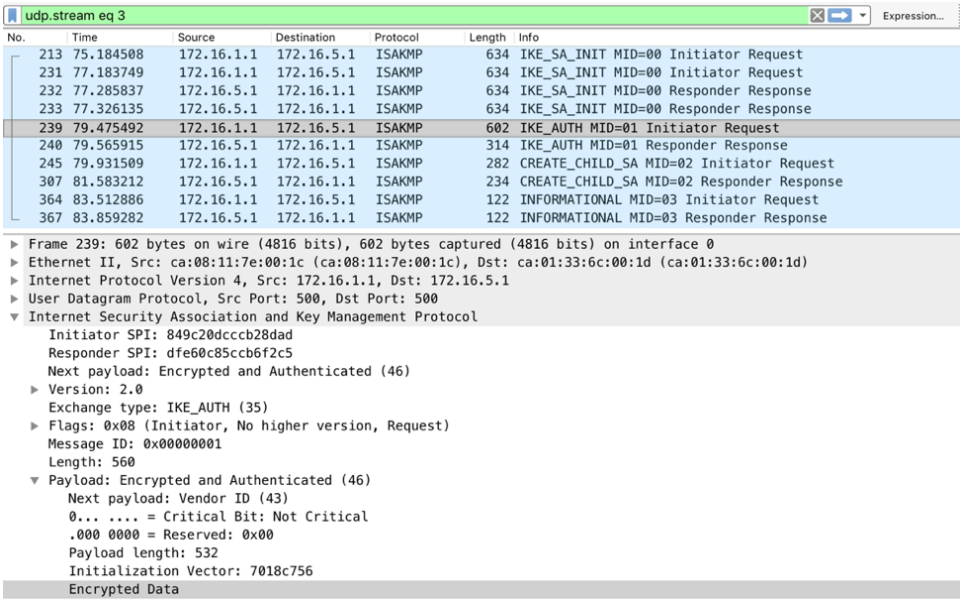


Figura 82 – IPSec. Wireshark. Mensaje IKE_AUTH Ikev2[112]

[111] Fuente: elaboración propia
[112] Fuente: elaboración propia

De la captura de pantalla anterior, notamos la presencia de dos mensajes `CREATE_CHILD_SA Exchange` y `INFORMATIONAL Exchange`. El primero de ellos se encarga de establecer un IPSec SA y permite la derivación de las claves para los protocolos ESP y/o AH. El segundo es un mensaje meramente informativo.

En la siguiente captura de pantalla podemos observar que los datos transmitidos en la red se ha incluido la cabecera ESP:

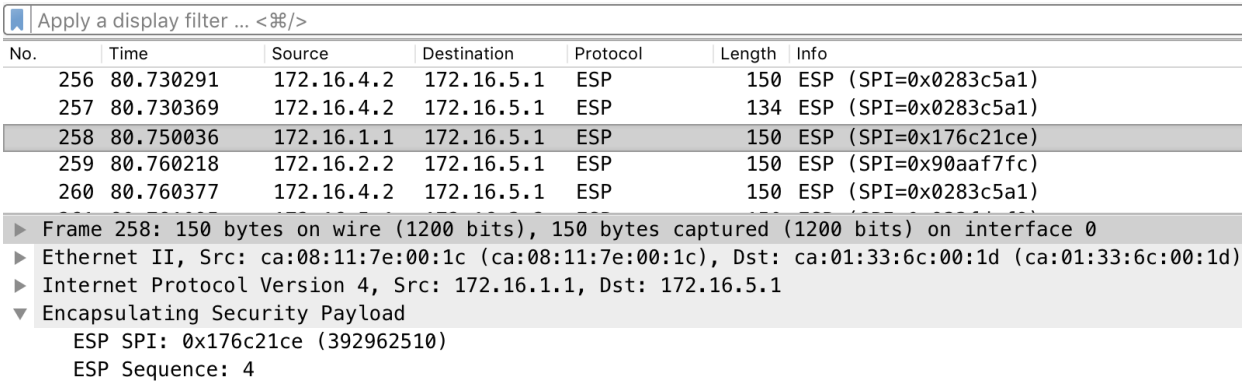


Figura 83 – IPSec. Wireshark. Cabecera ESP[113]

Para finalizar, si empleamos el comando `sh dmvpn details`, podemos observar los detalles de las sesiones crypto establecidas y el estado en el que se encuentran, entre muchos otros elementos. Ver apartado "Comandos Cisco para comprobar la conectividad" del Anexo.

[113] Fuente: elaboración propia

5.7 BLOQUE 5: NAGIOS

Es importante que conozcamos los potenciales problemas de nuestra infraestructura antes de que los clientes o los trabajadores de la entidad nos lo comuniquen. Por esto, Nagios nos permitirá condensar toda la información sobre el estado de nuestro despliegue en una única pantalla, fácilmente comprensible en un vistazo.

Una forma de integrar Nagios con nuestra simulación es anexar una máquina virtual Linux (Ubuntu 18.04 LTS en nuestro caso) en Virtual Box o VMWare a la plataforma GNS3. Dicha máquina virtual ha de estar provista de `Apache2` (`apt install apache2`), `Nagios Core` (`apt install nagios3`) y tendrá una dirección IP de la LAN 10.3.1.0/24 (sede central), concretamente la 10.3.1.3. Una vez haya terminado la instalación, podremos acceder a la interfaz de Nagios tecleando en el navegador web `http://localhost/nagios3` (o `http://10.3.1.3/nagios3`) e introduciendo las credenciales establecidas en la fase de instalación. Después, visualizaríamos un *Dashboard* y un menú lateral izquierdo con diversas opciones:

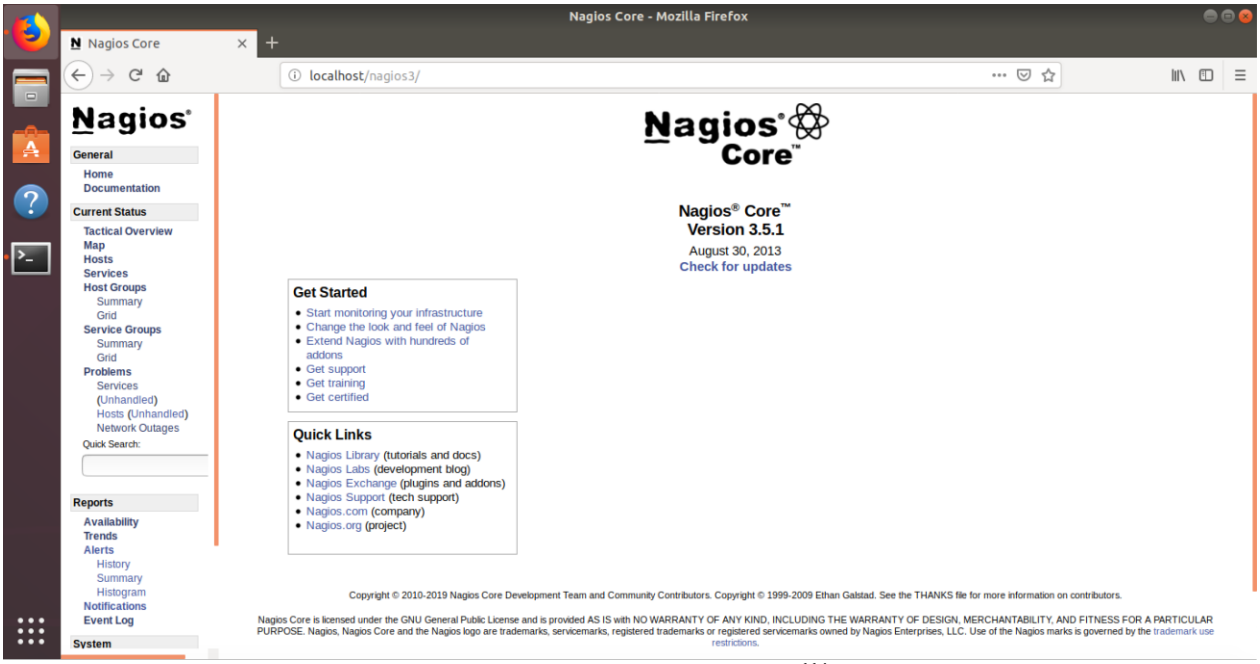


Figura 84 – Nagios. Pantalla inicial[114]

[114] Fuente: elaboración propia a partir de Nagios Core

Dicho esto, en este apartado se expondrá como se ha configurado la herramienta de monitorización Nagios Core en GNS3 y se reflejarán reportes de disponibilidad de la red de determinadas circunstancias.

5.7.1 CONFIGURACIÓN

Las dos carpetas principales de Nagios son `/etc/nagios-plugins/config` y `/etc/nagios3`. En la primera carpeta encontramos ficheros de configuración asociados a los *plugins* de Nagios (es decir, definiciones de comandos para su aplicabilidad a los servicios) y en la segunda carpeta otros archivos de configuración: definición de hosts, intervalos de tiempo para realizar los checks de los servicios; así como otros ficheros para configurar el envío de correos electrónicos, generación de alertas... De ambos directorios se hará especial hincapié en aquéllos que son necesarios modificar para que poder monitorizar todas las interfaces físicas y virtuales de los routers del Piloto.

Como el objetivo principal es comprobar la disponibilidad, principalmente de los enrutadores que componen la red, modificaremos el fichero `/etc/nagios-plugins/config/ping.cfg` y definiremos nuestras propios parámetros para hacer verificaciones de estado de los túneles, siendo aplicable para el resto de las interfaces (ver el resto de configuración en el apartado del *Servicio PING* del Anexo).

```
# 'check_ping_tunnel10'
define command{
        command_name    check_ping_tunnel10
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTTUNNEL10$' -w
'$ARG1$' -c '$ARG2$'
        }

# 'check_ping_tunnel1'
define command{
        command_name    check_ping_tunnel1
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTTUNNEL1$' -w
'$ARG1$' -c '$ARG2$'
        }
```

Más tarde, crearemos un archivo de configuración con extensión `.cfg` dentro del directorio `/etc/nagios3` para cada router de la DMVPN. En estos archivos se encontrarán definidos los servicios que deseamos monitorizar de los hosts

especificados. En esta ocasión, se mostrará un ejemplo del archivo de configuración del HUB1 (`hub1.cfg`). Es importante remarcar que, para poder utilizar más de una dirección IP en un solo host, se ha añadido el símbolo "_" seguido del nombre de la interfaz y su dirección IP. A continuación, se muestra la configuración para realizar los checks PING de las interfaces túnel del host HUB1:

```
define host{
        use                     generic-host
        host_name               HUB1
        alias                   HUB1
       _tunnel10                10.2.0.1
       _tunnel1                 10.1.0.1
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             Tunnel10 - 10.2.0.1
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             Tunnel1 - 10.1.0.1
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }
```

Procedemos de la misma manera para el resto de routers y reiniciamos Nagios mediante el comando `/etc/init.d/nagios3 restart`.

5.7.2 VALIDACIÓN

El estado de un servicio o recurso se determina a través de dos componentes: el estado de servicio o recurso (`OK`, `WARNING`, `UP`, `DOWN`…) y el tipo de estado en el que se encuentra un servicio: SOFT y HARD. Con la finalidad de prevenir falsas alarmas provocadas por problemas transitorios, Nagios realiza un número de reintentos antes de considerar que el problema es realmente real. El número de reintentos se define en la variable '`max_check_attempts`' del fichero '`generic-host_nagios2.cfg`', y en el instante en que se sobrepasa el límite especificado, el problema cambiará de *soft* a *hard*. Así mismo, para considerar un problema como solucionado se efectúa el mismo número de comprobaciones antes de cambiar el estado Hard a solucionado.

Por otro lado, existe un tercer estado denominado Flapping, que es asignado a aquellos servicios que cambian rápidamente entre el estado disponible y algún estado de error. Esta situación puede deberse a situaciones transitorias consideradas como normales (por ejemplo, reinicios periódicos en algún router durante la simulación, apagado e inicio del programa GNS3…) y puede crear diversas notificaciones sobre un falso problema. Por ello, se ha establecido como valor '`max_check_attempts`' a 10 para que estas circunstancias no generen este tipo de alertas.

Hecha esta aclaración inicial, imaginemos que todos los routers e interfaces virtuales y físicas se encuentran en pleno funcionamiento. Si utilizamos la opción "Services" dentro de la sección Current Status, podríamos observar esta situación de la siguiente manera:

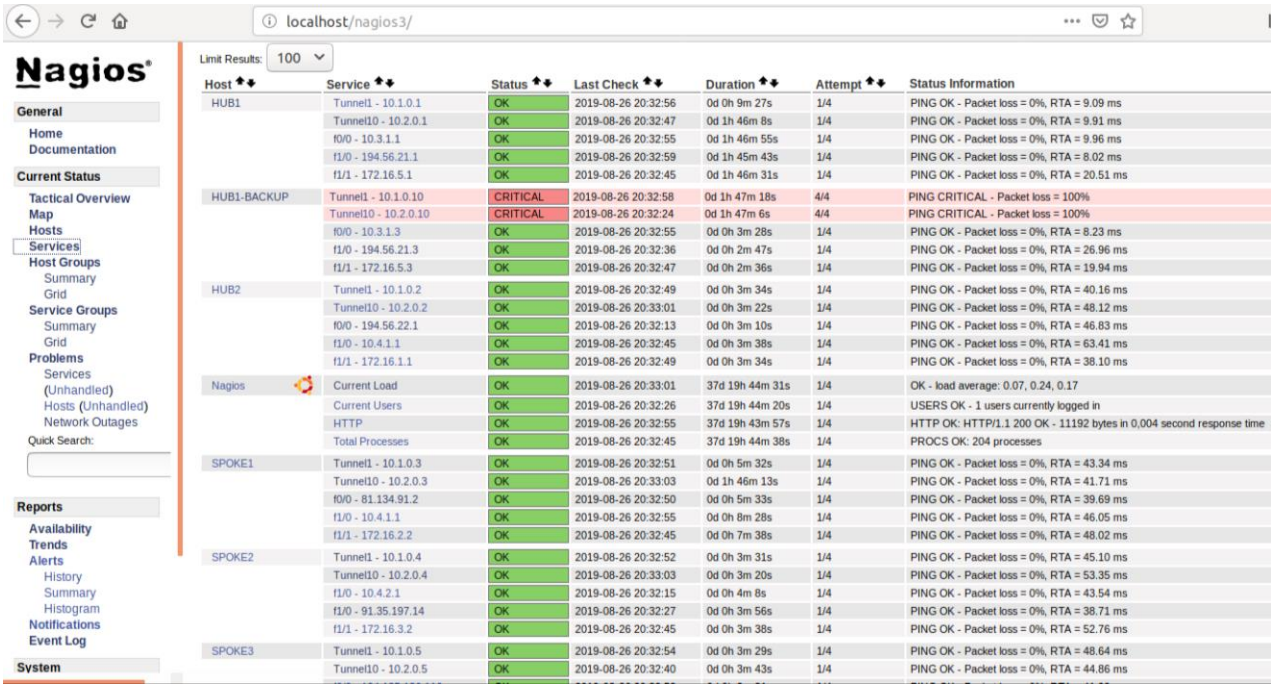


Figura 85 – Nagios. Servicios[115]

E incluso, podríamos obtener información más detallada de cada *check* (información de estado, último y siguiente check a realizar, datos acerca de la latencia…) si hacemos *click* en la interfaz que deseemos:

[115] Fuente: elaboración propia a partir de Nagios Core

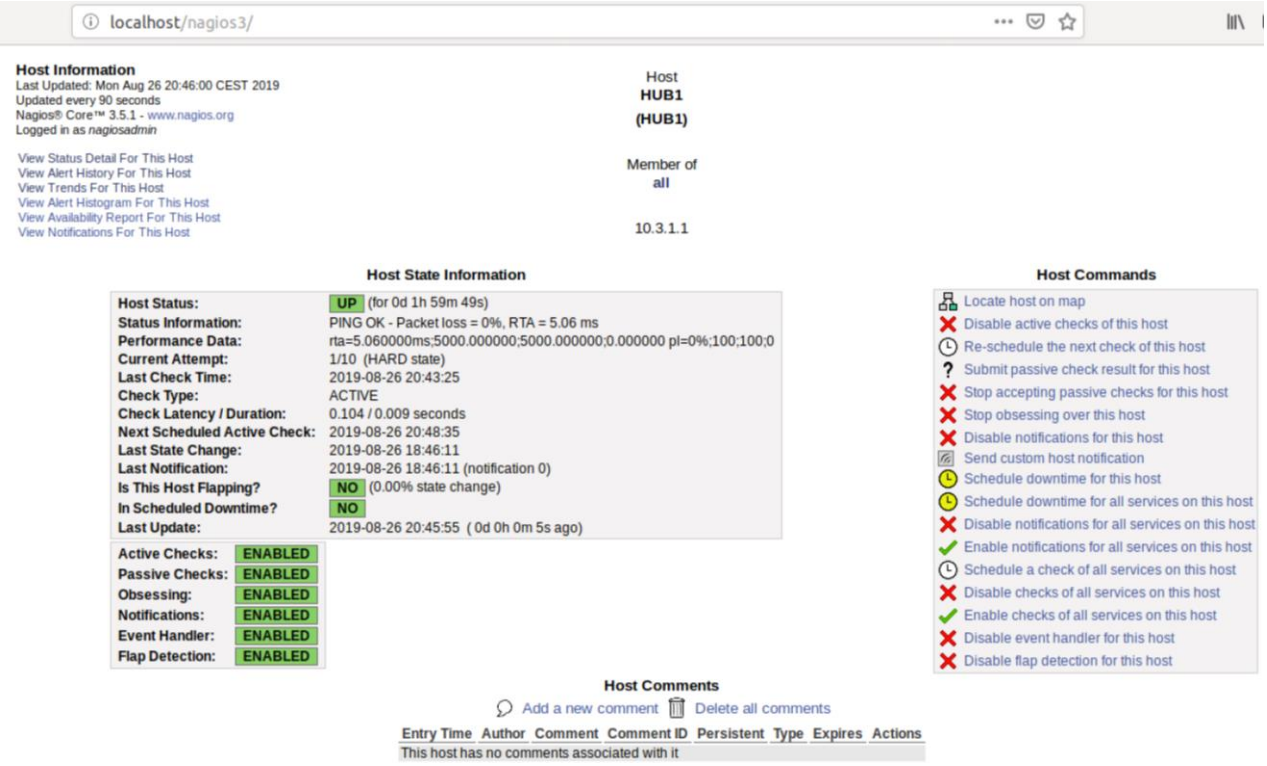


Figura 86 - Nagios. Información detallada de un host[116]

Por otro lado, una de las mejores utilidades que ofrece Nagios Core es la **generación de informes**. Para ello, habrá que especificar qué servicio o dispositivo se quiere obtener información, el periodo de tiempo deseado; así como otros parámetros. La información proporcionada por los informes que se visualizarán a continuación, se ha especificado el servicio Tunnel10 del HUB1 en las últimas 24 horas.

Los **Trends** muestran historiales de cambios de estado que han sufrido un dispositivo o servicio en combinación con la información obtenida de los diferentes *checks*. Se pueden obtener si hacemos *click* en la opción de "*View Trends for this Host*".

[116] Fuente: elaboración propia a partir de Nagios Core

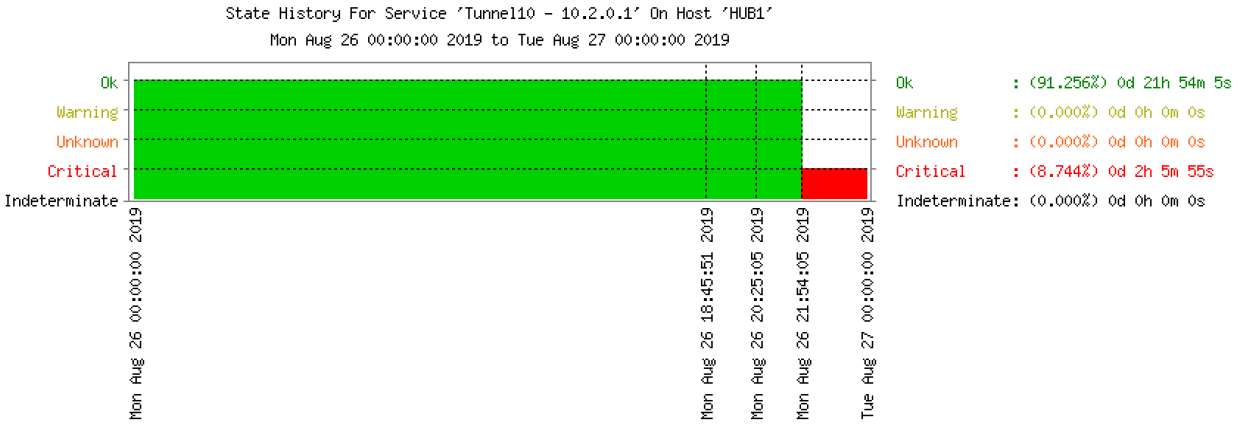


Figura 87 - Nagios. Trends[117]

El informe **Availability** muestra un historial de la cantidad de tiempo que ha estado un dispositivo en un estado determinado. Se ha obtenido haciendo click en la opción ("*View Availability Report for this Host*"). Tanto en esta visualización como en la anterior, observamos estados *Critical* debido al intervalo de tiempo que el Tunnel10 no ha estado operativo por motivos de apagado del programa GNS3:

Service 'Tunnel10 - 10.2.0.1' On Host 'HUB1'

Service State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
OK	Unscheduled	0d 21h 53m 5s	91.186%	91.186%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 21h 53m 5s	91.186%	91.186%
WARNING	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
UNKNOWN	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
CRITICAL	Unscheduled	0d 2h 6m 55s	8.814%	8.814%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 2h 6m 55s	8.814%	8.814%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	1d 0h 0m 0s	100.000%	100.000%

[117] Fuente: elaboración propia a partir de Nagios Core

Service Log Entries:

[View full log entries]

Event Start Time	Event End Time	Event Duration	Event/State Type	Event/State Information
2019-07-20 12:14:41	2019-07-20 12:15:41	0d 0h 1m 0s	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%
2019-07-20 12:15:41	2019-07-20 12:19:39	0d 0h 3m 58s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 6.97 ms
2019-07-20 12:21:41	2019-07-20 12:22:21	0d 0h 0m 40s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 60%, RTA = 9.37 ms
2019-07-20 12:22:21	2019-07-20 12:27:41	0d 0h 5m 20s	SERVICE CRITICAL (HARD)	CRITICAL - Host Unreachable (10.2.0.1)
2019-07-20 12:38:43	2019-07-20 12:43:06	0d 0h 4m 23s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 11.80 ms
2019-07-20 12:54:36	2019-07-20 13:20:01	0d 0h 25m 25s	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%
2019-07-20 13:47:29	2019-07-22 20:17:45	2d 6h 30m 16s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 7.27 ms
2019-07-22 20:18:15	2019-07-24 18:23:21	1d 22h 5m 6s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 7.82 ms
2019-07-24 18:23:21	2019-07-25 14:45:57	0d 20h 22m 36s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 9.03 ms
2019-07-25 14:45:57	2019-07-25 15:05:07	0d 0h 19m 10s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 10.92 ms
2019-07-25 15:05:07	2019-07-25 15:06:07	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-07-25 15:06:07	2019-07-25 15:07:07	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-07-25 15:07:07	2019-07-25 15:08:07	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-07-25 15:08:07	2019-07-25 15:13:57	0d 0h 5m 50s	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%
2019-07-25 15:13:57	2019-07-25 17:55:58	0d 2h 42m 1s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 9.45 ms
2019-07-25 17:55:58	2019-07-25 18:15:08	0d 0h 19m 10s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 9.32 ms
2019-07-25 18:15:08	2019-07-25 18:25:50	0d 0h 10m 42s	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%
2019-07-25 18:26:21	2019-08-18 15:10:38	23d 20h 44m 17s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 10.16 ms
2019-08-18 15:11:08	2019-08-26 18:45:51	8d 3h 34m 43s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 37.75 ms
2019-08-26 18:47:21	2019-08-26 20:25:05	0d 1h 37m 44s	SERVICE OK (HARD)	PING OK - Packet loss = 0%, RTA = 6.61 ms
2019-08-26 21:53:05	2019-08-26 21:54:05	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-08-26 21:54:05	2019-08-27 11:36:51	0d 13h 42m 46s	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%
2019-08-27 11:43:31	2019-08-27 11:44:31	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-08-27 11:44:31	2019-08-27 11:45:31	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-08-27 11:45:31	2019-08-27 11:46:31	0d 0h 1m 0s	SERVICE CRITICAL (SOFT)	PING CRITICAL - Packet loss = 100%
2019-08-27 11:46:31	2019-08-27 00:00:00	1158050440d 19h 13	SERVICE CRITICAL (HARD)	PING CRITICAL - Packet loss = 100%

Figura 88 - Nagios. Availability[118]

Por último, el informe **Alert** muestra, en modo de histograma, la cantidad de alertas que se han producido para un dispositivo o servicio en un intervalo de tiempo. Para generar este informe haremos *click* en la opción "*View Alert Histogram for This Host*":

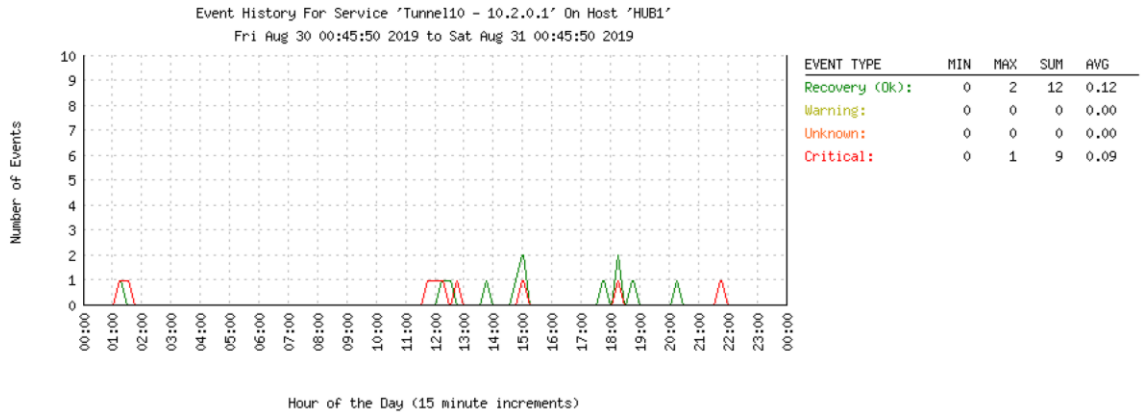


Figura 89 - Nagios. Alerts[119]

Otra característica interesante, es la **actualización de los estados** de los dispositivos en la interfaz gráfica y la generación de **alertas**. Para ello, vamos a simular escenarios de caída de routers e interfaces para visualizar el comportamiento de la plataforma. Supongamos, pues, que la sede central pierde la comunicación con

[118] Fuente: elaboración propia a partir de Nagios Core
[119] Fuente: elaboración propia a partir de Nagios Core

la oficina mediana, ya que su router, SPOKE2, se ha apagado por un corte de luz puntual. Para simular esta situación, apagaremos el router SPOKE2 (opción *Stop* en GNS3):

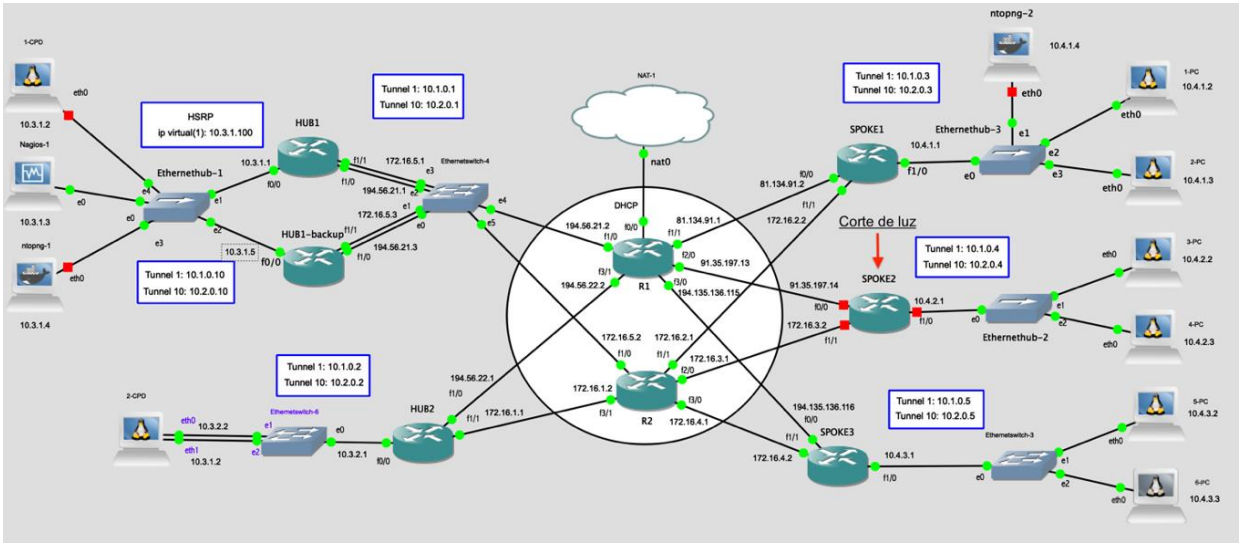


Figura 90 - Nagios. Simulación de corte de luz en una oficina[120]

Tras esperar unos cuantos minutos hasta el siguiente check, visualizaremos el cambio de estado de la interfaz gráfica de Nagios:

	f1/0 - 194.56.21.1	OK	2019-08-26 20:59:59	0d 2h 12m 42s	1/4	PING OK - Packet loss = 0%, RTA = 9.77 ms
	f1/1 - 172.16.5.1	OK	2019-08-26 20:59:45	0d 2h 13m 30s	1/4	PING OK - Packet loss = 0%, RTA = 19.47 ms
HUB1-BACKUP	Tunnel1 - 10.1.0.10	CRITICAL	2019-08-26 20:59:58	0d 2h 14m 17s	4/4	PING CRITICAL - Packet loss = 100%
	Tunnel10 - 10.2.0.10	CRITICAL	2019-08-26 20:59:24	0d 2h 14m 5s	4/4	PING CRITICAL - Packet loss = 100%
	f0/0 - 10.3.1.3	OK	2019-08-26 20:59:55	0d 0h 30m 27s	1/4	PING OK - Packet loss = 0%, RTA = 9.32 ms
	f1/0 - 194.56.21.3	OK	2019-08-26 20:59:36	0d 0h 29m 46s	1/4	PING OK - Packet loss = 0%, RTA = 28.51 ms
	f1/1 - 172.16.5.3	OK	2019-08-26 20:59:47	0d 0h 29m 35s	1/4	PING OK - Packet loss = 0%, RTA = 20.76 ms
HUB2	Tunnel1 - 10.1.0.2	OK	2019-08-26 20:59:49	0d 0h 30m 33s	1/4	PING OK - Packet loss = 0%, RTA = 38.75 ms
	Tunnel10 - 10.2.0.2	OK	2019-08-26 21:00:01	0d 0h 30m 21s	1/4	PING OK - Packet loss = 0%, RTA = 49.75 ms
	f0/0 - 194.56.22.1	OK	2019-08-26 20:59:13	0d 0h 30m 9s	1/4	PING OK - Packet loss = 0%, RTA = 47.52 ms
	f1/0 - 10.4.1.1	OK	2019-08-26 20:59:45	0d 0h 30m 37s	1/4	PING OK - Packet loss = 0%, RTA = 47.57 ms
	f1/1 - 172.16.1.1	OK	2019-08-26 20:59:49	0d 0h 30m 33s	1/4	PING OK - Packet loss = 0%, RTA = 38.37 ms
Nagios	Current Load	OK	2019-08-26 21:00:01	37d 20h 11m 30s	1/4	OK - load average: 0.00, 0.01, 0.06
	Current Users	OK	2019-08-26 20:59:26	37d 20h 11m 19s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	2019-08-26 20:59:55	37d 20h 10m 56s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0,002 second response time
	Total Processes	OK	2019-08-26 20:59:45	37d 20h 11m 37s	1/4	PROCS OK: 205 processes
SPOKE1	Tunnel1 - 10.1.0.3	OK	2019-08-26 20:59:51	0d 0h 32m 31s	1/4	PING OK - Packet loss = 0%, RTA = 48.46 ms
	Tunnel10 - 10.2.0.3	OK	2019-08-26 21:00:03	0d 2h 13m 12s	1/4	PING OK - Packet loss = 0%, RTA = 38.81 ms
	f0/0 - 81.134.91.2	OK	2019-08-26 20:59:50	0d 0h 32m 32s	1/4	PING OK - Packet loss = 0%, RTA = 48.00 ms
	f1/0 - 10.4.1.1	OK	2019-08-26 20:59:55	0d 0h 35m 27s	1/4	PING OK - Packet loss = 0%, RTA = 47.37 ms
	f1/1 - 172.16.2.2	OK	2019-08-26 20:59:45	0d 0h 34m 37s	1/4	PING OK - Packet loss = 0%, RTA = 48.32 ms
SPOKE2	Tunnel1 - 10.1.0.4	CRITICAL	2019-08-26 20:59:52	0d 0h 2m 30s	1/4	PING CRITICAL - Packet loss = 100%
	Tunnel10 - 10.2.0.4	CRITICAL	2019-08-26 21:00:03	0d 0h 2m 19s	1/4	PING CRITICAL - Packet loss = 100%
	f1/0 - 10.4.2.1	CRITICAL	2019-08-26 20:59:15	0d 0h 2m 7s	1/4	CRITICAL - Host Unreachable (10.4.2.1)
	f1/0 - 91.35.197.14	CRITICAL	2019-08-26 20:59:27	0d 0h 1m 55s	2/4	PING CRITICAL - Packet loss = 100%
	f1/1 - 172.16.3.2	CRITICAL	2019-08-26 20:59:45	0d 0h 2m 37s	1/4	PING CRITICAL - Packet loss = 100%
SPOKE3	Tunnel1 - 10.1.0.5	OK	2019-08-26 20:59:54	0d 0h 30m 28s	1/4	PING OK - Packet loss = 0%, RTA = 47.38 ms
	Tunnel10 - 10.2.0.5	OK	2019-08-26 20:59:40	0d 0h 30m 42s	1/4	PING OK - Packet loss = 0%, RTA = 46.26 ms
	f0/0 - 194.135.136.116	OK	2019-08-26 20:59:52	0d 0h 30m 30s	1/4	PING OK - Packet loss = 0%, RTA = 48.49 ms
	f1/0 - 10.4.3.1	OK	2019-08-26 21:00:05	0d 0h 30m 17s	1/4	PING OK - Packet loss = 0%, RTA = 39.38 ms
	f1/1 - 172.16.4.2	OK	2019-08-26 20:59:45	0d 0h 30m 37s	1/4	PING OK - Packet loss = 0%, RTA = 40.40 ms

Figura 91 - Nagios. Cambio de estado[121]

[120] Fuente: elaboración propia

[121] Fuente: elaboración propia a partir de Nagios Core

Ahora, supongamos que el Túnel principal y secundario del HUB1 están caídos por motivos de mantenimiento y configuración del router por parte del personal técnico de redes de la organización. Esta situación la podríamos simular, por ejemplo, apagando las interfaces virtuales Tunnel10 y Tunnel1, mediante el comando `shutdown` en la configuración del router. Observamos que los túneles de HUB1-backup se han levantado:

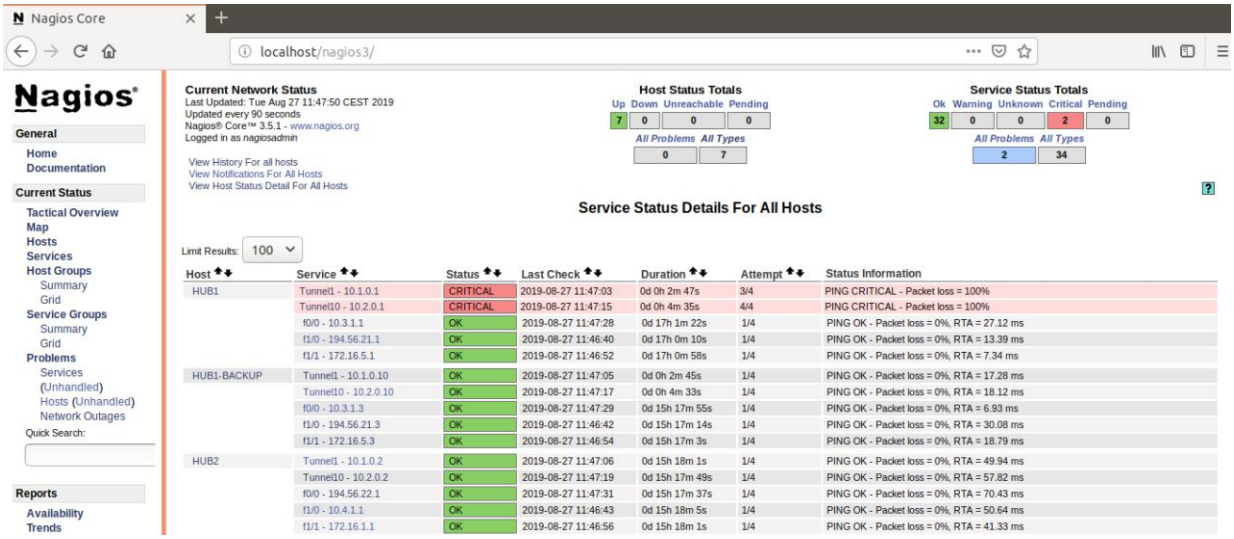


Figura 92 - Nagios. Cambio de estado II[122]

[122] Fuente: elaboración propia a partir de Nagios Core

CONCLUSIONES Y LÍNEAS FUTURAS

En los entornos de redes de comunicaciones (públicas o corporativas) han surgido tecnologías que facilitan el uso de Internet como medio de transmisión para establecer enlaces de datos entre sedes distantes. Entre las tecnologías surgidas, destaca DMVPN, que permite paliar los problemas de escalabilidad y rigidez existentes en las VPN seguras basadas en IPSec y en otros protocolos incapaces de transportar tramas de tipo Multicast dentro de túneles VPN, utilizando para ello protocolos de encaminamiento dinámicos. Generalmente, la implementación de redes DMVPN se aplican a redes corporativas de gran tamaño, como entidades bancarias y empresas con muchas delegaciones donde la disponibilidad de los recursos que ofrece la red es un factor determinante en el funcionamiento de la estructura empresarial. Además, también permite la configuración de redes compuestas de accesos a Internet a través de distintos proveedores, y con más de uno por sede.

En este proyecto, hemos diseñado y simulado el despliegue de una red DMVPN virtual utilizando el software de simulación de red GNS3, aplicable a una entidad con dos centros de cálculo y 200 sedes. Para ello, hemos elaborado un diagrama de bloques general de las diferentes partes que compondrán la solución y lograr así, tanto los objetivos propuestos como los resultados deseados.

1. En primer lugar, realizamos una estimación del consumo de ancho de banda de para cada tipo de oficina y HUB en función de su tamaño, partiendo como base de un ancho de banda estándar consumido por un usuario y un factor de simultaneidad obtenido empíricamente. A tenor de los resultados conseguidos, se observó que cada red requería un modelo de router concreto. Se escogió la serie Cisco ISR 4000 por motivos de eficiencia y funcionalidades y tuvimos en cuenta *benchmarks* realizados por otras para controlar el consumo de CPU de cada router y evitar problemas de congestionamiento y pérdidas de paquetes en el caso aplicabilidad real a una entidad.
2. En segundo lugar, implementamos y configuramos una red con 2 NBMA con tecnologías WAN de distintos operadores, comprobamos que la aplicación de la

tecnología DMVPN ofrece todas las ventajas de una VPN, ofrece escalabilidad en cuanto a sedes remotas, facilidad en la reconfiguración de las topologías, independencia con la distancia y del proveedor de acceso. Además, comprobamos la presencia de la norma RFC 1191 en la topología para hacer frente a la fragmentación IP; así como la alta disponibilidad y redundancia mediante pruebas de conmutación de red, cuyos tiempos oscilaban entre 5 y 24 segundos.

3. Debido a la convergencia de una gran cantidad de servicios diferentes con características dispares (aplicaciones interactivas, aplicaciones de transferencias masivas de datos, tráfico de datos…), creamos políticas QoS basadas en clases, mediante el QoS MCLI. Luego, generamos flujos de datos que permitieron la comprobación del correcto funcionamiento de las políticas QoS establecidas mediante el software `iperf`.
4. Comprobamos el soporte DMVPN para Multicast mediante el envío de un único flujo de datos con destino a 'N' máquinas en las redes finales (*streaming multicast*), permitiendo de esta manera la difusión de información con un gran ahorro de ancho de banda.
5. Logramos que las transmisiones de datos sean seguras mediante la aplicabilidad de IPSec con IKEv2 como tecnología de cifrado para la tunelización mGRE con NHRP para proporcionar confidencialidad, autenticidad y no repudio.
6. Por último, integramos Nagios Core en GNS3 como herramienta de monitorización para obtener reportes de disponibilidad y controlar el estado de la red en todo momento.

En futuras investigaciones podemos centrar nuestro foco de interés en la integración de componentes de seguridad. En el caso de que la organización tuviese servicios publicados en Internet y éstos se encuentren ubicados en la sede central y/o en el centro de respaldo, aplicaremos la configuración **Firewall Zone Based** para crear tres zonas: INSIDE (equipos de las oficinas), OUTSIDE (WAN) y DMZ (servicios públicos en Internet). Siguiendo la estela de la seguridad, implementaremos un **IDS/IPS**, como *Snort* o *Suricata*, para la generación de alertas en caso de detección

de alguna intrusión. Además, si bien es cierto que en el presente proyecto, se han centralizado los *syslogs* en los dos CPDs ubicados en la sede central y en el centro de respaldo, se podría instalar un **SIEM** que centralice y correlacione todos los *logs* generados en la red con la finalidad de proporcionar una respuesta proactiva ante un ataque a un servicio público en Internet o la detección de comportamiento inusual en las redes LAN de las oficinas. También, sería adecuado la disposición de programas informáticos que permitan la **detección de vulnerabilidades** en los servicios web, como **Nessus** o **Acunetix** y la integración de **Cisco Umbrella** para proporcionar una capa adicional de seguridad en los PCs existentes en cada oficina y sede.

Pero también este proyecto plantea la inclusión de algunas otras tecnologías aplicables a los encaminadores Cisco reales. Mediante la inclusión de la tecnología **BFD** en los túneles principales y secundarios, se podría incrementar la velocidad de detección y recuperación de fallos de la red. En el router HUB1 y en HUB1-backup, se podría integrar, también, la tecnología **SSO**, que permitiría mantener información de estado y de sesión durante la conmutación, proporcionando una mayor disponibilidad de red. También, se puede introducir la tecnología **PFR**, que proporcionaría una selección dinámica del mejor camino para una aplicación para llegar a su destino y control caudales de red consumidos del ancho de banda. Por otra parte, los **Offset Lists**, permitirían incrementar y optimizar las métricas EIGRP entrantes y salientes a las rutas aprendidas vía EIGRP. Por último, las políticas QoS podrían ser aplicadas en los túneles principales y secundarios (en lugar de en las interfaces físicas), utilizando la tecnología **Per-Tunnel QoS** de Cisco e incluso se podría agregar la tecnología **NetFlow** para obtener información detallada en todo momento de los flujos que circulan en la red.

REFERENCIAS

Cisco Integrated Services Router: Architectural Overview and Use Cases. (s.f.). Obtenido de Cisco: https://www.ciscolive.com/c/dam/r/ciscolive/us/docs/2018/pdf/BRKARC-3001.pdf

Cisco Umbrella. (s.f.). Obtenido de Cisco: https://www.cisco.com/c/dam/global/es_mx/solutions/security/pdf/cisco-only-umbrella-mx.pdf

Crespo, L. M. (s.f.). *Redes NBMA: Introducción a las redes DMVPN.* Universidad de Alicante.

Bartlett, G., & Inamdar, A. (2016). *IKEv2 IPSec Virtual Private Networks: Understanding and Deploying IKEv2, IPSec VPNs, and FlexVPN in Cisco IOS.* Cisco Press.

Berná, J. Á. (s.f.). *Interconexión de Redes. Práctica 2. Redes DMVPN.* Universidad de Alicante.

Berná, J. A. (s.f.). *Interconexión de Redes: Práctica 3. Calidad de Servicio (QoS).* Universidad de Alicante.

Diseño y configuración de IPS, IDS y SIEM en Sistemas de Control Industrial. (Noviembre de 2017). Obtenido de INCIBE-CERT: https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_diseno_configuracion_ips_ids_siem_en_sci.pdf

Dynamic Multipoint VPN Configuration Guide, Cisco IOS Release 15M&T. (s.f.). Obtenido de Cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/15-mt/sec-conn-dmvpn-15-mt-book.pdf

F. Kurose, J., & W. Ross, K. (2009). *Redes de Computadoras: Un Enfoque Descendente basado en Internet.* Pearson Addison Wesley.

Iglesias, S. P. (s.f.). *Facultad Regional La Plata.* Obtenido de http://www.frlp.utn.edu.ar/materias/internetworking/apuntes/IPSec/ipsec.pdf

Introduction to EIGRP. (s.f.). Obtenido de Cisco: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html

iPerf - The ultimate speed test tool for TCP, UDP and SCTP. (s.f.). Obtenido de iPerf: https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2015/pdf/BRKCOL-2616.pdf

Joseph, V., & Chapman, B. (2009). *Deploying QoS for Cisco IP and Next Generation Networks.* Morgan Kaufmann.

Official manuals, documentation, video tutorials, and FAQs for Nagios solutions. (s.f.). Obtenido de Nagios: https://www.nagios.org/documentation/

Overview, I. M. (s.f.). *IP Multicast Technology Overview.* Obtenido de Cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipmulti_pim/configuration/xe-16/imc-pim-xe-16-book/imc-tech-oview.pdf

Path MTU Discovery, RFC 1191.

Performance Analysis: Cisco ISR 4000 Family. Models 4321, 4331, 4351, 4431 & 4451. (2015). Obtenido de Miercom: https://www.cisco.com/c/dam/en/us/products/collateral/routers/4000-series-integrated-services-routers-isr/miercom-isr4k-report.pdf

Protocolo GRE, RFC 1701 RFC 2784.

Protocolo HSRP, RFC 2281.

Protocolo mGRE, RFC 2547.

Protocolo NHRP, RFC 2333 .

QoS Strategies and Smart Media Techniques for Collaboration Deployments. (s.f.). Obtenido de Cisco Live: https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2015/pdf/BRKCOL-2616.pdf

The official guide and reference for GNS3. (s.f.). Obtenido de GNS3: https://docs.gns3.com

The TCP MSS and related Topics, RFC 879.

Tiso, J. (2011). *Designing Cisco Network Service Architectures.* Cisco Press.

Ubuntu Server Guide - Monitoring - Nagios. (s.f.). Obtenido de Ubuntu: https://help.ubuntu.com/lts/serverguide/nagios.html

Wallace, K., Hucaby, D., & Lacoste, R. (s.f.). *CCNP Routing and Switching V2.0 Official Cert Guide Library.*

ANEXO

CONFIGURACIÓN DE LOS ROUTERS

HUB1

```
hostname HUB1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
ip name-server 8.8.8.8
ip multicast-routing
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
group 15
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
pre-shared-key TFG_2019_DMVPN
!
!
!
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
```

```
match destination-address mac CA08.117E.001C
class-map match-any ISP2.destination
match destination-address mac CA03.1122.001C
class-map match-all pInteractive
match dscp af31
class-map match-all Interactive
match access-group 102
class-map match-any ISP1.source
match source-address mac CA08.117E.001C
class-map match-any ISP2.source
match source-address mac CA03.1122.001C
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
policy-map GLOBAL_DOWN
class qos1
shape average 101693000
service-policy child
class qos2
shape average 101693000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 101693000
service-policy pchild
class ISP2.destination
```

```
shape average 101693000
service-policy pchild
!
!
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
!
!
!
interface Tunnel1
bandwidth 10000
ip address 10.1.0.1 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 100000
ip nhrp holdtime 15
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 300
keepalive 3 2
tunnel source 194.56.21.1
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.1 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip pim nbma-mode
ip pim sparse-mode
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 200000
ip nhrp holdtime 15
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 200
keepalive 3 2
tunnel source 172.16.5.1
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
```

```
interface FastEthernet0/0
ip address 10.3.1.1 255.255.255.0
ip nat inside
ip pim sparse-mode
standby 1 ip 10.3.1.100
standby 1 priority 110
standby 1 preempt
ip tcp adjust-mss 1360
duplex full
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/0
ip address 194.56.21.1 255.255.255.0
ip nat outside
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/1
ip address 172.16.5.1 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet2/0
no ip address
duplex full
!
!
router eigrp 100
network 10.0.0.0
passive-interface FastEthernet1/1
passive-interface FastEthernet1/0
!
ip nat inside source list 1 interface FastEthernet1/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip pim rp-address 10.2.0.1
ip route 0.0.0.0 0.0.0.0 194.56.21.2
ip route 172.16.0.0 255.255.0.0 172.16.5.2
!
logging facility local0
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.3.1.0 0.0.0.255
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
```

```
!
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

HUB1-BACKUP

```
hostname HUB1-backup
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
group 15
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
```

```
pre-shared-key TFG_2019_DMVPN
!
¡
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
match destination-address mac CA08.117E.001C
class-map match-any ISP2.destination
match destination-address mac CA03.1122.001C
class-map match-all pInteractive
match dscp af31
class-map match-any ISP1.source
match source-address mac CA08.117E.001C
class-map match-all Interactive
match access-group 102
class-map match-any ISP2.source
match source-address mac CA03.1122.001C
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
```

```
policy-map GLOBAL_DOWN
class qos1
shape average 101693000
service-policy child
class qos2
shape average 101693000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 101693000
service-policy pchild
class ISP2.destination
shape average 101693000
service-policy pchild
!
!
!
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
¡
interface Tunnel1
bandwidth 10000
ip address 10.1.0.10 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 100000
ip nhrp holdtime 15
ip tcp adjust-mss 1360
delay 300
keepalive 5 3
tunnel source 194.56.21.3
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.10 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
```

```
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 200000
ip nhrp holdtime 15
ip tcp adjust-mss 1360
delay 200
keepalive 5 3
tunnel source 172.16.5.3
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface FastEthernet0/0
ip address 10.3.1.5 255.255.255.0
ip nat inside
standby 1 ip 10.3.1.100
standby 1 preempt
ip tcp adjust-mss 1360
duplex full
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/0
ip address 194.56.21.3 255.255.255.0
ip nat outside
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/1
ip address 172.16.5.3 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet2/0
no ip address
shutdown
duplex full
!
!
router eigrp 100
network 10.0.0.0
offset-list 20 out 5000
passive-interface FastEthernet1/1
passive-interface FastEthernet1/0
!
ip nat inside source list 1 interface FastEthernet1/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 194.56.21.2
ip route 172.16.0.0 255.255.0.0 172.16.5.2
```

```
!
logging facility local0
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.3.1.0 0.0.0.255
access-list 20 permit any
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

HUB2

```
hostname HUB2
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
```

```
group 15
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
pre-shared-key TFG_2019_DMVPN
!
!
!
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
match destination-address mac CA08.117E.0055
class-map match-any ISP2.destination
match destination-address mac CA03.1122.0055
class-map match-all pInteractive
match dscp af31
class-map match-any ISP1.source
match source-address mac CA08.117E.0055
class-map match-all Interactive
match access-group 102
class-map match-any ISP2.source
match source-address mac CA03.1122.0055
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
```

```
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
policy-map GLOBAL_DOWN
class qos1
shape average 660720000
service-policy child
class qos2
shape average 660720000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 660720000
service-policy pchild
class ISP2.destination
shape average 660720000
service-policy pchild
!
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
!
interface Tunnel1
bandwidth 10000
ip address 10.1.0.2 255.255.255.0
no ip redirects
ip nat inside
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.1.0.1 194.56.21.1
ip nhrp map multicast 194.56.21.1
ip nhrp map 10.1.0.10 194.56.21.3
ip nhrp map multicast 194.56.21.3
ip nhrp network-id 100000
ip nhrp holdtime 15
ip nhrp nhs 10.1.0.1
ip nhrp nhs 10.1.0.10
ip tcp adjust-mss 1360
delay 350
```

```
keepalive 3 2
tunnel source 194.56.22.1
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.2 255.255.255.0
no ip redirects
ip nat inside
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.2.0.1 172.16.5.1
ip nhrp map multicast 172.16.5.1
ip nhrp map 10.2.0.10 172.16.5.3
ip nhrp map multicast 172.16.5.3
ip nhrp network-id 200000
ip nhrp holdtime 300
ip nhrp nhs 10.2.0.1
ip nhrp nhs 10.2.0.10
ip nhrp nhs cluster 1 max-connections 2
ip nhrp nhs fallback 15
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 250
keepalive 3 2
tunnel source 172.16.1.1
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface FastEthernet0/0
ip address 10.3.2.1 255.255.255.0
ip nat inside
ip tcp adjust-mss 1360
duplex full
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/0
ip address 194.56.22.1 255.255.255.0
ip nat outside
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/1
ip address 172.16.1.1 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
```

```
interface FastEthernet2/0
no ip address
shutdown
duplex full
!
!
router eigrp 100
network 10.0.0.0
redistribute static
passive-interface FastEthernet1/1
passive-interface FastEthernet1/0
!
ip nat inside source list 1 interface FastEthernet1/0 overload
ip nat inside source list 20 interface FastEthernet0/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 194.56.22.2
ip route 10.3.1.0 255.255.255.0 10.3.2.2 200
ip route 172.16.0.0 255.255.0.0 172.16.1.2
!
logging facility local0
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.3.1.2
access-list 1 permit 10.3.2.0 0.0.0.255
access-list 20 permit 10.3.1.0 0.0.0.255
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

SPOKE1

```
hostname SPOKE1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
ip name-server 8.8.8.8
ip multicast-routing
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
group 15
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
pre-shared-key TFG_2019_DMVPN
!
!
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
match destination-address mac CA08.117E.001D
class-map match-any ISP2.destination
match destination-address mac CA03.1122.001D
class-map match-all pInteractive
match dscp af31
class-map match-any ISP1.source
match source-address mac CA08.117E.001D
```

```
class-map match-all Interactive
match access-group 102
class-map match-any ISP2.source
match source-address mac CA03.1122.001D
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
policy-map GLOBAL_DOWN
class qos1
shape average 15840000
service-policy child
class qos2
shape average 15840000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 15840000
service-policy pchild
class ISP2.destination
shape average 15840000
service-policy pchild
!
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
```

```
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
!
!
interface Tunnel1
bandwidth 10000
ip address 10.1.0.3 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.1.0.2 194.56.22.1
ip nhrp map multicast 194.56.22.1
ip nhrp map 10.1.0.1 194.56.21.1
ip nhrp map multicast 194.56.21.1
ip nhrp map 10.1.0.10 194.56.21.3
ip nhrp map multicast 194.56.21.3
ip nhrp network-id 100000
ip nhrp holdtime 300
ip nhrp nhs 10.1.0.2
ip nhrp nhs 10.1.0.1
ip nhrp nhs 10.1.0.10
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 300
if-state nhrp
keepalive 3 2
tunnel source 81.134.91.2
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.3 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip pim sparse-mode
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.2.0.2 172.16.1.1
ip nhrp map multicast 172.16.1.1
ip nhrp map 10.2.0.1 172.16.5.1
ip nhrp map multicast 172.16.5.1
ip nhrp map 10.2.0.10 172.16.5.3
ip nhrp map multicast 172.16.5.3
ip nhrp network-id 200000
ip nhrp holdtime 300
ip nhrp nhs 10.2.0.2
ip nhrp nhs 10.2.0.1
ip nhrp nhs 10.2.0.10
```

```
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 200
if-state nhrp
keepalive 3 2
tunnel source 172.16.2.2
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface FastEthernet0/0
ip address 81.134.91.2 255.255.255.0
ip nat outside
duplex full
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/0
ip address 10.4.1.1 255.255.255.0
ip nat inside
ip pim sparse-mode
ip tcp adjust-mss 1360
speed auto
duplex auto
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/1
ip address 172.16.2.2 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet2/0
no ip address
shutdown
duplex full
!
!
router eigrp 100
network 10.0.0.0
passive-interface FastEthernet1/1
passive-interface FastEthernet1/0
passive-interface FastEthernet0/0
!
ip nat inside source list 1 interface FastEthernet0/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip pim rp-address 10.2.0.1
ip route 0.0.0.0 0.0.0.0 81.134.91.1
ip route 172.16.0.0 255.255.0.0 172.16.2.1
!
logging facility local0
```

```
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.4.1.0 0.0.0.255
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

SPOKE2

```
hostname SPOKE2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
ip name-server 8.8.8.8
ip multicast-routing
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
group 15
```

```
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
pre-shared-key TFG_2019_DMVPN
!
!
!
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
match destination-address mac CA08.117E.0038
class-map match-any ISP2.destination
match destination-address mac CA03.1122.0038
class-map match-all pInteractive
match dscp af31
class-map match-any ISP1.source
match source-address mac CA08.117E.0038
class-map match-all Interactive
match access-group 102
class-map match-any ISP2.source
match source-address mac CA03.1122.0038
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
```

```
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
policy-map GLOBAL_DOWN
class qos1
shape average 6340000
service-policy child
class qos2
shape average 6340000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 6340000
service-policy pchild
class ISP2.destination
shape average 6340000
service-policy pchild
!
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
!
!
interface Tunnel1
bandwidth 10000
ip address 10.1.0.4 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.1.0.2 194.56.22.1
ip nhrp map multicast 194.56.22.1
ip nhrp map 10.1.0.1 194.56.21.1
ip nhrp map multicast 194.56.21.1
ip nhrp map 10.1.0.10 194.56.21.3
ip nhrp map multicast 194.56.21.3
ip nhrp network-id 100000
ip nhrp holdtime 300
ip nhrp nhs 10.1.0.2
```

```
ip nhrp nhs 10.1.0.1
ip nhrp nhs 10.1.0.10
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 300
if-state nhrp
keepalive 3 2
tunnel source 91.35.197.14
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.4 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip pim sparse-mode
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.2.0.2 172.16.1.1
ip nhrp map multicast 172.16.1.1
ip nhrp map 10.2.0.1 172.16.5.1
ip nhrp map multicast 172.16.5.1
ip nhrp map 10.2.0.10 172.16.5.3
ip nhrp map multicast 172.16.5.3
ip nhrp network-id 200000
ip nhrp holdtime 300
ip nhrp nhs 10.2.0.2
ip nhrp nhs 10.2.0.1
ip nhrp nhs 10.2.0.10
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 200
if-state nhrp
keepalive 3 2
tunnel source 172.16.3.2
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface FastEthernet0/0
ip address 91.35.197.14 255.255.255.0
ip nat outside
duplex full
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/0
ip address 10.4.2.1 255.255.255.0
ip nat inside
ip pim sparse-mode
ip tcp adjust-mss 1360
speed auto
duplex auto
```

```
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/1
ip address 172.16.3.2 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet2/0
no ip address
duplex full
!
router eigrp 100
network 10.0.0.0
passive-interface FastEthernet0/0
passive-interface FastEthernet1/1
!
ip nat inside source list 1 interface FastEthernet0/0 overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.2.0.1
ip route 0.0.0.0 0.0.0.0 91.35.197.13
ip route 172.16.0.0 255.255.0.0 172.16.3.1
!
logging facility local0
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.4.2.0 0.0.0.255
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

SPOKE3

```
hostname SPOKE3
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
ip name-server 8.8.8.8
ip multicast-routing
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
crypto ikev2 proposal prop-1
encryption aes-cbc-256
integrity sha256
group 15
!
crypto ikev2 policy site-policy
proposal prop-1
!
crypto ikev2 keyring cisco-ikev2-keyring
peer dmvpn-node
description symmetric pre-shared key for the hub/spoke
address 0.0.0.0 0.0.0.0
pre-shared-key TFG_2019_DMVPN
!
!
!
crypto ikev2 profile cisco-ikev2-profile
match fvrf any
match identity remote address 0.0.0.0
authentication remote pre-share
authentication local pre-share
keyring local cisco-ikev2-keyring
!
!
!
ip tcp synwait-time 5
!
class-map match-all RealTime
match access-group 101
class-map match-any ISP1.destination
match destination-address mac CA08.117E.0054
class-map match-any ISP2.destination
match destination-address mac CA03.1122.0054
class-map match-all pInteractive
```

```
match dscp af31
class-map match-any ISP1.source
match source-address mac CA08.117E.0054
class-map match-all Interactive
match access-group 102
class-map match-any ISP2.source
match source-address mac CA03.1122.0054
class-map match-any qos1
match qos-group 1
class-map match-any qos2
match qos-group 2
class-map match-all pRealTime
match dscp ef
!
policy-map MARKING_DOWN
class ISP1.source
set qos-group 1
class ISP2.source
set qos-group 2
policy-map pchild
class pRealTime
priority percent 30
class pInteractive
bandwidth percent 20
class class-default
bandwidth percent 10
policy-map child
class RealTime
priority percent 30
set dscp ef
class Interactive
bandwidth percent 20
set dscp af31
class class-default
bandwidth percent 10
set dscp default
policy-map GLOBAL_DOWN
class qos1
shape average 2530000
service-policy child
class qos2
shape average 2530000
service-policy child
policy-map MARKING_UP
class RealTime
set dscp ef
class Interactive
set dscp af31
policy-map GLOBAL_UP
class ISP1.destination
shape average 2530000
service-policy pchild
class ISP2.destination
shape average 2530000
service-policy pchild
!
!
```

```
!
crypto ipsec transform-set cisco-ts esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile cisco-ipsec-ikev2
set transform-set cisco-ts
set ikev2-profile cisco-ikev2-profile
!
!
!
interface Tunnel1
bandwidth 10000
ip address 10.1.0.5 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.1.0.2 194.56.22.1
ip nhrp map multicast 194.56.22.1
ip nhrp map 10.1.0.1 194.56.21.1
ip nhrp map multicast 194.56.21.1
ip nhrp map 10.1.0.10 194.56.21.3
ip nhrp map multicast 194.56.21.3
ip nhrp network-id 100000
ip nhrp holdtime 300
ip nhrp nhs 10.1.0.2
ip nhrp nhs 10.1.0.1
ip nhrp nhs 10.1.0.10
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 300
if-state nhrp
keepalive 3 2
tunnel source 194.135.136.116
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface Tunnel10
bandwidth 10000
ip address 10.2.0.5 255.255.255.0
no ip redirects
ip mtu 1400
no ip next-hop-self eigrp 100
no ip split-horizon eigrp 100
ip pim sparse-mode
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp map 10.2.0.2 172.16.1.1
ip nhrp map multicast 172.16.1.1
ip nhrp map 10.2.0.1 172.16.5.1
ip nhrp map multicast 172.16.5.1
ip nhrp map 10.2.0.10 172.16.5.3
ip nhrp map multicast 172.16.5.3
ip nhrp network-id 200000
```

```
ip nhrp holdtime 300
ip nhrp nhs 10.2.0.2
ip nhrp nhs 10.2.0.1
ip nhrp nhs 10.2.0.10
ip nhrp registration timeout 3
ip tcp adjust-mss 1360
delay 200
if-state nhrp
keepalive 3 2
tunnel source 172.16.4.2
tunnel mode gre multipoint
tunnel key 200000
tunnel protection ipsec profile cisco-ipsec-ikev2
!
interface FastEthernet0/0
ip address 194.135.136.116 255.255.255.0
ip nat outside
duplex full
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet1/0
ip address 10.4.3.1 255.255.255.0
ip nat inside
ip pim sparse-mode
ip tcp adjust-mss 1360
speed auto
duplex auto
service-policy input MARKING_UP
service-policy output GLOBAL_DOWN
!
interface FastEthernet1/1
ip address 172.16.4.2 255.255.255.0
speed auto
duplex auto
service-policy input MARKING_DOWN
service-policy output GLOBAL_UP
!
interface FastEthernet2/0
no ip address
duplex full
!
!
router eigrp 100
network 10.0.0.0
passive-interface FastEthernet0/0
passive-interface FastEthernet1/1
!
ip nat inside source list 1 interface FastEthernet0/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip pim rp-address 10.2.0.1
ip route 0.0.0.0 0.0.0.0 194.135.136.115
ip route 172.16.0.0 255.255.0.0 172.16.4.1
```

```
!
logging facility local0
logging host 10.3.1.2
logging host 10.3.2.2
access-list 1 permit 10.4.3.0 0.0.0.255
access-list 101 permit udp any any range 16000 32000
access-list 101 remark permit_realtime
access-list 102 permit tcp any any range 1520 1560
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 remark permit_interactive
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end
```

R1

```
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
ip name-server 8.8.8.8
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
!
!
ip tcp synwait-time 5
!
!
```

```
interface FastEthernet0/0
ip address dhcp
ip nat outside
duplex full
!
interface FastEthernet1/0
ip address 194.56.21.2 255.255.255.0
ip nat inside
speed auto
duplex auto
!
interface FastEthernet1/1
ip address 81.134.91.1 255.255.255.0
ip nat inside
speed auto
duplex auto
!
interface FastEthernet2/0
ip address 91.35.197.13 255.255.255.0
ip nat inside
duplex full
!
interface FastEthernet3/0
ip address 194.135.136.115 255.255.255.0
ip nat inside
speed auto
duplex auto
!
interface FastEthernet3/1
ip address 194.56.22.2 255.255.255.0
ip nat inside
speed auto
duplex auto
!
ip nat inside source list 1 interface FastEthernet0/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 10.0.0.0 255.0.0.0 Null0
!
access-list 1 permit any
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
```

```
stopbits 1
line vty 0 4
login
!
!
end
```

R2

```
hostname R2
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
!
ip tcp synwait-time 5
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex full
!
interface FastEthernet1/0
ip address 172.16.5.2 255.255.255.0
speed auto
duplex auto
!
interface FastEthernet1/1
ip address 172.16.2.1 255.255.255.0
speed auto
duplex auto
!
interface FastEthernet2/0
ip address 172.16.3.1 255.255.255.0
duplex full
!
interface FastEthernet3/0
ip address 172.16.4.1 255.255.255.0
speed auto
duplex auto
!
interface FastEthernet3/1
```

```
ip address 172.16.1.2 255.255.255.0
speed auto
duplex auto
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
End
```

COMANDOS CISCO IOS PARA COMPROBAR LA CONECTIVIDAD

HUB1

ROUTING:

Tabla de vecinos:

```
HUB1#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
8   10.1.0.5                Tu1                      11 00:45:32   72   432  0  34
7   10.1.0.3                Tu1                      10 00:46:07  122   732  0  33
6   10.1.0.2                Tu1                      13 00:46:19  101   606  0  41
5   10.2.0.4                Tu10                     13 00:46:26   94   564  0  42
4   10.1.0.4                Tu1                      11 00:46:35  165   990  0  43
3   10.2.0.3                Tu10                     11 00:46:49  372  2232  0  31
2   10.2.0.5                Tu10                     12 00:46:55  244  1464  0  28
1   10.2.0.2                Tu10                     10 00:47:08   91   546  0  32
0   10.3.1.5                Fa0/0                    13 00:47:09   87   522  0  51
HUB1#
```

Tabla de topología:

```
HUB1#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(194.56.21.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Rep
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 309760
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.1.0.0/24, 1 successors, FD is 332800
        via Connected, Tunnel1
P 172.16.0.0/16, 0 successors, FD is Infinity
        via 10.1.0.2 (335360/28160), Tunnel1
        via 10.2.0.2 (309760/28160), Tunnel10
P 10.4.2.0/24, 1 successors, FD is 309760
        via 10.2.0.4 (309760/28160), Tunnel10
        via 10.1.0.4 (335360/28160), Tunnel1
P 10.4.1.0/24, 1 successors, FD is 309760
        via 10.2.0.3 (309760/28160), Tunnel10
        via 10.1.0.3 (335360/28160), Tunnel1
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.1.0.2 (335360/28160), Tunnel1
        via 10.2.0.2 (309760/28160), Tunnel10
P 10.4.3.0/24, 1 successors, FD is 309760
        via 10.2.0.5 (309760/28160), Tunnel10
        via 10.1.0.5 (335360/28160), Tunnel1
P 10.3.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

Tabla NHRP:

```
HUB1#sh ip nhrp
10.1.0.2/32 via 10.1.0.2
   Tunnel1 created 00:48:17, expire 00:00:14
   Type: dynamic, Flags: unique registered used
   NBMA address: 194.56.22.1
10.1.0.3/32 via 10.1.0.3
   Tunnel1 created 00:48:08, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 81.134.91.2
10.1.0.4/32 via 10.1.0.4
   Tunnel1 created 00:48:37, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 91.35.197.14
10.1.0.5/32 via 10.1.0.5
   Tunnel1 created 00:47:33, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 194.135.136.116
10.2.0.2/32 via 10.2.0.2
   Tunnel10 created 00:50:09, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.1.1
10.2.0.3/32 via 10.2.0.3
   Tunnel10 created 00:48:52, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.2.2
10.2.0.4/32 via 10.2.0.4
   Tunnel10 created 00:48:24, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.3.2
10.2.0.5/32 via 10.2.0.5
   Tunnel10 created 00:48:56, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.4.2
```

CALIDAD DE SERVICIO:

```
HUB1#sh policy-map int f0/0
 FastEthernet0/0

  Service-policy input: MARKING_UP

    Class-map: RealTime (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 101
      QoS Set
        dscp ef
          Packets marked 0

    Class-map: Interactive (match-all)
      4328 packets, 6085122 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 102
      QoS Set
        dscp af31
          Packets marked 4328

    Class-map: class-default (match-any)
      2709 packets, 180562 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_DOWN

    Class-map: qos1 (match-any)
      2616 packets, 175295 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: qos-group 1
        2616 packets, 175295 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 2616/175295
      shape (average) cir 101693000, bc 406772, be 406772
      target shape rate 101693000

      Service-policy : child

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: RealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 101
          Priority: 30% (9999 kbps), burst bytes 249950, b/w exceed drops: 0

          QoS Set
            dscp ef
              Packets marked 0


        Class-map: Interactive (match-all)
```

```
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 102
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (6666 kbps)
          QoS Set
            dscp af31
              Packets marked 0

        Class-map: class-default (match-any)
          2616 packets, 175295 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 2616/175295
          bandwidth 10% (3333 kbps)
          QoS Set
            dscp default
              Packets marked 2616

    Class-map: qos2 (match-any)
      45 packets, 3508 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: qos-group 2
        45 packets, 3508 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 45/3508
      shape (average) cir 101693000, bc 406772, be 406772
      target shape rate 101693000

      Service-policy : child

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: RealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 101
          Priority: 30% (9999 kbps), burst bytes 249950, b/w exceed drops: 0

          QoS Set
            dscp ef
              Packets marked 0

        Class-map: Interactive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 102
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
```

```
          (pkts output/bytes output) 0/0
          bandwidth 20% (6666 kbps)
          QoS Set
            dscp af31
              Packets marked 0

        Class-map: class-default (match-any)
          45 packets, 3508 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 45/3508
          bandwidth 10% (3333 kbps)
          QoS Set
            dscp default
              Packets marked 45

    Class-map: class-default (match-any)
      3365 packets, 245858 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
                           (pkts output/bytes output) 3365/245858
```

```
HUB1#sh policy-map int f1/1
 FastEthernet1/1

  Service-policy input: MARKING_DOWN

    Class-map: ISP1.source (match-any)
      12893 packets, 2180670 bytes
      5 minute offered rate 3000 bps, drop rate 0000 bps
      Match: source-address mac CA08.117E.001C
        12893 packets, 2180670 bytes
        5 minute rate 3000 bps
      QoS Set
        qos-group 1
          Packets marked 12893

    Class-map: ISP2.source (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: source-address mac CA03.1122.001C
        0 packets, 0 bytes
        5 minute rate 0 bps
      QoS Set
        qos-group 2
          Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_UP

    Class-map: ISP1.destination (match-any)
```

```
      14956 packets, 8371732 bytes
      5 minute offered rate 3000 bps, drop rate 0000 bps
      Match: destination-address mac CA08.117E.001C
        14956 packets, 8371732 bytes
        5 minute rate 3000 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 14956/8371732
      shape (average) cir 101693000, bc 406772, be 406772
      target shape rate 101693000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: pRealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (9999 kbps), burst bytes 249950, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          4315 packets, 6411714 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp af31 (26)
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 4315/6411714
          bandwidth 20% (6666 kbps)

        Class-map: class-default (match-any)
          10641 packets, 1960018 bytes
          5 minute offered rate 3000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 10641/1960018
          bandwidth 10% (3333 kbps)

    Class-map: ISP2.destination (match-any)
      1 packets, 60 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: destination-address mac CA03.1122.001C
        1 packets, 60 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 1/60
      shape (average) cir 101693000, bc 406772, be 406772
      target shape rate 101693000

      Service-policy : pchild

        queue stats for all priority classes:
```

```
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0

      Class-map: pRealTime (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match:  dscp ef (46)
        Priority: 30% (9999 kbps), burst bytes 249950, b/w exceed drops: 0


      Class-map: pInteractive (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match:  dscp af31 (26)
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 20% (6666 kbps)

      Class-map: class-default (match-any)
        1 packets, 60 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 1/60
        bandwidth 10% (3333 kbps)

    Class-map: class-default (match-any)
      532 packets, 57620 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
                           (pkts output/bytes output) 532/57620
```

MULTICAST:

Tabla vecinos Multicast:

```
HUB1#sh ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.1.1.1), 01:03:00/00:02:48, RP 10.2.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel10, 10.2.0.3, Forward/Sparse, 00:16:29/00:02:32
    Tunnel10, 10.2.0.4, Forward/Sparse, 00:36:18/00:02:48
    Tunnel10, 10.2.0.5, Forward/Sparse, 00:57:40/00:02:48

(10.3.1.2, 225.1.1.1), 00:57:23/00:01:46, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel10, 10.2.0.3, Forward/Sparse, 00:16:29/00:02:43
    Tunnel10, 10.2.0.4, Forward/Sparse, 00:36:18/00:03:10
    Tunnel10, 10.2.0.5, Forward/Sparse, 00:57:23/00:03:17

(*, 224.0.1.40), 02:59:56/00:03:22, RP 10.2.0.1, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel10, 10.2.0.3, Forward/Sparse, 02:58:07/00:03:22
    Tunnel10, 10.2.0.4, Forward/Sparse, 02:58:01/00:02:42
    Tunnel10, 10.2.0.5, Forward/Sparse, 02:58:01/00:02:50
    FastEthernet0/0, Forward/Sparse, 02:59:55/00:02:12

HUB1#
```

RP configurados:

```
HUB1#sh ip pim rp
Group: 225.1.1.1, RP: 10.2.0.1, next RP-reachable in 00:00:48
Group: 224.0.1.40, RP: 10.2.0.1, next RP-reachable in 00:01:01
```

Tabla de vecinos de PIM

```
HUB1#sh ip pim nei
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.2.0.4          Tunnel10                 03:06:35/00:01:24 v2    1 / S P G
10.2.0.5          Tunnel10                 03:06:35/00:01:17 v2    1 / DR S P G
10.2.0.3          Tunnel10                 03:06:41/00:01:26 v2    1 / S P G
HUB1#
```

Sesión DMVPN

```
HUB1#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
      N - NATed, L - Local, X - No Socket
      # Ent --> Number of NHRP entries with same NBMA peer
      NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
      UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel1 is up/up, Addr. is 10.1.0.1, VRF ""
   Tunnel Src./Dest. addr: 194.56.21.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds
Type:Hub, Total NBMA Peers (v4/v6): 4


# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.22.1           10.1.0.2    UP 00:03:12    D        10.1.0.2/32
    1 81.134.91.2           10.1.0.3    UP 00:03:09    D        10.1.0.3/32
    1 91.35.197.14          10.1.0.4    UP 00:03:05    D        10.1.0.4/32
    1 194.135.136.116       10.1.0.5    UP 00:02:59    D        10.1.0.5/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.1, VRF ""
   Tunnel Src./Dest. addr: 172.16.5.1/MGRE, Tunnel VRF ""
```

```
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds
Type:Hub, Total NBMA Peers (v4/v6): 4

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.1.1            10.2.0.2    UP 00:03:08    D        10.2.0.2/32
    1 172.16.2.2            10.2.0.3    UP 00:03:00    D        10.2.0.3/32
    1 172.16.3.2            10.2.0.4    UP 00:03:01    D        10.2.0.4/32
    1 172.16.4.2            10.2.0.5    UP 00:02:49    D        10.2.0.5/32


Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
Session: [0x65B1259C]
  IKEv2 SA: local 194.56.21.1/500 remote 194.56.22.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:56:47
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 194.56.22.1
  IPSEC FLOW: permit 47 host 194.56.21.1 host 194.56.22.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 119 drop 0 life (KB/Sec) 4253161/3407
        Outbound: #pkts enc'ed 115 drop 0 life (KB/Sec) 4253161/3407
   Outbound SPI : 0x4BB539D8, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B123AC]
  IKEv2 SA: local 194.56.21.1/500 remote 81.134.91.2/500 Active
          Capabilities:(none) connid:3 lifetime:23:56:49
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 81.134.91.2
  IPSEC FLOW: permit 47 host 194.56.21.1 host 81.134.91.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 138 drop 0 life (KB/Sec) 4216003/3409
        Outbound: #pkts enc'ed 130 drop 0 life (KB/Sec) 4216003/3409
   Outbound SPI : 0xEFC4D743, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B124A4]
  IKEv2 SA: local 194.56.21.1/500 remote 91.35.197.14/500 Active
          Capabilities:(none) connid:4 lifetime:23:56:53
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 91.35.197.14
  IPSEC FLOW: permit 47 host 194.56.21.1 host 91.35.197.14
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 145 drop 0 life (KB/Sec) 4351251/3418
        Outbound: #pkts enc'ed 131 drop 0 life (KB/Sec) 4351254/3418
   Outbound SPI : 0xEC7ECEBB, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B122B4]
  IKEv2 SA: local 194.56.21.1/500 remote 194.135.136.116/500 Active
          Capabilities:(none) connid:7 lifetime:23:56:58
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 194.135.136.116
  IPSEC FLOW: permit 47 host 194.56.21.1 host 194.135.136.116
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 124 drop 0 life (KB/Sec) 4348022/3431
        Outbound: #pkts enc'ed 117 drop 0 life (KB/Sec) 4348021/3431
   Outbound SPI : 0xE948B958, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B12694]
```

```
  IKEv2 SA: local 172.16.5.1/500 remote 172.16.1.1/500 Active
          Capabilities:(none) connid:2 lifetime:23:56:49
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 172.16.1.1
  IPSEC FLOW: permit 47 host 172.16.5.1 host 172.16.1.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 149 drop 0 life (KB/Sec) 4309457/3409
        Outbound: #pkts enc'ed 168 drop 0 life (KB/Sec) 4309453/3409
   Outbound SPI : 0xC1F1F6BD, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B1278C]
  IKEv2 SA: local 172.16.5.1/500 remote 172.16.2.2/500 Active
          Capabilities:(none) connid:5 lifetime:23:56:57
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 172.16.2.2
  IPSEC FLOW: permit 47 host 172.16.5.1 host 172.16.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 136 drop 0 life (KB/Sec) 4306841/3417
        Outbound: #pkts enc'ed 132 drop 0 life (KB/Sec) 4306841/3417
   Outbound SPI : 0x983C917E, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B1297C]
  IKEv2 SA: local 172.16.5.1/500 remote 172.16.3.2/500 Active
          Capabilities:(none) connid:6 lifetime:23:56:57
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 172.16.3.2
  IPSEC FLOW: permit 47 host 172.16.5.1 host 172.16.3.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 131 drop 0 life (KB/Sec) 4371499/3417
        Outbound: #pkts enc'ed 130 drop 0 life (KB/Sec) 4371499/3417
   Outbound SPI : 0xACABE82E, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B12884]
  IKEv2 SA: local 172.16.5.1/500 remote 172.16.4.2/500 Active
          Capabilities:(none) connid:8 lifetime:23:57:04
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 172.16.4.2
  IPSEC FLOW: permit 47 host 172.16.5.1 host 172.16.4.2
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 116 drop 0 life (KB/Sec) 4162872/3431
        Outbound: #pkts enc'ed 112 drop 0 life (KB/Sec) 4162872/3431
   Outbound SPI : 0x4FFA1B25, transform : esp-aes esp-sha-hmac
    Socket State: Open

Pending DMVPN Sessions:
```

HUB1-BACKUP

ROUTING:

Tabla de vecinos:

```
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
8   10.1.0.3                Tu1                      12 00:51:39   97   582  0  35
7   10.1.0.5                Tu1                      10 00:51:57  108   648  0  30
6   10.2.0.3                Tu10                     13 00:52:05   96   576  0  31
5   10.1.0.2                Tu1                      11 00:52:28  115   690  0  38
4   10.1.0.4                Tu1                      12 00:52:31  198  1188  0  43
3   10.2.0.5                Tu10                     11 00:52:44  171  1026  0  28
2   10.2.0.4                Tu10                     12 00:52:53  269  1614  0  42
1   10.2.0.2                Tu10                     13 00:52:59  128   768  0  34
0   10.3.1.1                Fa0/0                    12 00:53:02  106   636  0  47
HUB1-backup#
```

Tabla de topología:

```
HUB1-backup#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(194.56.21.3)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 309760
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.1.0.0/24, 1 successors, FD is 332800
        via Connected, Tunnel1
P 172.16.0.0/16, 0 successors, FD is Infinity
        via 10.1.0.2 (335360/28160), Tunnel1
        via 10.2.0.2 (309760/28160), Tunnel10
P 10.4.2.0/24, 1 successors, FD is 309760
        via 10.2.0.4 (309760/28160), Tunnel10
        via 10.1.0.4 (335360/28160), Tunnel1
P 10.4.1.0/24, 1 successors, FD is 309760
        via 10.2.0.3 (309760/28160), Tunnel10
        via 10.1.0.3 (335360/28160), Tunnel1
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.1.0.2 (335360/28160), Tunnel1
        via 10.2.0.2 (309760/28160), Tunnel10
P 10.4.3.0/24, 1 successors, FD is 309760
        via 10.2.0.5 (309760/28160), Tunnel10
        via 10.1.0.5 (335360/28160), Tunnel1
P 10.3.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

Tabla NHRP:

```
HUB1-backup#sh ip nhrp
10.1.0.2/32 via 10.1.0.2
   Tunnel1 created 00:54:03, expire 00:00:10
   Type: dynamic, Flags: unique registered used
   NBMA address: 194.56.22.1
10.1.0.3/32 via 10.1.0.3
   Tunnel1 created 00:53:12, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 81.134.91.2
10.1.0.4/32 via 10.1.0.4
   Tunnel1 created 00:54:05, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 91.35.197.14
10.1.0.5/32 via 10.1.0.5
   Tunnel1 created 00:53:33, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 194.135.136.116
10.2.0.2/32 via 10.2.0.2
   Tunnel10 created 00:56:44, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.1.1
10.2.0.3/32 via 10.2.0.3
   Tunnel10 created 00:53:38, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.2.2
10.2.0.4/32 via 10.2.0.4
   Tunnel10 created 00:54:30, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.3.2
10.2.0.5/32 via 10.2.0.5
   Tunnel10 created 00:54:19, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.4.2
```

Sesión DMVPN:

```
HUB1-backup#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
      N - NATed, L - Local, X - No Socket
      # Ent --> Number of NHRP entries with same NBMA peer
```

```
      NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
      UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel1 is up/up, Addr. is 10.1.0.10, VRF ""
   Tunnel Src./Dest. addr: 194.56.21.3/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
Type:Hub, Total NBMA Peers (v4/v6): 4

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.22.1           10.1.0.2    UP 00:05:14    D        10.1.0.2/32
    1 81.134.91.2           10.1.0.3    UP 00:05:12    D        10.1.0.3/32
    1 91.35.197.14          10.1.0.4    UP 00:05:11    D        10.1.0.4/32
    1 194.135.136.116       10.1.0.5    UP 00:05:06    D        10.1.0.5/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.10, VRF ""
   Tunnel Src./Dest. addr: 172.16.5.3/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
Type:Hub, Total NBMA Peers (v4/v6): 4

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.1.1            10.2.0.2    UP 00:05:09    D        10.2.0.2/32
    1 172.16.2.2            10.2.0.3    UP 00:05:09    D        10.2.0.3/32
    1 172.16.3.2            10.2.0.4    UP 00:05:04    D        10.2.0.4/32
    1 172.16.4.2            10.2.0.5    UP 00:04:59    D        10.2.0.5/32


Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
Session: [0x65B151E4]
  IKEv2 SA: local 194.56.21.3/500 remote 194.56.22.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:54:43
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 194.56.22.1
  IPSEC FLOW: permit 47 host 194.56.21.3 host 194.56.22.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 160 drop 0 life (KB/Sec) 4360987/3283
        Outbound: #pkts enc'ed 165 drop 0 life (KB/Sec) 4360986/3283
   Outbound SPI : 0xEFF86F39, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B14EFC]
  IKEv2 SA: local 194.56.21.3/500 remote 81.134.91.2/500 Active
          Capabilities:(none) connid:3 lifetime:23:54:46
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 81.134.91.2
  IPSEC FLOW: permit 47 host 194.56.21.3 host 81.134.91.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 197 drop 0 life (KB/Sec) 4210977/3286
        Outbound: #pkts enc'ed 208 drop 0 life (KB/Sec) 4210974/3286
   Outbound SPI : 0xBC656EB8, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B14FF4]
  IKEv2 SA: local 194.56.21.3/500 remote 91.35.197.14/500 Active
          Capabilities:(none) connid:4 lifetime:23:54:46
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 91.35.197.14
  IPSEC FLOW: permit 47 host 194.56.21.3 host 91.35.197.14
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 198 drop 0 life (KB/Sec) 4352759/3286
```

```
        Outbound: #pkts enc'ed 204 drop 0 life (KB/Sec) 4352757/3286
   Outbound SPI : 0x90CB9E28, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B150EC]
  IKEv2 SA: local 194.56.21.3/500 remote 194.135.136.116/500 Active
          Capabilities:(none) connid:7 lifetime:23:54:52
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 194.135.136.116
  IPSEC FLOW: permit 47 host 194.56.21.3 host 194.135.136.116
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 182 drop 0 life (KB/Sec) 4340677/3292
        Outbound: #pkts enc'ed 195 drop 0 life (KB/Sec) 4340674/3292
   Outbound SPI : 0x 36E6822, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B14D0C]
  IKEv2 SA: local 172.16.5.3/500 remote 172.16.1.1/500 Active
          Capabilities:(none) connid:2 lifetime:23:54:48
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.1.1
  IPSEC FLOW: permit 47 host 172.16.5.3 host 172.16.1.1
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 197 drop 0 life (KB/Sec) 4356593/3305
        Outbound: #pkts enc'ed 199 drop 0 life (KB/Sec) 4356591/3305
   Outbound SPI : 0x3A6DAFCE, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B14C14]
  IKEv2 SA: local 172.16.5.3/500 remote 172.16.2.2/500 Active
          Capabilities:(none) connid:5 lifetime:23:54:48
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.2.2
  IPSEC FLOW: permit 47 host 172.16.5.3 host 172.16.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 205 drop 0 life (KB/Sec) 4245460/3288
        Outbound: #pkts enc'ed 193 drop 0 life (KB/Sec) 4245460/3288
   Outbound SPI : 0x90C8AF91, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B14B1C]
  IKEv2 SA: local 172.16.5.3/500 remote 172.16.3.2/500 Active
          Capabilities:(none) connid:6 lifetime:23:54:52
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.3.2
  IPSEC FLOW: permit 47 host 172.16.5.3 host 172.16.3.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 197 drop 0 life (KB/Sec) 4293368/3292
        Outbound: #pkts enc'ed 191 drop 0 life (KB/Sec) 4293367/3292
   Outbound SPI : 0x25FBC51F, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B14E04]
  IKEv2 SA: local 172.16.5.3/500 remote 172.16.4.2/500 Active
          Capabilities:(none) connid:8 lifetime:23:54:58
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.4.2
  IPSEC FLOW: permit 47 host 172.16.5.3 host 172.16.4.2
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 195 drop 0 life (KB/Sec) 4298957/3313
        Outbound: #pkts enc'ed 179 drop 0 life (KB/Sec) 4298957/3313
   Outbound SPI : 0x24DA945A, transform : esp-aes esp-sha-hmac
    Socket State: Open

                                  Pending DMVPN Sessions:
```

HUB2

ROUTING:

Tabla de vecinos:

```
HUB2#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
9   10.1.0.5                Tu1                      10 00:00:29   73   438  0  25
8   10.2.0.5                Tu10                     10 00:00:29   24   144  0  26
7   10.2.0.4                Tu10                     12 00:00:36  267  1602  0  29
6   10.2.0.3                Tu10                     14 00:00:40  164   984  0  30
5   10.1.0.4                Tu1                      13 00:00:45  236  1416  0  28
4   10.1.0.3                Tu1                      13 00:00:45  182  1092  0  31
3   10.2.0.1                Tu10                     12 00:00:51  232  1392  0  53
2   10.2.0.10               Tu10                     12 00:00:56  294  1764  0  48
1   10.1.0.1                Tu1                      11 00:00:56  311  1866  0  56
0   10.1.0.10               Tu1                      13 00:00:58  247  1482  0  47
HUB2#
```

Tabla de topología:

```
HUB2#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(194.56.22.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R -
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
P 10.1.0.0/24, 1 successors, FD is 345600
        via Connected, Tunnel1
        via 10.2.0.5 (396800/332800), Tunnel10
        via 10.1.0.5 (422400/332800), Tunnel1
        via 10.2.0.4 (396800/332800), Tunnel10
        via 10.2.0.3 (396800/332800), Tunnel10
        via 10.1.0.3 (422400/332800), Tunnel1
        via 10.1.0.4 (422400/332800), Tunnel1
        via 10.2.0.1 (396800/332800), Tunnel10
        via 10.2.0.10 (401664/337664), Tunnel10
        via 10.1.0.1 (422400/332800), Tunnel1
        via 10.1.0.10 (427264/337664), Tunnel1
P 172.16.0.0/16, 1 successors, FD is 28160
        via Rstatic (28160/0)
P 10.4.2.0/24, 1 successors, FD is 322560
        via 10.2.0.4 (322560/28160), Tunnel10
        via 10.1.0.4 (348160/28160), Tunnel1
       10.2.0.4 via 10.2.0.1 (373760/309760), Tunnel10
        via 10.1.0.10 (404224/314624), Tunnel1
        via 10.1.0.1 (399360/309760), Tunnel1
       10.2.0.4 via 10.2.0.10 (378624/314624), Tunnel10
P 10.4.1.0/24, 1 successors, FD is 322560
        via 10.2.0.3 (322560/28160), Tunnel10
        via 10.1.0.3 (348160/28160), Tunnel1
       10.2.0.3 via 10.2.0.1 (373760/309760), Tunnel10
       10.2.0.3 via 10.2.0.10 (378624/314624), Tunnel10
        via 10.1.0.1 (399360/309760), Tunnel1
        via 10.1.0.10 (404224/314624), Tunnel1
P 10.2.0.0/24, 1 successors, FD is 320000
        via Connected, Tunnel10
        via 10.2.0.5 (371200/307200), Tunnel10
        via 10.1.0.5 (396800/307200), Tunnel1
        via 10.2.0.4 (371200/307200), Tunnel10
        via 10.2.0.3 (371200/307200), Tunnel10
        via 10.1.0.4 (396800/307200), Tunnel1
        via 10.1.0.3 (396800/307200), Tunnel1
        via 10.2.0.1 (371200/307200), Tunnel10
        via 10.2.0.10 (376064/312064), Tunnel10
        via 10.1.0.1 (396800/307200), Tunnel1
        via 10.1.0.10 (401664/312064), Tunnel1
P 0.0.0.0/0, 1 successors, FD is 28160
        via Rstatic (28160/0)
P 10.4.3.0/24, 1 successors, FD is 322560
        via 10.2.0.5 (322560/28160), Tunnel10
        via 10.1.0.5 (348160/28160), Tunnel1
       10.2.0.5 via 10.2.0.1 (373760/309760), Tunnel10
       10.2.0.5 via 10.2.0.10 (378624/314624), Tunnel10
        via 10.1.0.1 (399360/309760), Tunnel1
        via 10.1.0.10 (404224/314624), Tunnel1
P 10.3.1.0/24, 1 successors, FD is 322560
        via 10.2.0.1 (322560/28160), Tunnel10
       10.2.0.1 via 10.2.0.5 (373760/309760), Tunnel10
        via 10.1.0.5 (399360/309760), Tunnel1
       10.2.0.1 via 10.2.0.4 (373760/309760), Tunnel10
       10.2.0.1 via 10.2.0.3 (373760/309760), Tunnel10
        via 10.1.0.3 (399360/309760), Tunnel1
        via 10.1.0.4 (399360/309760), Tunnel1
        via 10.2.0.10 (327424/33024), Tunnel10
        via 10.1.0.1 (348160/28160), Tunnel1
        via 10.1.0.10 (353024/33024), Tunnel1
```

Tabla NHRP:

```
HUB2#sh ip nhrp
10.1.0.1/32 via 10.1.0.1
   Tunnel1 created 00:05:57, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.1
10.1.0.3/32 via 10.1.0.3
   Tunnel1 created 00:05:15, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 81.134.91.2
10.1.0.4/32 via 10.1.0.4
   Tunnel1 created 00:05:15, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 91.35.197.14
10.1.0.5/32 via 10.1.0.5
   Tunnel1 created 00:04:59, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 194.135.136.116
10.1.0.10/32 via 10.1.0.10
   Tunnel1 created 00:05:57, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.3
10.2.0.1/32 via 10.2.0.1
   Tunnel10 created 00:05:56, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.1
10.2.0.3/32 via 10.2.0.3
   Tunnel10 created 00:05:10, expire 00:04:59
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.2.2
10.2.0.4/32 via 10.2.0.4
   Tunnel10 created 00:05:05, expire 00:04:57
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.3.2
10.2.0.5/32 via 10.2.0.5
   Tunnel10 created 00:04:59, expire 00:04:58
   Type: dynamic, Flags: unique registered used
   NBMA address: 172.16.4.2
10.2.0.10/32 via 10.2.0.10
   Tunnel10 created 00:05:56, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.3
```

Sesión DMVPN:

```
HUB2#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
      N - NATed, L - Local, X - No Socket
      # Ent --> Number of NHRP entries with same NBMA peer
      NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
      UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel1 is up/up, Addr. is 10.1.0.2, VRF ""
   Tunnel Src./Dest. addr: 194.56.22.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled

IPv4 NHS:
10.1.0.1  RE priority = 0 cluster = 0
10.1.0.10  RE priority = 0 cluster = 0
Type:Hub/Spoke, Total NBMA Peers (v4/v6): 5

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.21.1           10.1.0.1    UP 00:06:54    S        10.1.0.1/32
    1 81.134.91.2           10.1.0.3    UP 00:06:46    D        10.1.0.3/32
    1 91.35.197.14          10.1.0.4    UP 00:06:40    D        10.1.0.4/32
    1 194.135.136.116       10.1.0.5    UP 00:06:32    D        10.1.0.5/32
    1 194.56.21.3          10.1.0.10    UP 00:06:52    S       10.1.0.10/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.2, VRF ""
   Tunnel Src./Dest. addr: 172.16.1.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.2.0.1  RE priority = 0 cluster = 0
10.2.0.10  RE priority = 0 cluster = 0
Type:Hub/Spoke, Total NBMA Peers (v4/v6): 5

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.5.1            10.2.0.1    UP 00:06:51    S        10.2.0.1/32
    1 172.16.2.2            10.2.0.3    UP 00:06:38    D        10.2.0.3/32
    1 172.16.3.2            10.2.0.4    UP 00:06:35    D        10.2.0.4/32
    1 172.16.4.2            10.2.0.5    UP 00:06:32    D        10.2.0.5/32
    1 172.16.5.3           10.2.0.10    UP 00:06:47    S       10.2.0.10/32
```

```
Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
Session: [0x65B12884]
  IKEv2 SA: local 194.56.22.1/500 remote 194.56.21.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:53:05
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 194.56.21.1
  IPSEC FLOW: permit 47 host 194.56.22.1 host 194.56.21.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 206 drop 0 life (KB/Sec) 4322846/3185
        Outbound: #pkts enc'ed 211 drop 0 life (KB/Sec) 4322847/3185
   Outbound SPI : 0xC77C47B3, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B1259C]
  IKEv2 SA: local 194.56.22.1/500 remote 81.134.91.2/500 Active
          Capabilities:(none) connid:5 lifetime:23:53:13
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 81.134.91.2
  IPSEC FLOW: permit 47 host 194.56.22.1 host 81.134.91.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 264 drop 0 life (KB/Sec) 4284119/3193
        Outbound: #pkts enc'ed 259 drop 0 life (KB/Sec) 4284117/3193
   Outbound SPI : 0x2997D885, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B12694]
  IKEv2 SA: local 194.56.22.1/500 remote 91.35.197.14/500 Active
          Capabilities:(none) connid:6 lifetime:23:53:18
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 91.35.197.14
  IPSEC FLOW: permit 47 host 194.56.22.1 host 91.35.197.14
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 240 drop 0 life (KB/Sec) 4324341/3198
        Outbound: #pkts enc'ed 243 drop 0 life (KB/Sec) 4324338/3198
   Outbound SPI : 0x7ECCFA0E, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B1278C]
  IKEv2 SA: local 194.56.22.1/500 remote 194.135.136.116/500 Active
          Capabilities:(none) connid:10 lifetime:23:53:27
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 194.135.136.116
  IPSEC FLOW: permit 47 host 194.56.22.1 host 194.135.136.116
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 217 drop 0 life (KB/Sec) 4213969/3210
        Outbound: #pkts enc'ed 220 drop 0 life (KB/Sec) 4213967/3210
   Outbound SPI : 0x7132F7D4, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B1297C]
  IKEv2 SA: local 194.56.22.1/500 remote 194.56.21.3/500 Active
          Capabilities:(none) connid:2 lifetime:23:53:07
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),  Phase1_id: 194.56.21.3
  IPSEC FLOW: permit 47 host 194.56.22.1 host 194.56.21.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 207 drop 0 life (KB/Sec) 4299123/3186
        Outbound: #pkts enc'ed 203 drop 0 life (KB/Sec) 4299125/3186
   Outbound SPI : 0x859CCA83, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
```

```
Session: [0x65B122B4]
  IKEv2 SA: local 172.16.1.1/500 remote 172.16.5.1/500 Active
          Capabilities:(none) connid:3 lifetime:23:53:08
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.5.1
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.5.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 299 drop 0 life (KB/Sec) 4303824/3188
        Outbound: #pkts enc'ed 268 drop 0 life (KB/Sec) 4303831/3188
   Outbound SPI : 0xE9685985, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B124A4]
  IKEv2 SA: local 172.16.1.1/500 remote 172.16.2.2/500 Active
          Capabilities:(none) connid:8 lifetime:23:53:20
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.2.2
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 255 drop 0 life (KB/Sec) 4166946/3200
        Outbound: #pkts enc'ed 237 drop 0 life (KB/Sec) 4166946/3200
   Outbound SPI : 0x87707FE2, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B123AC]
  IKEv2 SA: local 172.16.1.1/500 remote 172.16.3.2/500 Active
          Capabilities:(none) connid:9 lifetime:23:53:22
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.3.2
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.3.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 251 drop 0 life (KB/Sec) 4300611/3202
        Outbound: #pkts enc'ed 233 drop 0 life (KB/Sec) 4300612/3202
   Outbound SPI : 0x7796F363, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B121BC]
  IKEv2 SA: local 172.16.1.1/500 remote 172.16.4.2/500 Active
          Capabilities:(none) connid:11 lifetime:23:53:27
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.4.2
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.4.2
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 249 drop 0 life (KB/Sec) 4173652/3208
        Outbound: #pkts enc'ed 232 drop 0 life (KB/Sec) 4173652/3208
   Outbound SPI : 0x5B96A0A0, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B120C4]
  IKEv2 SA: local 172.16.1.1/500 remote 172.16.5.3/500 Active
          Capabilities:(none) connid:4 lifetime:23:53:11
  Crypto Session Status: UP-ACTIVE
  fvrf: (none),   Phase1_id: 172.16.5.3
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.5.3
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 256 drop 0 life (KB/Sec) 4189005/3203
        Outbound: #pkts enc'ed 253 drop 0 life (KB/Sec) 4189008/3203
   Outbound SPI : 0x62E5A281, transform : esp-aes esp-sha-hmac
    Socket State: Open

Pending DMVPN Sessions:
```

SPOKE1

ROUTING:

Tabla de vecinos:

```
SPOKE1#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
5   10.2.0.2                Tu10                     14 00:06:29  458  2748  0  49
4   10.2.0.1                Tu10                     11 00:06:33  466  2796  0  53
3   10.2.0.10               Tu10                     13 00:06:34  242  1452  0  48
2   10.1.0.2                Tu1                      13 00:06:36  322  1932  0  50
1   10.1.0.1                Tu1                      13 00:06:38  390  2340  0  56
0   10.1.0.10               Tu1                      11 00:06:38  352  2112  0  47
SPOKE1#
```

Tabla de topología:

```
SPOKE1#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(172.16.2.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 309760
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.1.0.0/24, 1 successors, FD is 332800
        via Connected, Tunnel1
P 172.16.0.0/16, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.2.0/24, 1 successors, FD is 360960
       10.2.0.4 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.4 via 10.2.0.10 (365824/314624), Tunnel10
       10.2.0.4 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.1 (386560/309760), Tunnel1
        via 10.1.0.2 (399360/322560), Tunnel1
        via 10.1.0.10 (391424/314624), Tunnel1
P 10.4.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet1/0
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.3.0/24, 1 successors, FD is 360960
       10.2.0.5 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.5 via 10.2.0.10 (365824/314624), Tunnel10
       10.2.0.5 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.10 (391424/314624), Tunnel1
        via 10.1.0.2 (399360/322560), Tunnel1
        via 10.1.0.1 (386560/309760), Tunnel1
P 10.3.1.0/24, 1 successors, FD is 309760
        via 10.2.0.1 (309760/28160), Tunnel10
        via 10.2.0.10 (314624/33024), Tunnel10
        via 10.1.0.1 (335360/28160), Tunnel1
        via 10.1.0.10 (340224/33024), Tunnel1
```

Tabla NHRP:

```
SPOKE1#sh ip nhrp
10.1.0.1/32 via 10.1.0.1
   Tunnel1 created 00:10:08, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.1
10.1.0.2/32 via 10.1.0.2
   Tunnel1 created 00:10:08, never expire
   Type: static, Flags: used
   NBMA address: 194.56.22.1
10.1.0.10/32 via 10.1.0.10
   Tunnel1 created 00:10:08, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.3
10.2.0.1/32 via 10.2.0.1
   Tunnel10 created 00:10:07, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.1
10.2.0.2/32 via 10.2.0.2
   Tunnel10 created 00:10:07, never expire
   Type: static, Flags: used
   NBMA address: 172.16.1.1
10.2.0.10/32 via 10.2.0.10
   Tunnel10 created 00:10:07, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.3
SPOKE1#
```

CALIDAD DE SERVICIO

```
SPOKE1#sh policy-map int f1/0
 FastEthernet1/0

  Service-policy input: MARKING_UP
```

```
  Class-map: RealTime (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: access-group 101
    QoS Set
      dscp ef
        Packets marked 0

  Class-map: Interactive (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: access-group 102
    QoS Set
      dscp af31
        Packets marked 0

  Class-map: class-default (match-any)
    3976 packets, 267909 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any

Service-policy output: GLOBAL_DOWN

  Class-map: qos1 (match-any)
    9967 packets, 11760465 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 1
      9967 packets, 11760465 bytes
      5 minute rate 0 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 9967/11760465
    shape (average) cir 15840000, bc 63360, be 63360
    target shape rate 15840000

    Service-policy : child

      queue stats for all priority classes:
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 3340/2703364

      Class-map: RealTime (match-all)
        3340 packets, 2703364 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 101
        Priority: 30% (4752 kbps), burst bytes 118800, b/w exceed drops: 0

        QoS Set
          dscp ef
            Packets marked 3340

      Class-map: Interactive (match-all)
        6433 packets, 9030951 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 102
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 6433/9030951
```

```
          bandwidth 20% (3168 kbps)
          QoS Set
            dscp af31
              Packets marked 6433

        Class-map: class-default (match-any)
          194 packets, 26150 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 194/26150
          bandwidth 10% (1584 kbps)
          QoS Set
            dscp default
              Packets marked 194

    Class-map: qos2 (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: qos-group 2
        0 packets, 0 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 15840000, bc 63360, be 63360
      target shape rate 15840000

      Service-policy : child

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: RealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 101
          Priority: 30% (4752 kbps), burst bytes 118800, b/w exceed drops: 0

          QoS Set
            dscp ef
              Packets marked 0

        Class-map: Interactive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 102
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (3168 kbps)
          QoS Set
            dscp af31
              Packets marked 0

        Class-map: class-default (match-any)
```

```
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 10% (1584 kbps)
          QoS Set
            dscp default
              Packets marked 0


    Class-map: class-default (match-any)
      1112 packets, 104308 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 1112/104308
```

```
SPOKE1#sh policy-map int f1/1
 FastEthernet1/1

  Service-policy input: MARKING_DOWN

    Class-map: ISP1.source (match-any)
      17686 packets, 13973784 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: source-address mac CA08.117E.001D
        17686 packets, 13973784 bytes
        5 minute rate 2000 bps
      QoS Set
        qos-group 1
          Packets marked 17686

    Class-map: ISP2.source (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: source-address mac CA03.1122.001D
        0 packets, 0 bytes
        5 minute rate 0 bps
      QoS Set
        qos-group 2
          Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_UP

    Class-map: ISP1.destination (match-any)
      12310 packets, 2110648 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: destination-address mac CA08.117E.001D
        12310 packets, 2110648 bytes
        5 minute rate 2000 bps
```

```
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 12310/2110648
      shape (average) cir 15840000, bc 63360, be 63360
      target shape rate 15840000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: pRealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (4752 kbps), burst bytes 118800, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          187 packets, 25058 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp af31 (26)
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 187/25058
          bandwidth 20% (3168 kbps)

        Class-map: class-default (match-any)
          12123 packets, 2085590 bytes
          5 minute offered rate 2000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 12123/2085590
          bandwidth 10% (1584 kbps)

    Class-map: ISP2.destination (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: destination-address mac CA03.1122.001D
        0 packets, 0 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 15840000, bc 63360, be 63360
      target shape rate 15840000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
```

```
        Class-map: pRealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (4752 kbps), burst bytes 118800, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp af31 (26)
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (3168 kbps)

        Class-map: class-default (match-any)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 10% (1584 kbps)

    Class-map: class-default (match-any)
      724 packets, 78347 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 724/78347
```

MULTICAST

Tabla vecinos Multicast:

```
SPOKE1#sh ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.1.1.1), 01:41:52/00:02:35, RP 10.2.0.1, flags: SJC
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 00:55:21/00:02:35

(*, 224.0.1.40), 03:37:27/00:02:42, RP 10.2.0.1, flags: SJCL
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 03:37:26/00:02:42
```

RP configurados:

```
SPOKE1#sh ip pim rp
 Group: 225.1.1.1, RP: 10.2.0.1, uptime 01:43:05, expires never
 Group: 224.0.1.40, RP: 10.2.0.1, uptime 03:38:40, expires never
```

Miembros del grupo *Multicast* registrados en el segmento local:

```
SPOKE1#sh ip pim nei
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.2.0.1          Tunnel10                 03:38:53/00:01:34 v2    1 / S P G
SPOKE1#
```

Sesión DMVPN:

```
SPOKE1#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
     N - NATed, L - Local, X - No Socket
     # Ent --> Number of NHRP entries with same NBMA peer
     NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
     UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel1 is up/up, Addr. is 10.1.0.3, VRF ""
   Tunnel Src./Dest. addr: 81.134.91.2/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds
```

```
IPv4 NHS:
10.1.0.2  RE priority = 0 cluster = 0
10.1.0.1  RE priority = 0 cluster = 0
10.1.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.21.1            10.1.0.1    UP 00:12:56    S        10.1.0.1/32
    1 194.56.22.1            10.1.0.2    UP 00:12:52    S        10.1.0.2/32
    1 194.56.21.3           10.1.0.10    UP 00:12:56    S       10.1.0.10/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.3, VRF ""
   Tunnel Src./Dest. addr: 172.16.2.2/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.2.0.2  RE priority = 0 cluster = 0
10.2.0.1  RE priority = 0 cluster = 0
10.2.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.5.1             10.2.0.1    UP 00:12:48    S        10.2.0.1/32
    1 172.16.1.1             10.2.0.2    UP 00:12:43    S        10.2.0.2/32
    1 172.16.5.3            10.2.0.10    UP 00:12:53    S       10.2.0.10/32


Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
Session: [0x65B1297C]
  IKEv2 SA: local 81.134.91.2/500 remote 194.56.21.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:47:02
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.1
  IPSEC FLOW: permit 47 host 81.134.91.2 host 194.56.21.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 451 drop 0 life (KB/Sec) 4245767/2822
        Outbound: #pkts enc'ed 458 drop 0 life (KB/Sec) 4245769/2822
   Outbound SPI : 0xA1A41749, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B1278C]
  IKEv2 SA: local 81.134.91.2/500 remote 194.56.22.1/500 Active
          Capabilities:(none) connid:3 lifetime:23:47:07
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.22.1
  IPSEC FLOW: permit 47 host 81.134.91.2 host 194.56.22.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 459 drop 0 life (KB/Sec) 4376696/2827
        Outbound: #pkts enc'ed 464 drop 0 life (KB/Sec) 4376699/2827
   Outbound SPI : 0x14A87BF1, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
```

```
Session: [0x65B12884]
  IKEv2 SA: local 81.134.91.2/500 remote 194.56.21.3/500 Active
          Capabilities:(none) connid:2 lifetime:23:47:03
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.3
  IPSEC FLOW: permit 47 host 81.134.91.2 host 194.56.21.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 464 drop 0 life (KB/Sec) 4339708/2823
        Outbound: #pkts enc'ed 454 drop 0 life (KB/Sec) 4339713/2823
   Outbound SPI : 0xC4354036, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B1259C]
  IKEv2 SA: local 172.16.2.2/500 remote 172.16.5.1/500 Active
          Capabilities:(none) connid:5 lifetime:23:47:10
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.1
  IPSEC FLOW: permit 47 host 172.16.2.2 host 172.16.5.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 488 drop 0 life (KB/Sec) 4272249/2830
        Outbound: #pkts enc'ed 487 drop 0 life (KB/Sec) 4272252/2830
   Outbound SPI : 0x201B07E9, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B12694]
  IKEv2 SA: local 172.16.2.2/500 remote 172.16.1.1/500 Active
          Capabilities:(none) connid:4 lifetime:23:47:15
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.1.1
  IPSEC FLOW: permit 47 host 172.16.2.2 host 172.16.1.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 437 drop 0 life (KB/Sec) 4336308/2835
        Outbound: #pkts enc'ed 475 drop 0 life (KB/Sec) 4336307/2835
   Outbound SPI : 0x7720FB22, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B124A4]
  IKEv2 SA: local 172.16.2.2/500 remote 172.16.5.3/500 Active
          Capabilities:(none) connid:6 lifetime:23:47:06
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.3
  IPSEC FLOW: permit 47 host 172.16.2.2 host 172.16.5.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 450 drop 0 life (KB/Sec) 4277732/2826
        Outbound: #pkts enc'ed 485 drop 0 life (KB/Sec) 4277732/2826
   Outbound SPI : 0xCDD7E55A, transform : esp-aes esp-sha-hmac
    Socket State: Open

                              Pending DMVPN Sessions:
```

SPOKE2

ROUTING:

Tabla de vecinos:

```
SPOKE2#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
5   10.2.0.2                Tu10                     11 00:10:17  392  2352  0  49
4   10.2.0.1                Tu10                     10 00:10:18  575  3450  0  53
3   10.2.0.10               Tu10                     13 00:10:23  260  1560  0  48
2   10.1.0.2                Tu1                      10 00:10:24  401  2406  0  50
1   10.1.0.1                Tu1                      12 00:10:27  375  2250  0  56
0   10.1.0.10               Tu1                      13 00:10:27  351  2106  0  47
SPOKE2#
```

Tabla de topología:

```
SPOKE2#sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(172.16.3.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 309760
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.1.0.0/24, 1 successors, FD is 332800
        via Connected, Tunnel1
P 172.16.0.0/16, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.2.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet1/0
P 10.4.1.0/24, 1 successors, FD is 360960
       10.2.0.3 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.3 via 10.2.0.10 (365824/314624), Tunnel10
       10.2.0.3 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.10 (391424/314624), Tunnel1
        via 10.1.0.1 (386560/309760), Tunnel1
        via 10.1.0.2 (399360/322560), Tunnel1
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.3.0/24, 1 successors, FD is 360960
       10.2.0.5 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.5 via 10.2.0.10 (365824/314624), Tunnel10
       10.2.0.5 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.10 (391424/314624), Tunnel1
        via 10.1.0.2 (399360/322560), Tunnel1
        via 10.1.0.1 (386560/309760), Tunnel1
P 10.3.1.0/24, 1 successors, FD is 309760
        via 10.2.0.1 (309760/28160), Tunnel10
        via 10.2.0.10 (314624/33024), Tunnel10
        via 10.1.0.1 (335360/28160), Tunnel1
        via 10.1.0.10 (340224/33024), Tunnel1
```

Tabla NHRP:

```
SPOKE2#sh ip nhrp
10.1.0.1/32 via 10.1.0.1
   Tunnel1 created 00:13:07, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.1
10.1.0.2/32 via 10.1.0.2
   Tunnel1 created 00:13:07, never expire
   Type: static, Flags: used
   NBMA address: 194.56.22.1
10.1.0.10/32 via 10.1.0.10
   Tunnel1 created 00:13:07, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.3
10.2.0.1/32 via 10.2.0.1
   Tunnel10 created 00:13:06, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.1
10.2.0.2/32 via 10.2.0.2
   Tunnel10 created 00:13:06, never expire
   Type: static, Flags: used
   NBMA address: 172.16.1.1
10.2.0.10/32 via 10.2.0.10
   Tunnel10 created 00:13:06, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.3
SPOKE2#
```

CALIDAD DE SERVICIO

```
SPOKE2#sh policy-map int f1/0
 FastEthernet1/0

  Service-policy input: MARKING_UP

    Class-map: RealTime (match-all)
      955 packets, 1336282 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 101
      QoS Set
        dscp ef
          Packets marked 955
```

```
  Class-map: Interactive (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: access-group 102
    QoS Set
      dscp af31
        Packets marked 0

  Class-map: class-default (match-any)
    17 packets, 1441 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any

Service-policy output: GLOBAL_DOWN

  Class-map: qos1 (match-any)
    15 packets, 1189 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 1
      15 packets, 1189 bytes
      5 minute rate 0 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 15/1189
    shape (average) cir 6340000, bc 25360, be 25360
    target shape rate 6340000

    Service-policy : child

      queue stats for all priority classes:
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0

      Class-map: RealTime (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 101
        Priority: 30% (1902 kbps), burst bytes 47550, b/w exceed drops: 0

        QoS Set
          dscp ef
            Packets marked 0

      Class-map: Interactive (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 102
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 20% (1268 kbps)
        QoS Set
          dscp af31
            Packets marked 0

      Class-map: class-default (match-any)
        15 packets, 1189 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
```

```
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 15/1189
        bandwidth 10% (634 kbps)
        QoS Set
          dscp default
            Packets marked 15

  Class-map: qos2 (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 2
      0 packets, 0 bytes
      5 minute rate 0 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    shape (average) cir 6340000, bc 25360, be 25360
    target shape rate 6340000

    Service-policy : child

      queue stats for all priority classes:
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0

      Class-map: RealTime (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 101
        Priority: 30% (1902 kbps), burst bytes 47550, b/w exceed drops: 0

        QoS Set
          dscp ef
            Packets marked 0

      Class-map: Interactive (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 102
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 20% (1268 kbps)
        QoS Set
          dscp af31
            Packets marked 0

      Class-map: class-default (match-any)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 10% (634 kbps)
```

```
          QoS Set
            dscp default
              Packets marked 0

    Class-map: class-default (match-any)
      1879 packets, 156350 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
                           (pkts output/bytes output) 1879/156350
```

```
SPOKE2#sh policy-map int f1/1
 FastEthernet1/1

  Service-policy input: MARKING_DOWN

    Class-map: ISP1.source (match-any)
      8071 packets, 1510146 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: source-address mac CA08.117E.0038
        8071 packets, 1510146 bytes
        5 minute rate 2000 bps
      QoS Set
        qos-group 1
          Packets marked 8071

    Class-map: ISP2.source (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: source-address mac CA03.1122.0038
        0 packets, 0 bytes
        5 minute rate 0 bps
      QoS Set
        qos-group 2
          Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_UP

    Class-map: ISP1.destination (match-any)
      10250 packets, 2990712 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: destination-address mac CA08.117E.0038
        10250 packets, 2990712 bytes
        5 minute rate 2000 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 10250/2990712
      shape (average) cir 6340000, bc 25360, be 25360
      target shape rate 6340000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
```

```
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 1750/1488676

        Class-map: pRealTime (match-all)
          1750 packets, 1488676 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (1902 kbps), burst bytes 47550, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp af31 (26)
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (1268 kbps)

        Class-map: class-default (match-any)
          8500 packets, 1502036 bytes
          5 minute offered rate 2000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 8500/1502036
          bandwidth 10% (634 kbps)

      Class-map: ISP2.destination (match-any)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: destination-address mac CA03.1122.0038
          0 packets, 0 bytes
          5 minute rate 0 bps
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        shape (average) cir 6340000, bc 25360, be 25360
        target shape rate 6340000

        Service-policy : pchild

          queue stats for all priority classes:
            Queueing
            queue limit 64 packets
            (queue depth/total drops/no-buffer drops) 0/0/0
            (pkts output/bytes output) 0/0

          Class-map: pRealTime (match-all)
            0 packets, 0 bytes
            5 minute offered rate 0000 bps, drop rate 0000 bps
            Match:  dscp ef (46)
            Priority: 30% (1902 kbps), burst bytes 47550, b/w exceed drops: 0


          Class-map: pInteractive (match-all)
            0 packets, 0 bytes
            5 minute offered rate 0000 bps, drop rate 0000 bps
            Match:  dscp af31 (26)
```

```
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (1268 kbps)

        Class-map: class-default (match-any)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 10% (634 kbps)

    Class-map: class-default (match-any)
      570 packets, 61926 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
                           (pkts output/bytes output) 570/61926
```

MULTICAST

Tabla vecinos Multicast:

```
SPOKE2#sh ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.1.1.1), 01:41:58/00:02:35, RP 10.2.0.1, flags: SJC
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 01:17:28/00:02:35

(*, 224.0.1.40), 03:39:29/00:02:33, RP 10.2.0.1, flags: SJCL
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 03:39:27/00:02:33

SPOKE2#
```

RP configurados:

```
SPOKE2#sh ip pim rp
Group: 225.1.1.1, RP: 10.2.0.1, uptime 01:43:34, expires never
Group: 224.0.1.40, RP: 10.2.0.1, uptime 03:41:05, expires never
SPOKE2#
```

Miembros del grupo *Multicast* registrados en el segmento local:

```
SPOKE2#sh ip pim nei
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.2.0.1          Tunnel10                 03:41:15/00:01:25 v2    1 / S P G
SPOKE2#
```

Sesión DMVPN:

```
SPOKE2#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
     N - NATed, L - Local, X - No Socket
     # Ent --> Number of NHRP entries with same NBMA peer
     NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
     UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel1 is up/up, Addr. is 10.1.0.4, VRF ""
   Tunnel Src./Dest. addr: 91.35.197.14/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.1.0.2  RE priority = 0 cluster = 0
10.1.0.1  RE priority = 0 cluster = 0
10.1.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.21.1            10.1.0.1    UP 00:11:43    S        10.1.0.1/32
    1 194.56.22.1            10.1.0.2    UP 00:11:38    S        10.1.0.2/32
    1 194.56.21.3           10.1.0.10    UP 00:11:45    S       10.1.0.10/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.4, VRF ""
   Tunnel Src./Dest. addr: 172.16.3.2/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.2.0.2  RE priority = 0 cluster = 0
10.2.0.1  RE priority = 0 cluster = 0
10.2.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
```

```
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.5.1             10.2.0.1    UP 00:11:39    S        10.2.0.1/32
    1 172.16.1.1             10.2.0.2    UP 00:11:32    S        10.2.0.2/32
    1 172.16.5.3            10.2.0.10    UP 00:11:39    S       10.2.0.10/32


Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
Session: [0x65B1303C]
  IKEv2 SA: local 91.35.197.14/500 remote 194.56.21.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:48:16
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.1
  IPSEC FLOW: permit 47 host 91.35.197.14 host 194.56.21.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 414 drop 0 life (KB/Sec) 4367921/2902
        Outbound: #pkts enc'ed 426 drop 0 life (KB/Sec) 4367921/2902
   Outbound SPI : 0x17FDDD0B, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B13134]
  IKEv2 SA: local 91.35.197.14/500 remote 194.56.22.1/500 Active
          Capabilities:(none) connid:3 lifetime:23:48:20
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.22.1
  IPSEC FLOW: permit 47 host 91.35.197.14 host 194.56.22.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 406 drop 0 life (KB/Sec) 4209014/2900
        Outbound: #pkts enc'ed 404 drop 0 life (KB/Sec) 4209018/2900
   Outbound SPI : 0x 7DFC1E4, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x65B1322C]
  IKEv2 SA: local 91.35.197.14/500 remote 194.56.21.3/500 Active
          Capabilities:(none) connid:2 lifetime:23:48:14
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.3
  IPSEC FLOW: permit 47 host 91.35.197.14 host 194.56.21.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 422 drop 0 life (KB/Sec) 4330454/2894
        Outbound: #pkts enc'ed 415 drop 0 life (KB/Sec) 4330458/2894
   Outbound SPI : 0x22EE3B98, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B1341C]
  IKEv2 SA: local 172.16.3.2/500 remote 172.16.5.1/500 Active
          Capabilities:(none) connid:5 lifetime:23:48:20
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.1
  IPSEC FLOW: permit 47 host 172.16.3.2 host 172.16.5.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 445 drop 0 life (KB/Sec) 4278215/2900
        Outbound: #pkts enc'ed 438 drop 0 life (KB/Sec) 4278219/2900
   Outbound SPI : 0xD94BCCAA, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B13324]
```

```
  IKEv2 SA: local 172.16.3.2/500 remote 172.16.1.1/500 Active
          Capabilities:(none) connid:4 lifetime:23:48:26
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.1.1
  IPSEC FLOW: permit 47 host 172.16.3.2 host 172.16.1.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 396 drop 0 life (KB/Sec) 4369575/2906
        Outbound: #pkts enc'ed 429 drop 0 life (KB/Sec) 4369575/2906
   Outbound SPI : 0xC32C0C95, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x65B13514]
  IKEv2 SA: local 172.16.3.2/500 remote 172.16.5.3/500 Active
          Capabilities:(none) connid:6 lifetime:23:48:20
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.3
  IPSEC FLOW: permit 47 host 172.16.3.2 host 172.16.5.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 410 drop 0 life (KB/Sec) 4334669/2900
        Outbound: #pkts enc'ed 434 drop 0 life (KB/Sec) 4334670/2900
   Outbound SPI : 0xD2403844, transform : esp-aes esp-sha-hmac
    Socket State: Open

Pending DMVPN Sessions:
```

SPOKE3

ROUTING:

Tabla de vecinos:

```
SPOKE3#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
5   10.1.0.2                Tu1                      11 00:13:24  109   654  0  50
4   10.2.0.2                Tu10                     14 00:13:25  135   810  0  49
3   10.2.0.1                Tu10                     13 00:13:28  485  2910  0  54
2   10.2.0.10               Tu10                     12 00:13:29  460  2760  0  48
1   10.1.0.1                Tu1                      11 00:13:30  358  2148  0  56
0   10.1.0.10               Tu1                      12 00:13:36  118   708  0  47
SPOKE3#
```

Tabla de topología:

```
SPOKE3# sh ip eigrp topo
EIGRP-IPv4 Topology Table for AS(100)/ID(194.135.136.116)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Repl
       r - reply Status, s - sia Status

P 10.3.2.0/24, 1 successors, FD is 309760
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.1.0.0/24, 1 successors, FD is 332800
        via Connected, Tunnel1
P 172.16.0.0/16, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.2.0/24, 1 successors, FD is 360960
       10.2.0.4 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.4 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.2 (399360/322560), Tunnel1
       10.2.0.4 via 10.2.0.10 (365824/314624), Tunnel10
        via 10.1.0.1 (386560/309760), Tunnel1
        via 10.1.0.10 (391424/314624), Tunnel1
P 10.4.1.0/24, 1 successors, FD is 360960
       10.2.0.3 via 10.2.0.1 (360960/309760), Tunnel10
       10.2.0.3 via 10.2.0.2 (373760/322560), Tunnel10
        via 10.1.0.2 (399360/322560), Tunnel1
       10.2.0.3 via 10.2.0.10 (365824/314624), Tunnel10
        via 10.1.0.1 (386560/309760), Tunnel1
        via 10.1.0.10 (391424/314624), Tunnel1
P 10.2.0.0/24, 1 successors, FD is 307200
        via Connected, Tunnel10
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.2.0.2 (309760/28160), Tunnel10
        via 10.1.0.2 (335360/28160), Tunnel1
P 10.4.3.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet1/0
P 10.3.1.0/24, 1 successors, FD is 309760
        via 10.2.0.1 (309760/28160), Tunnel10
        via 10.2.0.10 (314624/33024), Tunnel10
        via 10.1.0.1 (335360/28160), Tunnel1
        via 10.1.0.10 (340224/33024), Tunnel1
```

Tabla NHRP:

```
SPOKE3#sh ip nhrp
10.1.0.1/32 via 10.1.0.1
   Tunnel1 created 00:16:49, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.1
10.1.0.2/32 via 10.1.0.2
   Tunnel1 created 00:16:49, never expire
   Type: static, Flags: used
   NBMA address: 194.56.22.1
10.1.0.10/32 via 10.1.0.10
   Tunnel1 created 00:16:49, never expire
   Type: static, Flags: used
   NBMA address: 194.56.21.3
10.2.0.1/32 via 10.2.0.1
   Tunnel10 created 00:16:48, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.1
10.2.0.2/32 via 10.2.0.2
   Tunnel10 created 00:16:48, never expire
   Type: static, Flags: used
   NBMA address: 172.16.1.1
10.2.0.10/32 via 10.2.0.10
   Tunnel10 created 00:16:48, never expire
   Type: static, Flags: used
   NBMA address: 172.16.5.3
```

CALIDAD DE SERVICIO

```
SPOKE3#sh policy-map int f1/0
 FastEthernet1/0

  Service-policy input: MARKING_UP

    Class-map: RealTime (match-all)
      795 packets, 1095510 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 101
      QoS Set
        dscp ef
          Packets marked 795

    Class-map: Interactive (match-all)
      2122 packets, 2969867 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 102
      QoS Set
        dscp af31
          Packets marked 2122

    Class-map: class-default (match-any)
      177 packets, 242221 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_DOWN
```

```
  Class-map: qos1 (match-any)
    1348 packets, 91611 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 1
      1348 packets, 91611 bytes
      5 minute rate 0 bps
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 1348/91611
    shape (average) cir 2530000, bc 10120, be 10120
    target shape rate 2530000

    Service-policy : child

      queue stats for all priority classes:
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0

      Class-map: RealTime (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 101
        Priority: 30% (759 kbps), burst bytes 18950, b/w exceed drops: 0

        QoS Set
          dscp ef
            Packets marked 0

      Class-map: Interactive (match-all)
        0 packets, 0 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: access-group 102
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 20% (506 kbps)
        QoS Set
          dscp af31
            Packets marked 0

      Class-map: class-default (match-any)
        1348 packets, 91611 bytes
        5 minute offered rate 0000 bps, drop rate 0000 bps
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 1348/91611
        bandwidth 10% (253 kbps)
        QoS Set
          dscp default
            Packets marked 1348

  Class-map: qos2 (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 2
      0 packets, 0 bytes
```

```
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 2530000, bc 10120, be 10120
      target shape rate 2530000

      Service-policy : child

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: RealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 101
          Priority: 30% (759 kbps), burst bytes 18950, b/w exceed drops: 0

          QoS Set
            dscp ef
              Packets marked 0

        Class-map: Interactive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 102
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (506 kbps)
          QoS Set
            dscp af31
              Packets marked 0

        Class-map: class-default (match-any)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 10% (253 kbps)
          QoS Set
            dscp default
              Packets marked 0

    Class-map: class-default (match-any)
      1921 packets, 159566 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 1921/159566
```

```
SPOKE3#sh policy-map int f1/1
 FastEthernet1/1

  Service-policy input: MARKING_DOWN

    Class-map: ISP1.source (match-any)
      9590 packets, 1729468 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: source-address mac CA08.117E.0054
        9590 packets, 1729468 bytes
        5 minute rate 2000 bps
      QoS Set
        qos-group 1
          Packets marked 9590

    Class-map: ISP2.source (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: source-address mac CA03.1122.0054
        0 packets, 0 bytes
        5 minute rate 0 bps
      QoS Set
        qos-group 2
          Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: GLOBAL_UP

    Class-map: ISP1.destination (match-any)
      12570 packets, 6166692 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: destination-address mac CA08.117E.0054
        12570 packets, 6166692 bytes
        5 minute rate 2000 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/4/0
      (pkts output/bytes output) 12566/6160716
      shape (average) cir 2530000, bc 10120, be 10120
      target shape rate 2530000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 1431/1219530

        Class-map: pRealTime (match-all)
          1431 packets, 1219530 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (759 kbps), burst bytes 18950, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          2122 packets, 3139420 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
```

```
        Match:  dscp af31 (26)
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/4/0
        (pkts output/bytes output) 2118/3133444
        bandwidth 20% (506 kbps)

      Class-map: class-default (match-any)
        9017 packets, 1807742 bytes
        5 minute offered rate 2000 bps, drop rate 0000 bps
        Match: any
        Queueing
        queue limit 64 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 9017/1807742
        bandwidth 10% (253 kbps)

    Class-map: ISP2.destination (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: destination-address mac CA03.1122.0054
        0 packets, 0 bytes
        5 minute rate 0 bps
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 2530000, bc 10120, be 10120
      target shape rate 2530000

      Service-policy : pchild

        queue stats for all priority classes:
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

        Class-map: pRealTime (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          Priority: 30% (759 kbps), burst bytes 18950, b/w exceed drops: 0


        Class-map: pInteractive (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp af31 (26)
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
          bandwidth 20% (506 kbps)

        Class-map: class-default (match-any)
          0 packets, 0 bytes
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: any
          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
```

```
        bandwidth 10% (253 kbps)

  Class-map: class-default (match-any)
    586 packets, 63833 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
                          (pkts output/bytes output) 586/63833
```

MULTICAST

Tabla vecinos Multicast:

```
SPOKE3#sh ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.1.1.1), 01:31:44/00:02:45, RP 10.2.0.1, flags: SJC
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 01:28:44/00:02:45

(*, 224.0.1.40), 03:29:25/00:02:39, RP 10.2.0.1, flags: SJCL
  Incoming interface: Tunnel10, RPF nbr 10.2.0.1
  Outgoing interface list:
    FastEthernet1/0, Forward/Sparse, 03:29:24/00:02:39

SPOKE3#
```

RP configurados:

```
SPOKE3#sh ip pim rp
Group: 225.1.1.1, RP: 10.2.0.1, uptime 01:32:21, expires never
Group: 224.0.1.40, RP: 10.2.0.1, uptime 03:30:02, expires never
SPOKE3#
```

Miembros del grupo *Multicast* registrados en el segmento local:

```
SPOKE3#sh ip pim nei
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.2.0.1          Tunnel10                 03:42:58/00:01:39 v2    1 / S P G
SPOKE3#
```

Sesión DMVPN:

```
SPOKE3#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
     N - NATed, L - Local, X - No Socket
     # Ent --> Number of NHRP entries with same NBMA peer
     NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
     UpDn Time --> Up or Down Time for a Tunnel
==========

Interface Tunnel1 is up/up, Addr. is 10.1.0.5, VRF ""
   Tunnel Src./Dest. addr: 194.135.136.116/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.1.0.2  RE priority = 0 cluster = 0
10.1.0.1  RE priority = 0 cluster = 0
10.1.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 194.56.21.1           10.1.0.1    UP 00:10:06    S        10.1.0.1/32
    1 194.56.22.1           10.1.0.2    UP 00:10:00    S        10.1.0.2/32
    1 194.56.21.3          10.1.0.10    UP 00:10:11    S       10.1.0.10/32

Interface Tunnel10 is up/up, Addr. is 10.2.0.5, VRF ""
   Tunnel Src./Dest. addr: 172.16.4.2/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect "cisco-ipsec-ikev2"
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
IPv4 Registration Timer: 3 seconds

IPv4 NHS:
10.2.0.2  RE priority = 0 cluster = 0
10.2.0.1  RE priority = 0 cluster = 0
10.2.0.10  RE priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.5.1            10.2.0.1    UP 00:10:02    S        10.2.0.1/32
    1 172.16.1.1            10.2.0.2    UP 00:09:59    S        10.2.0.2/32
    1 172.16.5.3           10.2.0.10    UP 00:10:06    S       10.2.0.10/32


Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel1
```

```
Session: [0x693C91A0]
  IKEv2 SA: local 194.135.136.116/500 remote 194.56.21.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:49:50
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.1
  IPSEC FLOW: permit 47 host 194.135.136.116 host 194.56.21.1
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 354 drop 0 life (KB/Sec) 4288605/3000
        Outbound: #pkts enc'ed 360 drop 0 life (KB/Sec) 4288608/3000
   Outbound SPI : 0x9F35BBB5, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x693C9390]
  IKEv2 SA: local 194.135.136.116/500 remote 194.56.22.1/500 Active
          Capabilities:(none) connid:3 lifetime:23:49:59
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.22.1
  IPSEC FLOW: permit 47 host 194.135.136.116 host 194.56.22.1
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 337 drop 0 life (KB/Sec) 4370842/3002
        Outbound: #pkts enc'ed 334 drop 0 life (KB/Sec) 4370845/3002
   Outbound SPI : 0xFEC29659, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel1
Session: [0x693C9298]
  IKEv2 SA: local 194.135.136.116/500 remote 194.56.21.3/500 Active
          Capabilities:(none) connid:2 lifetime:23:49:48
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 194.56.21.3
  IPSEC FLOW: permit 47 host 194.135.136.116 host 194.56.21.3
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 367 drop 0 life (KB/Sec) 4214612/2988
        Outbound: #pkts enc'ed 354 drop 0 life (KB/Sec) 4214616/2988
   Outbound SPI : 0x4D52F798, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x693C8FB0]
  IKEv2 SA: local 172.16.4.2/500 remote 172.16.5.1/500 Active
          Capabilities:(none) connid:5 lifetime:23:49:57
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.1
  IPSEC FLOW: permit 47 host 172.16.4.2 host 172.16.5.1
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 375 drop 0 life (KB/Sec) 4155635/3000
        Outbound: #pkts enc'ed 375 drop 0 life (KB/Sec) 4155638/3000
   Outbound SPI : 0x 8FE3188, transform : esp-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel10
Session: [0x693C8EB8]
  IKEv2 SA: local 172.16.4.2/500 remote 172.16.1.1/500 Active
          Capabilities:(none) connid:4 lifetime:23:49:59
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.1.1
  IPSEC FLOW: permit 47 host 172.16.4.2 host 172.16.1.1
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 348 drop 0 life (KB/Sec) 4359844/3000
        Outbound: #pkts enc'ed 377 drop 0 life (KB/Sec) 4359843/3000
   Outbound SPI : 0x9DD10F8F, transform : esp-aes esp-sha-hmac
    Socket State: Open
```

```
Interface: Tunnel10
Session: [0x693C90A8]
  IKEv2 SA: local 172.16.4.2/500 remote 172.16.5.3/500 Active
          Capabilities:(none) connid:6 lifetime:23:49:50
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 172.16.5.3
  IPSEC FLOW: permit 47 host 172.16.4.2 host 172.16.5.3
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 351 drop 0 life (KB/Sec) 4226331/3002
        Outbound: #pkts enc'ed 385 drop 0 life (KB/Sec) 4226330/3002
   Outbound SPI : 0x6C1C53FE, transform : esp-aes esp-sha-hmac
    Socket State: Open

                               Pending DMVPN Sessions:
```

CONFIGURACIÓN DE NAGIOS CORE

Servicio PING

```
# 'check_ping' command definition
define command{
       command_name check_ping
       command_line /usr/lib/nagios/plugins/check_ping -H '$HOSTADDRESS$' -w '$ARG1$'
-c '$ARG2$'
       }

# 'check_ping_tunnel10'
define command{
        command_name    check_ping_tunnel10
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTTUNNEL10$' -w
'$ARG1$' -c '$ARG2$'
        }

# 'check_ping_tunnel1'
define command{
        command_name    check_ping_tunnel1
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTTUNNEL1$' -w
'$ARG1$' -c '$ARG2$'
        }

# 'check_ping_int1'
define command{
        command_name    check_ping_int1
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTINT1$' -w
'$ARG1$' -c '$ARG2$'
        }

# 'check_ping_int1'
define command{
        command_name    check_ping_int2
        command_line    /usr/lib/nagios/plugins/check_ping -H '$_HOSTINT2$' -w
'$ARG1$' -c '$ARG2$'
        }
```

Máquina Nagios

```
define host{
        use                     generic-host
        host_name               Nagios
        alias                   Nagios
        address                 127.0.0.1
        }

# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
        use                             generic-service         ; Name of service
template to use
        host_name                       Nagios
        service_description             Current Users
        check_command                   check_users!20!50
        }


# Define a service to check the number of currently running procs
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

define service{
        use                             generic-service
        host_name                       Nagios
        service_description             Total Processes
             check_command             check_procs!250!400
        }



# Define a service to check the load on the local machine.

define service{
        use                             generic-service         ; Name of service
template to use
        host_name                       Nagios
        service_description             Current Load
             check_command                   check_load!5.0!4.0!3.0!10.0!6.0!4.0
        }
```

HUB1

```
define host{
        use                     generic-host
        host_name               HUB1
        alias                   HUB1
        address                 10.3.1.1
       _tunnel10                10.2.0.1
       _tunnel1                 10.1.0.1
       _int1                    172.16.5.1
       _int2                    194.56.21.1
         }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             f0/0 - 10.3.1.1
        check_command                   check_ping!200.0,20%!300.0,30%
```

```
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             Tunnel10 - 10.2.0.1
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             Tunnel1 - 10.1.0.1
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             f1/1 - 172.16.5.1
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1
        service_description             f1/0 - 194.56.21.1
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```

HUB1-backup

```
define host{
        use                     generic-host
        host_name               HUB1-BACKUP
        alias                   HUB1-BACKUP
        address                 10.3.1.5
       _tunnel10                10.2.0.10
       _tunnel1                 10.1.0.10
       _int1                    172.16.5.3
       _int2                    194.56.21.3
        }

define service{
        use                             generic-service
        host_name                       HUB1-BACKUP
        service_description             f0/0 - 10.3.1.3
        check_command                   check_ping!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1-BACKUP
        service_description             Tunnel10 - 10.2.0.10
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1-BACKUP
        service_description             Tunnel1 - 10.1.0.10
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }
```

```
define service{
        use                             generic-service
        host_name                       HUB1-BACKUP
        service_description             f1/1 - 172.16.5.3
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB1-BACKUP
        service_description             f1/0 - 194.56.21.3
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```

HUB2

```
define host{
        use                     generic-host
        host_name               HUB2
        alias                   HUB2
        address                 10.3.2.1
      _tunnel10               10.2.0.2
      _tunnel1                10.1.0.2
      _int1                   172.16.1.1
      _int2                   194.56.22.1
        }

define service{
        use                             generic-service
        host_name                       HUB2
        service_description             f1/0 - 10.4.1.1
        check_command                   check_ping!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB2
        service_description             Tunnel10 - 10.2.0.2
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB2
        service_description             Tunnel1 - 10.1.0.2
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB2
        service_description             f1/1 - 172.16.1.1
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       HUB2
        service_description             f0/0 - 194.56.22.1
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```

SPOKE1

```
define host{
        use                     generic-host
        host_name               SPOKE1
        alias                   SPOKE1
        address                 10.4.1.1
       _tunnel10                10.2.0.3
       _tunnel1                 10.1.0.3
       _int1                    172.16.2.2
       _int2                    81.134.91.2
        }

define service{
        use                             generic-service
        host_name                       SPOKE1
        service_description             f1/0 - 10.4.1.1
        check_command                   check_ping!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE1
        service_description             Tunnel10 - 10.2.0.3
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE1
        service_description             Tunnel1 - 10.1.0.3
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE1
        service_description             f1/1 - 172.16.2.2
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE1
        service_description             f0/0 - 81.134.91.2
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```

SPOKE2

```
define host{
        use                     generic-host
        host_name               SPOKE2
        alias                   SPOKE2
        address                 10.4.2.1
       _tunnel10                10.2.0.4
       _tunnel1                 10.1.0.4
       _int1                    172.16.3.2
       _int2                    91.35.197.14
        }

define service{
        use                             generic-service
        host_name                       SPOKE2
```

```
        service_description             f1/0 - 10.4.2.1
        check_command                   check_ping!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE2
        service_description             Tunnel10 - 10.2.0.4
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE2
        service_description             Tunnel1 - 10.1.0.4
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE2
        service_description             f1/1 - 172.16.3.2
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE2
        service_description             f1/0 - 91.35.197.14
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```

SPOKE3

```
define host{
        use                     generic-host
        host_name               SPOKE3
        alias                   SPOKE3
        address                 10.4.3.1
       _tunnel10                10.2.0.5
       _tunnel1                 10.1.0.5
       _int1                    172.16.4.2
       _int2                    194.135.136.116
        }


define service{
        use                             generic-service
        host_name                       SPOKE3
        service_description             f1/0 - 10.4.3.1
        check_command                   check_ping!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service         ; Name of service
template to use
        host_name                       SPOKE3
        service_description             Tunnel10 - 10.2.0.5
        check_command                   check_ping_tunnel10!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE3
        service_description             Tunnel1 - 10.1.0.5
```

```
        check_command                   check_ping_tunnel1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE3
        service_description             f1/1 - 172.16.4.2
        check_command                   check_ping_int1!200.0,20%!300.0,30%
        }

define service{
        use                             generic-service
        host_name                       SPOKE3
        service_description             f0/0 - 194.135.136.116
        check_command                   check_ping_int2!200.0,20%!300.0,30%
        }
```