



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA
IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET**

Gersson Josué Guoz Cúmez

Asesorado por la Inga. Ingrid Salomé Rodríguez García de Loukota

Guatemala, agosto de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA
IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

GERSSON JOSUÉ GUOZ CÚMEZ

ASESORADO POR LA INGA. INGRID SALOMÉ RODRÍGUEZ GARCÍA DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, AGOSTO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Ing. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
EXAMINADOR	Ing. Helmut Federico Chicol Cabrera
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
SECRETARIA	Inga. Lesbia Magalí Herrera López de López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 27 de febrero de 2020.

Gersson Josué Guoz Cúmez

Guatemala 23 de febrero 2021

Ingeniero
Julio César Solares Peñate
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Apreciable Ingeniero Solares,

Me permito dar aprobación al trabajo de graduación titulado "**Diseño de un manual de configuración para la implementación de equipos Huawei en una red Ethernet**", del señor **Gersson Josué Guoz Cúmez**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones de este.

Sin otro particular, me es grato saludarle.

Atentamente,

A handwritten signature in black ink, reading "Ingrid Rodríguez de Loukota". The signature is written in a cursive style and is underlined with a single horizontal line.

Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiado 5356



Guatemala, 11 de marzo de 2021

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET**, desarrollado por el estudiante **Gersson Josué Guoz Cúmez**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS

A handwritten signature in blue ink, appearing to read 'Julio César Solares Peñate'.

Ing. Julio César Solares Peñate
Coordinador de Electrónica



REF. EIME 115. 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; GERSSON JOSUÉ GUOZ CÚMEZ titulado: DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET, procede a la autorización del mismo.


Ing. Armando Alonso Rivera Carrillo



GUATEMALA, 12 DE AGOSTO 2,021.

DTG. 359-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE UN MANUAL DE CONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE EQUIPOS HUAWEI EN UNA RED ETHERNET**, presentado por el estudiante universitario: **Gersson Josué Guoz Cúmez**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, agosto de 2021

AACE/asga

ACTO QUE DEDICO A:

Dios	Por darme el tiempo de vida para lograr esta meta.
Mis padres	Gloria Evangelina Cúmez García y José Eusebio Guoz Esquit, por ser mi más grande fuente de inspiración y consuelo.
Mis hermanos	Mishell y Eduardo Guoz Cúmez, por apoyarme y estar a mi lado siempre. Junto a ellos la vida ha sido grandiosa.
Mis abuelos	Rosa Esquit, Isidro Guoz, Sebastiana García y José Cúmez (q. e. p. d.), cuyas historias de vida me han motivado sobremanera.
Mis tíos	En especial a Reyna Patricia Tetzaguic Socop (q. e. p. d.) y Carlos Fermín Guoz Esquit (q. e. p. d.), por sus consejos y ánimos para seguir adelante sin importar la adversidad.
Mis amigos	Con quienes compartí mis años de carrera universitaria y gracias a su apoyo pude finalizar mis estudios.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	XV
GLOSARIO	XVII
RESUMEN	XXIII
OBJETIVOS.....	XXV
INTRODUCCIÓN	XXVII
1. BREVE HISTORIA DE HUAWEI TECHNOLOGIES CO., LTD.....	1
1.1. Estado actual.....	2
1.2. La guerra por el campo de las telecomunicaciones.....	3
2. MARCO TEÓRICO.....	5
2.1. Fundamentos de la comunicación en red	5
2.1.1. Modelo OSI.....	5
2.1.2. Modelo TCP/IP	8
2.2. El simulador eNSP	9
2.2.1. Interfaz del usuario	10
2.2.2. Iniciando dispositivos	10
2.2.3. La interfaz de línea de comandos.....	11
2.2.4. Configuración física de un dispositivo.....	12
2.2.5. Conexión entre dispositivos	13
2.2.6. WireShark.....	14
2.3. El sistema operativo Huawei VRP	15
2.3.1. Líneas de comandos	15

2.3.2.	Ingresando a un dispositivo por el puerto de consola	18
2.3.3.	Configuración básica	21
2.3.3.1.	Nombre del equipo	21
2.3.3.2.	Hora y fecha	21
2.3.3.3.	Dirección IP en una interfaz	22
2.3.3.4.	Restringiendo acceso al equipo	23
2.3.4.	Comandos básicos	26
2.4.	Ethernet.....	28
2.4.1.	Dominio de colisión	28
2.4.2.	Dominio de Broadcast	29
2.4.3.	CSMA/CD	30
2.4.4.	Comunicación Half- y Full-Dúplex	32
2.4.4.1.	Half-Dúplex.....	33
2.4.4.2.	Full-Dúplex	33
2.4.5.	Ethernet en la capa de enlace de datos	34
2.4.5.1.	Direccionamiento Ethernet	35
2.4.5.2.	Tramas Ethernet.....	36
2.4.6.	Ethernet en la capa física	37
2.4.6.1.	Cableado Ethernet	38
2.4.6.1.1.	Cable directo	42
2.4.6.1.2.	Cable cruzado	43
2.4.6.1.3.	Cable de fibra óptica	44
2.4.6.1.4.	Transceptores SFP	46
2.4.7.	<i>Switches</i> Ethernet	47
2.4.7.1.	Principio de operación.....	48
2.4.7.2.	Tabla de direcciones MAC	53
2.4.8.	Encapsulación de datos	54
2.4.9.	Address Resolution Protocol	56

	2.4.9.1.	Funcionamiento	57
	2.4.9.2.	Tabla ARP	60
	2.4.9.3.	Formato del paquete ARP	60
	2.4.10.	Envío de datos a través de una red Ethernet	63
2.5.		Enrutamiento IP	66
	2.5.1.	Proceso de enrutamiento.....	66
	2.5.2.	Analizando la tabla de enrutamiento.....	67
	2.5.3.	Enrutamiento estático	71
	2.5.3.1.	Ruta por defecto	76
	2.5.4.	Enrutamiento dinámico	78
	2.5.4.1.	Preferencia de la ruta	78
	2.5.4.2.	Costo de la ruta	79
	2.5.5.	Elección de la mejor ruta	80
	2.5.6.	Traza de un paquete.....	81
2.6.		Routing Information Protocol	82
	2.6.1.	Principio de operación	83
	2.6.2.	Temporizadores.....	88
	2.6.2.1.	Temporizador de actualización	88
	2.6.2.2.	Temporizador para invalidar rutas	88
	2.6.2.3.	Temporizador para eliminar rutas	88
	2.6.3.	RIP-1 vs. RIP-2.....	89
	2.6.4.	Implementando RIP	90
	2.6.5.	Verificando la configuración de RIP	93
2.7.		Open Shortest Path First	96
	2.7.1.	Principio de operación	96
	2.7.2.	Tipos de paquetes OSPF	101
	2.7.3.	Encapsulación de los paquetes OSPF	102
	2.7.4.	Tipos de redes OSPF	102
	2.7.5.	<i>Routers</i> vecinos.....	103

2.7.6.	Adyacencia OSPF	106
2.7.7.	OSPF en redes Broadcast y NBMA	107
2.7.7.1.	DR Y BDR	108
2.7.8.	Costo de una ruta	109
2.7.9.	Áreas en OSPF	111
2.7.10.	Tipos de <i>routers</i> OSPF.....	112
2.7.11.	Tipos de rutas OSPF	113
2.7.12.	Implementando OSPF área única	114
2.7.13.	Verificando la configuración de OSPF área única	120
2.7.14.	Diferencia al implementar OSPF multi-área	123
2.8.	VLAN y enrutamiento Inter-VLAN	123
2.8.1.	Funcionamiento.....	123
2.8.2.	Puerto en modo acceso y troncal	127
2.8.3.	GVRP	130
2.8.4.	VLANIF.....	133
2.8.5.	Enrutamiento Inter-VLAN	137
2.8.5.1.	Enrutamiento con subinterfaces	138
2.8.5.2.	Enrutamiento con VLANIF	141
2.9.	Spanning-Tree Protocol	144
2.9.1.	Principio de operación	144
2.9.2.	BPDU	149
2.9.3.	Elección de Root Bridge	152
2.9.4.	Elección del rol de los puertos.....	154
2.9.5.	Estados de los puertos	158
2.9.6.	Variantes de STP	160
2.9.6.1.	STP	160
2.9.6.2.	RSTP	161
2.9.6.3.	MSTP	163
2.9.7.	Bridge ID para RSTP y MSTP	164

2.9.8.	Edge Port y BPDU Protection	164
2.9.9.	Root Protection	166
2.9.10.	Implementando RSTP	167
2.9.11.	Verificando la configuración de RSTP	170
2.10.	Enlaces Ethernet-Trunk	173
2.10.1.	Implementando enlaces Eth-Trunk manual	173
2.10.2.	Verificando la configuración de Eth-Trunk	175
2.11.	Servicios IP	177
2.11.1.	DHCP	177
2.11.1.1.	Agente DHCP Relay	179
2.11.1.2.	Implementando DHCP	179
2.11.1.3.	Verificando la configuración de DHCP	182
2.11.1.4.	Implementando agente DHCP Relay.	183
2.11.2.	ACL	185
2.11.2.1.	Análisis de las sentencias	186
2.11.2.2.	Implementando ACL básica	187
2.11.2.3.	Implementando ACL avanzada	190
2.11.2.4.	Verificando una ACL	191
2.11.3.	NAT	192
2.11.3.1.	Implementando NAT estático	193
2.11.3.2.	Implementando NAT dinámico	195
2.11.3.3.	Implementando NAPT	197
2.11.3.4.	Easy-IP	198
2.11.4.	VRRP	199
2.11.4.1.	Implementando VRRP	202
2.11.4.2.	Verificando la configuración de VRRP	204
2.11.5.	SYSLOG	205

CONCLUSIONES.....209
RECOMENDACIONES211
BIBLIOGRAFÍA.....213

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Ejemplo de una red que comunica dos computadoras.....	5
2.	Modelo de referencia OSI	6
3.	Algunos protocolos de la pila TCP/IP	9
4.	Simulador eNSP.....	9
5.	Interfaz del usuario del simulador eNSP	10
6.	Métodos para iniciar y apagar un dispositivo en eNSP	11
7.	Abriendo la CLI en eNSP	12
8.	Vista frontal y trasera de un <i>router</i> Huawei AR1220	13
9.	Conexión entre dos <i>routers</i> empleando cables de cobre	14
10.	WireShark en el simulador eNSP	15
11.	Vista del usuario de un <i>router</i> AR201.....	16
12.	Vista del sistema de un <i>router</i> AR201	16
13.	Vista de la Interfaz Ethernet0/0/0 de un <i>router</i> AR201	17
14.	Comando <i>quit</i>	17
15.	Puerto de consola de un <i>router</i> AR201	18
16.	Cables para la conexión serial por el puerto de consola	19
17.	Diálogo inicial de un <i>router</i> AR201	20
18.	Configuración del <i>host-name</i>	21
19.	Configuración de hora y fecha en un <i>router</i>	22
20.	Configuración de dirección IP en una interfaz.....	23
21.	Interfaces del usuario de un <i>router</i> AR201.....	24
22.	Configuración de contraseña en el puerto de consola	24
23.	Creación de un usuario implementando autenticación AAA local	26

24.	Dominios de colisión y dominio de Broadcast.....	30
25.	Funcionamiento de CSMA/CD	32
26.	Half- y Full-Dúplex	34
27.	Modelo OSI y estándares Ethernet.....	34
28.	Dirección MAC	35
29.	Trama Ethernet.....	36
30.	Cable UTP y conector RJ-45	41
31.	Conexión T-568A y T-568B	42
32.	Cable directo.....	43
33.	Cable cruzado.....	44
34.	Tipo de conexión para cables de 1Gbps de ancho de banda	44
35.	Estructura del cable de fibra óptica.....	45
36.	Conectores para fibra óptica	46
37.	<i>Transceiver SFP</i>	47
38.	Parte trasera de un <i>switch</i> Huawei S3700	47
39.	Trama ingresando al <i>switch</i> S3700 rumbo a PC4.....	49
40.	Primera entrada de la tabla de direcciones MAC.....	50
41.	Ejemplo de <i>flooding</i> en un <i>switch</i>	51
42.	Nueva trama dirigida hacia PC1	52
43.	Segunda entrada de la tabla de direcciones MAC	52
44.	Proceso de encapsulación.....	56
45.	ARP Request	58
46.	ARP Reply	59
47.	Entrada en la tabla ARP de PC1 y PC3.....	59
48.	Estructura de una trama ARP	61
49.	Envío de información de PC1 a PC2	64
50.	Recepción de información de PC2 a PC1.....	66
51.	Ejemplo de enrutamiento IP.....	67
52.	Tabla de enrutamiento	68

53.	Consulta de una ruta específica y direcciones IP	70
54.	Ejemplo de una topología para implementar rutas estáticas	71
55.	Tabla de enrutamiento de R1	72
56.	Ruta estática en R1	73
57.	Ruta estática en la tabla de enrutamiento de R1	74
58.	Configuración de una ruta estática en R2	75
59.	Tabla de enrutamiento en R2	75
60.	<i>Router</i> conectado hacia Internet	76
61.	Configuración de una ruta por defecto en R1	77
62.	Tabla de enrutamiento de R1	77
63.	Tabla de enrutamiento filtrada para mostrar rutas estáticas	80
64.	Traza de PC1 a PC4	82
65.	Topología de red implementando RIP	83
66.	Mensajes RIP enviados por R1	85
67.	Mensajes RIP enviados por R2 y R5	85
68.	Métrica o costo de la ruta en RIP	86
69.	Topología para implementar RIP-2	90
70.	Configuración de R1	91
71.	Configuración de R2	92
72.	Configuración de R3	92
73.	Vecinos RIP de R1	93
74.	Configuración RIP de R1	94
75.	Tabla de enrutamiento de R1	95
76.	<i>Ping</i> de PC1 a PC3	95
77.	Traza de PC1 a PC3	96
78.	Paquetes Hello enviados	97
79.	Comparación del resumen de la LSDB de R1 y R2	98
80.	Solicitud de actualización para sincronizar la LSDB	99
81.	Envío de actualización para sincronizar la LSDB	99

82.	Encapsulación de los paquetes OSPF.....	102
83.	Tipos de redes en OSPF	103
84.	Proceso para formar adyacencia OSPF	106
85.	Adyacencia OSPF en una red Broadcast	107
86.	Ecuación para calcular el costo de una interfaz.....	109
87.	Costo de la ruta en OSPF.....	110
88.	Áreas OSPF en una red.....	111
89.	Tipos de <i>routers</i> en OSPF.	113
90.	Topología propuesta para implementar OSPF área única.....	114
91.	Costos de las interfaces.....	116
92.	Configuración de R1	117
93.	Configuración de R2	118
94.	Configuración de R3	119
95.	Configuración de R4	120
96.	Configuración de OSPF en R1	121
97.	Adyacencias OSPF de R1	122
98.	Rutas aprendidas por OSPF de R1	122
99.	Trama Ethernet con Etiqueta 802.1q	124
100.	Trama sin etiqueta 802.1q ingresando al <i>switch</i>	125
101.	Trama sin etiqueta saliendo del <i>switch</i>	126
102.	Configuración de un puerto en modo acceso	128
103.	Configuración de un puerto en modo troncal	129
104.	VLAN asignadas a los puertos de un <i>switch</i>	130
105.	Topología de ejemplo para implementar GVRP	131
106.	Configuración de SW1	131
107.	Configuración de SW2.....	132
108.	Configuración de SW3.....	132
109.	VLAN estáticas y dinámicas en SW1.....	133
110.	VLANIF en dos <i>switches</i>	134

111.	Configuración de SW1	135
112.	Configuración de SW2	135
113.	Tabla ARP de SW1	136
114.	<i>Ping</i> desde SW1 hacia SW2	137
115.	Topología propuesta para implementar enrutamiento Inter-VLAN con subinterfaces.....	138
116.	Configurando subinterfaces en R1	139
117.	Configurando puertos en SW1	140
118.	Estado de las interfaces de R1	140
119.	Etiquetado 802.1q en las subinterfaces de R1	141
120.	Topología propuesta para implementar enrutamiento Inter-VLAN con VLANIF	142
121.	Configurando VLANIF en SWL3	143
122.	Estado de las interfaces en SWL3	143
123.	Tabla ARP de SWL3	144
124.	Análisis de STP en una red.....	145
125.	Inundación de tramas STP.....	146
126.	Roles de los puertos.....	147
127.	Ruta alterna encontrada por STP.....	148
128.	Estructura de un BPDU	149
129.	Estructura de <i>Bridge ID</i>	150
130.	Estructura de <i>Port ID</i>	151
131.	Ejemplo de elección de Root Bridge	153
132.	Costo de las rutas hacia el Root Bridge	155
133.	Puertos <i>root</i> en la topología	156
134.	Rol de los puertos en la topología.....	157
135.	Estados de los puertos de STP.....	159
136.	Proceso para estabilizar el estado del puerto en STP	160
137.	Bridge ID con sistema extendido.....	164

138.	Configurando BPDU Protection y Edge Port.....	165
139.	Puerto bloqueado por recibir BPDU.....	166
140.	Configuración de Root Protection	166
141.	Red Broadcast para implementar RSTP	167
142.	Configuración de SW1	168
143.	Configuración de SW2.....	169
144.	Configuración de SW3.....	169
145.	Configuración de SW4.....	169
146.	Ruta de comunicación creada por RSTP	170
147.	Configuración STP en SW3	171
148.	Rol y estado de los puertos en SW3.....	171
149.	Configuración RSTP de GE0/0/3 de SW3	172
150.	Cambio de topología STP.....	173
151.	Enlace Eth-Trunk entre SW1 y SW2.....	174
152.	Configuración de SW1	175
153.	Estado del Puerto Eth-Trunk 1 en SW1	176
154.	Estado del Puerto Eth-Trunk 1 de SW1	176
155.	Proceso DHCP.....	177
156.	Topología para implementar DHCP	180
157.	Configuración de R1	181
158.	Dirección IP de PC 1 obtenida por DHCP.....	182
159.	Dirección IP de PC 1 obtenida por DHCP.....	183
160.	Topología para implementar <i>router</i> como agente DHCP Relay.....	184
161.	Configuración de R1	185
162.	Configuración del servidor DHCP	185
163.	Topología para implementar ACL	188
164.	Creación de una ACL básica en R1.....	189
165.	Creación de una ACL avanzada en R1.....	191
166.	Configuración de las ACL de R1.....	192

167.	Topología para implementar NAT	194
168.	Configurando NAT estático en R1.....	195
169.	NAT estático en R1	195
170.	Configurando NAT dinámico en R1.....	196
171.	NAT Dinámico en R1.....	197
172.	Configurando NAT en R1.....	197
173.	Configurando Easy-IP en R1.....	198
174.	Easy-IP en R1	199
175.	Topología para implementar VRRP	200
176.	Configuración de SW1	203
177.	Configuración de SW2	203
178.	Configuración de PC 1	204
179.	Configuración VRRP de SW1	205
180.	Estructura de los registros Syslog.....	207
181.	Registros de un <i>router</i> Huawei.....	208

TABLAS

I.	Valores predeterminados para establecer comunicación serial	20
II.	Listado de comandos básicos	27
III.	Estándares Ethernet.....	38
IV.	Ancho de banda y longitud para cables UTP y fibra óptica	40
V.	Definición de los campos de una trama ARP	61
VI.	Preferencias por defecto	79
VII.	Ejemplo de la tabla de enrutamiento de R1	87
VIII.	Diferencia entre RIP-1 y RIP-2.....	89
IX.	Tipos de paquetes OSPF	101
X.	Costos en base al estándar IEEE 802.1t.....	151
XI.	Roles de los puertos en STP y RSTP	162

XII.	Estado de los puertos en STP y RSTP	163
XIII.	Número identificador de las ACL	186
XIV.	Direcciones IPv4 privadas	192
XV.	Niveles de severidad de los registros	206

LISTA DE SÍMBOLOS

Símbolo	Significado
L1	Layer 1, referente a la primera capa del modelo de referencia OSI
L2	Layer 2, referente a la segunda capa del modelo de referencia OSI
L3	Layer 3, referente a la tercera capa del modelo de referencia OSI
P2MP	Point to Multi-Point.
P2P	Point-to-Point.
Rx	Recepción
Tx	Transmisión
1Gbps	Velocidad de transmisión de 1 gigabit por segundo
10Mbps	Velocidad de transmisión de 10 megabits por segundo
100Mbps	Velocidad de transmisión de 100 megabits por segundo

GLOSARIO

AP	Access Point, punto de acceso utilizado generalmente para distribuir señal Wi-Fi.
AS	Autonomous Systems, conjunto de equipos de red administrados por una misma entidad.
ARP	Address Resolution Protocol.
ATM	Asynchronous Transfer Mode, estándar para el envío de información digital de datos, video y voz.
<i>Bandwidth</i>	Ancho de banda que hace referencia a la tasa de transmisión de datos de un enlace.
Bits	Unidad básica de información que puede tener el valor de uno o cero lógicos.
BPDU	Bridge Protocol Data Unit, unidad de datos usada en STP.
<i>Bridge</i>	Dispositivo usando antiguamente como concentrador de red, actualmente es usado para referirse a los <i>switches</i> .

Bytes	Unidad de información digital compuesta por ocho bits.
CPU	Unidad Central de Procesamiento.
CRC	Cyclic Redundancy Check, técnica utilizada para detectar alteración de datos por el medio transmisión.
DWDM	Multiplexación por división de longitud de onda densa utilizada frecuentemente cuando se manejan anchos de banda elevados.
EGP	Exterior Gateway Protocols, protocolos para la comunicación entre sistemas autónomos.
eNSP	Enterprise Network Simulation Platform, herramienta de simulación propietaria de Huawei.
FDDI	Interfaz de datos distribuida por fibra óptica.
GARP	Generic Attribute Registration Protocol.
GSM	Global System for Mobile Communications, estándar que normaliza las redes celulares digitales.
HDLC	High-Level Data Link Control, estándar de comunicación para segmentos punto a punto.

Host	Nombre que se atribuye a cualquier dispositivo final en una red.
HTTP	Hypertext Transfer Protocol, protocolo de la capa de aplicación del modelo OSI para la distribución de hipermedia.
Hub	Primer concentrador de red que centralizaba los cables conectados a él para formar un único dominio de colisión.
IEEE	Instituto de Ingenieros Electricistas y Electrónicos.
IGP	Interior Gateway Protocols, protocolos para la comunicación dentro de un mismo sistema autónomo.
IP	Internet Protocol.
IS-IS	Intermediate System to Intermediate System, protocolo de enrutamiento dinámico de estado de enlace.
ISO	Organización Internacional de Normalización.
ISP	Proveedor de Servicios de Internet.
LAN	Red de área local.
NBMA	Non-Broadcast Multi-Access.

NIC	Network Interface Controller.
NSE	Network Security Expert.
OSI	Interconexión de Sistemas Abiertos.
OTN	Red de transporte óptico.
OUI	Organizationally Unique Identifier.
<i>Padding</i>	Proceso donde se rellena con unos lógicos los campos faltantes del paquete ARP.
<i>Patchcord</i>	Cable eléctrico u óptico usado para conectar dos equipos.
PBX	Private Branch Exchange, sistema telefónico que permite la administración de llamadas nacionales e internacionales.
PDU	Protocol Data Unit.
PVID	Port VLAN Identifier.
<i>Router</i>	Enrutador, dispositivo capaz de encontrar la mejor ruta hacia un destino y redirigir un paquete.
SFP	Small Form-factor Pluggable, transceptor de fibra óptica.

SSH	Secure Shell, protocolo que permite administrar y transferir datos encriptados.
<i>Subnetting</i>	Proceso que divide una red en pequeñas subredes basándose en la modificación de la máscara de subred.
<i>Switch</i>	Dispositivo capaz de conmutar tráfico al segmentar los dominios de colisión en una red.
TCP	Transmission Control Protocol.
TELNET	Teletype Network, Protocolo que permite administrar y transferir datos en texto plano.
<i>Transceiver</i>	Dispositivo capaz de recibir y enviar información por medios eléctricos u ópticos.
UTP	Unshielded Twisted Pair.
VLAN	Red de área local virtual, capaz de segmentar dominios de Broadcast a nivel lógico.
VLANIF	VLAN Interface.
VRP	Versatile Routing Platform, sistema operativo de los dispositivos Huawei.
VTY	Virtual Tele-type.

WAN

Wide Area Network o red de área amplia.

WI-FI

Wireless Fidelity, tecnología que permite crear redes LAN de forma inalámbrica en la frecuencia de 2.4GHz o 5GHz.

RESUMEN

La compañía Huawei Technologies ha tenido un crecimiento exponencial en Latinoamérica en los últimos años, a tal punto que los más grandes proveedores de servicios de Internet ya cuentan con topologías WAN implementando exclusivamente estos equipos debido a su alto rendimiento, bajo costo y el gran soporte que ofrecen.

A medida que Guatemala se adapta a este inminente cambio, resulta esencial que los profesionales en telecomunicaciones posean las herramientas y conocimiento teórico y práctico necesario para la implementación de equipos de red Huawei y hacer frente a los requerimientos actuales.

El presente trabajo describe brevemente el modo de operación de los protocolos de comunicación más utilizados en redes Ethernet con el propósito de formar una base teórica al momento de su implementación, adicional muestra el uso básico de la herramienta de simulación eNSP para la creación y estudio de topologías de red y describe los comandos básicos para la configuración y verificación de *routers*, *switches* y *switches* multicapa Huawei para iniciarlos en la producción.

OBJETIVOS

General

Diseñar un manual de configuración que incluya los conceptos teóricos y prácticos para la implementación de equipos Huawei en una red Ethernet.

Específicos

1. Presentar los fundamentos teóricos de los protocolos que rigen las comunicaciones en las redes Ethernet.
2. Mostrar las características más importantes de la herramienta eNSP para la simulación de topologías de red.
3. Detallar la configuración de equipos Huawei implementando la herramienta de simulación eNSP.
4. Describir ejemplos prácticos implementando múltiples protocolos de comunicación en equipos Huawei.

INTRODUCCIÓN

La empresa pionera en la creación de protocolos de comunicación y equipos de red desde los inicios del Internet fue Cisco Systems, su innovación fue tal que hasta hace algunos años gran cantidad de los ISP a nivel mundial utilizaban exclusivamente sus productos en su arquitectura WAN.

Los protocolos creados por Cisco Systems son registrados y usados exclusivamente en sus equipos, sin embargo, debido al crecimiento exponencial que Internet tuvo a finales del siglo XX el IEEE creó estándares abiertos análogos a los protocolos registrados para incentivar el crecimiento de la red global, logrando así que dispositivos de diferentes marcas logaran comunicarse.

Hoy en día Huawei Technologies, empresa creada en China en 1987 por Ren Zhengfei, ha logrado posicionarse como la empresa preferida para abastecer equipos de red a los grandes proveedores de servicios y al público en general. Esta marca ha implementado los estándares abiertos del IEEE a sus equipos logrando una comunicación multiplataforma.

Dicho cambio de tecnología ha llegado a tal punto que los grandes ISP a nivel centroamericano están migrando su arquitectura completa a esta nueva marca, ya que ofrece menor costo e iguales prestaciones. Es así como se ha logrado posicionar como la marca preferida para la migración de la arquitectura de red en los próximos años, cambiando el campo de las telecomunicaciones.

1. BREVE HISTORIA DE HUAWEI TECHNOLOGIES CO., LTD.

Huawei Technologies Co., Ltd. es una empresa China fundada en 1987 por Ren Zhengfei quien era distribuidor de productos PBX importados. Hacia 1993, luego de desarrollar y distribuir sus propios equipos PBX, logró un importante avance al crear el *switch* C&C08 marcando una gran diferencia en capacidad de procesamiento comparado con sus competidores, de esta forma se posiciona como el principal distribuidor para centrales telefónicas de china.

En el año 1995 las ventas ascendieron a 1 500 millones de renminbis, moneda de China, procedentes principalmente de los mercados rurales de China. En 1996 empieza a proveer equipo telefónico para la compañía transnacional Hongkong's Hutchison-Whampoa y a finales de 1997 lanza su producto GSM y se expande para abarcar tecnologías CDMA y UMTS.

En el año 2000 las cifras de ingreso aumentan a 100 millones de dólares estadounidenses. Y en 2003, Huawei crea una alianza estratégica llamada Huawei-3Com para iniciarse en el mundo de los *routers* y *switches* basados en protocolos de Internet. En 2007 firma con la empresa de seguridad Estadounidense Symantec para desarrollar equipos de seguridad y almacenamiento de datos.

A mediados del año 2008 la revista BusinessWeek reconoce a Huawei como la empresa más influyente del mundo. Para este momento se une con Optus, una empresa desarrolladora de tecnología móvil de Sídney, Australia, para encaminarse a la adopción de la banda ancha móvil e inalámbrica de alta

velocidad. Un año después, Huawei gana el Green Mobile Award en el GSMA Mobile Awards.

Para el 2011, Huawei ya contaba con 20 centros de datos de computación en la nube y su incursión en la fabricación de teléfonos celulares muestra un éxito rotundo alcanzando más de 20 millones de unidades vendidas. Un año después se da a conocer el primer sistema de transporte óptico DWDM de 400 Gbps y se lanza una tarjeta de línea de 480 Gbps que suponía la mayor capacidad de la industria en el área IP. En 2015, las redes LTE de Huawei dan cobertura a más de 140 capitales del mundo y en el área de transporte óptico colaboró con un operador europeo para construir la primera OTN de 1 Tbps del mundo y adicionalmente con BT para completar las pruebas de transmisión óptica de 3 Tbps en redes de tráfico real.

1.1. Estado actual

El crecimiento de Huawei ha llegado a tal nivel que en julio del 2019 la lista Fortune 500 la posicionó en el número 61 entre las empresas más influyentes de la actualidad, así mismo el estudio anual de las 100 Marcas Globales Más Valiosas realizada por Brandz la posicionó como el número 47.

Durante el mes de agosto, en la Conferencia para Desarrolladores Huawei se lanzó HarmonyOS, el nuevo sistema operativo, basado en microkernel, que logró ser compacto y ligero con poderosas funciones, y será usado en relojes, pantallas y bocinas inteligentes, así como en sistemas internos de vehículos.

Una de las principales características de los dispositivos, y por la cual está teniendo un gran impacto en el mundo, es la accesibilidad en el precio. El factor económico ha demostrado ser decisivo en la competición con sus adversarios y

está causando que los grandes proveedores de servicios de internet migren su arquitectura completa al conseguir las mismas prestaciones que otros equipos por un menor precio.

Es de esta forma que la empresa china ya cuenta con una gran presencia en el continente americano, y específicamente en Guatemala, tanto en el campo de las telecomunicaciones como en la telefonía móvil. Los grandes proveedores de servicios de internet en el país han optado por adoptar esta nueva tecnología en sus equipos Core, haciendo imprescindible que los profesionales en el campo de las telecomunicaciones sepan su manejo y configuración.

1.2. La guerra por el campo de las telecomunicaciones

Al hablar de los equipos de red más utilizados a nivel nacional no se puede pasar por alto a la empresa Cisco Systems. Cisco es una empresa estadounidense creada en California en el año 1984 y desde sus inicios fue pionera en la creación de protocolos de comunicación y equipos de red. Durante varios años sus equipos de red gobernaban el campo de las telecomunicaciones a nivel mundial. Esta empresa fue creadora de algunos de los protocolos más importantes que se usan en las redes hoy en día y fue hasta años posteriores que el IEEE creó estándares abiertos para que la comunicación entre dispositivos se lograra sin importar el fabricante.

No obstante, toda la investigación que se realizó para la creación de nuevas tecnologías y protocolos de comunicación tuvo impacto en el precio de los equipos. Y aunque inicialmente esto no representó un gran problema para la empresa, ya que poseían los mejores equipos, a medida que otras empresas como Ericsson, Juniper o Huawei fueron adaptando y mejorando estas

tecnologías, el monopolio de los equipos Cisco fue disminuyendo progresivamente.

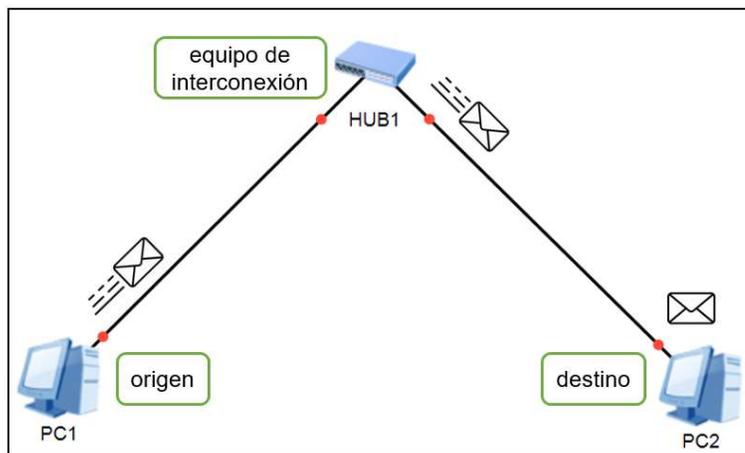
Actualmente se puede decir que la principal razón por la que los proveedores de servicio y los administradores de redes escogen una empresa para proveer los equipos es el precio y es en este campo en el que resulta vencedor Huawei Technologies. Esta nueva empresa china probó ser capaz de fabricar equipos con las mismas prestaciones que otras empresas, pero con un costo mucho menor. Es en base a esto que el futuro de las telecomunicaciones en Guatemala está inclinado en la adopción de equipos Huawei.

2. MARCO TEÓRICO

2.1. Fundamentos de la comunicación en red

Una red es una ruta o un camino que comunica a dos o más dispositivos. Existen múltiples protocolos que se pueden utilizar para crear una red, no obstante, el conjunto o *stack* de protocolos más usados en internet es TCP/IP en el cual va incluido el protocolo Ethernet. Antes de poder hablar de los protocolos que conforman TCP/IP es importante recordar el Modelo OSI.

Figura 1. **Ejemplo de una red que comunica dos computadoras**



Fuente: elaboración propia, empleando eNSP.

2.1.1. Modelo OSI

En el año 1980 la ISO creó el Modelo de referencia de interconexión de sistemas abiertos, Modelo OSI, que cumplió la tarea de estandarizar el proceso de comunicación entre dispositivos de distintos fabricantes.

El modelo de referencia OSI divide el proceso para establecer la comunicación en red en siete capas, cada una cumpliendo una tarea específica para lograr la transmisión e interpretación de datos. Las últimas tres capas se encargan de la interacción con el usuario final mientras que las primeras cuatro se centran en la manipulación de la información para su envío y recepción.

Figura 2. **Modelo de referencia OSI**



Fuente: elaboración propia.

A continuación, se brinda una breve descripción del propósito de cada capa:

- La capa de aplicación se encarga de la interacción entre las aplicaciones finales y la red. Ejemplos de aplicaciones finales pueden ser: navegadores web, procesadores de texto, hojas de cálculo, entre otros. Cabe mencionar que estas aplicaciones no residen en la capa de aplicación, sino interactúan con ella. Algunas de las aplicaciones que residen en la capa siete son: Telnet, SSH, FTP, TFTP, DNS, HTTP, DHCP, POP, SMTP, XMPP, entre otros.

- La capa de presentación se encarga de traducir o convertir la información para que las aplicaciones que interactúan con la capa siete, navegador web, visor de imágenes, Microsoft Word, entre otros, puedan hacer uso de ella. También se encarga de la compresión, encriptación y de dar formato a los datos. Un ejemplo de un proceso en la capa de presentación ocurre cuando una aplicación que representa sus caracteres con formato ASCII debe enviar datos a través de la red hacia otra aplicación que representa sus caracteres con formato EBCDIC.
- La capa de sesión se encarga de establecer, mantener separadas y terminar las sesiones entre los *hosts*, los cuales pueden tener varias sesiones funcionando de forma paralela. En una red local una sesión se puede dar cuando un *host* solicita información de un servidor web utilizando el puerto 80, o cuando se están transfiriendo archivos utilizando el protocolo FTP por el puerto 20, la capa de sesión se encarga de iniciar y mantener separadas estas sesiones para que funcionen al mismo tiempo sin afectarse una a la otra, es de esta forma que ocurre el diálogo entre los dispositivos.
- La capa de transporte se responsabiliza de establecer conexiones de forma lógica ya sean confiables o no, utilizando direcciones de transporte denominados puertos. También se encarga de controlar la velocidad de transmisión, el ancho de ventana, ajusta la secuencia y flujo de los datos transmitidos. Alguno de los protocolos que residen en esta capa son: TCP y UDP.
- La capa de red se encarga del direccionamiento lógico de los dispositivos y del enrutamiento de los paquetes. El direccionamiento lógico se logra empleando el protocolo IP en su versión cuatro o seis mientras que el

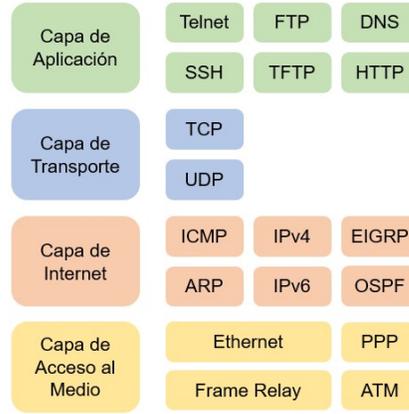
enrutamiento de paquetes hace referencia al hecho de encontrar la mejor ruta para transportar un paquete hacia un destino. Algunos de los protocolos que residen en esta capa son: IPv4, IPv6, ICMP y ARP.

- La capa de enlace de datos se encarga del direccionamiento físico de los dispositivos y establece un enlace lógico entre dispositivos conectados físicamente, también se encarga de la recuperación de datos corruptos. Los protocolos que funcionan en esta capa pueden ser Ethernet, ATM, Frame Relay, HDLC, entre otros
- Finalmente, la capa física se encarga del envío y recepción de la información representada por *bits*. La codificación ocurre en esta capa y se representa por medio de alteraciones de alguna magnitud físico como intensidad eléctrica, lumínico o de ondas electromagnéticas. También se responsabiliza de la coordinación para acceder al medio. En esta capa residen protocolos como RS 232, DSL, Ethernet y algunos otros para el manejo de fibra óptica.

2.1.2. Modelo TCP/IP

El protocolo TCP/IP realmente es una *stack* o pila de protocolos que se utilizan para la comunicación a través de Internet. En cada capa del modelo de referencia TCP/IP existe un conjunto de protocolos que pueden utilizarse.

Figura 3. **Algunos protocolos de la pila TCP/IP**

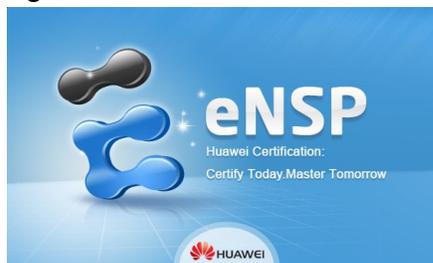


Fuente: elaboración propia.

2.2. El simulador eNSP

El simulador eNSP es una herramienta propietaria de Huawei que es capaz de virtualizar dispositivos de red como *routers*, *switches*, *switches* multicapa, firewalls, computadoras, entre otros. Una de las principales diferencias de eNSP frente a otros simuladores es que trabaja en conjunto con programas como: VirtualBox, que virtualiza los sistemas operativos de los equipos, y WireShark que da la opción de analizar el tráfico de datos en tiempo real. Una gran ventaja al utilizar un virtualizador es que permite descargar nuevos sistemas operativo a medida que estos salen al mercado.

Figura 4. **Simulador eNSP**

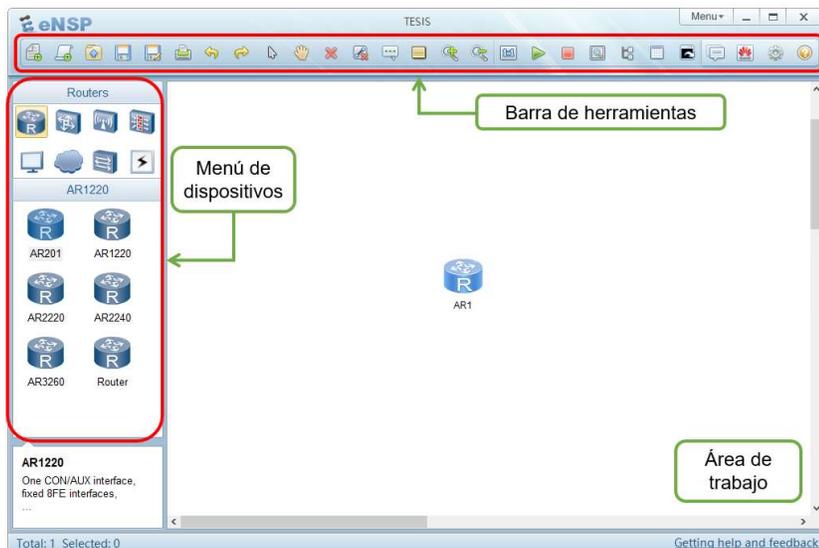


Fuente: elaboración propia, empleando eNSP.

2.2.1. Interfaz del usuario

La interfaz del usuario del simulador eNSP se divide en tres partes principales: el menú de los dispositivos de red disponibles, la barra de herramientas y el área de trabajo para crear la topología, ubicada en la parte central. En la parte izquierda de la pantalla se muestra el menú de dispositivos de red que incluyen *routers*, *switches*, *access points*, *firewalls*, dispositivos finales, cables de conexión, entre otros. Para seleccionar un dispositivo basta con hacer *click* sobre él y arrastrarlo hacia el área de trabajo.

Figura 5. Interfaz del usuario del simulador eNSP



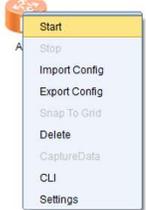
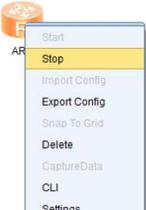
Fuente: elaboración propia, empleando eNSP.

2.2.2. Iniciando dispositivos

Una vez el equipo esté ubicado en el área de trabajo es necesario encenderlo de forma virtual para que su operación inicie. Para esto se debe seleccionar el dispositivo y pulsar la flecha verde *Start Device* de la barra de

herramienta o dando *click* derecho sobre el dispositivo y seleccionado la opción *Start*.

Figura 6. **Métodos para iniciar y apagar un dispositivo en eNSP**

	Botón para iniciar un dispositivo
	Botón para apagar un dispositivo
	Opción <i>Start</i> en el menú del dispositivo
	Opción <i>Stop</i> en el menú del dispositivo

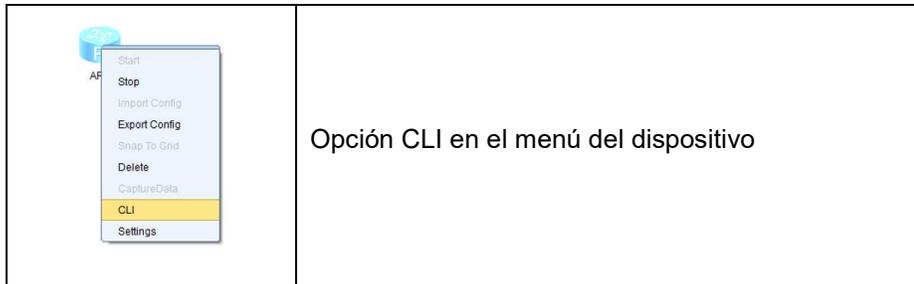
Fuente: elaboración propia, empleando eNSP.

Para apagar el dispositivo se puede seleccionar la opción *Stop* al dar *click* derecho sobre el equipo o pulsando el botón *Stop Device* ubicado en la barra de herramientas.

2.2.3. La interfaz de línea de comandos

Para ingresar a la CLI del equipo, que resulta equivalente a conectarse al dispositivo vía el puerto de consola, es requisito que el equipo esté encendido. Para abrir la CLI se tiene dos opciones: dando doble *click* sobre el dispositivo o dando *click* derecho y seleccionando la opción CLI.

Figura 7. **Abriendo la CLI en eNSP**



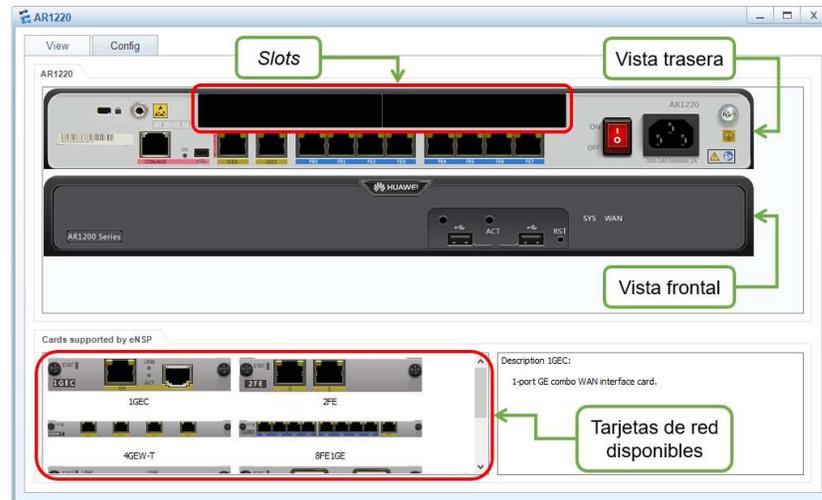
Fuente: elaboración propia, empleando eNSP.

2.2.4. **Configuración física de un dispositivo**

El simulador eNSP cuenta con dos características especiales que le permiten adaptarse a distintas topologías de red sin importar el medio físico empleado para la conexión de los equipos. La primera característica es que cuenta con una vista del equipo de la parte frontal y trasera permitiendo tener una idea del aspecto físico del dispositivo, y la segunda es la opción de agregar adaptadores de red para simular distintos protocolos de capa dos. Estas opciones se encuentran en la sección de configuración del equipo y para acceder a ellas se debe dar *click* derecho sobre el dispositivo y seleccionar la opción *settings*.

La opción de agregar tarjetas de red depende del modelo del equipo debido a que se necesitan *slots* en la parte trasera para insertarlos. Entre las tarjetas de red disponibles se encuentran tarjetas FastEthernet, GigabitEthernet, Serial, POS, Packet-over-SONET/SDH usada para fibra óptica, E1 y SHDSL. Para introducir las tarjetas en el *slot* basta con arrastrarla hasta el *slot* vacío del equipo, igualmente para retirarla se arrastra la tarjeta del equipo hasta la sección de tarjetas de red.

Figura 8. Vista frontal y trasera de un *router* Huawei AR1220



Fuente: elaboración propia, empleando eNSP.

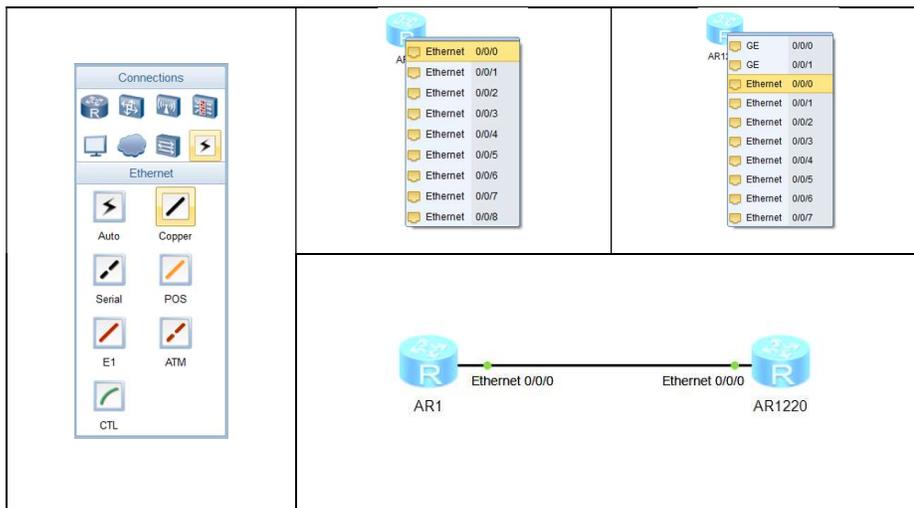
2.2.5. Conexión entre dispositivos

Para conectar dos o más dispositivos se pueden emplear cables de cobre, fibra óptica o por algún medio inalámbrico, con Wi-Fi. El simulador eNSP cuenta con cables de conexión de cobre Ethernet, Serial, E1 y ATM, y para fibra óptica cuenta con conexiones POS. Adicionalmente incorpora un cable de consola, una opción automática que detecta el tipo de tarjeta de red y selecciona el cable adecuado y la opción de conectar dispositivos a un AP utilizando Wi-Fi.

Elegir el cable de conexión adecuado depende de la tarjeta de red que tengan los equipos. Los cables de conexión disponibles se encuentran en la sección *Connections*, ubicado en el menú de dispositivos. Para unir dos dispositivos el primer paso es seleccionar el cable de conexión, luego se hacer *click* sobre uno de los equipos y se desplegará un menú donde se debe elegir el puerto donde se conectará el cable, una vez seleccionado se repite el proceso con el otro equipo. Para validar que existe comunicación a nivel físico entre los

dispositivos, el enlace que los une debe tener una luz verde en ambos extremos, si uno de los extremos tiene una luz roja significa que el puerto no está transmitiendo información, lo que se traduce a que el equipo esté apagado o el puerto desactivado.

Figura 9. **Conexión entre dos *routers* empleando cables de cobre**

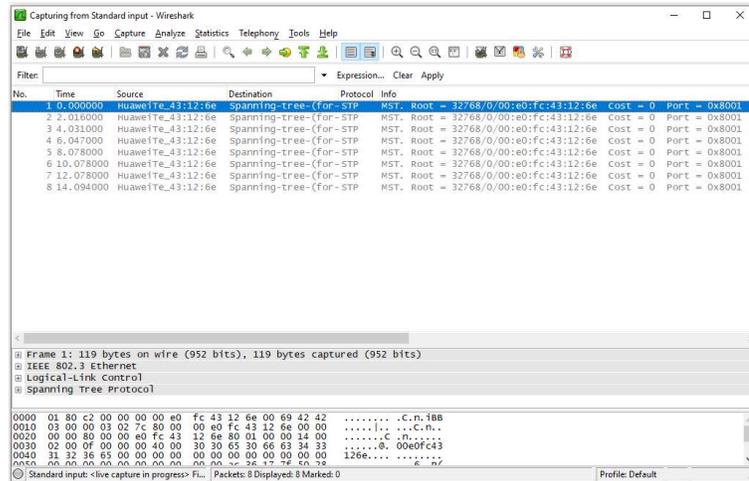


Fuente: elaboración propia, empleando eNSP.

2.2.6. WireShark

Una de las principales ventajas que posee eNSP es la integración de WireShark para el análisis de paquetes de datos en tiempo real. WireShark es conocido como un programa *sniffer*, es decir un capturador de paquetes que viajan a través de un enlace. Para hacer uso de esta herramienta se debe hacer *click* derecho sobre el dispositivo objetivo y seleccionar la opción *CaptureData* del menú desplegable. Esta acción abrirá un submenú con las interfaces activas del equipo, aquí se debe elegir la que se desee monitorear. De esta forma se abrirá una nueva ventana con el programa WireShark y se mostrarán una lista de los paquetes que se están transmitiendo y recibiendo en tiempo real.

Figura 10. WireShark en el simulador eNSP



Fuente: elaboración propia, empleando eNSP.

2.3. El sistema operativo Huawei VRP

Es un sistema operativo propietario de Huawei cuya primera versión es VRP1.0, salió en el año 1998 y, al momento de escrito este trabajo, la versión más actualizada es la octava, VRP8.x, lanzada en el año 2009. Este sistema operativo es aplicado especialmente a *routers* y *switches* y su principal función es la interacción entre el usuario y el dispositivo, empleando una serie de comandos para su manejo y configuración, estos comandos son aplicados a la CLI.

2.3.1. Líneas de comandos

Una línea de comandos VRP hace referencia a un conjunto de caracteres ordenados que se ingresan a un dispositivo para configurar algún parámetro. Cuando un equipo Huawei es iniciado por primera vez este no cuenta con ninguna configuración por lo que el administrador de redes tendrá que utilizar la CLI para ingresar estos comandos y así configurar el equipo.

La interfaz de línea de comandos cuenta con un sistema jerárquico de vistas que permiten consultar información general o configurar el equipo. Al iniciar un dispositivo por primera vez la vista que se muestra en la interfaz es la vista del usuario, esta permite consultar información del dispositivo y acceder a otros niveles de vistas.

Para configurar las funciones del equipo, mostrar la configuración que está siendo ejecutada o poder ingresar a otros niveles de vistas se debe ingresar a la vista del sistema ejecutando el comando: *system-view*, como se observa en la figura 12. Una vez dentro de la vista del sistema se puede acceder a otros tipos de vistas más específicas como la vista de interfaz, al introducir el comando: *interface interface_num* donde el parámetro *interface_num* es el tipo y número de interfaz.

Figura 11. Vista del usuario de un *router* AR201



Fuente: elaboración propia, empleando eNSP.

Figura 12. Vista del sistema de un *router* AR201



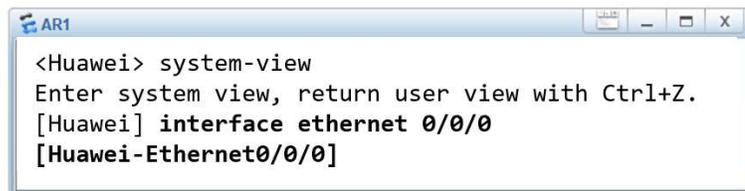
Fuente: elaboración propia, empleando eNSP.

El símbolo del sistema de cada vista es un indicador que muestra el nombre del dispositivo, también llamado *host-name*, encerrado en distintos tipos llaves. Esta cadena de caracteres es conocida frecuentemente como *command prompt*.

En el caso de la vista del usuario el *host-name* se encuentra encerrado entre los símbolos de mayor y menor que, <...>, en la vista del sistema se encuentra encerrado entre paréntesis cuadrados, [...], mientras que al ingresar a la vista de una interfaz se muestra el nombre de la misma, por ejemplo, Ethernet0/0/0, como descripción.

En la figura 13 se muestra la vista de la interfaz Ethernet0/0/0 de un *router* AR201.

Figura 13. **Vista de la interfaz Ethernet0/0/0 de un *router* AR201**

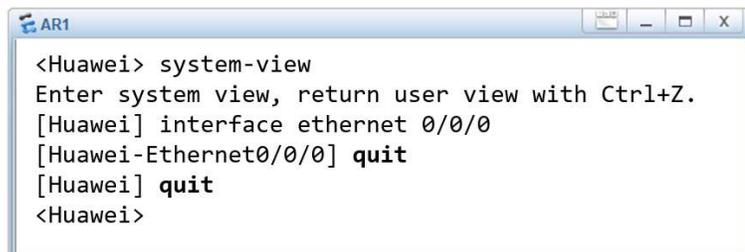


```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] interface ethernet 0/0/0
[Huawei-Ethernet0/0/0]
```

Fuente: elaboración propia, empleando eNSP.

Para salir de un nivel de vista y regresar al anterior se emplea el comando *quit* o la combinación de teclas *Ctrl+z*. Para retornar a la vista del usuario directamente se puede utilizar el comando *return*.

Figura 14. **Comando *quit***



```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] interface ethernet 0/0/0
[Huawei-Ethernet0/0/0] quit
[Huawei] quit
<Huawei>
```

Fuente: elaboración propia, empleando eNSP.

Al igual que muchos otros fabricantes, Huawei incorpora dos grandes características en el sistema VRP: el autocompletado de comandos, utilizando la tecla de tabulación y el uso del signo de interrogación, para solicitar información sobre los comandos disponibles que se pueden ejecutar en la vista seleccionada.

Dentro de la vista del usuario se puede consultar información sobre el estado de las interfaces, protocolos, CPU, entre otra información más. Para mostrar información general se emplea el comando *display*. Evidentemente la información que se puede mostrar de un dispositivo es basta y variada, es por esto que existe una gran lista de comandos que pueden ser escritos acompañados del comando *display*. La combinación de comandos *display current-configuration* muestra la configuración que el dispositivo está ejecutando en la memoria Flash.

2.3.2. Ingresando a un dispositivo por el puerto de consola

Existen tres métodos para ingresar a un dispositivo, a través de las líneas VTY, el puerto MiniUSB o el puerto de consola. Generalmente la configuración inicial del dispositivo se realiza utilizando el puerto de consola, por tal motivo se expone este método exclusivamente.

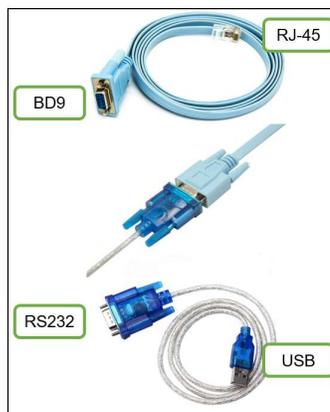
Figura 15. Puerto de consola de un *router* AR201



Fuente: elaboración propia, empleando eNSP.

Todos los dispositivos cuentan con un puerto físico llamado puerto de consola que, en la mayoría de los casos, se encuentra en la parte posterior del equipo. El puerto cuenta con ocho pines y es de tipo RJ-45, el mismo que se utiliza para la comunicación en red entre dispositivos. Para lograr la comunicación del equipo hacia la computadora se puede emplear un cable convertidor de RJ-45 a USB, no obstante, este tipo de cable no es muy común por lo que es más frecuente utilizar cables convertidores de RJ-45 a DB9 y RS232 a USB, la forma de acoplar estos cables se describe en la figura 16.

Figura 16. **Cables para la conexión serial por el puerto de consola**



Fuente: elaboración propia.

Una vez se tenga la conexión física entre los dos dispositivos se puede utilizar algún programa para establecer comunicación serial como PuTTY, HyperTerminal o SecureCRT, por defecto todos los dispositivos Huawei vienen con los valores predeterminados que se muestran en la tabla I, para establecer la comunicación serial.

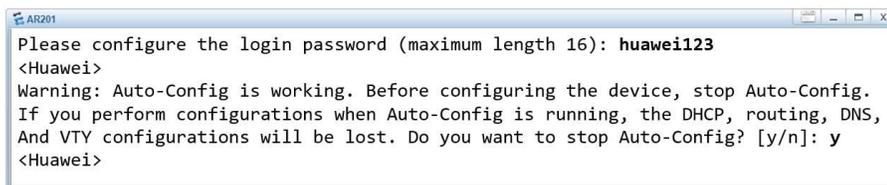
Tabla I. **Valores predeterminados para establecer comunicación serial**

Bits por segundo	9 600
Bits de Datos	8
Paridad	No
Bit de parada	1
Control de Flujo	No

Fuente: elaboración propia

Cuando se ingresa por primera vez a un *router* o *switch* Huawei se solicita una pequeña configuración inicial, en ella se debe establecer una contraseña que restringirá el acceso en futuras ocasiones. Los dispositivos nuevos también ofrecen una opción de autoconfiguración, sin embargo, esta opción no es recomendable y puede ser cancelada al presionar la tecla *y*, como se muestra en la figura 17.

Figura 17. **Diálogo inicial de un *router* AR201**



```
AR201
Please configure the login password (maximum length 16): huawei123
<Huawei>
Warning: Auto-Config is working. Before configuring the device, stop Auto-Config.
If you perform configurations when Auto-Config is running, the DHCP, routing, DNS,
And VTY configurations will be lost. Do you want to stop Auto-Config? [y/n]: y
<Huawei>
```

Fuente: elaboración propia, empleando eNSP.

Luego de la configuración inicial se muestra en pantalla la vista del usuario indicando que el equipo está listo para ser configurado y entrar en producción.

2.3.3. Configuración básica

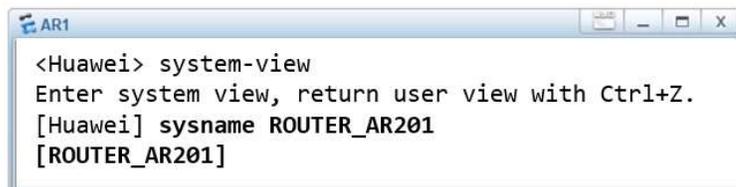
Antes que un equipo entre en producción es necesario configurar ciertos parámetros para su uso, tales como los usuarios para controlar el acceso, la habilitación de las interfaces a utilizar, la fecha y hora para mayor control de las actividades que se registran en el equipo y el nombre del dispositivo para una apropiada organización.

2.3.3.1. Nombre del equipo

El nombre del equipo, también denominado *host-name*, es la etiqueta que lo identifica y aparece siempre en el *command prompt* de la CLI. Para establecer el *host-name* se debe ingresar a la vista del sistema y emplear el comando: *sysname host-name*, donde el parámetro *host-name* es el nombre a asignar.

En la figura 18 se muestra el proceso para configurar el *host-name* a un *router*.

Figura 18. Configuración del *host-name*



```
AR1
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname ROUTER_AR201
[ROUTER_AR201]
```

Fuente: elaboración propia, empleando eNSP.

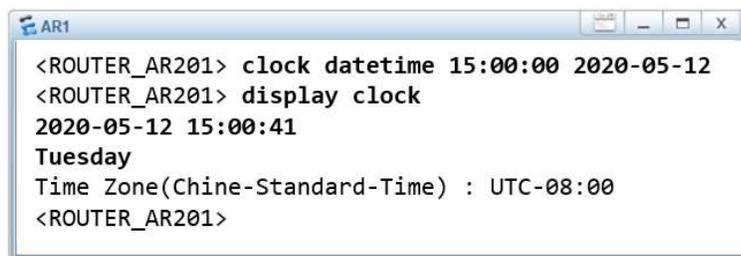
2.3.3.2. Hora y fecha

La hora y fecha del dispositivo puede configurarse de forma local o sincronizarse con un servidor NTP. Esta configuración resulta muy importante

porque los equipos Huawei guardan un registro de las actividades que ocurren, y estos registros pueden ser analizados de mejor forma si el tiempo está configurado correctamente. Para configurar la hora y fecha de forma local se ingresa a la vista del usuario y se emplea el comando: *clock datetime hh:mm:ss y-m-d*.

En la figura 19 se muestra el proceso para configurar la hora y fecha en un *router*.

Figura 19. **Configuración de hora y fecha en un *router***



```
AR1
<ROUTER_AR201> clock datetime 15:00:00 2020-05-12
<ROUTER_AR201> display clock
2020-05-12 15:00:41
Tuesday
Time Zone(Chine-Standard-Time) : UTC-08:00
<ROUTER_AR201>
```

Fuente: elaboración propia, empleando eNSP.

2.3.3.3. Dirección IP en una interfaz

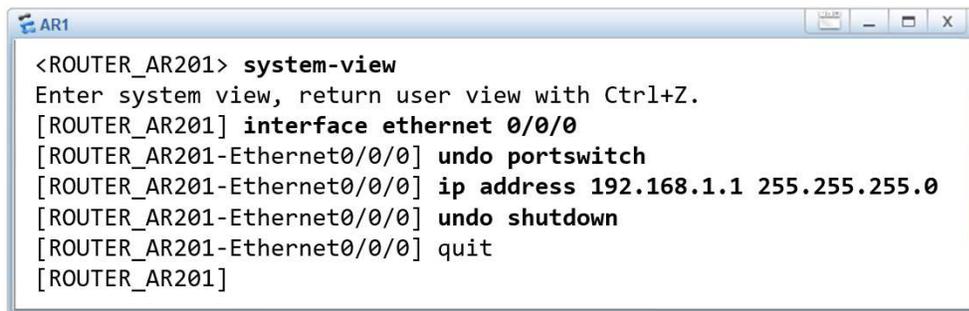
Por lo general en los *switches* Huawei todas las interfaces vienen habilitadas para su uso inmediato, ya que estos equipos cumplen la función de concentradores de red. No obstante, los *routers* necesitan direcciones IP para cumplir la función de enrutar los paquetes, más adelante se profundizará en el tema del enrutamiento.

Para configurar una dirección IP primero se debe habilitar la interfaz para que soporte la configuración L3, para esto, se ingresa a la vista de la interfaz y se emplea el comando *undo portswitch*. Seguido se configura la dirección IP y la máscara de subred con el comando: *ip address ip_address mask*, donde el

parámetro *ip_address* corresponde a la dirección IP y *mask* a la máscara de subred o a su longitud. Finalmente se enciende la interfaz con el comando *undo shutdown*.

El proceso completo para habilitar una dirección IP en una interfaz se puede visualizar en la figura 20.

Figura 20. **Configuración de dirección IP en una interfaz**



```
<ROUTER_AR201> system-view
Enter system view, return user view with Ctrl+Z.
[ROUTER_AR201] interface ethernet 0/0/0
[ROUTER_AR201-Ethernet0/0/0] undo portswitch
[ROUTER_AR201-Ethernet0/0/0] ip address 192.168.1.1 255.255.255.0
[ROUTER_AR201-Ethernet0/0/0] undo shutdown
[ROUTER_AR201-Ethernet0/0/0] quit
[ROUTER_AR201]
```

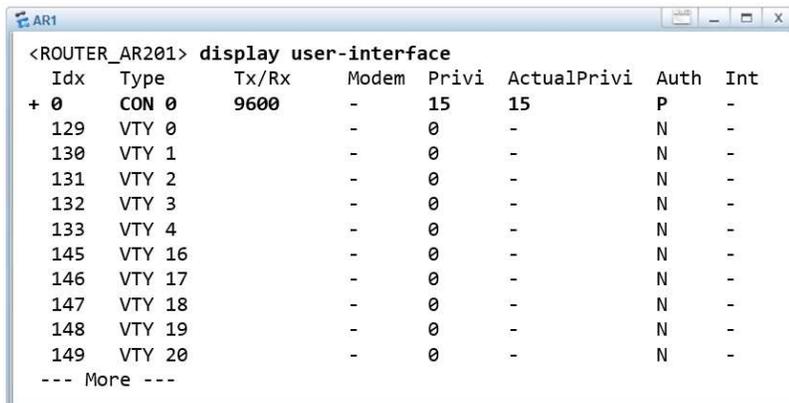
Fuente: elaboración propia, empleando eNSP.

2.3.3.4. Restringiendo acceso al equipo

Para ingresar a un equipo se puede utilizar el puerto de consola o VTY. Para el caso del puerto de consola se debe tener acceso físico al equipo mientras que con las líneas VTY se pueden emplear protocolos como Telnet o SSH para ingresar de forma remota. Si se toma como ejemplo el caso del *router* AR201, este cuenta con un puerto de consola y diez líneas VTY, enumeradas de la cero a la cuatro y de la dieciséis a la veinte, a través de las cuales se puede ingresar de forma separada, es decir que se pueden tener diez usuarios conectados simultáneamente al equipo.

En la figura 21 se puede observar que se tiene únicamente un usuario conectado por medio del puerto de consola, CON 0.

Figura 21. Interfaces del usuario de un *router* AR201



Idx	Type	Tx/Rx	Modem	Privi	ActualPrivi	Auth	Int
+ 0	CON 0	9600	-	15	15	P	-
129	VTY 0		-	0	-	N	-
130	VTY 1		-	0	-	N	-
131	VTY 2		-	0	-	N	-
132	VTY 3		-	0	-	N	-
133	VTY 4		-	0	-	N	-
145	VTY 16		-	0	-	N	-
146	VTY 17		-	0	-	N	-
147	VTY 18		-	0	-	N	-
148	VTY 19		-	0	-	N	-
149	VTY 20		-	0	-	N	-

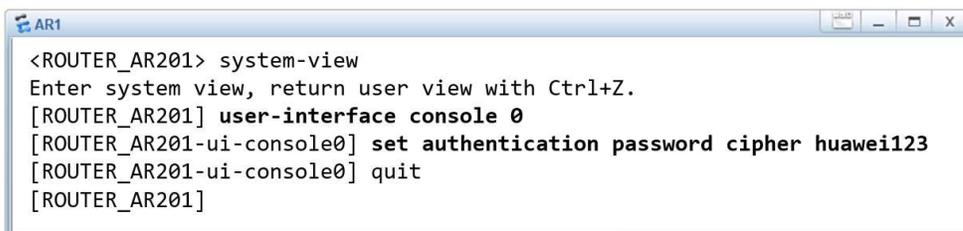
--- More ---

Fuente: elaboración propia, empleando eNSP.

Para restringir acceso al puerto de consola se puede implementar una contraseña local que se solicite cuando algún usuario intente ingresar al equipo. Para esto se debe ingresar a la vista de la interfaz de consola empleando el comando: *user-interface console 0* y se establece la contraseña con el comando: *set authentication password cipher password*, donde el parámetro *password* es la contraseña a configurar.

En la figura 22 se muestra el proceso para restringir el acceso al puerto de consola con la contraseña huawei123 como ejemplo.

Figura 22. Configuración de contraseña en el puerto de consola



```
<ROUTER_AR201> system-view
Enter system view, return user view with Ctrl+Z.
[ROUTER_AR201] user-interface console 0
[ROUTER_AR201-ui-console0] set authentication password cipher huawei123
[ROUTER_AR201-ui-console0] quit
[ROUTER_AR201]
```

Fuente: elaboración propia, empleando eNSP.

Otra forma de restringir el acceso, ya sea en el puerto de consola o en las líneas virtuales VTY, es a través de la creación de usuarios con el protocolo AAA, Authentication, Authorization and Accounting. La autenticación de usuarios implementando el protocolo AAA es de forma local, el uso de servidores RADIUS O TACACS+ está fuera del alcance de este trabajo.

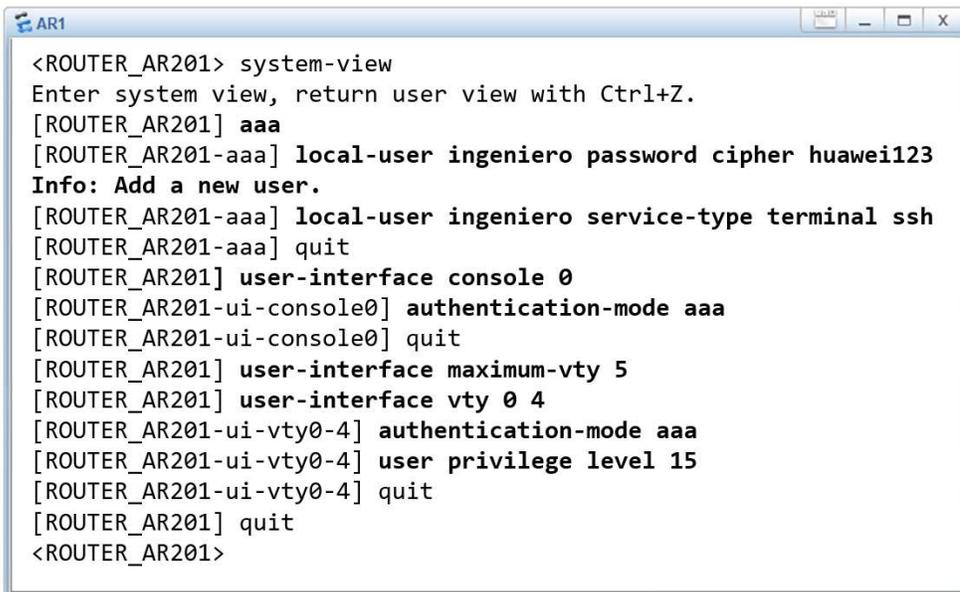
Para la creación de los usuarios se ingresa a la vista del protocolo AAA ejecutando el comando: *aaa* en la vista del sistema. Luego se crea el usuario y la contraseña con el comando: *local-user user password cipher password*. Finalmente se debe indicar en dónde se solicitarán las credenciales del usuario recién creado, es decir si el usuario se utilizará para autenticar el ingreso por el puerto de consola, por las líneas VTY o en ambos. Para autenticar el ingreso únicamente por el puerto de consola se ejecuta el comando: *local-user user service-type terminal*, mientras que para autenticar el ingreso por las líneas VTY se debe indicar si se usará el protocolo SSH, TELNET o ambos con el comando: *local-user user service-type telnet ssh*.

El último paso es ingresar a la vista del puerto de consola empleando el comando: *user-interface console 0*, o a la vista de las líneas VTY empleando el comando: *user-interface vty 0 4*, en este caso se usarán únicamente las líneas de la cero a la cuatro, e indicar que se utilizará la autenticación AAA con el comando: *authentication-mode aaa*. Aunque el *router* AR201 tiene la opción de tener diez líneas VTY activas, por defecto únicamente cinco vienen habilitadas y tienen un nivel de privilegio de cero. El nivel de privilegio es un número entero entre cero y quince que se asigna a las líneas VTY para dar permisos de configuración, siendo el nivel más bajo el cero que permite visualizar información limitada y el nivel más alto quince que permite la configuración completa del equipo. Para modificar el número máximo de líneas VTY habilitadas se ingresa a la vista del sistema y se emplea el comando: *user-interface maximum-vty 5*. Para

asignar un nivel de privilegio de quince a las líneas VTY se ingresa a la vista de las líneas y se emplea el comando: *user privilege level 15*.

En la figura 23 se muestra la configuración para crear un usuario llamado ingeniero con contraseña huawei123 y restringir el acceso vía Telnet y SSH.

Figura 23. **Creación de un usuario implementando autenticación AAA local**



```
<ROUTER_AR201> system-view
Enter system view, return user view with Ctrl+Z.
[ROUTER_AR201] aaa
[ROUTER_AR201-aaa] local-user ingeniero password cipher huawei123
Info: Add a new user.
[ROUTER_AR201-aaa] local-user ingeniero service-type terminal ssh
[ROUTER_AR201-aaa] quit
[ROUTER_AR201] user-interface console 0
[ROUTER_AR201-ui-console0] authentication-mode aaa
[ROUTER_AR201-ui-console0] quit
[ROUTER_AR201] user-interface maximum-vty 5
[ROUTER_AR201] user-interface vty 0 4
[ROUTER_AR201-ui-vty0-4] authentication-mode aaa
[ROUTER_AR201-ui-vty0-4] user privilege level 15
[ROUTER_AR201-ui-vty0-4] quit
[ROUTER_AR201] quit
<ROUTER_AR201>
```

Fuente: elaboración propia, empleando eNSP.

2.3.4. Comandos básicos

La cantidad de comandos para la configuración de un equipo Huawei es inmensa y varía dependiendo del modelo.

En la tabla II, se describe un listado de los comandos básicos que permiten mostrar la información elemental de un equipo.

Tabla II. **Listado de comandos básicos**

Comando	Descripción
<i>display current-configuration</i>	Muestra la configuración actual del equipo cargada en la memoria RAM.
<i>display saved-configuration</i>	Muestra la configuración del equipo cargada en memoria la no volátil.
<i>display this</i>	Muestra la configuración de la vista en la que se encuentre el usuario.
<i>Quit</i>	Retorna a la vista anterior.
<i>Reboot</i>	Reinicia el equipo.
<i>Sabe</i>	Guarda la configuración actual en la memoria no volátil.
<i>schedule reboot [at time delay interval]</i>	Programa un reinicio del equipo en el tiempo indicado.
<i>display interface description</i>	Muestra la descripción de cada interfaz.
<i>display ip interface brief</i>	Muestra un resumen de las direcciones IP configuradas en las interfaces.
<i>display users all</i>	Muestra el usuario que está conectado al equipo y a través de cuál puerto, puede ser consola o líneas VTY.
<i>display arp</i>	Muestra la tabla ARP.
<i>display mac-address</i>	Muestra la tabla de direcciones MAC.
<i>display ip routing-table</i>	Muestra la tabla de enrutamiento global.

Fuente: elaboración propia

2.4. Ethernet

El protocolo Ethernet fue creado en 1972 por el grupo DIX conformado por Digital, Intel y Xerox. Diez años después el protocolo evoluciona a Ethernet II el cual se utiliza en la pila de protocolos TCP/IP. Para el año 1983 el IEEE creó la norma IEEE 802.3 basado en Ethernet II cuyo propósito es estandarizar la capa física y la capa de enlace de datos del modelo de referencia OSI. Hoy en día el término Ethernet hace referencia al estándar IEEE 802.2 y 802.3 que rigen la comunicación en redes LAN y WAN. Es de esta forma que se introducen nuevos conceptos como los dominios de colisión, dominios de Broadcast, direcciones MAC, tramas, entre otros.

2.4.1. Dominio de colisión

Un dominio de colisión es una zona donde los paquetes de datos pueden colisionar y destruirse. En un escenario donde existen dos computadoras conectadas entre sí por un solo cable, existe un único dominio de colisión debido a que si cualesquiera de los dos *hosts* envían información al mismo tiempo el paquete chocará en el medio y dejará inhabilitada la conexión momentáneamente.

Si se utiliza un *hub* como concentrador de red para conectar más dispositivos ocurrirá el mismo resultado: se creará un único dominio de colisión que abarcará toda la topología. Esto es debido a que el *hub* es un dispositivo que trabaja en la capa física del modelo OSI y esencialmente centraliza todos los cables conectados a él.

La situación cambia si se utiliza un *switch* como concentrador. Los *switches* trabajan en la capa de enlace de datos del modelo OSI, y tienen la habilidad de

crear un dominio de colisión por cada puerto. De esta forma si un *switch* cuenta con 24 puertos podrá crear 24 dominios de colisión y los eventos que ocurran en cada dominio afectarán únicamente de forma local sin perturbar la topología completa. Resulta evidente, entonces, que mientras más dominios de colisión existan será mejor ya que se segmenta la inestabilidad en pequeños dominios.

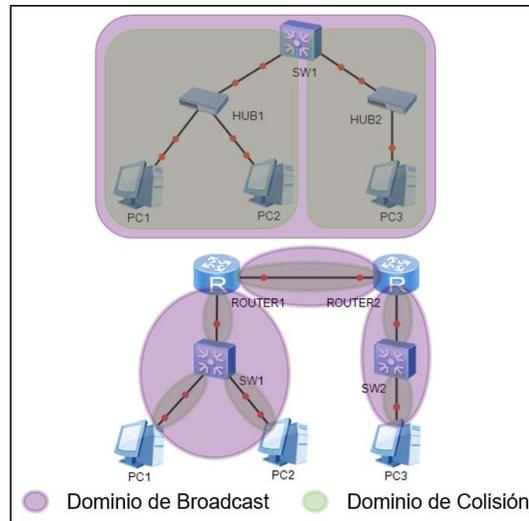
2.4.2. Dominio de Broadcast

Un Broadcast es un mensaje que se envía a todos los dispositivos pertenecientes a segmento de red. La comunicación Broadcast puede ocurrir en la capa de enlace de datos o en la capa de red del modelo OSI y resulta necesario en ARP y protocolos de enrutamiento dinámico.

En redes Ethernet se da un fenómeno llamado tormenta de Broadcast que ocurre cuando los dispositivos miembros de una red envían una gran cantidad de mensajes Broadcast y saturan el ancho de banda disponible causando que la comunicación sea inestable. Para evitar este problema es recomendable dividir una gran red en pequeños segmentos para que la inestabilidad quede confinada en su interior, cada segmento equivale a un dominio de Broadcast. Para separar o segmentar una red se utiliza *subnetting* y dispositivos de capa tres, como *routers*, *switches* multicapa o implementando VLAN.

En la figura 24 se muestran dos topologías donde se puede diferenciar el dominio de colisión del dominio de Broadcast.

Figura 24. **Dominios de colisión y dominio de Broadcast**



Fuente: elaboración propia, empleando eNSP.

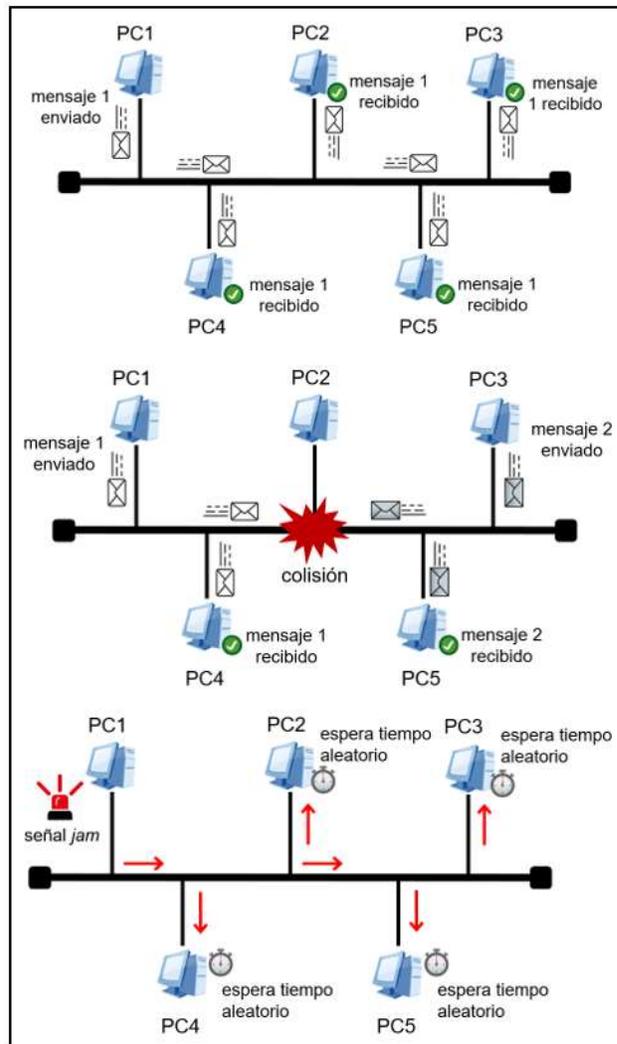
2.4.3. CSMA/CD

Es un conjunto de técnicas que se emplean en un medio de transmisión compartido y permite la detección de colisiones. Las iniciales *CS* se refieren a *Carrier Sense* que indica que el dispositivo conectado al medio tiene la habilidad de detectar cuando el canal de transmisión está ocupado por otras señales, las iniciales *MA* se refieren a *Multiple Acces* que indica que el medio puede ser compartido por múltiples dispositivos y, finalmente, las iniciales *CD* se refieren a *Collision Detection* que indica la habilidad del dispositivo para detectar colisiones en el medio.

Cuando múltiples dispositivos se encuentran conectados a una red Ethernet que implementa CSMA/CD, antes de establecer comunicación ocurre el siguiente procedimiento:

- El *host* que requiera transmitir información verificará si el medio se encuentra disponible. Cabe mencionar que todos los equipos de la red tienen la misma prioridad a la hora de enviar información.
- Si el medio se encuentra libre el *host* iniciará la transmisión de datos, de lo contrario esperará hasta que se encuentre libre.
- Mientras el *host* se encuentre enviando información monitoreará constantemente el canal para asegurarse de que sea el único que esté transmitiendo. Si llegara a detectar alguna otra señal ocurrirá una colisión y procederá a informar a todos los demás dispositivos de la red que detengan el envío de datos, esto se logra enviando una señal denominada *jam*.
- Los otros *hosts* que reciban esta señal *jam* esperarán un período de tiempo aleatorio antes de intentar transmitir nuevamente. De esta forma dos *hosts* no podrán transmitir información al mismo tiempo. Es importante recordar que CSMA/CD no evita las colisiones, simplemente las detecta y reacciona a ellas. En la figura 25 se muestra ilustrado el funcionamiento de CSMA/CD.

Figura 25. **Funcionamiento de CSMA/CD**



Fuente: elaboración propia, empleando eNSP.

2.4.4. **Comunicación Half- y Full-Dúplex**

Cuando varios *hosts* utilizan el mismo canal para comunicarse las colisiones son frecuentes y normales debido a que todos los equipos necesitan enviar información, sin embargo, esto afecta el rendimiento de la red porque se debe esperar a que el canal se encuentre libre para usarlo. Es así que únicamente un

equipo puede transmitir a la vez mientras los otros reciben, a este tipo de comunicación se le denomina Half-Dúplex.

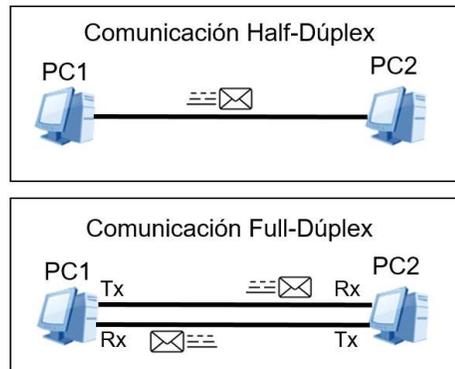
2.4.4.1. Half-Dúplex

La comunicación Half-Dúplex existe cuando hay un único canal de comunicación que une a dos o más *hosts*. En este caso los paquetes de datos sólo pueden enviarse en una dirección, es decir sólo un *host* puede estar enviando información a la vez. En un dispositivo trabajando en Half-Dúplex se necesita un solo canal de comunicación y las colisiones son completamente normales. Un ejemplo de comunicación Half-Dúplex es cuando dos equipos se comunican a través de un único cable de cobre.

2.4.4.2. Full-Dúplex

En la comunicación Full-Dúplex se necesitan dos canales de comunicación, uno de recepción y otro de transmisión. Esto permite que un *host* pueda enviar y recibir datos al mismo tiempo, mejorando así la eficiencia de la comunicación. En un entorno Full-Dúplex no existen las colisiones ya que los paquetes que viajan en direcciones opuestas no interactúan entre sí. Un ejemplo de comunicación Full-Dúplex es la comunicación serial ya que utiliza dos cables de cobre, uno para emisión, Tx, y otro para recepción, Rx, de datos.

Figura 26. **Half- y Full-Dúplex**



Fuente: elaboración propia, empleando eNSP.

2.4.5. Ethernet en la capa de enlace de datos

En la capa de enlace de datos Ethernet se divide en dos: la subcapa de Control de Enlace Lógico, LLC por sus siglas en inglés, estandarizada en la norma IEEE 802.2, y la subcapa de Control de Acceso al Medio, MAC por sus siglas en inglés, está estandarizada en la norma IEEE 802.3. La subcapa LLC se encarga de identificar el protocolo usado en la capa de red y pasar la información hacia la capa de enlace de datos, mientras que la subcapa MAC se encarga de encapsular la información, crear las tramas y controlar el acceso al medio.

Figura 27. **Modelo OSI y estándares Ethernet**



Fuente: elaboración propia, empleando eNSP.

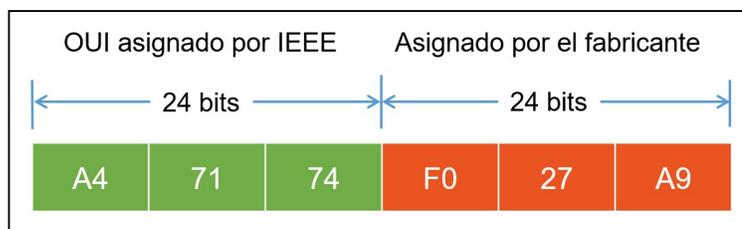
2.4.5.1. Direccionamiento Ethernet

El direccionamiento en la capa de enlace de datos tiene como principal función identificar los equipos a nivel local, es decir, equipos que pertenezcan al mismo segmento de red. Para que un *host* pueda conectarse a una red Ethernet se necesita una tarjeta de interfaz de red o NIC que se encargue de convertir la información digital a variaciones de alguna magnitud física para enviarla a través del medio de transmisión.

Cada tarjeta NIC tiene una dirección única que la identifica y viene quemada en la placa con el propósito de no ser modificada. Esta dirección se denomina MAC, su nombre deriva de la subcapa de Control de Acceso al Medio, y está escrita en formato hexadecimal contando con 48 caracteres, 48 bits o 6 Bytes.

Los primeros 24 bits corresponden al identificador del fabricante, OUI, y es asignado por el IEEE. Cada fabricante tiene su propio OUI con el cual identifica los equipos que produce. Los últimos 24 bits de la dirección MAC corresponden a un número correlativo que el fabricante asigna a sus equipos a medida que estos son fabricados. La duplicación de direcciones MAC es poco probable pero posible, y en esos casos remotos se puede cambiar la dirección de forma lógica para evitar problemas.

Figura 28. Dirección MAC



Fuente: elaboración propia.

La dirección MAC es una dirección física y se utiliza como origen o destino para enviar o recibir información de forma local. Existen direcciones MAC específicas para enviar mensajes en modo Broadcast o Multicast, es importante recalcar que estas direcciones solo pueden ser de destino, nunca de origen.

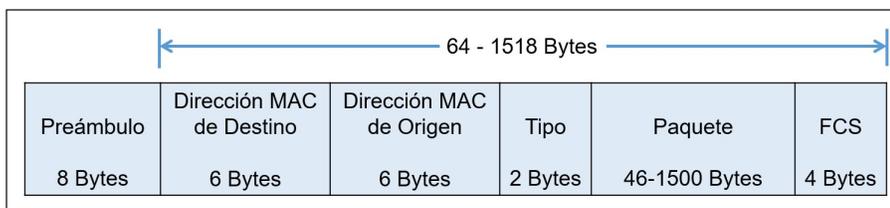
- Dirección MAC de Broadcast: FF-FF-FF-FF-FF-FF
- Dirección MAC de Multicast: 01-00-5E-XX-XX-XX

En este caso las Xs de la dirección Multicast pueden ser cualquier valor hexadecimal.

2.4.5.2. Tramas Ethernet

La trama es la unidad de datos que se utiliza en la capa de enlace de datos y dentro de ella se encapsula toda la información necesaria para que los datos puedan llegar a su destino. Evidentemente una trama solo tiene alcance local ya que maneja direcciones MAC por lo que el destino tiene que estar en el mismo segmento de red que el origen.

Figura 29. Trama Ethernet



Fuente: elaboración propia.

La estructura de una trama está conformada de 7 campos como se describe:

- El preámbulo, 7 Bytes, que contiene a su vez el campo SFD, 1 Byte, y ambos se encargan de sincronizar al emisor y receptor y los prepara para intercambiar datos.
- La dirección MAC de destino corresponde a la dirección MAC de la interfaz del *host* receptor. Esta puede ser una dirección individual, de Broadcast o Multicast.
- La dirección MAC de origen corresponde a la dirección MAC de la interfaz del *host* emisor. Esta dirección siempre es individual.
- El campo tipo es un número que identifica al protocolo de capa de red que se está utilizando, ejemplos de este identificador pueden ser: 0x800 para IPv4, 0x86DD para IPv6, 0x806 para ARP o 0x8848 para MPLS.
- El paquete es la unidad de datos de la capa de red y en él está contenida la información o la carga. El paquete puede variar entre los 64 hasta los 1 500 bytes.
- El campo FCS, Frame Check Sequence, está diseñado para detectar errores en la trama. Utiliza los CRC, como contadores que aumentan si algún bit de la trama fue modificado durante el transporte. Normalmente los CRC aumentan debido a mala negociación entre dos puertos o perturbaciones en el medio.

2.4.6. Ethernet en la capa física

La primera versión de Ethernet creada en 1972 funcionaba a una velocidad de 2,85 Mbps sobre un cable coaxial en topología de bus. Tiempo después se

desarrolló el cable de par trenzado que evitaba la interferencia por inducciones en el cable de cobre y se logró llegar a velocidades de 10 Mbps. A medida que la tecnología evolucionaba y se desarrollaban velocidades mayores y otros tipos de medios de transmisión, como la fibra óptica. Ethernet fue acoplado estos nuevos avances para mejorar su rendimiento. En la tabla III, se muestra una lista de los estándares Ethernet más usados en la actualidad.

Tabla III. **Estándares Ethernet**

Nombre Común	Velocidad	Nombre Alternativo	Estándar IEEE	Tipo de Cable
Ethernet	10 Mbps	10 Base-T	802,3	Cobre
Fast Ethernet	100 Mbps	100 Base-TX	802,3u	Cobre
Fast Ethernet	100 Mbps	100 Base-FX	802,3u	Fibra
Gigabit Ethernet	1 Gbps	1 000 Base-LX	802,3z	Fibra
Gigabit Ethernet	1 Gbps	1 000 Base-T	802,3ab	Cobre
10Gigabit Ethernet	10 Gbps	10 GBase-T	802,3an	Cobre

Fuente: elaboración propia.

2.4.6.1. **Cableado Ethernet**

El protocolo Ethernet puede ser implementado en topologías de cobre o fibra óptica, cada material tiene ventajas y desventajas y su elección dependerán de diversos factores. Los cables de cobre son más económicos y fáciles de instalar, pero tienen como desventaja la limitada velocidad y la longitud máxima del cable. Los cables de fibra óptica son más costosos y de complicada instalación, pero la velocidad de transmisión es mucho mayor, además de ser más eficientes para largas distancias ya que no sufren de inducción electromagnética ni pérdidas por efecto Joule.

Los cables de cobre que utiliza el protocolo Ethernet pueden ser de tipo UTP y estar conformados por cuatro pares de hilos de cobre trenzados y protegidos por un material aislante. Cada hilo de cobre está identificado con un patrón de colores que ayuda a la instalación con los conectores. Existen tres tipos de cables de par trenzados:

- Cable de Par Trenzado Sin Blindaje o UTP. Utiliza conectores RJ-45.
- Cable de Par Trenzado con Blindaje Individual o STP. Cuenta con una lámina conductora que cubre cada par de hilos que cumple la función blindarlos frente a inducciones externas. Requiere una puesta a tierra para desviar las inducciones por lo que utiliza un conector especial tipo RJ-49.
- Cable de Par Trenzado con Blindaje Total o FTP. Cuenta con una lámina conductora que cubre los ocho hilos y cumple la función de blindar el cable. Es menos eficiente para proteger contra inducciones que el cable STP. Emplea conectores RJ-45.

El cable UTP es usado en entornos donde no exista mayor ruido electromagnético, mientras que los cables STP y FTP son usados en entornos industriales. Para los tres casos la longitud máxima del cable es de 100 metros y su velocidad máxima de transferencia dependerá del calibre de los hilos de cobre, que se clasifican por categoría. Cabe mencionar que la velocidad de transferencia, que se mide en bits por segundo, también es conocida como ancho de banda o BW.

En la tabla IV, se muestra el ancho de banda para cada tipo de categoría de cable:

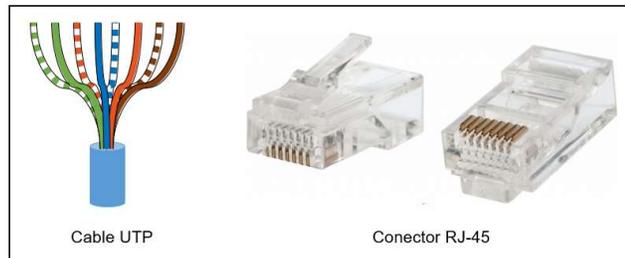
Tabla IV. **Ancho de banda y longitud para cables UTP y fibra óptica**

Ethernet	Ancho de Banda	Tipo de Cable	Distancia Máxima
10 Base-T	10 Mbps	Cat3/Cat5 UTP	100 m
100 Base-TX	100 Mbps	Cat5 UTP	100 m
100 Base-TX	200 Mbps	Cat5 UTP	100 m
100 Base-FX	100 Mbps	Fibra Multimodo	400 m
100 Base-FX	200 Mbps	Fibra Multimodo	2 Km
1 000 Base-T	1 Gbps	Cat5e UTP	100 m
1 000 Base-TX	1 Gbps	Cat6 UTP	100 m
1 000 Base-SX	1 Gbps	Fibra Multimodo	550 m
1 000 Base-LX	1 Gbps	Fibra Monomodo	2 Km
10 GBase-T	10 Gbps	Cat6a/Cat7 UTP	100 m
10 GBase-SX4	10 Gbps	Fibra Multimodo	550 m
10 GBase-LX4	10 Gbps	Fibra Monomodo	2 Km

Fuente: elaboración propia.

El cable, también llamado *patchcord*, más utilizado es el UTP con conector RJ-45, sin embargo, su forma de acople es similar para el conector RJ-49. El conector RJ-45 cuenta con ocho pines, uno por cada hilo de cobre del cable que se conecta en un orden en específico gracias al patrón de colores con el que cuenta cada hilo.

Figura 30. **Cable UTP y conector RJ-45**



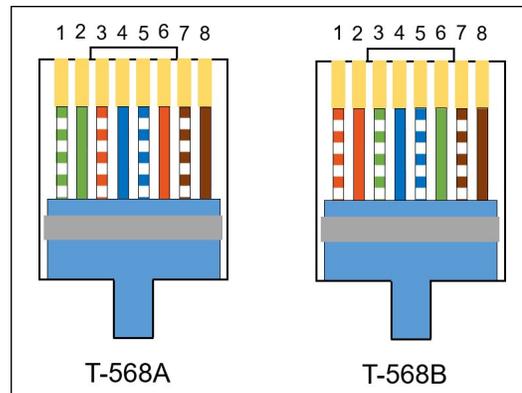
Fuente: elaboración propia.

Cada pin del conector RJ-45 tiene una función en específico que depende del tipo de dispositivo al que se conecte el cable. Por ejemplo, las tarjetas de Red de las computadoras, *routers*, *Wireless AP* y demás *hosts* finales emplean los pines 1 y 2 para transmitir datos, Tx, y los pines 3 y 6 para recibir, Rx. Caso contrario con los concentradores de red como los *hubs* y *switches* que utilizan los pines 3 y 6 para transmitir y los pines 1 y 2 para recibir.

En resumen, los pines 1, 2, 3 y 6 del conector RJ-45 se utilizan para el envío y recepción de información cuando el ancho de banda es menor a 1 Gbps. Cuando el ancho de banda es igual o superior al 1 Gbps se utilizan los 8 pines para el envío y recepción de datos.

Evidentemente para enviar datos de un dispositivo a otro se necesita conectar el pin Tx con el pin Rx, para cumplir este propósito podría utilizarse cualquier hilo de cobre del cable UTP. Sin embargo, el conjunto de estándares TIA/EIA-568-B define dos formas de conexión: T-568A y T-568B, como se observa en la figura 31.

Figura 31. **Conexión T-568A y T-568B**



Fuente: elaboración propia.

Es con estos estándares que se logran crear cables directos o cruzados para comunicar los equipos.

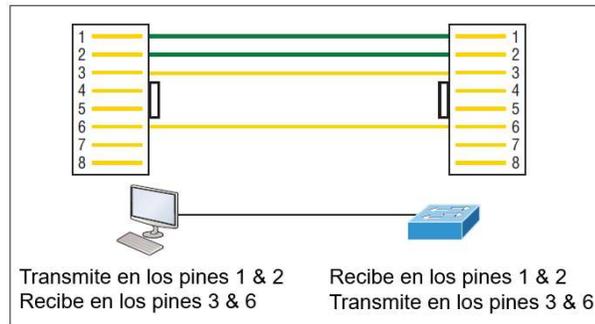
2.4.6.1.1. **Cable directo**

El cable directo se construye utilizando la misma conexión en ambos extremos, ambos pueden ser T-568A o T-568B. Este tipo de cable es el más común y se utiliza para conectar los siguientes equipos:

- *Hosts con switches*
- *Hosts con hubs*
- *Routers con switches*
- *Routers con hubs*

En la figura 32, se muestra la forma en la que funciona el cable directo.

Figura 32. **Cable directo**



Fuente: LAMMLE, Todd. *CCNA, Routing and Switching, Study Guide*. p. 60.

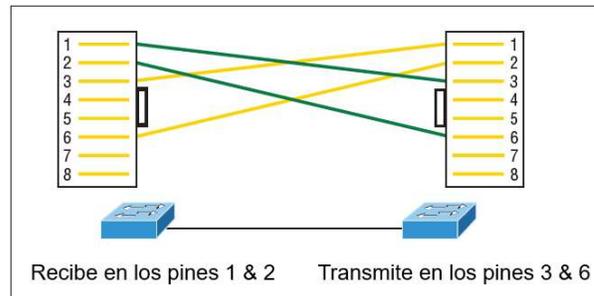
2.4.6.1.2. **Cable cruzado**

El cable cruzado se construye utilizando en un extremo del cable la conexión T-568A y en el otro T-568B. Este tipo de cable se utiliza para conectar los siguientes equipos:

- *Switches con switches*
- *Hubs con hubs*
- *Hosts con hosts*
- *Hubs con switches*
- *Routers directamente con hosts*
- *Routers con routers*

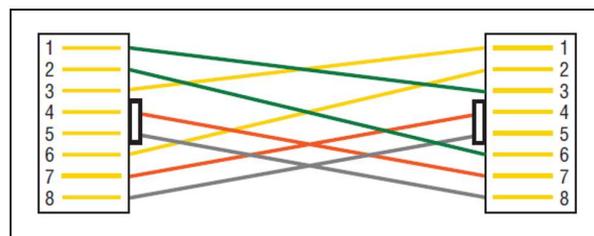
En la figura 33, se muestra la forma en la que funciona el cable cruzado.

Figura 33. **Cable cruzado**



Fuente: LAMMLE, Todd. *CCNA, Routing and Switching, Study Guide*. p. 61.

Figura 34. **Tipo de conexión para cables de 1Gbps de ancho de banda**



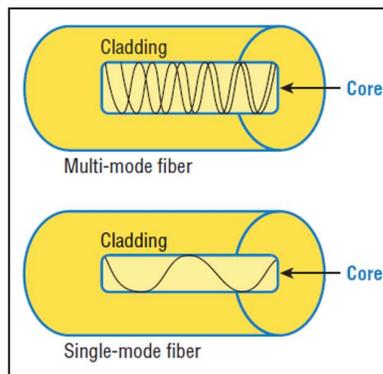
Fuente: LAMMLE, Todd. *CCNA, Routing and Switching, Study Guide*. p. 62.

2.4.6.1.3. **Cable de fibra óptica**

Otro tipo de cableado empleado con el protocolo Ethernet es el de fibra óptica, y puede ser multimodo o monomodo. En los cables multimodo se pueden enviar varios haces de luz sobre el mismo medio, técnica que se conoce como Multiplexación por División de Longitud de Onda o WDM, esto permite enviar varias portadoras y aumentar la tasa de transferencia de datos a costa de reducir la longitud máxima del cable. Por el contrario, los cables de fibra monomodo son capaces de enviar únicamente un haz de luz, disminuyendo así su tasa de transferencia de datos, pero aumentando la longitud máxima de los hilos de fibra.

Ambos tipos de cables de fibra óptica están diseñados de forma similar: cuentan con un núcleo por donde viaja el haz de luz y una capa que lo rodea llamada *cladding* que evita que el haz de luz escape del núcleo.

Figura 35. **Estructura del cable de fibra óptica**



Fuente: LAMMLE, Todd. *CCNA, Routing and Switching, Study Guide*. p. 65.

Como convención general los cables monomodo vienen con forro color amarillo y los cables multimodo con forro color naranja. Ambos utilizan distintos tipos de conectores, los más comunes son: Ferrule Connector o FC, Straight Tip o ST, Lucent Connector o LC, y Suscriptor Connector o SC.

En la figura 36 se muestran los distintos tipos de conectores para fibra óptica.

Figura 36. **Conectores para fibra óptica**



Fuente: PROMAX. *Tipos de conectores de fibra óptica.*

www.promax.es/esp/noticias/578/tipos-de-conectores-de-fibra-optica-guia-sencilla/.

Consulta: junio de 2020.

2.4.6.1.4. Transceptores SFP

Los transceptores o *transceivers SFP* son módulos que reciben información por pulsos eléctricos de un dispositivo y se encargan de convertirlos a variaciones de ondas electromagnéticas para enviarlos a través de cables de fibra óptica. Existen una gran cantidad de modelos que se clasifican en base a la velocidad de transmisión, al tipo de conectores empleados, entre otras características.

En la figura 37 se muestra un *transceiver SFP* con longitud de onda de 1 310 nm y con capacidad de transmitir hasta 2 km.

Figura 37. **Transceiver SFP**



Fuente: FS. *Generic Compatible 1000BASE-SX SFP 1310nm 2km DOM Transceiver Module*. www.fs.com/products/48928.html. Consulta: junio de 2020.

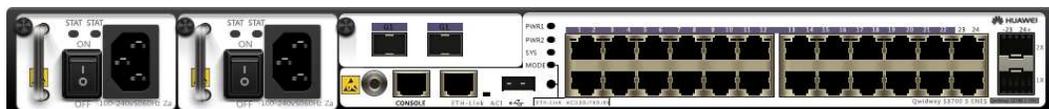
2.4.7. **Switches Ethernet**

Un *switch* Ethernet es un concentrador de red cuya función principal es conmutar o reenviar las tramas Ethernet hacia su destino basándose en la dirección MAC.

Huawei dispone de una gran cantidad de modelos de *switches* con diferentes prestaciones, desde el *switch* S3700 que cuenta con 24 puertos GigabitEthernet hasta el *switch* ATN 910 C que se utiliza frecuentemente en la infraestructura de ISP.

En la figura 38 se pueden observar los puertos disponibles en un *switch* S3700.

Figura 38. **Parte trasera de un switch Huawei S3700**



Fuente: elaboración propia, empleando eNSP.

En general, todos los tipos de *switches* se basan en la dirección MAC de origen y destino, el puerto por donde ingresa la trama y el puerto que dirige hacia el destino para conmutar las tramas.

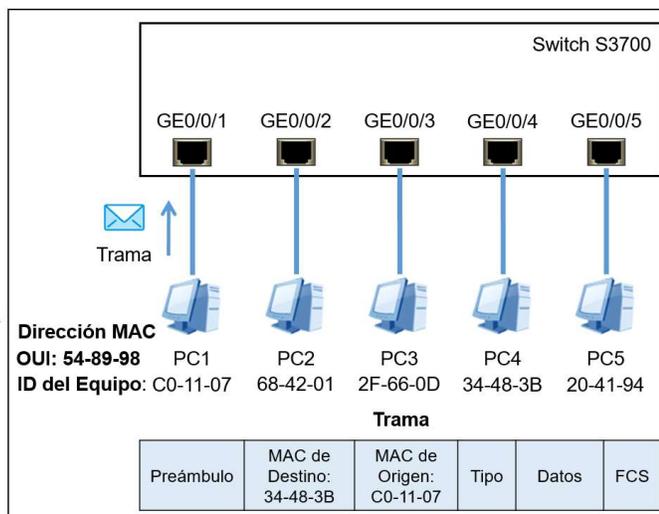
2.4.7.1. Principio de operación

Para comprender de forma clara el principio de operación de un *switch*, se toma como ejemplo una topología con cinco computadoras conectadas a los puertos GigabitEthernet de un *switch* S3700. Para fines ilustrativos se omite el resto de puertos del *switch* y se representan únicamente los últimos tres Bytes de la dirección MAC que corresponden al ID del dispositivo.

Inicialmente, el *switch* S3700 no cuenta con información almacenada en la memoria por lo que desconoce qué equipos se encuentran conectados a él. A medida que las tramas comienzan a ingresar por los puertos se crea un mapa que se almacena en la memoria RAM.

Para este ejemplo la PC1 envía información hacia la PC4, esto se traduce a una trama que sale de la PC1 con dirección MAC de origen: C0-11-07 y se dirige hacia la dirección MAC de destino 34-48-3B. Cabe mencionar que este ejemplo únicamente busca explicar la operación de un *switch*, en un entorno real el equipo emisor no conoce con antelación la dirección MAC de destino y se requiere del apoyo de ARP, el cual se abarcará más adelante.

Figura 39. Trama ingresando al *switch* S3700 rumbo a PC4



Fuente: elaboración propia, empleando eNSP.

La primera acción que realiza el *switch* cuando la trama ingresa por el puerto es asociar la dirección MAC del emisor y el puerto por el que se recibe la trama. De esta forma el *switch* sabe que a través del puerto GE0/0/1 se puede alcanzar el *host* con dirección MAC: C0-11-07. Esta información se almacena en una tabla llamada tabla de direcciones MAC alojada en la memoria RAM. A grandes rasgos, la tabla de direcciones MAC es un mapa que indica el puerto por el que se puede alcanzar una dirección MAC.

En la figura 40, se puede observar la primera entrada a la tabla de direcciones MAC del *switch* S3700.

Figura 40. **Primera entrada de la tabla de direcciones MAC**

```
<S5700> display mac-address
MAC address table of slot 0:
-----
MAC Address      VLAN/      PEVLAN CEVLAN  Port          Type          LSP/LSR-ID
                  VSI/SI
-----
5489-98c0-1107  1          -      -      GE0/0/1      dynamic      0/-
-----
Total matching items on slot 0 displayed = 1
<S5700>
```

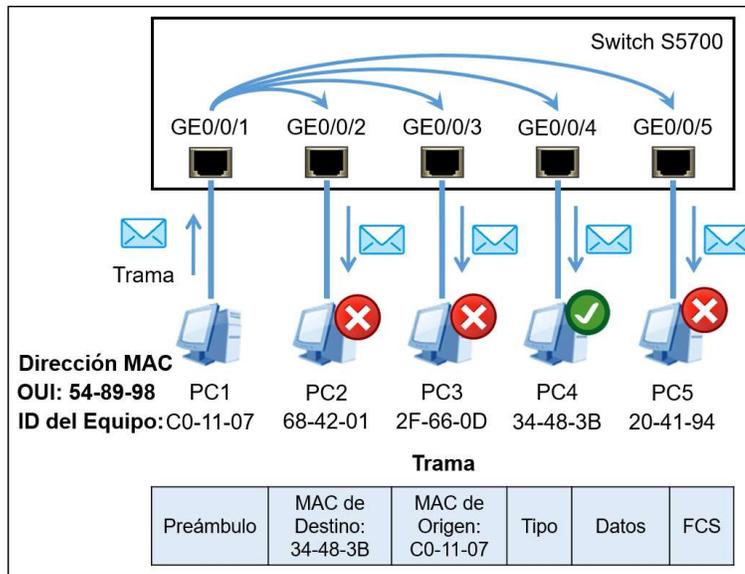
MAC Address	VLAN/ VSI/SI	PEVLAN	CEVLAN	Port	Type	LSP/LSR-ID MAC-Tunnel
5489-98c0-1107	1	-	-	GE0/0/1	dynamic	0/-

Fuente: elaboración propia, empleando eNSP.

Una vez recibida la trama, el *switch* analiza la dirección MAC de destino y consulta en la tabla de direcciones MAC para ver si conoce algún puerto por el cual llegar a ella. Para este ejemplo, el *switch* busca en la tabla de direcciones MAC si conoce algún puerto para llegar al *host* con la dirección MAC: 34-48-3B. En este caso, el *switch* no conoce qué puerto emplear para alcanzar el *host* PC4 ya que no lo ha aprendido.

Cuando el *switch* no conoce algún puerto para llegar a una dirección MAC recurre a una operación llamada *flooding* o inundación, que consiste en reenviar la trama en todos los puertos excepto por el que la recibió. En términos del ejemplo, la trama se reenvía por los puertos GE0/0/2, GE0/0/3, GE0/0/4 y GE0/0/5 pero no por el GE0/0/1. Los *hosts* conectados a estos puertos reciben la trama y verifican si la dirección MAC de destino es igual a la suya, si no es igual descartan la trama, si son iguales significa que la trama alcanzo su destino.

Figura 41. Ejemplo de *flooding* en un *switch*



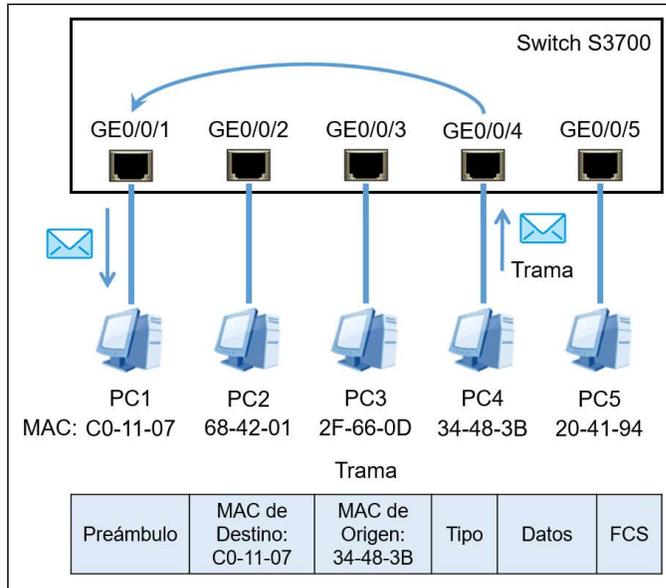
Fuente: elaboración propia, empleando eNSP.

El *host* PC4 recibe la trama y valida que su dirección MAC es igual a la dirección MAC de destino de la trama, lo que indica que la trama está dirigida para él.

Posteriormente, PC4 envía una respuesta dirigida hacia la PC1 construyendo una nueva trama con dirección MAC de destino: C0-11-07 y dirección MAC de origen: 34-48-3B. Esta nueva trama ingresa por el puerto GE0/0/4 del *switch* y el proceso se repite: el puerto y la dirección MAC del *host* emisor se guardan en la tabla de direcciones MAC creando así una segunda entrada.

En la figura 42 se observa la nueva trama ingresando por el puerto GE0/0/4 y proveniente del *host* PC4, mientras que en la figura 43 se observa la nueva entrada en la tabla de direcciones MAC del *switch* S3700.

Figura 42. Nueva trama dirigida hacia PC1



Fuente: elaboración propia, empleando eNSP.

Figura 43. Segunda entrada de la tabla de direcciones MAC

```

SW1
<S5700> display mac-address
MAC address table of slot 0:
-----
MAC Address      VLAN/      PEVLAN  CEVLAN  Port          Type      LSP/LSR-ID
                  VSI/SI
-----
5489-9834-483b  1          -        -        GE0/0/4       dynamic   0/-
5489-98c0-1107  1          -        -        GE0/0/1       dynamic   0/-
-----
Total matching items on slot 0 displayed = 2
<S5700>
  
```

Fuente: elaboración propia, empleando eNSP.

Como siguiente paso el *switch* revisa su tabla de direcciones MAC para verificar si conoce cómo alcanzar la dirección MAC: C0-11-07 que, dada la primera entrada, se sabe que se conoce a través del puerto GE0/0/1.

Finalmente, la trama es reenviada únicamente al puerto GE0/0/1 alcanzando al *host* de destino.

2.4.7.2. Tabla de direcciones MAC

En general, la tabla de direcciones MAC es un mapa que indica el puerto por el que se puede alcanzar una dirección MAC. Para visualizarla en un *switch* debe ingresar a la vista del usuario y emplear el comando: *display mac-address*. Es importante mencionar que un puerto puede conocer varias direcciones MAC cuando se implementan VLAN, puertos troncales o *hubs* funcionando como concentradores de red. La tabla de direcciones MAC es dinámica, lo que se traduce a que puede aprender nuevas entradas y eliminar las que no están en uso, sin embargo, también se pueden configurar entradas estáticas que restringe el puerto por el que se puede recibir tramas con una dirección MAC específica. Estas entradas estáticas, a diferencia de las dinámicas, siempre permanecen en la tabla. Para crear una entrada estática se debe ingresar a la vista del sistema y emplear el comando: *mac-address static mac_address interface_type interface_number vlan vlan_id*, donde el parámetro *mac_address* indica la dirección MAC de origen de la trama, el parámetro *interface_type* e *interface_number* indican el puerto por el que se debe recibir la trama y el parámetro *vlan_id* indica la VLAN asociada a la dirección MAC.

Por defecto, una entrada dinámica que no esté en uso tiene un tiempo de expiración de 300 segundos antes de ser eliminada de la tabla, este tiempo es reiniciado cuando se actualice la entrada nuevamente. El tiempo de expiración es configurable a conveniencia y puede ser modificado desde la vista del sistema con el comando: *mac-address aging-time aging-time*, donde el parámetro *aging-time* especifica el tiempo de expiración, que es un valor entero entre 60 a 1 000 000 de segundos.

Una función especial de los *switches* Huawei es la creación de entradas *blackhole* en la tabla de dirección MAC. Una entrada *blackhole* tiene la función de descartar cualquier trama que provenga o se dirija hacia una dirección MAC en específico. Para crear una entrada de tipo *blackhole* se debe ingresar a la vista del sistema y emplear el comando: `mac-address blackhole mac_address [vlan vlan_id]`, donde el parámetro *mac_address* es la dirección MAC de origen o destino de la trama que se desee descartar y el parámetro opcional *vlan_id* es la VLAN asociada a la dirección MAC.

2.4.8. Encapsulación de datos

Cuando un *host* envía un mensaje a través de una red se adjunta información adicional, por ejemplo, direcciones IP, direcciones MAC, número de puerto, entre otros, con el propósito de que los dispositivos de red intermedios puedan conducir el mensaje hacia su destino final. A este proceso se le denomina encapsulación.

El proceso de encapsulación recorre las siete capas del modelo OSI iniciando en la capa de aplicación donde se interactúa con las aplicaciones del usuario y se obtiene la información, seguidamente pasa a la capa de presentación donde se le da formato para que la aplicación del usuario pueda interpretarla y finalmente pasa a la capa de sesión responsable de mantener separada la sesión de comunicación para que no exista conflicto con otras sesiones. En cada una de las tres capas se adjunta información adicional que aumenta el tamaño de mensaje y recibe el nombre de unidad de datos del protocolo o PDU. Para las tres primeras capas la PDU se denomina datos y la información que se adjunta en cada una de ellas escapa del alcance del presente trabajo.

Cuando los datos llegan a la capa de transporte se adjunta un encabezado o *header*, creando así una nueva PDU llamada segmento si se está empleando TCP o datagrama en el caso de UDP. El contenido del *header* de un segmento es distinto al del datagrama ya que el primero debe llevar información para crear una conexión y por lo tanto tiene un mayor tamaño. Generalmente el *header* de un segmento puede tener un tamaño de 20 a 21 Bytes mientras que el *header* de un datagrama tiene un tamaño fijo de 8 Bytes.

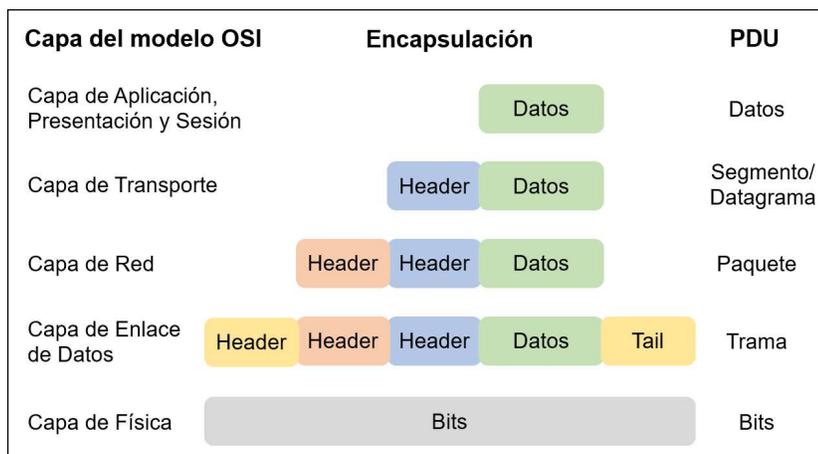
Luego de pasar por la capa de transporte, la PDU se dirige hacia la capa de red. En esta capa se adjunta un nuevo *header* y el segmento o datagrama se convierte en un paquete. El contenido de este nuevo *header* dependerá del protocolo de capa de red empleado: IPv4, IPv6, ICMP, entre otros. El *header* de un paquete IPv4 tiene un tamaño de 20 Bytes mientras que el *header* de un paquete IPv6 tiene un tamaño de 40 Bytes. Ambos *headers* tienen un campo de 16 bits llamado *total length* que indica el tamaño total del paquete y que, en teoría, puede llegar a ser de 65 535 Bytes.

Sin embargo, en la práctica se utilizan distintos tamaños dependiendo del tipo de protocolo de enlace de datos, por ejemplo, el protocolo Ethernet acepta un paquete máximo de 1 500 Bytes. Este parámetro también es conocido como MTU, *Maximum Transmission Unit*, y puede ser configurado a conveniencia restringiendo así la cantidad de información que puede ser enviada en cada paquete, este proceso se conoce como fragmentación de la información y es muy importante cuando se implementan protocolos de enrutamiento dinámicos.

La encapsulación continúa cuando el paquete llega a la capa de enlace de datos donde existen protocolos como Ethernet, Frame Relay, ATM, Point-to-Point Protocol o PPP, entre otros. En el caso de Ethernet se agrega un nuevo *header* y una cola o *tail* al paquete para convertirlo en una trama.

El *header* está constituido por los campos: preámbulo, dirección MAC de destino, Dirección MAC de origen y tipo, mientras que la cola está conformada por el campo FCS. La trama puede tener un tamaño mínimo de 64 Bytes y máximo de 1 518 Bytes, se aumenta a 1 522 Bytes al implementar VLAN.

Figura 44. **Proceso de encapsulación**



Fuente: elaboración propia, empleando eNSP.

Finalmente, la trama se traslada hacia la capa física que se encarga de convertir los bits de información, unos y ceros lógicos, en alteraciones de alguna magnitud física, como intensidad eléctrica o lumínica, para luego modularla y enviarla por el medio de transmisión. De esta forma los bits viajan a través de cables o por medios inalámbricos hacia un equipo receptor, que puede ser un *host* o un equipo de red intermedio que demodule e interprete la información.

2.4.9. Address Resolution Protocol

ARP es un protocolo de la capa de red que, a grandes rasgos, busca una dirección MAC desconocida en base a una dirección IP conocida. ARP se emplea al inicio de una transmisión debido a que el equipo emisor conoce la dirección IP

de destino, pero desconoce la dirección MAC, por lo que no podría llenar el campo correspondiente en la trama. Es así como ARP se vale de los mensajes *ARP Request* y *ARP Reply* para encontrar la dirección MAC de un equipo en base a la dirección IP de destino.

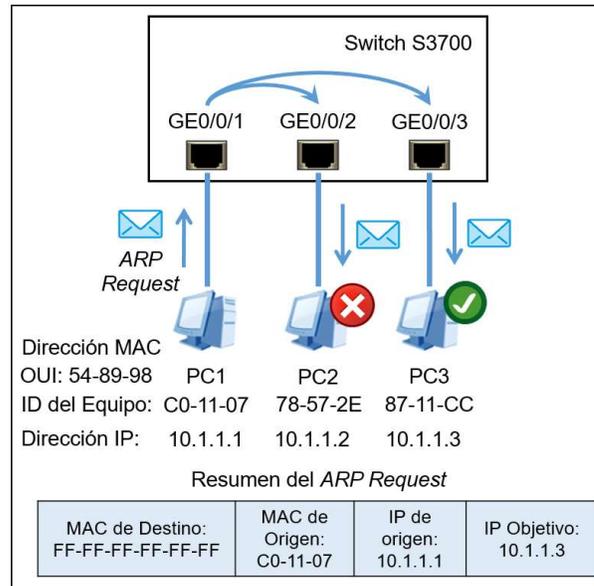
2.4.9.1. Funcionamiento

Para entender el funcionamiento de ARP se toma como ejemplo una topología donde se tienen tres computadoras conectadas a un *switch* S3700. El *host* PC1 desea enviar información hacia el *host* PC3 por lo que el paquete tendrá la dirección IP de destino: 10.1.1.3. Ahora bien, el equipo emisor conoce la dirección IP de destino, pero al intentar realizar el proceso de encapsulación cae en la cuenta que desconoce la dirección MAC de PC3 y no puede llenar el *header* de la trama.

Frente a este dilema PC1 envía antes un paquete especial llamado *ARP Request* a todos los equipos del segmento de red. Este paquete es de tipo Broadcast y tiene la dirección MAC de destino: FF-FF-FF-FF-FF-FF. Cuando el paquete ingresa por el puerto GE0/0/1, el *switch* realiza una inundación o *flooding* por todos los puertos que pertenecen al dominio, es decir GE0/0/2 y G0/0/3, exceptuando el puerto donde ingresó el paquete. El objetivo del *ARP Request* es encontrar el equipo que tenga configurada la dirección IP: 10.1.1.3.

En la figura 45 se observa el proceso de envío del mensaje *ARP Request* por PC1.

Figura 45. **ARP Request**

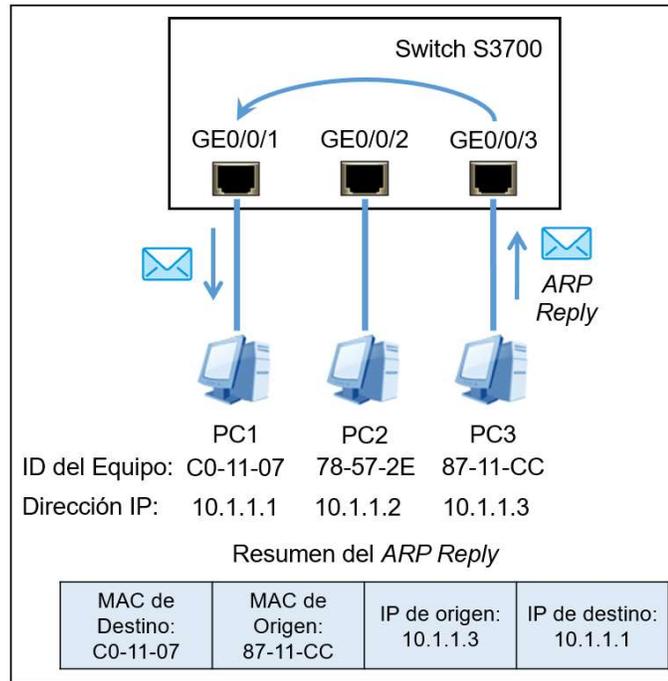


Fuente: elaboración propia, empleando eNSP.

El paquete *ARP Request* es recibido por PC2 y PC3. En el caso de PC2, la dirección IP no coincide, por lo que el mensaje es descartado. En el caso de PC3 la dirección IP sí coincide, indicando que el equipo objetivo fue encontrado. Para completar el proceso, PC3 responde con un mensaje unicast llamado *ARP Replay* dirigido hacia PC1, en dicho mensaje incluye su propia dirección MAC.

En la figura 46 se observa el procedo del *ARP Replay* hacia PC1.

Figura 46. **ARP Reply**



Fuente: elaboración propia, empleando eNSP.

Cuando el *ARP Reply* arriba a PC1 se extrae la dirección MAC de PC3 y se guarda en una tabla llamada tabla ARP, almacenada en la memoria caché del equipo. Ahora que se ejecutó ARP, PC1 ya conoce la dirección MAC de destino y puede llenar el *header* de la trama para continuar con el proceso de encapsulación.

Figura 47. **Entrada en la tabla ARP de PC1 y PC3**

PC1> arp -a		
Internet Address	Physical Address	Type
10.1.1.3	54-89-98-87-11-CC	dynamic
PC3> arp -a		
Internet Address	Physical Address	Type
10.1.1.1	54-89-98-C0-11-07	dynamic

Fuente: elaboración propia, empleando eNSP.

2.4.9.2. Tabla ARP

En general, la tabla ARP es un mapa que relaciona la dirección MAC y la dirección IP de un equipo. Para visualizarla en los *hosts* se debe ingresar a la vista del usuario y emplear el comando: *display arp* o emplear el comando: *arp -a* cuando son *hosts*. La tabla ARP se utiliza en equipos que puedan manejar direcciones IP como: *routers*, *switches* multicapa y *hosts*.

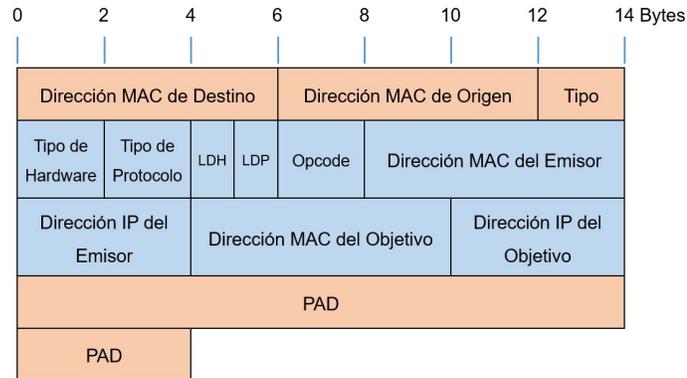
La tabla ARP es dinámica, es decir que puede aprender nuevas entradas y eliminar las que no están en uso, sin embargo, también se pueden configurar entradas estáticas que relacionarán direcciones IP con direcciones MAC. Por defecto, una entrada dinámica que no está en uso tiene un tiempo de expiración de 1 200 segundos o 20 minutos antes de ser eliminada de la tabla ARP, este tiempo es reiniciado cuando se actualiza la entrada. El tiempo de expiración es configurable a conveniencia desde la vista del sistema con el comando: *arp timeout expire-time*, donde el parámetro *expire-time* especifica el tiempo de expiración, que es un valor entero 30 a 62 640 segundos.

2.4.9.3. Formato del paquete ARP

La estructura del paquete ARP es la misma tanto para las solicitudes *ARP Request* como para las respuestas *ARP Reply*, lo único que varía es el valor de sus campos. Durante el proceso de encapsulación al paquete ARP, creando en la capa de red, se le agrega un *header* y un *PAD* para convertirlo en una trama. La trama creada cuenta con un tamaño mínimo de 64 Bytes.

En la figura 48 se muestran los campos que conforman una trama ARP, la forma de interpretarla es de derecha a izquierda y de arriba hacia abajo. Los campos que se muestran en celeste son los que conforman el paquete ARP.

Figura 48. Estructura de una trama ARP



Fuente: JIANG, Yonghong. *HCNA Networking Study Guide*. p. 95.

En la tabla V se describe cada campo que conforma la trama ARP.

Tabla V. Definición de los campos de una trama ARP

Campo	ARP Request	ARP Reply
Dirección MAC de Destino	FF-FF-FF-FF-FF-FF	Dirección MAC del Solicitante
Dirección MAC de Origen	Dirección MAC del Solicitante	Dirección MAC Solicitada
Tipo	2 Bytes. Indica el tipo de paquete. Para ARP el valor hexadecimal es 0x0806	
Tipo de Hardware	2 Bytes. Especifica el tipo de red. Para Ethernet el valor es 1	
Tipo de Protocolo	2 Bytes. Especifica el tipo de dirección de protocolo. Para mapeos basados en direcciones IP el valor hexadecimal es 0x0800	

Continuación de la tabla V.

Longitud de la Dirección de Hardware o LDH	1 Byte. Especifica la longitud de la dirección de Hardware. Para el caso de Ethernet el valor es 6, indicando que la dirección MAC tiene una longitud de 6 Bytes.	
Longitud de la Dirección del Protocolo o LDP	1 Byte. Especifica la longitud de la dirección del protocolo. Para IP el valor es 4, indicando que la dirección IP tiene una longitud de 4 bytes.	
Opcode	2 Bytes. Especifica el tipo de paquete ARP. El valor es de 1.	2 Bytes. Especifica el tipo de paquete ARP. El valor es de 2.
Dirección MAC del Emisor	Dirección MAC del Solicitante.	Dirección MAC Solicitada.
Dirección IP del Emisor	Dirección IP del Solicitante.	Dirección IP Solicitada.
Dirección MAC del Objetivo	Este campo es ignorado porque el solicitante no conoce esta Dirección MAC.	Dirección MAC del Solicitante.
Dirección IP del Objetivo	La dirección IP que el solicitante quiere mapear, misma que la dirección IP del solicitante.	Dirección IP del solicitante.
PAD	18 Bytes. El campo PAD cumple la función de relleno o <i>padding</i> para asegurar que la trama Ethernet alcance un tamaño mínimo de 64 Bytes.	

Fuente: JIANG, Yonghong. *HCNA Networking Study Guide*. p. 96.

2.4.10. Envío de datos a través de una red Ethernet

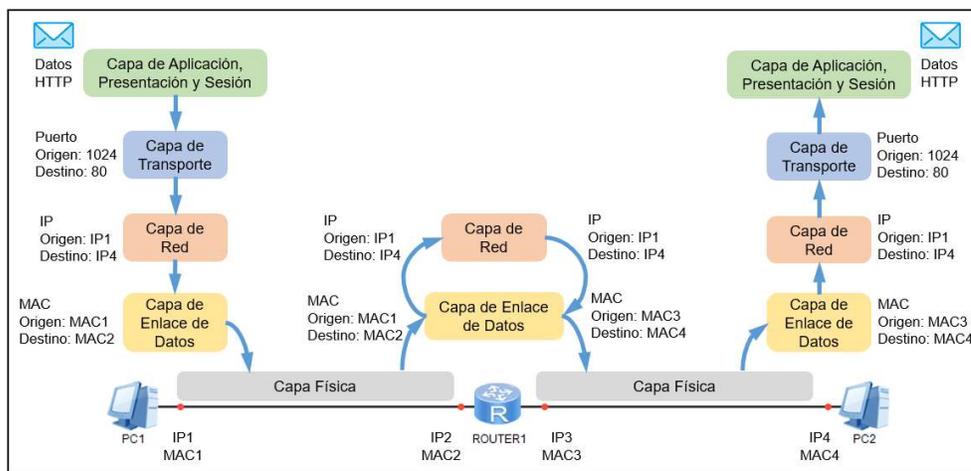
Para que la información sea enviada a través de una red, cada dispositivo intermedio debe realizar el proceso de encapsulación y desencapsulación, comprobando las direcciones IP y MAC para redireccionar el mensaje hasta su destino final.

Para el siguiente ejemplo se desea envía una solicitud HTTP desde el *host* PC1 hasta PC2. El proceso inicia en las primeras tres capas del modelo OSI donde se obtienen los datos. Luego se encapsulan en un segmento con el número de puerto de destino: 80, el cual es un puerto designado a HTTP, y el número de puerto origen: 1024, el cual es un puerto asignado de forma aleatoria comprendido entre 1024 a 65 535. Posteriormente el segmento se encapsula en un paquete IP con la dirección de origen del equipo emisor: IP1 y la dirección IP del equipo objetivo: IP4. En este punto el paquete desciende a la capa de enlace de datos donde se encapsula y forma una trama Ethernet con dirección MAC de origen del equipo emisor: MAC1. Al intentar completar la dirección MAC de destino, se cae en la cuenta de que no se conoce por lo tanto se ejecuta ARP para encontrarla. No obstante, el ARP no se completa porque la dirección IP1 e IP4 están en redes distintas y ARP solo tiene alcance en el dominio de Broadcast.

En este caso PC1 no sabe cómo llegar a PC2 y necesita de la ayuda de su puerta de enlace por defecto o *default gateway* para dirigir el tráfico. El *default gateway* es un equipo a donde se envía todo el tráfico con destino desconocido. Para este ejemplo la dirección IP del *default gateway* es IP2, así que PC1 ejecuta nuevamente ARP para consultar la dirección MAC del equipo con la dirección IP2. El ROUTER1 responde a este *ARP Request* con su dirección MAC2 y es así que la trama se completa con la dirección MAC de destino MAC2.

En la figura 49 se puede observar el proceso de encapsulación de los datos enviados de IP1 a IP4.

Figura 49. Envío de información de PC1 a PC2



Fuente: elaboración propia, empleando eNSP.

En la capa física la trama es convertida a bits para viajar por el medio de transmisión hasta el equipo que se encuentra directamente conectado. El ROUTER1 recibe los bits y reconstruye la trama para dirigirla hacia la capa de enlace de datos. Aquí se compara la dirección MAC de destino de la trama, MAC2, con la dirección MAC que se tiene quemada en la NIC de ROUTER1, MAC2, y ya que ambas coinciden la trama es desencapsulada y dirigida hacia la capa de red. En esta capa se compara la dirección IP de destino del paquete, IP4, con la dirección IP que se tiene configurada en la interfaz de ROUTER1, IP2. Estas direcciones IP resultan no ser iguales por lo que el *router* consulta en una tabla llamada Tabla de Enrutamiento, para verificar si conoce cómo alcanzar al equipo con dirección IP4. El resultado de esta búsqueda indica que para llegar a IP4 se debe enviar el paquete por la interfaz configurada con la dirección IP3. Es importante recalcar que las direcciones IP de origen y destino del paquete no

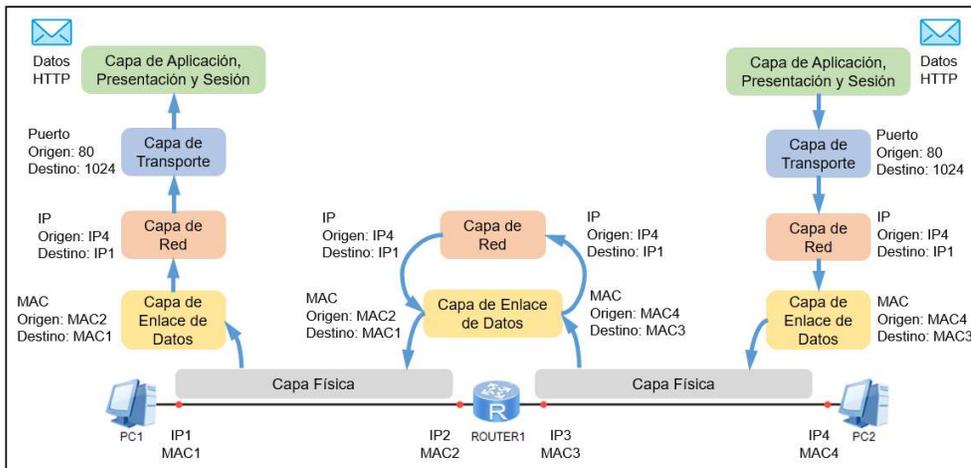
cambian en ningún momento, lo único que cambia es la dirección MAC de origen y destino. El paquete es encapsulado nuevamente formando una trama con dirección MAC de origen MAC3 y, suponiendo que en este nuevo segmento de Broadcast ya se ejecutó ARP, la dirección MAC de destino MAC4.

Nuevamente la trama se convierte en bits y se envía por el medio hacia PC2. Este equipo reconstruye la trama y verifica si la dirección MAC de destino, MAC4, es igual a la dirección MAC quemada en su tarjeta NIC, MAC4. Al encontrar ambas direcciones MAC iguales, desencapsula la trama volviéndola un paquete y la asciende hacia la capa de red. Aquí se valida que la dirección IP de destino del paquete, IP4, es igual a la dirección IP configurada en PC2, IP4, lo que indica que el paquete llegó a su destino a nivel lógico. El paquete es dirigido hacia la capa de transporte, donde el segmento ingresa por el puerto por el puerto 80 y se crea la conexión para obtener los datos. Finalmente, se muestran con el apoyo de la capa de aplicación, presentación y sesión.

El proceso anterior se realiza cuando los datos se dirigen de PC1 hacia PC2, no obstante, cuando el mensaje se responde de PC2 hacia PC1 el proceso es muy similar. La diferencia es que esta vez el puerto de origen del segmento es 80 y el de destino 1024, la dirección IP de origen del paquete es IP4 y el de destino IP1 y la dirección MAC de origen de la trama es MAC4 y la de destino MAC3. Las comparaciones de las direcciones durante el transporte de la información son iguales hasta llegar a PC1.

En la figura 50 se puede observar el proceso de encapsulación de datos enviados desde PC2 hacia PC1.

Figura 50. **Recepción de información de PC2 a PC1**



Fuente: elaboración propia, empleando eNSP.

2.5. Enrutamiento IP

El enrutamiento o ruteo es un proceso que se desarrolla en la capa de red del modelo OSI y su propósito es encontrar el camino o ruta más corta hacia una dirección de destino. Los dispositivos de capa tres, como *routers* o *switches* multicapa, pueden aprender una gran cantidad de rutas hacia un destino, pero siempre elegirán la más corta para enviar el paquete. La elección de la ruta más corta se basa en la comparación de varios parámetros que se describen más adelante.

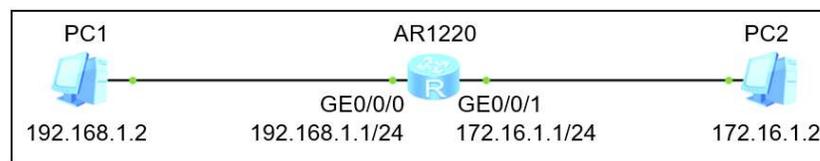
2.5.1. Proceso de enrutamiento

Cuando se configura una dirección IP en la interface de un *router*, este interpreta que por dicha interfaz puede alcanzar todas las direcciones que pertenezcan al segmento de red. A manera de ejemplo se toma un *router* Huawei

AR1220 con la dirección IP: 192.168.1.1/24 configurada en la interfaz GE0/0/0 y la dirección IP: 172.16.1.1/24 en la interfaz GE0/0/1.

Mediante esta acción el *router* comprende que cualquier paquete que ingrese por alguna de sus interfaces y cuente con una dirección IP de destino entre 192.168.1.2 - 192.168.1.255, se excluye la dirección de red y la dirección configurada en la interfaz del *router*, debe ser reenviada por la interfaz GE0/0/0. De igual forma, si la dirección IP de destino se encuentra en el rango 172.16.1.2 - 172.16.1.255 debe ser reenviada por la interfaz GE0/0/1.

Figura 51. **Ejemplo de enrutamiento IP**



Fuente: elaboración propia, empleando eNSP.

Esta información es almacenada en el *router* en una tabla llamada: tabla de enrutamiento. En este caso, la tabla de enrutamiento tendrá dos entradas denominadas: directamente conectadas, debido a que hay una interfaz que pertenece al segmento de red de destino.

2.5.2. Analizando la tabla de enrutamiento

Para visualizar la tabla de enrutamiento de un *router* Huawei se ingresa a la vista del usuario y se emplea el comando: *display ip routing-table*. En esta tabla se muestran todas las redes que se conocen, la forma en que las conoce y la ruta a tomar para llegar a ellas. En la figura 52, se muestra la tabla de enrutamiento del *router* AR1220 del ejemplo anterior.

Figura 52. Tabla de enrutamiento

```
<AR1220> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 10      Routes : 10

Destination/Mask    Proto    Pre    Cost    Flags NextHop         Interface
-----
 127.0.0.0/8        Direct   0      0        D    127.0.0.1       InLoopBack0
 127.0.0.1/32       Direct   0      0        D    127.0.0.1       InLoopBack0
127.255.255.255/32  Direct   0      0        D    127.0.0.1       InLoopBack0
172.16.1.0/24       Direct   0      0        D    172.16.1.1      GigabitEthernet0/0/1
172.16.1.1/32       Direct   0      0        D    127.0.0.1       GigabitEthernet0/0/1
 172.16.1.255/32    Direct   0      0        D    127.0.0.1       GigabitEthernet0/0/1
 192.168.1.0/24     Direct   0      0        D    192.168.1.1     GigabitEthernet0/0/0
 192.168.1.1/32     Direct   0      0        D    127.0.0.1       GigabitEthernet0/0/0
 192.168.1.255/32   Direct   0      0        D    127.0.0.1       GigabitEthernet0/0/0
255.255.255.255/32  Direct   0      0        D    127.0.0.1       InLoopBack0

<AR1220>
```

Rutas directas

Fuente: elaboración propia, empleando eNSP.

Cada fila de la tabla representa una ruta, es decir, el camino a tomar para llegar a una red de destino. Para interpretar cada entrada se tienen 7 columnas que describen cada parte de una ruta.

- Destination/Mask: indica la dirección de red o *host* de destino seguido de la longitud de la máscara de subred.
- Proto: indica el protocolo de enrutamiento por el cual se aprende la ruta. A continuación, se describen algunas de las formas por las que un *router* puede aprender rutas:
 - *Direct*: rutas directas, cuando el *router* tiene una interfaz que pertenece al segmento de red de destino.
 - *Static*: rutas estáticas.

- *OSPF*: rutas por OSPF.
 - *RIP*: rutas por RIP.
 - *UNR*: rutas por el usuario.
- Pre: indica la preferencia de la ruta.
 - Cost: indica el costo de la ruta.
 - Flag: indica las banderas de ruteo.
 - NextHop: indica la dirección IP del siguiente salto. Se trata de la dirección IP de la interfaz cuando es una ruta directa o la dirección IP del equipo que conoce la ruta para llegar a la red de destino.
 - Interface: indica la interfaz de salida por la cual se puede alcanzar el NextHop o el *host* de destino.

Si se analiza la tabla de enrutamiento del *router* AR1220 del ejemplo anterior, se pueden visualizar 10 rutas, no obstante, tres de ellas son para alcanzar la dirección de Loopback, 172.0.0.0/8, y una para alcanzar la dirección de Broadcast, 255.255.255.255/32. Es de esta manera que quedan seis rutas restantes que hacen referencia a las redes directamente conectadas al *router*. Para consultar por una ruta en específico se emplea el comando: *display ip routing-table ip_address*, donde el parámetro *ip_address* especifica alguna dirección IP dentro del segmento de red de destino.

Por ejemplo, si se consulta por la ruta que dirige hacia la red 192.168.1.0 nos indicará que esa red está directamente conectada a la interfaz GE0/0/0, como se muestra en la figura 53. Esta información puede ser corroborada con la

dirección IP configurada en la interfaz, para esto se puede emplear el comando: *display ip interface brief*. Es importante mencionar que la ruta se muestra en la tabla de enrutamiento únicamente si la interfaz de salida se encuentra activa, es decir, en estado up a nivel físico, *Physical*, y de protocolo, *Protocol*.

Figura 53. Consulta de una ruta específica y direcciones IP

```

<AR1220> display ip routing-table 192.168.1.0
Route Flags: R - relay, D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask  Proto  Pre  Cost    Flags NextHop        Interface
-----
192.168.1.0/24   Direct  0    0       D    192.168.1.1        GigabitEthernet0/0/0

<AR1220> display ip interface brief
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
The number of interface that is UP in Physical is 3
The number of interface that is DOWN in Physical is 0
The number of interface that is UP in Protocol is 3
The number of interface that is DOWN in Protocol is 0

Interface                IP Address/Mask    Physical  Protocol
-----
GigabitEthernet0/0/0     192.168.1.1/24    up        up
GigabitEthernet0/0/1     172.16.1.1/24     up        up
NULL0                    unassigned         up        up(s)
<AR1220>

```

Fuente: elaboración propia, empleando eNSP.

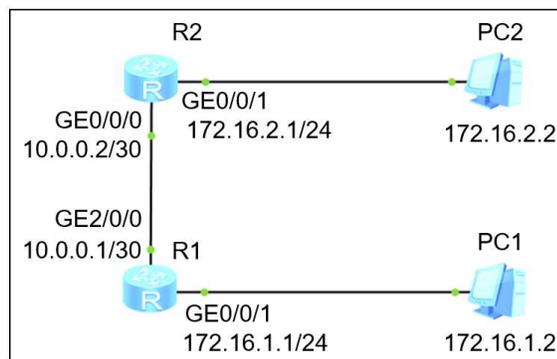
Resulta evidente que un *router* sepa cómo llegar a las redes que están directamente conectadas a él, pero ¿qué ocurre con las redes que no lo están? En este caso se puede emplear enrutamiento estático o dinámico para indicar al *router* el camino a seguir para llegar a esas redes distantes. La implementación de protocolos de enrutamiento y rutas estáticas depende de la complejidad de la topología de red.

2.5.3. Enrutamiento estático

El enrutamiento estático es común en topologías pequeñas ya que deben ser configuradas manualmente por el usuario. Este proceso consiste en agregar entradas estáticas a la tabla de enrutamiento con el propósito de instruir al *router* sobre cuál interfaz de salida emplear para llegar a una red de destino que no esté directamente conectada a él.

En la topología mostrada en la figura 54, la computadora PC1 desea enviar un paquete ICMP o *ping*, hacia la computadora PC2. En la tabla ARP de PC1 no existe una entrada para la dirección IP de destino 172.16.2.2 y al ejecutar ARP no habría respuesta al *ARP Request*, por tanto, el paquete es enviado al *default gateway*, que en este caso es la interfaz GE0/0/1 de R1.

Figura 54. **Ejemplo de una topología para implementar rutas estáticas**



Fuente: elaboración propia, empleando eNSP.

Una vez ejecutado ARP, el *host* PC1 envía el paquete hacia R1. El *router* R1 recibe el paquete por la interfaz GE0/0/1 y busca en su tabla de enrutamiento algún camino o ruta con destino hacia el segmento 172.16.2.0/24. En este caso, R1 únicamente sabe cómo llegar a las dos redes que están directamente

conectadas a este, 172.16.1.0/24 y 10.0.0.0/30, y no cuenta con ningún camino trazado para alcanzar el segmento de destino 172.16.2.0/24 por lo que el paquete nunca llegaría hasta PC2. En la figura 55, se muestra la tabla de enrutamiento del *router* R1.

Figura 55. **Tabla de enrutamiento de R1**

```

<R1> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 13      Routes : 13

Destination/Mask    Proto    Pre    Cost    Flags NextHop         Interface
-----
10.0.0.0/30        Direct   0      0      D    10.0.0.1      GigabitEthernet2/0/0
10.0.0.1/32        Direct   0      0      D    127.0.0.1     GigabitEthernet2/0/0
10.0.0.3/32        Direct   0      0      D    127.0.0.1     GigabitEthernet2/0/0
127.0.0.0/8        Direct   0      0      D    127.0.0.1     InLoopBack0
127.0.0.1/32       Direct   0      0      D    127.0.0.1     InLoopBack0
127.255.255.255/32 Direct   0      0      D    127.0.0.1     InLoopBack0
172.16.1.0/24      Direct   0      0      D    172.16.1.1    GigabitEthernet0/0/1
172.16.1.1/32      Direct   0      0      D    127.0.0.1     GigabitEthernet0/0/1
172.16.1.255/32    Direct   0      0      D    127.0.0.1     GigabitEthernet0/0/1
255.255.255.255/32 Direct   0      0      D    127.0.0.1     InLoopBack0
  
```

Fuente: elaboración propia, empleando eNSP.

Por el contrario, el *router* R2 sí conoce cómo alcanzar el segmento de destino, ya que está configurado directamente a él. Por lo que resulta necesario crear una ruta para indicar a R1 que envíe los paquetes destinados al segmento 172.16.2.0/24 hacia R2. Para esto se puede crear una ruta estática en R1 que envíe los paquetes hacia R2. Ahora bien, para dirigir los paquetes hacia R2 se tienen dos opciones:

- Expulsar los paquetes por la interfaz GE2/0/0, ya que ambos *routers* están directamente conectados.
- Dirigir los paquetes hacia la dirección IP de R2 del segmento compartido con R1, es decir 10.0.0.2. A esta dirección se le conoce como siguiente

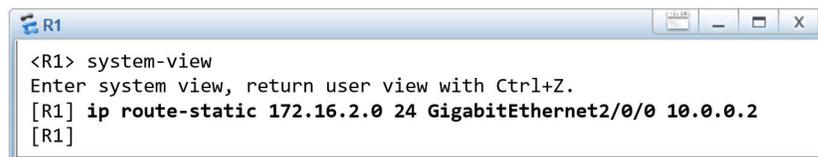
salto o NextHop, ya que esa dirección IP sería el siguiente salto del paquete para llegar a su destino.

En la práctica, se recomienda crear las rutas estáticas con ambos parámetros, la interfaz de salida y la dirección del NextHop, debido a que los dos *routers* no siempre estarán conectados directamente y es una forma eficiente para que el *router* evite búsquedas recursivas en la tabla de enrutamiento.

Para crear una ruta estática se ingresa a la vista del sistema y se emplea el comando: `ip route-static ip_address {mask | mask_length} {nexthop_address | interface_number [nexthop_address]} [preference preference]`, donde el parámetro `ip_address` indica la dirección de red de destino, `mask` la máscara de subred o su longitud, `nexthop_address` la dirección IP del siguiente salto, `interface_number` la interfaz de salida y finalmente el parámetro opcional `preference` que indica la preferencia de la ruta.

En la figura 56, se muestra la creación de la ruta estática en R1 para indicarle que para llegar al segmento 172.16.2.0/24 se debe usar la interfaz de salida GE2/0/0 y el NextHop 10.0.0.2, ya que de ambas formas se alcanza al *router* R2.

Figura 56. Ruta estática en R1

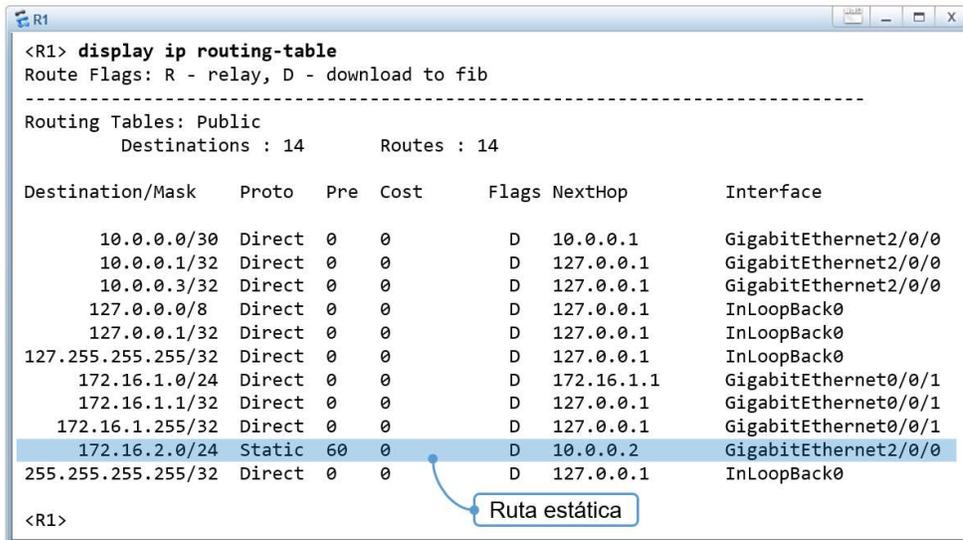


```
<R1> system-view
Enter system view, return user view with Ctrl+Z.
[R1] ip route-static 172.16.2.0 24 GigabitEthernet2/0/0 10.0.0.2
[R1]
```

Fuente: elaboración propia, empleando eNSP.

De esta forma se crea una nueva entrada en la tabla de enrutamiento de R1 de tipo *Static*, como se muestra en la figura 57.

Figura 57. Ruta estática en la tabla de enrutamiento de R1



```
<R1> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 14      Routes : 14

Destination/Mask    Proto    Pre  Cost    Flags NextHop         Interface
-----
 10.0.0.0/30        Direct  0    0        D  10.0.0.1         GigabitEthernet2/0/0
 10.0.0.1/32        Direct  0    0        D  127.0.0.1        GigabitEthernet2/0/0
 10.0.0.3/32        Direct  0    0        D  127.0.0.1        GigabitEthernet2/0/0
 127.0.0.0/8        Direct  0    0        D  127.0.0.1        InLoopBack0
 127.0.0.1/32       Direct  0    0        D  127.0.0.1        InLoopBack0
127.255.255.255/32  Direct  0    0        D  127.0.0.1        InLoopBack0
 172.16.1.0/24      Direct  0    0        D  172.16.1.1       GigabitEthernet0/0/1
 172.16.1.1/32      Direct  0    0        D  127.0.0.1        GigabitEthernet0/0/1
 172.16.1.255/32    Direct  0    0        D  127.0.0.1        GigabitEthernet0/0/1
 172.16.2.0/24      Static  60   0        D  10.0.0.2         GigabitEthernet2/0/0
255.255.255.255/32  Direct  0    0        D  127.0.0.1        InLoopBack0

<R1>
```

Fuente: elaboración propia, empleando eNSP.

Con esta nueva ruta en R1 todos los paquetes que estén dirigidos hacia el segmento 172.16.2.0/24 serán reenviados por la interfaz GE2/0/0 hacia R2.

Con esta nueva ruta, R2 recibe los paquetes y los redirecciona a la interfaz GE0/0/1 para alcanzar a PC2. Todo este tiempo el paquete que viajó por la red fue un *ICMP Echo Request*, y ahora que alcanzó a PC2, este debe responder con un paquete *ICMP Echo Reply* para completar la solicitud y asegurar que haya comunicación bidireccional punto a punto.

PC2 crea un paquete de respuesta con dirección IP de destino 172.16.1.1 y lo envía hacia su *default gateway*, ya que no tiene una entrada en su tabla ARP para PC1. R2 recibe este nuevo paquete por la interfaz GE0/01 y busca en su

tabla de enrutamiento alguna ruta con destino a la red 172.16.1.0/24 pero ocurre lo mismo que con R1: no conoce cómo llegar al segmento de destino puesto que solo conoce las redes que están conectadas directamente a él.

Entonces es necesario crear una nueva ruta estática en R2 que le indique enviar los paquetes destinados al segmento 172.16.1.0/24 por la interfaz GE0/0/0 rumbo al NextHop 10.0.0.1 que es R1.

En la figura 58, se puede observar la creación de la ruta estática en R2 y en la figura 59 se muestra la tabla de enrutamiento.

Figura 58. **Configuración de una ruta estática en R2**

```

<R2> system-view
Enter system view, return user view with Ctrl+Z.
[R2] ip route-static 172.16.1.0 24 GigabitEthernet0/0/0 10.0.0.1
[R2]
  
```

Fuente: elaboración propia, empleando eNSP.

Figura 59. **Tabla de enrutamiento en R2**

```

<R2> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 11      Routes : 11

Destination/Mask    Proto    Pre    Cost    Flags NextHop         Interface
-----
 10.0.0.0/30        Direct  0      0        D   10.0.0.2       GigabitEthernet0/0/0
 10.0.0.2/32        Direct  0      0        D   127.0.0.1      GigabitEthernet0/0/0
 10.0.0.3/32        Direct  0      0        D   127.0.0.1      GigabitEthernet0/0/0
 127.0.0.0/8        Direct  0      0        D   127.0.0.1      InLoopBack0
 127.0.0.1/32       Direct  0      0        D   127.0.0.1      InLoopBack0
127.255.255.255/32  Direct  0      0        D   127.0.0.1      InLoopBack0
172.16.1.0/24      Static   60     0        D   10.0.0.1       GigabitEthernet0/0/0
172.16.2.0/24      Direct  0      0        D   172.16.2.1     GigabitEthernet0/0/1
 172.16.2.1/32      Direct  0      0        D   127.0.0.1      GigabitEthernet0/0/1
 172.16.2.255/32    Direct  0      0        D   127.0.0.1      GigabitEthernet0/0/1
255.255.255.255/32 Direct  0      0        D   127.0.0.1      InLoopBack0
  
```

A blue box labeled "Ruta estática" points to the row for 172.16.1.0/24.

Fuente: elaboración propia, empleando eNSP.

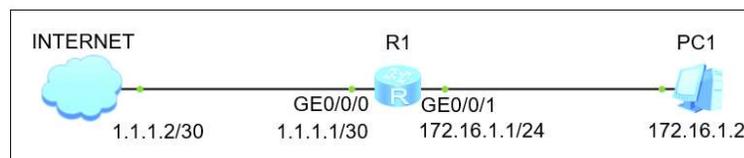
Finalmente, R1 recibe el paquete y lo dirige hacia PC1 completando así la comunicación bidireccional solicitada por el protocolo ICMP.

2.5.3.1. Ruta por defecto

Un *router* debe tener una ruta en su tabla de enrutamiento para alcanzar una red de destino, sin embargo, cuando las redes se vuelven muy grandes, como es el caso de Internet, esto se vuelve imposible, ya que se saturarían la tabla. Es por esta razón que un *router* no necesariamente debe tener una ruta específica para cada destino, basta con tener una sola ruta hacia un equipo que conozca cómo llegar a esa gran cantidad de redes y así poder mandarle todo el tráfico, a este tipo de ruta se conoce como ruta por defecto.

Este tipo de ruta es frecuentemente usado en los *routers* de usuarios finales que requieren conectarse a Internet. En estos casos se configura una ruta por defecto que dirija todo el tráfico hacia los equipos del ISP, quien posteriormente se encarga de redireccionarlo hacia la salida a Internet. En la topología mostrada en la figura 60, se muestra un ejemplo clásico donde se requiere la implementación de una ruta por defecto.

Figura 60. **Router conectado hacia Internet**



Fuente: elaboración propia, empleando eNSP.

Para crear una ruta por defecto se ingresa a la vista del sistema y se emplea el comando: `ip route-static 0.0.0.0 0 {nexthop_address | interface_number`

[*nexthop_address*]], donde el parámetro *nexthop_address* es la dirección IP del siguiente salto, generalmente el equipo del ISP, y el parámetro *interface_number* la interfaz de salida.

En la figura 61 se muestra la creación de una ruta por defecto hacia la dirección 1.1.1.2 y en la figura 62 se muestra la ruta en la tabla de enrutamiento.

Figura 61. **Configuración de una ruta por defecto en R1**

```

R1
<R1> system-view
Enter system view, return user view with Ctrl+Z.
[R1] ip route-static 0.0.0.0 0 GigabitEthernet0/0/0 1.1.1.2
[R1]
  
```

Fuente: elaboración propia, empleando eNSP.

Figura 62. **Tabla de enrutamiento de R1**

```

R1
<R1> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 11      Routes : 11

Destination/Mask    Proto    Pre  Cost    Flags NextHop          Interface
-----
 0.0.0.0/0          Static   60   0        D    1.1.1.2           GigabitEthernet0/0/0
 1.1.1.0/30         Direct   0     0        D    1.1.1.1           GigabitEthernet0/0/0
 1.1.1.1/32         Direct   0     0        D    127.0.0.1         GigabitEthernet0/0/0
 1.1.1.3/32         Direct   0     0        D    127.0.0.1         GigabitEthernet0/0/0
 127.0.0.0/8        Direct   0     0        D    127.0.0.1         InLoopBack0
 127.0.0.1/32       Direct   0     0        D    127.0.0.1         InLoopBack0
127.255.255.255/32  Direct   0     0        D    127.0.0.1         InLoopBack0
 172.16.1.0/24      Direct   0     0        D    172.16.1.1       GigabitEthernet0/0/1
 172.16.1.1/32      Direct   0     0        D    127.0.0.1         GigabitEthernet0/0/1
 172.16.1.255/32    Direct   0     0        D    127.0.0.1         GigabitEthernet0/0/1
255.255.255.255/32  Direct   0     0        D    127.0.0.1         InLoopBack0

<R1>
  
```

Fuente: elaboración propia, empleando eNSP.

2.5.4. Enrutamiento dinámico

El enrutamiento dinámico otorga a un *router* la habilidad de aprender nuevas rutas de sus *routers* vecinos al implementar protocolos de enrutamiento. Los protocolos de enrutamiento dinámicos se dividen en dos grupos: los IGP o Interior Gateway Protocols y los EGP o Exterior Gateway Protocols.

Los protocolos IGP se encargan de la comunicación dentro de un mismo sistema autónomo o AS, por ejemplo: RIP, OSPF e IS-IS.

Los protocolos EGP se encargan de la comunicación entre distintos sistemas autónomos, actualmente el único EGP empleado en Internet es BGP o Border Gateway Protocol.

En un *router* pueden existir una ruta estática y dinámica apuntando hacia la misma red de destino, es por esta razón que debe existir un criterio de elección de la ruta óptima para ser instalada en la tabla de enrutamiento, ya que en ella se instalan únicamente las mejores rutas. Para esta elección el *router* se basa en la preferencia y en el costo de la ruta.

2.5.4.1. Preferencia de la ruta

La preferencia es un número entero que califica la confiabilidad del protocolo de enrutamiento por el que se aprende una ruta, está comprendida entre 0 y 255, mientras más cercana a 0, más confiable será la información. En otras palabras, si un *router* cuenta con dos rutas hacia el mismo destino, una ruta estática con preferencia 60, y una dinámica aprendida por OSPF con preferencia 10, instalará la ruta dinámica en la tabla de enrutamiento ya que la información de enrutamiento aprendida por OSPF es más confiable que la información

aprendida con la ruta estática. Es importante mencionar que en el caso de las rutas estáticas el parámetro de preferencia puede ser modificado a conveniencia.

El término preferencia también es conocido por otros fabricantes como distancia administrativa. En la tabla VI, se muestra una tabla con la preferencia de distintos protocolos de enrutamiento para dispositivos Huawei.

Tabla VI. **Preferencias por defecto**

Ruta	Preferencia por Defecto
Ruta directa	0
Ruta aprendida por OSPF	10
Ruta aprendida por IS-IS	15
Ruta estática	60
Ruta aprendida por RIP	100
Ruta aprendida por BGP	255

Fuente: elaboración propia.

2.5.4.2. Costo de la ruta

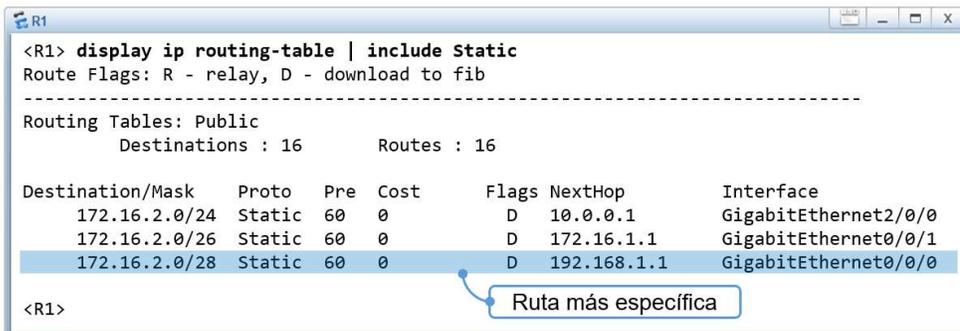
Cuando un *router* conoce varias rutas hacia una red de destino aprendidas por el mismo protocolo, se debe elegir la más corta para ser instalada en la tabla de enrutamiento. El costo es un valor que califica a una ruta e indica la distancia hacia la red de destino. Cada protocolo de enrutamiento tiene una forma específica para medir el costo de una ruta, puede ser calculado en base al número de saltos, en base al ancho de banda acumulado de la ruta, entre otros.

2.5.5. Elección de la mejor ruta

Para que una ruta sea inyectada en la tabla de enrutamiento debe cumplir dos condiciones: la primera es contar con la preferencia más baja y la segunda es contar con el costo más bajo. Sin embargo, habrá ocasiones en que el *router* aprenda dos o más rutas con el mismo costo hacia una red, es entonces que se elige la ruta más específica.

Para comprender mejor esta situación se toma como ejemplo un *router* que desea enviar un paquete hacia la IP: 172.16.2.5. El *router* cuenta con 3 rutas estáticas apuntando hacia segmentos que incluyen a la IP de destino, como se muestra en figura 63.

Figura 63. **Tabla de enrutamiento filtrada para mostrar rutas estáticas**



```
<R1> display ip routing-table | include Static
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 16      Routes : 16

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
172.16.2.0/24      Static  60   0        D   10.0.0.1      GigabitEthernet2/0/0
172.16.2.0/26      Static  60   0        D   172.16.1.1   GigabitEthernet0/0/1
172.16.2.0/28      Static  60   0        D   192.168.1.1  GigabitEthernet0/0/0
-----
<R1>
```

Fuente: elaboración propia, empleando eNSP.

Debido a que la máscara de subred no viaja en el paquete IP, el *router* no puede determinar hacia cuál de las tres subredes se dirige el paquete. En estos casos se elige la ruta más específica, es decir, la ruta que incluya la IP de destino y que cuenta con la máscara de subred de más alto valor. Para este ejemplo el

paquete sería enviado por la interfaz de salida GE0/0/0 hacia el NextHop 192.168.1.1 debido a que la máscara de subred es 255.255.255.240.

2.5.6. Traza de un paquete

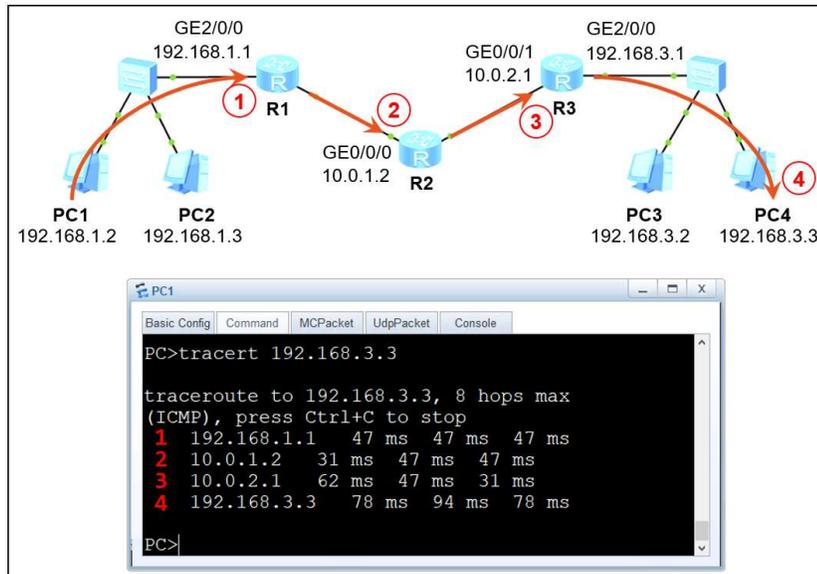
Una traza es una forma de representar la ruta por la que viaja un paquete hasta llegar su destino. Resulta muy útil cuando se desea verificar si la comunicación está siendo establecida por la ruta deseada. En la mayoría de los equipos de red la traza se representa como un listado de los equipos que atraviesa el paquete hasta llegar a su destino. Este listado se construye gracias al parámetro Time-To-Live o TTL de los paquetes IP. El parámetro TTL es un número que generalmente inicia en 128 y va disminuyendo una unidad cada vez que el paquete atraviesa un equipo de capa tres, al llegar a 0 el paquete es descartado. La principal función del parámetro TTL es evitar que los paquetes queden circulando por tiempo indefinido en una red.

Para iniciar la construcción de la traza, el equipo envía un *ping* hacia la dirección IP de destino con un TTL=1. El primer equipo en recibir el paquete resta una unidad al TTL volviéndolo 0 y envía una respuesta hacia el emisor indicando que el destino es inalcanzable, es decir TTL=0, y es de esta forma que el emisor recibe el paquete y detecta el primer equipo intermedio. Seguidamente el equipo emisor envía nuevamente un *ping* hacia la IP de destino, pero aumenta una unidad el TTL, es decir TTL=2. Con este procedimiento se logra detectar las direcciones IP de las interfaces de los equipos por donde ingresa el paquete en su camino hacia la dirección IP de destino.

Para obtener la traza hacia una dirección IP de destino se emplea el comando: `tracert [-a source_ip_address] ip_address`, donde el parámetro `source_ip_address` especifica la dirección IP de origen e `ip_address` la dirección

IP de destino. En la figura 64 se muestra la traza de un paquete que atraviesa 3 *routers* hasta llegar a su destino.

Figura 64. Traza de PC1 a PC4



Fuente: elaboración propia, empleando eNSP

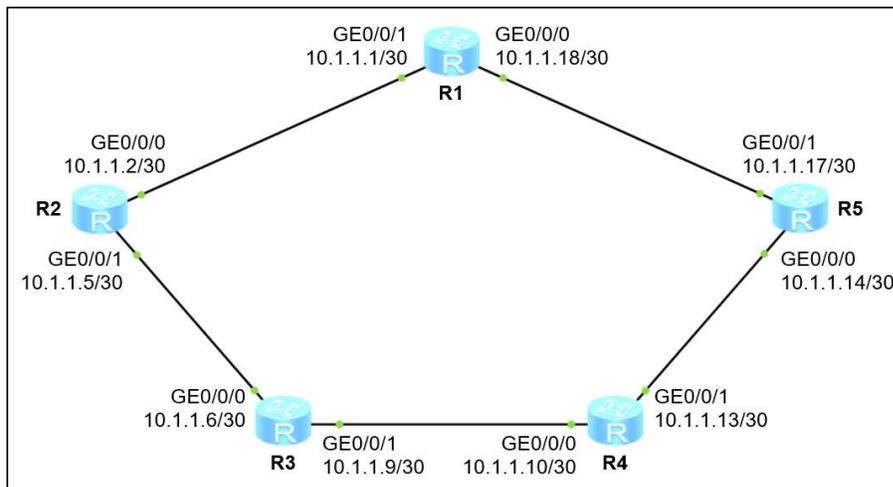
2.6. Routing Information Protocol

RIP es el protocolo de enrutamiento menos utilizado en la actualidad, pero debido a su simplicidad es de vital importancia analizar su comportamiento para comprender protocolos más complejos. Se considera un protocolo IGP vector-distancia, es decir, que las rutas aprendidas por RIP almacenan información sobre la distancia hacia una red de destino y la interfaz de salida, dirección, para llegar a ella. Cuenta con una preferencia de 100 y determina el costo de una ruta en base a la cantidad de saltos que se deben dar para llegar a un destino. Una red se considera inalcanzable cuando se deben realizar 16 saltos o más para llegar al destino. RIP trabaja en la capa de aplicación del modelo TCP/IP.

2.6.1. Principio de operación

Para analizar el funcionamiento de RIP se toma como ejemplo la topología mostrada en la figura 65.

Figura 65. Topología de red implementando RIP



Fuente: elaboración propia, empleando eNSP.

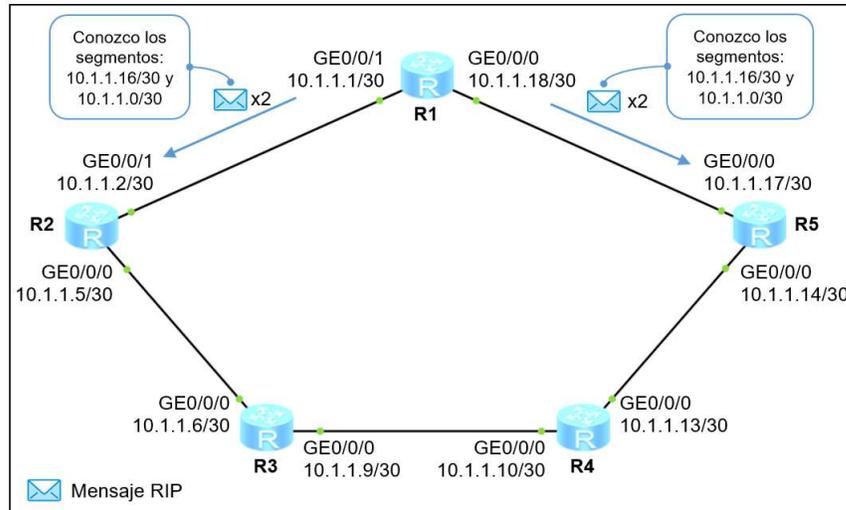
Luego de configurar las direcciones IP en las interfaces de cada *router* se crean de forma automática las rutas directas y se instalan en la tabla de enrutamiento. Inicialmente R1 únicamente conoce dos rutas directas y sabe que para alcanzar el segmento 10.1.1.0/30 debe utilizar la interfaz GE0/0/1 mientras que para alcanzar el segmento 10.1.1.16/30 debe utilizar la interfaz GE0/0/0. Debido a esto es evidente que R1 no cuenta con una ruta para alcanzar los segmentos 10.1.1.4/30, 10.1.1.8/30 y 10.1.1.12/30. El mismo escenario se repite con el resto de *routers* de la topología: únicamente conocen los segmentos de red a los cuales están directamente conectados.

El propósito de RIP es lograr que todos los *routers* tengan por lo menos una ruta para alcanzar todos los segmentos de red y que se actualicen de forma automática. Luego de habilitar RIP en todos los *routers*, el proceso inicia de la siguiente forma: R1 crea un mensaje RIP en el cual encapsula información que indica que conoce cómo alcanzar el segmento 10.1.1.0/30 y lo envía a través de sus dos interfaces activas. Este mensaje es recibido por R2 a través de su interfaz GE0/0/1 y por R5 a través de su interfaz GE0/0/0. Ahora R2 sabe que el segmento 10.1.1.0/30 lo puede alcanzar utilizando como interfaz de salida la GE0/0/1, este dato ya lo conocía R2 debido a que tiene una ruta directa hacia ese segmento, no obstante, con R5 el caso es distinto. Con esta nueva información, R5 sabe que el segmento 10.1.1.0/30 puede ser alcanzado utilizando como interfaz de salida la GE0/0/0. Este mismo procedimiento se realiza para cada ruta directa de R1, que para este ejemplo resultan ser solo dos. Es importante mencionar que primero se analiza el comportamiento de R1 únicamente por fines didácticos, en realidad todos los *routers* realizan el procedimiento anterior de forma paralela.

En la figura 66, se muestran los dos mensajes RIP enviados por R1 para dar a conocer las redes que conoce.

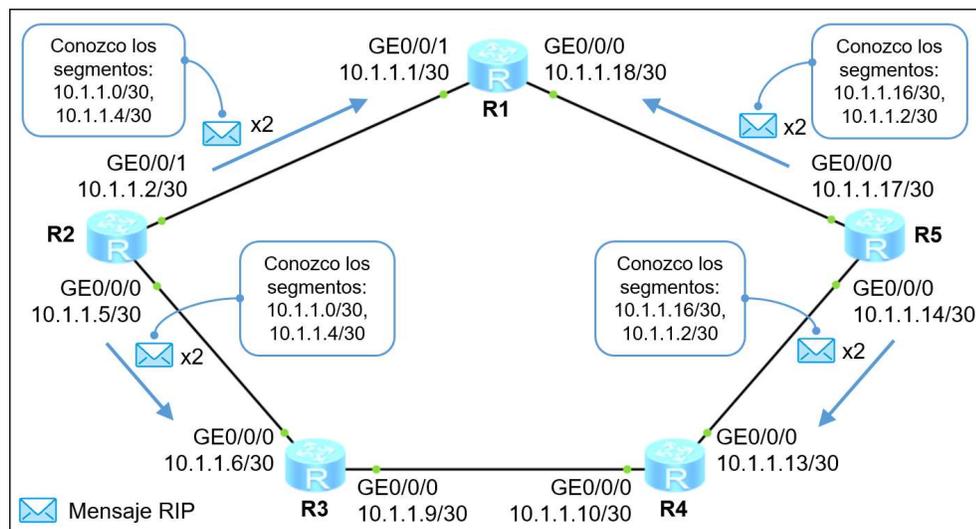
El proceso continúa con R2 y R5 quienes repiten el mismo procedimiento que R1, es decir, envían dos mensajes RIP indicando que conocen dos rutas directas. Es importante mencionar que R2 ya no anuncia el segmento 10.1.1.16/30 a R1 ya que fue este quien se lo anunció, esta propiedad del protocolo es denominada *split horizon*. En la figura 67, se muestran los paquetes RIP enviados por R2 y R5.

Figura 66. Mensajes RIP enviados por R1



Fuente: elaboración propia, empleando eNSP.

Figura 67. Mensajes RIP enviados por R2 y R5



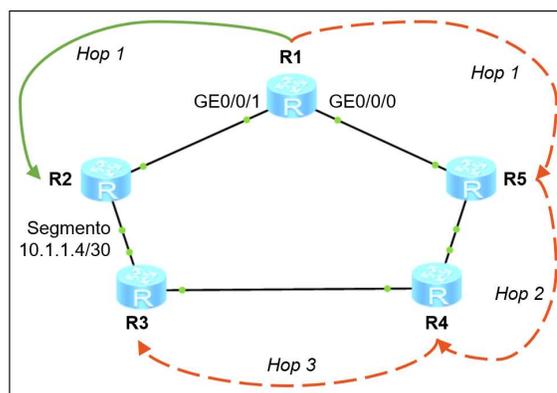
Fuente: elaboración propia, empleando eNSP.

Progresivamente todos los *routers* de la topología realizan el mismo procedimiento hasta conocer todos los segmentos de red. A este estado se le denomina convergencia de la red.

Analizando el caso de R1, RIP encontró dos rutas para llegar al segmento 10.1.1.4/30, la primera es saliendo por la interfaz GE0/0/1 hacia R2 y la segunda es saliendo por la interfaz GE0/0/0 hacia R5, luego hacia R4 y finalmente hacia R3. En este caso se debe elegir cuál de las dos rutas es la óptima y así instalarla en la tabla de enrutamiento. Para esto el protocolo se basa en el costo o métrica de la ruta, que para RIP es la cantidad de saltos o *hops* que se deben dar para llegar a la dirección de red de destino. R1 debe dar 1 saltos para llegar a R2 al salir por la GE0/0/1 mientras que debe dar 3 saltos para llegar a R3 al salir por la GE0/0/0, por tanto, la ruta que se inyecta en la tabla de enrutamiento es la que tiene interfaz de salida GE0/0/1 y costo 1.

En la figura 68, se muestra la cantidad de saltos que debe dar R1 para alcanzar el segmento 10.1.1.4/30.

Figura 68. **Métrica o costo de la ruta en RIP**



Fuente: elaboración propia, empleando eNSP.

En el caso de las rutas aprendidas por RIP únicamente se indica la interfaz de salida o dirección y la cantidad de saltos que se deben dar para llegar a un destino o distancia, es por esto por lo que se denomina un protocolo de vector distancia. Si existen dos o más rutas hacia el mismo destino y con el mismo costo, RIP las agrega en la tabla de enrutamiento realizando un balanceo de cargas, es decir que el tráfico se divide entre las rutas. RIP puede agregar hasta un máximo de 4 rutas para balancear cargas a la tabla de enrutamiento, pero este valor puede ser disminuido a conveniencia empleando el comando: *maximum load-balancing {1-4}*, en la vista del protocolo.

En la tabla VII, se muestra un bosquejo de la tabla de enrutamiento de R1 del ejemplo anterior, para el segmento 10.1.1.8/30 se tienen dos rutas con el mismo costo y se realizaría balanceo de cargas.

Tabla VII. **Ejemplo de la tabla de enrutamiento de R1**

Segmento de red	Interfaz de salida	Costo
10.1.1.0/30	GE0/0/1	0
10.1.1.4/30	GE0/0/1	1
10.1.1.8/30	GE0/0/1 GE0/0/0	2
10.1.1.12/30	GE0/0/0	1
10.1.1.16/30	GE0/0/0	0

Fuente: elaboración propia.

Para mantener actualizada la tabla de enrutamiento, cada *router* envía actualizaciones cada 30 segundos indicando los segmentos de red que conoce. Si un *router* detecta la caída de una interfaz envía una actualización inmediata

para no esperar la cuenta regresiva o *countdown* de 30 segundos y así agilizar la convergencia de red.

2.6.2. Temporizadores

Se describen a continuación.

2.6.2.1. Temporizador de actualización

Como se mencionó anteriormente un *router* que implementa RIP envía actualizaciones cada 30 segundos a todos sus vecinos. La forma en que esto sucede es similar al funcionamiento de un temporizador: el *router* envía las actualizaciones y comienza una cuenta regresiva que inicia en 30 segundos y finaliza en 0, al llegar al final del tiempo se envían las actualizaciones nuevamente.

2.6.2.2. Temporizador para invalidar rutas

Cuando un *router* aprende una ruta por RIP se le asigna un temporizador de 180 segundos e inicia la cuenta regresiva. Cada vez que el *router* recibe actualizaciones de esa ruta por algún vecino el temporizador regresa a 180 segundos. Si el temporizador llega a 0 significa que ningún vecino actualizó esa ruta y por lo tanto el *router* la considerará como inalcanzable, es decir, le asignará un costo de 16.

2.6.2.3. Temporizador para eliminar rutas

Cuando el temporizador de 180 segundos que considera una ruta como invalida llega a 0 la ruta es considerada como inalcanzable, no obstante, el *router*

no la elimina inmediatamente, en su lugar inicia un tercer temporizador de 120 segundos. En este período de tiempo el *router* anuncia a sus vecinos que la ruta tiene un costo de 16, es decir es inalcanzable.

Si algún vecino le anuncia que conoce la ruta y tiene una métrica inferior a 16 entonces la ruta vuelve a ser alcanzable y el segundo temporizador regresa a 180. Por el contrario, si el tercer temporizador de 120 segundos culmina y no se tuvo ninguna actualización entonces el *router* procede a eliminar la ruta de la tabla de enrutamiento.

2.6.3. RIP-1 vs. RIP-2

RIP inició con su primera versión manejando direccionamiento *classful*, no obstante, a medida que se implementaba el direccionamiento *classless* evolucionó a su segunda versión, que es la más usada hoy en día. En la tabla VIII se describen las diferencias entre ambas versiones.

Tabla VIII. Diferencia entre RIP-1 y RIP-2

RIP-1	RIP-2
Únicamente soporta rutas <i>classful</i>	Soporta rutas <i>classless</i>
Utiliza el puerto UDP 520	
Envía actualizaciones por Broadcast a la dirección: 255.255.255.255	Envía actualizaciones por Multicast o Broadcast a las direcciones: 224.0.0.9 o 255.255.255.255, respectivamente.
Envía actualizaciones por Broadcast a la dirección MAC: FF-FF-FF-FF-FF-FF	Envía actualizaciones por Broadcast o Multicast.
Identifica el campo del paquete llamado protocolo, con el número hexadecimal: 0x11	
Identifica el campo de la trama llamado: Tipo, con el número hexadecimal: 0x0800	
Consumo mucho recurso de procesamiento	Consumo poco recurso de procesamiento
No hay opción de autenticación	Posee la opción de autenticación

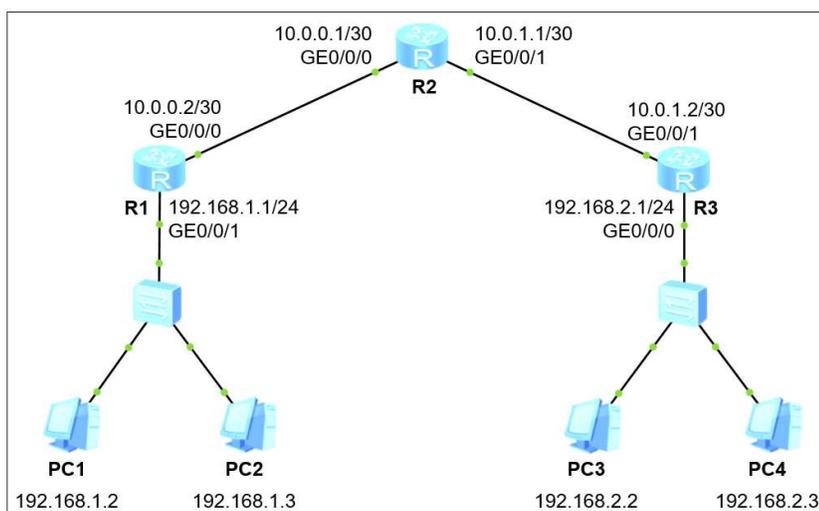
Fuente: elaboración propia.

2.6.4. Implementando RIP

Para iniciar la configuración de RIP en un *router* se debe ingresar a la vista del sistema y emplear el comando: *rip process_id*, donde el parámetro *process_id* es un número entero entre 1 y 65 535 que identifica proceso RIP, este número no necesariamente debe ser igual entre dos *router* para ser vecinos ya que pueden ejecutar varios procesos RIP simultáneamente. Seguidamente se emplea el comando: *version {1 | 2}*, para indicar la versión de RIP a implementar. Finalmente se agregan los segmentos de red directamente conectados al *router*, rutas directas, empleando el comando: *network ip_address*, donde *ip_address* indica la dirección de red del segmento.

Es importante mencionar que RIP no anuncia rutas, lo que hace realmente es anunciar los segmentos de red que se encuentran configurados en sus interfaces. El único protocolo capaz de anunciar rutas es BGP. En la figura 69 se muestra la topología de red propuesta para implementar RIP.

Figura 69. Topología para implementar RIP-2

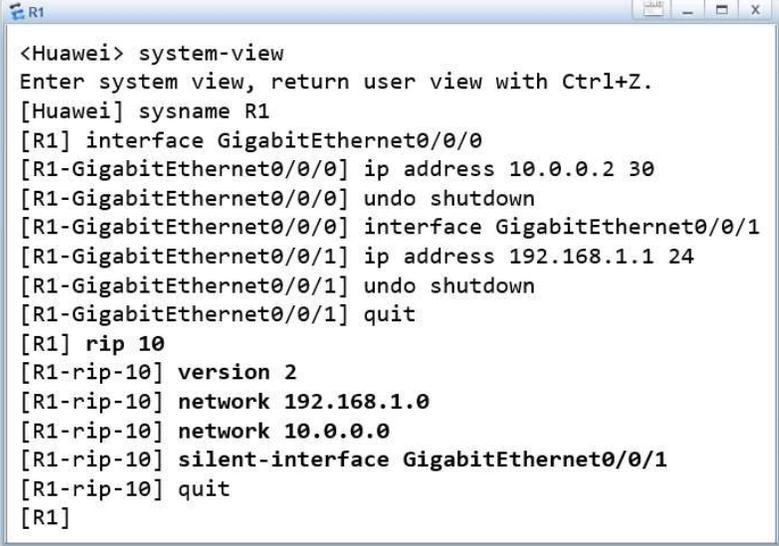


Fuente: elaboración propia, empleando eNSP.

En las cuatro computadoras únicamente se configura la dirección IP, máscara de subred y la dirección del *gateway*. En las figuras de la 70 a la 72 se muestra la configuración de los *routers* R1, R2 y R3 respectivamente.

En la configuración de R1 y R3 se aplica un comando llamado: *silent-interface* a la interfaz GE0/0/1 de R1 y GE0/0/0 de R3. Este comando indica que las actualizaciones que RIP envía cada 30 segundos no deben ser envidadas por las interfaces especificadas. El propósito de esta acción es impedir que los equipos de la LAN, PC1, PC2, PC3 y PC4, reciban las actualizaciones y puedan usarlas de forma maliciosa, además de evitar el uso innecesario del ancho de banda de esos segmentos.

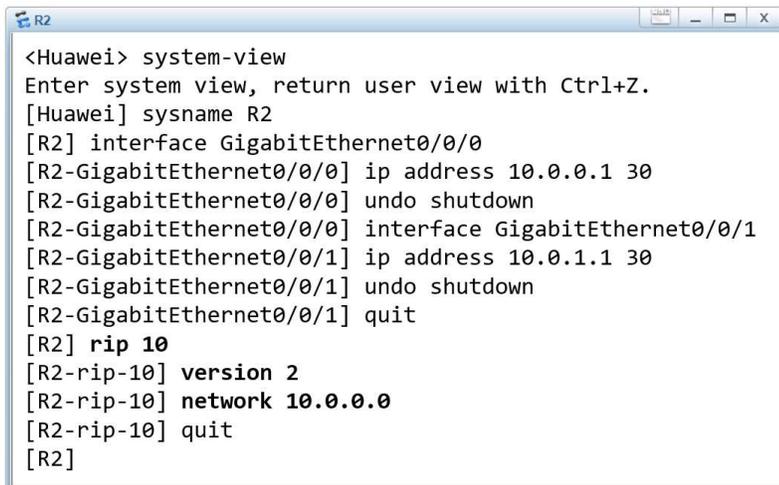
Figura 70. **Configuración de R1**



```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R1
[R1] interface GigabitEthernet0/0/0
[R1-GigabitEthernet0/0/0] ip address 10.0.0.2 30
[R1-GigabitEthernet0/0/0] undo shutdown
[R1-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R1-GigabitEthernet0/0/1] ip address 192.168.1.1 24
[R1-GigabitEthernet0/0/1] undo shutdown
[R1-GigabitEthernet0/0/1] quit
[R1] rip 10
[R1-rip-10] version 2
[R1-rip-10] network 192.168.1.0
[R1-rip-10] network 10.0.0.0
[R1-rip-10] silent-interface GigabitEthernet0/0/1
[R1-rip-10] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

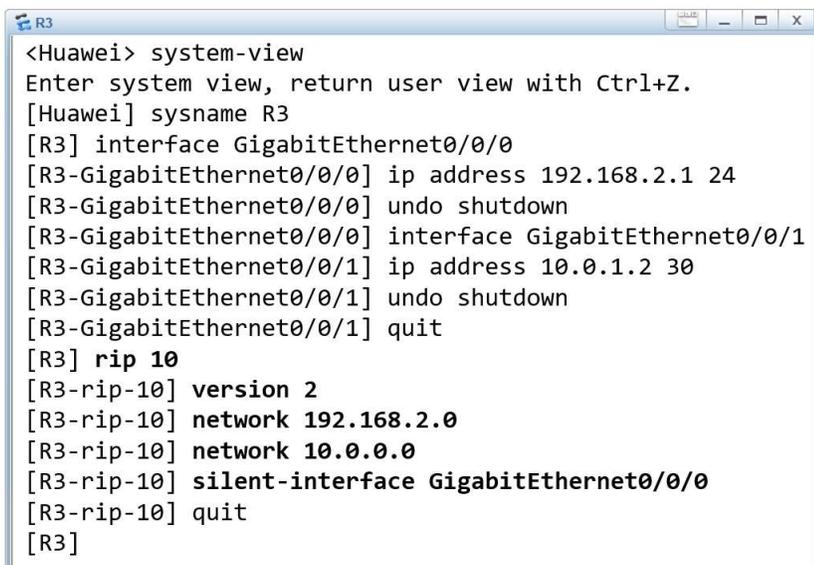
Figura 71. Configuración de R2



```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R2
[R2] interface GigabitEthernet0/0/0
[R2-GigabitEthernet0/0/0] ip address 10.0.0.1 30
[R2-GigabitEthernet0/0/0] undo shutdown
[R2-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R2-GigabitEthernet0/0/1] ip address 10.0.1.1 30
[R2-GigabitEthernet0/0/1] undo shutdown
[R2-GigabitEthernet0/0/1] quit
[R2] rip 10
[R2-rip-10] version 2
[R2-rip-10] network 10.0.0.0
[R2-rip-10] quit
[R2]
```

Fuente: elaboración propia, empleando eNSP.

Figura 72. Configuración de R3



```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R3
[R3] interface GigabitEthernet0/0/0
[R3-GigabitEthernet0/0/0] ip address 192.168.2.1 24
[R3-GigabitEthernet0/0/0] undo shutdown
[R3-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R3-GigabitEthernet0/0/1] ip address 10.0.1.2 30
[R3-GigabitEthernet0/0/1] undo shutdown
[R3-GigabitEthernet0/0/1] quit
[R3] rip 10
[R3-rip-10] version 2
[R3-rip-10] network 192.168.2.0
[R3-rip-10] network 10.0.0.0
[R3-rip-10] silent-interface GigabitEthernet0/0/0
[R3-rip-10] quit
[R3]
```

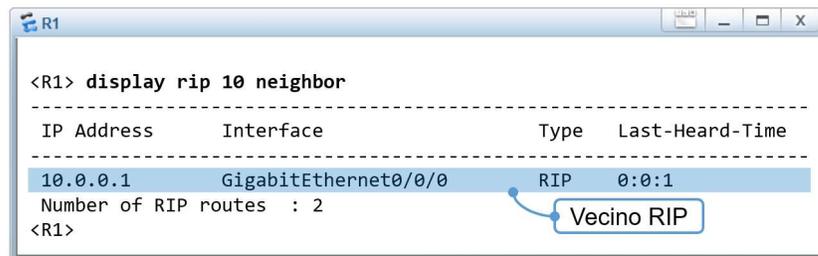
Fuente: elaboración propia, empleando eNSP.

2.6.5. Verificando la configuración de RIP

Para comprobar la correcta implementación de RIP se pueden emplear diversos comandos, dos de los más importantes son: *display rip process_id neighbor*, para verificar las interfaces por las que se conocen a los vecinos RIP, y *display rip process_id*, para visualizar la configuración de RIP que se está ejecutando.

En la figura 73 se muestran los *routers* vecinos de R1 que también están ejecutando RIP.

Figura 73. Vecinos RIP de R1



```
<R1> display rip 10 neighbor
-----
IP Address      Interface      Type      Last-Heard-Time
-----
10.0.0.1        GigabitEthernet0/0/0    RIP      0:0:1
Number of RIP routes : 2
<R1>
```

Fuente: elaboración propia, empleando eNSP.

En la figura 74 se muestra la configuración de RIP que se está ejecutando en R1.

Figura 74. Configuración RIP de R1

```
<R1> display rip 10
Public VPN-instance
RIP process : 10
  RIP version : 2 ①
  Preference : 100 ②
  Checkzero : Enabled
  Default-cost : 0
  Summary : Enabled
  Host-route : Enabled
  Maximum number of balanced paths : 4 ③
  Update time : 30 sec
  Age time : 180 sec ④
  Garbage-collect time : 120 sec
  Graceful restart : Disabled
  BFD : Disabled
  Silent-interfaces : ⑤
  GigabitEthernet0/0/1
  Default-route : Disabled
  Verify-source : Enabled
  Networks : ④
  10.0.0.0 192.168.1.0
  Configured peers : None
  Number of routes in database : 5
  Number of interfaces enabled : 2
  Triggered updates sent : 3
  Number of route changes : 5
  Number of replies to queries : 1
  Number of routes in ADV DB : 4
<R1>
```

- | | |
|--------------------------------------|--------------------------|
| ① Versión | ④ Timers |
| ② Preferencia | ⑤ Interfaces silenciadas |
| ③ Máximo número de rutas balanceadas | ⑥ Interfaces anunciadas |

Fuente: elaboración propia, empleando eNSP.

El último paso es verificar que en la tabla de enrutamiento exista una ruta para alcanzar todas las redes de la topología, En la figura 75 se muestra la tabla de enrutamiento de R1.

Figura 75. **Tabla de enrutamiento de R1**

```

<R1> display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 12      Routes : 12

Destination/Mask    Proto    Pre    Cost    Flags NextHop          Interface
-----
 10.0.0.0/30        Direct  0      0        D   10.0.0.2          GigabitEthernet0/0/0
 10.0.0.2/32        Direct  0      0        D   127.0.0.1         GigabitEthernet0/0/0
 10.0.0.3/32        Direct  0      0        D   127.0.0.1         GigabitEthernet0/0/0
 10.0.1.0/30        RIP     100    1        D   10.0.0.1          GigabitEthernet0/0/0
 127.0.0.0/8        Direct  0      0        D   127.0.0.1         InLoopBack0
 127.0.0.1/32       Direct  0      0        D   127.0.0.1         InLoopBack0
127.255.255.255/32 Direct  0      0        D   127.0.0.1         InLoopBack0
 192.168.1.0/24     Direct  0      0        D   192.168.1.1       GigabitEthernet0/0/1
 192.168.1.1/32     Direct  0      0        D   127.0.0.1         GigabitEthernet0/0/1
 192.168.1.255/32   Direct  0      0        D   127.0.0.1         GigabitEthernet0/0/1
 192.168.2.0/24     RIP     100    2        D   10.0.0.1          GigabitEthernet0/0/0
255.255.255.255/32 Direct  0      0        D   127.0.0.1         InLoopBack0
  
```

Fuente: elaboración propia, empleando eNSP.

Finalmente, para validar que exista comunicación bidireccional entre las dos redes LAN, se realiza un *ping* desde la PC1 hasta la PC3. Adicionalmente se puede realizar una traza desde PC1 hasta PC3 para visualizar la ruta que está tomando el paquete para llegar a su destino.

Figura 76. **Ping de PC1 a PC3**

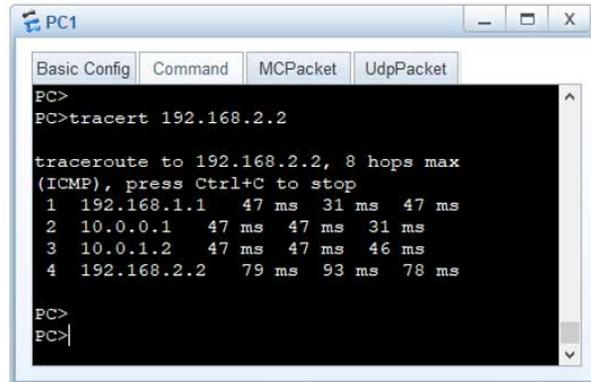
```

PC1>PING 192.168.2.2
Ping 192.168.2.2: 32 data bytes, Press Ctrl_C to break
From 192.168.2.2: bytes=32 seq=1 ttl=125 time=94 ms
From 192.168.2.2: bytes=32 seq=2 ttl=125 time=93 ms
From 192.168.2.2: bytes=32 seq=3 ttl=125 time=110 ms
From 192.168.2.2: bytes=32 seq=4 ttl=125 time=78 ms
From 192.168.2.2: bytes=32 seq=5 ttl=125 time=94 ms

--- 192.168.2.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 78/93/110 ms
PC1>
  
```

Fuente: elaboración propia, empleando eNSP.

Figura 77. Traza de PC1 a PC3



```
PC1
Basic Config  Command  MCPacket  UdpPacket
PC>
PC>tracert 192.168.2.2

tracert to 192.168.2.2, 8 hops max
(ICMP), press Ctrl+C to stop
 1  192.168.1.1    47 ms  31 ms  47 ms
 2  10.0.0.1     47 ms  47 ms  31 ms
 3  10.0.1.2     47 ms  47 ms  46 ms
 4  192.168.2.2   79 ms  93 ms  78 ms

PC>
PC>
```

Fuente: elaboración propia, empleando eNSP.

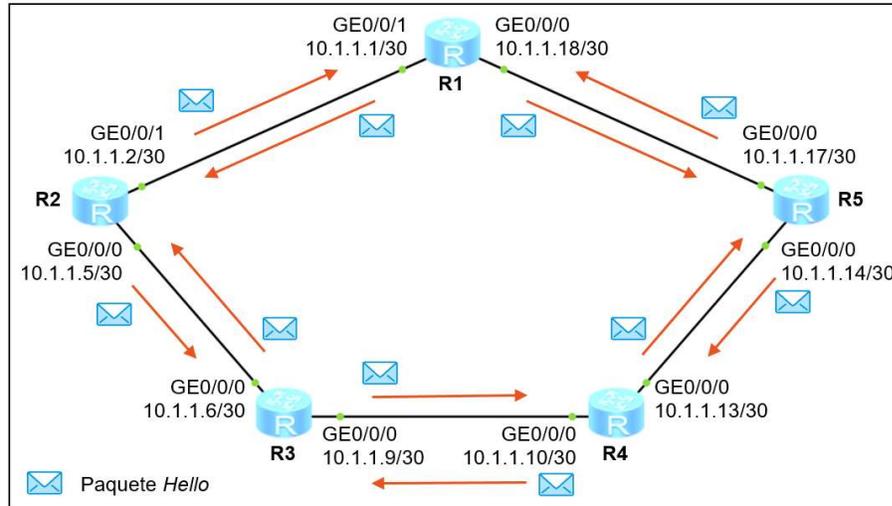
2.7. Open Shortest Path First

OSPF es el protocolo de enrutamiento dinámico más utilizado por los ISP. Se considera un protocolo IGP de estado de enlace ya que construye un mapa de toda la topología de red y en base a ella encuentra la mejor ruta. A diferencia de RIP que únicamente construye la tabla de enrutamiento, OSPF crea tres: la tabla de topología, tabla de enrutamiento y tabla de adyacencias. Cuenta con una preferencia de 10 y determina el costo en base al ancho de banda de la ruta completa. OSPF es un protocolo que trabaja en la capa de red Internet del modelo TCP/IP.

2.7.1. Principio de operación

Para analizar el funcionamiento de OSPF se tomará la misma topología empleada en el análisis de RIP. Luego de habilitar el protocolo en todos los *routers* de la topología, cada uno inicia enviando paquetes llamados Hello cada 30 segundos a través de sus interfaces habilitadas, como se muestra en la figura 78.

Figura 78. Paquetes Hello enviados

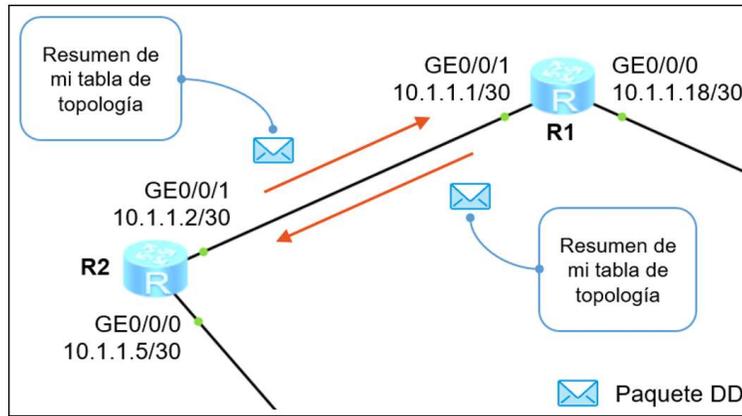


Fuente: elaboración propia, empleando eNSP.

El paquete Hello tiene el propósito de descubrir *routers* que también estén implementando OSPF y así crear y mantener la relación de vecindad con ellos. Para el siguiente ejemplo, luego que R1 reciba el paquete Hello de R2 y R2 reciba el paquete Hello de R1, estos se convierten en vecinos OSPF. Esta misma situación ocurre con todos los *routers* de la topología hasta que, progresivamente, todos formen relación de vecindad con los *routers* a los cuales están conectado.

Luego de formar relación de vecindad, R1 envía un paquete especial llamado Database Description o DD que contiene un resumen de su tabla de topología o LSDB hacia R2 y R5, para fines de análisis en la figura 79 solo se observa el *link* entre R1 y R2. La tabla de topología es un mapa completo de toda la red y contiene información sobre cómo llegar a todos los segmentos existentes.

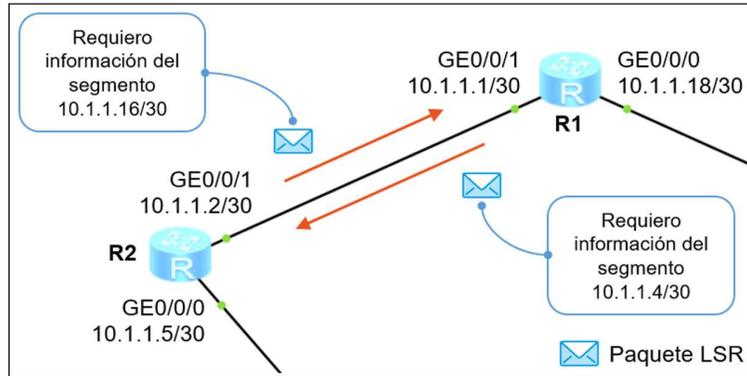
Figura 79. **Comparación del resumen de la LSDB de R1 y R2**



Fuente: elaboración propia, empleando eNSP.

En este punto, R1 únicamente contiene rutas directas en su tabla de topología así que envía un resumen de esa información. R2 recibe este resumen y lo compara con el propio resumen de su propia tabla de topología, si ambos resultan ser iguales significa que conocen cómo llegar a las mismas redes y están sincronizadas, sin embargo, R2 también contiene únicamente sus propias rutas directas por lo que los resúmenes de las tablas de topología no son iguales. De esta forma, R2 procede a enviar un paquete llamado Link State Request o LSR a R1 solicitando la información faltante para igualar las tablas de topología, como se muestra en la figura 80.

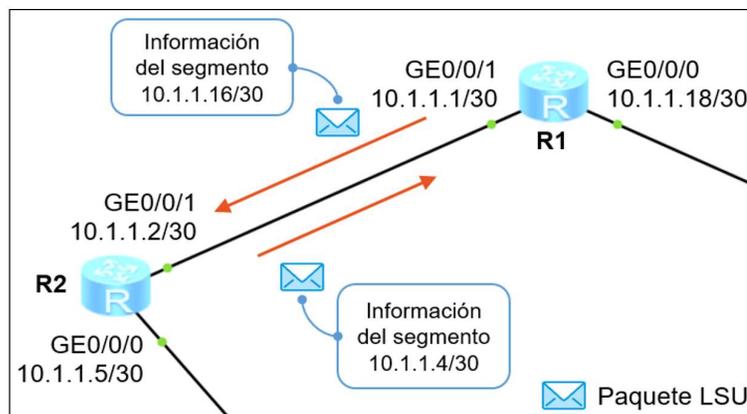
Figura 80. **Solicitud de actualización para sincronizar la LSDB**



Fuente: elaboración propia, empleando eNSP.

Para responder a la solicitud LSR de R2, R1 devuelve un paquete llamado Link State Update o LSU hacia R1 el cual contiene la información solicitada, como se observa en la figura 81. Luego de intercambiar la información se igualan las tablas de topología y se llega a un estado llamado convergencia, es únicamente en este estado que se consigue la adyacencia OSPF entre dos *routers*. Este mismo proceso se repite en todos los *routers* de la topología hasta que todos sincronicen su tabla de topología.

Figura 81. **Envío de actualización para sincronizar la LSDB**



Fuente: elaboración propia, empleando eNSP.

Para mantener actualizada la tabla de topología, los *routers* envían periódicamente su propio resumen a sus vecinos. A diferencia de RIP que envía toda la información, OSPF solamente envía una pequeña parte para lograr una rápida convergencia y optimizar el ancho de banda. Otras de las grandes ventajas de OSPF es que puede crear áreas para agrupar *routers*. Las áreas tienen el propósito de marcar una frontera a las actualizaciones para que afecten únicamente a los *routers* agrupados en ella y no a todos en la topología.

El estudio a profundidad de OSPF debe empezar haciendo énfasis en el término Link State o LS, cuya traducción literal es estado de enlace, y hace referencia al estado de la interfaz de un *router*.

Un *router* ejecutando OSPF realmente anuncia el estado de sus propias interfaces y la de los otros *routers* y no propiamente las rutas, recordando que el único protocolo que puede anunciar rutas es BGP. Es así que los *routers* recolectan la información del estado de todas las interfaces o LS y crean una base de datos llamada *Link State DataBase* o LSDB.

Dentro de la LSDB se encuentra información que describe las conexiones entre los *routers* y el tipo y costo de las interfaces que conforman estas conexiones. Luego de crear la LSDB se ejecuta el algoritmo Dijkstra SPF para crear el mapa de la topología. Es importante mencionar que la LSDB tiene alcance a nivel de área OSPF, así que se puede considerar un mapa de la topología únicamente dentro del área. Una vez se tenga el mapa de la topología se elige la mejor o mejores rutas hacia cada segmento de red y se inyectan a la tabla de enrutamiento. Por defecto, OSPF puede albergar hasta un máximo de 4 rutas para balanceo de cargas en su tabla de enrutamiento, pero este valor puede variar dependiendo del modelo del *router*.

2.7.2. Tipos de paquetes OSPF

El funcionamiento de OSPF se basa en cinco tipos de paquetes que buscan crear y mantener la relación de vecindad y adyacencia, así como distribuir la información del estado de las LS de los *routers*. A continuación, se brinda una pequeña descripción del propósito de cada uno.

Tabla IX. Tipos de paquetes OSPF

Tipo de paquete	Definición
Hello	Los paquetes Hello son enviados periódicamente por el <i>router</i> para descubrir <i>routers</i> vecinos, además de crear y mantener la adyacencia OSPF.
<i>Database Description o DD</i>	Los paquetes DD contienen un resumen de la LSDB del <i>router</i> emisor. Son enviados para compararse con el resumen de la LSDB de los vecinos y así comprobar si está actualizada.
<i>Link State Request o LSR</i>	Los paquetes LSR son empleados para solicitar información de una entrada específica de la LSDB y son enviados para solicitar la información faltante cuando la LSDB de un <i>router</i> no está sincronizada con la de su vecino.
<i>Link State Update o LSU</i>	Los paquetes LSU son empleados a modo de respuesta de los LSR y transportan LSAs que son los bloques que construyen la LSDB. Dentro de los LSU se envía la información necesaria para sincronizar la LSDB.
<i>Link State Acknowledgment o LSAck</i>	Los paquetes LSAck son un tipo de acuse de recibo. Son enviados para confirmar que el paquete LSU fue recibido con éxito.

Fuente: elaboración propia.

2.7.3. Encapsulación de los paquetes OSPF

En la figura 82, se visualiza la estructura del mensaje que emplea OSPF para enviar los cinco tipos de paquetes.

Figura 82. Encapsulación de los paquetes OSPF

Header de la trama	Header del paquete IP	Header del paquete OSPF	Tipo de paquete OSPF
Dirección MAC Origen: MAC de la interfaz emisora Dirección MAC Destino: Multicast 0100-5E00-0005 o 0100-5E00-0006			
Dirección IP Origen: IP de la interfaz emisora Dirección IP Destino: Multicast 224.0.0.5 o 224.0.0.6 Protocolo: 89 para OSPF			
Router ID Área ID Type Code para el tipo de paquete OSPF			
			Paquete Hello Paquete DD Paquete LSR Paquete LSU Paquete LSAck

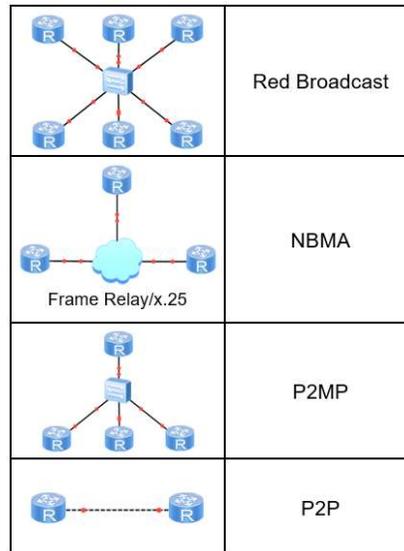
Fuente: elaboración propia.

2.7.4. Tipos de redes OSPF

OSPF se puede implementar en varios tipos de redes, entre ellas están las redes Broadcast como Ethernet e FDDI, redes No-Broadcast Multi-Acceso o NBMA como Frame Relay, ATM y X.25, redes Punto a Multi-Punto o P2MP, que es independiente al protocolo de la capa de estado de enlace, y redes Punto a Punto o P2P como Point to Point Protocol y HDLC.

En la figura 83, se muestra un bosquejo de los tipos de redes donde se puede implementar OSPF.

Figura 83. Tipos de redes en OSPF



Fuente: elaboración propia, empleando eNSP.

2.7.5. Routers vecinos

La creación de vecinos OSPF se basa en el intercambio de paquetes Hello entre dos *routers*. Un paquete Hello transporta información de la interfaz del *router* emisor. El paquete es destinado a la dirección de Multicast 224.0.0.5 para llegar a todos los *routers* que estén ejecutando OSPF dentro del dominio Broadcast. Cuando algún *router* recibe este paquete Hello, lo primero que realiza es una comparación entre los parámetros recibidos en el paquete y los parámetros configurados en la interfaz receptora.

Existen cinco parámetros que deben coincidir entre el *router* emisor y receptor para que se cree la relación de vecindad OSPF, estos parámetros son:

- Intervalo Hello
- Intervalo Dead

- Tipo de red y máscara de subred de las interfaces
- Área ID
- Contraseña de autenticación

Solamente si estos parámetros coinciden entonces el *router* emisor del paquete Hello será vecino del *router* receptor. La relación de vecindad se completa cuando ambos *routers* hayan intercambiado paquetes Hello cuyos cinco parámetros sean iguales.

A continuación, se brinda una breve descripción de algunos de los parámetros que transporta el paquete Hello.

- Versión de OSPF: OSPFv2 se emplea como protocolo de enrutamiento de IPv4 y OSPFv3 se emplea para IPv6.
- Router ID: El Router ID es un número en formato IPv4 cuya función es identificar al *router*. Este valor debe ser único para no causar conflicto con otros equipos y puede ser configurado de forma automático o manual. El proceso para su elección es el siguiente:
 - El Router ID puede ser configurado manualmente empleando el comando: *ospf router-id ip_address*, donde el parámetro *ip_address* es el Router ID en formato IPv4.
 - Si no se especifica manualmente entonces el *router* elige como Router ID la dirección IPv4 de mayor valor de las interfaces Loopback activas.
 - Si no se tiene ninguna dirección Loopback entonces el *router* elige como Router ID la dirección IPv4 de mayor valor de las interfaces activas.

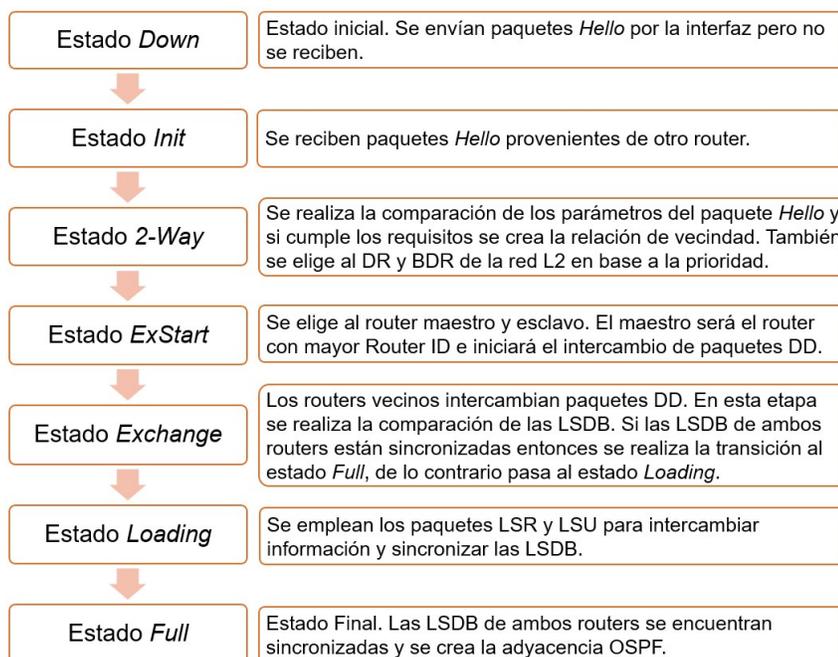
- Área ID: Número entero que especifica el área donde reside la interfaz del *router*.
- Intervalo Hello: Período con que se envían los paquetes Hello por la interfaz. Por defecto, en redes P2P o Broadcast los paquetes Hello se envían cada 10 segundos, mientras que en redes NBMA o P2MP se envían cada 30 segundos. Para modificar este valor se debe ingresar a la vista de la interfaz y emplear el comando: *ospf timer hello interval*, donde el parámetro *interval* es el período al cual se enviarán los paquetes Hello, puede variar entre 1 y 65 535 segundos.
- Intervalo Dead: Período de tiempo que debe transcurrir sin que se reciban paquetes Hello de un vecino para considerarlo inalcanzable y, consecuentemente, perder la relación de vecindad. Por lo general es cuatro veces el intervalo Hello, es decir, de 30 segundos para redes P2P y Broadcast y 120 segundos para redes NBMA y P2MP. Cuando se recibe un paquete Hello se inicia una cuenta regresiva de 30 segundos, en el caso de redes Broadcast, si este temporizador llega a 0 y no se recibe otro paquete Hello entonces se elimina la relación de vecindad, por el contrario, si se recibe un paquete Hello entonces el intervalo Dead es reiniciado a su valor máximo. Para modificar este valor se debe ingresar a la vista de la interfaz y emplear el comando: *ospf timer dead interval*, donde el parámetro *interval* es el tiempo que deben transcurrir para considerar inalcanzable al vecino, puede variar entre 1 y 235 926 000 segundos.
- Tipo de red y máscara de subred de la interfaz emisora: Los tipos de red más conocidos son P2P, Broadcast, NBMA y P2MP.
- Prioridad de la interfaz emisora: Se emplea para la elección del DR y BDR.

- Contraseña para autenticación: Únicamente si se está implementando autenticación.
- DR de la red Layer 2: Router ID del DR de la red L2 donde reside la interfaz del *router* emisor.
- BDR de la red Layer 2: Router ID de BDR de la red L2 donde reside la interfaz del *router* emisor.

2.7.6. Adyacencia OSPF

La adyacencia OSPF entre dos *routers* se da cuando ambos sincronizan completamente su LSDB. Para esto cada *router* debe cumplir una serie de pasos como se describe en la figura 84.

Figura 84. **Proceso para formar adyacencia OSPF**



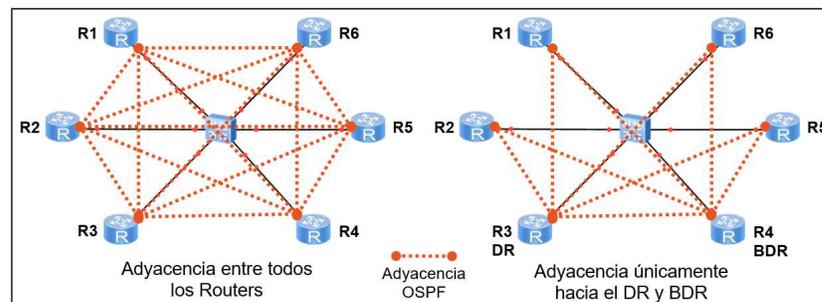
Fuente: elaboración propia.

2.7.7. OSPF en redes Broadcast y NBMA

Los paquetes Hello son destinados a la dirección Multicast 224.0.0.5 para llegar a todos los *routers* que estén ejecutando OSPF. Esta acción causa conflicto en redes Broadcast o NBMA porque se forma adyacencia entre todos los *routers* de la red.

Como se muestra en la figura 85, en una red Broadcast con seis *routers* conectados a un *switch* central, se formarán cinco adyacencias por cada *router* lo cual causará tormentas de Broadcast y de paquetes Hello en la red L2.

Figura 85. **Adyacencia OSPF en una red Broadcast**



Fuente: elaboración propia, empleando eNSP.

Para mitigar este problema, OSPF elige un Designated Router o DR y un Backup Designated Router o BDR para que todos los *routers* formen adyacencia únicamente con ellos dos. Así, para el ejemplo anterior, cada *router* tendrá únicamente 2 adyacencias en lugar de 5. Evidentemente se exceptúa al DR y BDR.

2.7.7.1. DR Y BDR

Para evitar que todos los *routers* formen adyacencia entre sí en una red Broadcast, OSPF elige un DR que será el responsable de centralizar todas las actualizaciones de OSPF en la red. El DR forma adyacencia con todos los *routers* asegurando que cualquier cambio en la LSDB le llegue primero a él, quien luego se encarga de informar al resto de *routers* de la topología.

El DR tiene un equipo de respaldo denomina BDR que entra en funcionamiento solamente si el DR falla. Todos los *routers* de la topología forman adyacencia OSPF con el DR y BDR quienes recolectan las actualizaciones de la topología, no obstante, es únicamente el DR quien se encarga de anunciar las actualizaciones al resto de equipos, el BDR solamente las conserva y las anuncia si el DR se vuelve inalcanzable.

La elección del DR y BDR se hace en base a la prioridad de la interfaz que conecta hacia la red L2 y puede ser un valor entero comprendido entre 0 y 255. El valor 0 indica que el *router* no puede ser electo como DR o BDR a través de esa interfaz y el valor 255 indica la máxima prioridad, lo que se traduce a una mayor probabilidad de ser electo como DR.

OSPF maneja un valor de prioridad por defecto de 1 en todas las interfaces. El *router* con la prioridad de mayor valor se convierte en el DR y el *router* con la segunda prioridad de mayor valor se convierte en el BDR. Si todos los *routers* tiene la misma prioridad se toma como criterio de desempate el valor del Router ID, siguiendo la misma metodología.

Para establecer la prioridad de una interfaz se ingresa a la vista de la interfaz y se emplea el comando: `ospf dr-priority priority`, donde el parámetro *priority* es la

prioridad. Para regresar la prioridad a su valor por defecto de 1, se emplea el comando: *undo ospf dr-priority*.

2.7.8. Costo de una ruta

OSPF toma como base el ancho de banda para encontrar la ruta más corta hacia un destino. El costo de una ruta es la suma de todos los costos individuales de las interfaces de salida por los que pasa un paquete en su camino hacia su destino. Para encontrar el costo de una interfaz se emplea la fórmula mostrada en la figura 86.

Figura 86. Ecuación para calcular el costo de una interfaz

$$Costo_{int} = \frac{10^8}{BW}$$

Fuente: elaboración propia.

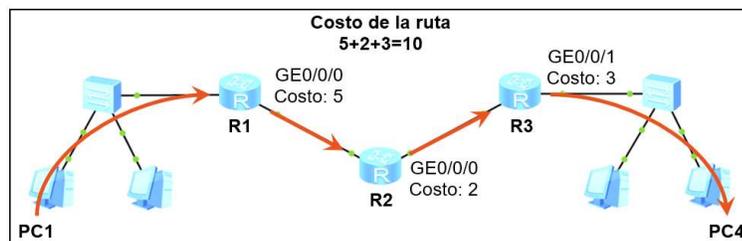
Donde el valor 10^8 es el ancho de banda de referencia y el parámetro *BW* es el ancho de banda configurado en la interfaz de salida, en bits por segundo. De esta forma, una interfaz FastEthernet, que trabaja a 100 Mbps, tendría un costo de 1 y una interfaz GigabitEthernet, que trabaja a 1 000 Mbps, tendría un costo de 0,1. No obstante, OSPF no maneja costos con puntos decimales así que se aproxima al valor entero más cercano, es decir 1.

Por defecto, cuando se implementa OSPF en *routers* Huawei, estos asignan un costo de 1 a todas las interfaces sin importar el ancho de banda. Para habilitar la opción del uso de la fórmula para el cálculo del costo de cada interfaz se debe ingresar a la vista del proceso OSPF y emplear el comando: *bandwidth-config enable*.

El ancho de banda de referencia utilizado en la fórmula de la figura 86, puede ser modificada ingresando a la vista del proceso OSPF y se empleando el comando: *bandwidth-reference bw*, donde el parámetro *bw* es el ancho de banda de referencia en Mbps. Es importante mencionar que el cambio del ancho de banda de referencia debe ser a nivel de toda la topología de red. El ancho de banda de una interfaz puede ser modificado ingresando a la vista de la interfaz y se empleando el comando: *bandwidth bw*, donde el parámetro *bw* es el ancho de banda en bits por segundo. Es importante recalcar que la configuración del ancho de banda de una interfaz no es una limitante ni una restricción en la velocidad de transmisión del enlace, en este caso el término enlace hace referencia al tramo que conecta dos equipos. Finalmente, otra opción es configurar un costo explícito en cada interfaz de salida, para esto se ingresa a la vista de la interfaz y se emplea el comando: *ospf cost cost*, donde el parámetro *cost* es el costo deseado que varía entre 1 y 65 535.

El costo de la ruta completa es la suma de los costos individuales de las interfaces de salida por donde viaja el paquete hacia su destino. En la figura 87, se muestra un ejemplo para determinar el costo de una ruta.

Figura 87. Costo de la ruta en OSPF



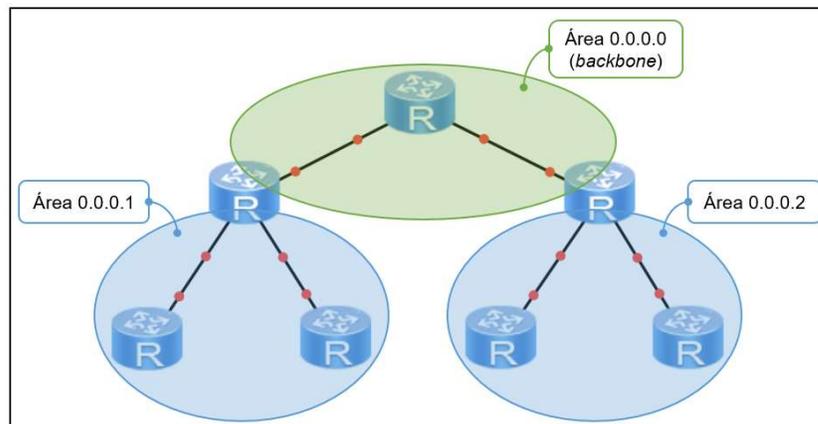
Fuente: elaboración propia.

2.7.9. Áreas en OSPF

La LSDB de un *router* contiene toda la información necesaria para crear un mapa completo de la topología. Sin embargo, cuando se implementa topología con una gran cantidad de *routers*, la LSDB se vuelve compleja y difícil de manipular debido a su gran tamaño. Es por esta razón que OSPF permite crear áreas para agrupar *routers* y así poder construir LSDB's de la topología dentro de cada área y no de la red completa. Esto beneficia enormemente a la reducción del tamaño de la LSDB y, consecuentemente, al uso efectivo de los recursos de procesamiento de cada *router* y del ancho de banda de cada enlace o *link*.

Toda implementación de OSPF nace con el área cero, también denominada *área backbone*, que es el área central de la topología. Cualquier área extra que se desee crear debe estar conectada únicamente al *área backbone* para evitar crear bucles de capa tres.

Figura 88. Áreas OSPF en una red.



Fuente: elaboración propia, empleando eNSP.

Para indicar el área al cual pertenece un *router* se debe hacer a nivel de interfaz, esto es debido a que un *router* puede tener interfaces que pertenezcan a distintas áreas, como se puede visualizar en la figura 88. De hecho, uno de los parámetros del paquete Hello que debe ser igual en ambos *routers* para ser vecinos es el Area ID, que es el número que identifica el área.

Para indicar el área al cual pertenece la interfaz de un *router* se debe ingresar a la vista del proceso ospf y emplear el comando: *área, area_id*, donde el parámetro *area_id* es el número identificador del área, que puede ser un número entero de 32 bits comprendido entre 0 y 4 294 967 295 o puede representar en formato de dirección IPv4. Luego de crear el área se deben agregar las interfaces empleando el comando: *network ip_address wild_card_mask*, donde el parámetro *ip_address* es la dirección de red del segmento configurado en la interfaz y el parámetro *wild_card_mask* es la máscara *wild card*. En este caso el comando *network* tiene la función de habilitar la interfaz para que envíe paquete Hello, indicar que el estado de esa interfaz o LS debe ser anunciado a los vecinos y agregar la interfaz al área seleccionada.

2.7.10. Tipos de *routers* OSPF

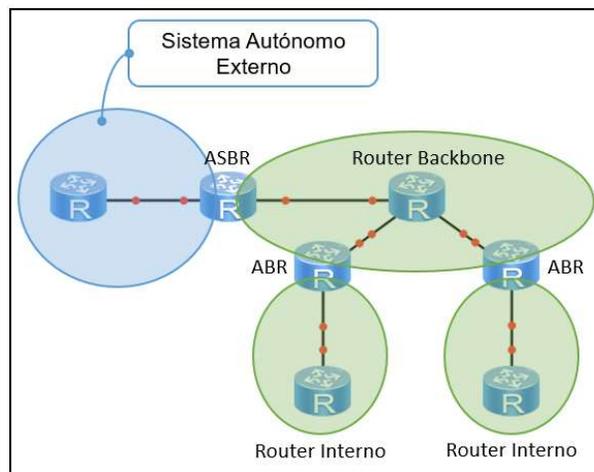
Se tienen cuatro tipos de *routers* que reciben su nombre en base a la posición que ocupan en la topología. A continuación, se brinda una breve descripción de cada uno de ellos.

- Router Interno: todas las interfaces pertenecen a la misma área.
- Area Border Router o ABR: es el encargado de la comunicación hacia el área backbone. Posee una interfaz en el área 0 y al menos otra en

cualquier otra área. Cuenta con una LSDB por cada área a la que pertenecen.

- Router Backbone: contiene al menos una interfaz en el área 0, incluye a los ABR y ASBR.
- Autonomous System Boundary Router o ASBR: es el encargado de la comunicación hacia un sistema autónomo externo.

Figura 89. Tipos de *routers* en OSPF.



Fuente: elaboración propia, empleando eNSP.

2.7.11. Tipos de rutas OSPF

Cuando un *router* inyecta rutas aprendidas por OSPF en su tabla de enrutamiento, estas pueden provenir de la misma área o de un área distinta de donde se encuentra el *router*. Se tienen cuatro tipos de rutas que puede aprender un *router* por OSPF.

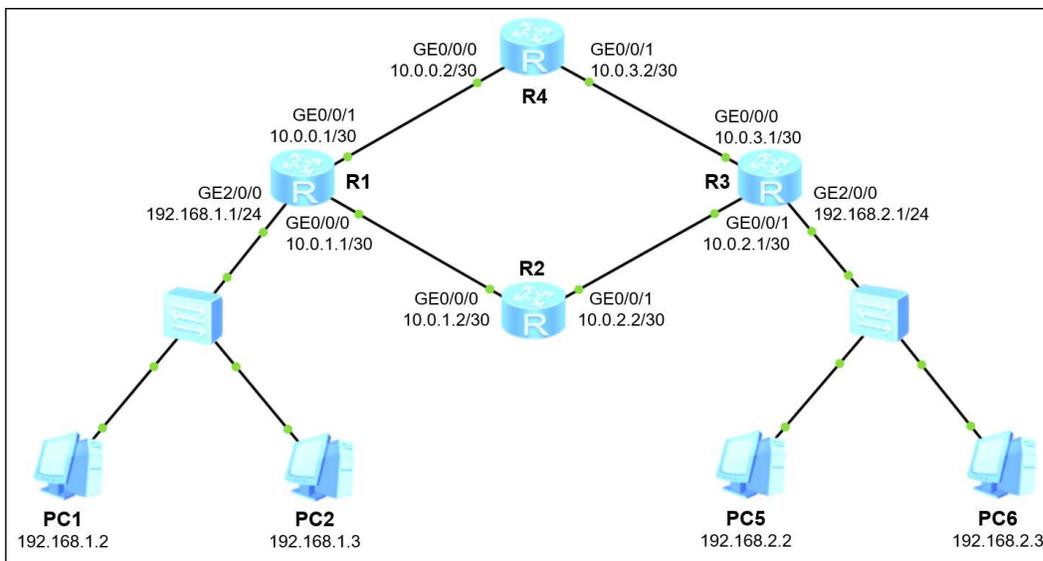
- Ruta Intra-área: rutas aprendidas por *routers* en la misma área.

- Ruta Inter-área: rutas aprendidas por *routers* en áreas distintas.
- Ruta Externa Tipo 1.
- Ruta Externa Tipo 2.

2.7.12. Implementando OSPF área única

En la figura 90, se muestra la topología propuesta para la implementación de OSPF de área única.

Figura 90. Topología propuesta para implementar OSPF área única



Fuente: elaboración propia, empleando eNSP.

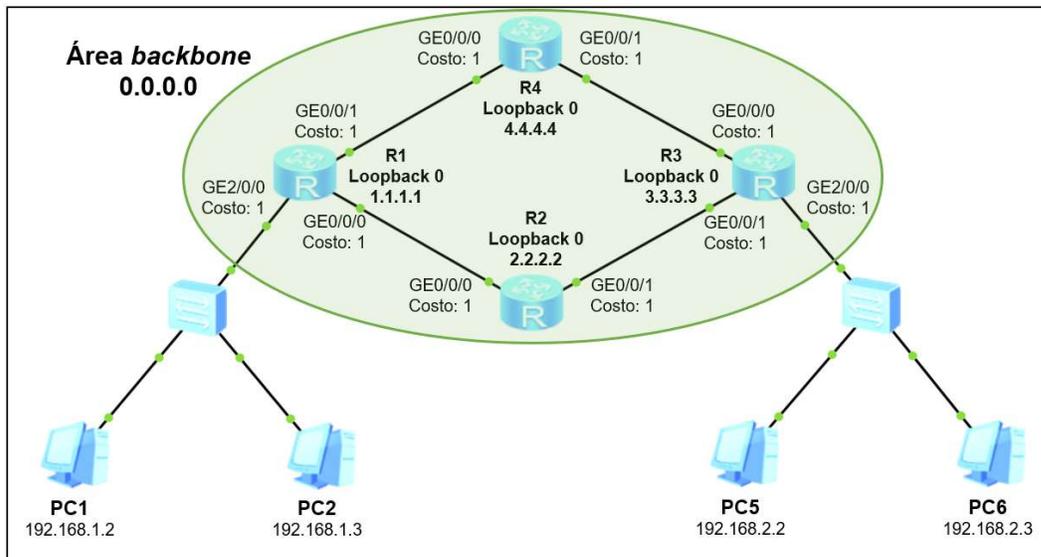
El inicio de la configuración parte activando OSPF en el *router*, para esto se ingresa a la vista del sistema y se emplea el comando: `ospf process_id router-id router_id`, donde el parámetro *process_id* es un numero entero comprendido entre 1 y 65 535 que identifica el proceso OSPF y el parámetro *router_id* es el Router ID en formato IPv4.

El proceso OSPF únicamente tiene importancia a nivel local, es decir que dos *router* no necesariamente deben tener el mismo número de proceso para formar adyacencia. Así mismo, un *router* puede tener varios procesos funcionando de forma paralela. En cuanto al Router ID, siempre es aconsejable configurar una dirección Loopback con la misma dirección IP para realizar *ping* y determinar si el *router* es alcanzable.

El siguiente paso es crear el área OSPF y agregar las interfaces, para esto se emplea el comando *area area_id* para luego agregar las interfaces con el comando *network ip_address wild_card_mask*. Finalmente, se puede silenciar una interfaz para que no se envíen paquetes Hello por ella empleando el comando: *silent-interface interface*, donde el parámetro *interface* es el tipo y número de la interfaz. Esto se hace por motivos de seguridad y para aprovechar el ancho de banda del enlace.

Como se observa en la topología de la figura 91, existen dos rutas para llegar del segmento 192.168.1.0/24 al 192.168.2.0/24, la primera es R1-R4-R3 y la segunda R1-R2-R3. Si se implementa OSPF con los valores por defecto ambas rutas serán agregadas a la tabla de enrutamiento, ya que cada una posee un costo de 3, y se realizaría balanceo de cargas.

Figura 91. Costos de las interfaces



Fuente: elaboración propia, empleando eNSP.

De la figura 92 a la 95 se muestra la configuración que está siendo ejecutada por los *routers* R1, R2, R3 y R4, respectivamente. Se puede visualizar que se crea una interfaz Loopback 0 en cada *router* y se anuncia con el comando *network* en el proceso OSPF, con el propósito de poder hacer *ping* directamente al Router ID de cada equipo.

Figura 92. Configuración de R1

```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R1
[R1] interface GigabitEthernet0/0/0
[R1-GigabitEthernet0/0/0] ip address 10.0.1.1 30
[R1-GigabitEthernet0/0/0] undo shutdown
[R1-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R1-GigabitEthernet0/0/1] ip address 10.0.0.1 30
[R1-GigabitEthernet0/0/1] undo shutdown
[R1-GigabitEthernet0/0/1] interface GigabitEthernet2/0/0
[R1-GigabitEthernet2/0/0] ip address 192.168.1.1 24
[R1-GigabitEthernet2/0/0] undo shutdown
[R1-GigabitEthernet2/0/0] interface Loopback 0
[R1-LoopBack0] ip address 1.1.1.1 32
[R1-LoopBack0] quit
[R1] ospf 1 router-id 1.1.1.1
[R1-ospf-1] silent-interface GigabitEthernet2/0/0
[R1-ospf-1] area 0
[R1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0] network 10.0.0.0 0.0.0.3
[R1-ospf-1-area-0.0.0.0] network 10.0.1.0 0.0.0.3
[R1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0] quit
[R1-ospf-1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

Figura 93. Configuración de R2

```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R2
[R2] interface GigabitEthernet0/0/0
[R2-GigabitEthernet0/0/0] ip address 10.0.1.2 30
[R2-GigabitEthernet0/0/0] undo shutdown
[R2-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R2-GigabitEthernet0/0/1] ip address 10.0.2.2 30
[R2-GigabitEthernet0/0/1] undo shutdown
[R2-GigabitEthernet0/0/1] interface LoopBack 0
[R2-LoopBack0] ip address 2.2.2.2 32
[R2-LoopBack0] quit
[R2] ospf 1 router-id 2.2.2.2
[R2-ospf-1] area 0
[R2-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0] network 10.0.1.0 0.0.0.3
[R2-ospf-1-area-0.0.0.0] network 10.0.2.0 0.0.0.3
[R2-ospf-1-area-0.0.0.0] quit
[R2-ospf-1] quit
[R2]
```

Fuente: elaboración propia, empleando eNSP.

Figura 94. Configuración de R3

```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R3
[R3] interface GigabitEthernet0/0/0
[R3-GigabitEthernet0/0/0] ip address 10.0.3.1 30
[R3-GigabitEthernet0/0/0] undo shutdown
[R3-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R3-GigabitEthernet0/0/1] ip address 10.0.2.1 30
[R3-GigabitEthernet0/0/1] undo shutdown
[R3-GigabitEthernet0/0/1] interface GigabitEthernet2/0/0
[R3-GigabitEthernet2/0/0] ip address 192.168.2.1 24
[R3-GigabitEthernet2/0/0] undo shutdown
[R3-GigabitEthernet2/0/0] interface LoopBack 0
[R3-LoopBack0] ip address 3.3.3.3 32
[R3-LoopBack0] quit
[R3] ospf 1 router-id 3.3.3.3
[R3-ospf-1] silent-interface GigabitEthernet2/0/0
[R3-ospf-1] area 0
[R3-ospf-1-area-0.0.0.0] network 3.3.3.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0] network 10.0.2.0 0.0.0.3
[R3-ospf-1-area-0.0.0.0] network 10.0.3.0 0.0.0.3
[R3-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0] quit
[R3-ospf-1] quit
[R3]
```

Fuente: elaboración propia, empleando eNSP.

Figura 95. Configuración de R4

```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] sysname R4
[R4] interface GigabitEthernet0/0/0
[R4-GigabitEthernet0/0/0] ip address 10.0.0.2 30
[R4-GigabitEthernet0/0/0] undo shutdown
[R4-GigabitEthernet0/0/0] interface GigabitEthernet0/0/1
[R4-GigabitEthernet0/0/1] ip address 10.0.3.2 30
[R4-GigabitEthernet0/0/1] undo shutdown
[R4-GigabitEthernet0/0/1] interface LoopBack 0
[R4-LoopBack0] ip address 4.4.4.4 32
[R4-LoopBack0] quit
[R4] ospf 1 router-id 4.4.4.4
[R4-ospf-1] area 0
[R4-ospf-1-area-0.0.0.0] network 4.4.4.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0] network 10.0.0.0 0.0.0.3
[R4-ospf-1-area-0.0.0.0] network 10.0.3.0 0.0.0.3
[R4-ospf-1-area-0.0.0.0] quit
[R4-ospf-1] quit
[R4]
```

Fuente: elaboración propia, empleando eNSP.

2.7.13. Verificando la configuración de OSPF área única

Existe una gran cantidad de comandos para verificar el funcionamiento de OSPF. Uno de los más útiles es: *display ospf brief*, el cual permite visualizar el Router ID, la preferencia, la cantidad de áreas configuradas en el *router* y las interfaces que pertenecen a cada una de ellas, así mismo permite ver el tipo, costo, prioridad, intervalo Hello y Dead de cada interfaz y las interfaces silenciadas.

Figura 96. Configuración de OSPF en R1

```

<R1> display ospf brief

      OSPF Process 1 with Router ID 1.1.1.1
      OSPF Protocol Information

RouterID: 1.1.1.1 ①      Border Router: ②
Multi-VPN-Instance is not enabled
Global DS-TE Mode: Non-Standard IETF Mode
Graceful-restart capability: disabled
Helper support capability : not configured
Applications Supported: MPLS Traffic-Engineering
Spf-schedule-interval: max 10000ms, start 500ms, hold 1000ms
Default ASE parameters: Metric: 1 Tag: 1 Type: 2
Route Preference: 10 ③
ASE Route Preference: 150
SPF Computation Count: 16
RFC 1583 Compatible
Retransmission limitation is disabled
Area Count: 1  Nssa Area Count: 0
Exchange/Loading Neighbors: 0
Process total up interface count: 4
Process valid up interface count: 3

Area: 0.0.0.0 ④      (MPLS TE not enabled)
Authtype: None  Area flag: Normal
SPF scheduled Count: 16
Exchange/Loading Neighbors: 0
Router ID conflict state: Normal
Area interface up count: 4 ⑤

Interface: 10.0.1.1 (GigabitEthernet0/0/0)
Cost: 1  State: BDR  Type: Broadcast  MTU: 1500 ⑥
Priority: 1
Designated Router: 10.0.1.2
Backup Designated Router: 10.0.1.1
Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1 ⑦

Interface: 10.0.0.1 (GigabitEthernet0/0/1)
Cost: 1  State: BDR  Type: Broadcast  MTU: 1500
Priority: 1
Designated Router: 10.0.0.2 ⑧
Backup Designated Router: 10.0.0.1
Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

Interface: 1.1.1.1 (LoopBack0)
Cost: 0  State: P-2-P  Type: P2P  MTU: 1500
Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

Interface: 192.168.1.1 (GigabitEthernet2/0/0)
Cost: 1  State: DR  Type: Broadcast  MTU: 1500
Priority: 1 ⑨
Designated Router: 192.168.1.1
Backup Designated Router: 0.0.0.0
Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1
Silent interface, No hellos ⑩

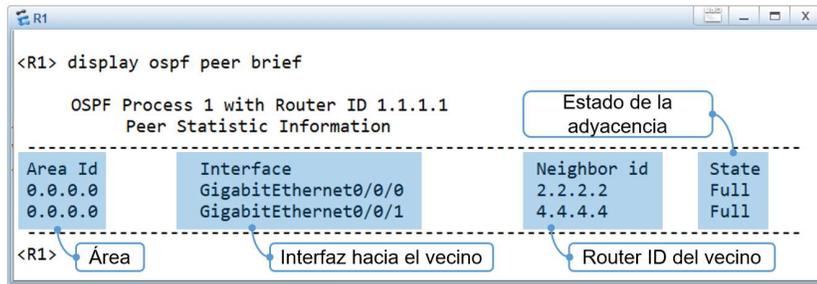
<R1>
    
```

- | | | |
|------------------|-------------------------------------|-----------------------------------|
| ① Router ID | ⑤ Cantidad de interfaces en el área | ⑧ IP del DR y BDR del segmento L2 |
| ② Tipo de router | ⑥ Costo, estado y tipo de interfaz | ⑨ Prioridad de la interfaz |
| ③ Preferencia | ⑦ Timers | ⑩ Interfaces silenciadas |
| ④ Area ID | | |

Fuente: elaboración propia, empleando eNSP.

Para verificar la interfaz por la cual se ha formado adyacencia con un vecino se emplea el comando: *display ospf peer brief*.

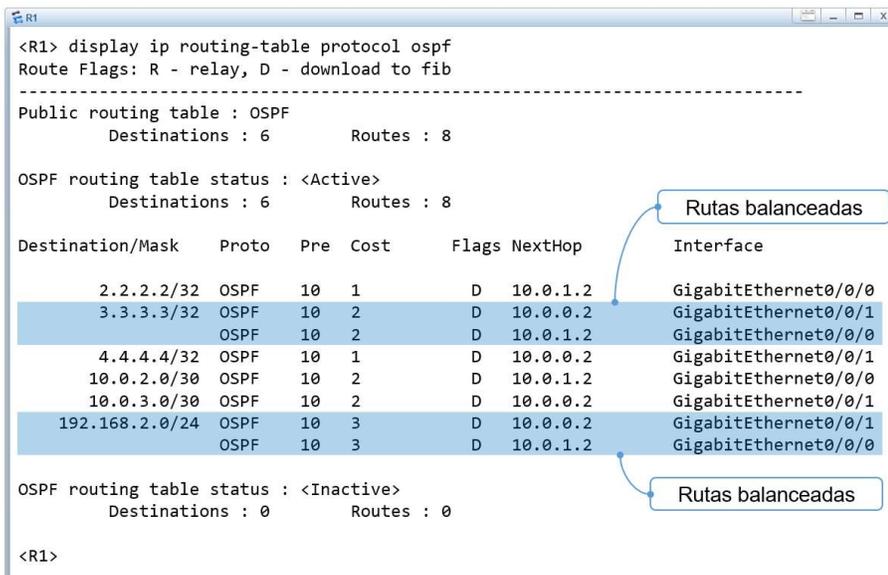
Figura 97. **Adyacencias OSPF de R1**



Fuente: elaboración propia, empleando eNSP.

Para validar que las rutas se hayan inyectado en la tabla de enrutamiento se puede filtrar para observar únicamente las que se hayan aprendido por OSPF emplea el comando: *display ip routing-table protocol ospf*.

Figura 98. **Rutas aprendidas por OSPF de R1**



Fuente: elaboración propia, empleando eNSP.

2.7.14. Diferencia al implementar OSPF multi-área

La única diferencia al implementar OSPF multi-área es que se debe ingresar a la vista del área OSPF con el comando: *area area_id*, e ir agregando las interfaces con el comando: *network*. Recordando que las interfaces que unen a dos *routers* deben pertenecer a la misma área para formar adyacencia.

2.8. VLAN y enrutamiento Inter-VLAN

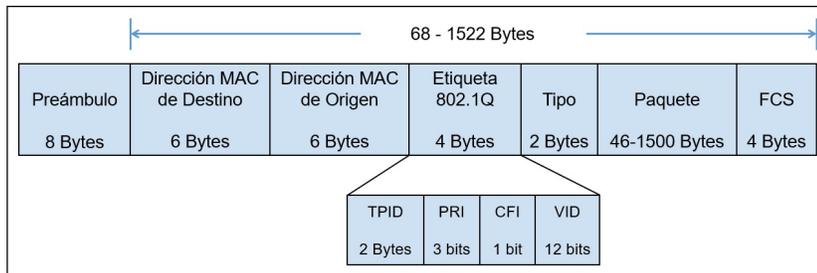
El término VLAN hace referencia a *Virtual Local Area Network* y es una tecnología que se implementa en *switches* para separar dominios de Broadcast. Inicialmente un *switch* Huawei tiene todos sus puertos asignados a la VLAN por defecto, VLAN 1, y es por esa razón que no se requiere configuración previa para que comunique *hosts* conectados a sus puertos.

Un *switch* únicamente puede comunicar equipos que se encuentren en la misma VLAN, mientras que para realizar comunicación entre VLAN se requiere de algún dispositivo de capa tres. Las VLAN están identificadas con un número entero, VLAN ID, entre 1 y 4 094, siendo la VLAN 1 la VLAN por defecto, que no puede ser eliminada o modificada de los *switches*.

2.8.1. Funcionamiento

Para tener la habilidad de manejar VLAN, un *switch* debe realizar una modificación a las tramas que recibe y envía para identificarlas con el ID de la VLAN, VID, este proceso recibe el nombre de etiquetado. El estándar IEEE 802.1q brinda las bases para normalizar el etiquetado de tramas al agregar 4 Bytes a la trama Ethernet, como se muestra en la figura 99.

Figura 99. Trama Ethernet con etiqueta 802.1q



Fuente: elaboración propia, empleando eNSP.

La etiqueta 802.1q se divide en cuatro campos.

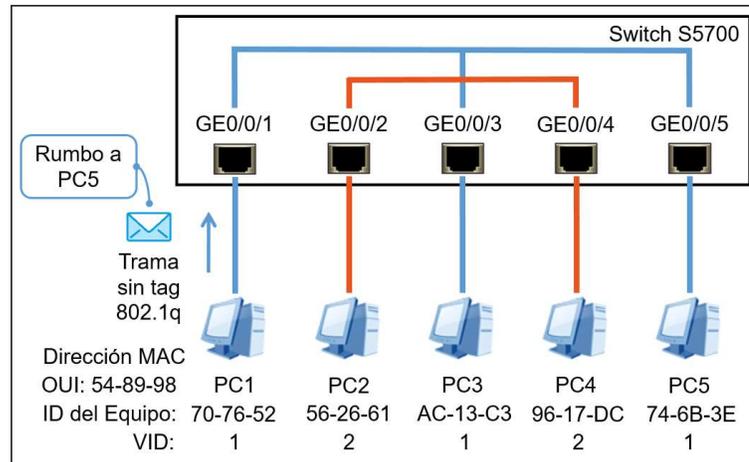
- TPID o *Tag Protocol ID*: Tiene un valor hexadecimal de 0x8100 si la trama maneja etiquetado 802.1q. Cualquier otro valor indica que la trama se maneja sin etiqueta.
- PRI o *Priority*: Tiene un valor de 0 a 7 y especifica la prioridad de la trama. Este campo se utiliza para *Class of Service* o CoS.
- CFI o *Canonical Format Indicator*.
- VID o *VLAN ID*: Tiene un valor de 1 a 4094 e indica el identificador de la VLAN.

Gracias a esta nueva etiqueta, todos los *switches* de una red pueden identificar la VLAN a la que pertenece una trama.

Para analizar el comportamiento del etiquetado de trama se toma como ejemplo un *switch* S5700 con cinco computadoras conectadas a sus puertos, como se muestra en la figura 100.

Para fines didácticos se tomará el *switch* S5700 en lugar del S3700 ya que el primero contiene puertos GigabitEthernet.

Figura 100. **Trama sin etiqueta 802.1q ingresando al *switch***



Fuente: elaboración propia, empleando eNSP.

Las computadoras PC1, PC3 y PC5 pertenecen a la VLAN 1 y las computadoras PC2 y PC4 a la VLAN 2. Con esta configuración las computadoras en la VLAN 1 no tendrán comunicación con las computadoras en la VLAN 2 ya que no hay equipo de capa tres que las comunique.

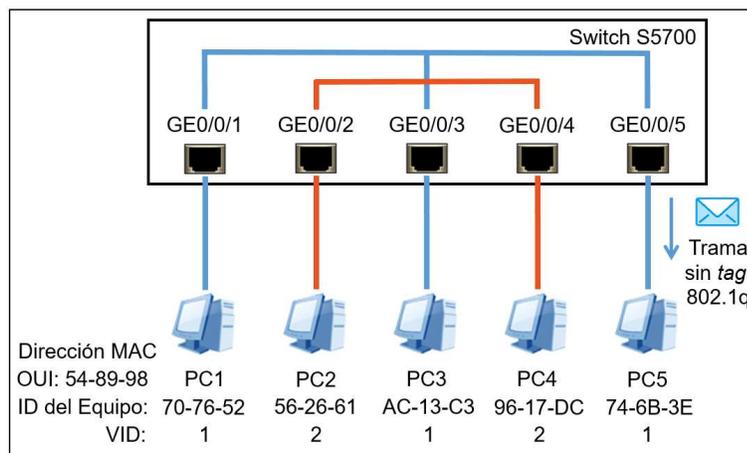
El proceso inicia con PC1 enviando una trama dirigida hacia PC5, como se muestra en la figura 100. En esta ocasión se asumirá que la trama que sale de PC1, e ingresa al puerto GE0/0/1 del *switch*, es una trama Ethernet sin etiqueta, no obstante, algunas computadoras tienen la opción de configurar VLAN y enviar tráfico etiquetado.

En el momento en que la trama ingresa por el puerto GE0/0/1, el *switch* la recibe y le aplica una etiqueta 802.1q con el VID 1 para identificar que proviene

de la VLAN 1. Posteriormente el *switch* revisa la tabla MAC correspondiente a la VLAN 1 para consultar el puerto por el que se llega a la dirección MAC de PC5. El *switch* contiene una tabla MAC por cada VLAN, es por esa razón que si el emisor y receptor no se encuentran en la misma VLAN no habrá comunicación. En este caso PC5 y PC1 pertenecen a la VLAN 1 y la trama etiquetada es dirigida hacia el puerto de salida GE0/0/5. Antes de expulsar la trama por la GE0/0/5, el *switch* retira la etiqueta 802.1q por dos razones, la primera es debido a que se supone que PC5 recibe y envía tramas sin etiqueta, y la segunda es porque la etiqueta tiene la función de identificar a la trama únicamente durante su manipulación en la red de capa dos y ya no es necesaria, en la mayoría de los casos, cuando va rumbo a su destino final. Es importante recordar que existen equipos que aceptan y envían tráfico etiquetado, en estos casos las tramas salen con la etiqueta 802.1q desde el puerto del *switch* hacia destinatario.

En la figura 101 se muestra la trama sin etiqueta 802.1q saliendo del puerto GE0/0/5 del *switch* rumbo a la computadora PC5.

Figura 101. **Trama sin etiqueta saliendo del *switch***



Fuente: elaboración propia, empleando eNSP.

En general, cuando el tráfico viaja sin etiqueta, *untagged*, las tramas tienen el formato Ethernet y cuando el tráfico viaja etiquetado, *tagged*, las tramas cuentan con la etiqueta de la norma IEEE 802.1q. En un *switch*, cuando se reciben o expulsan tramas sin etiqueta por el puerto se dice que está en modo acceso, por el contrario, cuando se reciben o expulsan tramas etiquetadas el puerto está en modo troncal.

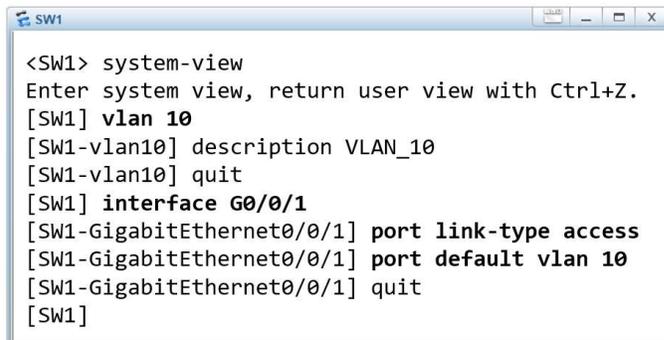
En un *switch* capa dos o capa tres, para ingresar a la vista o crear una VLAN, se debe ingresar a la vista del sistema y se emplea el comando: *vlan vlan_id*, donde el parámetro *vlan_id* corresponde al VID, que varía entre 1 y 4 094. Posteriormente se puede agregar una breve descripción de la VLAN con el comando: *description text*. Una vez se tenga creada la VLAN se deben agregar los puertos físicos, puede ser en modo acceso o troncal, para establecer el nuevo dominio de Broadcast.

2.8.2. Puerto en modo acceso y troncal

El modo acceso permite asignar una sola VLAN al puerto ya que las tramas son recibidas y enviadas sin etiqueta. Esta configuración es mayormente empleada cuando el puerto conecta directamente hacia un *host* final. Para configurar un puerto en modo acceso se ingresa a la vista de la interfaz y se emplea el comando: *port link-type access*. Seguidamente se debe asignar una VLAN al puerto lo cual se realiza empleando el comando: *port default vlan vlan_id*, donde el parámetro *vlan_id* es el identificador de la VID.

En la figura 102, se muestra la configuración del puerto GE0/0/1 en modo acceso para la VLAN 10.

Figura 102. **Configuración de un puerto en modo acceso**



```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] vlan 10
[SW1-vlan10] description VLAN_10
[SW1-vlan10] quit
[SW1] interface G0/0/1
[SW1-GigabitEthernet0/0/1] port link-type access
[SW1-GigabitEthernet0/0/1] port default vlan 10
[SW1-GigabitEthernet0/0/1] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

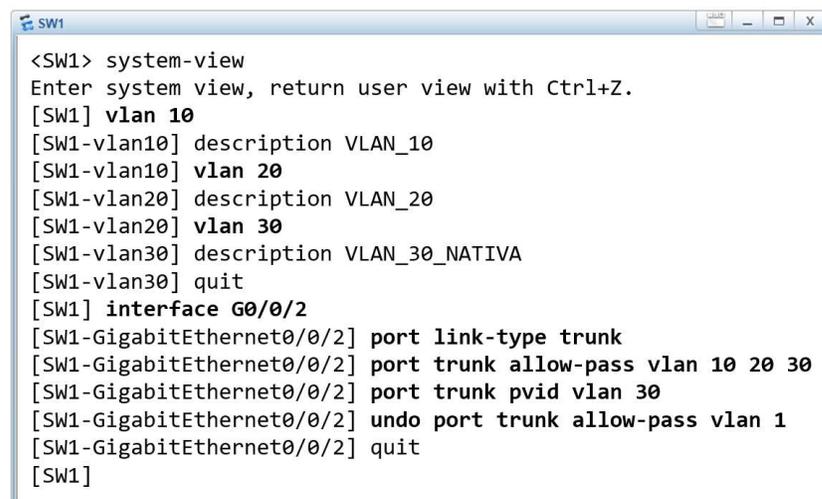
El modo troncal permite agregar más de una VLAN al puerto ya que emplea las etiquetas 802.1q para identificarlas. Generalmente esta configuración se emplea en enlaces que unen dos *switches*, es decir enlaces troncales, para transportar el tráfico de las VLAN permitidas a través de él. Si por el enlace troncal se recibe una trama con la etiqueta de una VLAN que no está permitida entonces el *switch* desecha la trama automáticamente. En los enlaces troncales se introduce un nuevo concepto denominado VLAN nativa.

Las tramas de la VLAN nativa son las únicas que viajan sin etiqueta a través de un enlace troncal y su función es transportar información de configuración entre los *switches*, por ejemplo, las actualizaciones de GVRP. Por defecto, la VLAN nativa es la 1 y viene agregada en todos los puertos troncales, no obstante, siempre es recomendable retirarla y asignar una VLAN distinta por motivos de seguridad. Para configurar un puerto en modo troncal se ingresa a la vista de la interfaz y se emplea el comando: *port link-type trunk*. Posteriormente se agregan las VLAN al troncal con el comando: *port trunk allow-pass vlan vlan_id [vlan2_id] [vlan3_id]*. Para cambiar de VLAN nativa previamente se debe permitir la VLAN en el puerto y luego emplear el comando: *port trunk pvid vlan vlan_id*, el término PVID hace referencia número de VLAN que enviará tramas sin etiqueta por el

puerto, es decir la VLAN nativa. Finalmente, para retirar una o varias VLAN del puerto se emplea el comando: *undo port trunk allow-pass vlan vlan_id [vlan2_id]*.

En la figura 103, se muestra la configuración del puerto G0/0/2 en modo troncal permitiendo las VLAN 10, 20 y 30, siendo esta última la VLAN y retirando la 1 del troncal.

Figura 103. **Configuración de un puerto en modo troncal**

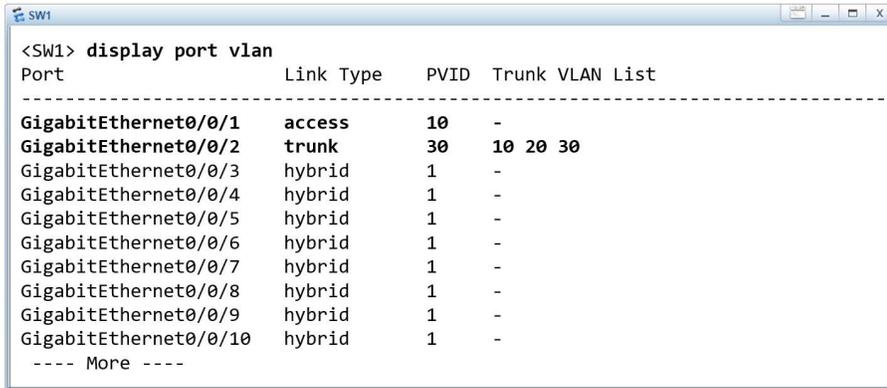


```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] vlan 10
[SW1-vlan10] description VLAN_10
[SW1-vlan10] vlan 20
[SW1-vlan20] description VLAN_20
[SW1-vlan20] vlan 30
[SW1-vlan30] description VLAN_30_NATIVA
[SW1-vlan30] quit
[SW1] interface G0/0/2
[SW1-GigabitEthernet0/0/2] port link-type trunk
[SW1-GigabitEthernet0/0/2] port trunk allow-pass vlan 10 20 30
[SW1-GigabitEthernet0/0/2] port trunk pvid vlan 30
[SW1-GigabitEthernet0/0/2] undo port trunk allow-pass vlan 1
[SW1-GigabitEthernet0/0/2] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

Para visualizar las VLAN asignadas a los puertos se emplea el comando: *display port vlan*, como se muestra en la figura 104.

Figura 104. **VLAN asignadas a los puertos de un switch**



```
<SW1> display port vlan
Port                Link Type  PVID  Trunk VLAN List
-----
GigabitEthernet0/0/1  access    10    -
GigabitEthernet0/0/2  trunk     30    10 20 30
GigabitEthernet0/0/3  hybrid    1     -
GigabitEthernet0/0/4  hybrid    1     -
GigabitEthernet0/0/5  hybrid    1     -
GigabitEthernet0/0/6  hybrid    1     -
GigabitEthernet0/0/7  hybrid    1     -
GigabitEthernet0/0/8  hybrid    1     -
GigabitEthernet0/0/9  hybrid    1     -
GigabitEthernet0/0/10 hybrid    1     -
---- More ----
```

Fuente: elaboración propia, empleando eNSP.

2.8.3. GVRP

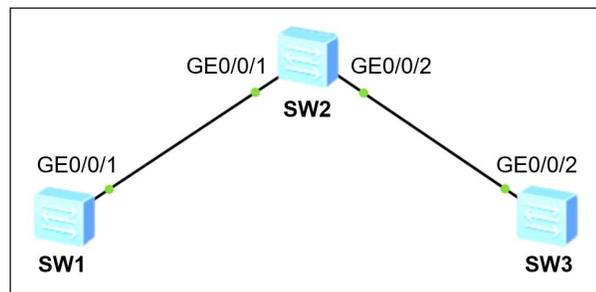
El protocolo GVRP, iniciales para GARP VLAN Registration Protocol, viene incluido en el estándar IEEE GARP y su función es automatizar la creación de VLAN en una red de *switches*. En una red que implementa GVRP solo es necesario crear manualmente las VLAN en uno de los *switches*, cualquiera de ellos, y el protocolo se encargará de transmitir esa actualización al resto de *switches* con el objetivo de que también se creen las mismas VLAN automáticamente.

Las VLAN creadas manualmente en un *switch* se denominan VLAN estáticas mientras que las VLAN creadas automáticamente gracias a GVRP se denominan VLAN dinámicas. Para implementar GVRP en una red se inicia activando el protocolo en la vista del sistema con el comando: *gvrp*. Seguidamente se crean las VLAN y se configura en modo troncal el puerto que conecta con el *switch* vecino aceptando las VLAN recién creadas. En los *switches* vecinos también es necesario declarar el puerto como troncal aceptando todas

las VLAN, para eso se puede emplear el comando: *port trunk allow-pass vlan all*. Finalmente se activa el protocolo en el puerto troncal con el comando: *gvrp*.

En la figura 105, se muestra un ejemplo de la implementación del protocolo GVRP en tres *switches* S5700.

Figura 105. **Topología de ejemplo para implementar GVRP**



Fuente: elaboración propia, empleando eNSP.

En la figura 106, se muestra la configuración del *switch* SW1 donde se crea la VLAN 7 y se declara en modo troncal el puerto GE0/0/1 permitiendo todas las VLAN.

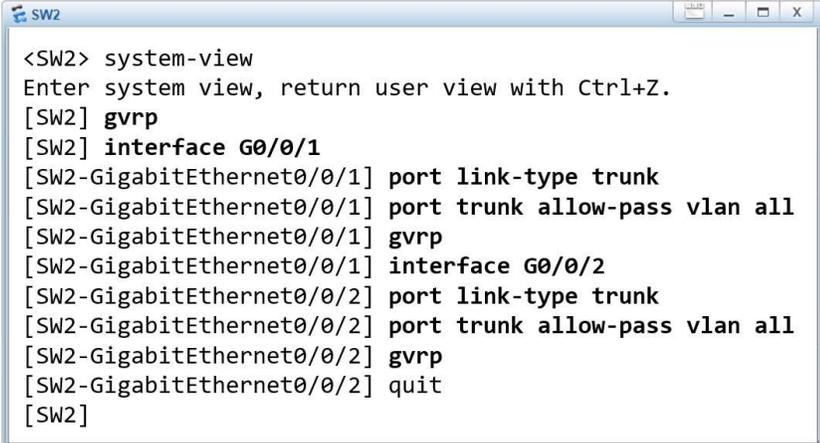
Figura 106. **Configuración de SW1**

```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] gvrp
[SW1] vlan 7
[SW1-vlan7] description VLAN_7
[SW1-vlan7] quit
[SW1] interface G0/0/1
[SW1-GigabitEthernet0/0/1] port link-type trunk
[SW1-GigabitEthernet0/0/1] port trunk allow-pass vlan all
[SW1-GigabitEthernet0/0/1] gvrp
[SW1-GigabitEthernet0/0/1] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 107, se muestra la configuración del *switch* SW2 donde se crean dos puertos troncales, GE0/0/1 y GE0/0/2, permitiendo todas las VLAN.

Figura 107. **Configuración de SW2**

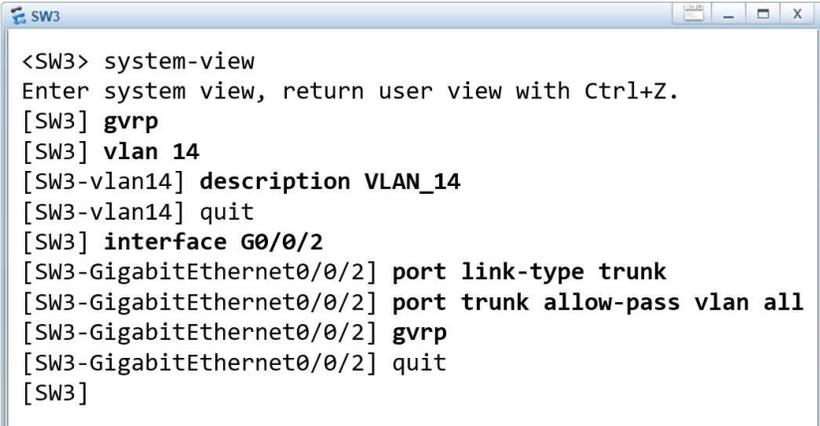


```
<SW2> system-view
Enter system view, return user view with Ctrl+Z.
[SW2] gvrp
[SW2] interface G0/0/1
[SW2-GigabitEthernet0/0/1] port link-type trunk
[SW2-GigabitEthernet0/0/1] port trunk allow-pass vlan all
[SW2-GigabitEthernet0/0/1] gvrp
[SW2-GigabitEthernet0/0/1] interface G0/0/2
[SW2-GigabitEthernet0/0/2] port link-type trunk
[SW2-GigabitEthernet0/0/2] port trunk allow-pass vlan all
[SW2-GigabitEthernet0/0/2] gvrp
[SW2-GigabitEthernet0/0/2] quit
[SW2]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 108, se muestra la configuración el *switch* SW3 donde se crea la VLAN 14 y se declara el puerto GE0/0/2 en modo troncal permitiendo todas las VLAN.

Figura 108. **Configuración de SW3**



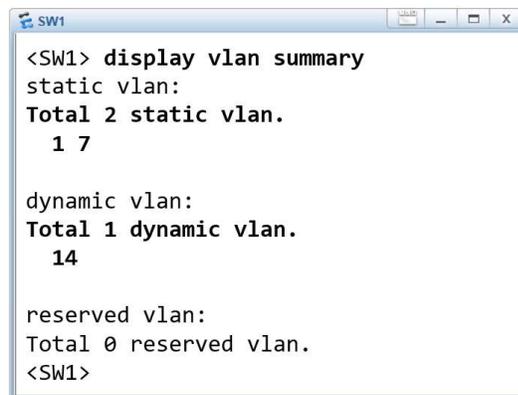
```
<SW3> system-view
Enter system view, return user view with Ctrl+Z.
[SW3] gvrp
[SW3] vlan 14
[SW3-vlan14] description VLAN_14
[SW3-vlan14] quit
[SW3] interface G0/0/2
[SW3-GigabitEthernet0/0/2] port link-type trunk
[SW3-GigabitEthernet0/0/2] port trunk allow-pass vlan all
[SW3-GigabitEthernet0/0/2] gvrp
[SW3-GigabitEthernet0/0/2] quit
[SW3]
```

Fuente: elaboración propia, empleando eNSP.

Para verificar las VLAN estáticas o dinámicas creadas en un *switch* se emplea el comando: *display vlan summary*.

En la figura 109, se observan las VLAN estáticas y dinámicas creadas en el *switch* SW1.

Figura 109. **VLAN estáticas y dinámicas en SW1**



```
<SW1> display vlan summary
static vlan:
Total 2 static vlan.
  1 7

dynamic vlan:
Total 1 dynamic vlan.
  14

reserved vlan:
Total 0 reserved vlan.
<SW1>
```

Fuente: elaboración propia, empleando eNSP.

2.8.4. VLANIF

VLANIF, iniciales para VLAN Interface, es una función que permite asociar una interfaz virtual a una VLAN y asignarle una dirección IP. La VLANIF, también conocida como SVI o Switch Virtual Interface, resulta útil para asignar una dirección IP de gestión para los *switches* de la red, aunque también es utilizada cuando se implementa enrutamiento Inter-VLAN con *switches* multicapa.

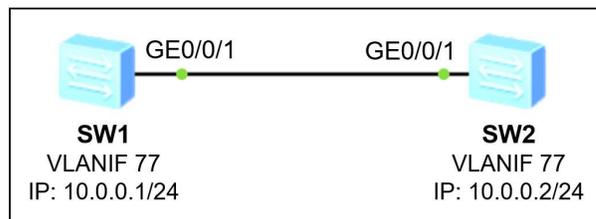
Para configurar una VLANIF en un *switch* se inicia creando una VLAN asignada para la gestión. Posteriormente se crea la interfaz virtual con el comando: ***interface vlanif*** *vlan_id*, donde el parámetro *vlan_id* es el VID de la

VLAN de gestión. Seguidamente se configura la dirección IP con el comando: *ip address ip_address mask*.

Finalmente se debe encender la interfaz de forma lógica con el comando: *undo shutdown*. Es importante mencionar que la interfaz virtual permanecerá caída o *down* si no se tiene alguna interfaz física activa en la que se permita la VLAN de gestión, ya sea en modo acceso o troncal.

En la figura 110, se brinda un ejemplo implementando VLANIF para comunicar dos *switches*.

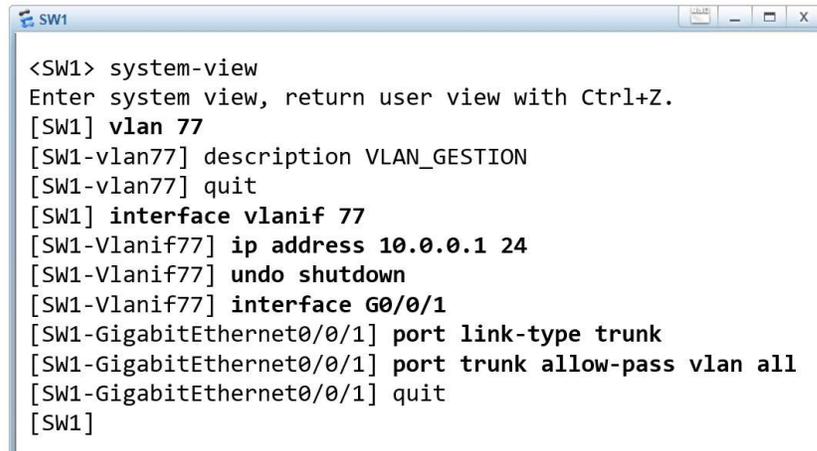
Figura 110. **VLANIF en dos switches**



Fuente: elaboración propia, empleando eNSP.

En la figura 111, se muestra la configuración del *switch* SW1 donde se crea la VLANIF 77 y se declara el puerto GE0/0/1 como troncal permitiendo todas las VLAN.

Figura 111. Configuración de SW1

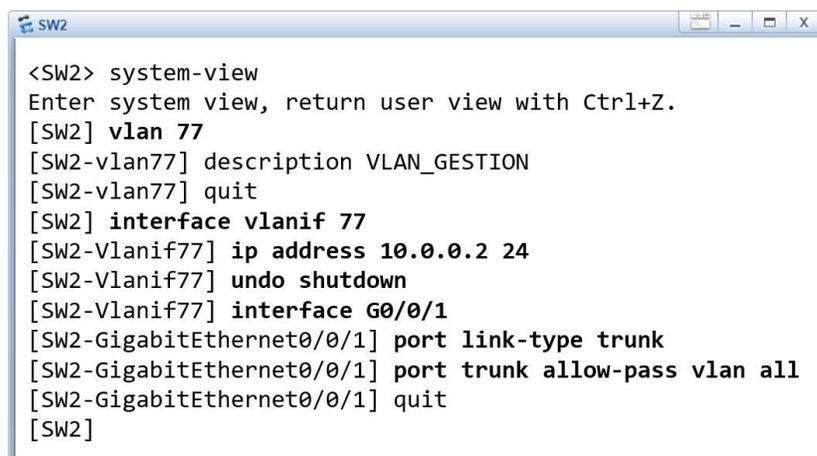


```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] vlan 77
[SW1-vlan77] description VLAN_GESTION
[SW1-vlan77] quit
[SW1] interface vlanif 77
[SW1-Vlanif77] ip address 10.0.0.1 24
[SW1-Vlanif77] undo shutdown
[SW1-Vlanif77] interface G0/0/1
[SW1-GigabitEthernet0/0/1] port link-type trunk
[SW1-GigabitEthernet0/0/1] port trunk allow-pass vlan all
[SW1-GigabitEthernet0/0/1] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 112, se muestra la configuración del *switch* SW2 donde se crea la VLANIF 77 y se declara la interfaz GE0/0/1 en modo troncal permitiendo todas las VLAN.

Figura 112. Configuración de SW2

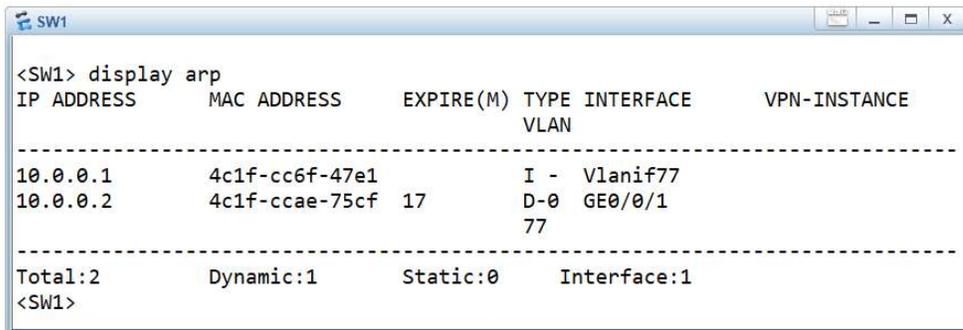


```
<SW2> system-view
Enter system view, return user view with Ctrl+Z.
[SW2] vlan 77
[SW2-vlan77] description VLAN_GESTION
[SW2-vlan77] quit
[SW2] interface vlanif 77
[SW2-Vlanif77] ip address 10.0.0.2 24
[SW2-Vlanif77] undo shutdown
[SW2-Vlanif77] interface G0/0/1
[SW2-GigabitEthernet0/0/1] port link-type trunk
[SW2-GigabitEthernet0/0/1] port trunk allow-pass vlan all
[SW2-GigabitEthernet0/0/1] quit
[SW2]
```

Fuente: elaboración propia, empleando eNSP.

Se puede validar que existe comunicación a nivel de capa dos al verificar que el ARP completa, como se muestra en la figura 113.

Figura 113. **Tabla ARP de SW1**



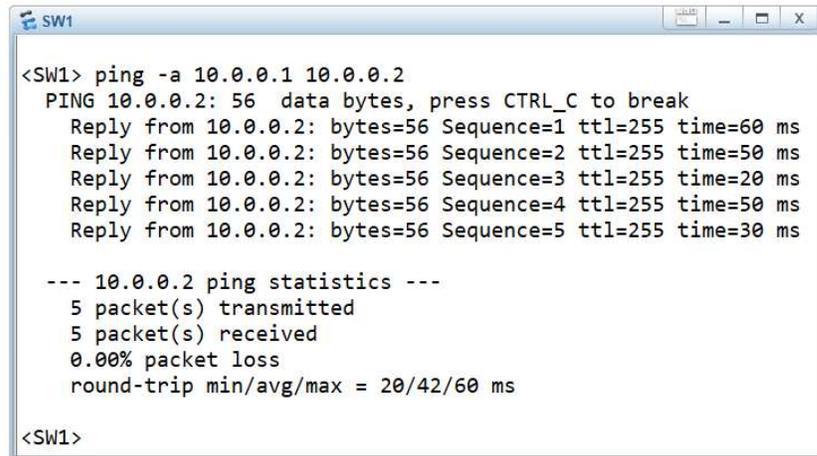
```
<SW1> display arp
IP ADDRESS      MAC ADDRESS      EXPIRE(M)  TYPE  INTERFACE      VPN-INSTANCE
-----
10.0.0.1        4c1f-cc6f-47e1   I - Vlanif77
10.0.0.2        4c1f-ccae-75cf   17         D-0  GE0/0/1
              77
-----
Total:2         Dynamic:1        Static:0    Interface:1
<SW1>
```

Fuente: elaboración propia, empleando eNSP.

También se puede verificar que exista comunicación a nivel de capa tres entre los dos *switches*, para esto se emplea el comando: *ping -a ip_address_source ip_address*, donde el parámetro *ip_address_source* es la dirección IP donde se origina el paquete ICMP y tiene que estar configurada en alguna interfaz activa del equipo, y el parámetro *ip_address* es la dirección IP de destino.

En la figura 114, se muestra un *ping* desde la VLANIF 77 de SW1 hacia la VLANIF 77 de SW2.

Figura 114. **Ping desde SW1 hacia SW2**



```
<SW1> ping -a 10.0.0.1 10.0.0.2
PING 10.0.0.2: 56 data bytes, press CTRL_C to break
  Reply from 10.0.0.2: bytes=56 Sequence=1 ttl=255 time=60 ms
  Reply from 10.0.0.2: bytes=56 Sequence=2 ttl=255 time=50 ms
  Reply from 10.0.0.2: bytes=56 Sequence=3 ttl=255 time=20 ms
  Reply from 10.0.0.2: bytes=56 Sequence=4 ttl=255 time=50 ms
  Reply from 10.0.0.2: bytes=56 Sequence=5 ttl=255 time=30 ms

--- 10.0.0.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 20/42/60 ms

<SW1>
```

Fuente: elaboración propia, empleando eNSP.

2.8.5. Enrutamiento Inter-VLAN

Al implementar VLAN en un *switch* se segmenta el dominio de Broadcast lo cual resulta análogo a tener un *switch* independiente por cada VLAN. Para tener comunicación entre distintas VLAN es necesario algún equipo con capacidad de enrutamiento, como un *router* o un *switch* multicapa, ya que cada VLAN es una red distinta.

Existen tres formas para implementar enrutamiento Inter-VLAN, la primera es una técnica llamada: Multi-Armed Router, y consiste en asignar una interfaz de un *router* por cada VLAN, es decir, si se tienen 10 VLAN se necesitarán 10 interfaces en un *router* y 10 puertos en un *switch*. Esta técnica es rara vez utilizada debido al desperdicio de recursos que supondría.

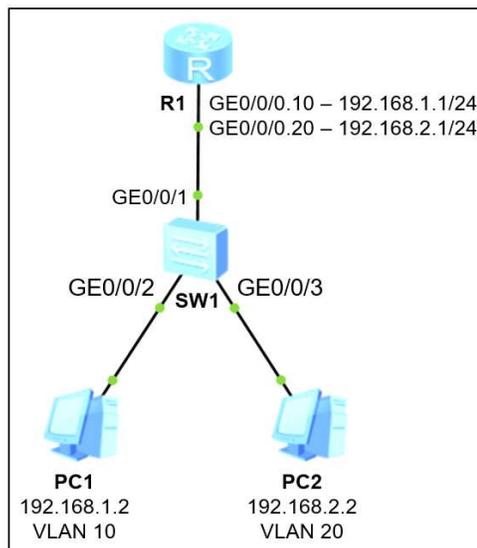
La segunda técnica es llamada: One-Armed Router, y consiste en crear subinterfaces en la interfaz física de un *router* y habilitarla para manejar

etiquetado 802.1q. Finalmente la tercera técnica consiste en implementar un *switch* con la capacidad de enrutamiento, *switches* multicapa como el Huawei S5700 o S3700, y crear VLANIF por cada VLAN para luego crear puertos troncales y comunicarse con el resto de la red L2 mediante el etiquetado 802.1q.

2.8.5.1. Enrutamiento con subinterfaces

Para implementar subinterfaces en un *router* se toma como ejemplo la topología mostrada en la figura 115.

Figura 115. **Topología propuesta para implementar enrutamiento Inter-VLAN con subinterfaces**



Fuente: elaboración propia, empleando eNSP.

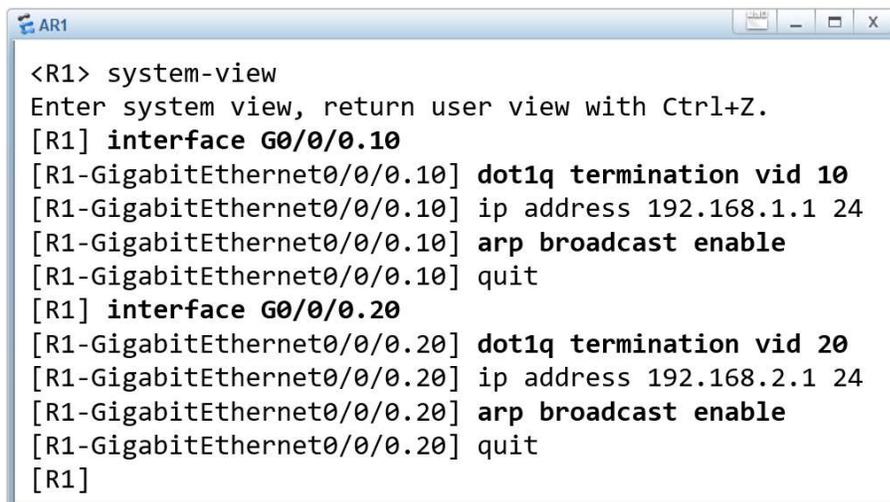
Se inicia creando las subinterfaces en el *router*, para esto se ingresa a la vista del sistema y se emplea el comando: `interface type_interface interface_number.sub_interface_number`, donde el parámetro `sub_interface_number` es un número entero que identifica a la subinterfaz, este

número no tiene que coincidir obligatoriamente con el ID de la VLAN asignada, no obstante, siempre es aconsejable para mantener el orden.

Posteriormente se configura la dirección IP a la subinterfaz con el comando: *ip address ip_address mask*, y se habilita la encapsulación 802.1q para recibir tramas etiquetadas con el comando: *dot1q termination vid vlan_id*, donde el parámetro *vlan_id* es el identificador de la VLAN. Finalmente se agrega el comando: *arp broadcast enable*, con el objetivo de habilitar la recepción de paquetes Broadcast para completar el ARP en cada subinterfaz.

En la figura 116, se muestra la configuración de 2 subinterfaces en R1.

Figura 116. **Configurando subinterfaces en R1**



```
<R1> system-view
Enter system view, return user view with Ctrl+Z.
[R1] interface G0/0/0.10
[R1-GigabitEthernet0/0/0.10] dot1q termination vid 10
[R1-GigabitEthernet0/0/0.10] ip address 192.168.1.1 24
[R1-GigabitEthernet0/0/0.10] arp broadcast enable
[R1-GigabitEthernet0/0/0.10] quit
[R1] interface G0/0/0.20
[R1-GigabitEthernet0/0/0.20] dot1q termination vid 20
[R1-GigabitEthernet0/0/0.20] ip address 192.168.2.1 24
[R1-GigabitEthernet0/0/0.20] arp broadcast enable
[R1-GigabitEthernet0/0/0.20] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

Es importante mencionar que el puerto G0/0/1 de SW1 debe ir en modo troncal aceptando las VLAN que coincidan con la cantidad de subinterfaces configuradas en R1. En la figura 117, se muestra la configuración del troncal en SW1.

Figura 117. Configurando puertos en SW1

```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] vlan batch 10 20
Info: This operation may take a few seconds. Please wait for a moment...done.
[SW1] interface G0/0/1
[SW1-GigabitEthernet0/0/1] port link-type trunk
[SW1-GigabitEthernet0/0/1] port trunk allow-pass vlan 10 20
[SW1-GigabitEthernet0/0/1] interface G0/0/2
[SW1-GigabitEthernet0/0/2] port link-type access
[SW1-GigabitEthernet0/0/2] port default vlan 10
[SW1-GigabitEthernet0/0/2] interface G0/0/3
[SW1-GigabitEthernet0/0/3] port link-type access
[SW1-GigabitEthernet0/0/3] port default vlan 20
[SW1-GigabitEthernet0/0/3] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

Al momento de verificar la configuración, una función útil es visualizar el estado de las subinterfaces del *router*, para esto se emplea el comando: *display ip interface brief*, como se muestra en la figura 118.

Figura 118. Estado de las interfaces de R1

```
<R1> display ip interface brief
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
The number of interface that is UP in Physical is 4
The number of interface that is DOWN in Physical is 5
The number of interface that is UP in Protocol is 3
The number of interface that is DOWN in Protocol is 6

Interface                IP Address/Mask    Physical  Protocol
GigabitEthernet0/0/0     unassigned         up        down
GigabitEthernet0/0/0.10  192.168.1.1/24    up        up
GigabitEthernet0/0/0.20  192.168.2.1/24    up        up
GigabitEthernet0/0/1     unassigned         down      down
<R1>
```

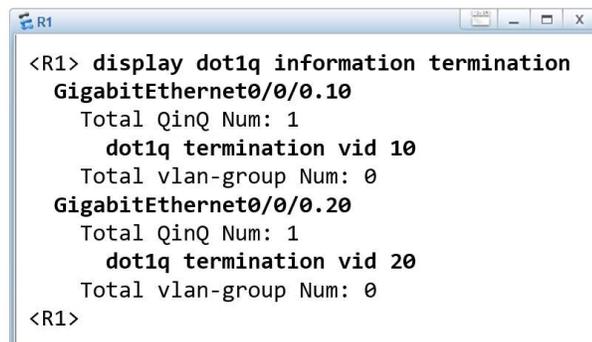
Interfaz física debe estar up/down

Fuente: elaboración propia, empleando eNSP.

Para visualizar el VID asignado a cada subinterfaz se emplea el comando: *display dot1q information termination*. Adicional se puede revisar la tabla ARP para verificar el puerto y la VLAN por donde se está aprendiendo una dirección MAC.

En la figura 119, se puede visualizar el etiquetado 802.1q de las subinterfaces de R1.

Figura 119. **Etiquetado 802.1q en las subinterfaces de R1**



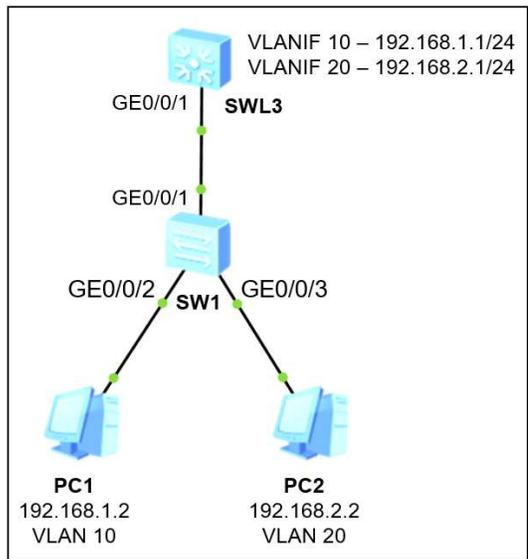
```
<R1> display dot1q information termination
GigabitEthernet0/0/0.10
  Total QinQ Num: 1
    dot1q termination vid 10
  Total vlan-group Num: 0
GigabitEthernet0/0/0.20
  Total QinQ Num: 1
    dot1q termination vid 20
  Total vlan-group Num: 0
<R1>
```

Fuente: elaboración propia, empleando eNSP.

2.8.5.2. Enrutamiento con VLANIF

Para implementar enrutamiento con interfaces virtuales en un *switch* capa 3 se toma como ejemplo la topología mostrada en la figura 120.

Figura 120. **Topología propuesta para implementar enrutamiento Inter-VLAN con VLANIF**



Fuente: elaboración propia, empleando eNSP.

En el *switch* SWL3 se crean dos VLANIF, una para la VLAN 10 y otra para la VLAN 20, y se configura el puerto *downstream* G0/0/1 en modo troncal aceptando las 2 VLAN. La configuración de SW1 es idéntica al ejemplo de la sección anterior.

En la figura 121, se muestra la creación de las dos VLANIF y se declara el puerto *downstream* GE0/0/1 como troncal.

Figura 121. Configurando VLANIF en SWL3

```
<SWL3> system-view
Enter system view, return user view with Ctrl+Z.
[SWL3] vlan batch 10 20
Info: This operation may take a few seconds. Please wait for a moment...done.
[SWL3] interface vlanif 10
[SWL3-Vlanif10] ip address 192.168.1.1 24
[SWL3-Vlanif10] interface vlanif 20
[SWL3-Vlanif20] ip address 192.168.2.1 24
[SWL3-Vlanif20] interface G0/0/1
[SWL3-GigabitEthernet0/0/1] port link-type trunk
[SWL3-GigabitEthernet0/0/1] port trunk allow-pass vlan 10 20
[SWL3-GigabitEthernet0/0/1] quit
[SWL3]
```

Fuente: elaboración propia, empleando eNSP.

Para verificar el estado de las interfaces se emplea el comando: *display ip interface brief*. También se puede verificar si existe comunicación a nivel de capa dos con el comando: *display arp* o *display mac-address*.

En la figura 122, se muestra el estado de las VLANIFs configuradas en SWL3.

Figura 122. Estado de las interfaces en SWL3

```
<SWL3> display ip interface brief
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
The number of interface that is UP in Physical is 4
The number of interface that is DOWN in Physical is 1
The number of interface that is UP in Protocol is 3
The number of interface that is DOWN in Protocol is 2

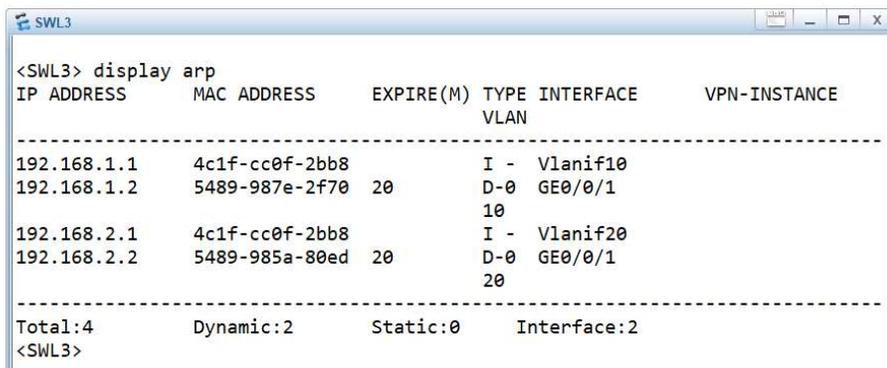
Interface                IP Address/Mask    Physical  Protocol
MEth0/0/1                unassigned         down     down
NULL0                    unassigned         up       up(s)
Vlanif1                  unassigned         up       down
Vlanif10                  192.168.1.1/24    up       up
Vlanif20                  192.168.2.1/24    up       up
<SWL3>
```

Interfaces deben estar up/up

Fuente: elaboración propia, empleando eNSP.

En la figura 123, se muestra la tabla ARP de SWL3.

Figura 123. **Tabla ARP de SWL3**



IP ADDRESS	MAC ADDRESS	EXPIRE(M)	TYPE	INTERFACE	VPN-INSTANCE
192.168.1.1	4c1f-cc0f-2bb8		I -	Vlanif10	
192.168.1.2	5489-987e-2f70	20	D-0	GE0/0/1	
192.168.2.1	4c1f-cc0f-2bb8		I -	Vlanif20	
192.168.2.2	5489-985a-80ed	20	D-0	GE0/0/1	

Total:4 Dynamic:2 Static:0 Interface:2

Fuente: elaboración propia, empleando eNSP.

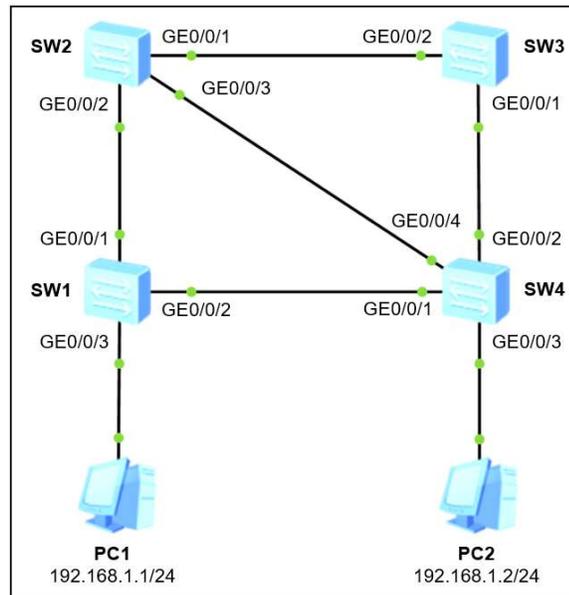
2.9. Spanning-Tree Protocol

STP es un protocolo definido en el estándar IEEE 802.1d cuya principal función es gestionar el estado de los puertos de los *switches* para crear una ruta principal libre de bucles en la capa de estado de enlace de la red. Para esto, STP elige un Root Bridge, es decir un *switch* raíz, en donde se centraliza todo el tráfico de datos para evitar los *loops* de capa 2. STP es la implementación más básica de esta tecnología ya que solo maneja una instancia, posteriormente se tuvo mejoras como RSTP definido en el estándar IEEE 802.1w y MSTP definido en el estándar IEEE 802.1s.

2.9.1. Principio de operación

Para analizar el funcionamiento de STP se toma como ejemplo la red Broadcast mostrada en la figura 124.

Figura 124. **Análisis de STP en una red**

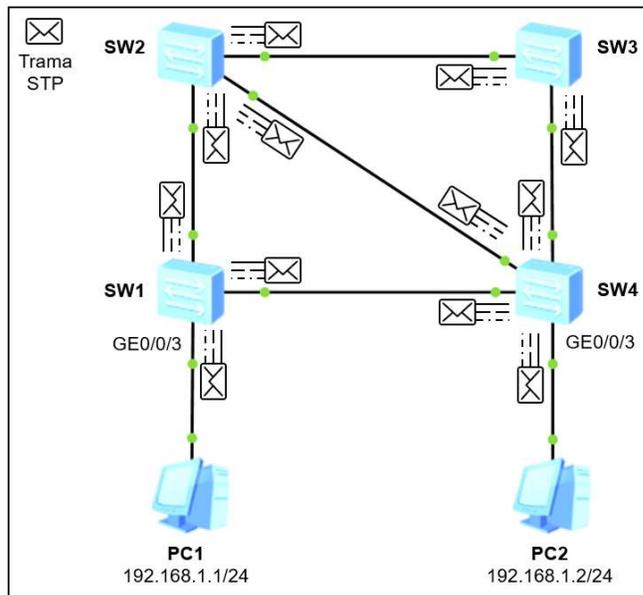


Fuente: elaboración propia, empleando eNSP.

Todos los *switches* Huawei vienen con STP habilitado por defecto, así que entra en funcionamiento al encender el equipo. El primer paso que realiza el protocolo se una inundación de tramas STP por los puertos activos del *switch*, estas tramas tienen el propósito de intercambiar información como el Bridge ID, Port ID, costo de la ruta, entre otros parámetros del *switch* emisor para encontrar bucles en la red y elegir el *switch* raíz.

En la figura 125 se muestra la inundación de tramas STP en los puertos activos de los *switches*.

Figura 125. Inundación de tramas STP



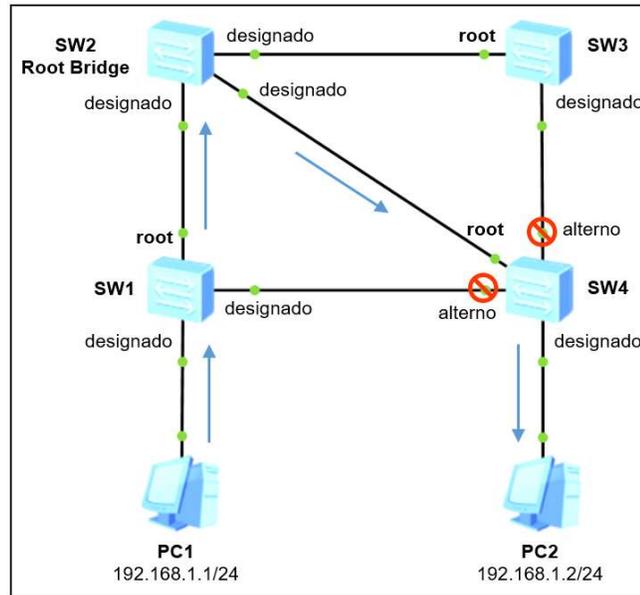
Fuente: elaboración propia, empleando eNSP.

El parámetro por evaluar para elegir el *switch* raíz de la red es el *Bridge ID*. Para este ejemplo se supondrá que el BID o Bridge ID de menor valor lo posee SW2, por tanto, será electo como raíz o *root*.

Una vez electo el Root Bridge de la red, todos los demás *switches* analizan las rutas para llegar a este y proceden a configurar sus puertos en tres posibles roles: raíz, designado o alternativo. Algunos de los puertos serán bloqueados, puerto alternativo, para evitar bucles en la red mientras el resto serán alternos que enviarán y recibirán tramas de datos. Los puertos designados y raíz son capaces de enviar y recibir tramas por lo que la información puede fluir por ellos sin problema.

En la figura 126 se muestra un ejemplo de cómo serían los roles de los puertos en una topología implementando STP.

Figura 126. Roles de los puertos



Fuente: elaboración propia, empleando eNSP.

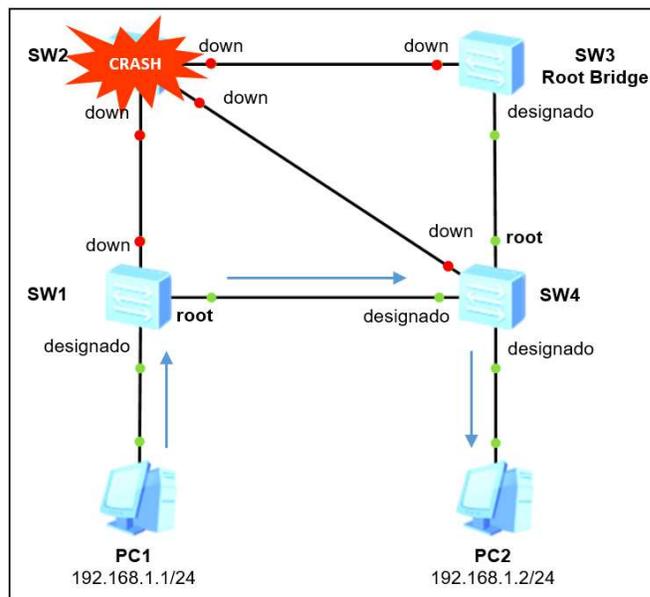
Con estas disposiciones cualquier trama que viaje por la red debe atravesar forzosamente el *switch* raíz, formando así una topología STP con una sola ruta de comunicación que evita tormentas de Broadcast, tramas duplicadas e inestabilidad en la tabla de direcciones MAC de los *switches*. A este estado se le conoce como convergencia STP.

Ahora, emulando un entorno real, se supone que el *switch* SW2 falla y, debido a que este es el *switch* raíz de la red, la comunicación entre los *hosts* cae momentáneamente. Al detectar la caída de SW2, STP elige un nuevo *switch* raíz, siempre basándose en el *Bridge ID*, y desbloquea alguno de los puertos de los *switches* para mantener la comunicación. Para este ejemplo se supone que SW3 toma el papel del *switch* raíz y el puerto GE0/0/1 de SW4 cambia a designado para crear una ruta hacia SW1.

En general, los eventos que causan cambios de los roles de los puertos provocan reconvergencia STP ya que el protocolo debe crear una nueva topología. La reconvergencia STP debe evitarse en la medida de lo posible debido a que causa inestabilidad en la red.

En la figura 127, se observa una nueva ruta creada por STP para mantener la comunicación entre los *hosts* a pesar de que SW2 se encuentra fuera.

Figura 127. Ruta alterna encontrada por STP



Fuente: elaboración propia, empleando eNSP.

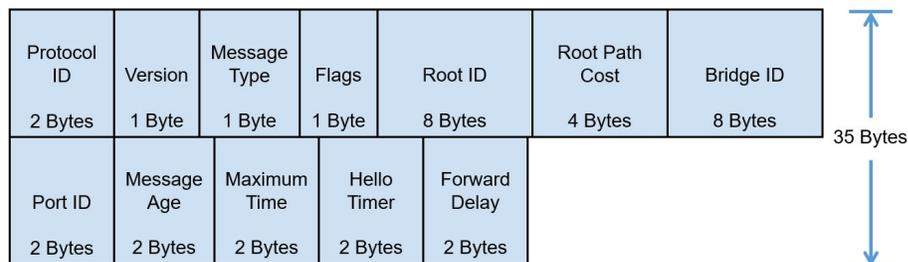
Si el *switch* SW2 vuelve a estar activo, entonces STP reconoce que este cuenta con el BID más bajo de toda la red y lo posiciona nuevamente como *switch* raíz y los puertos de SW4 vuelven a su estado original, es decir en alterno GE0/0/1 y GE0/0/2.

2.9.2. BPDU

El funcionamiento de STP se basa en el intercambio de tramas STP cuyo paquete de información son los BPDU. Estas tramas se envían por los puertos activos de los *switches* hacia la dirección Multicast 0180-C200-0000 cada 2 segundos, no obstante, este valor puede modificarse a conveniencia. Un BPDU o Bridge Protocol Data Unit es un mensaje especial que acarrea información del *switch* emisor y tiene la función de encontrar bucles en la topología e intercambiar información para la elección del Root Bridge. Entre la información que se incluye en un BPDU se puede mencionar el *Bridge ID*, *Root ID*, el costo hacia el Root Bridge, el *Port ID*, entre otros.

En la figura 128 se muestra la estructura de un BPDU, la forma de interpretarla es de arriba hacia abajo y de izquierda a derecha.

Figura 128. Estructura de un BPDU

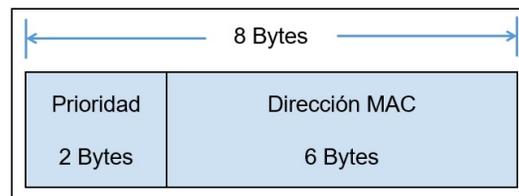


Fuente: elaboración propia, empleando eNSP.

Es importante mencionar que el término *bridge* se emplea para referirse a un *switch*, esto es debido a que en la antigüedad los *bridges* se empleaban como concentradores de red y el término quedó acuñado desde entonces. A continuación, se brinda una breve descripción de alguno de los parámetros que conforman un BPDU.

- *Protocol ID*: El identificador del protocolo siempre es 0x0000 para STP.
- *Version*: La versión es 0x00 para STP, 0x02 para RSTP y 0x03 par MSTP.
- *Bridge ID*: Es un número único que identifica al *switch* en una red Broadcast, está conformado por la prioridad y la dirección MAC del *bridge*. La prioridad es un número entero de 16 bits, entre 0 y 65 535, y es el primer parámetro por evaluar a la hora de elegir al Root Bridge de la red L2. Por defecto, todos los *switches* Huawei vienen configurados con una prioridad de 32 768. Para la dirección MAC del *bridge* por lo general se elige la dirección MAC del número de puerto de menor valor. En la figura 129 se muestra la estructura del Bridge ID.

Figura 129. **Estructura de *Bridge ID***

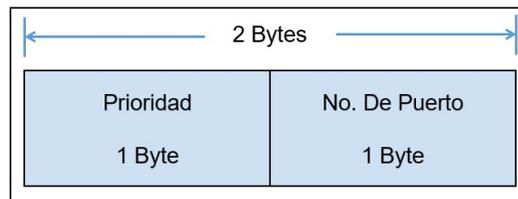


Fuente: elaboración propia, empleando eNSP.

- *Root ID*: es el Bridge ID del *switch* raíz o Root Bridge de la red Broadcast.
- *Port ID*: Es un número entero que identifica a los puertos de un *switch*. Tiene una longitud de 2 bytes y está conformado por la prioridad del puerto, 1 byte, y el número del puerto, 1 byte. Su uso es crucial al momento de decidir el rol del puerto en la topología STP. Por defecto, los puertos de un *switch* Huawei vienen configurados con una prioridad de 128, pero pueden ser modificando a conveniencia, esto se logra ingresando a la vista de la

interfaz y empleando el comando: *stp port priority port_priority*, donde el parámetro *port_priority* puede variar entre 0 y 240 con saltos de 16. En la figura 130 se muestra la estructura del Port ID.

Figura 130. **Estructura de Port ID**



Fuente: elaboración propia, empleando eNSP.

- **Root Path Cost:** El costo es un parámetro asignado a los puertos en base a su velocidad de operación y está regido por el estándar IEEE 802.1t, como se muestra en la tabla X. El costo de la ruta hacia el Root Bridge es la suma algebraica de todos los costos de los puertos de salida hasta el Root Bridge. En base al costo de la ruta se puede determinar la ruta más corta hacia el Root Bridge.

Tabla X. **Costos en base al estándar IEEE 802.1t**

Velocidad del puerto	Costo
10 Mbps	2 000 000
100 Mbps	200 000
1 Gbps	20 000
10 Gbps	2 000

Fuente: elaboración propia.

- Hello Time: El período de tiempo al cual se envían los BPDU por los puertos activos del *switch*. Por defecto es de 2 segundos.
- Maximum Time: Es el período de tiempo máximo que un puerto puede estar en estado Blocking. Por defecto es de 20 segundos.
- Forward Delay: Es el período de tiempo que el puerto permanece en estado Listening y Learning. Por defecto es de 15 segundos.

2.9.3. Elección de Root Bridge

Como se mencionó anteriormente, la elección del Root Bridge se realiza al comparar el Bridge ID de los *switches*, el *switch* que tenga el Bridge ID de menor valor será electo como Root Bridge. Inicialmente cada *switch* se considera a él mismo como Root Bridge por lo que llena el campo Root ID del BPDU con su propio Bridge ID e inicia la inundación. Cuando un *switch* recibe un BPDU compara su Bridge ID con el Root ID que viene en el BPDU recibido, si su Bridge ID es mayor que el Root ID entonces continúa considerándose a él mismo como Root Bridge, en cambio si su Bridge ID es menor entonces considera Root al *switch* con Bridge ID igual al Root ID recibido en el BPDU. Ahora que el *switch* conoce al Root Bridge, al momento de crear su BPDU llenará el campo Root ID con el Bridge ID del Root Bridge y serán enviados por los demás puertos activos hacia el resto de los equipos de la red quienes repetirán el mismo proceso. Progresivamente todos los *switches* conocerán el Bridge ID del Root Bridge.

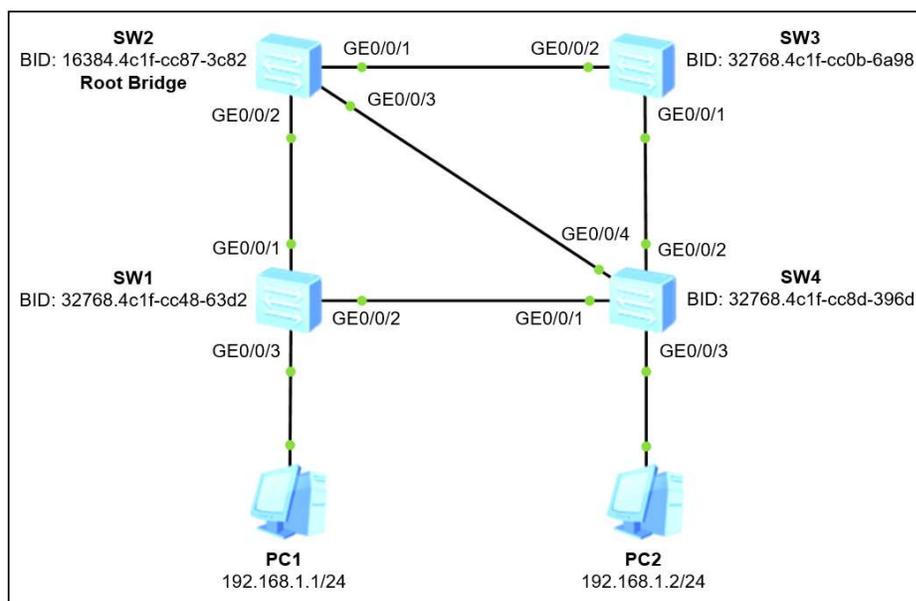
A la hora de construir una red es importante ser conscientes que todos los *switches* Huawei vienen con la misma prioridad del Bridge ID por defecto, así que la elección del Root Bridge se basa en la dirección MAC del Bridge. La dirección MAC es ascendente a medida que los equipos salen de producción de la fábrica, es decir que el equipo más antiguo tendrá la dirección MAC de menor valor y, por

lo tanto, será considerado el Root Bridge. Esto muchas veces resulta no ser beneficioso porque los equipos antiguos pueden tener menor capacidad de procesamiento que los recientes, es por esta razón que la modificación de la prioridad es importante.

Para modificar la prioridad del Bridge ID se ingresa a la vista del sistema y se emplea el comando: `stp priority bridge_priority`, donde el parámetro `bridge_priority` es la prioridad que puede estar entre 0 y 61 440 con saltos de 4 096, esto es debido a que los *switches* Huawei manejan múltiples instancias STP, lo cual reduce el tamaño del campo del Bridge ID.

Para comprender la elección del Root Bridge y del rol de los puertos se toma como ejemplo la topología mostrada en la figura 131, donde la prioridad del *switch* SW2 fue modificada a 16 384 para ser electo como el Root Bridge.

Figura 131. Ejemplo de elección de Root Bridge



Fuente: elaboración propia, empleando eNSP.

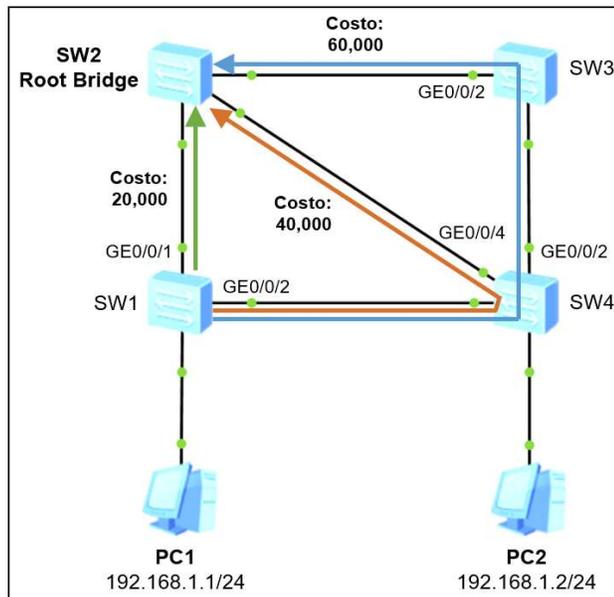
2.9.4. Elección del rol de los puertos

Ahora que todos los *switches* de la red conocen el Bridge ID del Root o Root ID, es momento de encontrar la ruta más corta hacia este. Para esto los *switches* utilizan el costo de los puertos de salida para encontrar el costo total de la ruta hacia el Root Bridge. Retomando la topología mostrada en la figura 131, todos los puertos están trabajando a una velocidad de operación de 1 Gbps, y por ende tienen un costo de 20 000.

Ahora cada *switch* debe encontrar todas las rutas posibles hacia el Root Bridge y elegir la más corta. Iniciando en análisis con *switch* SW1, este cuenta con tres rutas o caminos hacia el Root Bridge, la primera es saliendo por su interfaz GE0/0/1 directo hacia el Root Bridge, costo de 20 000, la segunda es saliendo hacia SW4 y luego hacia el Root Bridge, costo total de 40 000, y la tercera es saliendo hacia SW4, luego hacia SW3 y finalmente hacia el Root Bridge, costo total de 60 000. Evidentemente la ruta más corta es la que tiene un costo de 20 000, es por esta razón que SW1 cambia el rol del puerto GE0/0/1 a raíz o *root*, indicando que por él se puede llegar al Root Bridge con la ruta más corta.

En la figura 132 se muestra el costo de las rutas que puede tomar SW2 para llegar al Root Bridge, es decir SW1.

Figura 132. Costo de las rutas hacia el Root Bridge

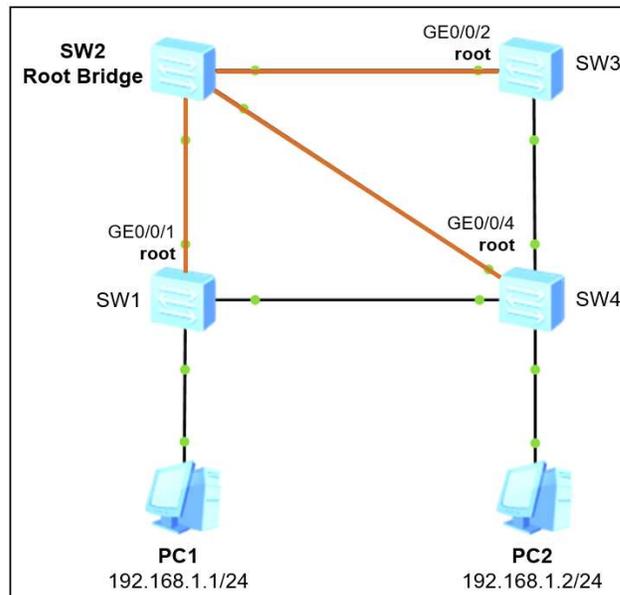


Fuente: elaboración propia, empleando eNSP.

En el caso de que se tengan dos rutas con el mismo costo hacia el Root Bridge, el siguiente criterio de desempate es el Bridge ID. Se elegirá puerto *root* el puerto del *switch* con menor Bridge ID. En el caso de que los dos Bridge ID sean iguales, es decir que las dos rutas atravesen el mismo *switch* en su camino hacia el Root Bridge, el siguiente criterio de desempate es la prioridad del puerto, eligiendo como puerto *root* al que posea la prioridad más baja, sin embargo, ya que todos los puertos poseen una prioridad por defecto de 128, el último criterio de desempate es el número del puerto. Se elige puerto *root* al que posea el número de puerto de menor valor.

Este mismo procedimiento lo repiten todos *switches* hasta que cada uno de ellos tenga un único puerto *root*. Los puertos *root* pueden mandar y recibir tramas sin problemas ya que se dirige hacia el Root Bridge. En la figura 133 se muestran los puertos *root* electos en la topología.

Figura 133. Puertos *root* en la topología



Fuente: Elaboración propia, empleando eNSP.

El siguiente paso es elegir cuales puertos estarán bloqueados, puertos alternos, y cuales podrán enviar y recibir tramas, puertos designados, sin el peligro de que se creen bucles. Por definición se tienen dos reglas: solo debe existir un puerto designado por enlace, segmento que une a dos *switches*, y todos los puertos del Root Bridge deben ser designados.

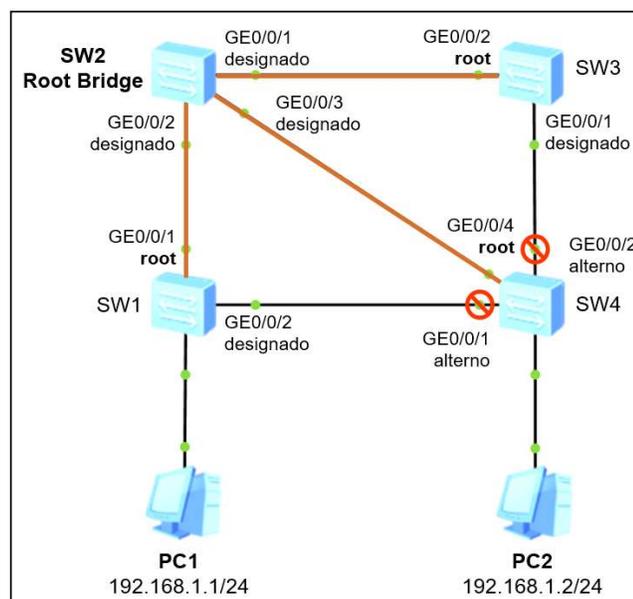
Para la elección de los puertos designados, STP también se basa en el costo, de esta forma el puerto que posea el menor costo hacia el Root Bridge será electo como designado. Si ambos costos hacia el Root Bridge son iguales el siguiente criterio de desempate es el Bridge ID, donde el *switch* que posea el Bridge ID de menor valor tendrá el puerto designado, y por consiguiente el de mayor Bridge ID tendrá el puerto alternativo. Si se da el caso de que ambos Bridge ID sean iguales, ambos puertos pertenecen al mismo *switch*, entonces el siguiente criterio de desempate será el Port ID, donde el puerto que tenga el

menor Port ID será designado. Finalmente, el último criterio de desempate es el número de puerto, donde el puerto que tenga el menor número de puerto será electo como designado.

Para el ejemplo de la figura 131, el puerto GE0/0/2 de SW1 y el puerto GE0/0/1 de SW4 poseen el mismo costo hacia el Root Bridge de 40 000, es por esta razón que se debe recurrir al segundo parámetro de desempate: el Bridge ID. En este caso el BID de SW1 es menor que el BID de SW4, por lo tanto, el puerto GE0/0/2 de SW1 será designado y el puerto GE0/0/1 será alternativo. El mismo escenario ocurre entre SW3 y SW4, ya que el BID de SW3 es menor entonces el puerto GE0/0/1 de SW3 será designado y el puerto GE0/0/2 de SW4 será alternativo.

En la figura 134 se muestra el rol de los puertos de todos los *switches*.

Figura 134. **Rol de los puertos en la topología**



Fuente: elaboración propia, empleando eNSP.

En resumen, cuando la topología logra la convergencia STP se pueden tener tres roles de los puertos: *root*, designado y alterno. Es importante mencionar que en muchas fuentes al puerto alterno también se le denomina no designado, esto es debido a que el nombre alterno fue introducido hasta la llegada de RSTP.

2.9.5. Estados de los puertos

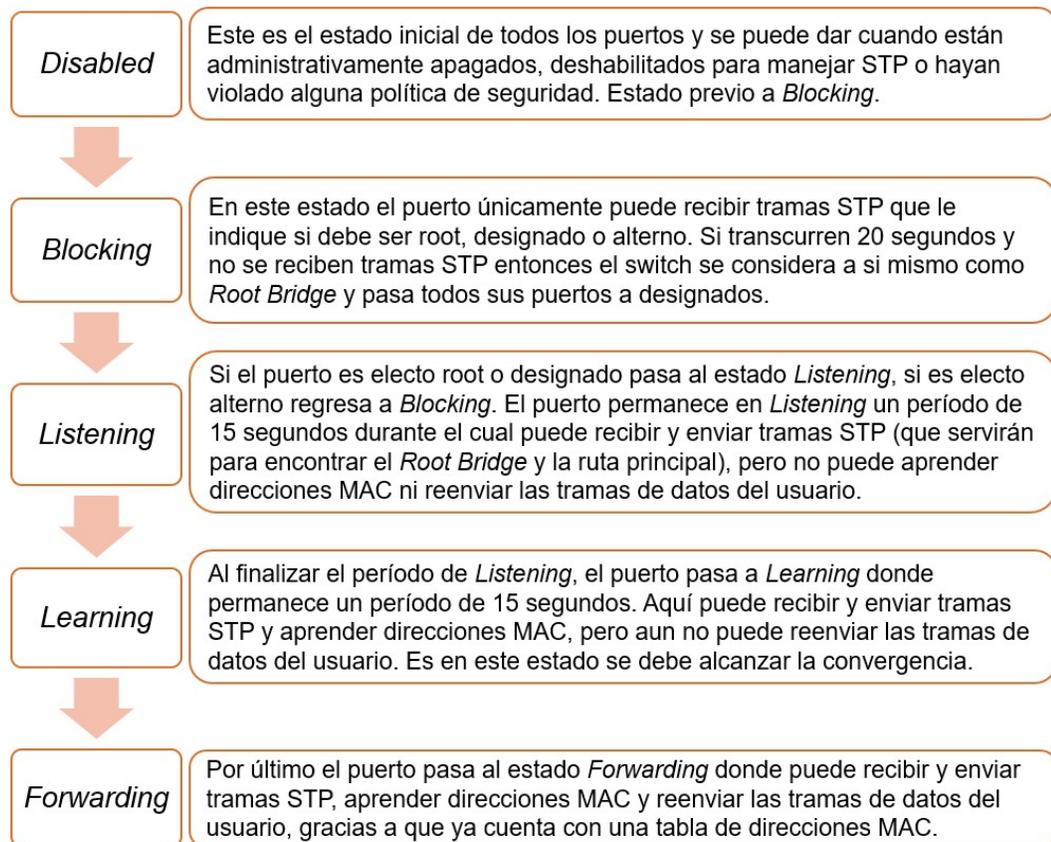
Para alcanzar el estado de convergencia STP se deben seguir cuatro pasos:

- Elegir el Root Bridge de la red en base al BID más bajo. Tomar en cuenta que sólo debe existir un Root Bridge por red y todos sus puertos deben ser designados.
- Elegir los puertos *root* de los *switches* en base al costo más bajo hacia el Root Bridge. Solamente debe existir un puerto *root* por *switch*.
- Elegir los puertos designados de cada segmento en base al costo más bajo hacia el Root Bridge. Solamente debe existir un puerto designado por segmento.
- Elegir los puertos alternos, es decir los puertos bloqueados que no envían tramas. Solamente debe existir un puerto alterno por segmento.

Con esto se elige una ruta principal para evitar bucles de capa dos. Sin embargo, cuando algún puerto deja de detectar tramas STP de su vecino lo considerará inalcanzable se perderá la comunicación. En este momento STP ejecuta los cuatro pasos previamente mencionados para elegir una nueva ruta y, de ser necesario, un nuevo Root Bridge, alterando así el rol de todos los puertos de los *switches*.

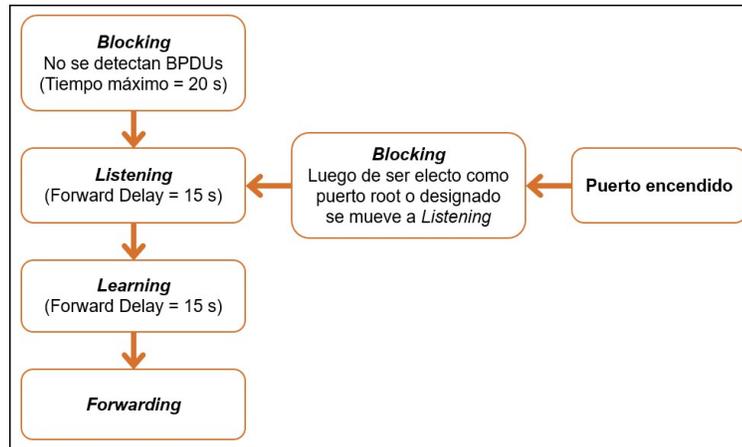
STP debe encontrar una nueva ruta lógica, y durante este período de tiempo cada puerto pasa por cinco estados antes de ser electos como *root*, designado o alterno también conocido como no designado. Estos estados son denominados: *Disabled*, *Blocking*, *Listening*, *Learning* y *Forwarding*. En las figuras 135 y 136 se describen los estados por los que deben pasar los puertos para lograr la convergencia de la red.

Figura 135. **Estados de los puertos de STP**



Fuente: elaboración propia.

Figura 136. **Proceso para estabilizar el estado del puerto en STP**



Fuente: elaboración propia.

Para que el puerto pase del estado *Blocking* a *Forwarding* debe permanecer 15 segundos en *Listening* y 15 segundos en *Learning* sumando un tiempo de, al menos, 30 segundos para encontrar una nueva ruta, lograr la convergencia y restablecer la comunicación en la red. A lo largo de los años IEEE fue mejorando STP para reducir el tiempo de convergencia y acoplarse a las necesidades actuales.

2.9.6. Variantes de STP

A continuación, se brinda una breve descripción de las tres variantes del protocolo Spanning Tree creadas por el IEEE.

2.9.6.1. STP

El protocolo Spanning Tree es estandarizado bajo la norma IEEE 802.1d y es capaz de crear solamente una instancia STP para toda la red L2, su convergencia es lenta, al menos 30 segundos, y requiere pocos recursos del

switch para ser ejecutado. Maneja tres roles para los puertos: *root*, designado y alterno o no designado, los cuales pueden pasar por cinco estados: *Disabled*, *Blocking*, *Listening*, *Learning* y *Forwarding*.

2.9.6.2. RSTP

El protocolo Rapid Spanning Tree es estandarizada bajo la norma IEEE 802.1w. Con RSTP el tiempo de convergencia se reduce considerablemente, menos de 10 segundos, continúa creando una instancia STP para toda la red L2 y cuenta con una demanda de recursos media. RSTP maneja cinco roles para los puertos: *root*, designado, alterno, *backup* y deshabilitado, los cuales pueden pasar por tres estados: *Discarding*, *Learning* y *Forwarding*.

En la tabla XI, se muestra un cuadro comparativo entre los roles de los puertos en STP y RSTP.

Tabla XI. Roles de los puertos en STP y RSTP

Rol del puerto en RSTP	Rol del puerto en STP	Descripción
<i>Root</i>	<i>Root</i>	Puerto donde se recibe el mejor BPDU que dirige hacia el Root Bridge.
Designado	Designado	Todos los puertos del Root Bridge. En un <i>switch</i> que no es el Root Bridge, será el puerto que anuncie el mejor BPDU.
Alternativo	Alternativo o no designado	Puerto que recibe un BPDU subóptimo.
Backup	-	Puerto que está conectado al mismo segmento, dominio de colisión, que otro puerto del mismo <i>switch</i> . Generalmente se presenta cuando se usan <i>hubs</i> como concentradores de red.
Deshabilitado	-	Un puerto administrativamente apagado o defectuoso.

Fuente: elaboración propia.

En la tabla XII, se muestra un cuadro comparativo entre los estados de los puertos en STP y RSTP.

Tabla XII. **Estado de los puertos en STP y RSTP**

Estado del puerto RSTP	Estado del puerto STP	Descripción
Discarding	Disabled	El puerto no puede aprender direcciones MAC ni tramas de datos del usuario.
Discarding	Blocking	
Discarding	Listening	
Learning	Learning	El puerto puede aprender direcciones MAC pero no puede reenviar tramas de datos del usuario.
Forwarding	Forwarding	El puerto puede aprender direcciones MAC y reenviar tramas de datos del usuario.

Fuente: elaboración propia.

2.9.6.3. MSTP

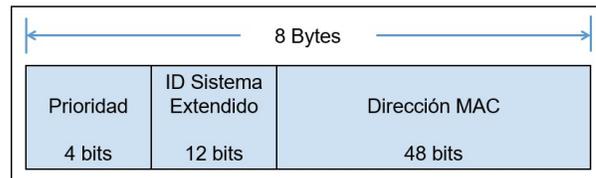
El protocolo Multiple Spanning Tree es estandarizado bajo la norma IEEE 802.1s y es la variable que todos los *switches* Huawei traen habilitada por defecto. Cuenta con la rápida convergencia de RSTP, pero tiene la gran ventaja de permitir crear instancias STP para agrupar VLAN y así elegir un Root Bridge por instancia. Su consumo de recursos es alto ya que maneja una topología STP por cada instancia. El rol y estado de sus puertos es el mismo empleado en RSTP.

2.9.7. Bridge ID para RSTP y MSTP

La razón por la cual la prioridad del Bridge ID debe ser configurada en saltos de 4096 en RSTP y MSTP es debido a una reducción al campo Prioridad para agregar instancias STP a cada VLAN. De esta forma la prioridad se reduce a 4 bits dando lugar a un nuevo campo llamado ID de Sistema Extendido con una longitud de 12 bits y que tiene el propósito de identificar la instancia STP.

En la figura 137 se muestra la nueva estructura del Bridge ID cuando se implementa RSTP o MSTP.

Figura 137. **Bridge ID con sistema extendido**



Fuente: elaboración propia.

2.9.8. Edge Port y BPDU Protection

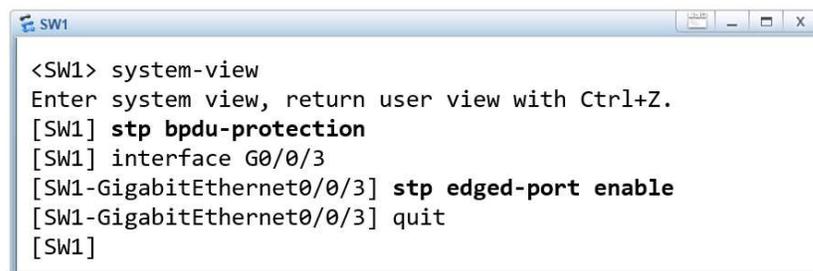
Si un puerto conecta directamente hacia un *host* final entonces no es necesario que sea incluido en el proceso STP ya que, a través de él, no se puede llegar al Root Bridge ni se pueden crear bucles en la red. A este tipo de puertos especiales se les denomina Edge Port.

En general, un Edge Port es un puerto que pasa directamente del estado *Disabled* al *Forwarding* para aprender direcciones MAC, reenviar las tramas de datos del usuario y acortar el tiempo que el *host* final se mantiene sin comunicación hacia la red. A través de un Edge Port no se deben recibir tramas

STP, de lo contrario se entenderá que se conectó otro *switch*, lo que puede provocar bucles de capa dos. Para proteger un Edge Port de recibir BPDU de otro *switch* se debe habilitar la opción de filtrado BPDU a nivel global, con esta configuración el puerto Edge se apagará si se recibe una BPDU.

Para configurar un puerto como Edge se ingresa a la vista de la interfaz y se emplea el comando: *stp edged-port enable*. Para activar la protección de PBDUs a nivel global, se ingresa a la vista del sistema y se emplea el comando: *stp bpdu-protection*.

Figura 138. **Configurando BPDU Protection y Edge Port**

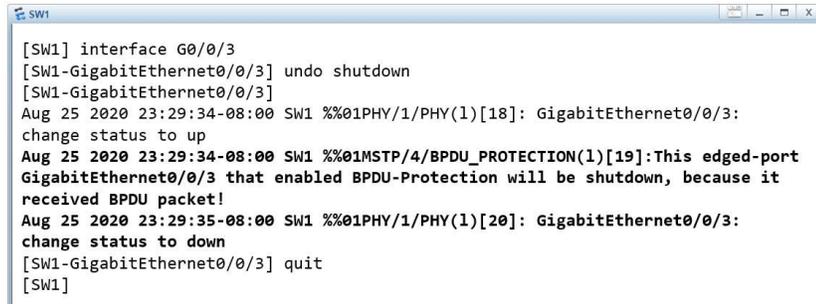


```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] stp bpdu-protection
[SW1] interface G0/0/3
[SW1-GigabitEthernet0/0/3] stp edged-port enable
[SW1-GigabitEthernet0/0/3] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

En la imagen 139 se muestra el error registrado en el *logbuffer* o historial del *switch* cuando se recibe un BPDU por un puerto declarado como Edge.

Figura 139. Puerto bloqueado por recibir BPDU



```
[SW1] interface G0/0/3
[SW1-GigabitEthernet0/0/3] undo shutdown
[SW1-GigabitEthernet0/0/3]
Aug 25 2020 23:29:34-08:00 SW1 %%01PHY/1/PHY(1)[18]: GigabitEthernet0/0/3:
change status to up
Aug 25 2020 23:29:34-08:00 SW1 %%01MSTP/4/BPDU_PROTECTION(1)[19]:This edged-port
GigabitEthernet0/0/3 that enabled BPDU-Protection will be shutdown, because it
received BPDU packet!
Aug 25 2020 23:29:35-08:00 SW1 %%01PHY/1/PHY(1)[20]: GigabitEthernet0/0/3:
change status to down
[SW1-GigabitEthernet0/0/3] quit
[SW1]
```

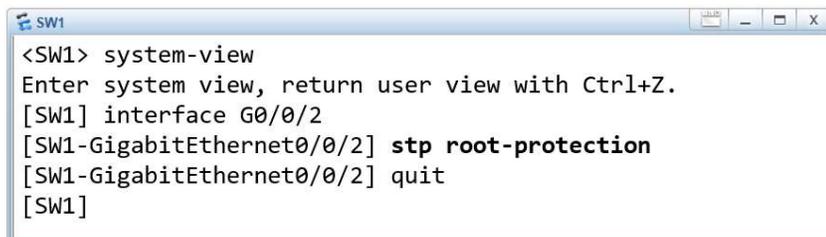
Fuente: elaboración propia, empleando eNSP.

2.9.9. Root Protection

La función Root Protection en un *switch* tiene el propósito de asegurar que un puerto se mantenga como Root Port a pesar de recibir mejores BPDU por otro puerto. La finalidad de esta protección es asegurar que un *switch* determinado se mantenga como Root Bridge de la red y evitar que le elija otro por ataques maliciosos.

Para configurar Root Protection en un puerto se ingresa a la vista de la interfaz y se emplea el comando: *stp root-protection*, como se muestra en la figura 140.

Figura 140. Configuración de Root Protection



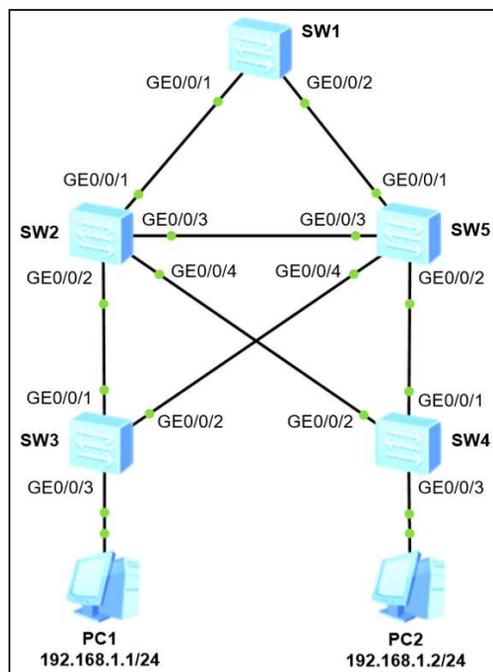
```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] interface G0/0/2
[SW1-GigabitEthernet0/0/2] stp root-protection
[SW1-GigabitEthernet0/0/2] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

2.9.10. Implementando RSTP

Para implementar RSTP se toma como ejemplo la red Broadcast mostrada en la figura 141, en donde existe únicamente la VLAN 1, y por tal motivo todos los puertos están configurados en modo acceso.

Figura 141. Red Broadcast para implementar RSTP



Fuente: elaboración propia, empleando eNSP.

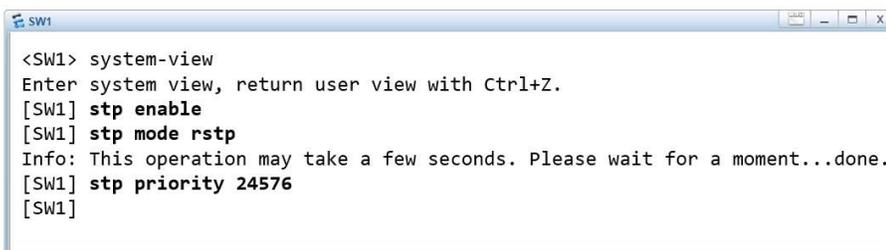
En todos los *switches* Huawei el protocolo Spanning Tree viene activado por defecto, no obstante, si se requiere activarlo de forma manual se debe ingresar a la vista del sistema y emplear el comando: *stp enable*. Adicional a esto, todos los *switches* vienen con MSTP activado, para cambiar la versión a RSTP se emplea el comando: *stp mode {stp | rstp | mstp}*, y se elige *rstp*.

Para cambiar la prioridad del Bridge ID se tienen dos opciones: la primera, y más recomendada, es emplear el comando: `stp priority bridge_id`, la segunda es emplear el comando: `stp root primary`, que asigna una prioridad de 0, asegurando que el *switch* se convierta en Root Bridge. También se puede elegir un segundo *switch* que se convertirá en Root Bridge si el principal deja de estar disponible, para esto se emplear el comando: `stp root secondary`, que asigna una prioridad de 4 096.

Llegado a este punto es importante recordar las instancias STP. Como se mencionó anteriormente una instancia STP permite agrupar VLAN y crear una topología distinta por cada grupo. Por defecto, los *switches* ya vienen con la instancia 0 creada y agrupa a la VLAN 1 únicamente.

Para este ejemplo se asigna una prioridad de 24 576 al *switch* SW1 para asegurar que sea electo como Root Bridge y una prioridad de 28 672 al *switch* SW2 para ser electo Root Bridge si SW1 deja de estar disponible. Adicional se activa la protección BPDU a los *switches* SW3 y SW4 y se convierte en Edge los puertos que conectan directamente hacia los *hosts*, es decir GE0/0/3. De la figura 142 a la 145 se muestra la configuración de cada *switch*.

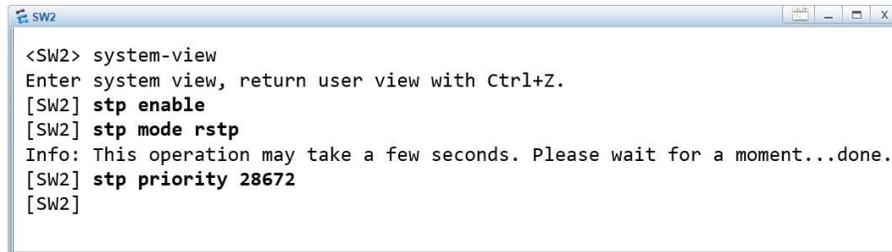
Figura 142. **Configuración de SW1**



```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] stp enable
[SW1] stp mode rstp
Info: This operation may take a few seconds. Please wait for a moment...done.
[SW1] stp priority 24576
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

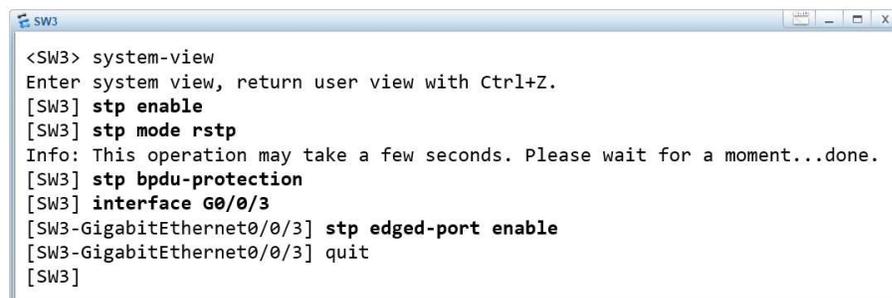
Figura 143. Configuración de SW2

A screenshot of a terminal window titled 'SW2'. The terminal shows the following commands and output:

```
<SW2> system-view
Enter system view, return user view with Ctrl+Z.
[SW2] stp enable
[SW2] stp mode rstp
Info: This operation may take a few seconds. Please wait for a moment...done.
[SW2] stp priority 28672
[SW2]
```

Fuente: elaboración propia, empleando eNSP.

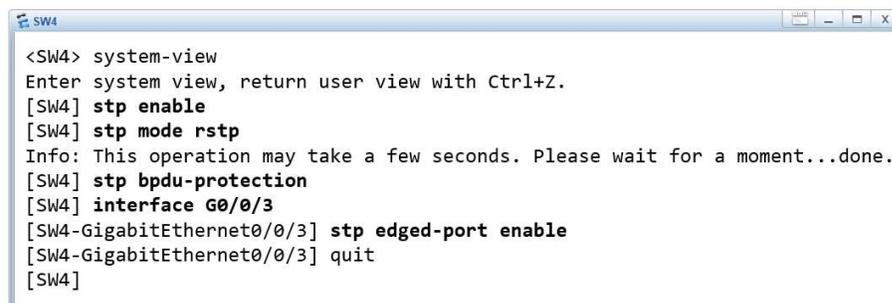
Figura 144. Configuración de SW3

A screenshot of a terminal window titled 'SW3'. The terminal shows the following commands and output:

```
<SW3> system-view
Enter system view, return user view with Ctrl+Z.
[SW3] stp enable
[SW3] stp mode rstp
Info: This operation may take a few seconds. Please wait for a moment...done.
[SW3] stp bpdu-protection
[SW3] interface G0/0/3
[SW3-GigabitEthernet0/0/3] stp edged-port enable
[SW3-GigabitEthernet0/0/3] quit
[SW3]
```

Fuente: elaboración propia, empleando eNSP.

Figura 145. Configuración de SW4

A screenshot of a terminal window titled 'SW4'. The terminal shows the following commands and output:

```
<SW4> system-view
Enter system view, return user view with Ctrl+Z.
[SW4] stp enable
[SW4] stp mode rstp
Info: This operation may take a few seconds. Please wait for a moment...done.
[SW4] stp bpdu-protection
[SW4] interface G0/0/3
[SW4-GigabitEthernet0/0/3] stp edged-port enable
[SW4-GigabitEthernet0/0/3] quit
[SW4]
```

Fuente: elaboración propia, empleando eNSP.

Con la configuración anterior, la ruta de comunicación queda como se muestra en la figura 146.

Figura 147. Configuración STP en SW3

```

<SW3> display stp instance 0
-----[CIST Global Info][Mode RSTP]----- 1
CIST Bridge      :32768.4c1f-cc09-1b43 2
Config Times    :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
Active Times    :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
CIST Root/ERPC  :24576.4c1f-cc0e-61a9 / 40000 3
CIST RegRoot/IRPC :32768.4c1f-cc09-1b43 / 0
CIST RootPortId :128.1 4
BPDU-Protection :Enabled 5
TC or TCN received :114
TC count per hello :0
STP Converge Mode :Normal
Time since last TC :0 days 0h:2m:9s 6
Number of TC      :40
Last TC occurred  :GigabitEthernet0/0/1
---- More ----

```

- 1 Versión de STP
- 2 Bridge ID
- 3 Root ID / Costo hacia el Root
- 4 Port ID del puerto Root
- 5 Protección BPDU
- 6 Tiempo desde el último cambio de topología (TC)

Fuente: elaboración propia, empleando eNSP.

Para verificar el rol, estado, protección BPDU de los puertos y la instancia a la que pertenecen se emplea el comando: *display stp brief*. Por defecto, todos los puertos perteneces a la instancia 0, como se observa en la figura 148.

Figura 148. Rol y estado de los puertos en SW3

```

<SW3> display stp brief
MSTID Port          Role STP State Protection
0      GigabitEthernet0/0/1  ROOT FORWARDING NONE
0      GigabitEthernet0/0/2  ALTE DISCARDING NONE
0      GigabitEthernet0/0/3  DESI FORWARDING BPDU
<SW3>

```

Fuente: elaboración propia, empleando eNSP.

Para revisar la configuración de RSTP de cada puerto se puede emplear el comando: *display stp interface interface_type interface_number*. En la figura 149 se observa la configuración RSTP del puerto GE0/0/3 del switch SW3.

Figura 149. Configuración RSTP de GE0/0/3 de SW3

```

<SW3> display stp interface G0/0/3
----[Port3(GigabitEthernet0/0/3)][FORWARDING]---- ①
Port Protocol      :Enabled
Port Role          :Designated Port ②
Port Priority       :128
Port Cost(Dot1T ) :Config=auto / Active=20000 ③
Designated Bridge/Port :32768.4c1f-cc09-1b43 / 128.3 ④
Port Edged         :Config=enabled / Active=enabled ⑤
BPDU-Protection    :Enabled
Point-to-point     :Config=auto / Active=true
Transit Limit      :147 packets/hello-time
Protection Type    :None
Port STP Mode      :RSTP
Port Protocol Type :Config=auto / Active=dot1s
BPDU Encapsulation :Config=stp / Active=stp
PortTimes          :Hello 2s MaxAge 20s FwDly 15s RemHop 20 ⑥
TC or TCN send     :0
TC or TCN received :0
BPDU Sent          :239
                   TCN: 0, Config: 0, RST: 239, MST: 0
BPDU Received      :0
                   TCN: 0, Config: 0, RST: 0, MST: 0
<SW3>

```

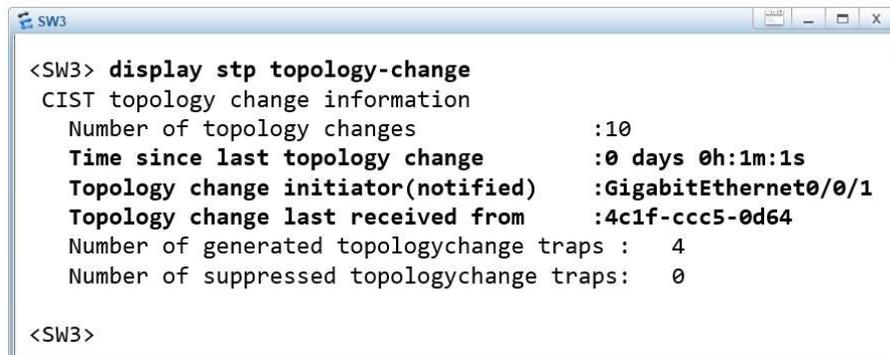
- | | |
|---------------------|-----------------------|
| ① Estado del puerto | ④ Bridge ID / Port ID |
| ② Rol del puerto | ⑤ Edge Port |
| ③ Costo del puerto | ⑥ Timers |

Fuente: elaboración propia, empleando eNSP.

Finalmente, otra opción útil para verificar la estabilidad de la red es ver el tiempo desde la última reconvergencia STP y así encontrar el evento que la ocasionó. Para esto se emplea el comando: *display stp topology-change*, que muestra el tiempo desde el último cambio de topología y el puerto que notificó dicho cambio.

En la figura 150, se muestra que el último cambio de topología o TC fue hace 1 minutos 1 segundo y fue notificado por G0/0/1

Figura 150. **Cambio de topología STP**



```
<SW3> display stp topology-change
CIST topology change information
Number of topology changes           :10
Time since last topology change      :0 days 0h:1m:1s
Topology change initiator(notified)  :GigabitEthernet0/0/1
Topology change last received from   :4c1f-ccc5-0d64
Number of generated topologychange traps : 4
Number of suppressed topologychange traps: 0

<SW3>
```

Fuente: elaboración propia, empleando eNSP.

2.10. Enlaces Ethernet-Trunk

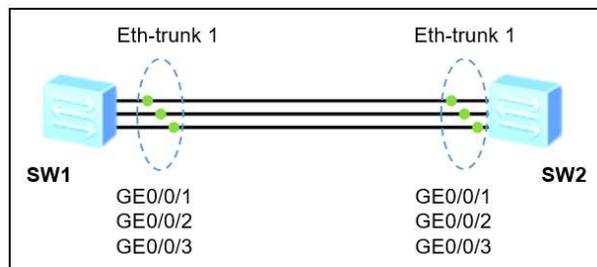
Un Ethernet-Trunk es un enlace lógico que agrupa varios enlaces físicos de la misma capacidad con el propósito de aumentar el BW e implementar balanceo de cargas entre ellos. Frecuentemente esta tecnología se encuentra a nivel WAN en redes Metro Ethernet, no obstante, también está presente en redes LAN cuando se requiere manejar un tráfico elevado de datos.

Un enlace Eth-Trunk se puede crear manualmente o de forma automática empleando el protocolo LACP que está regido bajo el estándar IEEE 802.3ad.

2.10.1. Implementando enlaces Eth-Trunk manual

En la topología mostrada en la figura 151, se requiere crear un enlace Eth-Trunk con capacidad de 3Gbps que comunique al *switch* SW1 y SW2. Para esto es necesario crear un puerto Eth-Trunk que incluya tres puertos GigabitEthernet.

Figura 151. **Enlace Eth-Trunk entre SW1 y SW2**



Fuente: elaboración propia, empleando eNSP.

Para crear un puerto Eth-Trunk se ingresa a la vista del sistema y se emplea el comando: `interface eth-trunk interface_number`, donde el parámetro `interface_number` es un número entero que identifica al puerto Ethernet-Trunk y solo tiene importancia a nivel local, variar estar entre 0 y 63. Posteriormente se indica el modo para establecer el puerto eth-trunk con el comando: `mode {lacp | manual} load-balance`, y se elige `manual` para crear el enlace manualmente. Finalmente se agregan los puertos físicos al puerto Eth-Trunk con el comando: `trunkport interfece_type interface_number to interface_number`.

Luego de la creación del puerto Eth-Trunk se procede a configurarlo en modo troncal para manejar tráfico etiquetado o en modo acceso para manejar tráfico sin etiqueta.

En la figura 152, se observa la configuración de SW1 para crear el puerto Eth-trunk 1 conformado por los puertos físicos GE0/0/1 al GE0/0/3. Es importante mencionar que el número de puerto Eth-trunk tiene importancia únicamente a nivel local y no necesariamente debe coincidir en ambos *switches* para formar el enlace Eth-Trunk. Tampoco es necesario que coincidan los números de puertos que conforman el enlace Eth-Trunk, sin embargo, si debe coincidir el tipo de

puerto y la negociación, que para este ejemplo se utilizan puertos GigabitEthernet.

Figura 152. Configuración de SW1

```
<SW1> system-view
Enter system view, return user view with Ctrl+Z.
[SW1] interface eth-trunk 1
[SW1-Eth-Trunk1] mode manual load-balance
[SW1-Eth-Trunk1] trunkport Gigabitethernet 0/0/1 to 0/0/3
[SW1-Eth-Trunk1] port link-type trunk
[SW1-Eth-Trunk1] port trunk allow-pass vlan all
[SW1-Eth-Trunk1] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

La configuración del *switch* SW2 es exactamente igual.

2.10.2. Verificando la configuración de Eth-Trunk

Para revisar la configuración de un puerto Eth-Trunk y verificar los puertos físicos que lo confirman, su estado, el BW, la dirección MAC y la PVID se puede emplear el comando: *display interface eth_trunk interface_number*, como se muestra en la figura 153.

Figura 153. Estado del Puerto Eth-Trunk 1 en SW1

```
<SW1> display interface Eth-Trunk 1
Eth-Trunk1 current state : UP
Line protocol current state : UP
Description:
Switch Port, PVID : 1, Hash arithmetic : According to SIP-XOR-DIP, Maximal BW: 3G,
Current BW: 3G, The Maximum Frame Length is 9216
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 4c1f-ccfb-7381
Current system time: 2020-08-31 22:29:19-08:00
Input bandwidth utilization : 0%
Output bandwidth utilization : 0%
-----
PortName                Status    Weight
-----
GigabitEthernet0/0/1    UP        1
GigabitEthernet0/0/2    UP        1
GigabitEthernet0/0/3    UP        1
-----
The Number of Ports in Trunk : 3
The Number of UP Ports in Trunk : 3
```

Fuente: elaboración propia, empleando eNSP.

Otra información importante que se puede revisar en un puerto Eth-Trunk es el modo de trabajo, la cantidad mínima de puertos activos que se debe tener o *Least Active-linknumber*, la cantidad máxima de puertos que pueden conformarlo o *Max Bandwidth-affected-linknumber* y la cantidad de puertos actualmente activos con su peso, *weight*, para esto se emplea el comando: *display eth-trunk interface_number*, como se muestra en la figura 154.

Figura 154. Estado del Puerto Eth-Trunk 1 de SW1

```
<SW1> display eth-trunk 1
Eth-Trunk1's state information is:
WorkingMode: NORMAL      Hash arithmetic: According to SIP-XOR-DIP
Least Active-linknumber: 1 Max Bandwidth-affected-linknumber: 8
Operate status: up      Number Of Up Port In Trunk: 3
-----
PortName                Status    Weight
-----
GigabitEthernet0/0/1    Up        1
GigabitEthernet0/0/2    Up        1
GigabitEthernet0/0/3    Up        1
-----
<SW1>
```

Fuente: elaboración propia, empleando eNSP.

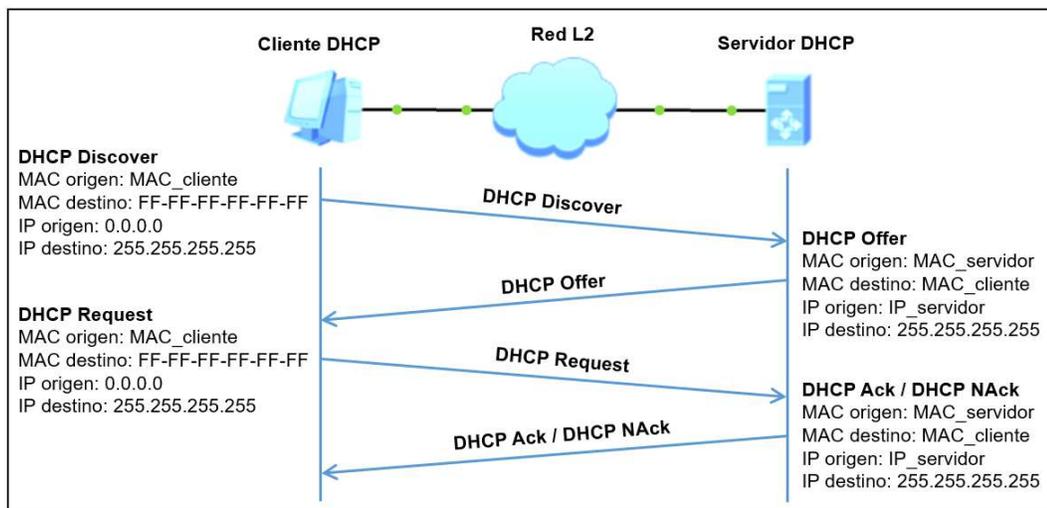
2.11. Servicios IP

Entre los servicios IP más importantes se pueden mencionar el direccionamiento dinámico por DHCP, uso de ACL para bloqueo de tráfico, NAT para acceder a internet y VRRP para el manejo de redundancia a nivel LAN.

2.11.1. DHCP

Dynamic Host Configuration Protocol o DHCP es un protocolo que permite la asignación de direcciones IPv4 de forma dinámica a través de un servidor. Su funcionamiento se basa en cuatro pasos llamados: DHCPDiscover, DHCPOffer, DHCPRequest y DHCPAck/DHCPSNack, como se muestra en la figura 155.

Figura 155. Proceso DHCP



Fuente: elaboración propia, empleando eNSP.

A continuación, se brinda una breve descripción de cada paso que se da entre el cliente y el servidor DHCP para obtener una dirección IPv4.

- DHCP Discover: el proceso inicia cuando el cliente envía un mensaje DHCP Discover con el propósito de descubrir o encontrar algún servidor DHCP en la red. El DHCP Discover es un mensaje Broadcast con dirección IP de origen: 0.0.0.0, dirección IP de destino: 255.255.255.255 y protocolo 0x11. La dirección MAC de origen es la que posee el cliente en su interfaz de salida y la dirección MAC de destino es: FF-FF-FF-FF-FF-FF. El puerto de destino del datagrama es UDP 67 y el puerto de origen UDP 68.
- DHCP Offer: el mensaje DHCP Discover llega al servidor quien consulta en un listado o *pool* de direcciones alguna que esté disponible. Luego envía esta dirección IPv4 disponible al cliente por medio de un DHCP Offer encapsulado en un mensaje Broadcast. El mensaje Broadcast tiene la dirección IP de origen del servidor DHCP y dirección IP de destino: 255.255.255.255. La dirección MAC de origen es la que posee el servidor DHCP en su interfaz de salida y la dirección MAC de destino es la del cliente. El puerto de destino del datagrama es UDP 68 y el puerto de origen UDP 67.
- DHCP Request: en una red pueden existir varios servidores DHCP, es por esta razón que el cliente elige la primera dirección IPv4 recibida a través del DHCP Offer. Seguidamente, el cliente envía un mensaje DHCP Request dirigido al servidor seleccionado para solicitar la dirección IPv4 recibida. El DHCP Request es recibido por todos los servidores DHCP y su función es informarles que ya se seleccionó un servidor, además de confirmar la dirección IP. DHCP Request tiene la dirección IP de origen 0.0.0.0 y la dirección IP de destino: 255.255.255.255. La dirección MAC de origen es la que posee el cliente en su interfaz de salida y la dirección MAC de destino es FF-FF-FF-FF-FF-FF.

- DHCP Acknowledgment / No-Acknowledgment: Luego que el servidor recibe el DHCP Request, responde con un mensaje DHCP Ack para confirmar la dirección asignada al cliente. Cuando el cliente recibe este mensaje procede a configurar la IP adquirida para su uso por primera vez. Si por alguna razón el servidor no acepta el DHCP Request entonces responde con un DHCP NACK y el proceso se repite nuevamente. DHCP Ack tiene la dirección IP de origen del servidor DHCP y la dirección IP de destino 255.255.255.255. La dirección MAC de origen es la que posee el servidor DHCP en su interfaz de salida y la dirección MAC de destino es la que posee el cliente.

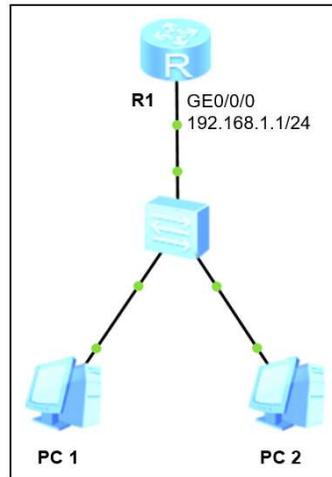
2.11.1.1. Agente DHCP Relay

En muchas ocasiones se tendrán escenarios donde el servidor DHCP no se encuentre en la misma red capa dos que los *hosts* a los cuales se requiere brindar servicio de asignación de direcciones. En estos casos los *routers* Huawei pueden ser configurados como agentes DHCP Relay cuyo propósito es convertir los mensajes Broadcast, usados en los 4 pasos del proceso, en Multicast para ser enviados fuera de la red hacia el servidor DHCP externo.

2.11.1.2. Implementando DHCP

Para configurar un *router* como servidor DHCP se toma como ejemplo la topología mostrada en la figura 156.

Figura 156. **Topología para implementar DHCP**



Fuente: elaboración propia, empleando eNSP.

La configuración inicia al activar DHCP a nivel global en el *router*, para esto se ingresa a la vista del sistema y se emplea el comando: *dhcp enable*. El siguiente paso es crear la lista o *pool* de direcciones IP para ser asignadas a los *hosts*, para esto se emplea el comando: *ip pool pool_name*, donde el parámetro *pool_name* es el nombre del *pool*. Con el comando: *network network_address mask mask_length*, se establece el segmento de red para el *pool*. Luego se establece el *gateway* y opcionalmente los servidores DNS, para esto se emplean los comandos: *gateway-list ip_gateway* y *dns_list ip_dns*, respectivamente. Es importante mencionar que la dirección IP del gateway tiene que estar configurada en alguna interfaz activa del *router*.

Por defecto, el *router* no asigna la dirección de red, la dirección Broadcast ni el *gateway* especificado a los *hosts*. Finalmente se puede establecer el tiempo máximo que un *host* puede poseer la dirección IP recibida antes de renovarla con el servidor, esto se logra con el comando: *lease day day_num hour hour_num minute min_num*. Por defecto el cliente renueva la dirección cada 24 horas.

La asignación de direcciones IP puede ser en base a un *pool* global o a un *pool* por interfaz. Para que el *router* asigne direcciones en base a un *pool* global se ingresa a la vista de la interfaz donde está configurado el *gateway* y se emplea el comando: *dhcp select global*.

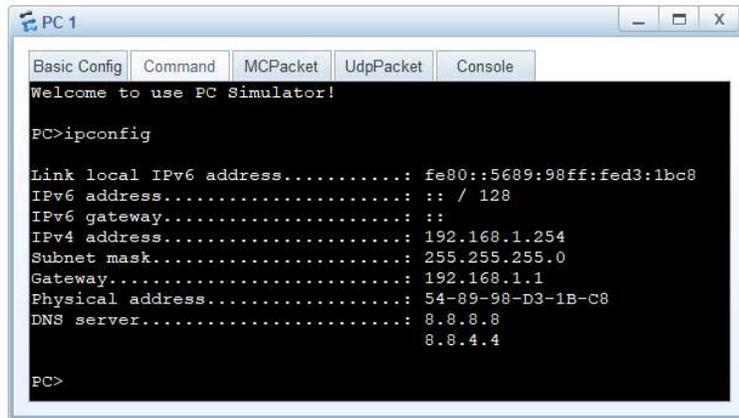
Figura 157. Configuración de R1

```
[R1] dhcp enable
Info: The operation may take a few seconds. Please wait for a moment...done.
[R1] ip pool red_lan
Info: It's successful to create an IP address pool.
[R1-ip-pool-red_lan_1] network 192.168.1.0 mask 24
[R1-ip-pool-red_lan_1] gateway-list 192.168.1.1
[R1-ip-pool-red_lan_1] dns-list 8.8.8.8 8.8.4.4
[R1-ip-pool-red_lan_1] lease day 1 hour 0 minute 0
[R1-ip-pool-red_lan_1] quit
[R1] interface G0/0/0
[R1-GigabitEthernet0/0/0] ip add 192.168.1.1 24
[R1-GigabitEthernet0/0/0] dhcp select global
[R1-GigabitEthernet0/0/0] undo shutdown
[R1-GigabitEthernet0/0/0] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 158, se puede observar la dirección IP recibida mediante DHCP en PC 1.

Figura 158. Dirección IP de PC 1 obtenida por DHCP



```
PC 1
Basic Config  Command  MCPacket  UdpPacket  Console
Welcome to use PC Simulator!
PC>ipconfig
Link local IPv6 address.....: fe80::5689:98ff:fed3:1bc8
IPv6 address.....: :: / 128
IPv6 gateway.....: ::
IPv4 address.....: 192.168.1.254
Subnet mask.....: 255.255.255.0
Gateway.....: 192.168.1.1
Physical address.....: 54-89-98-D3-1B-C8
DNS server.....: 8.8.8.8
                  8.8.4.4
PC>
```

Fuente: elaboración propia, empleando eNSP.

2.11.1.3. Verificando la configuración de DHCP

Para verificar la cantidad de direcciones IP usadas, las disponibles, las que cuentan con algún conflicto o están deshabilitadas del *pool* de direcciones se emplea el comando: *display ip pool name name_pool used*, donde *name_pool* es el nombre del pool, como se muestra en la figura 159.

Figura 159. Dirección IP de PC 1 obtenida por DHCP

```
<R1> display ip pool name red_lan used
Pool-name      : red_lan
Pool-No       : 0
Lease         : 1 Days 0 Hours 0 Minutes
Domain-name   : -
DNS-server0   : 8.8.8.8
DNS-server1   : 8.8.4.4
NBNS-server0  : -
Netbios-type  : -
Position      : Local          Status      : Unlocked
Gateway-0    : 192.168.1.1
Mask         : 255.255.255.0
VPN instance  : --

-----
      Start      End      Total  Used  Idle(Expired)  Conflict  Disable
-----
    192.168.1.1 192.168.1.254  253    2    251(0)         0         0
-----

Network section :
-----
Index      IP          MAC          Lease  Status
-----
   252    192.168.1.253  5489-98f9-3db1    15  Used
   253    192.168.1.254  5489-98d3-1bc8   2073  Used
-----

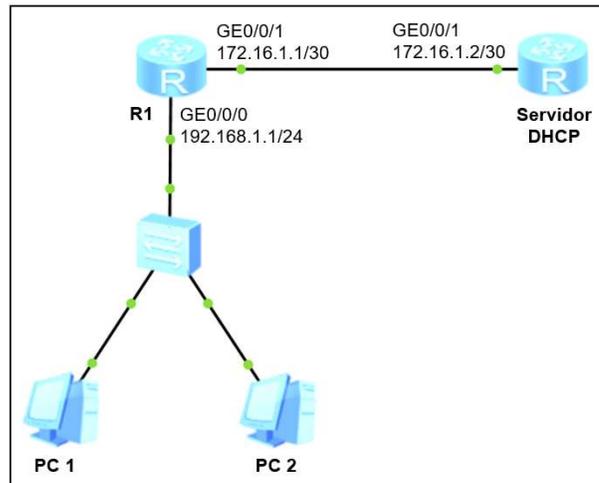
<R1>
```

Fuente: elaboración propia, empleando eNSP.

2.11.1.4. Implementando agente DHCP Relay

Para mostrar la configuración de un *router* como agente DHCP Relay se toma como ejemplo la topología mostrada en la figura 160.

Figura 160. **Topología para implementar *router* como agente DHCP Relay**



Fuente: elaboración propia, empleando eNSP.

En esta ocasión R1 es el agente DHCP Relay así que encapsulará todo el Broadcast recibido en su interfaz GE0/0/0 y la enviará por GE0/0/1 hacia el servidor DHCP externo. Para realizar esta acción se emplea el comando: *dhcp select relay* en la interfaz GE0/0/0 de R1. Adicionalmente se debe indicar la dirección IP del servidor DHCP externo con el comando: *dhcp relay server-ip ip_server_dhcp*, se pueden especificar hasta un máximo de 20 servidores DHCP externos.

En la figura 161, se muestra la configuración de R1 como agente DHCP Relay.

Figura 161. Configuración de R1

```
[R1] dhcp enable
Info: The operation may take a few seconds. Please wait for a moment...done.
[R1] interface G0/0/0
[R1-GigabitEthernet0/0/0] ip add 192.168.1.1 24
[R1-GigabitEthernet0/0/0] dhcp select relay
[R1-GigabitEthernet0/0/0] dhcp relay server-ip 172.16.1.2
[R1-GigabitEthernet0/0/0] undo shutdown
[R1-GigabitEthernet0/0/0] interface G0/0/1
[R1-GigabitEthernet0/0/1] ip address 172.16.1.1 30
[R1-GigabitEthernet0/0/1] undo shutdown
[R1-GigabitEthernet0/0/1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 162, se muestra la configuración del servidor DHCP.

Figura 162. Configuración del Servidor DHCP

```
[SERVIDOR_DHCP] dhcp enable
Info: The operation may take a few seconds. Please wait for a moment...done.
[SERVIDOR_DHCP] ip pool LAN_01
Info: It's successful to create an IP address pool.
[SERVIDOR_DHCP-ip-pool-LAN_01] network 192.168.1.0 mask 24
[SERVIDOR_DHCP-ip-pool-LAN_01] gateway-list 192.168.1.1
[SERVIDOR_DHCP-ip-pool-LAN_01] dns-list 8.8.8.8 8.8.4.4
[SERVIDOR_DHCP-ip-pool-LAN_01] quit
[SERVIDOR_DHCP] interface G0/0/1
[SERVIDOR_DHCP-GigabitEthernet0/0/1] ip add 172.16.1.2 30
[SERVIDOR_DHCP-GigabitEthernet0/0/1] dhcp select global
[SERVIDOR_DHCP-GigabitEthernet0/0/1] undo shutdown
[SERVIDOR_DHCP-GigabitEthernet0/0/1] quit
[SERVIDOR_DHCP] ip route-static 192.168.1.0 255.255.255.0 GigabitEthernet0/0/1 172.16.1.1
[SERVIDOR_DHCP]
```

Fuente: elaboración propia, empleando eNSP.

2.11.2. ACL

Las listas de control de acceso o ACL son normas que analizan el tráfico entrante o saliente del *router* para limitar, restringir o dar prioridad a ciertos

paquetes o tramas en base a la dirección IP de origen, dirección IP de destino, número de puerto, protocolo, entre otros parámetros. Están conformadas por un listado de sentencias o *rules* a los cuales se somete el paquete o trama para ser negado, *deny*, o permitido, *permit*.

Las ACL están identificadas con un número entero del 2 000 al 5 999 que las clasifica en cuatro diferentes tipos: listas de control de acceso básicas, avanzadas, de L2 y las definidas por el usuario. En la tabla XIII se muestra el tipo de ACL y el rango asignado.

Tabla XIII. **Número identificador de las ACL**

Tipo de ACL	Rango
Básica	2 000 – 2 999
Avanzada	3 000 – 3 999
L2	4 000 – 4 999
Definida por el usuario	5 000 – 5 999

Fuente: elaboración propia.

2.11.2.1. Análisis de las sentencias

Las sentencias o *rules* están identificadas con un número entero o ID que inicia en 5 y aumenta en 5 unidades a medida que son creadas. El valor de salto de 5 viene configurado por defecto en todos los equipos Huawei, sin embargo, puede ser modificado a conveniencia. El propósito del ID es ordenar de forma ascendente las sentencias para luego ser analizadas una a una.

Cuando se analiza un paquete o trama la primera sentencia que se revisa es la que tiene el ID 5. Si el paquete coincide con la *rule* entonces se realiza la acción correspondiente, *deny/permit*, y ya no entra en comparación con el resto de las sentencias. Por el contrario, si no coincide entonces se procede a analizar con la siguiente sentencia, es decir la que tiene el ID 10, si en dado caso existe. Este procedimiento se repite hasta que el paquete coincida con una sentencia y así se realice la acción correspondiente. Todas las ACL básicas y avanzadas tienen una sentencia final implícita que permite todo, *implicit permit*, así que, si el paquete no coincide con ninguna sentencia anterior, entonces es permitido y es procesado por el *router*.

Finalmente, la ACL debe ser configurada a alguna interfaz del *router* para analizar el tráfico de entrada, *inbound*, o de salida, *outbound*. En algunas ocasiones resulta difícil identificar si la ACL debe configurarse como entrada o salida, es por eso que se puede emplear la analogía de tomar el lugar físico del *router*, es decir, imaginar que las extremidades de una persona son las interfaces y la persona en sí es el *router*, con esto se puede analizar si el tráfico debe ser bloqueado antes de entrar al *router* o al salir de este. Una ACL también puede emplearse en las líneas VTY para restringir el acceso vía Telnet o SSH.

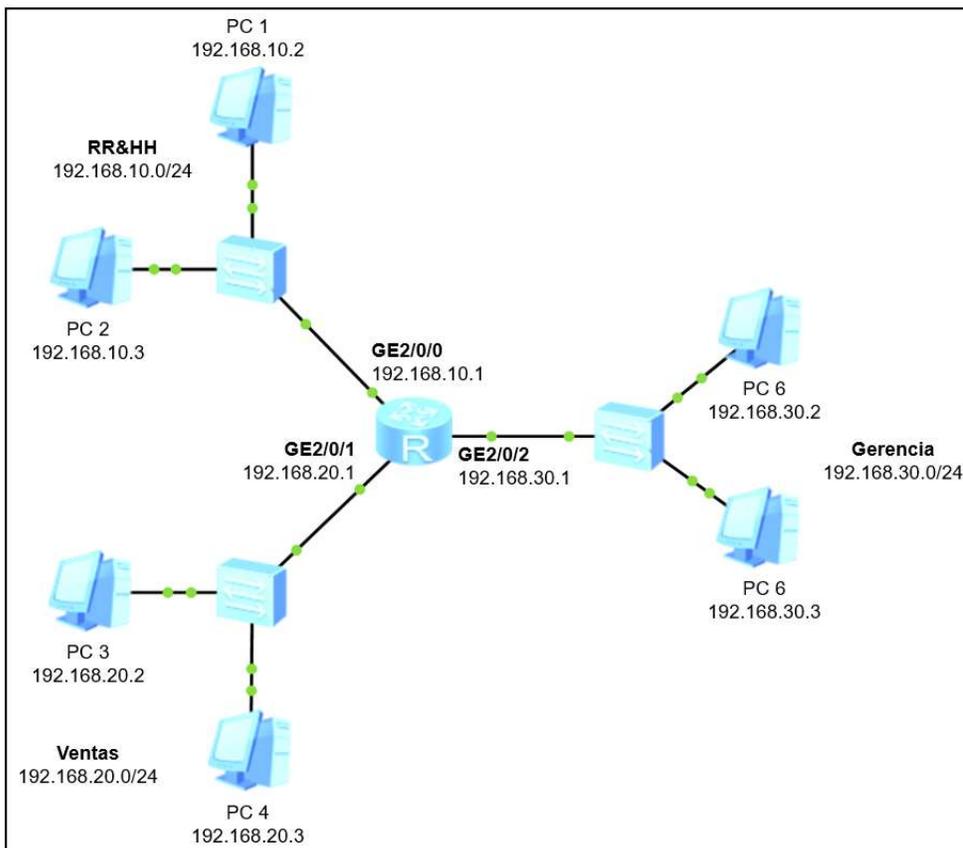
Por regla general, para optimizar el funcionamiento de la red, las ACL básicas se deben configurar como *outbound* en la interfaz más cercana al destino mientras que las ACL avanzadas deben configurarse como *inbound* en la interfaz más cercana al origen.

2.11.2.2. Implementando ACL básica

Una ACL básica únicamente verifica la dirección IP de origen del paquete, es por esta razón que debe ser configurada lo más cercana al destino posible. La

topología de ejemplo mostrada en la figura 163 servirá para implementar ACL básicas y avanzadas, contiene la red ventas con el segmento 192.168.10.0/24, recursos humanos con el segmento 192.168.20.0/24 y gerencia con el segmento 192.168.30.0/24.

Figura 163. Topología para implementar ACL



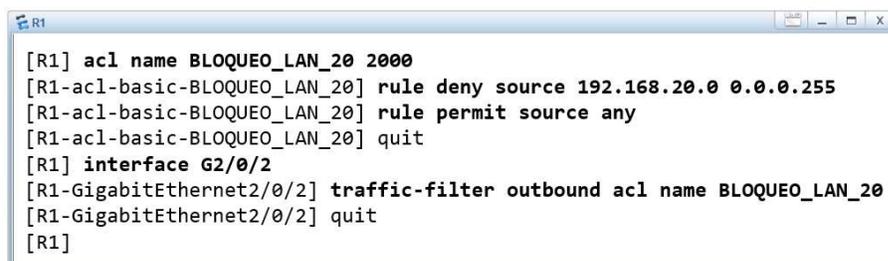
Fuente: elaboración propia, empleando eNSP.

Por cuestiones de seguridad es necesario bloquear la comunicación entre los *hosts* de la red ventas y los *hosts* de la red gerencia, para esto se puede emplear una ACL básica. Para crear la ACL básica se ingresa a la vista del sistema y se emplea el comando: `acl name acl_name acl_number`, donde el parámetro `acl_name` es el nombre para identificar a la ACL y el parámetro

acl_number es el identificador, que para ACL básicas se encuentra entre 2 000 y 2 999. Posteriormente se crea la primera sentencia con el comando: *rule {permit / deny} source ip_address wild_card_mask*, donde se debe definir si la ACL permitirá, permit, o negará, deny, el tráfico, el parámetro *ip_address* representa la dirección de red del segmento a analizar, y la máscara wildcard representa el alcance del segmento. Por último, para agregar la ACL a una interfaz se ingresa a la vista de la interfaz y se emplea el comando: *traffic-filter {outbound / inbound} acl name acl_name*.

En la figura 164, se muestra la configuración de R1 donde se crea una ACL llamada BLOQUEO_LAN_20 y se aplica como *outbound* en la interfaz GE2/0/2. Con esta configuración el *router* bloqueará todos los paquetes que sean originados en el segmento 192.168.20.0/24 y que intenten salir por la interfaz GE2/0/2, sin embargo, dejará pasar el resto de tráfico gracias al *permit* implícito al final de la ACL, por motivos didácticos se especifica la sentencia que permite el resto del tráfico. Los *hosts* que provengan de la red gerencia sí pueden alcanzar a los *hosts* de la red ventas, pero será una comunicación unidireccional, es decir, si se realiza una prueba de *ping* desde algún *host* de la red gerencia hacia algún *host* de la red ventas no será exitoso porque el *ICMP echo-request* es originado en el segmento 192.168.30.0/24 pero el *ICMP echo-reply* es originado en el segmento 192.168.20.0/24.

Figura 164. Creación de una ACL básica en R1



```
[R1] acl name BLOQUEO_LAN_20 2000
[R1-acl-basic-BLOQUEO_LAN_20] rule deny source 192.168.20.0 0.0.0.255
[R1-acl-basic-BLOQUEO_LAN_20] rule permit source any
[R1-acl-basic-BLOQUEO_LAN_20] quit
[R1] interface G2/0/2
[R1-GigabitEthernet2/0/2] traffic-filter outbound acl name BLOQUEO_LAN_20
[R1-GigabitEthernet2/0/2] quit
[R1]
```

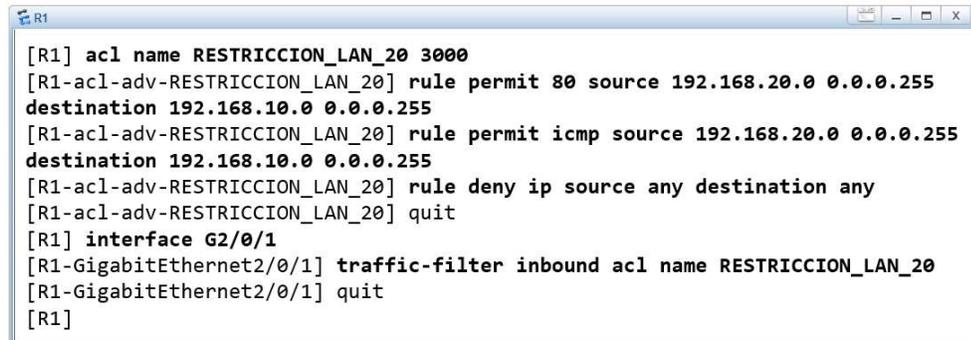
Fuente: elaboración propia, empleando eNSP.

2.11.2.3. Implementando ACL avanzada

Las ACL avanzadas permiten revisar la dirección IP de origen, dirección IP de destino y en número de puerto, es por esta razón que deben ser configuradas en la interfaz más cercana al origen. Para su creación se ingresa a la vista del sistema y se emplea el comando: *acl name acl_name acl_number*, donde el parámetro *acl_number* puede variar entre 3 000 y 3 999. Una vez creada la ACL se procede a configurar la primera sentencia con el comando: *rule {permit / deny} port_number source source_ip_address wild_card_mask destination destination_ip_address wild_card_mask*, donde el parámetro *port_number* puede ser el número de puerto por donde se establece la comunicación o alguna de las siguientes opciones: *gre, icmp, igmp, ip, ospf, tcp o udp*. Finalmente, la ACL se debe agregar a la interfaz con el comando: *traffic-filter {outbound / inbound} acl name acl_name*.

En la figura 165, se muestra la configuración de una ACL llamada *RESTRICCION_LAN_20* que permite únicamente el tráfico que se origine en el segmento 192.168.20.0/24 y que se dirija hacia el segmento 192.168.10.0/24 por el puerto 80 para *html* o empleando el protocolo ICMP para *ping*. Con esta configuración los equipos ubicados en la red ventas podrán alcanzar los servidores WEB y tener respuesta a nivel de ping de los equipos ubicados en la red RR&HH. Esta ACL está configurada como *inbound* en la interfaz GE2/0/1, es decir que el *router* verificará cada paquete que ingrese por la interfaz y si es aceptado por la ACL entonces consultará su tabla de enrutamiento para reenviarlo, de lo contrario será descartado.

Figura 165. Creación de una ACL avanzada en R1



```
[R1] acl name RESTRICCIÓN_LAN_20 3000
[R1-acl-adv-RESTRICCIÓN_LAN_20] rule permit 80 source 192.168.20.0 0.0.0.255
destination 192.168.10.0 0.0.0.255
[R1-acl-adv-RESTRICCIÓN_LAN_20] rule permit icmp source 192.168.20.0 0.0.0.255
destination 192.168.10.0 0.0.0.255
[R1-acl-adv-RESTRICCIÓN_LAN_20] rule deny ip source any destination any
[R1-acl-adv-RESTRICCIÓN_LAN_20] quit
[R1] interface G2/0/1
[R1-GigabitEthernet2/0/1] traffic-filter inbound acl name RESTRICCIÓN_LAN_20
[R1-GigabitEthernet2/0/1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

2.11.2.4. Verificando una ACL

Para verificar la configuración de una ACL se puede emplear el comando: *display acl acl_number*, donde el parámetro *acl_number* es el número que identifica a la ACL, también se pueden mostrar todas las ACL existentes con el comando: *display acl all*. Al ejecutar el comando se puede verificar las sentencias que confirman la ACL y los paquetes que han coincidido con cada una de ellas.

En la figura 166, se puede visualizar la configuración de las dos ACL creadas en R1 en el ejemplo anterior. Se observa que la ACL 2 000 ha tenido cinco coincidencias con la *rule 10* que permite paquetes con cualquier IP de origen, mientras que la ACL 3 000 ha contabilizado cinco coincidencias con la *rule 10* que permite el protocolo ICMP entre la red ventas y la red RR&HH y ha bloqueado 834 paquetes con la *rule 15* que niega el resto del tráfico, entre estos paquetes bloqueados están los BPDU recibidos por el *switch*.

Figura 166. Configuración de las ACL de R1

```
<R1> display acl all
Total quantity of nonempty ACL number is 2

Basic ACL BLOQUEO_LAN_20 2000, 2 rules
Acl's step is 5
rule 5 deny source 192.168.20.0 0.0.0.255
rule 10 permit (5 matches)

Advanced ACL RESTRICCIÓN_LAN_20 3000, 3 rules
Acl's step is 5
rule 5 permit 80 source 192.168.20.0 0.0.0.255 destination 192.168.10.0 0.0.0.255
rule 10 permit icmp source 192.168.20.0 0.0.0.255 destination 192.168.10.0 0.0.0.255 (5 matches)
rule 15 deny ip (834 matches)

<R1>
```

Fuente: elaboración propia, empleando eNSP.

2.11.3. NAT

Las direcciones IPv4 se dividen en públicas y privadas, las direcciones públicas son accesibles a través de internet mientras que las privadas son usadas exclusivamente dentro de un AS. La traducción de direcciones de red o NAT es una técnica que permite convertir una dirección privada en una pública para tener salida hacia Internet. En la tabla XIV se muestra el rango de direcciones privadas en base a su clase.

Tabla XIV. Direcciones IPv4 privadas

Clase	Rango
A	10.0.0.0 – 10.255.255.255
B	172.16.0.0 – 172.31.255.255
C	192.168.0.0 – 192.168.255.255

Fuente: elaboración propia.

NAT cuenta con tres variantes:

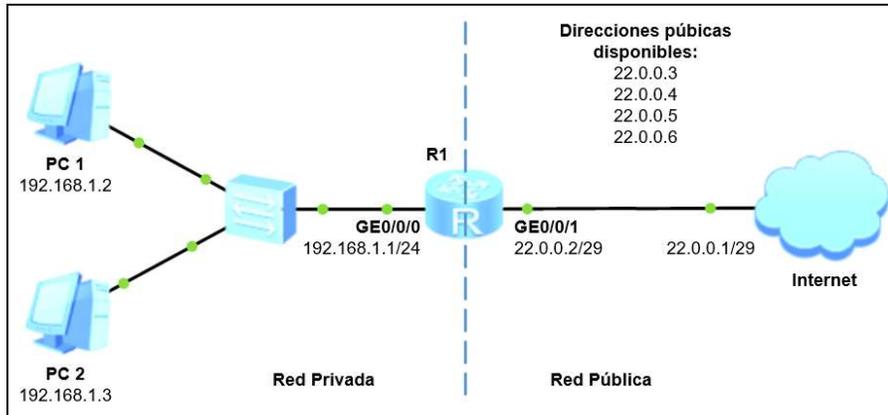
- NAT estático que asigna una dirección pública por cada dirección privada que se desee dar a conocer.
- NAT dinámico que requiere la creación de un listado o grupo de direcciones públicas que serán utilizadas aleatoriamente a medida que una dirección privada requiere salir hacia Internet, al igual que NAT estático se asigna una dirección pública por cada dirección privada.
- NAPT o Network Address and Port Translation, también conocido por otros fabricantes como PAT o NAT sobrecargado, permite que varias direcciones privadas utilicen una misma dirección pública para salir hacia Internet empleando los puertos virtuales.

En la actualidad la variante más utilizada es NAPT ya que permite que una gran cantidad de direcciones privadas tengan salida a internet empleando una dirección pública, no obstante, NAT estático también es usado cuando se requiere anunciar la dirección de algún servidor.

2.11.3.1. Implementando NAT estático

En la figura 167, se muestra la topología a emplear para implementar NAT estático, dinámico y NAPT.

Figura 167. Topología para implementar NAT



Fuente: elaboración propia, empleando eNSP.

Para implementar NAT estático se supondrá que se tiene contratado el segmento público 22.0.0.0/29 a un ISP. De esta forma es necesario asignar una dirección pública a las dos computadoras PC 1 y PC 2 para que tengan salida y sean alcanzables desde Internet. Una vez se tenga configurado el direccionamiento en el *router* se debe ingresar a la vista de la interfaz de salida, es decir la que cuenta con la dirección pública y emplear el comando: *nat static global ip_public inside ip_private*, donde el parámetro *ip_public* es una dirección pública disponible y *ip_private* es la dirección privada del *host*.

En la figura 168, se muestra la configuración aplicada a R1 para traducir la dirección privada 192.168.1.2 a la pública 22.0.0.3 y 192.168.1.3 a 22.0.0.4.

Figura 168. **Configurando NAT estático en R1**

```
[R1] interface G0/0/1
[R1-GigabitEthernet0/0/1] nat static global 22.0.0.3 inside 192.168.1.2
[R1-GigabitEthernet0/0/1] nat static global 22.0.0.4 inside 192.168.1.3
[R1-GigabitEthernet0/0/1] quit
[R1]
```

Fuente: Elaboración propia, empleando eNSP.

Para verificar la configuración de NAT estático se emplea el comando: *display nat static*, como se muestra en la figura 169.

Figura 169. **NAT estático en R1**

```
<R1> display nat static
Static Nat Information:
Interface : GigabitEthernet0/0/1
Global IP/Port : 22.0.0.3/----
Inside IP/Port : 192.168.1.2/----
Protocol : ----
VPN instance-name : ----
Acl number : ----
Netmask : 255.255.255.255
Description : ----

Global IP/Port : 22.0.0.4/----
Inside IP/Port : 192.168.1.3/----
Protocol : ----
VPN instance-name : ----
Acl number : ----
Netmask : 255.255.255.255
Description : ----

Total : 2
<R1>
```

Fuente: elaboración propia, empleando eNSP.

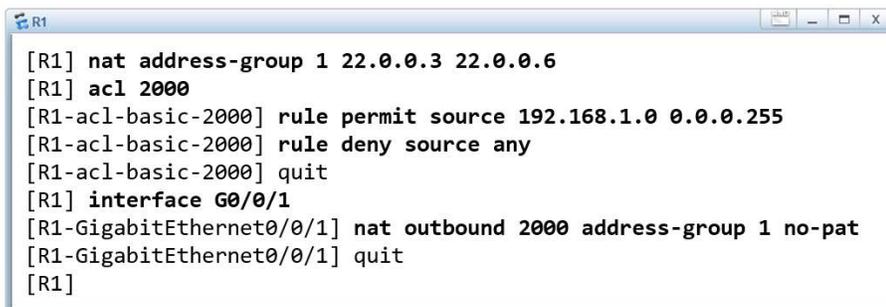
2.11.3.2. Implementando NAT dinámico

Para implementar NAT dinámico se debe crear una lista que agrupe todas las direcciones públicas disponibles, para esto se ingresa a la vista del sistema y

se emplea el comando: `nat address-group number_group start_ip_address end_ip_address`, donde el parámetro `number_group` es el número que identifica al grupo de 0 a 7, `start_ip_address` es la primera dirección pública disponible y `end_ip_address` la última dirección disponible. Seguidamente se debe crear una ACL que restringa las direcciones privadas que tendrán acceso a Internet. Finalmente se debe habilitar la interfaz de salida, es decir la que cuenta con la dirección pública configurada, para manejar NAT dinámico sin PAT, para esto se emplea el comando: `nat outbound acl_number address-group number_group no-pat`.

En la figura 170, se muestra la configuración de R1 para implementar NAT dinámico y dar acceso a internet al segmento 192.168.1.0/24 con cualquier dirección pública disponible.

Figura 170. **Configurando NAT dinámico en R1**



```
[R1] nat address-group 1 22.0.0.3 22.0.0.6
[R1] acl 2000
[R1-acl-basic-2000] rule permit source 192.168.1.0 0.0.0.255
[R1-acl-basic-2000] rule deny source any
[R1-acl-basic-2000] quit
[R1] interface G0/0/1
[R1-GigabitEthernet0/0/1] nat outbound 2000 address-group 1 no-pat
[R1-GigabitEthernet0/0/1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

Para verificar la configuración de NAT o NAT en el *router* se emplea el comando: `display nat outbound`, como se observa en la figura 171.

Figura 171. NAT Dinámico en R1

```
<R1> display nat outbound
NAT Outbound Information:
-----
Interface                Ac1      Address-group/IP/Interface  Type
-----
GigabitEthernet0/0/1     2000    1                            no-pat
-----
Total : 1
<R1>
```

Fuente: elaboración propia, empleando eNSP.

2.11.3.3. Implementando NAPT

Para implementar NAPT los pasos son exactamente iguales a la configuración de NAT dinámico, con la diferencia de que se debe eliminar el comando: *no-pat* de la interfaz de salida. En la figura 172 se muestra la configuración de NAPT en R1.

Figura 172. Configurando NAPT en R1

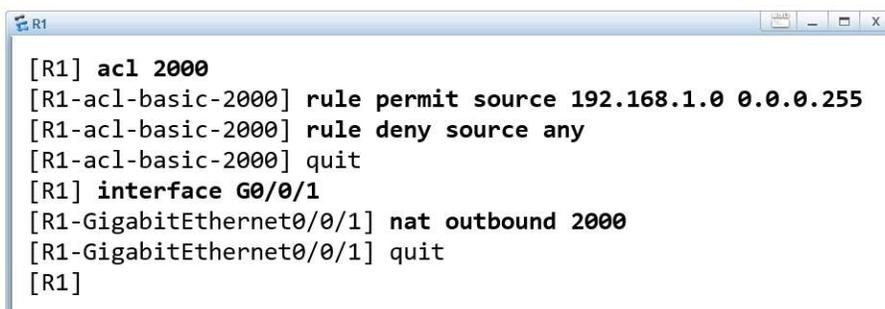
```
[R1] nat address-group 1 22.0.0.3 22.0.0.6
[R1] acl 2000
[R1-acl-basic-2000] rule permit source 192.168.1.0 0.0.0.255
[R1-acl-basic-2000] rule deny source any
[R1-acl-basic-2000] quit
[R1] interface G0/0/1
[R1-GigabitEthernet0/0/1] nat outbound 2000 address-group 1
[R1-GigabitEthernet0/0/1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

2.11.3.4. Easy-IP

Existe otra técnica propietaria de Huawei llamada Easy-IP que permite realizar NATP utilizando la dirección pública configurada en la interfaz de salida del *router* para traducir las direcciones privadas por medio de los puertos. Con esta técnica no es necesario crear un grupo de direcciones públicas, *address-group*, ya que se utilizará la dirección que está configurada en la interfaz. En la figura 173 se muestra la configuración de Easy-IP en R1 para dar acceso a internet al segmento 192.168.1.0/24.

Figura 173. Configurando Easy-IP en R1

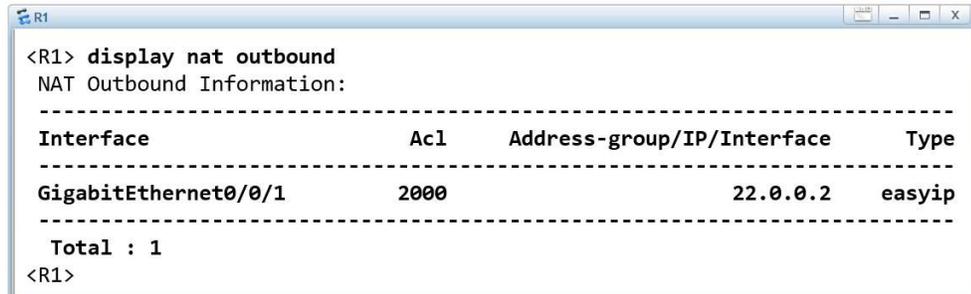


```
[R1] acl 2000
[R1-acl-basic-2000] rule permit source 192.168.1.0 0.0.0.255
[R1-acl-basic-2000] rule deny source any
[R1-acl-basic-2000] quit
[R1] interface G0/0/1
[R1-GigabitEthernet0/0/1] nat outbound 2000
[R1-GigabitEthernet0/0/1] quit
[R1]
```

Fuente: elaboración propia, empleando eNSP.

Para verificar la configuración de Easy-IP se puede emplear el comando: *display nat outbound*, como se muestra en la figura 174.

Figura 174. Easy-IP en R1



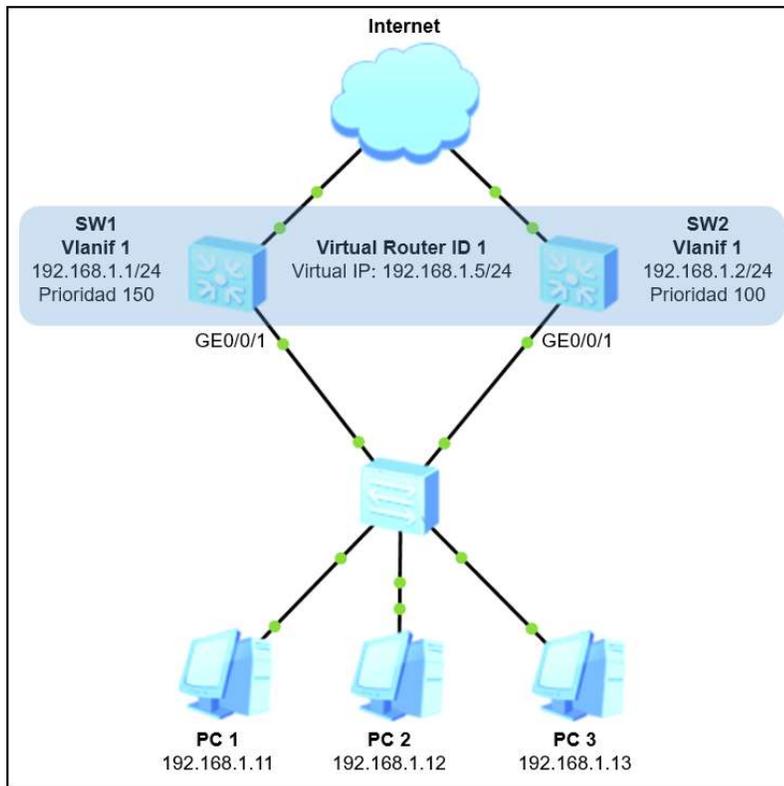
```
<R1> display nat outbound
NAT Outbound Information:
-----
Interface                Ac1      Address-group/IP/Interface  Type
-----
GigabitEthernet0/0/1     2000    22.0.0.2                    easyip
-----
Total : 1
<R1>
```

Fuente: elaboración propia, empleando eNSP.

2.11.4. VRRP

Virtual Router Redundancy Protocol es un protocolo abierto definido en el estándar RFC 3 768. Su propósito es brindar múltiples redundancias para el *default gateway* de una red L2. Para analizar su funcionamiento se toma como ejemplo la topología mostrada en la figura 175.

Figura 175. Topología para implementar VRRP



Fuente: elaboración propia, empleando eNSP.

Se requiere tener una alta disponibilidad para las computadoras en la red 192.168.1.0/24 por lo que se instalan dos *switches* multicapa para funcionar como *default gateway* de la red. Cada *switch* multicapa tiene configurada una dirección IP dentro del segmento 192.168.1.1 y 192.168.1.2, sin embargo, en los *hosts* no es posible configurar dos direcciones IP para el *default gateway*, es aquí donde VRRP entra en funcionamiento.

VRRP permite crear una dirección IP virtual para funcionar como dirección de *default gateway* para los *hosts*, así mismo se crea una dirección MAC virtual para completar el ARP. Esta dirección IP virtual es compartida por un grupo de *routers* identificados con un ID o Router Virtual ID, no obstante, solamente un

router funciona realmente como *gateway* a la vez, los demás serán de respaldo y entrarán a funcionar si el principal deja de estar disponible. Para esto, VRRP elige un *router* maestro en base a la prioridad entre todos los *routers* que conforman el grupo RVID. El parámetro de prioridad puede variar entre 1 y 254, donde el *router* que posea la prioridad de mayor valor será electo como maestro. La prioridad por defecto para todos los *routers* o *switches* multicapa Huawei es de 100.

Luego de activar VRRP en los *routers* y agregarlos al grupo RVID, empezarán a intercambiar mensajes Hello en formato Multicast que tendrán la función de detectar cuando uno de ellos deje de estar disponible. Para la topología mostrada en la figura 175, SW1 y SW2 intercambiaran mensajes Hello que salen de las interfaces GE0/0/1 y atraviesan el *switch*.

Debido a que SW1 tiene una prioridad de 150 será electo como maestro del grupo RVID por lo que físicamente todos los paquetes de los *hosts* viajarán hacia él para salir de la red. Si se dejan de recibir paquetes Hello del maestro entonces el *router* con la siguiente prioridad más alta tomará su lugar, que para este ejemplo es SW2, haciendo que todo el tráfico que busque salir de la red conmute hacia este. Por defecto los mensajes Hello son enviados cada segundo, sin embargo, este parámetro puede ser modificado a conveniencia.

Si SW1 vuelve a estar disponible y los *routers* del grupo RVID detectan los paquetes Hello entonces vuelve a tomar el papel del maestro del grupo y conmuta el tráfico nuevamente hacia él. Ahora bien, es posible configurar tiempo de espera antes que el *router* con mayor prioridad vuelva a tomar el papel del maestro, esta función es conveniente cuando se está trabajando con redes muy inestables. Por defecto este tiempo de espera es de cero segundos.

2.11.4.1. Implementando VRRP

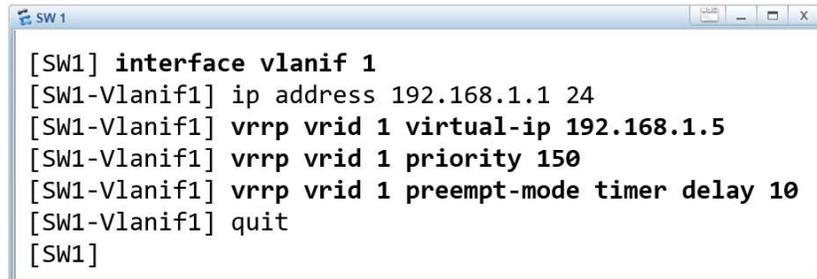
Para implementar VRRP se tomará como ejemplo la topología mostrada en la figura 175. En ella es necesario elegir a SW1 como maestro con una prioridad de 150 y programar un tiempo de espera de 10 segundos antes que vuelva a tomar el rol del maestro del grupo.

Para activar VRRP en un *router* se debe ingresar a la vista de la interfaz capa tres en la cual se desea activar el protocolo y emplear el comando: `vrrp vrid vrid_group virtual-ip ip_address`, donde el parámetro *vrid_group* es el número identificador del grupo VRID que puede variar entre 1 y 255 y el parámetro *ip_address* es dirección IP virtual elegida, que por obvias razones debe estar dentro del segmento de red configurada en la interfaz.

El siguiente paso es definir la prioridad, para esto se emplea el comando: `vrrp vrid vrid_group priority priority_value`, donde el parámetro *priority_value* es la prioridad. Finalmente se puede elegir el tiempo de espera que el *router* maestro esperará antes de anunciar su disponibilidad a los otros *routers* del grupo, para esto se emplea el comando: `vrrp vrid vrid_group preempt-mode timer delay time`, donde el parámetro *time* es el tiempo en segundos.

En la figura 176, se muestra la configuración de VRRP para SW1 estableciendo una prioridad de 150 y un *delay* de 10 segundos.

Figura 176. Configuración de SW1

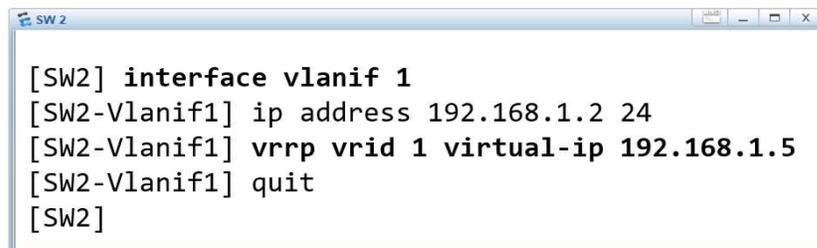


```
[SW1] interface vlanif 1
[SW1-Vlanif1] ip address 192.168.1.1 24
[SW1-Vlanif1] vrrp vrid 1 virtual-ip 192.168.1.5
[SW1-Vlanif1] vrrp vrid 1 priority 150
[SW1-Vlanif1] vrrp vrid 1 preempt-mode timer delay 10
[SW1-Vlanif1] quit
[SW1]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 177, se muestra la configuración de VRRP para SW2.

Figura 177. Configuración de SW2

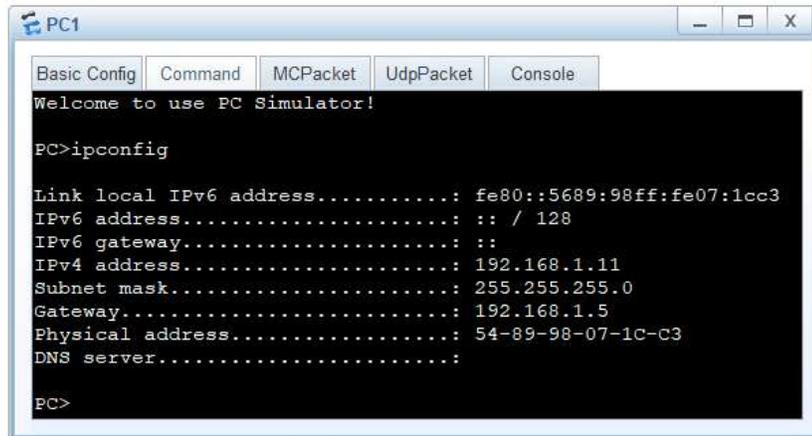


```
[SW2] interface vlanif 1
[SW2-Vlanif1] ip address 192.168.1.2 24
[SW2-Vlanif1] vrrp vrid 1 virtual-ip 192.168.1.5
[SW2-Vlanif1] quit
[SW2]
```

Fuente: elaboración propia, empleando eNSP.

En la figura 178, se muestra la configuración del *default gateway* en el *host* PC 1.

Figura 178. Configuración de PC 1



The image shows a window titled 'PC1' with a 'Console' tab selected. The console output displays the following configuration details:

```
PC>ipconfig
Link local IPv6 address.....: fe80::5689:98ff:fe07:1cc3
IPv6 address.....: :: / 128
IPv6 gateway.....: ::
IPv4 address.....: 192.168.1.11
Subnet mask.....: 255.255.255.0
Gateway.....: 192.168.1.5
Physical address.....: 54-89-98-07-1C-C3
DNS server.....:
```

Fuente: elaboración propia, empleando eNSP.

2.11.4.2. Verificando la configuración de VRRP

Los parámetros más importantes por revisar en la configuración VRRP son: el estado, que puede ser maestro o backup, la dirección IP virtual, la dirección IP del *router* maestro, la prioridad configurada, el período de envío de los paquetes Hello y el tiempo desde el último cambio de estado. Para visualizar esta información se emplea el comando: *display vrrp vrid_group*, como se observa en la figura 179.

Figura 179. Configuración VRRP de SW1

```
<SW1> display vrrp
Vlanif1 | Virtual Router 1 1
  State : Master 2
  Virtual IP : 192.168.1.5 3
  Master IP : 192.168.1.1 4
  PriorityRun : 150
  PriorityConfig : 150 5
  MasterPriority : 150
  Preempt : YES Delay Time : 10 s 6
  TimerRun : 1 s
  TimerConfig : 1 s 7
  Auth type : NONE
  Virtual MAC : 0000-5e00-0101
  Check TTL : YES
  Config type : normal-vrrp
  Create time : 2020-09-12 18:26:55 UTC-08:00
  Last change time : 2020-09-12 18:27:12 UTC-08:00 8
<SW1>
```

- | | |
|----------------------------|---|
| 1 Interfaz L3 Grupo VRID | 5 Prioridad configurada |
| 2 Estado del router | 6 <i>Timer</i> para Preempt |
| 3 Dirección IP virtual | 7 <i>Timer</i> para envío de <i>hello</i> |
| 4 Dirección IP del maestro | 8 Hora del último cambio de estado |

Fuente: elaboración propia, empleando eNSP.

2.11.5. SYSLOG

Syslog es una función que permite registrar todas las actividades que ocurren en el equipo y guardarlas de forma local en una memoria *buffer* o de forma remota en un servidor Syslog. Algunas de las actividades que se pueden registrar en un equipo son caídas de interfaz a nivel físico o de protocolo, establecimiento de protocolos de enrutamiento, cambios de estado VRRP, cambios de topología STP, entre otros.

Por defecto esta función viene activada en todos los *routers* y *switches* Huawei, sin embargo, es necesario indicar el lugar donde se guardarán los

registros. En general, para activar la función Syslog en un equipo se ingresa a la vista del sistema y se emplea el comando: *info-center enable*. Para guardar los registros de forma local en el *buffer* del equipo se agrega el comando: *info-center logbuffer*. Por defecto, la memoria buffer puede albergar hasta 512 registros, sin embargo, este valor puede aumentarse hasta 1 024 con el comando: *info-center logbuffer size 1024*.

Los registros o *logs* se categorizan en niveles de severidad, en la tabla XV se muestra los nombres y niveles que existen.

Tabla XV. **Niveles de severidad de los registros**

Nivel	Descripción
0	Emergencia: Indica que el sistema no está disponible.
1	Alerta: Indica que es requerido una intervención inmediata.
2	Crítico: Indica un importante mensaje.
3	Error: Indica un error general.
4	Advertencia o <i>Warning</i> : Indica una advertencia general.
5	Notificación: Indica una notificación general.
6	Información: Indica información general.
7	<i>Debugging</i> : indica mensajes <i>debugging</i> .

Fuente: elaboración propia.

Por defecto, los registros generados no son guardados con la hora en la cual ocurrió el evento, para activar esta función se usa el comando: *info-center timestamp log date*.

Los registros almacenados en la memoria buffer tiene una estructura específica que informa sobre la severidad, autor, descripción, fecha y hora en el cual ocurrió el evento. La estructura de los registros se describe en la figura 180.

Figura 180. **Estructura de los registros Syslog**

```
Tiempo Hostname %%ddMódulo/Severidad/Desc(1)[#]:Descripción
* Tiempo: Mes, día, año y hora
* Hostname: Nombre del equipo
* %: Indica que el log fue generado por un equipo Huawei
* dd: Versión
* Módulo: Nombre del módulo que generó el registro
* Severidad: Nivel de severidad [0-7]
* Desc: Breve descripción del registro
* (1): Registro informativo
* [#]: Número correlativo identificador del registro
* Descripción: Descripción detallada sobre el evento registrado
```

Fuente: elaboración propia.

Finalmente, para ver el historial de eventos registrados en el equipo se emplea el comando: *display logbuffer*. Es importante recordar que los registros se muestran en orden cronológico, siendo el primero el más reciente. En la figura 181, se puede ver el historial de eventos de un *router* Huawei.

Figura 181. Registros de un router Huawei

```
<Router> display logbuffer
Logging buffer configuration and contents: enabled 1
Allowed max buffer size: 1024
Actual buffer size: 512
Channel number: 4, Channel name: logbuffer
Dropped messages: 0
Overwritten messages: 0
Current messages: 8

Sep 14 2020 01:25:00-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[0]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=LoadingDone, NeighborPreviousState>Loading, NeighborCurrentState=Full)
Sep 14 2020 01:25:00-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[1]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=ExchangeDone, NeighborPreviousState=Exchange, NeighborCurrentState>Loading)
Sep 14 2020 01:25:00-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[2]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=NegotiationDone, NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)
Sep 14 2020 01:24:57-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[3]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=AdjOk?, NeighborPreviousState=2Way, NeighborCurrentState=ExStart)
Sep 14 2020 01:24:33-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[4]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=2WayReceived, NeighborPreviousState=Init, NeighborCurrentState=2Way)
Sep 14 2020 01:24:14-08:00 R3 %%01OSPF/4/NBR_CHANGE_E(1)[5]:Neighbor changes event: neighbor status changed. (ProcessId=256, NeighborAddress=2.3.0.10, NeighborEvent=HelloReceived, NeighborPreviousState=Down, NeighborCurrentState=Init)
Sep 14 2020 01:24:10-08:00 R3 %%01IFNET/4/LINK_STATE(1)[6]:The line protocol IP on the interface GigabitEthernet0/0/1 has entered the UP state.
Sep 14 2020 01:24:00-08:00 Huawei %%01IFPDT/4/IF_STATE(1)[7]:Interface GigabitEthernet0/0/1 has turned into UP state.
```

- 1 Logging Buffer activado
- 2 Tamaño del Buffer
- 3 Cantidad de registros en el Buffer
- 4 Registro anunciando la creación de adyacencia OSPF
- 5 Registro del estado UP de una interfaz a nivel de protocolo
- 6 Registro del estado UP de una interfaz a nivel físico

Fuente: elaboración propia, empleando eNSP.

Para limpiar la memoria *buffer* del equipo se emplea el comando: *reset logbuffer*.

CONCLUSIONES

1. Los protocolos de comunicación más utilizados en la pila TCP/IP que trabaja en conjunto con Ethernet, son: ARP, RIP, OSPF, STP, DHCP, NAT Y VRRP, y su comprensión teórica es indispensable al momento de su implementación.
2. El simulador eNSP es la herramienta recomendada para la configuración de dispositivos Huawei, ya que permite virtualizar el sistema operativo VRP de *routers*, *switches* y *switches* multicapa, haciendo que los comandos de configuración sean iguales a los empleados en dispositivos reales.
3. La herramienta de simulación eNSP permite crear topologías de red complejas, gracias a que ejecuta sistemas operativos independientes para cada equipo, con esto se logra implementar protocolos de comunicación avanzados como: OSPF, STP, VRRP, enrutamiento Inter-VLAN, entre otros.

RECOMENDACIONES

1. Continuar el estudio en el campo de las telecomunicaciones con cualquier tecnología ofrecida en el mercado, tomando como base las certificaciones correspondientes, ya que su contenido se mantiene en constante actualización por parte de los fabricantes.
2. Complementar el estudio teórico con la práctica a nivel de laboratorio. Para simular las topologías de red se recomienda la herramienta eNSP, pero existen otras opciones como GNS3 o EVE que también permiten virtualizar sistemas operativos y crear topologías complejas.
3. Complementar el estudio de telecomunicaciones con la seguridad de la información. Para esto existen múltiples compañías que ofrecen estos servicios, siendo Fortinet una de las más utilizadas en Guatemala. Es por esta razón que se aconseja el estudio a profundidad del contenido de las certificaciones NSE.

BIBLIOGRAFÍA

1. FAIRHURST, Gorry. *Encapsulation of Protocol Data Units*. [en línea]. <<https://erg.abdn.ac.uk/users/gorry/course/intro-pages/encapsulation.html>>. [Consulta: 15 de julio de 2020].
2. IEEE. *IEEE 802.2-1985 – IEEE Standard for Local Area Network*. [en línea]. <https://standards.ieee.org/standard/802_2-1985.html>. [Consulta: 20 de diciembre de 2020].
3. JIANG, Yonghong. *HCNA Networking Study Guide*. China: Springer, 2016. 342 p.
4. KOZIEROK, Charles. *ARP Message Format*. [en línea]. <http://www.tcpipguide.com/free/t_ARPMessageFormat.htm>. [Consulta: 17 de julio de 2020].
5. LAMMLE, Todd. *CCNA Routing and Switching Study Guide*. Estados Unidos: Sybex, 2013. 1100 p.
6. PROMAX. *Tipos de conectores de fibra óptica*. [en línea]. <<https://www.promax.es/esp/noticias/578/tipos-de-conectores-de-fibra-optica-guia-sencilla/>>. [Consulta: 15 de octubre de 2020].

