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

PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Kenie Gary Chuy Azurdia
Asesorado por el Ing. Byron Odilio Arrivillaga Méndez

Guatemala, septiembre de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

KENIE GARY CHUY AZURDIA
ASESORADO POR EL ING. BYRON ODILIO ARRIVILLAGA MÉNDEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, SEPTIEMMBRE DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. 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. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
EXAMINADOR	Ing. José Antonio de León Escobar
EXAMINADORA	Inga. María Magdalena Puente Romero
SECRETARIA	Inga. Lesbia Magalí Herrera 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:

PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 11 de agosto de 2017.

Kenie Gary Chuy Azurdia

Guatemala 14 de noviembre del 2018

Ingeniero
Julio Cesar Solares Peñate
Coordinador
Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero Solares:

Por este medio hago de su conocimiento que he concluido la revisión del trabajo de graduación del estudiante Kenie Gary Chuy Azurdia, titulado:

PROPUESTA PARA LA CREACION Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECANICA ELECTRICA FACULTAD DE INGENIERIA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.

El cual cumple con los objetivos que se propusieron pasa su elaboración. Por lo que, el estudiante Kenie Gary Chuy Azurdia puede continuar con el trámite que la Universidad tiene para concluir su proceso de graduación.

Hago la salvedad que, tanto el señor Chuy con el suscrito en calidad de Asesor nombrado, somos responsables del contenido del trabajo de graduación referido

Reciba un cordial saludo,

Ing. Byron Odilio Arrivillaga Méndez

Byron Arrivillaga Méndez
Ingeniero Electrónico
Colegiado 5217

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

REF. EIME 02. 2019.
15 DE ENERO 2019.

Señor Director
Ing. Otto Fernando Andrino González
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me **permito dar aprobación al trabajo de Graduación titulado: PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, del estudiante; Kenie Gary Chuy Azurdia, que cumple con los requisitos establecidos para tal fin.**

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Julio César Solares Peñate
Coordinador de Electrónica

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

REF. EIME 02. 2019.

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: KENIE GARY CHUY AZURDIA Titulado: PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, procede a la autorización del mismo.

Ing. Otto Fernando Andrino González

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

GUATEMALA, 5 DE FEBRERO 2019.

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

DTG. 324.2019

El Decano 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: **PROPUESTA PARA LA CREACIÓN Y DISEÑO DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES II DE LA ESCUELA DE MECÁNICA ELÉCTRICA FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,** presentado por el estudiante universitario: **Kenie Gary Chuy Azurdia,** 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, septiembre de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

/gdech

ACTO QUE DEDICO A:

Dios	Por todas las bendiciones y permitirme culminar mis estudios de licenciatura.
Mis padres	Mario Chuy y Mayra Azurdia de Chuy, por todo su apoyo y comprensión en los momentos que más lo necesitaba, por nunca dejarme sola y sobre todo por amarme tal como soy.
Mis hermanos	Por estar a mi lado observando mi proceso de formación académica profesional.
Mi novia	Ana Abril, por creer en mí, por apoyarme cuando más lo necesitaba, por darme todo ese amor que me motiva a ser mejor persona.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi segundo hogar, en donde me formé como profesional.
Facultad de Ingeniería	Por darme las herramientas necesarias para aportar de manera positiva al país.
Mis padres	Roberto López y Aracely Ortega de López, por todas sus enseñanzas a lo largo de mi vida, sobre todo amor.
Mi novia	Ana Abril, por ser mi incondicional y por preocuparse de que siempre esté bien.
Mis amigos de la carrera	Por todas las convivencias, apoyo y motivaciones para cumplir nuestra meta en común.
Mi asesor	Byron Arrivillaga, por tener siempre la mejor disposición en ayudarme y ser una fuente de motivación para terminar mi trabajo de graduación

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES XI
LISTA DE SÍMBOLOS XXI
GLOSARIO XXIII
RESUMEN XXVII
OBJETIVOS XXXI
INTRODUCCIÓN XXXIII

1. GENERALIDADES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA 1
    1.1. Historia 1
    1.2. Misión 2
    1.3. Visión 2
    1.4. Facultad de Ingeniería 2
        1.4.1. Historia 2
        1.4.2. Misión 3
        1.4.3. Visión 3
        1.4.4. Escuela de Mecánica Eléctrica 3
            1.4.4.1. Misión 4
            1.4.4.2. Visión 4

2. DISTRIBUCIÓN DEL LABORATORIO PROPUESTO 5
    2.1. Misión 5
    2.2. Visión 5
    2.3. Objetivos 5
        2.3.1. Objetivo general 5

2.3.2. Objetivos específicos 6
2.4. Perfil del auxiliar de laboratorio 6
2.5. Responsabilidades del auxiliar del laboratorio 7
2.6. Metodología 8

3. MARCO TEÓRICO 9
3.1. BGP 9
3.1.1. Asignación de direcciones IP públicas 9
3.1.2. Fundamentos 13
3.1.3. Opciones *multihoming* de BGP 17
3.1.4. Opción 1: rutas por defecto por todos los proveedores 18
3.1.5. Opción 2: rutas por defecto y actualizaciones parciales 20
3.1.6. Opción 3: rutas completas de todos los proveedores 22
3.1.7. Enrutamiento BGP entre sistemas autónomos 24
3.1.8. Funcionalidad vector distancia 26
3.1.9. Políticas de ruteo BGP 28
3.1.10. Características de BGP 30
3.1.11. Base de datos BGP 34
3.2. IBGP y EBGP 36
3.2.1. Relaciones con los vecinos BGP 36
3.2.2. Establecer una conexión entre vecinos externos BGP 37
3.2.3. Estableciendo una conexión entre los vecinos internos de BGP 38
3.2.4. Sincronización dentro de un sistema autónomo 40
3.2.5. IBGP en un sistema no transitorio 42

3.2.6. Problemas de enrutamiento en un sistema autónomo de tránsito 46
3.3. Configuración BGP 48
3.3.1. Configuración básica BGP 48
3.3.2. Activación de una sesión BGP 49
3.3.3. Apagando un vecino BGP 52
3.3.4. Consideraciones de configuración de BGP 53
3.3.5. Problema de emparejamiento IBGP 54
3.3.6. Comando *neighbor update-source* en BGP 55
3.3.7. Problemas de emparejamiento EBGP 58
3.3.8. Comportamiento del siguiente salto 61
3.3.9. Inyección de rutas BGP 64
3.3.10. Comando *network* en BGP 66
3.4. Seleccionando el camino BGP 68
3.4.1. Características y atributos de BGP 68
3.4.2. Atributos BGP 71
3.4.3. Atributo de ruta de un sistema autónomo 72
3.4.4. Atributo del siguiente salto 73
3.4.5. Atributo de origen 75
3.4.6. Atributo de preferencia local 76
3.4.7. Atributo *Multi Exit Discriminator* (*MED*) 78
3.4.8. Atributo de peso 79
3.4.9. Determinando la selección de ruta BGP 81
3.4.10. Selección de ruta con conexión *multihomed* 84
3.5. MPLS 87
3.5.1. Introducción a MPLS (*multiprotocol label switching*) 87
3.5.1.1. Beneficios de MPLS 88
3.5.2. Arquitectura MPLS: bloques de construcción 91

3.5.3. MPLS capa 3 VPN 93
3.5.4. Capa 3 IP/MPLS VPN 94
3.5.5. Topologías y aprovisionamiento de servicios IP/MPLS VPN 94
3.5.6. IP/MPLS VPN: una fundación para servicios de red 96
3.5.7. Transparencia IP/MPLS VPN 96
3.5.8. IP/MPLS VPN administración de red SLA 97
3.5.9. Arquitectura MPLS 98
3.5.10. Introducción a etiquetas MPLS 99
3.5.11. Apilamiento de etiqueta 100
3.5.12. Codificación de MPLS 101
3.5.13. MPLS y el modelo de referencia OSI 103
3.5.14. *Label switch router* 104
3.5.15. *Label switched* PATH 105
3.5.16. Clase de equivalencia de reenvío 107
3.5.17. Distribución de etiquetas 110
3.5.18. Agarre de etiquetas en un protocolo de enrutamiento IP existente 111
3.5.19. Ejecución de un protocolo separado para la distribución de etiquetas 112
3.5.20. Distribución de etiquetas con LDP 113
3.5.21. Base de instancia de reenvío de etiquetas 115
3.5.22. Carga útil de MPLS 116
3.5.23. Espacios de etiquetas MPLS 117
3.5.24. Diferentes modos MPLS 119
3.5.25. Modo de distribución de etiquetas 119
3.5.26. Modos de retención de etiquetas 120
3.5.27. Modos de control LSP 121

3.5.28. Reenvío de paquetes etiquetados 122
3.5.29. Operación de etiquetas..... 123
3.5.30. Paquetes etiquetados de balanceo de cargas..... 130
3.5.31. Etiqueta desconocida 132
3.5.32. Etiquetas reservadas 133
3.5.33. Etiqueta NULL implícita 133
3.5.34. Etiqueta de alerta del *router*..... 136
3.5.35. Etiquetas sin reserva 137
3.5.36. Comportamiento TTL de paquetes etiquetados.... 138
3.5.37. Comportamiento TTL en el caso de IP a la etiqueta o etiqueta a la IP 139
3.5.38. Comportamiento TTL en el caso de etiqueta a etiqueta..... 140
3.5.39. Expiración del TTL..... 141
3.5.40. MPLS MTU 144
3.5.41. Comando MPLS MTU..... 145
3.5.42. Unidad de recepción máxima MPLS..... 146
3.5.43. Fragmentación de paquetes MPLS 148
3.5.44. Path MTU Discovery..... 149
3.5.45. *Label distribution protocol*..... 150
3.5.46. Descripción del LDP 151
3.5.47. El descubrimiento de LSR que están ejecutando LDP..... 153
3.5.48. Establecimiento y mantenimiento de sesión LDP . 158
3.5.49. Número de sesiones LDP..... 163
3.5.50. Publicidad de asignaciones de etiquetas..... 165
3.5.51. Retiro de etiqueta 171
3.5.52. La limpieza por medio de la notificación 173
3.5.53. Sesión dirigida LDP 174

3.5.54. Autenticación LDP 178
3.5.55. Control de anuncio de etiquetas a través de LDP . 179
3.5.56. Filtrado de enlace de etiqueta de entrada MPLS LDP 184
3.5.57. Autoconfiguración LDP 186
3.5.58. Sincronización MPLS LDP-IGP 188
3.5.59. Funcionamiento de la sincronización MPLS LDP-IGP 190
3.5.60. Configuración de sincronización MPLS LDP-IGP .. 191
3.5.61. Protección de sesión MPLS LDP 197
3.5.62. MPLS VPN 201
3.5.63. Definición de una VPN 201
3.5.64. Modelo MPLS VPN 202
3.5.65. Descripción arquitectónica de MPLS VPN 205
3.5.66. *Virtual routing forwarding* 205
3.5.67. *Route distinguisher* (*RD*) 209
3.5.68. *Route Targets* RTs 211
3.5.69. Propagación de una ruta VPNv4 en la red MPLS VPN 217
3.5.70. Reenvío de paquetes en una red MPLS VPN 220
3.5.71. BGP 222
3.5.72. Extensiones y capacidades multiprotocolo BGP ... 223
3.5.73. Comunidad extendida BGP: RT 227
3.5.74. Rutas VPNv4 228
3.5.75. BGP llevando la etiqueta 229
3.5.76. RR'S 233
3.5.77. GRUPO RR 235
3.5.78. Ruta de selección BGP 237
3.5.79. Usando múltiples RD 238

3.5.80. Reenvío de paquetes 240
3.5.81. Protocolos de enrutamiento PE-CE 245
3.5.82. Rutas conectadas 245
3.5.83. Enrutamiento estático 246
3.5.84. RIP versión 2 247
3.5.85. OSPF 249
3.5.86. Configuración OSPF VRF 251
3.5.87. Propagación de métrica OSPF 253
3.5.88. Comunidades extendidas BGP para OSPF 254
3.5.89. Diseño de red OSPF MPLS VPN 256
3.5.90. Enlace simulado 257
3.5.91. *Bit Down* y etiqueta de dominio 261
3.5.92. EIGRP MPLS 263
3.5.93. Configuración 267
3.5.94. POI *pre-bestpath* 268
3.5.95. EIGRP pe-ce con enlaces *backdoor* 270
3.5.96. eBGP 272
3.5.97. Anulación del sistema autónomo 274
3.5.98. *Allowas-in* 275
3.5.99. *Hub-and-spoke* 277
3.5.100. SOO 279
3.5.101. Acceso VRF 282
3.5.102. Acceso a internet 283
3.5.103. Internet en una VPN 284
3.5.104. Acceso a internet a través de la tabla de enrutamiento global 284
3.5.105. Acceso a internet a través de la tabla de enrutamiento global con rutas estáticas 286

3.5.106. Acceso a internet a través de un sitio central VRF 288
3.5.107. Multi-VRF CE 289
3.5.108. Gestión CE 291
3.6. VPN 294
3.6.1. Fundamentos 295
3.6.2. Arquitectura VPN de superposición y de punto a punto 297
3.6.3. VPN superpuestas 297
3.6.4. EL modelo de superposición incluye VPN L2 y L3 297
3.6.5. VPN con aprovisionamiento del proveedor de servicio 299
3.6.6. Topologías de VPN 300
3.6.6.1. VPN de acceso remoto 301
3.6.6.2. VPN de intranet de punto a punto 302
3.6.6.3. Características de una VPN segura ... 304
3.6.7. Seguridad de VPN: encapsulación 306
3.6.8. Cifrado asimétrico 309
3.6.9. Algoritmos de cifrado simétrico 311
3.6.10. Cifrado simétrico: DES 313
3.6.11. Cifrado simétrico: 3DES 315
3.6.12. Cifrado simétrico: AES 316
3.6.13. Intercambio de llaves *Diffie-Hellman* 317
3.6.14. Ejemplo clásico de *Diffie-Hellman*: Alice y Bob 319
3.6.15. Primeros números y aritmética modular 321
3.6.16. Seguridad VPN: IPSEC Y GRE 323
3.6.17. Túneles VPN de punto a punto 324
3.6.18. Túneles: acceso remoto 325

3.6.19. Características de seguridad de IPSEC 326
3.6.20. Protocolos y encabezados IPsec 329
3.6.21. Intercambio de claves de internet 332
3.6.22. Fases y modos IKE 335
3.6.23. Otras funciones IKE 339
3.6.24. *Dead Peer Protection* (*DPD*) y Cisco IOS *Keepalives* 339
3.6.25. Autenticación extendida 345
3.6.26. Protocolos de ESP y AH, transporte y modos de túnel 346
3.6.27. Encabezados ESP y AH 348
3.6.28. Autenticación e integridad AH 351
3.6.29. Protocolo ESP 352
3.6.30. Autenticación de mensajes y verificación de integridad 355
3.6.31. Entorno PKI 357
3.6.32. Autoridad certificadora 359
3.6.33. PKI jerárquica: múltiples CA 359
3.6.34. Certificado X.509 v3 361
3.6.35. Intercambio de mensajes PKI 363
3.6.36. Credenciales de PKI 365
3.7. IPv6 366
3.7.1. Multicast 386
3.7.2. Tunelización Q-IN-Q 388
3.7.3. Red metro 392
3.7.4. Ethernet en metro 398
3.7.5. Una vista de datos de la red metro 398

CONCLUSIONES 403

RECOMENDACIONES 405
BIBLIOGRAFÍA.....407

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Vista conceptual de asignación de direcciones públicas IPv4 11
2. Sistemas autónomos 15
3. Utilización BGP para conectarse a internet 16
4. Rutas por defecto para todos los proveedores 20
5. Por defecto de todos los proveedores y tabla parcial 22
6. Rutas completas de todos los proveedores 23
7. BGP ruta vector enrutamiento 26
8. BGP evita los bucles de enrutamiento 27
9. Políticas de enrutamiento BGP 29
10. *Peers* = *Neighbors* (compañeros = vecinos) 37
11. BGP externo 38
12. BGP interno 39
13. IBGP en un sistema autónomo de tránsito (proveedor de servicios de Internet) 41
14. Malla parcial BGP 44
15. Malla completa BGP 45
16. Problemas de enrutamiento si BGP no está activado en todos los routers en la ruta de transito 47
17. Comandos BGP 49
18. Parametros BGP 50
19. *Neighbor remote-as* commando y parametros BGP 51
20. Ejemplo del comando BGP neighbor 52
21. Problema de peering IBGP 54

22. Comando BGP *neighbor update-source* 55
23. BGP utilizando direcciones de loopback 57
24. Comando y parámetros de neighbor ebgp-multihop en BGP 59
25. Ejemplo del comando EBGP-multihop 60
26. Ejemplo del comportamiento del siguiente salto 63
27. Comando y parámetros de *network* en BGP 65
28. Ejemplo del comando *network* en BGP 66
29. Ejemplo del comando *network* (contenido) en BGP 67
30. Atributo de ruta de un sistema autónomo 73
31. Atributo del siguiente salto 74
32. Ejemplo del atributo de origen 75
33. Atributo de preferencia local 77
34. Atributo *MED* 79
35. Atributo de peso (solo equipos Cisco) 80
36. Sintaxis de una etiqueta MPLS 99
37. Pila de etiquetas 101
38. Encapsulación para paquetes de etiqueta 102
39. Valores de identificador de protocolo MPLS para los tipos de encapsulación de capa 2 103
40. Un LSP a través de una red MPLS 106
41. LSP anidado 107
42. Una red MPLS que ejecuta iBGP 109
43. Una red IPv4 sobre MPLS que ejecuta LDP 114
44. Red IPv4 sobre MPLS con LDP: conmutación de paquetes 115
45. Espacio de etiqueta por interfaz 118
46. Espacio de etiquetas por plataforma 119
47. Operación en etiquetas 123
48. Búsqueda en CEF o LFIB 124
49. Ejemplo de una entrada en la tabla CEF 125

50. Extracto de la LFIB 126
51. Ejemplo del comando MPLS *forwarding-table* (detallado) 127
52. Ejemplo de una entrada en la LFIB para un prefijo VPN de MPLS 128
53. Ejemplo de una tabla de adyacencia 129
54. Ejemplo de paquetes etiquetados con balanceo de cargas 130
55. Cambiando un camino a un estado sin etiqueta 132
56. Penúltimo salto de popping 134
57. Depuración que muestra la etiqueta 1 en un paquete MPLS 137
58. Cambio de rango de etiquetas MPLS 138
59. Comportamiento de propagación de TTL entre encabezado IP y etiquetas MPLS 140
60. Propagacion TTL en la operación de etiqueta a etiqueta en el caso de una operación de intercambio, push y pop 141
61. ICMP, tiempo excedido devuelto por un enrutador en una red IP 142
62. ICMP, tiempo excedido enviado por un router en una red MPLS 143
63. Cambio de MPLS MTU 146
64. Permitiendo tramas Jumbo en *switches* Ethernet 148
65. Red ejemplificada a nivel mundial MPLS 151
66. Configuración básica de MPLS LDP 153
67. Comando LDP Discovery 154
68. Comando show MPLS interfaces 155
69. Problema 'no route' 158
70. Tiempo de espera del vecino LDP e intervalo KA 160
71. Comando show MPLS LDP parameters 161
72. Cambio de la dirección de transporte LDP predeterminada 162
73. Cambio de la dirección de transporte LDP predeterminada a nivel de consola 162
74. Ejemplos del número de sesiones LDP entre un par de LSR 164
75. Direcciones IP enlazadas LDP 166

76. Ejemplo de un LIB, 1 167
77. Ejemplo de un LIB, 2 168
78. Relación entre direcciones enlazadas, RIB,LIB y LFIB 169
79. Utilizando No LDP Split Horizon 170
80. Fijado por 10.200.254.2/32 de la figura 79 170
81. Etiqueta retirada 172
82. Etiqueta implícita de retiro 173
83. Hello target aceptado en la red 176
84. Configuracion de Sydney para LDP dirigido (corregir) 177
85. Configuración de Miami para LDP dirigido 177
86. Sesión LDP dirigida en router Miami 178
87. Anuncio controlado LDP 181
88. Anuncio controlado LDP: configuración (arreglar Yakarta) 181
89. Anuncio controlado LDP 182
90. Listados de LSR Lisboa para el vecino 10.200.254.4 183
91. LFIB en LSR Lisboa 183
92. Anuncio controlado LDP 184
93. Ejemplo de filtrado de enlace de etiquetas entrantes LDP 186
94. Ejemplo de configuración de autoconfiguración LDP 187
95. Sesión de LDP entre LSR 189
96. Ejemplo de configuración de la sincronización de MPLS LDP-IGP 193
97. Sincronización MPLS LDP-IGP 194
98. Ejemplo de sincronización MPLS LDP-IGP con temporizador de espera 194
99. Sincronización MPLS LDP-IGP: métrica máxima de publicidad 195
100. Información de depuración MPLS LDP-IGP 196
101. Peer no alcanzable 197
102. Sesión de protección LDP 199
103. Ejemplo de sesión de protección LDP 200

104. Vista esquemática de una red MPLS VPN 202
105. Modelo MPLS VPN 205
106. VRF’s en un router PE 206
107. Configurando VRF 208
108. Configurando un RD 210
109. Ilustración de RTs 212
110. Configuración de RTs 212
111. Configuración VRF 213
112. Ejemplo de una extranet 214
113. Ejemplo de una extranet con RTs 215
114. Configurando RTs para una extranet 216
115. Ruta extranet 216
116. Propagación de ruta en una red VPN MPLS 218
117. Propagación de ruta paso a paso en una red VPN MPLS 219
118. Reenvío de paquetes en una red VPN MPLS 222
119. Intercambio de capacidades BGP 224
120. Configurando las familias de direcciones BGP 226
121. Atributo BGP RT 228
122. Rutas VPNv4 229
123. Capacidad de anuncio de etiqueta BGP 230
124. Configuración de VPNv4 de la familia de direcciones BGP 232
125. Actualizaciones Unicast para el comando *debug* ip bgp vpnv4 232
126. Publicidad BGP y etiquetas MPLS 233
127. Ruta vpnv4 rechazada 235
128. Ejemplo de una red MPLS VPN con grupos RR 236
129. Ejemplo de grupos RR 237
130. RR anuncia solo la mejor ruta BGP 239
131. Uso de multiples RD’s 241

132. Vida de un paquete IPv4 a través de una red *backbone* VPN de MPLS: reenvio de paquetes....242
133. VRF CEF cust one en ingreso PE....244
134. Ruta VPNv4 en ingreso PE....244
135. Entrada de LFIB en ingreso PE....244
136. Redistribución de rutas conectadas en BGP....246
137. Configuración VRF OSPF....246
138. Distribución de rutas estáticas en BGP....247
139. Configuración RIPv2 VRF....248
140. Rutas OSPF internas a través de la red *backbone* MPLS VPN....250
141. Posibles escenarios de VPN MPLS de OSPF....251
142. Configuración básica de OSPF VRF....252
143. Comando *show* IP OSPF....253
144. Comunidad extendida BGP para OSPF....256
145. Ejemplo de un enlace simulado....259
146. Enlace simulado OSPF....260
147. *Down Bit*....262
148. Etiqueta de dominio....263
149. Comunidades extendidas BGP para EIGRP....265
150. Propagación de una ruta EIGRP a través de la red troncal MPLS VPN....266
151. Ejemplo de configuración EIGRP VRF....267
152. Comunidad de costos para EIGRP sobre MPLS VPN....269
153. Enlace *Backdoor* entre sitios EIGRP....271
154. Set de SOO para una ruta EIGRP....272
155. Configuración básica de BGP como protocolo de enrutamiento PE-CE....273
156. Uso de AS-Override....274
157. *Hub-and-spoke* con BGP como protocolo de enrutamiento PE-CE....276

158. Escenario *hub-and-spoke* 278
159. SOO previniendo bucles de enrutamiento 280
160. Configuración SOO de mapas de ruta 281
161. Aplicando mapas de ruta SOO para BGP 281
162. Aplicando mapas de ruta SOO sobre las interfaces VRF 282
163. Aplicando mapas de ruta SOO en rutas estáticas 282
164. Comandos VRF *ping*, *traceroute* y *telnet* 283
165. Configuración del túnel GRE en el espacio de enrutamiento global en el PE 286
166. Acceso a internet a través de la tabla de enrutamiento global con rutas estáticas 288
167. Acceso a internet a través de un sitio VRF central 289
168. Ejemplo de un CE Multi-VRF 290
169. Ejemplo de acceso de gestión 292
170. Configuración de un *router* PE que proporciona acceso de administración 293
171. Gestión de la configuración del *router* PE 293
172. Topología básica VPN 296
173. VPN de superposición basada en CPE 298
174. Modelo provisional VPN de un proveedor de servicios 300
175. Cliente inicializado en acceso remoto VPN 301
176. Intranet VPN de sitio a sitio 303
177. Extranet VPN de sitio a sitio 304
178. Seguridad VPN: encapsulación de paquetes 307
179. Seguridad VPN: ejemplo de encapsulación y proceso de túnel 308
180. Encriptación asimétrica: Deffie-Hellman y RSA 309
181. Encriptación simétrica: 3DES 312
182. Intercambio de llaves *Deffie-Hellman* 315
183. Seguridad VPN: acceso remoto VPN usando IPsec 318

184. Características de seguridad IPsec 323
185. Encabezados IPsec 327
186. IKE 331
187. Modos IKE 334
188. Operación IKE 337
189. El problema: IPsec: IPsec y NAT 338
190. La solución: IPsec NAT-T 340
191. Modo de configuración 341
192. Opción modo de configuración 344
193. Xauth 345
194. Encabezado ESP y AH 348
195. Encriptación ESP 350
196. ESP anidado en AH 350
197. Autenticación e integridad AH 351
198. Formato de trama AH en modo túnel 352
199. Protocolo ESP 353
200. Tunel ESP y modos de transporte 355
201. Mensaje de autenticación y chequeo de integridad usando hash 356
202. Funciones Hash utilizadas comúnmente 357
203. Topología de jerarquía de tres niveles 360
204. Certificado X.509 V3 362
205. Modelo de consumo de direcciones IPv4 por regiones a nivel mundial 367
206. Encabezado IPv4 370
207. Encabezado IPv6 370
208. Configuración IPv6 375
209. Composición IPv6 376
210. Tamaño de dirección IPv6 377
211. Ejemplo de dirección IPv6 378

212. Configuración IPv6 en una interfaz de Router 378
213. Colocación de dirección IPv6 con autocompletado 379
214. Utilización de NDP 380
215. Comando Unicast-Routing 380
216. Verificación de procesos IPv6 381
217. Anuncio de verificación IPv6 381
218. Autoconfiguración IPv6 381
219. Resultado de la autoconfiguración IPv6 382
220. Comando Show ipv6 en la verificación de direcciones 382
221. Verificación NDP 383
222. Proceso EUI-64 384
223. Habilitación del comando neighbor Discovery debugging 384
224. Comando Ping de verificación 384
225. Mensajes NS y NA 385
226. Comando show neighbors 385
227. Representación de una topología multicast 387
228. Representación de las distintas tramas en Q-in-Q 388
229. Representación del envío de etiquetas Q-in-Q 389
230. Representación de una topología de red metro 393
231. Representación de una red TDM 395
232. Comparación de red ethernet y TDM 397
233. Representación de una red metro operativa 399

TABLAS

I. Formato de tupla 225
II. Números de AFI y sus descripciones 225
III. Números SAFI y sus descripciones para la familia de direcciones IP 226
IV. Codificación del campo NLRI para MPLS VPN 231

V. Topología 264
VI. Características de una vpn segura 305
VII. Algoritmos de encriptación simétrica comunes y niveles de seguridad 313
VIII. Direcciones multicast utilizadas en IPv6 374
IX. Comparación de configuración IPv6 375

LISTA DE SÍMBOLOS

Símbolo	Significado
	Concentrador VPN
	Conexión serial
	*Firewall*
	Nube de internet o de datos
	*Router*
	Servidores de puerta de enlaces predeterminadas
	Switch multicapa
	Tunel VPN

GLOSARIO

Algoritmo Es una serie de pasos lógicos para llevar a cabo una tarea específica. Los algoritmos son independientes tanto en el lenguaje de programación en el que se expresan como la computadora que los ejecuta.

Bucle Este se define como *routing loop* también, y ocurre cuando los encaminadores o routers disponen de una información acerca de la red y en lugar de enviar el trafico a su destino, se pasan los paquetes entre ellos creyendo que el otro *router* sabra el camindo.

Cifrado Método por el cual permite aumentar la seguridad de un mensaje o de un archivo mediante la codificación de contenido, de manera que solo pueda leerlo la persona que cuente con la clave de cifrado adecuada para descodificarlo.

Encabezado shim Se utiliza par proporcionar el soporte MPLS. Esta configuración de muestra utiliza el encapsulado HDCL predeterminado en las interfaces del POS de Cisco.

Encapsulado Informacion que se envia a través de una red se denomina datos o paquetes de datos. A este proceso se le llama encapsulado.

Frame relay Es una técnica de comunicación mediante retransmisión de tramas para redes de circuito virtual, introducida por la ITU-T a partir de la recomendación I.122 de 1988. Consiste en una forma simplificada de tecnología de conmutación de paquetes que transmite una variedad de tamaños de tramas o marcos (*frames*) para datos, perfecto para la transmisión de grandes cantidades de datos.

Keepalive Este se refiere generalmente a las conexiones de comunicaciones en una red que no están terminadas pero que se mantienen hasta que el cliente o servidor interrumpe la conexión. La característica clave de mantener las *keep alive* es el envio de un mensaje sin contenido entre un servidor y un cliente.

Loopback Es una interfaz virtual. Estas direcciones redefinidas en los dispositivos, incluso con direcciones IP publicas, es una practica común en los routers y son usualmente utilizadas para probar la capacidad de la tarjeta interna si se están enviando datos del protocolo BGP.

Neighbor Este se refiere a un vecino entre un *router* y otro *router* en una red para la intercomunicación de una red.

Null La interfaz *null* no es una interfaz física; es una interfaz virtual y siempre esta activa. La intefaz nula

	nunca reenvía ni recibe tráfico, pero la ruta del paquete a la interfa nula se descarta.
Paquete ip	Corresponde a la capa de red del modelo OSI. Y contiene datagamas y contenido de información de datos para el viaje del paquete.
Protocolos	Es el termino que se emplea para denominar al conjunto de normas, reglas y pautas que sirven para guiar una conducta o acción. En este caso un conjunto de reglas para que un paquete de red siga reglas para llegar a su destino de una red estructurada.
Redundancia	Es una parte fundamental en una red. Ya que permite que una red sea tolerante a las fallas. Las topologías redundantes proporcionan protección contra el tiempo de inactividad, o no disponibilidad, en una red el tiempo de inactividad puede deberse a la falla de un solo enlace, puerto o dispositivo de red.
RFC	Sigla en inglés (*Request For Comments*) que significa solicitud de comentarios y consiste en un documento que puede ser escrito por cualquier persona y que contiene una propuesta para una nueva tecnología, información acerca del uso de tecnologías y/o recursos existentes, propuestas para mejoras de tecnologías, proyectos experimentales y demás.

Router	Conocido como enrutador. Se trata de un producto de hardware que permite interconectar compuradoras que funcionan en el marco de una red. La función del *router* es establecer que ruta destinara a cada paquete de datos dentro de una red informática.
Sistema autónomo	Conjunto de redes, o de routers, una única política de enrutamiento y que se ejecuta bajo la administracion común, utiliando habitualmente un único IGP. Para el mundo exterior, el sistema autónomo es visto como una única.
Topología	Es una representación lógica o ya sea física de una red en la cual se intercambia información. Esta se puede definir como un conjunto de nodos interconectados entre si para el manejo de paquetes de información a nivel lógico.
Tunelización	Se puede definir como un protocolo que encapsula en su datagrama otro paquete de datos completo que utiliza un protocolo de comunicaciones diferente. Esencialmente, se crea un túnel entre dos puntos de una red por el cual se puede transmitir de forma segura cualquier tipo de datos.

RESUMEN

En Telecomunicaciones y redes existen diversas tecnologías para lograr una conexión, esto se logra gracias a distintos protocolos de comunicación para el correcto funcionamiento de lo que conocemos como internet, este trabajo de graduación contiene lo mas utilizado por los proveedores de servicios y que se puede tomar en cuenta para lograr ampliar el contenido del laboratorio de electrónica de la facultad de ingeniería. Este esta conformado de cuatro capítulos en donde se explican las distintas tecnologías para comprensión del funcionamiento de internet.

El capitulo uno es la explicación de uno de los protocolos de enrutamiento mas utilizados a nivel mundial, y asi lograr una interconexión de todos los países sin interferir en el área en que se encuentra, BGP siendo uno de los protocolos mas utilizados se encuentra en estos instantes como dominante en la interconexión entre países, gracias a su característica fundamental que es el segmentado de red por medio de áreas para delimitar redes entre cada nación. Otras características por las cuales es utilizado, es la confiabilidad del protocolo como tal, para escoger la ruta asi como su utilización que se vulve sencilla.

En el capitulo dos podemos apreciar una tecnología lo cual facilita una conexión metropolitana entre sucursales de negocios dentro de un país, esto debido a que MPLS no se categoriza como un protocolo de enrutamiento, mas bien una técnica que mezcla dos capas del modelo OSI, lo cual se encuentra entre capa 2 y capa 3 el cual otro elemento que utilizan los proveedores de servicio para mantener un orden y una fácil distribución de enlaces de datos, cuenta con la característica mas importante que es la conexión a base de

etiquetas en sus paquetes de datos que viajan a travez de una conexión de internet para llegar a su destino.

El capitulo tres que es VPN es una conexión que permite intercomunicar de un punto hacia otro punto, sin lograr observar todas las conexiones de distintos protocolos de enrutamiento o algún proceso de interconexión, además que cuenta con una característica principal lo cual es una conexión cifrada lo cual no permite que cualquier otro pueda observar que tipo de datos pueden transitar en esta especie de túnel de conexión de punto a punto, es utilizado para empleados que pueden realizar su trabajo desde cualquier punto geografico, es una herramienta mas para que una empresa pueda obtener mas sucursales en cualquier parte del mundo.

En el Capitulo cuatro se puede observar que es un avance de la evolución de IPv4 a IPv6 el cual ya se encuentra aplicado en algunos proveedores de servicio el cual se lanzo en el año 2012, el cual comienza una era de nuevos horizontes en el direccionamiento IP a nivel mundial, siendo la función mas importante el anycast el cual convierte de una herramienta muy útil para este protocolo de direccionamiento IP, asi mismo se ven las redes metro el cual es una solución mas de un enlace punto a punto asistido de MPLS, en conjunto con las redes WAN lo cual es un gran apoyo para los proveedores de servicios, la tunelización Q-in-Q ofrece una conectividad a nivel de capa dos del modelo OSI, siendo esta capaz de trabajar en conjunto con MPLS, asi mismo ethernet metro presta una técnica a nivel de capa dos para la interconexión para los proveedores de servicios.

En este trabajo de graduación se exponen las distintas maneras de comunicación y funcionamiento del internet como una red mundial de interconectar personas en distintos dispositivos, pero a nivel de infraestructura

se detalla el funcionamiento total de este para llevar a cabo una red mundial de comunicación.

OBJETIVOS

General

Presentar una propuesta de diseño del laboratorio de telecomunicaciones y redes locales II para complementar el conocimiento adquirido en otros cursos de la carrera de Ingeniería Electrónica relacionados con las telecomunicaciones.

Específicos

1. Crear un curso acorde a las exigencias de la universidad, apegado a los reglamentos de evaluación de la Escuela de Mecánica Eléctrica de la Facultad de Ingeniería.

2. Presentar el perfil que debe poseer el auxiliar a cargo del laboratorio de telecomunicaciones y redes locales II, y las responsabilidades y la metodología.

3. Proporcionar al alumno un contenido disponible en español y totalmente ilustrado con ejemplos para contribuir al entendimiento pleno del curso.

INTRODUCCIÓN

El presente trabajo de graduación presenta los temas propuestos para ser impartidos en el laboratorio de telecomunicaciones y redes locales II en donde se abordarán tecnologías ampliamente utilizadas por los proveedores de servicio de internet tales como:

Border Gateway Protocol (BGP), uno de los protocolos más utilizados, provee enrutamiento entre sistemas autónomos, permite la interconexión de todos los proveedores a nivel mundial.

Multi Protocol Label Switching (MPLS), protocolo encargado en la estructuración de un sistema autónomo interior, utilizado por su confiabilidad y versatilidad para la implementación de un ruteo más confiable.

Virtual Private Network (VPN), protocolo utilizado como una interconexión en forma de túnel hacia otro punto geográfico. Este alcance tiene como fin crear una comunicación confiable, ya que este túnel posee un grado de seguridad muy alto el cual se traduce en encriptación de este mismo; contribuye a una total confianza en la comunicación de un punto hacia otro; estas encriptaciones se utilizan sin que el usuario final deba configurar en su computador, ya que se realizan en dispositivos especializados para esa tarea en específico.

En cuanto a otras tecnologías se observa que se tiene VRF Lite por el cual es una tecnología utilizada ampliamente por los proveedores de servicios para así complementar sus topologías de red a nivel de ruteo y no tener ningún inconveniente; Metro Ethernet es un modelo que se utiliza ampliamente, ya que

proporciona un ordenado crecimiento de las topologías y tener una ampliación de la misma con distintas tecnologías; Q in Q siendo un complemento de estas; así mismo, Multicast para la propagación de televisión por medio de IP; IPv6 lo cual se está adoptando actualmente para el cambio que se prevee en el futuro.

1. GENERALIDADES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

1.1. Historia

La universidad de San Carlos de Guatemala, también conocida por sus siglas como Usac, es la universidad más antigua que posee Guatemala; es la única universidad estatal en este país, fue fundada en 1676.

En la época de la colonia se establece bajo el nombre de Real y Pontifica Universidad de San Carlos de Borromeo, y desde esa época sufre cinco cambios por las distintas coyunturas sociales y políticas que ocurren en el país, hasta adoptar en 1944 tras la revolución que se lleva a cabo en Guatemala, el nombre que hoy en día conocemos como Universidad de San Carlos de Guatemala.

En el transcurso de su existencia se han dado distintos acontecimientos importantes, desde la elección del primer claustro de catedráticos acontecimiento que se dio entre 1677 y 1678 el cual se da la oposición en Puebla México; este se vio en polémica ya que se asignan catedráticos interinos; inicia así oficialmente las clases en 1680, hasta su separación de la corona española en 1821 junto con la independencia de Guatemala; en el transcurrir del tiempo, la Universidad de San Carlos se convierte en un pilar, que declarada en 1875 como la universidad nacional de Guatemala; las facultades más importantes en esa época de 1882 son las de Medicina y Farmacia, Derecho y Notariado, Ingeniería, Filosofía y Literatura. En la época de 1918 se exigen varios cambios los cuales prevalecen hoy día, libertad de catedra, autonomía, un gobierno entre docentes estudiantes y graduados, una extensión con inclusión social, concursos de oposición para los docentes, fomento de la investigación, solidaridad latinoamericana, unidad obrero estudiantil.

Hoy en día la Universidad de San Carlos de Guatemala posee la mayor oferta académica de carreras en el país, aproximadamente 65 diferentes licenciaturas y 15 profesorados y técnicos universitarios, las cuales son impartidas dentro de sus 10 facultades, más de 4 escuelas no facultativas y 17 centros regionales que están distribuidos en todo el territorio de Guatemala[1].

[1] Universidad de San Carlos de Guatemala. *Historia*. https://es.wikipedia.org/wiki/ Universidad_de_San_Carlos_de_Guatemala. Consulta: 29 de julio de 2018.

1.2. Misión

"En su carácter de única universidad estatal le corresponde con exclusividad dirigir, organizar y desarrollar la educación superior del estado y la educación estatal, así como la difusión de la cultura en todas sus manifestaciones. Promoverá por todos los medios a su alcance la investigación en todas las esferas del saber humano y cooperará al estudio y solución de problemas nacionales[2]."

1.3. Visión

"La Universidad de San Carlos de Guatemala es la institución de educación superior estatal, autónoma, con una cultura democrática, con enfoque multi e intercultural, vinculada y comprometida con el desarrollo científico, social y humanista, con una gestión actualizada, dinámica y efectiva y con recursos óptimamente utilizados para alcanzar sus fines y objetivos, formadora de profesionales con principios éticos y excelencia académica[3]."

1.4. Facultad de Ingeniería

1.4.1. Historia

En 1834, se creó la Academia de Ciencias, la cual iba a ser la sucesora de la Universidad de San Carlos, en donde la enseñanza consistía de Álgebra, Geometría, Trigonometría y Física; se otorgaban títulos de agrimensores.

Según decretos gubernativos de la época de 1875 se crean las carreras de ingeniería en la recién fundada Escuela Politécnica; carreras que más tarde serían incorporadas a la universidad.

En 1879 se estableció la Escuela de Ingeniería en la Universidad de San Carlos de Guatemala y por decreto del gobierno en 1882 se eleva la categoría de facultad dentro de la misma universidad, separándose así de la Escuela Politécnica.

Después de estos años se tuvo una incertidumbre sobre la existencia de la facultad dentro de la universidad; logra su estabilidad hasta 1944, año cuando la Universidad de San Carlos logra su autonomía asignándole recursos financieros fijados según la Constitución de la República de Guatemala; consolidándose así

[2] Usac, Tricentenaria. *Misión y visión*. https://www.usac.edu.gt/misionvision.php. Consulta: 29 de julio de 2018.
[3] Ibíd.

la incorporación total de la Facultad de Ingeniería a la Universidad de San Carlos de Guatemala independizándose así de todas las instituciones gubernamentales[4].

1.4.2. Misión

"Formar profesionales en las distintas áreas de la ingeniería que, a través de la aplicación de la ciencia y la tecnología, conscientes de la realidad nacional y regional, y comprometidos con nuestras sociedades, sean capaces de generar soluciones que se adapten a los desafíos del desarrollo sostenible y los retos del contexto global"[5].

1.4.3. Visión

"Ser una institución académica con incidencia en la solución de la problemática nacional, formando profesionales en las distintas áreas de la ingeniería, con sólidos conceptos científicos, tecnológicos, éticos y sociales, fundamentados en la investigación y promoción de procesos innovadores orientados hacia la excelencia profesional"[6].

1.4.4. Escuela de Mecánica Eléctrica

La creación de la Escuela de Mecánica Eléctrica fue aprobada por el Honorable Consejo Superior Universitario en agosto de 1967. Inició sus labores a principios del año de 1968 bajo la dirección del fundador el ingeniero Rodolfo Koenigberger. Inicialmente tenía a su cargo las carreras de Ingeniería Eléctrica e Ingeniería Mecánica Electricista; posteriormente, en 1988 se creó en la carrera de Ingeniería en Electrónica[7].

---

[4] Usac, Facultad de Ingeniería. *Historia*. https://portal.ingenieria.usac.edu.gt/index.php/aspirante/antecedentes. Consulta: 29 de julio de 2018.

[5] Usac, Facultad de Ingeniería, *Misión y visión*. https://www.usac.edu.gt/catalogo/ingenieria.pdf. Consulta: 29 de julio de 2018.

[6] Ibíd.

[7] Usac, Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica. *Inicio*. http://eime.ingenieria.usac.edu.gt/. Consulta: 29 de julio de 2018.

1.4.4.1. Misión

"Formar profesionales competentes, con principios éticos y conciencia social, en los campos de Ingenierías Mecánica Eléctrica, Eléctrica y Electrónica, mediante técnicas de enseñanza actualizadas y fundamentos en la investigación, comprometidos con la sociedad, con el fin de contribuir al bien común y al desarrollo sostenible del país y de la región"[8].

1.4.4.2. Visión

> Se la institución académica líder a nivel nacional y regional, con incidencia en la problemática nacional, en la formación de profesionales de calidad, en los campos de las Ingenierías Mecánica Eléctrica, Eléctrica y Electrónica, emprendedores, con sólidos conocimientos científicos, tecnológicos, éticos, sociales, fundamentados en la investigación, orientados hacia la excelencia, reconocidos internacionalmente y comprometidos con el desarrollo sostenible de Guatemala y de la región[9].

[8] Usac, Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica. *Misión y visión*. http://eime.ingenieria.usac.edu.gt/. Consulta: 29 de julio de 2018.
[9] Ibíd.

2. DISTRIBUCIÓN DEL LABORATORIO PROPUESTO

2.1. Misión

Contribuir a cumplir los objetivos de la Escuela de Ingeniería Mecánica Eléctrica al ayudar a formar a futuros profesionales en materia de redes de área local y proveedor de servicio, logrando un balance entre la teoría y la práctica, utilizando siempre los mejores recursos para conseguirlo.

2.2. Visión

Ser un laboratorio donde el estudiante sea el recurso más valioso, siempre en constante evolución para adaptarse y anticiparse a las necesidades del mercado nacional.

2.3. Objetivos

2.3.1. Objetivo general

Forma al estudiante de manera que posea una fuerte comprensión de los fundamentos del funcionamiento tanto de las redes locales como las redes de proveedores de servicio y contribuir a la capacidad de adaptarse rápidamente a los constantes cambios los que están sujetas estas tecnologías.

2.3.2. Objetivos específicos

- Presentar un contenido de fácil comprensión hacia el alumno y de fácil acceso en español.

- Proveer de ejercicios prácticos para complementar el conocimiento del alumno.

- Facilitar el contenido al alumno de una forma digital para que pueda ser estudiado de una manera más práctica.

2.4. Perfil del auxiliar de laboratorio

El auxiliar a cargo deberá tener un conocimiento sólido de los fundamentos de las tecnologías cubiertas en el programa del laboratorio e idealmente poseer alguna certificación internacional en la materia; debido a que el equipo en existencia es de la marca Cisco se recomienda el Cisco Certified Network Associate (CCNA) o el Cisco Certified Network Professional (CCNP).

Además, deberá contar con facilidad de expresión a manera que pueda transmitir conceptos complejos utilizando ideas sencillas y poseer habilidades suaves especialmente paciencia.

De no existir una persona con un perfil semejante, se recomienda realizar un examen de oposición para esta plaza entre los alumnos que hayan aprobado el curso con las puntuaciones más altas.

2.5. Responsabilidades del auxiliar del laboratorio

Además de presentarse a dar clases, el auxiliar del laboratorio propuesto debe al inicio de cada semestre:

- Acordar el horario, el lugar y la duración del laboratorio con los estudiantes de la carrera y el ingeniero a cargo.

- Actualizar el material didáctico, el software necesario y la documentación a medida que sea necesario y asegurarse de que el mismo esté disponible para todos los estudiantes interesados.

- Determinar el estado del equipo de red presente en el laboratorio y dar mantenimiento de ser necesario.

- Asegurar que el estado de las computadoras del laboratorio sea óptimo y que el software necesario ha sido instalado.

- Informar a los nuevos estudiantes de la existencia del material de apoyo (documentación, software, entre otros) y cómo pueden obtenerlo, así como la calendarización del curso, la distribución de exámenes y tareas y cualquier otra información que el auxiliar considere pertinente.

- Preparar, recibir y calificar tareas y exámenes.

- Acordar con el tutor del laboratorio la fecha en la que se presentará el proyecto final del curso.

2.6. Metodología

El contenido del laboratorio está pensado para ser impartido en clases de 2 o 4 horas de manera semanal, este debe ser presentado de manera que la teoría siempre sea respaldada y reforzada por ejercicios prácticos guiados y tareas relacionadas con el tópico de cada clase.

La creación y ponderación de exámenes, tareas y ejercicios, así como sus respectivas fechas de entrega quedan a criterio del auxiliar del laboratorio; aunque se presentan sugerencias más adelante en este mismo trabajo.

Como nota final de este apartado, se sugiere que tanto tareas como investigaciones sean presentadas en formato digital únicamente a manera de economizar papel y ayudar al medio ambiente.

3. MARCO TEÓRICO

3.1. BGP

Prototocolo de enrutamiento utilizado en telecomunicaciones, este se denomina protocolo de enlace de frontera (Border Gateway Protocol), en proveedores de servicios, es el mas utilizado.

Este protocolo funciona por medio de sistemas autónomos, en los cuales pueden funcionar como BGP interno o BGP externo. Y presenta varias características que son funtamentales para su funcionamiento que se expone a continuación.

3.1.1. Asignación de direcciones IP públicas

La corporación para la asignación de nombre y números (ICANN, The Internet Corporation for Assigned Names and Numbers) es propietaria de los procesos mediante los cuales se asignan las direcciones públicas (IPV4 y IPV6). Existe otra organización relacionada, esta es la Autoridad de Asignación de Números de Internet, el cual es una organización que forma parte del ICANN (IANA, Internet Assigned Numbers Authority) que cumple también políticas del ICANN. Estas organizaciones definen las direcciones IPv4 que se pueden asignar a diferentes regiones geográficas, además de gestionar y desarrollar el sistema de nombres de dominio (DNS, Domain Name System) y los nuevos dominios de nivel superior (TLD, Top Level Domains), como los dominios que terminan en .com.

ICANN trabaja con varios grupos para la administración de direcciones IPv4, pero para esta asignación se necesita una estrategia en la cual se puede resumir de la siguiente manera:

- Paso 1. El ICANN y el IANA agrupan las direcciones públicas por región geográfica.

- Paso 2. El IANA asigna rangos de direcciones según los Registros Regionales de Internet (RIR, Regional Internet Registries).

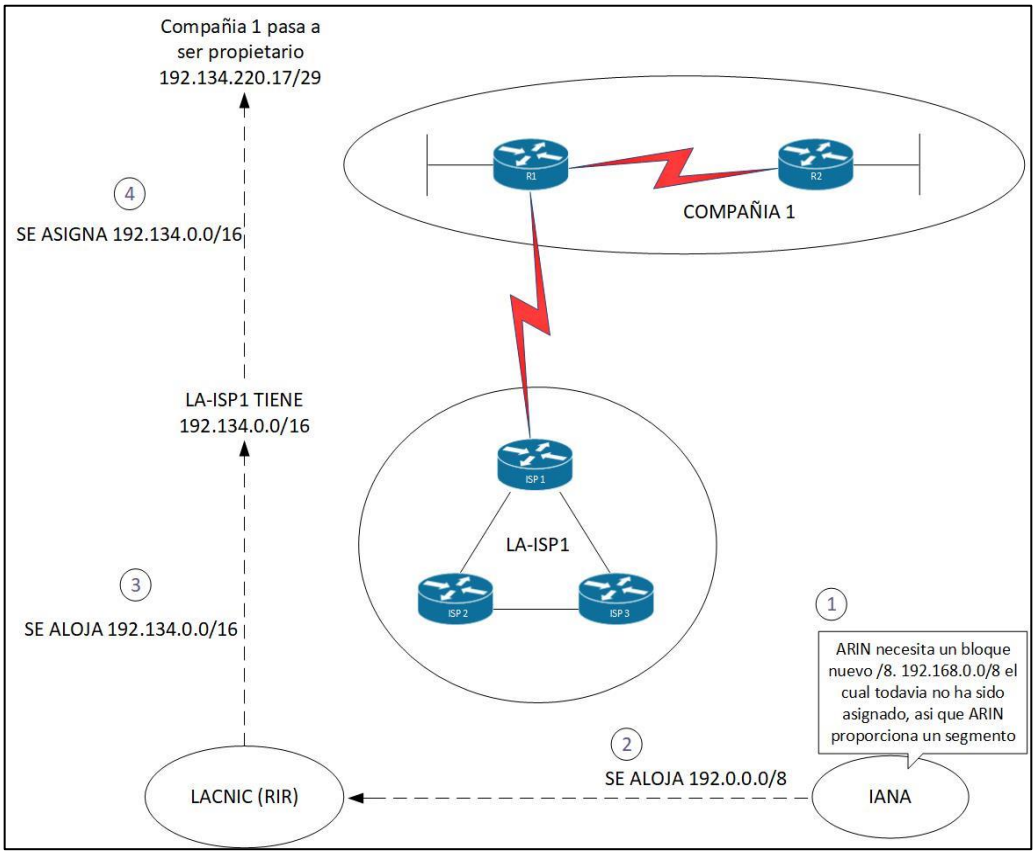
- Paso 3. Cada RIR posee un espacio de direcciones públicas, en rangos asignados al Registro Nacional de Internet o a los Registros Locales de Internet. Los proveedores de servicio de Internet por lo regular pertenecen a estos últimos.

- Paso 4. Cada tipo de registro de internet puede subdividir y asignar rangos de direcciones para un usuario final como lo es una organización. El 3 de febrero del 2011, la Autoridad para la Asignación de Números de Internet (IANA), Anuncia oficialmente el agotamiento de los últimos cinco bloques de direcciones IPv4, lo cual deja paso a una alta probabilidad a una implementación y utilización de IPv6.

El proceso inicia con estas dos organizaciones que son el ICANN y el IANA. Estas organizaciones mantienen un conjunto de direcciones IPv4 públicas aún sin asignar actualmente. (Vea en www.iana.org/numbers y busque en el enlace de IPv4 para verificar la lista actual). Cuando el Registro de Direcciones de Internet para Latinoamérica y el Caribe (LACNIC), el RIR para Latinoamérica, se da cuenta que se están quedando sin espacio de direcciones IPv4, LACNIC solicita un nuevo bloque de direcciones públicas. LACNIC

examina la solicitud, empieza la búsqueda para encontrar un nuevo bloque de direcciones públicas no asignadas (paso 1 en la figura 1), luego asigna el bloque LACNIC (paso 2 en la figura 1). A continuación, un ISP llamado LA-ISP1 (abreviatura de ISP latinoamericano) solicita a LACNIC un prefijo de asignación para un bloque de direcciones de tamaño /16. Después de que LACNIC se asegura de que LA-ISP1 cumple con algunos requisitos. LACNIC asigna un prefijo de 192.134.0.0/16 (paso 3 en la figura 1). Luego, cuando la compañía 1 se convierte en un cliente de LA-ISP1, LA-ISP1 puede asignar un prefijo a la compañía 1 (192.134.220.17/29 en este ejemplo, paso 4).

Figura 1. **Vista conceptual de asignación de direcciones públicas IPv4**



Fuente: elaboración propia, empleando Visio 2013.

Aunque en la figura 1 se muestra el proceso, los grandes ahorros para las direcciones públicas se producen porque el usuario de las direcciones IP se puede asignar un grupo mucho más pequeño que una única red de clase C. En algunos casos, las empresas sólo necesitan una ip pública; en otros casos, podrían necesitar solo unos pocas, como ocurre con la compañía 1 como se muestra en la figura 1. Esta práctica permite al registro de internet (IR, Internet Registry) asignar un bloque de direcciones de tamaño adecuado a cada cliente; reduce así el desperdicio de IP's.

- Sistema autónomo

Un sistema autónomo (AS) es un grupo de redes IP las cuales son gestionadas por uno o más operadores de servicio de red lo cual poseen una clara política y sola política de ruteo.

Cada sistema autónomo (AS) tiene un número el cual es utilizado como identificador único del sistema autónomo para el intercambio de rutas con otros sistemas externos, esto con el intercambio de información con protocolos externos tales como lo es BGP el cual es utilizado para intercambiar información de ruteo entre los sistemas autónomos.

Para la disminución de la complejidad que puede manejar las tablas de enrutamiento, que en este caso son tablas de ruteo globales, un nuevo número de sistema autónomo (ASN), debe ser asignado solamente en el caso en que una nueva política de ruteo sea necesaria.

Compartir un mismo ASN entre un grupo de redes que no están bajo la misma gestión va requerir una coordinación adicional entre los administradores de las redes y en algunos casos, va a requerir algún nivel de rediseño. Sin

embargo, esta es probablemente la única forma de implementar una política de ruteo deseada.

LACNIC distribuirá números de sistema autónomo a las organizaciones que cumplan los siguientes requisitos:

- La organización debe tener necesidad de interconexión con otros sistemas autónomos al momento de la solicitud, o tener programada la necesidad de interconexión en menos de 6 meses a partir del momento de la solicitud, luego de cumplido este plazo LACNIC podrá revocar el ASN asignado en caso el recurso no haya sido utilizado.

- Detallar la política de ruteo de la organización solicitante, indicando los ASN con los que se interconectarán y las direcciones IP que serán anunciadas a través del ASN solicitado.

3.1.2. Fundamentos

Las empresas que desean conectarse a internet lo hacen a través de uno o más ISP. Si una organización tiene una sola conexión a un ISP, es probable que no necesiten usar BGP. En su lugar, utilizarían una ruta predeterminada. Sin embargo, sí tienen múltiples conexiones a uno o a varios ISP, BGP puede ser apropiado porque les permite manipular atributos de ruta para seleccionar la ruta óptima.

Los protocolos que se ejecutan dentro de una empresa se llaman Protocolos Interiores (IGP). Los ejemplos de IGP incluyen RIP versiones 1 y 2, EIGRP y OSPF. Los protocolos que se ejecutan fuera de una empresa, o entre

sistemas autónomos, se llaman protocolos exteriores (EGP). Normalmente, los EGP se utilizan para intercambiar información de enrutamiento entre proveedores de servicios de internet (ISP), figura 2.

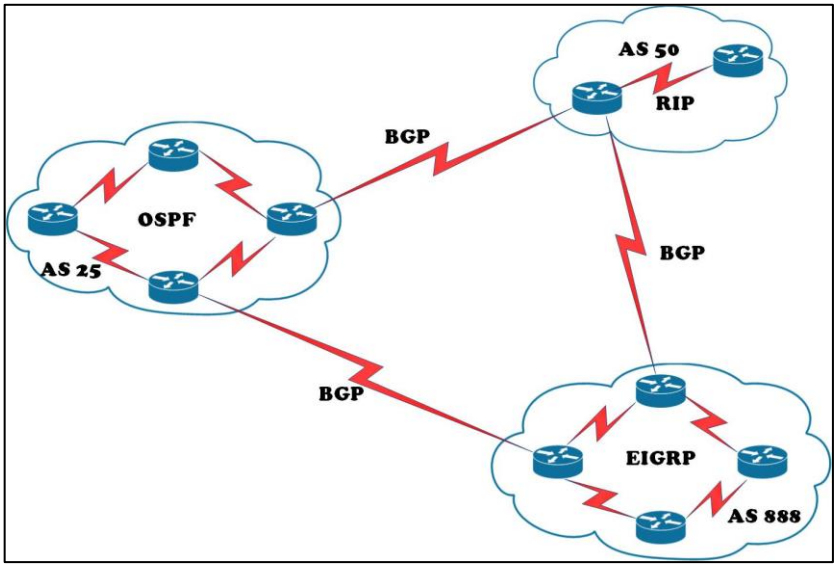
Desde 1994, Border Gateway Protocol Versión 4 (BGP4) se ha convertido en el principal protocolo de enrutamiento de internet. Todas las versiones anteriores se consideran obsoletas. La mayoría de los ISP deben usar BGP para establecer en el enrutamiento entre unos y otros.

La comprensión de las características importantes de BGP y la forma en que se comporta de manera diferente de IGPs es necesario saber cuándo utilizar y cuándo no utilizar BGP.

Un administrador de BGP debe entender las diversas opciones para configurar este protocolo correctamente para una implementación escalable.

Internet es una colección de sistemas autónomos que están interconectados para permitir la comunicación entre ellos. BGP proporciona el encaminamiento entre estos sistemas autónomos.

Figura 2. **Sistemas autónomos**



Fuente: elaboración propia, empleando Visio 2013.

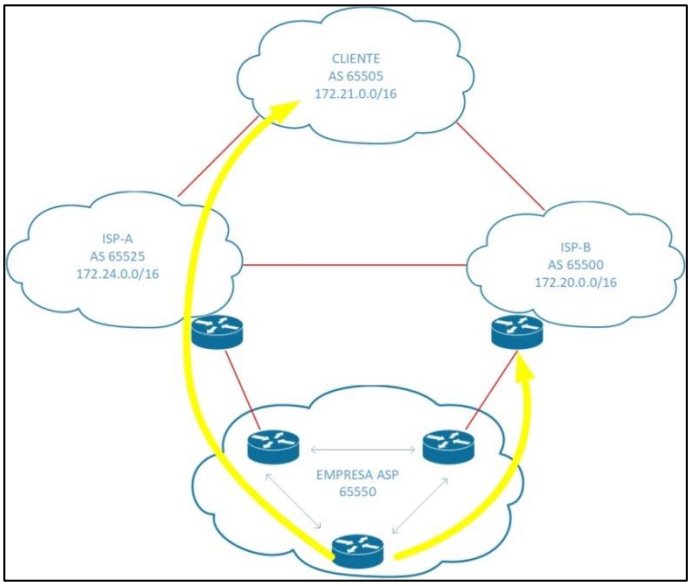
Los sistemas autónomos pueden utilizar más de un IGP, potencialmente con varios conjuntos de métricas. Desde el punto de vista del BGP, la característica más importante de un sistema autónomo es que otros sistemas autónomos les parece que tienen un único plan de enrutamiento interior coherente y presentan una imagen consistente de destinos accesibles. Todas las partes de un sistema autónomo deben conectarse entre sí.

Cuando un BGP se ejecuta entre routers en diferentes sistemas autónomos, se denomina External BGP (EBGP). Cuando un BGP se ejecuta entre enrutadores en el mismo sistema autónomo, se llama Internal BGP (IBGP).

Por ejemplo, la empresa AS 65550 de la figura 3 está aprendiendo rutas de ISP-A e ISP-B a través de EBGP, y también está ejecutando IBGP en todos

los routers internos. AS aprende sobre las rutas y elige la mejor manera para cada una basada en la configuración de los enrutadores en el sistema autónomo y las rutas BGP pasadas de los ISP.

Figura 3. **Utilización BGP para conectarse a internet**



Fuente: elaboración propia, empleando Visio 2013.

Una de las rutas que AS 65550 aprende de ISP-A es 172.24.0.0/16. Si esta ruta se pasa a través de AS 65550 utilizando IBGP y se anuncia erróneamente a ISP-B, ISP-B puede decidir que la mejor manera de llegar a 172.24.0.0/16 es a través de AS 65550, en lugar de a través del internet. El AS 65550 sería entonces considerado un sistema autónomo de tránsito lo cual es una situación muy indeseable. AS 65550 quiere tener una conexión a internet redundante, pero no quiere actuar como un sistema autónomo de tránsito entre los ISP. Se requiere una configuración cuidadosa de BGP para evitar esta situación.

3.1.3. Opciones *multihoming* de BGP

Multihoming es cuando un sistema autónomo tiene más de una conexión a internet. Dos razones típicas para el *multihoming* son las siguientes:

- Para aumentar la fiabilidad de la conexión a Internet: si una conexión falla, la otra conexión permanece disponible.

- Para aumentar el rendimiento de la conexión: se pueden utilizar mejores rutas en determinados destinos.

Los beneficios de BGP son evidentes cuando un sistema autónomo tiene múltiples conexiones EBGP a uno o múltiples sistemas autónomos. Múltiples conexiones redundantes a internet, de manera que la conectividad se pueda mantener si una única ruta de acceso no está disponible.

Una organización puede ser multihomed a un solo ISP o a múltiples ISP. Un inconveniente de tener todas sus conexiones a un único ISP es que los problemas de conectividad en ese ISP único pueden hacer que su sistema autónomo pierda la conectividad en ese ISP único pueden hacer que su sistema autónomo pierda conectividad a internet. Al tener conexiones con varios ISP, una organización obtiene los siguientes beneficios:

- Redundancia con las múltiples conexiones
- No está vinculado a la política de enrutamiento de un único ISP
- Más rutas a las mismas redes para una mejor manipulación de políticas

Si una organización quiere realizar *multihoming* con BGP, hay tres maneras comunes de hacer esto:

- Cada ISP solo pasa una ruta predeterminada al sistema autónomo: la ruta predeterminada se pasa a los routers internos.

- Cada ISP pasa solamente una ruta por defecto y rutas específicas del propietario al sistema autónomo: estas rutas pueden ser pasadas a routers internos, o todos los *routers* internos en la ruta de tránsito pueden ejecutar BGP y pasar estas rutas entre ellos.

- Cada ISP pasa todas las rutas al sistema autónomo: todos los routers internos en el camino de tránsito BGP y pasan estas rutas entre ellos.

3.1.4. Opción 1: rutas por defecto por todos los proveedores

Recibir solo una ruta predeterminada de cada ISP requiere el menor número de recursos dentro del sistema autónomo, ya que se utiliza una ruta predeterminada para llegar a cualquier destino externo. El sistema autónomo envía todas sus rutas a los ISP, los cuales los procesan y los pasan a otros sistemas autónomos. Si un enrutador en el sistema autónomo aprende sobre varias rutas predeterminadas, el protocolo de enrutamiento interior local instala la mejor ruta predeterminada en la tabla de enrutamiento, que es la que tiene la métrica IGP de menor costo. Esta ruta predeterminada IGP encamina los paquetes destinados a las redes externas a un enrutador de borde de este sistema autónomo, lo que está ejecutando EBGP con los ISP. El enrutador de borde utiliza la ruta predeterminada BGP para acceder todas las redes externas.

La ruta que toman los paquetes entrantes para llegar al sistema autónomo se decide fuera del sistema autónomo (dentro de los ISP y otros sistemas autónomos).

Los ISP regionales que tienen múltiples conexiones con ISP nacionales o internacionales implementan comúnmente esta opción. Los ISP regionales no utilizan BGP para la manipulación de la ruta. Sin embargo, requieren la capacidad de agregar nuevos clientes y las redes de los clientes que usan BGP. Si el ISP regional no utiliza BGP, cada vez que el ISP regional agrega un nuevo conjunto de redes, los clientes deben esperar hasta que los ISP nacionales agreguen estas redes a su proceso BGP y ubiquen rutas estáticas apuntando al ISP regional. Al ejecutar EBGP con los ISP nacionales o internacionales, el ISP regional necesita agregar solo las nuevas redes de los clientes a su proceso BGP. Estas nuevas redes se propagan automáticamente a través de internet con un retraso considerable, ya que BGP debe de converger.

Un cliente que elija recibir las rutas predeterminadas de todos los proveedores debe entender las limitaciones de esta opción:

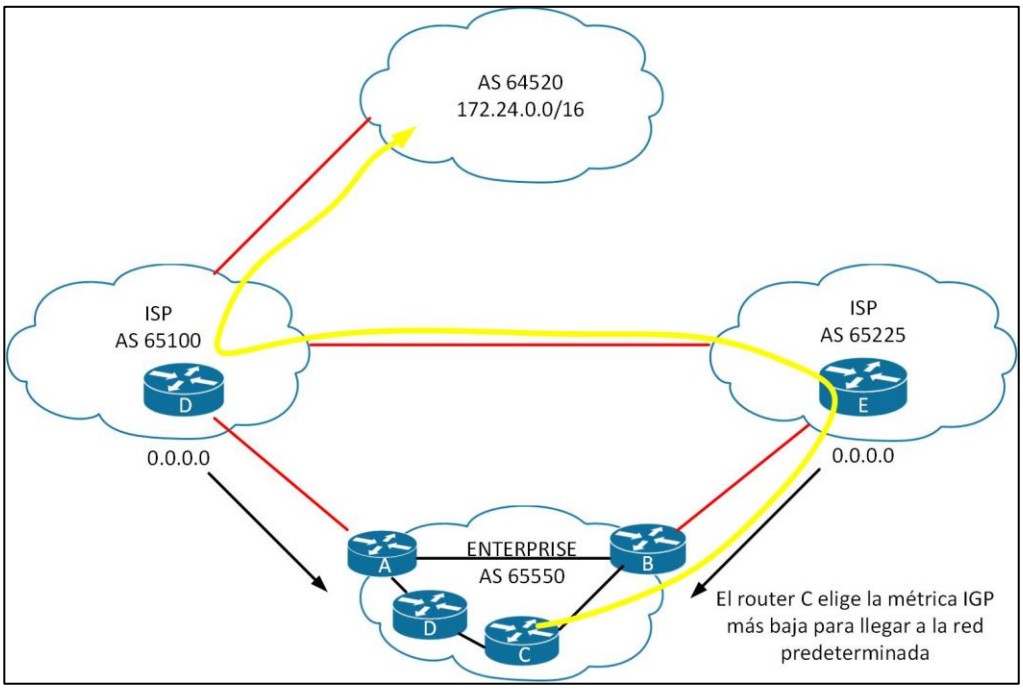
- No se realiza la manipulación de la ruta porque solo se recibe una sola ruta de cada ISP.

- La manipulación de ancho de banda es extremadamente difícil y solo se puede lograr manipulando la métrica IGP de la ruta predeterminada.

- Desviar un poco el tráfico de un punto de salida a otro es un reto porque todos los destinos utilizan la misma ruta predeterminada para la selección de ruta.

En la figura 4, el AS 65100 y el AS 65225 envían rutas predeterminadas al AS 65550. La métrica IGP que se utiliza para alcanzar la ruta predeterminada dentro del sistema autónomo determina que ISP utiliza un enrutador específico dentro 65550.

Figura 4. **Rutas por defecto para todos los proveedores**



Fuente: elaboración propia, empleando Visio 2013.

Por ejemplo, si usa RIP dentro del Sistema Autónomo 65550, el enrutador C selecciona la ruta con el conteo de saltos más bajo a la ruta predeterminada al enviar paquetes a la red 172.18.0.0.

3.1.5. Opción 2: rutas por defecto y actualizaciones parciales

En esta opción de diseño *multihoming*, todos los ISP pasan rutas por defecto y seleccionan rutas específicas del sistema autónomo.

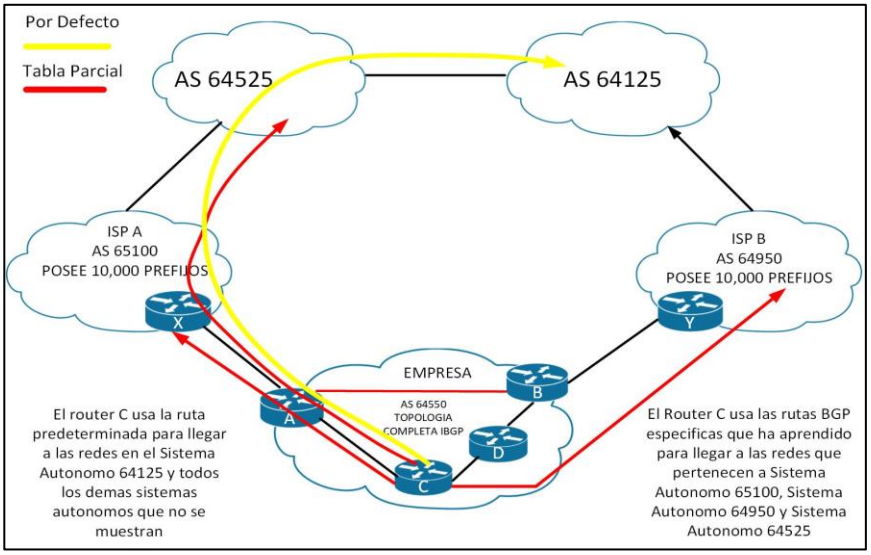
Una empresa que ejecuta EBGP con un ISP que desea una tabla de enrutamiento parcial generalmente recibe las redes que el ISP y sus otros

clientes poseen. La empresa también puede recibir las rutas desde cualquier otro sistema autónomo.

Los ISP principales se asignan entre 2 000 y 10 000 bloques de direcciones de IP de la Autoridad de Números Asignados de Internet (IANA), que reasignan a sus clientes, si el ISP pasa la información a un cliente que solo desea una tabla parcial de enrutamiento BGP, el cliente puede redistribuir estas rutas en su IGP. Los *routers* internos del cliente (estos enrutadores no están ejecutando BGP) pueden recibir estas rutas a través de la redistribución. Pueden tomar el punto de salida más cercano basado en la mejor métrica de redes específicas, en lugar de tomar el punto de salida más cercano basado en la ruta predeterminada.

En la figura 5, los ISP en el Sistema Autónomo 65100 y el Sistema Autónomo 64950 envían rutas predeterminadas y las rutas que cada ISP posee en el Sistema Autónomo 64550. La empresa (Sistema Autónomo 64550) solicitó a ambos proveedores que también envíen rutas a redes en el Sistema Autónomo 64525 y el Sistema Autónomo 64550.

Figura 5. **Por defecto de todos los proveedores y tabla parcial**



Fuente: elaboración propia, empleando Visio 2013.

Al ejecutar IBGP entre los enrutadores internos dentro del Sistema Autónomo 64550 y el Sistema Autónomo 64550 puede elegir la ruta óptima para llegar a las redes de los clientes (Sistema Autónomo 64525, en este caso). Las rutas hacia el Sistema Autónomo 64125 y hacia otros sistemas autónomos que no están anunciados específicamente para el Sistema Autónomo 64550 por el ISP A y el ISP B se deciden por la métrica IGP que se utiliza para alcanzar la ruta predeterminada dentro del sistema autónomo.

3.1.6. Opción 3: rutas completas de todos los proveedores

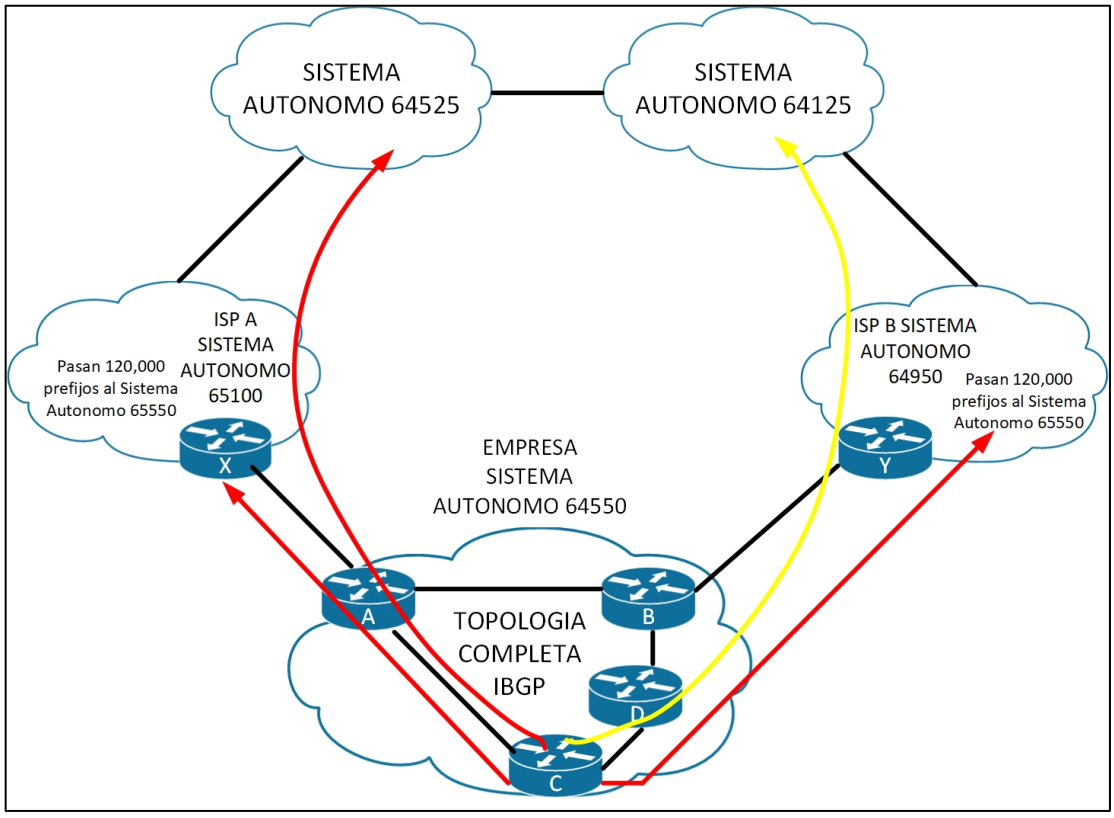
En la tercera opción *multihoming*, todos los ISP pasan todas las rutas al sistema, y IBGP se ejecuta en todos los *routers* en la ruta de tránsito en este sistema autónomo. Esta opción permite a los *routers* internos del sistema autónomo tomar el camino a través del mejor ISP para cada ruta.

Esta configuración requiere una gran cantidad de recursos dentro del sistema autónomo, ya que debe procesar todas las rutas externas.

El sistema autónomo envía todas sus rutas a los ISP que procesan las rutas y las pasan a otros sistemas autónomos.

En la figura 6, el AS 65100 y el AS 64950 envían todas las rutas al AS 64550. El protocolo BGP determina el ISP que utiliza un router específico dentro del AS 64550 para llegar a las redes externas.

Figura 6. **Rutas completas de todos los proveedores**



Fuente: elaboración propia, empleando Visio 2013.

Los *routers* en los sistemas autónomo 64550 se pueden configurar para influir en la ruta a ciertas redes. Por ejemplo, los enrutadores A y B pueden influir en el tráfico saliente desde el Sistema Autónomo 64550.

3.1.7. Enrutamiento BGP entre sistemas autónomos

El objetivo principal de BGP es proporcionar un sistema de enrutamiento entre dominios que garantice el intercambio sin bucle de información de enrutamiento entre sistema autónomos. Los *routers* intercambian información sobre rutas a las redes destino.

BGP es un sucesor del Exterior Gateway Protocol, que se desarrolló para aislar las redes entre sí a medida que crecía internet. Es importante no confundir el protocolo de puerta exterior con la categoría EGP.

Hay muchas RFC relacionadas con BGP4, la versión actual de BGP, que incluye 1772, 1773, 1774, 1930, 1966, 1997, 1998, 2042, 2385, 2439, 2545, 2547, 2796, 2858, 2918, 3065, 3107, 3392, 4223 y 4271.

BGP4 tiene muchas mejoras sobre los protocolos anteriores. Internet utiliza BGP4 exclusivamente para conectar a las empresas con los ISP y conectar a los ISP entre sí.

BGP4 y sus extensiones son las únicas versiones aceptables de BGP disponibles para su uso en la internet pública. BGP4 lleva una máscara de red para cada red anunciada y soporta tanto enmascaramiento de subred de longitud variable (VLSM) como CIDR. Los predecesores de BGP4 no apoyaban estas capacidades, que son actualmente obligatorias en Internet.

Cuando se utiliza CIDR en un enrutador central para un ISP importante, la tabla de enrutamiento IP, que se compone principalmente de rutas BGP, tiene más de 170 000 bloques CIDR. No utilizar CIDR en el nivel de internet haría que la tabla de enrutamiento IP tenga más de 2 000 000 de entradas. El uso de BGP4 y CIDR evita que la tabla de enrutamiento de internet sea demasiada grande para interconectar a millones de usuarios.

Los números de un sistema autónomo son de 16 bits, que van de 1 a 65535. RFC 1930 proporciona directrices para el uso de números. Los números 64512 a 65535 están reservados para uso privado, al igual que las direcciones IP privadas. Los números de sistema autónomo utilizados en este documento son todos de ámbito privado para evitar la publicación de números pertenecientes a las organizaciones.

El uso de un número de sistema autónomo asignada por IANA en lugar de un número privado solo es necesario si su organización planea usar un EGP, como BGP, y conectarse a una red pública, como internet.

- Comparación con IGPs

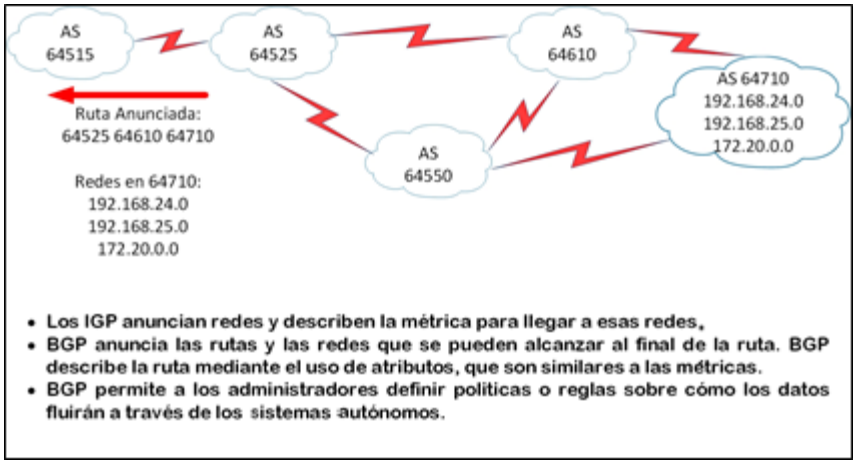
BGP funciona de forma diferente a los IGP. Un protocolo de enrutamiento interno busca la ruta más rápida desde un punto en una red corporativa a otra basada en ciertas métricas. RIP usa recuentos de salto que buscan cruzar el menor número de dispositivos de capa 3 para llegar a la red destino. OSPF y EIGRP buscan la mejor velocidad de acuerdo con la declaración de ancho de banda en la interfaz. Todos los protocolos de enrutamiento interno miran el coste de la ruta de acceso a un destino.

En contraste, BGP, un protocolo de enrutamiento externo, no mira la velocidad para el mejor camino. Más bien, BGP es un protocolo de enrutamiento basado en directivas que permite a un sistema autónomo controlar el flujo de tráfico utilizando múltiples atributos de ruta BGP. BGP permite a un proveedor utilizar completamente todo su ancho de banda mediante la manipulación de estos atributos de ruta.

3.1.8. Funcionalidad vector distancia

Los protocolos de enrutamiento interno anuncian una lista de redes y las métricas para llegar a cada red. En cambio, los enrutadores BGP intercambian información de accesibilidad de red, llamados vectores de trayectoria, formados por atributos de ruta. La información de vector de trayecto incluye una lista de la ruta completa de números de sistema autónomo BGP (salto por salto) necesarios para alcanzar una red de destino y las redes que son accesibles al final de la ruta, ver figura 7.

Figura 7. **BGP ruta vector enrutamiento**



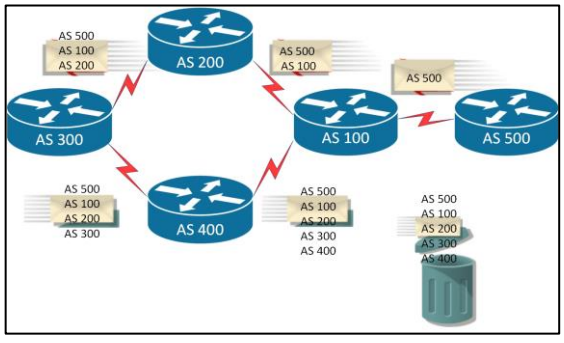
Fuente: elaboración propia, empleando Visio 2013.

Otros atributos incluyen la dirección IP para llegar al siguiente sistema autónomo (el siguiente atributo de salto) y una indicación de cómo las redes al final de la ruta se introdujeron en BGP (el atributo de código origen). Esta información del camino del sistema autónomo es útil para construir un gráfico de sistema autónomos basados en la información intercambiada entre vecinos BGP.

BGP ve toda la red como un gráfico, o árbol, de sistemas autónomos. La conexión entre dos sistemas cualquiera forma un camino. La recopilación de información de trayecto se expresa como una secuencia de números de sistemas autónomos llamados el camino AS. Esta secuencia forma una ruta para llegar a un destino específico.

La ruta AS siempre está sin bucles. Un enrutador que ejecuta BGP no acepta una actualización de enrutamiento que ya incluye el número de sistema autónomo al cual el enrutador pertenece en la lista de rutas, ya que la actualización ya ha pasado a través de su sistema autónomo y aceptarlo de nuevo resultaría en un bucle de enrutamiento, ver figura 8.

Figura 8. **BGP evita los bucles de enrutamiento**



Fuente: elaboración propia, empleando Visio 2013.

3.1.9. Políticas de ruteo BGP

BGP permite el enrutamiento de las decisiones de política a nivel del sistema autónomo que se aplicará. Estas políticas pueden implementarse para todas las redes propiedad de un sistema autónomo, para un cierto bloque CIDR de números de red (prefijos), o para redes individuales o subredes.

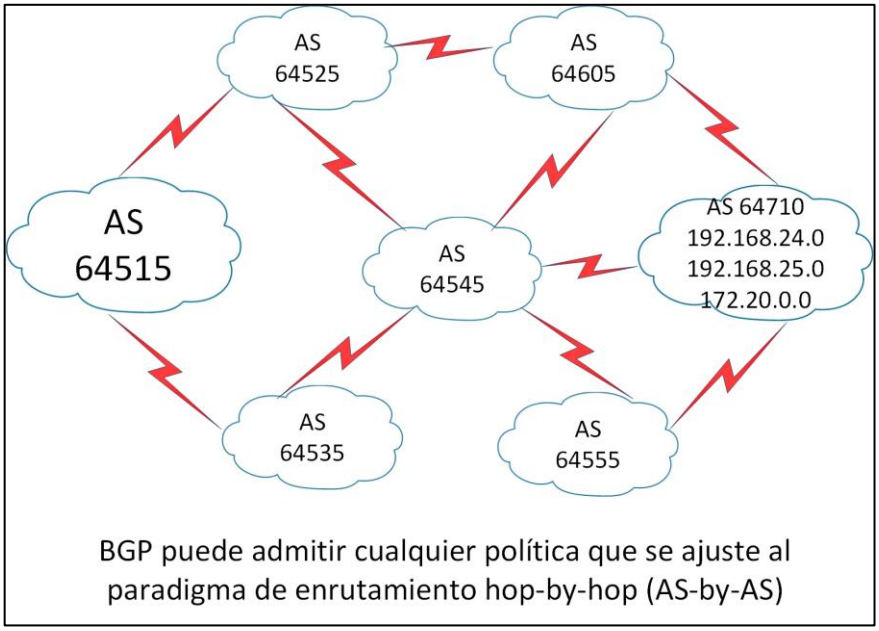
BGP especifica que un enrutador BGP puede anunciar a los sistemas autónomos vecinos solo las rutas que usa por sí mismo. Esta regla refleja el paradigma de enrutamiento *hop-by-hop* que generalmente usa internet.

El paradigma de enrutamiento *hop-by-hop* no admite todas las políticas posibles. Por ejemplo, no puede influir en como un sistema autónomo vecino en ruta el tráfico, pero puede influir en cómo su tráfico llega a un sistema autónomo vecino. BPG es compatible con cualquier política que se ajuste al paradigma de enrutamiento *hop-by-hop*.

Debido a que actualmente internet utiliza el paradigma de enrutamiento hop-by-hop solamente, y porque BGP puede soportar cualquier política que se ajuste a ese paradigma, BGP es altamente aplicable como un protocolo de enrutamiento de sistema Inter-autónomo.

Por ejemplo, en la figura 9, las siguientes rutas son posibles para el Sistema Autónomo 64515 para llegar a las redes en Sistema Autónomo 64710 a través del Sistema Autónomo 64525:

Figura 9. **Políticas de enrutamiento BGP**



Fuente: elaboración propia, empleando Visio 2013.

- 64525, 64605, 64710
- 64525, 64605, 64545, 64555, 64710
- 64525, 64545, 64605, 64710
- 64525, 64545, 64555, 64710

El sistema autónomo 64515 no ve todas estas posibilidades.

El sistema autónomo 64525 anuncia al sistema autónomo 64515 solo su mejor ruta, 64525, 64605, 64710, de la misma manera que los IGP anuncian solo sus mejores rutas de menor costo. Esta ruta es la única ruta a través del sistema autónomo 64525 que ve a sistema autónomo 64515. Todos los paquetes que están destinados a 64710 a 64525 toman esta ruta.

Aunque existen otras rutas, sistema autónomo 64515 solo puede usar lo que sistema autónomo 64525 anuncia para las redes en sistema autónomo 64710. La ruta de los sistema autónomos anunciadas para, 64525, 64605, 64710, es la ruta AS-by-AS (*Hop-by-Hop*) que sistema autónomo 64525 utiliza para llegar a las redes en sistema autónomo 64710. El sistema autónomo 64525 no anunciará otra ruta, como 64525, 64545, 64605, 64710, porque no eligió esa como la mejor ruta basada en la política de enrutamiento BGP en el sistema autónomo 64525.

El AS 64515 no conoce la segunda mejor ruta ni ninguna otra ruta desde el AS 64525, a menos que la mejor ruta del AS 64525 deje de estar disponible.

Incluso si el AS 64515 conocía otra ruta a través del AS 64525 y deseaba usarla, el AS 64525 no enrutaría paquetes a lo largo de esta otra ruta porque el AS 64525 seleccionó 64525, 64605, 64710, como su mejor ruta, y todos los enrutadores del AS 64525 usan esa ruta como asunto de la política BGP. BGP no permite que un sistema autónomo envié tráfico a un sistema autónomo vecino, con la intención de que el tráfico tome una ruta diferente de la tomada por el tráfico que se origina en el sistema autónomo vecino.

Para llegar a las redes en el AS 64710, el AS 64515 opta por utilizar el AS 64525 o puede optar por seguir la ruta que anuncia el AS 64535. El AS 64515 selecciona la mejor ruta para tomar en función de sus propias políticas de enrutamiento BGP.

3.1.10. Características de BGP

BGP es utilizado por los ISP para que puedan comunicarse e intercambiar paquetes. Los ISP tienen múltiples conexiones entre sí y acuerdos para

intercambiar actualizaciones. BGP implementa a los acuerdos entre dos o más sistemas autónomos.

BGP es más apropiado cuando al menos una de las siguientes condiciones exista:

- Un sistema autónomo permite que los paquetes transitan a través de los mismos sistemas autónomos (por ejemplo, en un proveedor de servicios).

- Un sistema autónomo tiene múltiples conexiones con otros sistemas autónomos.

- La política de enrutamiento y la selección de rutas para el tráfico que entra y sale del sistema autónomo debe manipularse.

El control y filtrado inapropiados de las actualizaciones de BGP pueden permitir que un sistema autónomo externo afecte el flujo de tráfico a su sistema autónomo. Es importante saber cómo funciona BGP y como configurarlo correctamente para evitar que esto ocurra.

Por ejemplo, si es un cliente conectado a ISP-A e ISP-B (para redundancia), desea implementar una directiva de enrutamiento para asegurarse de que ISP-A no envíe tráfico a ISP-B a través de su sistema autónomo. No desea desperdiciar valiosos recursos y ancho de banda dentro de su sistema autónomo para enrutar el tráfico de sus ISP, pero si desea recibir tráfico destinado a su sistema autónomo a través de cada ISP.

BGP no siempre es una solución adecuada para interconectar sistemas autónomos. Por ejemplo, si solo existe una ruta de salida del sistema autónomo, una ruta predeterminada es la solución más adecuada. En este caso, BGP usaría innecesariamente los recursos del CPU del enrutador y la memoria.

BGP no siempre es apropiado. No se debe de usar BGP si se tiene las siguientes condiciones:

- Comprensión limitada de filtrado de rutas y proceso de selección de rutas BGP.

- Una única conexión a internet u otro sistema autónomo.

- Falta de memoria o poder del procesador para manejar actualizaciones constantes en los *routers* BGP.

Si la política de enrutamiento que implementa en un sistema autónomo es coherente con la política del sistema de ISP, no es necesario o deseable configurar BGP en ese sistema autónomo.

BGP se clasifica como un protocolo vector distancia avanzado, pero en realidad es un protocolo de vector-camino. BGP es muy diferente de los protocolos estándar de vector distancia, como RIP.

BGP es un protocolo de vector distancia con las siguientes mejoras:

- BGP se ejecuta en la parte superior de TCP (puerto 179)
- Solo actualizaciones incrementales y activas

- Mensajes periódicos *Keepalive* para verificar la conectividad TCP
- Métricas ricas (llamadas vectores de trayecto o atributos)
- Diseñado para escalar a grandes redes (por ejemplo, Internet)

BGP utiliza TCP como su protocolo de transporte, que proporciona una entrega fiable orientada a la conexión. Dos enrutadores que utilizan BGP forman una conexión TCP entre sí e intercambian mensajes para abrir y confirmar los parámetros de conexión. Estos dos enrutadores BGP se denominan routers de pares o vecinos.

Después de realizar la conexión, los interlocutores BGP intercambian tablas de enrutamiento completas. Sin embargo, como la conexión es fiable, los pares BGP envían posteriormente solo cambios (actualizaciones incrementales o activadas) después de eso. Los enlaces confiables no requieren actualizaciones periódicas de enrutamiento; por lo tanto, los enrutadores utilizan actualizaciones activadas en su lugar. BGP envía mensajes *keepalive*, similares a los mensajes de saludo enviados por OSPF, IS-IS e EIGRP.

BGP es el único protocolo de enrutamiento IP que utiliza TCP como su capa de transporte. EIGRP y OSPF residen directamente por encima de la capa IP, y RIPv1 y RIPv2 utilizan el protocolo de datagramas de usuario (UDP) para su capa de transporte.

EIGRP y OSPF tienen su propia función interna para garantizar que los paquetes se reciban correctamente. Estos protocolos utilizan una ventana de uno por uno, por lo que, para varios paquetes, el siguiente paquete no puede ser enviado hasta que se reciba un acuse de recibido del primer paquete de actualización. Este proceso puede ser muy ineficiente y causar problemas de latencia si miles de paquetes de actualización deben ser intercambiados a

través de enlaces seriales relativamente lentos. Sin embargo, EIGRP y OSPF rara vez tienen miles de paquetes de actualización para enviar. EIGRP puede contener más de 100 redes en un paquete de actualización por lo que 100 paquetes de actualización EIGRP puede contener hasta 10 000 redes, y la mayoría de las organizaciones no tienen 10 000 subredes en la empresa.

BGP, por otro lado, cuenta con más de 170 000 redes (y creciendo) en internet para anunciarse, utiliza TCP para manejar la función de reconocimiento. TCP utiliza una ventana dinámica, que permite 65 576 bytes para estar pendientes antes de que se detenga y espera un acuse de recibido. Por ejemplo, si se envían paquetes de 1 000 bytes, BGP se detendrá y esperará un reconocimiento sólo cuando no se hayan reconocido 65 paquetes cuando se utiliza el tamaño máximo de la ventana.

TCP está diseñado para utilizar una ventana deslizante en la que el receptor reconoce a mitad de camino de la ventana de envío. Este método permite que cualquier aplicación TCP, como BGP, continúe transmitiendo paquetes sin tener que detenerse y esperar como requerían OSPF o EIGRP.

3.1.11. Base de datos BGP

Un router que ejecuta BGP mantiene varias tablas para almacenar información BGP que recibe de y envía a otros enrutadores. Estas tablas incluyen una tabla vecina, una tabla BGP (también llamada base de datos de reenvío o base de datos de topología) y una tabla de enrutamiento IP.

- Tabla de vecinos
    - Lista de vecinos BGP

- Tabla BGP (base de datos de reenvío)
    - Lista de todas las redes aprendidas de cada vecino
    - Puede contener múltiples rutas de acceso a las redes de destino
    - Contiene atributos BGP para cada ruta

- Tabla de enrutamiento
    - Lista de los mejores caminos a las redes de destino

Para que BGP establezca adyacencia, se debe configurar explícitamente para cada vecino. BGP utiliza TCP con cada uno de los vecinos configurados y realiza un seguimiento del estado de estas relaciones mediante el envío periódico de un mensaje BGP TCP *keepalive*. Tomar en cuenta una nota muy importante, lo cual es que BGP envía TCP *keepalives* cada 60 segundos de forma predeterminada.

Los enrutadores que ejecutan un proceso de enrutamiento BGP a menudo se denominan altavoces BGP. Dos altavoces BGP que forman una conexión TCP entre sí con el fin de intercambiar información de enrutamiento se denominan vecinos o pares.

Después de establecer una adyacencia, los vecinos intercambian las rutas BGP que están en su tabla de enrutamiento IP. Cada enrutador recoge estas rutas de cada vecino que establece con éxito una adyacencia y luego los coloca en su base de datos de reenvío BGP. Todas las rutas que se han prendido de cada vecino se colocan en la base de datos de reenvío BGP. Las mejores rutas para cada red se seleccionan de la base de datos de reenvío BGP. Las mejores rutas para cada red se seleccionan de la base de datos de reenvío BGP utilizando el proceso de selección de rutas BGP y luego se ofrecen a la tabla de enrutamiento IP.

Cada enrutador compara las rutas BGP ofrecidas con otras posibles rutas a estas redes, y la mejor ruta, basada en la distancia administrativa, se instala en la tabla de enrutamiento IP.

Las rutas EBGP (rutas BGP aprendidas desde un sistema externo) tienen una distancia administrativa de 20. Las rutas IBGP (rutas BGP aprendidas desde dentro del sistema autónomo) tienen una distancia administrativa de 200.

3.2. IBGP y EBGP

Protocolo interior y exterior por lo que su función es reconocer una red interna a nivel de proveedor de servicio asi como el exterior que es para la interconexión entre otros paises

3.2.1. Relaciones con los vecinos BGP

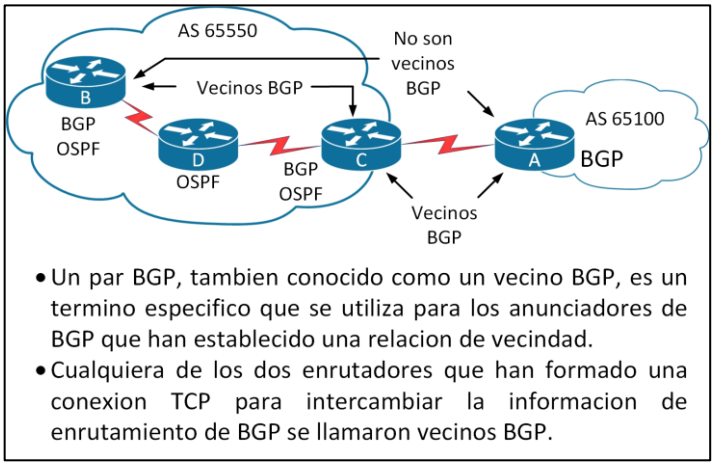
Ningún *router* puede manejar todas las conexiones con todos los *routers* que ejecutan BGP. Decenas de miles de routers funcionan con BGP y están conectados a internet, representando más de 21 000 sistemas autónomos.

Un enrutador BGP forma una relación de vecino directo con un número limitado de otros enrutadores BGP. A través de estos vecinos BGP, un router BGP aprende de los caminos a través de internet para llegar a cualquier red anunciada. Cualquier router que ejecute BGP se conoce como un anunciador BGP.

El término *BGP peer* tiene un significado específico: un anunciador de BGP que está configurado para formar una relación de vecino con otro anunciador BGP con el fin de intercambiar directamente información de

enrutamiento BGP entre sí. Un anunciador BGP tiene un número limitado de vecinos BGP con lo que se asocia y forma una relación basada en TCP.

Figura 10. ***Peers* = *Neighbors* (compañeros = vecinos)**



Fuente: elaboración propia, empleando Visio 2013.

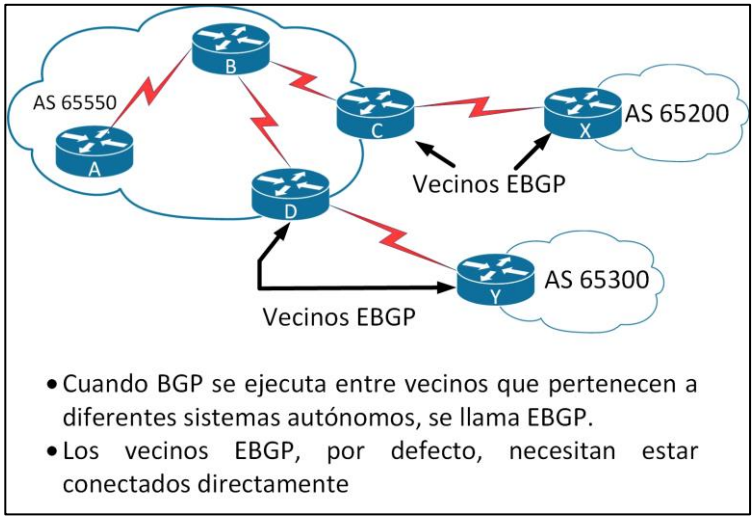
Los BGP peers también son conocidos como vecinos BGP y pueden ser internos o externos al sistema autónomo.

Un vecino BGP debe configurarse con el comando BGP *neighbor*. El administrador incluye al hablante BGP para establecer una relación con la dirección que aparece en el comando del vecino e intercambiar las actualizaciones de enrutamiento BGP con ese vecino.

3.2.2. Establecer una conexión entre vecinos externos BGP

Recuerde que cuando BGP se ejecuta entre *routers* en diferentes sistemas autónomos, se llama EBGP. En general, los enrutadores que ejecutan EBGP están conectados directamente entre sí, ver figura 10.

Figura 11. **BGP externo**



Fuente: elaboración propia, empleando Visio 2013.

Para que dos enrutadores intercambien actualizaciones de enrutamiento BGP, la capa de transporte TCP-confiable en cada lado debe pasar satisfactoriamente el *TCP three-way handshake* antes de que se pueda establecer la sesión BGP. Por lo tanto, la dirección IP utilizada en el comando de BGP neighbors debe ser accesible sin utilizar un IGP, lo que puede lograrse apuntando a una dirección que es accesible a través de una red conectada directamente o mediante rutas estáticas a esa a esa dirección IP.

3.2.3. Estableciendo una conexión entre los vecinos internos de BGP

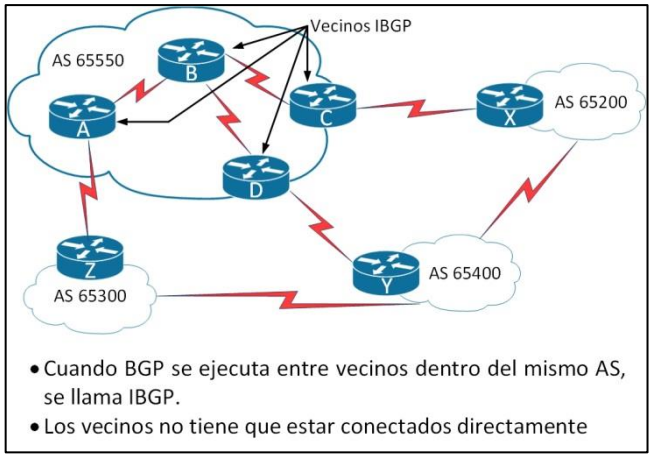
Cuando BGP funciona entre *routers* dentro del mismo sistema autónomo, se llama IBGP. IBGP intercambia información BGP para que todos los hablantes BGP tengan la misma información de enrutamiento BGP sobre sistemas autónomos externos.

Los enrutadores que ejecutan IBGP no tienen que estar conectados directamente entre sí, siempre y cuando puedan llegar entre sí de modo que TCP *handshaking* se puede realizar para establecer las relaciones de vecino BGP. El vecino IBGP puede ser alcanzado por una red directamente conectada, rutas estáticas o por el protocolo de enrutamiento interno.

Debido a que existen múltiples rutas en general dentro de un sistema autónomo para llegar a los otros enrutadores IBGP; generalmente, se usa una dirección de bucle en el comando BGP *neighbors* para establecer las sesiones IBGP.

Por ejemplo, cuando varios enrutadores en un sistema autónomo están ejecutando BGP, intercambian actualizaciones de enrutamiento BGP entre sí. En la figura 12, los enrutadores A, C y D aprenden las rutas de acceso a los sistemas autónomos externos desde sus respectivos vecinos EBGP (enrutadores Z, Y y X).

Figura 12. **BGP interno**



Fuente: elaboración propia, empleando Visio 2013.

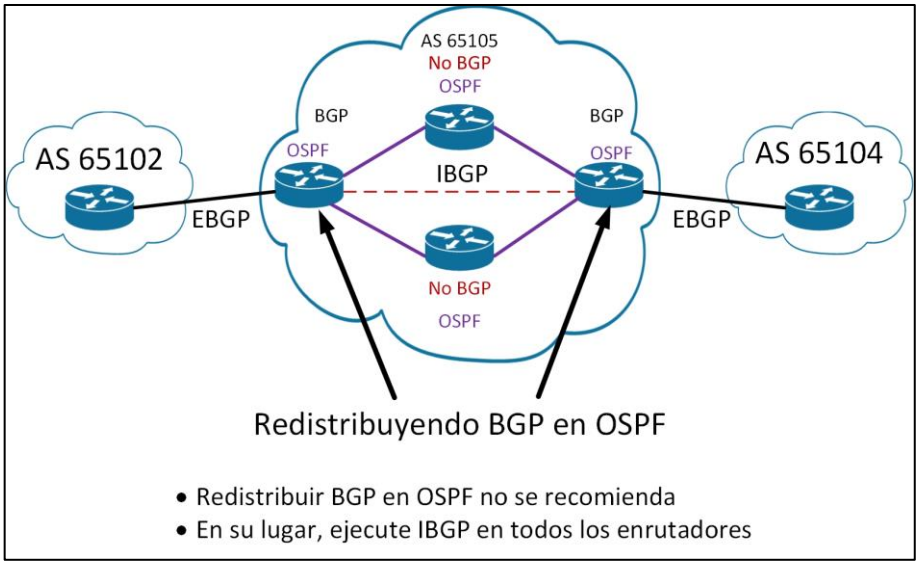
Si el enlace entre los enrutadores D y Y se reduce, el enrutador D debe aprender nuevas rutas a los sistemas autónomos externos. Otros enrutadores BGP dentro de AS 65525 que estaban utilizando el enrutador D para llegar a redes externas también deben ser informados de que la ruta a través del enrutador D no está disponible. Estos enrutadores BGP dentro de AS 65525 necesitan tener las rutas alternativas a través de *routers* A y C en su base de datos de reenvío BGP. Debe configurar sesiones IBGP entre todos los enrutadores BGP en AS 65525 para cada enrutador dentro del sistema autónomo aprenda sobre rutas a las redes externas a través de IBGP.

3.2.4. Sincronización dentro de un sistema autónomo

BGP originalmente estaba destinado a correr a lo largo de las fronteras de un sistema autónomo con los enrutadores en el medio del sistema autónomo ignorantes de los detalles de BGP (de ahí el nombre Border Gateway Protocol).

Un sistema autónomo de tránsito, como el de la figura 13, es un sistema autónomo que enruta el tráfico de un sistema autónomo externo a otro sistema autónomo externo. Típicamente, los sistemas autónomos de tránsito son ISP.

Figura 13. **IBGP en un sistema autónomo de tránsito (proveedor de servicios de Internet)**



Fuente: elaboración propia, empleando Visio 2013.

Todos los enrutadores de un sistema autónomo de tránsito deben tener un conocimiento completo de las rutas externas. Teóricamente, una forma de lograr este objetivo es reducir las rutas BGP en un IGP en los enrutadores de borde. Sin embargo, hay problemas asociados con este enfoque. Dado que la tabla de enrutamiento de internet actual es muy grande, la redistribución de todas las rutas BGP en un IGP no es un método escalable para que los enrutadores interiores dentro de un sistema autónomo conozcan las redes externas.

Otro método que puede usar es ejecutar IBGP dentro del sistema autónomo.

Por definición, el comportamiento predeterminado de BGP requiere que se sincronice con el IGP antes de que BGP pueda anunciar rutas de tránsito a sistemas autónomos externos. La regla de sincronización de BGP establece que un enrutador BGP no debe publicarse en destinos de vecinos externos, aprendidos de vecinos IBGP, a menos que esos destinos también sean conocidos a través de un IGP. Si un enrutador conoce estos destinos a través de un IGP, asume que la ruta ya se ha propagado dentro del sistema autónomo y que se garantiza la accesibilidad interna.

3.2.5. IBGP en un sistema no transitorio

Un sistema autónomo no transitorio, como una organización que tiene *multihoming* con los dos ISP, no pasa las rutas entre los ISP. Sin embargo, los enrutadores BGP dentro del sistema autónomo aún requieren el conocimiento de todas las rutas BGP pasadas al sistema autónomo para tomar decisiones de enrutamiento adecuadas.

BGP no funciona de la misma manera que los IGP. Debido a que los diseñadores de BGP no podían garantizar que un sistema autónomo ejecutará BGP en todos los enrutadores, se tuvo que desarrollar un método para asegurar que los anunciadores de IBGP pudieran pasar las actualizaciones entre ellos mientras se aseguraba de que no existirán bucles de enrutamiento.

Para evitar los bucles de enrutamiento dentro de un sistema autónomo, BGP especifica que las rutas aprendidas a través de un IBGP nunca se propagan a otros pares IBGP.

El comando *neighbor* habilita las actualizaciones de BGP entre los anunciadores BGP. Por defecto, se supone que cada altavoz BGP tiene una

conexión directa con todos los anunciadores IBGP en el sistema autónomo, lo que se conoce como un IBGP de malla completa. Sin embargo, los enrutadores no tienen que estar conectados directamente para ser completamente mallados.

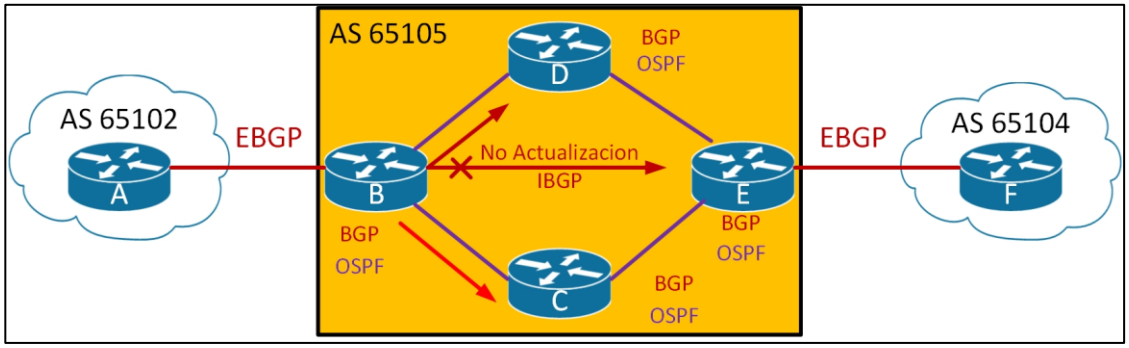
Si el vecino emisor de IBGP no está directamente conectado con los demás enrutadores IBGP, los *router* que no están viéndose con este router tienen diferentes tablas de enrutamiento IP de los routers que lo están viendo. Las tablas de enrutamiento inconsistentes pueden causar bucles de enrutamiento u hoyos negros de enrutamiento, porque la suposición predeterminada por todos los enrutadores que ejecutan BGP dentro de un sistema autónomo es que cada enrutador BGP está intercambiando información IBGP directamente con todos los otros enrutadores BGP en el sistema autónomo.

Si todos los vecinos IBGP están completamente engranados, cuando se recibe un cambio de un sistema autónomo externo, el enrutador BGP para el sistema autónomo local es responsable de informar a todos los demás vecinos de IBGP sobre el cambio. Los vecinos de IBGP que reciben esta actualización no la envían a ningún otro vecino IBGP, ya que suponen que el vecino emisor de IBGP está totalmente vinculado con todos los demás hablantes de IBGP y ha enviado a cada IBGP vecino la actualización.

- Ejemplo: IBGP en malla parcial

La figura 14 muestra el comportamiento de actualización de IBGP en un entorno vecino parcialmente mallado.

Figura 14. **Malla parcial BGP**



Fuente: elaboración propia, empleando Visio 2013.

El enrutador B recibe una actualización BGP del enrutador A. El enrutador B tiene dos vecinos IBGP, enrutadores C y D, pero no tiene una relación de vecino IBGP con el enrutador E. los enrutadores C y D aprenden acerca de las redes que se agregan o retiran detrás del enrutador B. Incluso si los enrutadores C y D tienen sesiones contiguas IBGP con el enrutador E, suponen que el sistema autónomo está completamente mallado para IBGP y no replican la actualización por lo que no la envían al enrutador E.

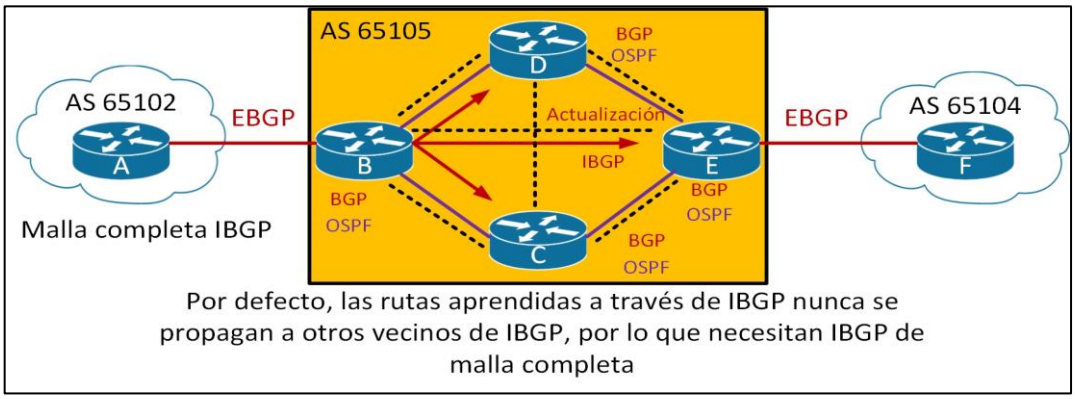
Enviar una actualización de IBGP al enrutador E es responsabilidad del enrutador B, ya que es el enrutador con conocimiento de primera mano de las redes en y más allá de AS 65105. El enrutador E no conoce ninguna red a través del enrutador B y no usa enrutador B para llegar a cualquier red en AS 65105 u otros sistemas autónomos detrás de AS 65015.

- Ejemplo: IBGP full mesh

La figura 15 muestra un IBGP completamente mallado. Cuando el enrutador B recibe una actualización del enrutador A, actualiza sus tres pares

IBGP: los enrutadores C, D y E. El IGP enruta el segmento TCP que contienen la actualización BGP del enrutador A al enrutador E porque los enrutadores no están conectados directamente. La actualización se envía una vez a cada vecino y no esta duplicada por ningún otro vecino de IBGP, lo que reduce el tráfico innecesario. En IBGP completamente mallado, cada enrutador supone que cada otro enrutador interno tiene una declaración de vecino que apunta a cada vecino de IBGP.

Figura 15. **Malla completa BGP**



Fuente: elaboración propia, empleando Visio 2013.

- TCP y malla completa

Se seleccionó TCP como la capa de transporte para BGP porque puede mover grandes volúmenes de datos de manera confiable. Con la tabla de enrutamiento de internet muy grande en constante cambio, se determinó que TCP era la mejor solución para ventanas y confiabilidad, en lugar de desarrollar una capacidad de ventana BGP uno por uno como OSPF o EIGRP.

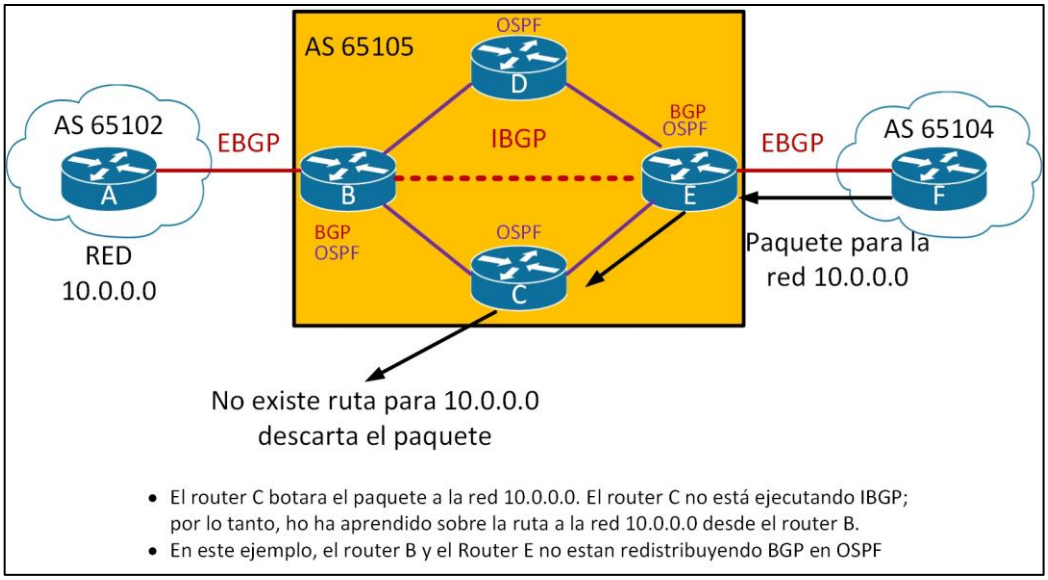
Debido a que cada enrutador IBGP necesita enviar rutas a todos los demás vecinos IBGP en el mismo sistema autónomo (para que todos tengan una imagen completa de las rutas enviadas al sistema autónomo), deben usar sesiones BGP (TCP) totalmente malladas. Recuerdese, BGP usa TCP para garantizar la entrega confiable de paquetes, por lo tanto, no pueden transmitir o multidifundir sus rutas a otros vecinos IBGP.

Cuando todos los enrutadores que ejecutan BGP en un sistema autónomo están completamente mallados y tienen la misma base de datos como resultado de una política de enrutamiento coherente, pueden aplicar la misma fórmula de selección de ruta. Los resultados de la selección de ruta son, por lo tanto, uniformes en todo el sistema autónomo, lo que garantiza que no haya bucles de enrutamiento y una política coherente para salir y entrar en el sistema autónomo.

3.2.6. Problemas de enrutamiento en un sistema autónomo de tránsito

Todos los enrutadores en la ruta entre los vecinos de IBGP, conocidos como la ruta de tránsito, también deben ejecutar BGP, como se ilustra en la figura 16. En este ejemplo, lo enrutadores A, B, E y F son los únicos que ejecutan BGP. El enrutador B tiene una declaración vecina EBGP para el enrutador A y una declaración vecina IBGP para el enrutador E. el enrutador E tiene una declaración vecina EBGP para el enrutador F y una declaración vecina IBGP para enrutador B. los enrutadores C y D no ejecutan BGP. Los enrutadores B, C, D y E están ejecutando OSPF como su IGP.

Figura 16. **Problemas de enrutamiento si BGP no está activado en todos los routers en la ruta de transito**



Fuente: elaboración propia, empleando Visio 2013.

La red 10.0.0.0 es propiedad de AS 65101 y se anuncia al enrutador B a través de una sesión EBGP. El enrutador B lo anuncia al enrutador E a través de una sesión IBGP. Los enrutadores C y D nunca aprenden acerca de esta red, porque no se redistribuye en el protocolo de enrutamiento local (OSPF), y los enrutadores C y D no ejecutan BGP. Si el enrutador E anuncia esta red al enrutador F en AS 65103, y el enrutador F comienza a reenviar paquetes a la red 10.0.0.0 a través de AS 65102, el enrutador E enviaría los paquetes a su par BGP, enrutador B. Sin embargo, para llegar al enrutador B, los paquetes deben pasar por el enrutador C o D, pero estos enrutadores no tienen una entrada en sus tablas de enrutamiento para la red 10.0.0.0.; por lo tanto, cuando el enrutador E envía paquetes con una dirección de destino en la red 10.0.0.0 al enrutador C o D, esos enrutadores descartan los paquetes.

Incluso si los enrutadores C y D tienen una ruta predeterminada que apunta a los puntos de salida del sistema autónomo (enrutadores B y E); es muy probable que cuando el enrutador E envié un paquete para la red 10.0.0.0 al enrutador C o D, aquellos enrutadores pueden enviarlo de vuelta al enrutador E, que lo reenvía al enrutador C o D, causando un bucle de enrutamiento. Para resolver este problema, BGP debe implementarse en los enrutadores C y D. En otras palabras, todos los enrutadores en la ruta de tránsito dentro del sistema autónomo deben ejecutar BGP, y las secciones de IBGP deben estar completamente malladas.

3.3. Configuración BGP

En la configuración de BGP se utilizan una diversidad de comandos en los routers por lo cual, se tienen que tomar en cuenta la diversidad de funcionalidades que puede tener este protocolo, los cuales se explican a continuación.

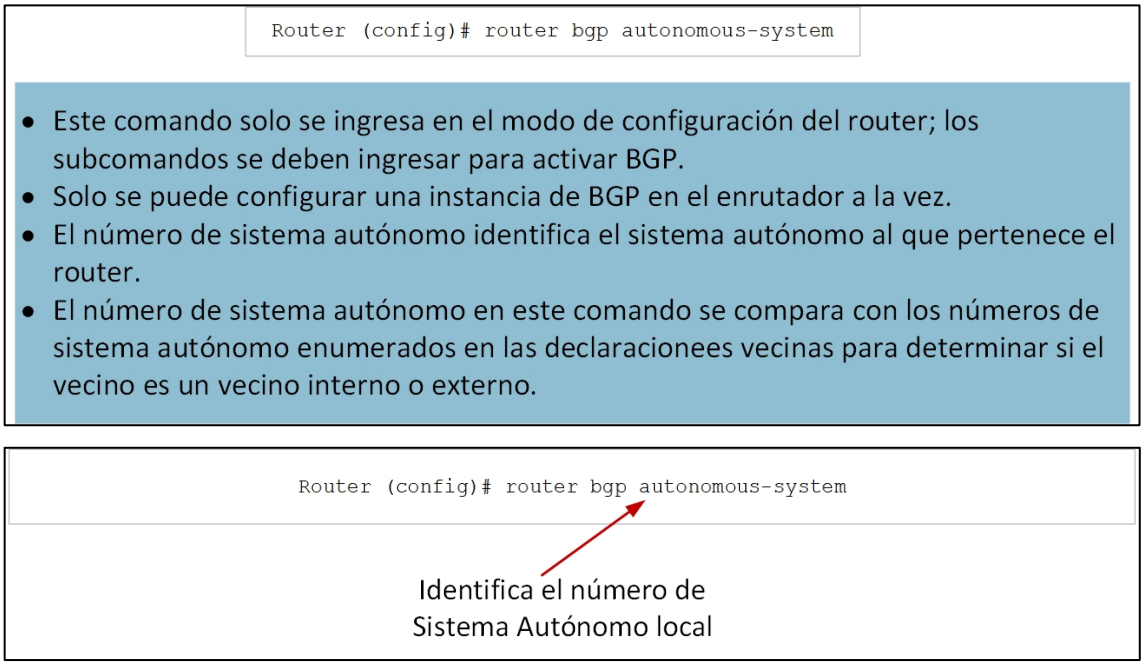
3.3.1. Configuración básica BGP

La sintaxis de los comandos básicos de configuración de BGP es similar a la sintaxis para configurar los protocolos de enrutamiento interno. Sin embargo, existen diferencias significativas en cómo funciona BGP.

Use el comando 'router bgp autonomous-system' para identificar al enrutador que cualquier subcomando subsiguiente pertenece a este proceso de enrutamiento (figura 17). Este comando también identifica el sistema autónomo local al que pertenece este enrutador. El enrutador debe estar informado del sistema autónomo para que pueda determinar si los vecinos BGP que se

configuran a continuación son vecinos IBGP o EBGP. La figura 17 muestra el parámetro para el comando router bgp.

Figura 17. **Comandos BGP**



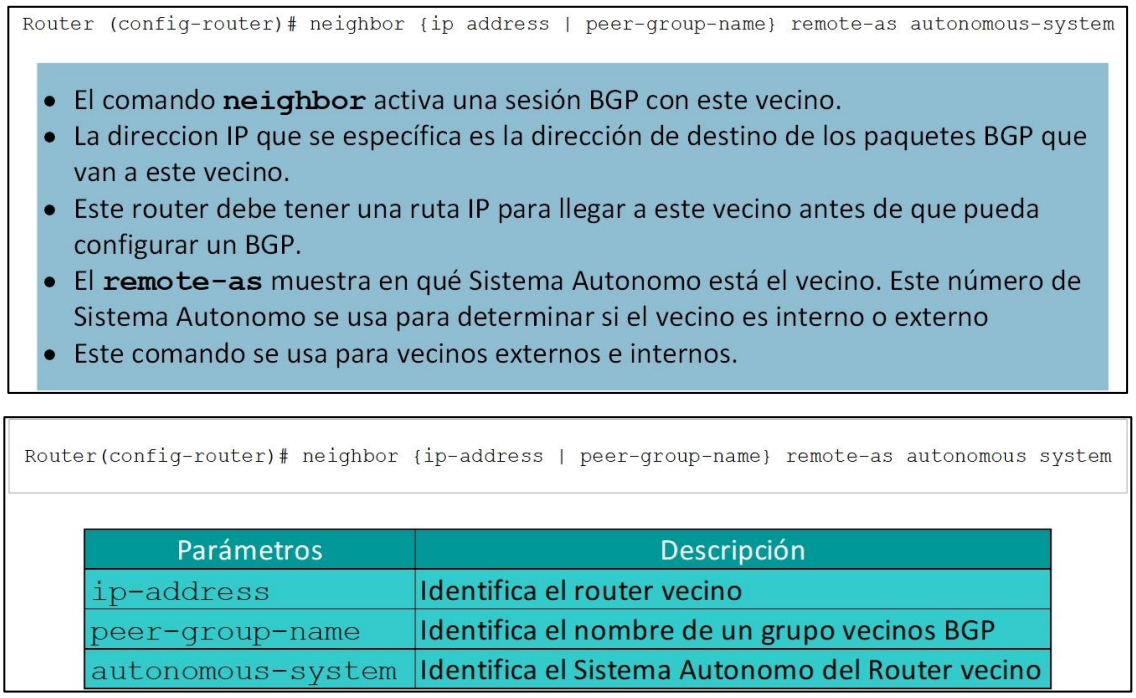
Fuente: elaboración propia, empleando Visio 2013.

El comando router bgp solo no activa BGP en un enrutador. Debe ingresar al menos un subcomando para activar el proceso BGP.

3.3.2. Activación de una sesión BGP

Utilice el comando neighbor ip-address remote-as autonomous-system para activar una sesión BGP para enrutadores vecinos externos e internos. Este comando identifica un enrutador con el que el enrutador local establece una sesión. La figura muestra los parámetros para este comando, figura 19.

Figura 18. **Parametros BGP**



Fuente: elaboración propia, empleando Visio 2013.

Un grupo de iguales es un grupo de vecinos BGP que tienen todas las mismas políticas de actualización. Los grupos de pares se describen más adelante en esta sección.

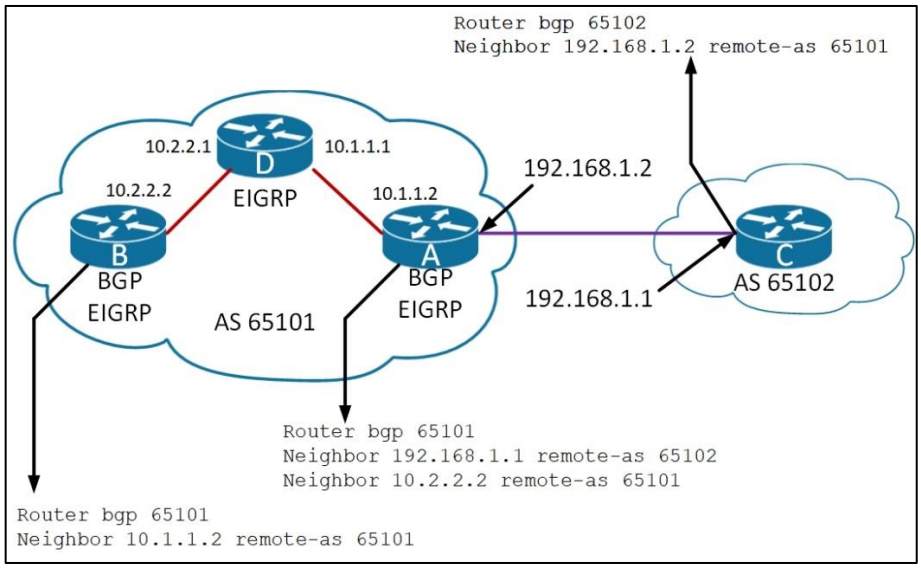
La dirección es la dirección de destino para todos los paquetes BGP que van a este enrutador vecino. La dirección debe ser accesible, porque BGP intenta establecer una sesión TCP e intercambiar las actualizaciones BGP con el dispositivo en esta dirección IP.

El número del sistema autónomo identifica si este vecino es un vecino EBGP o IBGP. Si el número es el mismo que el número de sistema autónomo para este enrutador, este vecino es un vecino IBGP, y la dirección IP

enumerada en el comando vecino no tiene que estar conectada directamente. Si el número es diferente, el vecino es un EBGP y la dirección en el comando vecino debe estar conectada directamente por defecto.

En la figura 19, el enrutador A en AS 65101 tiene dos declaraciones vecinas. El enrutador A sabe que el enrutador C (vecino 192.168.1.1 AS remoto 65102) es un vecino externo, porque tiene a el AS 65102 en la declaración vecina para enrutador. C no coincide con el número de sistema autónomo del enrutador A, que es AS 65101. El enrutador A puede llegar a AS 65102 a través de 192.168.1.1, que está conectado directamente al enrutador A.

Figura 19. ***Neighbor remote-as* commando y parametros BGP**



Fuente: elaboración propia, empleando Visio 2013.

El vecino 10.2.2.2 (enrutador B) está en el mismo sistema autónomo que el enrutador A. E el segundo enunciado en router A define al router B como un vecino de IBGP.

AS 65101 ejecuta EIGRP entre todos los enrutadores internos. El enrutador A tiene una ruta EIGRP para llegar a la dirección IP 10.2.2.2. Como vecino de IBGP, el enrutador B puede estar en enrutadores múltiples lejos del enrutador A.

3.3.3. Apagando un vecino BGP

Utilice el comando neighbor ip-address shutdown para cerrar administrativamente y volver habilitar un vecino BGP, figura 20.

Figura 20. **Ejemplo del comando BGP neighbor**

Fuente: elaboración propia, empleando Visio 2013.

Si implementa cambios o políticas importantes en un *router* vecino y cambia varios parámetros, debe cerrar administrativamente el *router* vecino, implementar los cambios luego hacer que el *router* vecino, haga una copia de seguridad con el comando *no neighbor ip-address shutdown*.

3.3.4. Consideraciones de configuración de BGP

La declaración del vecino BGP informa al router de la dirección IP de destino para cada paquete de actualización. El router debe decidir qué dirección IP usar como dirección IP de origen en la actualización de enrutamiento BGP.

- Al crear un paquete BGP, la declaración vecina define las direcciones IP de destino, y la interfaz saliente define la dirección IP de origen.

- Cuando se recibe un paquete BGP para una nueva sesión BGP, la dirección de origen del paquete es una comparación con la lista de declaraciones vecinas.

    - Si se encuentra una coincidencia, se establece la relación

    - Si no se encuentra ninguna coincidencia, se ignora el paquete

- Asegurarse de que la dirección IP origen coincida con la dirección que el otro en su declaración vecina.

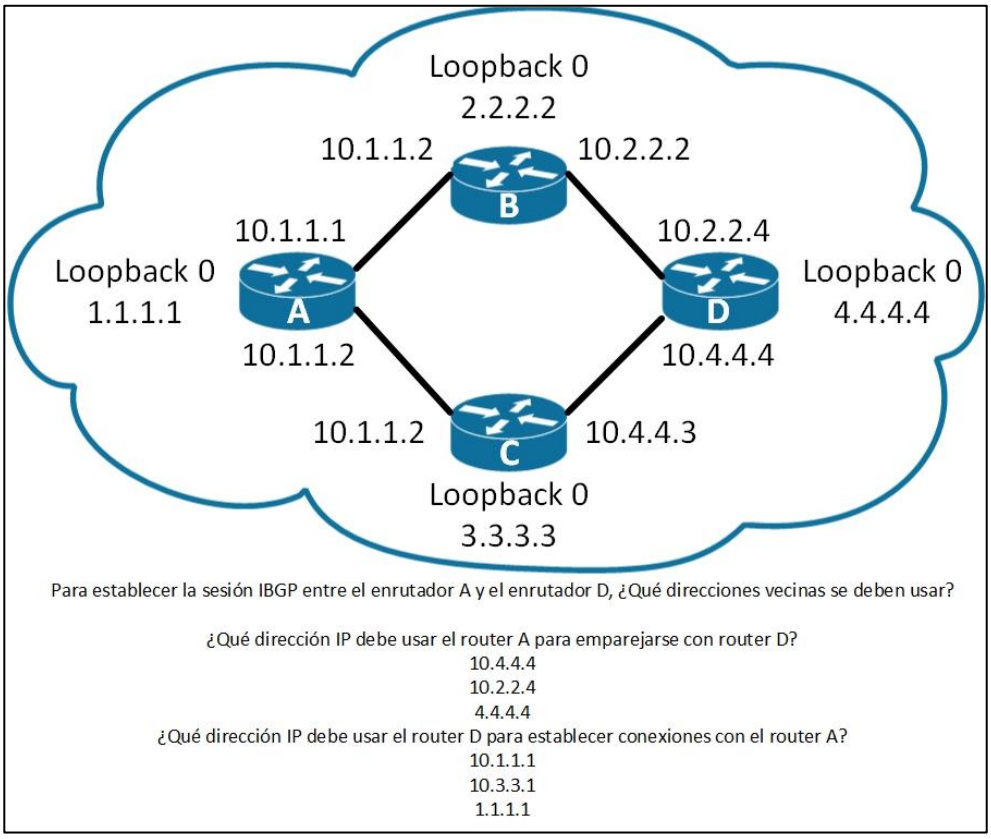
Cuando un *router* crea un paquete BGP para un vecino, verifica que la tabla de enrutamiento de la red de destino llegue a ese vecino. la dirección IP de la interfaz de salida, como indica la tabla de enrutamiento, se utiliza como la dirección IP de origen del paquete BGP.

Esta dirección IP de origen debe coincidir con la dirección en la declaración vecina correspondiente en el otro enrutador. De lo contrario, los enrutadores no serán pares BGP porque no pueden establecer una sesión BGP.

3.3.5. Problema de emparejamiento IBGP

Para establecer la sesión de IBGP entre los enrutadores A y D, como se muestra en la figura 21, ¿qué dirección IP vecina debería de utilizarse?

Figura 21. **Problema de peering IBGP**



Fuente: elaboración propia, empleando Visio 2013.

El problema es el siguiente. Si el enrutador D usa el vecino 10.3.3.1 remoto, como 65102, pero el *router* A está enviando los paquetes BGP al enrutador D a través del *router* B, la dirección IP origen es 10.1.1.1.

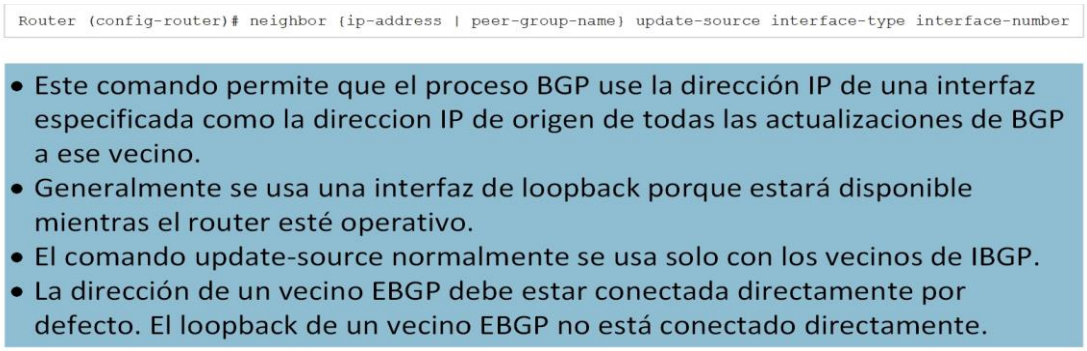
Cuando el *router* D recibe este paquete BGP a través del *router* B, no reconoce este paquete BGP porque 10.1.1.1. no se configuró como un vecino del *router* D. Por lo tanto, la sesión IBGP entre los router A y D no se puede establecer.

Una solución es establecer una sesión de IBGP utilizando una interfaz de *loopback* cuando haya múltiples rutas entre los vecinos IBGP.

3.3.6. Comando *neighbor update-source* en BGP

La opción 'update-source' del comando 'neighbor' anula la dirección IP de origen predeterminada utilizada para los paquetes BGP. Es necesario indicar al enrutador qué dirección IP utilizar como dirección de origen para todos los paquetes BGP si desea utilizar una interfaz de *loopback* en lugar de la interfaz física, figura 22.

Figura 22. **Comando BGP *neighbor update-source***



Fuente: elaboración propia, empleando Visio 2013.

Si no usa la opción update-source, un anuncio dirigido a un vecino usa la dirección IP de la interfaz que sale como la dirección de origen de un paquete.

Cuando un *router* crea un paquete, ya sea una actualización de enrutamiento, un ping o cualquier otro tipo de paquete IP, el enrutador busca la dirección de destino en la tabla de enrutamiento. La tabla de enrutamiento enumera la interfaz adecuada para llegar a la dirección de destino. La dirección de esta interfaz de salida se utiliza como la dirección de origen de este paquete de forma predeterminada.

Considere lo que sucedería si un *router* vecino usa la dirección de *loopback* en su comando *neighbor* para este *router*, pero el otro *router* vecino no usa el comando *neighbor update-source.* Cuando el enrutador vecino recibe un paquete de actualización y mira la dirección de origen del paquete, ve que no tiene una relación de vecino con esa dirección de origen, por lo que descarta el paquete.

BGP no acepta actualizaciones no solicitadas. Debe conocer todos los *routers* vecinos y tener un enunciado vecino.

Pueden existir múltiples rutas para llegar a cada vecino cuando se mira con los routers vecinos IBGP. Si el enrutador BGP está utilizando una dirección vecina que está asignada a una interfaz específica en otro router, y esa interfaz falla, el enrutador que apunta a esta dirección pierde su sesión BGP con ese vecino.

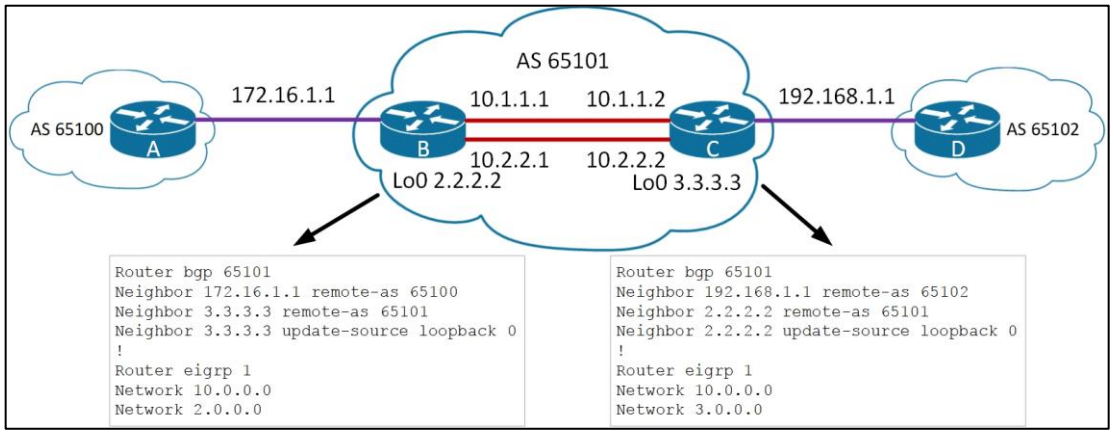
Si el router iguala con la interfaz de *loopback* en lugar del otro router, la interfaz de loopback siempre estará disponible siempre que el enrutador no falle. Este arreglo de emparejamiento agrega elasticidad a las sesiones de IBGP porque los enrutadores no están atados a una interfaz física, que puede bajar por cualquier cantidad de razones.

Para observar el *loopback* de otro vecino interno, el primer router apunta a la instrucción neighbor en la dirección *loopback* del otro vecino interno. Asegúrese de que ambos *routers* tengan una ruta a la dirección de *loopback* del otro vecino en su tabla de enrutamiento. También, asegúrese de que ambos routers están anunciando sus direcciones de loopback en su protocolo de enrutamiento local.

- Ejemplo: uso de direcciones de loopback con BGP

En la figura 23, el *router* B tiene el *router* A como vecino EBGP. La única dirección accesible para el enrutador B se usa para una dirección vecina en BGP es la dirección directamente conectada de 172.16.1.1. El *router* B tiene varias rutas para llegar al *router* C, un vecino de IBGP.

Figura 23. **BGP utilizando direcciones de loopback**



Fuente: elaboración propia, empleando Visio 2013.

Todas las redes, incluida la red IP para la interfaz de *loopback* del *router* C, se pueden alcanzar desde el *router* B. El *router* B puede llegar a estas redes

porque los *routers* B y C intercambian las actualizaciones de EIGRP; los enrutadores B y A no intercambian actualizaciones de EIGRP.

La relación de vecinos entre los routers B y C no está vinculada a una interfaz física porque el *router* B está a la par con la interfaz de *loopback* en el router C y utiliza su dirección de *loopback* como dirección IP de origen, y viceversa. si el *router* B en cambio utilizara la interfaz 10.1.1.2 en el router C y esa interfaz cayese, la relación de vecino BGP también terminaría.

El comando *neighbor update-source* debe usarse en ambos routers. Si el router B apunta a la dirección de *loopback* 3.3.3.3 del *router* C, y el *router* C apunta a la dirección de *loopback* 2.2.2.2 del router B, y ninguno utiliza el comando *neighbor update-source*, la sesión BGP entre estos *routers* no se inicia.

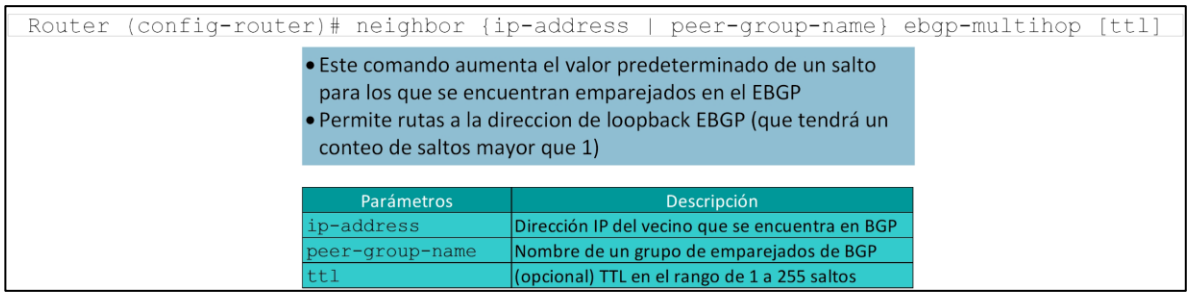
El *router* B enviaría un paquete BGP abierto al *router* C con la dirección IP de origen como 10.1.1.1 o 10.2.2.1. El *router* C revisará la dirección IP de origen e intentará compararla con su lista de vecinos conocidos. El *router* C no encontraría una coincidencia y no respondería al mensaje abierto del *router* B.

3.3.7. Problemas de emparejamiento EBGP

Cuando un *router* EBGP está analizando con un vecino externo, la única dirección a la que puede acceder sin configuración adicional es la interfaz que está conectada directamente con ese enrutador EBGP. Recuerde que la información de enrutamiento interno no se intercambia con compañeros externos. Por lo tanto, el *router* debe apuntar a una dirección directamente conectada para ese vecino externo.

Si se usa una interfaz de *loopback* en lugar de la interfaz que está directamente conectada, se requiere configuración adicional. Para permitir que el *router* acepte e intente conexiones BGP a pares externos que residen en redes que no están conectadas directamente, se debe configurar el comando de configuración del *router* vecino 'ip-address ebgp-multihop [ttl]'. La figura 24 muestra los parámetros del comando.

Figura 24. **Comando y parámetros de neighbor ebgp-multihop en BGP**



Fuente: elaboración propia, empleando Visio 2013.

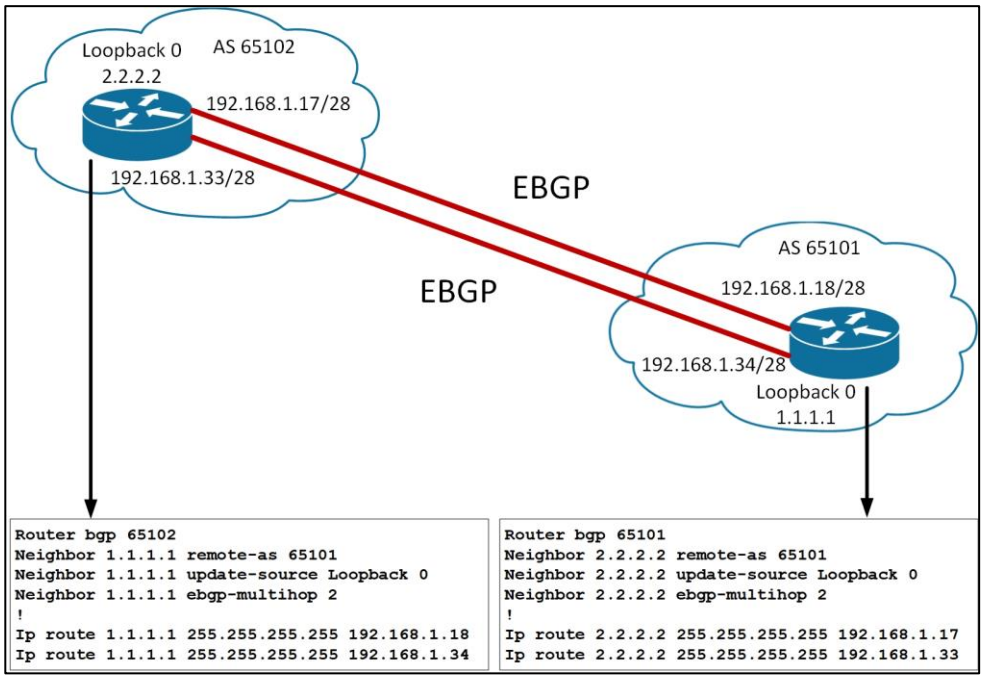
Los emparejamientos EBGP generalmente están a solo un salto uno del otro. El comando neighbor ebgp-multihop aumenta el valor de salto predeterminado para permitir rutas a la dirección de loopback EBGP con un valor TTL mayor que 1. Este comando es valioso cuando existen rutas redundantes entre los vecinos EBGP.

- Ejemplo: comando EBGP-multihop

En la figura 25, el *router* A en el sistema autónomo 65102 tiene dos rutas al router B en el Sistema Autónomo 65101. Si el *router* A usa una declaración de un solo vecino y apunta a 192.168.1.18 en el *router* B de sistema autónomo 65101 y ese enlace falla, no hay sesión de BGP entre estos sistemas

autónomos. Como resultado ningún paquete pasa de un Sistema Autónomo a otro, aunque existe otro enlace. Si el enrutador A en su lugar usa dos declaraciones vecinas apuntando a 192.168.1.18 y 192.168.1.34 en el *router* B, resuelve parcialmente el problema. Sin embargo, cada actualización de BGP que recibe el *router* A se envía al *router* B dos veces porque hay dos declaraciones vecinas.

Figura 25. **Ejemplo del comando EBGP-multihop**



Fuente: elaboración propia, empleando Visio 2013.

Como se muestra en la figura 25, el *router* A apunta a la dirección de *loopback* del *router* B y viceversa, y cada *router* usa su dirección de *loopback* como la dirección IP de origen para sus actualizaciones BGP. Debido a que un IGP no se usa entre sistemas autónomos, ninguno de los *routers* puede alcanzar el *loopback* del otro *router* sin asistencia.

Cada *router* necesita dos rutas estáticas para informar a BGP de las rutas disponibles para llegar a la dirección de *loopback* del otro *router*. Una dirección de vecino EBGP debe estar conectada directamente por defecto. El comando *neighbor ebgp-multihop* se debe usar para cambiar la configuración predeterminada de BGP e informar a BGP que esta dirección IP vecina está a más de un salto de distancia. En la figura 25, el comando utilizado en el *router* A informa a BGP que la dirección del vecino 1.1.1.1 está a dos pasos de distancia.

BGP no está diseñado para realizar el equilibrio de cargas. Las rutas se eligen debido a políticas, no basadas en el ancho de banda. BGP elige solo una mejor ruta. El uso de las direcciones de *loopback* y el comando *neighbor ebgp-multihop* como se muestra en la figura 25 permite el equilibrio de carga y la redundancia en las dos rutas entre los sistemas autónomos.

3.3.8. Comportamiento del siguiente salto

La forma en que BGP establece una relación IBGP es muy diferente de la forma en que se comportan los IGP. El método que usa BGP para denotar su dirección del siguiente salto también es muy diferente.

BGP informa al próximo sistema autónomo sobre las rutas a otros sistemas autónomos y las redes que poseen esos otros sistemas autónomos. BGP, al igual que IGP, es un protocolo de enrutamiento salto a salto. Sin embargo, a diferencia de los IGP, BGP enruta desde el sistema autónomo al sistema autónomo, y el siguiente salto predeterminado es el próximo sistema autónomo. Un *router* vecino de IBGP que aprende sobre una red fuera de sus sistemas autónomos ve, como la dirección del siguiente salto, el punto de

entrada para los próximos sistemas autónomos a lo largo de la ruta para llegar a la red distante.

- BGP es un protocolo de enrutamiento de sistema autónomo a sistema autónomo, no un protocolo de enrutamiento de *router* a *router*.

- En BGP, el próximo salto no significa el siguiente enrutador; significa la dirección IP para llegar al siguiente sistema autónomo.

- Para EBGP, el siguiente salto predeterminado es la dirección IP del router vecino que envió la actualización.

- Para IBGP, el protocolo BGP establece que el próximo salto publicado por EBGP debe llevarse a IBGP.

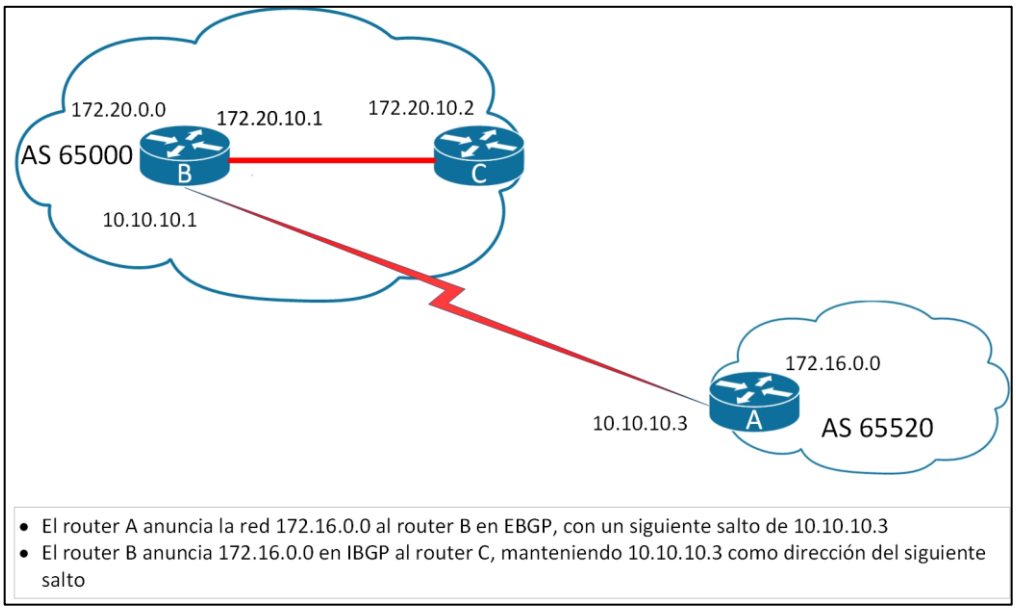
Para EBGP, el siguiente salto predeterminado es la dirección IP del enrutador vecino que envió la actualización.

Para IBGP, el protocolo BGP establece que el próximo salto publicitado por EBGP se debe llevar a IBGP.

- Ejemplo del comportamiento del siguiente salto

En la figura 26, el *router* A anuncia 172.16.0.0 al *router* B con un siguiente salto de 10.10.10.3. El *router* B anuncia 172.20.0.0 al enrutador A con un siguiente salto de 10.10.10.1.

Figura 26. **Ejemplo del comportamiento del siguiente salto**



Fuente: elaboración propia, empleando Visio 2013.

Para IBGP, el protocolo BGP establece el próximo salto publicado por EBGP se debe llevar a IBGP. Debido a esta regla, el *router* B anuncia 172.16.0.0 a su *router* de emparejados de IBGP C con un siguiente salto de 10.10.10.3, la dirección del router A. El *router* C sabe que el siguiente salto para llegar a 172.16.0.0 es 10.10.10.3, no 172.20.10.1, como era de esperar.

Por lo tanto, es muy importante que el *router* C sepa cómo llegar a la subred 10.10.10.0, ya sea a través de un IGP o una ruta estática. De lo contrario, el router C descarta los paquetes destinados a 172.16.0.0 porque no puede acceder a la dirección del siguiente salto para esa red.

Un *router* vecino IBGP realiza una búsqueda recursiva para averiguar cómo llegar a una dirección BGP de siguiente salto utilizando sus entradas IGP

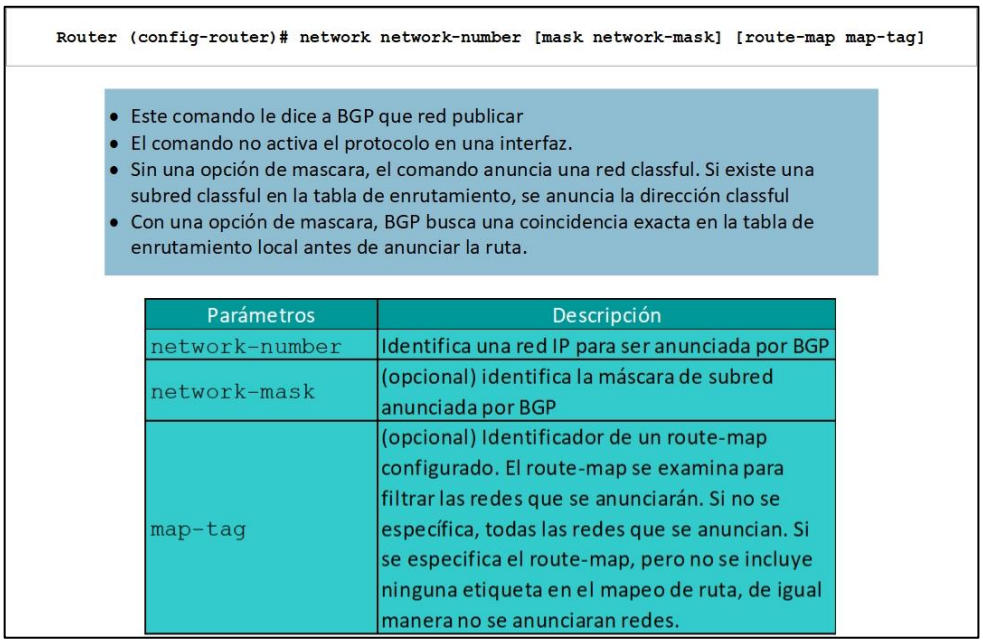
en la tabla de enrutamiento. Por ejemplo, el *router* C aprende en una actualización de BGP sobre la red 172.16.0.0/16 desde un origen de ruta de 172.20.10.1 (*router* B), con un siguiente salto de 10.10.10.3 (*router* A). El *router* C instala la ruta a 172.16.0.0/16 en la tabla de enrutamiento con un siguiente salto de 10.10.10.3. El *router* B debe anunciar la red 10.10.10.0/24 usando su IGP al *router* C para que el router C pueda instalar esa ruta en su tabla de enrutamiento con un siguiente salto de 172.20.10.1.

Un IGP usa la dirección IP de origen de una actualización de enrutamiento (fuente de ruta) como la dirección del siguiente salto, mientras que BGP usa un campo separado por red para registrar la dirección del siguiente salto. Si el *router* C tiene un paquete para enviar a 172.16.100.1, busca la red en la tabla de enrutamiento y encuentra una entrada BGP, el *router* C completa la búsqueda recursiva en la tabla de enrutamiento para una ruta a la red 10.10.10.3. El IGP ha colocado una ruta a la red 10.10.10.0 en la tabla de enrutamiento con un siguiente salto de 172.20.10.1, por lo que el *router* C reenvía el paquete destinado a 172.16.100.1. a 172.20.10.1.

3.3.9. Inyección de rutas BGP

Utilice el comando *network network-number* para permitir que BGP anuncie una red si está presente en la tabla de enrutamiento IP. La figura 28 muestra los parámetros del comando.

Figura 27. **Comando y parámetros de *network* en BGP**



Fuente: elaboración propia, empleando Visio 2013.

El comando *network* determina qué redes origina el router. Este concepto es diferente de usar el comando *network* cuando está configurado un IGP. A diferencia de un IGP, el comando *network* no inicia BGP en interfaces específicas. En cambio, indica a BGP qué redes debería originar desde este *router*.

El parámetro de máscara indica que BGP4 puede manejar subredes y superredes. La lista de comandos de *network* debe incluir todas las redes en su sistema autónomo que desee publicar, no solo aquellas que están conectadas localmente al *router*.

El comando *neighbor* le dice a BGP dónde anunciar, y el comando *network* le dice a BGP qué publicar.

3.3.10. Comando *network* en BGP

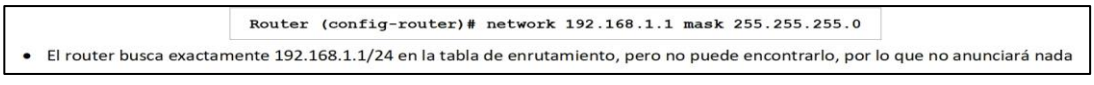
El único propósito del comando *network* es notificar a BGP qué red se va a publicar. Sin la opción de máscara, este comando anuncia sólo el número de red con clase. Al menos una subred de la red principal especificada debe estar presente en la tabla de enrutamiento de IP para permitir que BGP comience a anunciar la red con clase como una ruta BGP.

Cuando se especifica una opción *network-mask*, debe existir una coincidencia exacta con la red (tanto la dirección como la máscara) en la tabla de enrutamiento antes de que BGP anuncie las rutas. BGP comprueba si puede alcanzarlo antes de que comience a anunciar la red como una ruta BGP.

Los siguientes son dos ejemplos de cómo el comando *network network-mask* puede estar mal configurado.

En la figura 28, el comando *network* 192.168.1.1 mask 255.255.255.0 hace que BGP compruebe la ruta específica 192.168.1.0/24 en la tabla de enrutamiento. Puede encontrar 192.168.1.0/29 o 192.168.1.1/32. Sin embargo, si nunca encuentra una coincidencia específica para la red 192.168.1.1/24, BGP no anuncia la red 192.168.1.1/24 a ningún vecino.

Figura 28. **Ejemplo del comando *network* en BGP**

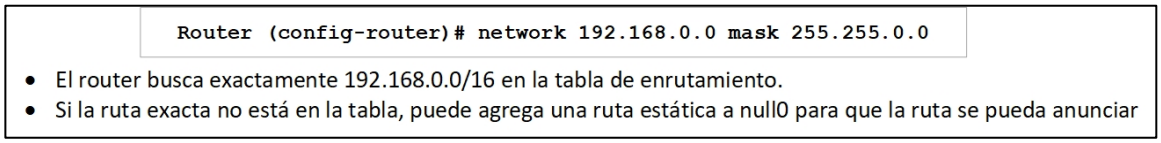


Fuente: elaboración propia, empleando Visio 2013.

En la figura 29, el comando *network* 192.168.0.0 *mask* 255.255.0.0 anuncia un bloque CIDR. Por lo tanto, BGP busca 192.168.0.0/16 en la tabla de enrutamiento. Puede encontrar 192.168.1.0/24 o 192.168.1.1/32. Si BGP nunca encuentra 192.168.0.0/16, no anuncia la red 192.168.0.0/16 a ningún vecino. En este caso, puede configurar la siguiente ruta estática hacia la interfaz nula, por lo que BGP puede encontrar una coincidencia exacta en la tabla de enrutamiento.

ip route 192.168.0.0 255.255.0.0 null0

Figura 29. **Ejemplo del comando *network* (contenido) en BGP**



Fuente: elaboración propia, empleando Visio 2013.

Después de encontrar una coincidencia exacta en la tabla de enrutamiento, BGP anuncia la red 192.168.0.0/16 a cualquier vecino.

El comando de configuración de BGP *auto-summary* en el *router* determina cómo BGP maneja las rutas redistribuidas. Cuando la sumarización BGP está habilitada (con *auto-summary*), todas las subredes redistribuidas se resumen a sus límites con clase en la tabla BGP. Cuando esta deshabilitado (con no 'auto-summary'); todas las subredes redistribuidas están presentes en su forma original en la tabla BGP, por lo que solo se anuncian las subredes.

3.4. Seleccionando el camino BGP

Dentro de las funciones que puede ejecutar este protocolo, tiene como prioridad las rutas en la forma de escogerlas, ya que tienen que cubrir distintas características las cuales serán explicadas cada una de ellas.

3.4.1. Características y atributos de BGP

Los enrutadores BGP envían mensajes de actualización de BGP sobre redes de destino a otros enrutadores BGP. Los mensajes de actualización contienen una o más rutas y un conjunto de métricas BGP, que se llaman atributos de ruta, adjuntos a las rutas.

- Las métricas de BGP se llaman atributos de ruta
- Las características de los atributos de ruta incluyen:

    - Bien conocido versus opcional
    - Obligatorio versus discrecional
    - Transitivo versus no transitivo
    - Parcial

Un atributo es bien conocido u opcional, obligatorio o discrecional, y transitivo o no transitivo. Un atributo también puede ser parcial.

No todas las combinaciones de estas características son válidas. los atributos de ruta se incluyen en las siguientes cuatro categorías:

- Conocido obligatorio
- Conocido discrecional

- Transitivo opcional
- Opcional no transitivo

Solo los atributos transitivos opcionales se pueden marcar como parciales. Todos los routers BGP deben reconocer un atributo conocido y propagarlo a otros vecinos BGP.

- Atributos bien conocidos
  - Debe ser reconocido por todas las implementaciones BGP compatibles.
  - Se propagan a otros vecinos.

- Atributos obligatorios bien conocidos
  - Debe estar presente en todos los mensajes de actualización.

- Atributos discrecionales bien conocidos
  - Puede estar presente en los mensajes de actualización.

Los atributos bien conocidos son obligatorios o discrecionales. Un atributo obligatorio bien conocido debe estar presente en todas las actualizaciones de BGP. Un atributo discrecional bien conocido no tiene que estar presente en todas las actualizaciones de BGP.

Los atributos que son conocidos se llaman opcionales. Los enrutadores BGP no tienen que admitir un atributo opcional. Los atributos opcionales son transitivos o no transitivos.

- Atributos opcionales
  - Reconocido por algunas implementaciones (podría ser privado); esperado no ser reconocido por todos los enrutadores.
  - Los atributos opcionales reconocidos se propagan a otros vecinos en función de su significado.

- Atributos transitivos opcionales
  - si no se reconoce, se marcan como parciales y se propagan a otros vecinos.

- Atributos opcionales no transitivos

  - Descartado si no es reconocido.

Las siguientes afirmaciones se aplican a los atributos opcionales:

- Los routers BGP que implementan el atributo opcional pueden propagarlo a los otros vecinos BGP, en función de su significado.

- Los routers BGP que no implementan un atributo transitivo opcional deben pasarlo a otros routers BGP intactos y marcar el atributo como parcial.

- Los routers BGP que no implementan un atributo opcional no transitorio deben eliminar el atributo y no deben pasarlo a otros routers BGP.

3.4.2. Atributos BGP

La siguiente es una lista de los atributos comunes de BGP según las categorías a las que pertenecen:

- Atributos obligatorios bien conocidos
  - Trayectoria del sistema autónomo
  - Siguiente salto
  - Origen

- Atributos discrecionales bien conocidos
  - Preferencia local
  - Agregado atómico

- Atributo transitivo opcional
  - Agregador

- Atributo no transitorio opcional
  - Discriminador de salida múltiple (MED)

- Los atributos de BGP incluyen los siguientes:
  - Trayectoria del sistema autónomo (atributo obligatorio conocido)
  - Próximo salto (atributo obligatorio bien conocido)
  - Origen (atributo obligatorio bien conocido)
  - Preferencia local
  - MED
  - Otros

Además, Cisco define un atributo de peso para BGP. El peso se configura localmente en un router y no se propaga a ningún otro router BGP.

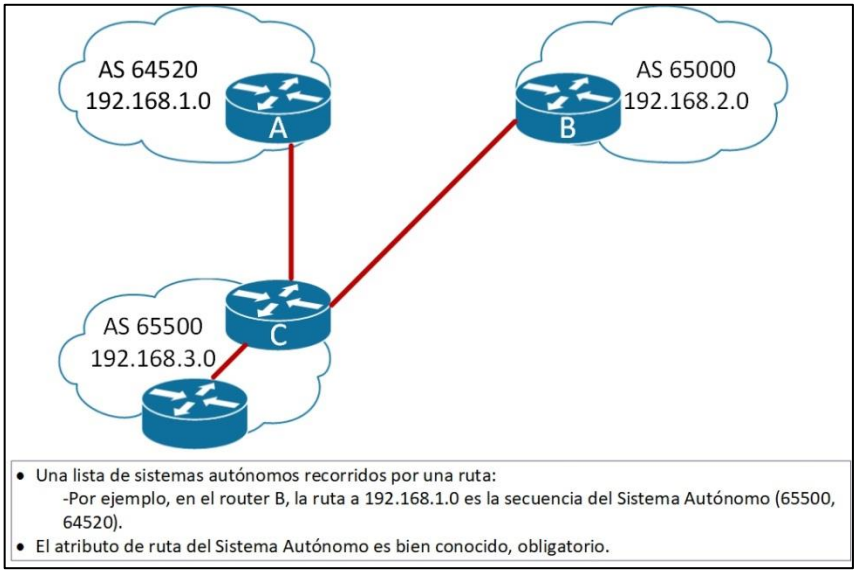
3.4.3. Atributo de ruta de un sistema autónomo

La ruta de un sistema autónomo es un atributo obligatorio bien conocido. Cuando una actualización de ruta pasa a través de un sistema autónomo, el número del sistema autónomo se antepone (agrega) a esta actualización cuando se anuncia al siguiente vecino EBGP.

El atributo de ruta del sistema autónomo es en realidad la lista de números de sistema autónomo que una ruta ha atravesado para llegar a un destino, con el número del sistema autónomo que originó la ruta al final de la lista.

En la figura 30, el *router* A en el sistema autónomo 64520 anuncia la red 192.168.1.0. Cuando esa ruta atraviesa el sistema autónomo 65500, el *router* C lo antepone a su propio número de sistema autónomo. Cuando 192.168.1.0 llega al *router* B, tiene dos números de sistema autónomo conectados. Desde la perspectiva del *router* B, la ruta para llegar a 192.168.1.0 es (65000, 64520).

Figura 30. **Atributo de ruta de un sistema autónomo**



Fuente: elaboración propia, empleando Visio 2013.

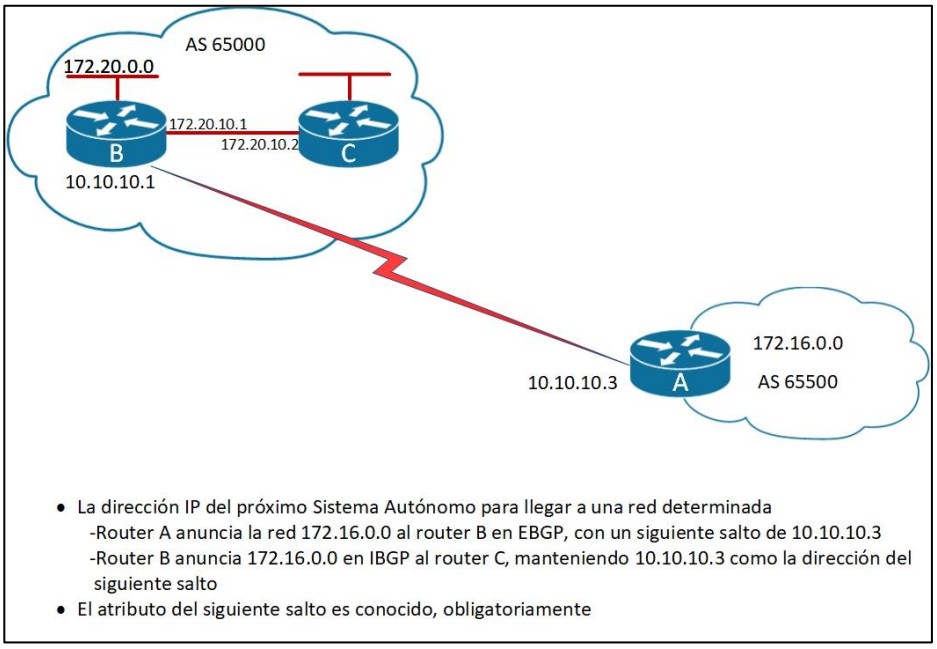
Se aplica un proceso similar para las rutas a las redes 192.168.2.0 y 192.168.3.0. La ruta desde el *router* A a 192.168.2.0 es (65500, 65000), lo que significa que recorre el Sistema Autónomo 65500 y luego el Sistema Autónomo 65000. El *router* C debe atravesar la ruta (65000) para llegar a 192.168.2.0, y la ruta (64520) para llegar a 192.168.1.0.

3.4.4. Atributo del siguiente salto

El atributo BGP del siguiente salto (*next-hop*) es un atributo obligatorio bien conocido que indica la dirección IP del siguiente salto que se utilizara para llegar a un destino. BGP enruta el sistema autónomo por sistema autónomo, no *router* por *router*. El atributo de siguiente salto (*next -hop*) define la dirección IP del *router* de la frontera que debe usarse como el próximo salto destino.

Para EBGP, el siguiente salto es la dirección IP del vecino que envió la actualización. En la figura 31, el *router* A anuncia 172.16.0.0 al *router* B, con un siguiente salto de 10.10.10.3 y el *router* B anuncia 172.20.0.0 al *router* A, con un siguiente salto de 10.10.10.1.

Figura 31. **Atributo del siguiente salto**



Fuente: elaboración propia, empleando Visio 2013.

Para IBGP, el protocolo establece que el próximo salto publicitado por EBGP se debe llevar a IBGP. Debido a esa regla, el *router* B anuncia 172.16.0.0 a su *router* de emparejados IBGP C con un siguiente salto de 10.10.10.3 (dirección del *router* A). Por lo tanto, el *router* C sabe que el siguiente salto para llegar a 172.16.0.0 es 10.10.10.3, no 172.20.10.1, como debería de esperarse.

Es muy importante que el router C sepa cómo llegar a la subred 10.10.10.0, ya sea a través de un IGP o una ruta estática. De lo contrario, descartará los paquetes destinados a 172.16.0.0, porque no puede llegar a la dirección del siguiente salto para esa red.

Alternativamente, el *router* B puede cambiar el atributo del próximo salto a si mismo si usa el comando *neighbor next-hop-self*.

3.4.5. Atributo de origen

El atributo de origen define el origen de la información de ruta. El atributo de origen puede ser uno de estos tres valores:

Figura 32. **Ejemplo del atributo de origen**

```
RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/24      0.0.0.0                  0         32768 i
* i                 10.1.0.2                 0    100      0 i
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*>i10.1.2.0/24      10.1.0.2                 0    100      0 i
*> 10.97.97.0/24    172.31.1.3                             0 64998 64997 i
*                   172.31.11.4                            0 64999 64997 i
* i                 172.31.11.4              0    100      0 64999 64997 i
*> 10.254.0.0/24    172.31.1.3               0             0 64998 i
*                   172.31.11.4                            0 64999 64998 i
* i                 172.31.1.3               0    100      0 64998 i
r> 172.31.1.0/24    172.31.1.3               0             0 64998 i
r                   172.31.11.4                            0 64999 64998 i
r i                 172.31.1.3               0    100      0 64998 i
*> 172.31.2.0/24    172.31.1.3               0             0 64998 i
<output omitted>
```

Fuente: elaboración propia, empleando Visio 2013.

- IGP: la ruta es interior al sistema autónomo de origen. Este valor normalmente se produce cuando el comando de red se utiliza para publicar la ruta a través de BGP. Un origen de IGP se indica con una 'i' en la tabla BGP.

- EGP: la ruta se aprendió a través de EGP. este valor se indica con una 'e' en la tabla BGP. EGP se considera un protocolo de enrutamiento histórico y no es compatible con internet porque solo realiza un enrutamiento *classful* y no admite el enrutamiento interdominio *classless*.

- Incompleto: el origen de la ruta es desconocido o se ha aprendido por otros medios. Este valor generalmente resulta cuando una ruta se redistribuye en BGP. Un origen incompleto se indica con un signo de interrogación (?) en la tabla BGP.

En la figura 33 muestra el resultado del ejemplo del comando show ip bgp. El código de origen, que refleja el atributo de origen, está en la última columna al final de cada línea. En este ejemplo, todos los códigos de origen son 'i', lo que indica un atributo de origen de IGP; las rutas son interiores al sistema autónomo de origen.

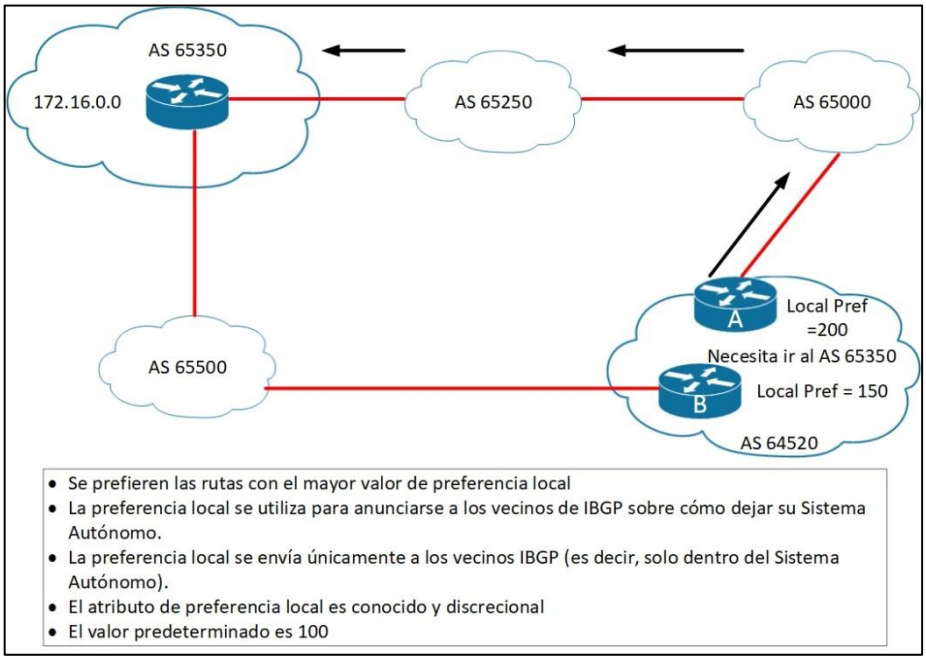
3.4.6. Atributo de preferencia local

La preferencia local es un atributo discrecional bien conocido que proporciona una indicación a los enrutadores en el sistema autónomo sobre qué camino se prefiere para salir del sistema autónomo. se prefiere una ruta con una preferencia local más alta.

La preferencia local es un atributo que se configura en un *router* y se intercambia entre *routers* dentro del mismo sistema autónomo solamente. el valor predeterminado para la preferencia local de un *router* Cisco es 100.

En la figura 34, AS 64520 recibe actualizaciones sobre la red 172.16.0.0 desde dos direcciones. La preferencia local en el *router* A la red 172.16.0.0 se establece en 200 y la preferencia local en el *router* B para la red 172.16.0.0 se establece en 150.

Figura 33. **Atributo de preferencia local**



Fuente: elaboración propia, empleando Visio 2013.

Debido a que la información de preferencia local se intercambia dentro del Sistema Autónomo 64520, todo el tráfico en sistema autónomo 64520 dirigido a

la red 172.16.0.0 se envía al *router* A como punto de salida del sistema autónomo 64520 (debido a su preferencia local más alta).

3.4.7. Atributo *Multi Exit Discriminator* (*MED*)

El atributo *MED*, también llamado métrica, es un atributo opcional no transitivo. El *MED* es una indicación para los vecinos EBGP sobre la ruta preferida en un sistema autónomo. El atributo *MED* es una forma dinámica de influir en otro sistema autónomo sobre la ruta que debe elegir para alcanzar una determinada ruta en su sistema autónomo cuando existen múltiples puntos de entrada. Se prefiere una métrica más baja.

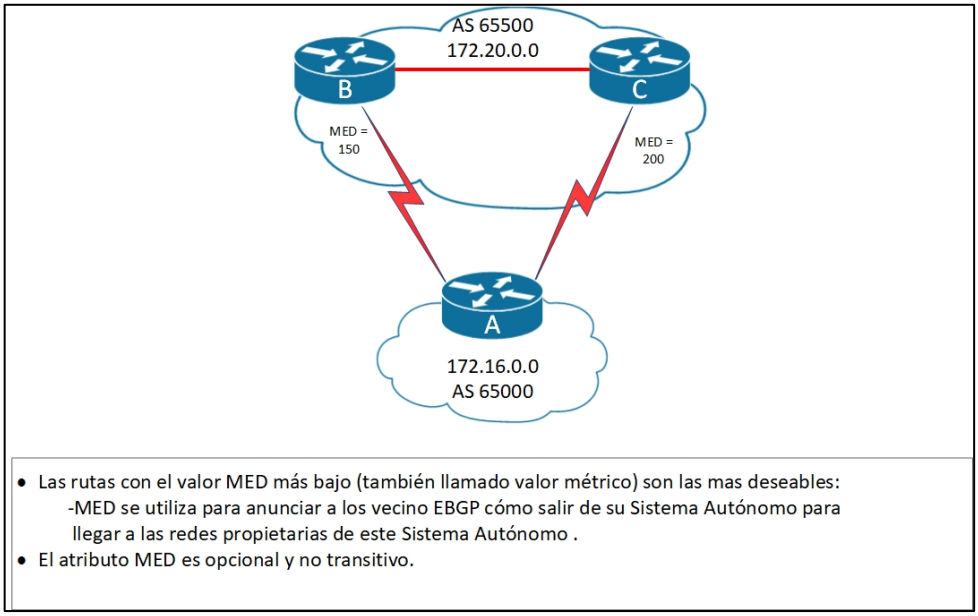
A diferencia de las preferencias locales, el *MED* se intercambia entre sistemas autónomos. El *MED* se envía a los pares EBGP. Esos *routers* propagan el *MED* dentro de su sistema autónomo, y los *routers* dentro del sistema autónomo usan el *MED* pero no lo pasan al siguiente sistema autónomo. Cuando la misma actualización se transfiere a otro sistema autónomo, la métrica vuelve al valor predeterminado 0.

MED influye en el tráfico entrante a un sistema autónomo y la preferencia local influye en el tráfico saliente.

De forma predeterminada, un *router* compara el atributo *MED* solo para las rutas de los vecinos en el mismo sistema autónomo.

El atributo *MED* significa que BGP es el único protocolo que puede afectar como se envían las rutas a un sistema autónomo.

Figura 34. **Atributo *MED***



Fuente: elaboración propia, empleando Visio 2013.

En la figura 34, el atributo B *MED* de *router* se establece en 150 y el atributo C *MED* del *router* se establece en 200. Cuando el *router* A recibe actualizaciones de los *routers* B y C, elige el *router* B como el mejor próximo salto porque su *MED* de 150 es menos que el *router* C.

3.4.8. Atributo de peso

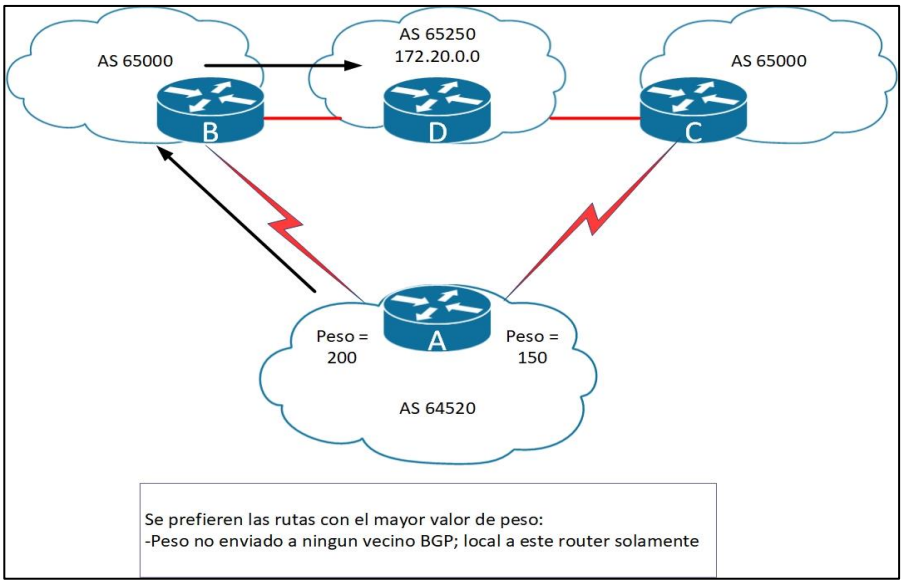
El atributo de peso es un atributo de Cisco para la selección de ruta. El peso se configura localmente en un router y no se propaga a ningún otro router. Este atributo se aplica cuando está utilizando un router con múltiples puntos de salida en el sistema autónomo, en oposición al atributo de preferencia local, que se usa cuando dos o más *routers* proporcionan múltiples puntos de salida.

El peso puede tener un valor de 0 a 65535. De forma predeterminada, las rutas que origina el *router* tienen un peso de 32768 y otras rutas tienen un peso de 0.

Se prefieren las rutas con mayor peso cuando existen varias rutas al mismo destino.

En la figura 35, los routers B y C aprenden sobre la red 172.20.0.0 desde el Sistema Autónomo 65250 y propagan la actualización al *router* A. El *router* A tiene dos formas de llegar a 172.20.0.0, y debe decidir que ruta tomar.

Figura 35. **Atributo de peso (solo equipos Cisco)**



Fuente: elaboración propia, empleando Visio 2013.

En el ejemplo, el *router* A establece el peso de las actualizaciones provenientes del *router* B en 200 y el peso de las que provienen del *router* C en

150. Como el peso del *router* B es más alto que el *router* C, el *router* A usa el *router* B como próximo salto para llegar a 172.20.0.0.

3.4.9. Determinando la selección de ruta BGP

Pueden existir múltiples rutas para llegar a una red determinada. A medida que se evalúan las rutas para la red, aquellas que no son la mejor ruta se eliminan de los criterios de selección, pero se guardan en la tabla de reenvío BGP (que se puede mostrar usando el comando *show ip bgp*) en caso de que se convierta en la mejor ruta inaccesible.

- La tabla de reenvío de BGP generalmente tiene varias rutas entre las que elegir para cada red.

- BGP no está diseñado para realizar balanceo de cargas:

    - Los caminos son elegidos debido a la política
    - Las rutas no eligen según el ancho de banda

- El proceso de selección de BGP elimina cualquier ruta múltiple a través del desgaste hasta que quede una sola mejor ruta.

- Esa mejor ruta se envía al proceso del administrador de la tabla de enrutamiento y se evalúa con los métodos de otros protocolos de enrutamiento para llegar a esa red (utilizando la distancia administrativa).

- La ruta desde la fuente con la distancia administrativa más baja se instala en la tabla de enrutamiento.

BGP no está diseñado para realizar balanceo de carga. Las rutas se eligen debido a políticas, no basadas en el ancho de banda. El proceso de selección BGP elimina cualquier ruta múltiple hasta que quede una sola mejor ruta.

La mejor ruta se envía al proceso del administrador de la tabla de *router* y se evalúa frente a cualquier otro protocolo de enrutamiento de también pueda llegar a esa red. La ruta desde la fuente con la distancia administrativa más baja se instala en la tabla de enrutamiento.

El proceso de decisión se basa en los atributos descritos anteriormente.

Después de que BGP recibe actualizaciones sobre diferentes destinos de diferentes sistemas autónomos, elige la mejor ruta para llegar a un destino específico. el proceso de decisión se basa en los atributos BGP. BGP considera solo las rutas sincronizadas sin bucles del sistema autónomo y un próximo salto válido.

El siguiente proceso resume como BGP elige la mejor ruta en un *router* Cisco:

- Prefiere la ruta con el mayor peso. (El atributo de peso es propiedad de Cisco y solo local para el *router*).

- Si varias rutas tienen el mismo peso, prefiera la ruta con el valor de preferencia local más alto. (La preferencia local se usa dentro de un sistema autónomo).

- Si hay varias rutas con la misma preferencia local, prefiera la ruta en la que se originó el *router* local. Una ruta localmente originaba tiene un siguiente salto de 0.0.0.0 en la tabla BGP.

- Si ninguna de las rutas originó localmente, prefiera la ruta con la ruta más corta del sistema autónomo.

- Si la longitud de la ruta del sistema autónomo es la misma, prefiera el código de origen más bajo (IGP<EGP<incompleto).

- Si todos los códigos de origen son iguales, prefiera la ruta con el *MED* más bajo. (El *MED* se intercambia entre sistemas autónomos). La comparación *MED* solo se realiza si el sistema autónomo vecino es el mismo para todas las rutas consideradas, a menos que el comando esté habilitado.

- Si las rutas tienen el mismo *MED*, prefiera las rutas externas a las rutas internas.

- Si la sincronización está desactivada y solo quedan rutas internas, prefiera la ruta a través del vecino IGP más cercano, lo que significa que el enrutador prefiere la ruta interna más corta dentro del sistema autónomo para llegar al destino (la ruta más corta al siguiente salto BGP).

- Para las rutas EBGP, seleccione la ruta más antigua para minimizar el efecto de las rutas que suben y bajan.

- Prefiera la ruta con el valor de ID de enrutador BGP más cercano.

- Si las ID del router BGP son las mismas, prefiera el router con la dirección IP más baja contigua.

Solo la mejor ruta se ingresa en la tabla de enrutamiento y se propaga a los vecinos BGP del *router*.

Por ejemplo, supóngase que hay siete rutas para llegar a la red 10.0.0.0. Todas las rutas no tienen bucles de sistemas autónomos y tienen direcciones válidas de siguiente salto, por lo que las siete rutas pasan al paso 1, que examina el peso de las rutas.

Las siete rutas tienen un peso de 0, por lo que todas proceden al paso 2, que examina la preferencia local de las rutas. Cuatro de de las rutas tienen una preferencia local de 200 y las otras tres tienen preferencias locales de 100, 100 y 150.

Los cuatro con una preferencia local de 200 continúan el proceso de evaluación hasta el próximo paso. Los otros tres todavía están en la tabla de reenvío BGP, pero actualmente están descalificados como el mejor camino.

BGP continúa el proceso de evaluación hasta que solo queda una única ruta, que se envía a la tabla de enrutamiento IP como la mejor ruta BGP.

3.4.10. Selección de ruta con conexión *multihomed*

Un sistema autónomo raramente implementa BGP con solo una conexión EBGP. Esta situación generalmente significa que existen múltiples rutas para cada red en la base de datos de reenvío de BGP.

Si solo existe una ruta, si está libre de bucles y sincronizada con el IGP para IBGP, y si se puede acceder al siguiente salto, la ruta se envía a la tabla de enrutamiento de IP. No hay una selección de ruta porque solo hay una ruta, y manipularla no produce ningún beneficio.

Se relatan los motivos más comunes para la selección de ruta. sin la manipulación de ruta, la razón más común para la selección de ruta es el paso 4, la preferencia por la ruta más corta del sistema autónomo.

- El paso 1 analiza el peso, que de forma predeterminada se establece en 0 para las rutas que no fueron originadas por este *router*.

- El paso 2 comparar las preferencias locales, que por defecto se establece en 100 para todas las redes. Ambos pasos tienen un solo efecto sólo si el administrador de red configura el peso o la preferencia local a un valor no predeterminado.

- El paso 3 analiza las redes que son propiedad de este sistema autónomo. Si el *router* local inyecta una de las rutas en la tabla BGP, el router local la prefiere a cualquier ruta recibida de otros routers BGP.

- El paso 4 selecciona la ruta que tiene menos sistemas autónomos para cruzar. Esta es la razón más común por la que se selecciona una ruta en BGP. Si a un administrador de red no le gusta la ruta con el menor número de sistemas autónomos, el administrador necesita manipular el peso o la preferencia local para cambiar qué ruta de salida elige BGP.

- El paso 5 analiza cómo se introdujo una red en BGP. Esta introducción generalmente se realiza con declaraciones de red (i para un código de origen) o mediante redistribución (para un código de origen).

- El paso 6 examina a *MED* para determinar donde desea el sistema autónomo vecino que este sistema autónomo envíe paquetes para una red determinada. Cisco establece el *MED* 0 por defecto; por lo tanto, MED no participa en la selección de ruta a menos que el administrador de red del sistema autónomo vecino manipule las rutas usando *MED*.

Si múltiples rutas tienen el mismo número de sistemas autónomo para atravesar, el segundo punto de decisión más común es el paso 7, que establece que se prefiere una ruta aprendida de un vecino IBGP. Un *router* en un sistema autónomo prefiere usar el ancho de banda ISP para llegar a una red en lugar de usar el ancho de banda interno para llegar a un vecino IBGP en el otro lado de su propio sistema autónomo.

Si la ruta del sistema autónomo es igual y el *router* de un sistema autónomo no tiene vecinos EBGP para esa red (solo vecino IBGP), tiene sentido tomar la ruta más rápida al punto de salida más cercano.

El paso 8 busca al vecino de IBGP más cercano. La métrica IGP determina que significa 'más cercano', por ejemplo, RIP utiliza el recuento de saltos, y OSPF utiliza el menor costo en función del ancho de banda.

Si la ruta del sistema autónomo es igual y los costos a través de todos los vecinos de IBGP son iguales, o si todos los vecinos de esta red son EBGP, el paso 9 es la siguiente razón más común para seleccionar una ruta sobre la otra. Los vecinos de EBGP rara vez establecen sesiones al mismo tiempo. Es

probable que una sesión sea más antigua que otra, por lo que las rutas a través de ese vecino más viejo se consideran más estables porque han estado activas durante más tiempo.

Si todos los criterios enumerados son iguales, la siguiente decisión más común es tomar el vecino con la ID más baja del *router* BGP, que es paso 10.

Si los ID del router BGP son los mismos (por ejemplo, si las rutas son para el mismo *router* BGP), el paso 11 establece que se utiliza la ruta con la dirección IP más baja del vecino.

3.5. MPLS

Este es denominado como una técnica, el cual ofrece una VPN IP hasta una ethernet metropolitana. Este funciona como una conmutación de etiquetas multiple, es muy utilizado en redes WAN, por lo que es altamente eficiente.

Es una técnica que no encaja perfectamente en ninguna capa del modelo OSI, lo cual separa mecanismos de reenvio de servicio de enlace de datos subyacentes, este principalmente trabaja con etiquetas, lo cual hace la distribución de paquetes mucho mas eficiente.

3.5.1. Introducción a MPLS (*multiprotocol label switching*)

La conmutación de etiquetas multiprotocolo (MPLS) existe desde hace varios años. Es una tecnología de red popular que usa etiquetas adjuntas a paquetes para enviarlas a través de la red. En las próximas secciones se explicará a detalle por qué MPLS se hizo tan popular en tan poco tiempo.

Las etiquetas MPLS se anuncian entre los enrutadores para que puedan crear una asignación de etiqueta a etiqueta. Estas etiquetas se adjuntan a los paquetes IP, lo que permite a los enrutadores reenviar el tráfico mirando la etiqueta y no la dirección IP de destino. Los paquetes se envían por cambio de etiqueta en lugar de conmutación IP.

El reenvío tradicional de paquetes IP analiza la dirección IP de destino contenida en el encabezado de la capa de red de cada paquete a medida que el paquete viaja desde su origen hasta su destino final. Un *router* analiza la dirección IP de destino de forma independiente en cada salto en la red. Los protocolos de enrutamiento dinámico o la configuración estática construyen la base de datos necesaria para analizar la dirección IP de destino (la tabla de enrutamiento). El proceso de implementación del enrutamiento tradicional IP también se denomina enrutamiento unidifusión basado en el destino salto por salto (*hop-by-hop*).

Aunque exitoso, y obviamente ampliamente implementado, existen ciertas restricciones, que se han realizado durante algún tiempo, para éste método de reenvío de paquetes se disminuye su flexibilidad. Por lo tanto, se requieren nuevas técnicas para abordar y expandir la funcionalidad de una infraestructura de red basada en IP.

3.5.1.1. Beneficios de MPLS

Está sección explica brevemente los beneficios de ejecutar MPLS en su red. Estos beneficios incluyen los siguientes:

- El uso de una infraestructura de red unificada
- Mejor integración IP sobre ATM

- Núcleo libre de BGP
- El modelo punto a punto para MPLS VPN
- Flujo de tráfico óptimo
- Ingeniería de tráfico

La conmutación de etiquetas multiprotocolo (MPLS) es una tecnología emergente que tiene como objetivo abordar muchos de los problemas existentes asociados con el reenvío de paquetes en el entorno actual de *internetworking*. Los miembros de la comunidad de IETF trabajaron extensamente para llevar un conjunto de estándares al mercado y para desarrollar las ideas de varios vendedores e individuos en el área del cambio de etiquetas. El documento IETF *draft-ietf-mpls-frameworks* contiene el marco de esta iniciativa y describe el objetivo principal de la siguiente manera: el objetivo principal del grupo de trabajo MPLS es estandarizar una tecnología base que integre el paradigma de reenvío de etiquetas con el enrutamiento de capa de red. Se espera que esta tecnología base (intercambio de etiquetas) mejore la relación precio/rendimiento del *router* de capa de red; mejore la escalabilidad de capa de red y brinde una mayor flexibilidad en la entrega de nuevos servicios de enrutamiento (permitiendo agregar nuevos servicios de enrutamiento sin un cambio en el paradigma de reenvío).

La arquitectura MPLS describe los mecanismos para realizar la conmutación de etiquetas, que combina los beneficios del reenvío de paquetes basado en la conmutación de capa 2 con los beneficios del enrutamiento de capa 3. De forma similar a las redes de capa 2 (por ejemplo, *Frame Relay* o ATM), MPLS asigna etiquetas a los paquetes para su transporte a través de redes basadas en paquetes o celdas. El mecanismo de reenvío en toda la red es el intercambio de etiquetas, en el que las unidades de datos (por ejemplo, un paquete o una celda) llevan una etiqueta corta de longitud fija que indica a los

nodos de conmutación a lo largo de la ruta de paquetes como procesar y reenviar los datos.

La diferencia significativa entre MPLS y las tecnologías WAN tradicionales es la forma en que se asignan las etiquetas y la capacidad de llevar una pila de etiquetas unidas a un paquete. El concepto de una pila de etiquetas permite nuevas aplicaciones, tales como ingeniería de tráfico, redes privadas virtuales, redireccionamiento rápido en fallas de enlace y nodos, y más.

El reenvío de paquetes en MPLS contrasta con el entorno de red sin conexión de hoy en día, donde cada paquete se analiza salto por salto, se comprueba su encabezado de capa 3 y se toma una decisión de reenvío independiente basada en la información extraída de una red algoritmo de capa de enrutamiento.

La arquitectura se divide en dos componentes separados: el componente de reenvío (también llamado el plano de datos) y el componente de control (también llamado plano de control). El componente de reenvío utiliza una base de datos de reenvío de etiquetas mantenida por un conmutador de etiquetas para realizar el reenvío de paquetes de datos en función de etiquetas transportadas por paquetes. El componente de control es responsable de crear y mantener información de reenvío de etiquetas (a la que hace referencia como enlaces) entre un grupo de conmutadores de etiquetas interconectados.

Cada nodo MPLS debe ejecutar uno o más protocolos de enrutamiento IP (o confiar en el enrutamiento estático) para intercambiar información de enrutamiento IP con otros nodos MPLS en la red. En este sentido, cada nodo MPLS (incluidos los conmutadores ATM) es un *router* de IP en el plano de control.

Al igual que en los *routers* tradicionales, los protocolos de enrutamiento IP llenan las tablas de enrutamiento. En los *routers* IP tradicionales, la tabla de enrutamiento IP se usa para construir el caché de reenvío IP (caché de conmutación rápida en Cisco IOS) o la tabla de reenvío IP (*forwarding information base,* FIB) utilizado por Cisco Express Forwarding (CEF).

En un nodo MPLS, la tabla de enrutamiento de IP se usa para determinar el intercambio de enlace de etiqueta, donde los nodos MPLS adyacentes intercambian etiquetas para subredes individuales que están contenidas dentro de la tabla de enrutamiento IP. El intercambio de enlaces de etiquetas para enrutamiento IP basado en destinos de unidifusión se realiza utilizando el protocolo de distribución de etiquetas (TDP) patentado por Cisco o el protocolo de etiquetas (LDP) especificado por IETF.

El proceso *MPLS IP Routing Control* utiliza etiquetas intercambiadas con nodos MPLS adyacentes para crear la tabla de reenvío que se utiliza para reenviar paquetes etiquetados a través de la red MPLS.

3.5.2. Arquitectura MPLS: bloques de construcción

Al igual que con cualquier tecnología nueva, se introducen varios términos nuevos para describir los dispositivos que componen la arquitectura. Estos nuevos términos describen la funcionalidad de cada dispositivo y sus roles dentro de la estructura de dominio MPLS.

El primer dispositivo en ser introducido es el *Label Switch Router* (LSR). Cualquier *router* o *switch* que implemente los procedimientos de distribución de etiquetas y pueda reenviar paquetes según las etiquetas y pueda reenviar paquetes según las etiquetas se incluye en esta categoría. La función básica de

los procedimientos de distribución de etiquetas es permitir que un LSR distribuya sus enlaces de etiquetas a otros LSR dentro de la red MPLS.

Existen varios tipos diferentes de LSR que se diferencian por la funcionalidad que proporcionan dentro de la infraestructura de red. Estos diferentes tipos de LSR se describen dentro de la arquitectura como Edge-LSR, ATM-LSR y ATM edge-LSR. La distinción entre varios tipos de LSR es puramente arquitectónica: un solo cuadro puede servir para varios roles.

Un Edge-LSR es un *router* que realiza la imposición de etiqueta (a veces también denominada acción de inserción) o la disposición de etiqueta (también llamada acción emergente) en el borde de la red MPLS. La imposición de etiqueta es el acto de anteponer una etiqueta, o una pila de etiquetas, a un paquete en el punto de entrada (con respecto al flujo de tráfico desde el origen al destino) del dominio MPLS. La disposición de la etiqueta es al revés de esto es el acto de eliminar la última etiqueta de un paquete en un punto de salida antes de que se envíe a un vecino que está fuera del dominio MPLS.

Cualquier LSR que tenga vecinos que no sean MPLS se considera un Edge-LSR. Sin embargo, si ese LSR tiene interfaces que se conectan a través de MPLS a un ATM-LSR; entonces, también, se considera que es un ATM-LSR de borde. Los LSR de borde o LSR-Edge utilizan una tabla de reenvío de IP tradicional, aumentada con información de etiquetado, para etiquetar paquetes IP o para eliminar etiquetas de paquetes etiquetados antes de enviarlos a nodos MPLS.

3.5.3. MPLS capa 3 VPN

Las prioridades comerciales actuales de muchas empresas multinacionales ciertamente no son comunes. A medida a que una empresa multinacional se vuelva más unida como una empresa global, su departamento de TI debe estar preparado para manejar un cambio dramático en su carga de aplicaciones. El arquitecto de red de muchas empresas multinacionales debe considerar varias tecnologías para alcanzar los objetivos comerciales de estas empresas han establecido.

Una empresa multinacional debería comenzar por considerar la capa 3 (L3) IP / MPLS VPN. Está tecnología le permite a estas empresas a subcontratar efectivamente el núcleo de su WAN; eliminar el esfuerzo necesario para planificar y construir una arquitectura compleja y concentrada; y aprovechar la escala de red de un proveedor de servicios. Este tipo de servicio generalmente brinda importantes ahorros de costos para toda la red de malla completa, la piedra angular de la colaboración empresarial a gran escala. Sin VPN L3 IP/MPLS, esta empresa puede encontrarse trabajando muy duro para administrar una red de latencia óptima.

Otra tecnología que las empresas pueden considerar es el uso de servicios VPN de capa 2 (L2). Estos servicios, cuando se construyen en una red IP/MPLS, permiten a estas empresas mantener el control completo sobre el enrutamiento de capa 3 dentro de su red, porque el proveedor de servicios no intercambia información de enrutamiento IP. Las empresas tienen la opción entre un par de servicios L2 VPN, un servicio de punto a punto basado en el servicio de cable privado virtual (VPWS) y una LAN de emulación multipunto basada en el servicio de LAN privada virtual (VPLS). Es posible que estas

empresas intentan consolidar alguna red de área metropolitana (MAN) y luego enlazar la red metropolitana a la red local (WAN) VPN IP/MPLS L3.

3.5.4. Capa 3 IP/MPLS VPN

El arquitecto de red de una empresa cree que los servicios L3 IP/MPLS VPN son precisamente lo que quiere para proporcionar una base para su WAN empresarial y para respaldar sus iniciativas comerciales.

La solución IP/MPLS VPN se basa en IETF RFC 2547. Está mecanismo cuenta con un amplio soporte de la industria para VPN basadas en red y se está convirtiendo rápidamente en un estándar común en todo el mundo para la conectividad IP.

3.5.5. Topologías y aprovisionamiento de servicios IP/MPLS VPN

L3 IP/MPLS VPN virtualiza el núcleo de la red de proveedores de servicios, lo que permite a la red agrupar el tráfico a su destino en función de la tabla de enrutamiento IP de la empresa, compartida con la red del proveedor del servicio. Esto indica que ya no se requiere que la empresa diseñe y mantenga una malla de la ubicación del concentrador y enlaces de interconexión. En cambio, cada sitio es un sitio final en la nube, y la empresa necesita administrar un solo puerto a la nube. L3 IP/MPLS VPN también simplifica la planificación de capacidad que una empresa debe realizar en su red.

La figura muestra el papel de la empresa en las redes de ingeniería basadas en la tecnología de multiplicación por división de tiempo (TDM), ATM o

Frame Relay. La figura muestra el rol en las redes L3 IP/MPLS basadas en VPN. Las porciones discontinuas y curvas de las redes detalladas en la figura representan los PVC que la empresa debe diseñar, aprovisionar y administrar la capacidad. Por el contrario, este requisito de la empresa se elimina en la VPN L3 IP/MPLS, donde el enrutamiento inteligente toma el lugar del aprovisionamiento de PVC. También, que tenga en cuenta el aprovisionamiento de ancho de banda; en lugar de determinar las capacidades de punto a punto, el personal de la red de la empresa necesita mantener solo la planificación de la capacidad sitio por sitio.

Para el ingeniero de redes de la empresa, la conexión de un sitio a la nube L3 IP/MPLS VPN parece una conexión a otro router en la red de la empresa. Los protocolos de enrutamiento externo, como BGP, o los protocolos de enrutamiento interno, como OSPF, RIP o EIGRP, intercambian información de enrutamiento con el router frontera (PE, *Provider Edge*). Los *router* transportan a través de la red IP/MPLS del proveedor de servicios en multiprotocolo BGP.

Además, muchas de las capacidades de red empresarial requeridas, como QoS y multicast IP, pueden ser soportadas por el servicio IP/MPLS VPN. Estos servicios se aprovisionan de forma nativa, como los protocolos de enrutamiento, y aparecen como si otro router empresarial estuviera en el otro lado del enlace.

La diferencia clave entre está tecnología típica WAN y subcontratada es que una única red de proveedores de servicios construida una vez con las tecnologías L3 IP/VPN puede venderse muchas veces a muchos clientes, a diferencia de un proveedor de servicios o experto que diseña una WAN individual para cada cliente.

El acceso al servicio L3 IP/MPLS VPN puede realizarse a través de cualquier tecnología de capa 2. entre el *router* de cliente (CE, *Customer Edge*) ubicado en el sitio de la empresa y el *router* de PE en el proveedor de servicios, se pueden usar tecnologías tradicionales como líneas arrendadas, cajeros automáticos y *Frame Relay*. Alternativamente, se pueden usar tecnologías de acceso más nuevas como Ethernet metropolitana. La disponibilidad de acceso *Frame Relay* o ATM a un servicio L3 IP/MPLS VPN proporciona una ruta de migración sencilla desde una red existente a L3 IP/MPLS VPN.

El servicio L3 IP/MPLS VPN también elimina la necesidad de la empresa de alquilar instalaciones y recambios de *rack* para equipos de red en ubicaciones de concentradores. Debido a que la inteligencia está integrada en la red del proveedor de servicios, la empresa no necesita interconectar PVC o circuitos punto a punto en sus propios equipos de red y puntos de enrutamiento clave en la red.

3.5.6. IP/MPLS VPN: una fundación para servicios de red

Las tecnologías L3 IP/MPLS VPN ofrecen una mejor capacidad de integración para servicios de red avanzados. En lugar de diseñar una conectividad especial para un servicio ofrecido por un proveedor de servicios, que puede ser una puerta de enlace VoIP a la PSTN, un proveedor de servicios puede integrar el servicio simplemente importando las rutas empresariales hacia y desde la VPN del cliente en la VPN de servicio.

3.5.7. Transparencia IP/MPLS VPN

Uno de los aspectos más importantes que debe tenerse en cuenta en un servicio L3 IP/MPLS VPN es su transparencia. Empresas que han operado una

red por algún tiempo han establecido parámetros clave para el funcionamiento de sus redes. Elementos como clases de servicios y sus valores de punto de código de servicios diferenciados (*differentiated services code point*, DSCP) asociados, el protocolo de enrutamiento utilizado en la red y la capacidad de multidifusión IP se requieren en la red de empresas multinacionales o muy grandes con sucursales, y la introducción del servicio L3 IP/MPLS VPN no obligará a las empresas a rediseñar su red para que el servicio se ajuste. Se podría decir que hay una buena razón para llamarlo una red privada virtual, la idea que debe parecerse mucho a la red privada de cualquier empresa.

3.5.8. IP/MPLS VPN administración de red SLA

En las redes basadas en capa 2, las empresas tienen control sobre toda la red de capa 3, lo que permite una capacidad de solución de problemas sin restricciones en toda la red. En los servicios L3 IP/MPLS VPN, ahora existe una responsabilidad compartida para los aspectos de capa 3 de la red entre la empresa y el proveedor de servicios, lo que puede hacer que la administración y el monitoreo sean más complejos.

Uno de los temas delicados entre las empresas y sus proveedores de servicio hoy en dia es el acuerdo de nivel de servicio (SLA). A medida que las empresas se vuelven cada vez más dependientes de una infraestructura de comunicaciones cubierta, el administrador de la red empresarial espera más de sus proveedores de servicios. Las tecnologías como las redes de voz, video y almacenamiento exigen estrictamente ciertas características de la red de datos, como el retraso y la inestabilidad. Antes de que la red IP adoptara estos tipos de tráfico, los requisitos de retardo y la fluctuación de fase eran bastante imprecisos. Dado que L3 IP/MPL VPN se centra en la entrega de QoS, las capacidades y los conjuntos de herramientas adecuados para gestionar los SLA

con estas características están disponibles y son una parte clave del servicio de red.

Finalmente, las empresas deben considerar la administración del servicio. En la mayoría de los casos, el proveedor de servicios ofrece un servicio totalmente administrado o un servicio no administrado, o ambos. En el caso del servicio totalmente gestionado, el proveedor del servicio suministra y gestiona por completo la configuración, la monitorización y la solución de problemas del *router* CE (*Customer Edge*) y la conectividad WAN conectada utilizado sus herramientas y procedimientos. Un servicio no administrado permite a la empresa mantener la configuración y administración de los *routers Customer Edge* (CE), dejando que el proveedor de servicios administre solo los *routers* PE. El primero (servicio totalmente administrado) es ligeramente flexible para la empresa, pero permite que el proveedor de servicios ofrezca un acuerdo de nivel de servicio más completo, con control sobre toda la porción de la WAN. Este último le permite a la empresa más control para usar sus herramientas de medición y monitoreo. Los capítulos posteriores discuten las diferencias entre los modelos e introducen algunos híbridos entre los dos, dependiendo de los requisitos de la empresa y del proveedor del servicio.

3.5.9. Arquitectura MPLS

MPLS significa cambio de etiquetas multiprotocolo. El aspecto multiprotocolo de MPLS se cumplió después de la implementación inicial de MPLS en una amplia gama de marcas de equipos. Aunque al principio solo se cambiaba la etiqueta de IPv4, más adelante se sumaron más protocolos.

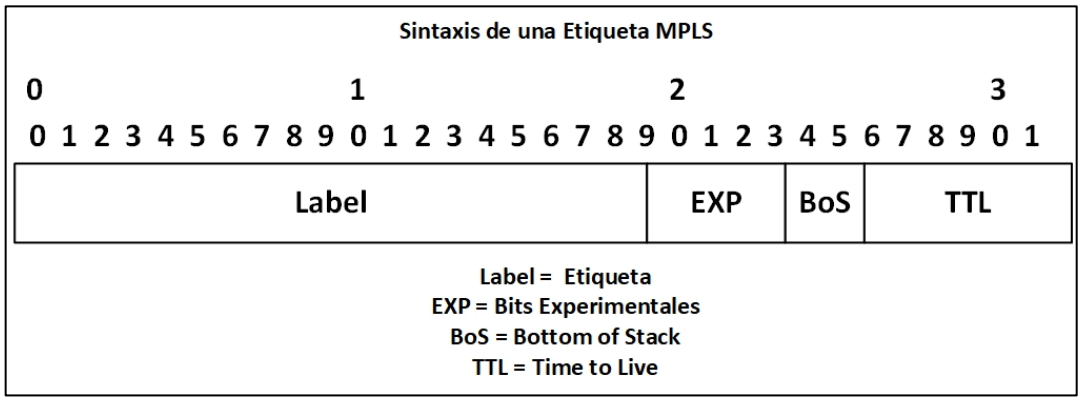
La conmutación de etiquetas indica que los paquetes conmutados ya no son paquetes IPv4, paquetes IPv6 o incluso tramas de nivel 2 cuando se

cambian, sino que están etiquetados. El elemento más importante para MPLS es la etiqueta.

3.5.10. Introducción a etiquetas MPLS

Una etiqueta MPLS es un campo de 32 bits con una cierta estructura. En la figura 36 se muestra la sintaxis de una etiqueta MPLS.

Figura 36. **Sintaxis de una etiqueta MPLS**



Fuente: elaboración propia, empleando Visio 2013.

Los primeros 20 bits son el valor de la etiqueta. Este valor puede estar entre 0 y 2^20 - 1, o 1 048 575. Sin embargo, los primeros 16 valores están exentos del uso normal; es decir, tienen un significado especial. Los *bits* 20 a 22 son los tres bits experimentales (EXP). Estos bits se usan únicamente para calidad de servicio (QoS).

El bit 23 es el bit de la parte inferior de la pila (*bottom of stack*, BoS). Es 0, a menos que esta sea la etiqueta inferior en la pila. Si es así, el *bit BoS* se establece en 1. La pila es la colección de etiquetas que se encuentran en la

parte superior del paquete. La pila puede consistir en una sola etiqueta, o podría tener más. El número de etiquetas (es decir, el campo de 32 *bits*) que puede encontrar en la pila no tiene límites, aunque rara vez debería ver una pila que consta de cuatro o más etiquetas.

Los bits 24 a 31 son los ocho bits utilizados para Time To Live (TTL). Está TTL tiene la misma función que el TTL encontrado en el encabezado IP. Simplemente se reduce en 1 en cada salto, y su función principal es evitar que un paquete se atasque en un bucle de enrutamiento. Si se produce un bucle de enrutamiento y no hay TTL presente, el paquete se repite para siempre. Si el TTL de la etiqueta llega a 0, el paquete se descarta.

3.5.11. Apilamiento de etiqueta

Los *routers* compatibles con MPLS pueden necesitar más de una etiqueta en la parte superior del paquete para enrutar ese paquete a través de la red MPLS. Esto se hace empacando las etiquetas en una pila. La primera etiqueta en la pila se llama etiqueta superior, y la última etiqueta se llama etiqueta inferior. En el medio, puede tener cualquier cantidad de etiquetas. La figura 37. muestra la estructura de la pila de etiquetas.

Figura 37. **Pila de etiquetas**

Fuente: elaboración propia, empleando Visio 2013.

Observe que la pila de etiquetas en la figura 37 muestra que el bit BoS es 0 para todas las etiquetas, excepto la etiqueta inferior. Para la etiqueta inferior, el bit BoS se establece en 1.

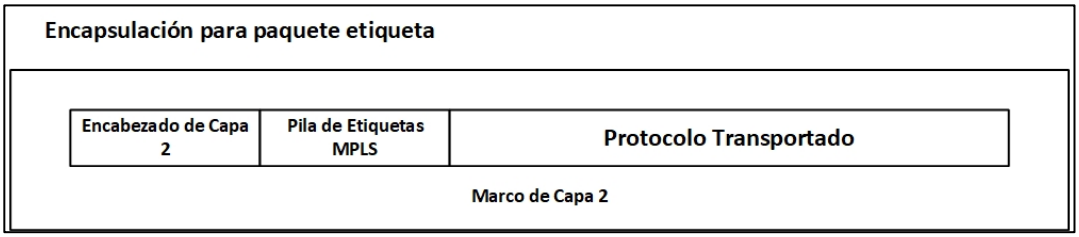
Algunas aplicaciones MPLS realmente necesitan más de una etiqueta en la pila de etiquetas para reenviar los paquetes etiquetados. Dos ejemplos de tales aplicaciones MPLS son MPLS VPN y AToM. Tanto MPLS VPN como AToM ponen dos etiquetas en la pila de etiquetas.

3.5.12. Codificación de MPLS

La pila de etiquetas se encuentra frente al paquete de capa 3, es decir, antes del encabezado del protocolo transportado, pero después del encabezado de capa 2. A menudo, la pila de etiquetas MPLS se denomina encabezado shim debido a su ubicación.

En la figura 38 muestra la ubicación de la pila de etiquetas para los paquetes etiquetados.

Figura 38. **Encapsulación para paquetes de etiqueta**



Fuente: elaboración propia, empleando Visio 2013.

La encapsulación de capa 2 del enlace puede ser casi cualquier encapsulación compatible con Cisco IOS:

PPP, control de enlace de datos de alto nivel (HDLC, *high-level data link control*), Ethernet, entre otros. Suponiendo que el protocolo transportado es IPv4, y la encapsulación de un enlace PPP, la pila de etiquetas está presente después del encabezado PPP, pero antes del encabezado IPv4. Como la pila de etiquetas en el marco de capa 2 se coloca antes del encabezado de capa 3 u otro protocolo transportado, debe tener valores nuevos para el campo protocolo de capa de enlace de datos, lo que indica que lo que sigue al encabezado de capa 2 es un paquete etiquetado MPLS.

El campo de protocolo de capa de enlace de datos es un valor que indica que tipo de carga está transportando el marco de capa 2. en la tabla se muestra cuales son lo nombre y valores para el campo identificador de protocolo en el encabezado de capa 2 para los diferentes tipos de encapsulación de capa 2.

Figura 39. **Valores de identificador de protocolo MPLS para los tipos de encapsulación de capa 2**

Valores de Identificador de Protocolo MPLS para los tipos de Encapsulación de Capa 2		
Tipo de encapsulamiento de Capa 2	**Nombre de Identificador de Protocolo de Capa 2**	**Valor (Hex)**
PPP	Campo de Protocolo PPP	0281
Encapsulacion Ethernet / 802.3 LCC / SNAP	Valor de Ethertype	8847
HDLC	Protocolo	8847
Frame Relay	NLPID (ID de protocolo de nivel de red)	80

Fuente: elaboración propia, empleando Visio 2013.

3.5.13. MPLS y el modelo de referencia OSI

La capa inferior es capa 1, o la capa física, y la capa superior es capa 7, o la capa de aplicación. Mientras que la capa física se refiere a las características de cableado, mecánicas y eléctricas, la capa 2, la capa de enlace de datos, se ocupa del formateo de los cuadros. Algunos ejemplos de la capa de enlace de datos json Ethernet, PPP, HDLC y *Frame Relay*. La importancia de la capa de enlace de datos solo se encuentra en un enlace entre dos máquinas, pero no más allá. Esto significa que el encabezado de la capa de enlace de datos siempre es reemplazado por la máquina en el otro extremo del enlace. La capa 3, la capa de red, está relacionada con el formateo de paquetes de extremo a extremo. Tiene significado más allá del enlace de datos. El ejemplo más conocido de un protocolo que opera en capa 3 es IP.

MPLS no es un protocolo de capa 2 porque la encapsulación de capa 2 aún está presente con los paquetes etiquetados. MPLS tampoco es realmente

un protocolo de capa 3 porque el protocolo de capa 3 aún está presente también. Por lo tanto, MPLS no encaja demasiado bien en la estratificación OSI. Quizás lo más difícil de hacer es ver MPLS como la capa 2.5 y terminar con esto.

3.5.14. *Label switch router*

Un *router* de cambio de etiqueta (LSR) es un router que admite MPLS. Es capaz de entender las etiquetas MPLS y de recibir y transmitir un paquete etiquetado en un enlace de datos. Existen tres tipos de LSR en una red MPLS:

- LSR de entrada: los LSR de entrada reciben un paquete que aún no está etiquetado, inserta una etiqueta (pila) delante del paquete y lo envía en un enlace de datos.

- LSR de egreso: los LSR de egreso reciben paquetes etiquetados, eliminan las etiquetas y los envían en un enlace de datos. Los LSR de entrada y salida son LSR de borde.

- LSR intermedios: los LSR intermedios reciben un paquete etiquetado entrante realizan una operación en él, cambian el paquete y envía el paquete al enlace de datos correcto.

Un LSR puede hacer las tres operaciones: por, *push* o *swap*.

Debe mostrar una o más etiquetas (eliminar una o más etiquetas de la parte superior de la pila de etiquetas) antes de apagar el paquete. Un LSR también debe poder insertar una o más etiquetas en el paquete recibido. Si el paquete recibido ya está etiquetado, el LSR empuja una o más etiquetas en la pila de etiquetas y cambia el paquete. Si el paquete aún no está etiquetado, el

LSR crea una pila de etiquetas y lo empuja al paquete. Un LSR también debe poder intercambiar una etiqueta. Esto simplemente significa que cuando se recibe un paquete etiquetado, la etiqueta superior de la pila de etiquetas se intercambia con una nueva etiqueta y el paquete se activa en el enlace de datos de salida.

Un LSR que empuja etiquetas a un paquete a un paquete que aún no fue etiquetado se llama LSR imponente porque es el primer LSR que impone etiquetas en el paquete. Uno que está haciendo imposición es un ingreso LSR. Un LSR que elimina todas las etiquetas del paquete etiquetado antes de desconectar el paquete es un LSR eliminador. Uno que hace disposición es un egreso LSR.

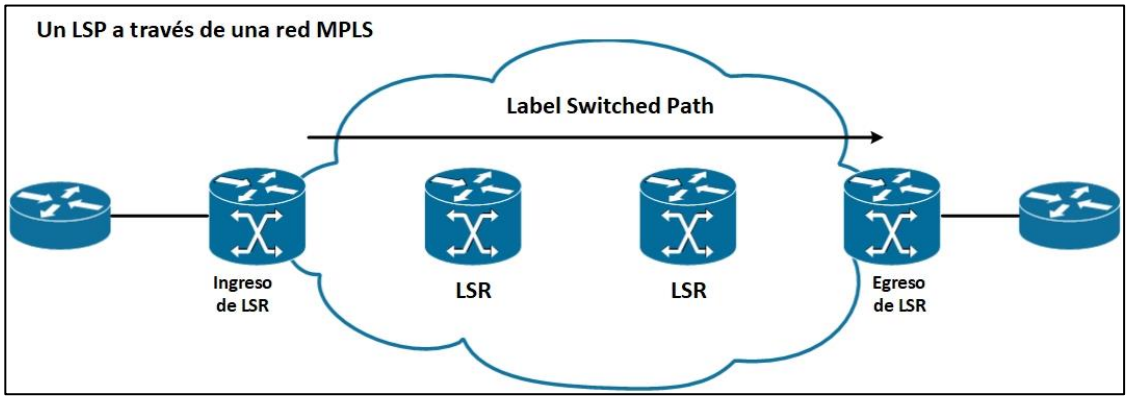
En el caso de MPLS VPN, los LSR de entrada y salida se denominan routers de borde de proveedor (PE, *Provider Edge*). Los LSR intermedios se conocen como *routers* de proveedor (P). Los términos *routers* PE y P se han vuelto tan populares que también se usan cuando la red MPLS no se ejecuta MPLS VPN.

3.5.15. *Label switched* PATH

Una ruta conmutada de etiqueta (LSP) es una secuencia de LSR que conmuta un paquete etiquetado a través de una red MPLS o parte de una red MPLS. Básicamente, el LSP es una ruta a través de la red MPLS o una parte de la que toman los paquetes. El primer LSR de un LSP es un ingreso LSR para ese LSP, mientras que el último LSR del LSP es un LSR de salida. Todos los LSR entre los LSR de entrada y salida son los LSR intermedios.

En la figura 40, la flecha en la parte superior indica la dirección, porque un LSP es unidireccional. El flujo de paquetes etiquetados en la otra dirección, de derecha a izquierda, entre los mismos LSR de borde seria otros LSP.

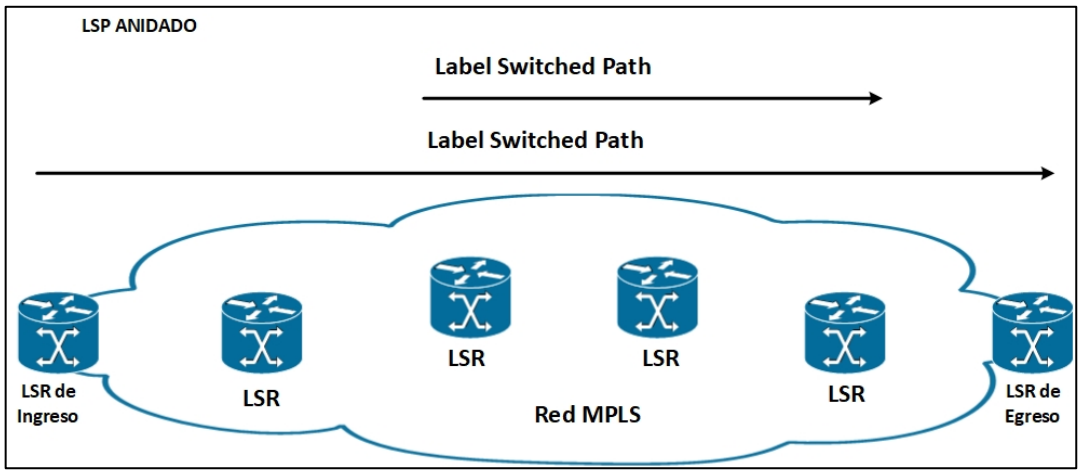
Figura 40. **Un LSP a través de una red MPLS**



Fuente: elaboración propia, empleando Visio 2013.

El ingreso LSR de un LSP no es necesariamente el primer *router* para etiquetar el paquete. Es posible que el paquete ya haya sido etiquetado por un LSR anterior. Tal caso sería un LSP anidado, es decir, un LSP dentro de otro LSP. En la figura, se puede ver un LSP que abarca todo el ancho de la red MPLS. Otro LSP comienza en el tercer LSR y finaliza en el penúltimo LSR. Por lo tanto, cuando el paquete ingresa al segundo LSP en un ingreso LSR (esto significa el tercer LSR), ya está etiquetado. Está LSR de ingreso del LSP anidado luego empuja una segunda etiqueta de paquete. La pila de etiquetas del paquete en el segundo LSP tiene ahora dos etiquetas. La etiqueta superior pertenece al LSP anidado y la etiqueta inferior pertenece al LSP que abarca toda la red MPLS. Un túnel de ingeniería de tráfico de respaldo (TE) es un ejemplo de dicho LSP anidado.

Figura 41. **LSP anidado**



Fuente: elaboración propia, empleando Visio 2013.

3.5.16. Clase de equivalencia de reenvío

Una clase de equivalencia de reenvío (*Forwarding equivalence class*) es un grupo o flujo de paquetes que se envían a lo largo de la misma ruta y se tratan de la misma manera con respecto al tratamiento de reenvío. Todos los paquetes que pertenecen a la misma FEC tiene la misma etiqueta. Sin embargo, no todos los paquetes tienen la misma etiqueta pertenecen al mismo FEC, porque sus valores de EXP pueden diferir; el tratamiento de reenvió podría ser diferente, y podrían pertenecer a un FEC diferente. El *router* que decide qué paquetes pertenecen a qué FEC es el ingreso LSR. Esto es lógico porque el ingreso LSR clasifica y etiqueta los paquetes.

Los siguientes son algunos ejemplos FEC:

- Paquetes con direcciones IP de destino de capa 3 que coincidan con un cierto prefijo.

- Paquetes de multidifusión que pertenecen a un cierto grupo.

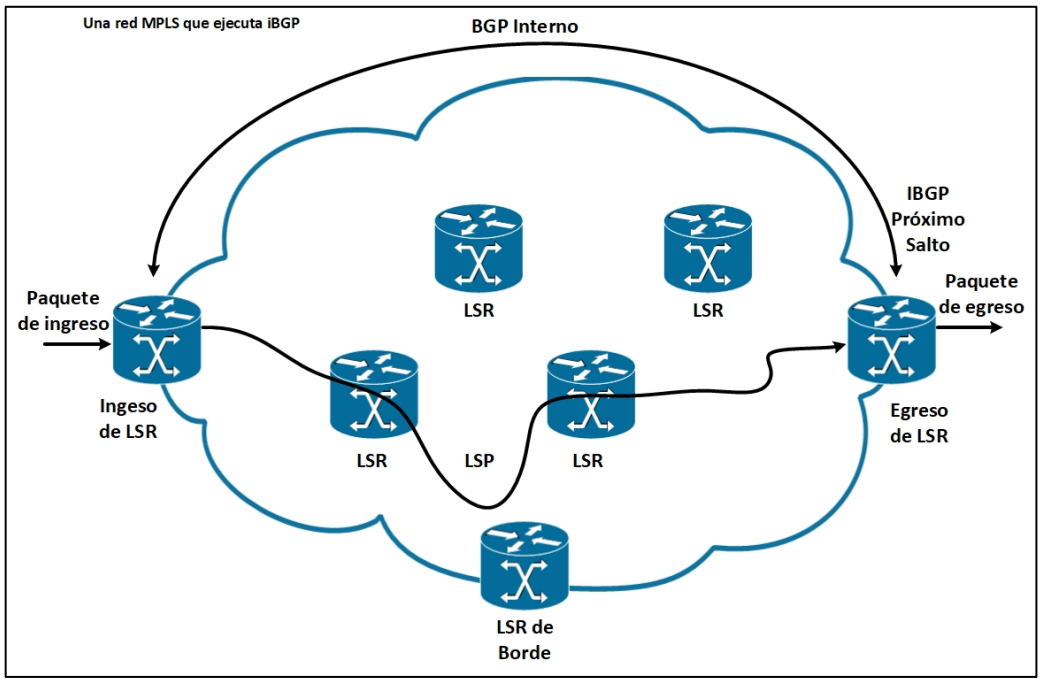
- Paquetes con el mismo tratamiento de reenvío, basado en el campo *Preference Code* o IP *DiffServ Code Point* (DSCP).

- Las tramas de capa 2 se transportan a través de una red MPLS recibida en un VC o (sub) interfaz en el LSR de entrada y se transmiten en un VC o (sub) interfaz en el LSR de salida.

- Paquetes con direcciones IP de destino de capa 3 que pertenecen a un conjunto de prefijos de protocolo de puerta de enlace de frontera (BGP), todos con el mismo salto siguiente de BGP.

Este último ejemplo de FEC es particularmente interesante. Todos los paquetes del LSR de ingreso para los cuales la dirección IP de destino apunta a un conjunto de rutas BGP en la tabla de enrutamiento, todos con la misma dirección BGP del siguiente salto, pertenecen a un FEC. Significa que todos los paquetes que ingresan a la red MPLS obtienen una etiqueta dependiendo de cuál sea el siguiente salto de BGP. La figura 42 muestra una red MPLS en la que todos los LSR de borde ejecutan BGP interno (iBGP).

Figura 42. **Una red MPLS que ejecuta iBGP**



Fuente: elaboración propia, empleando Visio 2013.

La dirección IP de destino de todos los paquetes IP que ingresan al LSR de ingreso se buscará en la tabla de reenvío de IP. Todas estas direcciones pertenecen a un conjunto de prefijos que se conocen en la tabla de enrutamiento como prefijos BGP. Muchos prefijos BGP en la tabla de enrutamiento tienen la misma dirección BGP del siguiente salto, es decir, una salida LSR. Todos los paquetes con una dirección IP de destino para la cual la búsqueda IP en la tabla de enrutamiento recurre a la misma dirección BGP del siguiente salto se asignará a la misma FEC. Como ya se mencionó, todos los paquetes que pertenecen a la misma FEC obtienen la misma etiqueta impuesta por el ingreso LSR.

3.5.17. Distribución de etiquetas

La primera etiqueta se impone en el ingreso LSR y la etiqueta pertenece a un LSP. La ruta del paquete a través de la red MPLS está ligada a ese único LSP. Todo lo que cambia es que la etiqueta superior de la pila de etiquetas se intercambia en cada salto. El ingreso LSR impone una o más etiquetas en el paquete. Los LSR intermedios intercambian la etiqueta superior (la etiqueta entrante) del paquete etiquetado recibido con otra etiqueta (la etiqueta saliente) y transmiten el paquete en el enlace saliente. El LSR de salida del LSP elimina las etiquetas de está LSP y reenvía el paquete.

Se considera un caso en el que IPv4 está sobre MPLS simple, que es un ejemplo más simple de una red MPLS. Simple IPv4 sobre MPLS es una red que consta de LSR que ejecutan un protocolo *IPv4 Interior Gateway* (IGP) (por ejemplo, *Open Shortest Path First [OSPF], Intermediate System-to-Intermediate System [IS-IS], y Enhanced Interior Gateway Routing Protocol [EIGRP*]. El ingreso LSR busca la dirección IPv4 de destino del paquete, impone una etiqueta y reenvía el paquete. El siguiente LSR (y cualquier otro LSR intermedio) recibe el paquete etiquetado, intercambia la etiqueta entrante con una etiqueta saliente y reenvía el paquete. El LSR de salida muestra la etiqueta y reenvía el paquete IPv4 sin etiquetas en el enlace de salida. Para que esto funcione, los LSR adyacentes deben acordar que etiqueta usar para cada prefijo IGP. Por lo tanto, cada LSR intermedio debe ser capaz de determinar con qué etiqueta de salida debe intercambiarse la etiqueta entrante. Esto significa que necesita un mecanismo para indicar a los routers qué etiquetas usar al reenviar un paquete. Las etiquetas son locales para cada par de routers adyacentes. Las etiquetas no tienen un significado global en la red. para que los routers adyacentes acuerden que etiqueta usar para que prefijo, necesitan alguna forma de comunicación entre ellos; de lo contrario, los routers no saben

que etiqueta de salida necesita coincidir con que etiqueta entrante. Se necesita un protocolo de distribución de etiquetas.

Se puede distribuir etiquetas de dos maneras:

- Agarre las etiquetas en un protocolo de enrutamiento IP existente
- Tener un protocolo separado para distribuir etiquetas

3.5.18. Agarre de etiquetas en un protocolo de enrutamiento IP existente

El primer método tiene la ventaja de no se necesita un nuevo protocolo para ejecutarse en los LSR, pero cada protocolo de enrutamiento IP existente debe extenderse para que lleve las etiquetas. Esto no siempre es una cosa fácil de hacer. La gran ventaja de tener el protocolo de enrutamiento con las etiquetas es que el enrutamiento y la distribución de etiquetas siempre están sincronizados, lo que significa que no se puede tener una etiqueta si falta prefijo o viceversa. También, elimina la necesidad de otro protocolo ejecutándose en el LSR para hacer la distribución de etiquetas. La implementación de los protocolos de enrutamiento vector distancia (como EIGRP) es sencilla, ya que cada router origina un prefijo de su tabla de enrutamiento. El *router* simplemente une una etiqueta a ese prefijo.

Los protocolos de enrutamiento de estado del enlace (como IS-IS y OSPF) no funcionan de esta manera. Cada *router* origina actualizaciones de estado de enlace que luego son enviadas sin cambios por todos los routers dentro de un área. El problema es que para que MPLS funcione, cada *router* necesita distribuir una etiqueta para cada prefijo IGP, incluso los routers que no son creadores de ese prefijo. Los protocolos de enrutamiento de estado de enlace

deben empaparse de forma intrusiva para poder hacer esto. El hecho de que un router necesite anunciar una etiqueta para un prefijo que no se origina es contradictorio con la forma en que los protocolos de enrutamiento de estado de enlace funcionan de todos modos. Por lo tanto, para los protocolos de enrutamiento de estado de enlace, se prefiere un protocolo separado para distribuir etiquetas.

Ninguno de los IGP se ha cambiado para implementar el primer método. Sin embargo, BGP es un protocolo de enrutamiento que puede llevar prefijos y distribuir etiquetas al mismo tiempo. Sin embargo, BGP no es un IGP; se usa para llevar prefijos externos. BGP se utiliza principalmente para la distribución de etiquetas en redes MPLS VPN.

3.5.19. Ejecución de un protocolo separado para la distribución de etiquetas

El segundo método (ejecutar un protocolo separado para la distribución de etiquetas) tiene la ventaja de ser un protocolo de enrutamiento independiente. Cualquiera que sea el protocolo de enrutamiento IP, ya sea que sea capaz de distribuir etiquetas o no, un protocolo separado distribuye las etiquetas y permite que el protocolo de enrutamiento distribuye los prefijos. La desventaja de este método es que se necesita un nuevo protocolo en los LSR.

La elección de todos los proveedores de *routers* era tener un nuevo protocolo de distribución de etiquetas para distribuir las etiquetas para los prefijos IGP. este es el protocolo de distribución de etiquetas (*label distribution protocol,* LDP); sin embargo, no es el único protocolo que puede distribuir etiquetas MPLS.

Varias variedades de protocolos distribuyen etiquetas:

- *Tag distribution protocol* (TDP)
- Protocolo de distribución de etiquetas (LDP)
- Protocolo de reserva de recursos (RSVP)

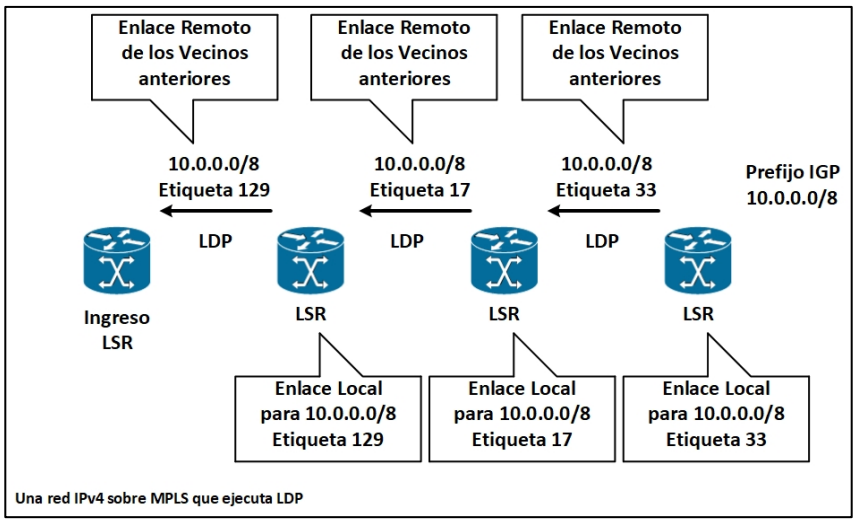
TDP, que es anterior a LDP, fue el primer protocolo para la distribución de etiquetas desarrollando e implementando por Cisco. Sin embargo, TDP es propiedad de Cisco. El IETF luego formalizo el LDP. LDP y TDP son similares en la forma en que operan, pero LDP tiene más funcionalidad que TDP. con la disponibilidad generalizada de LDP en lanzamientos de Cisco IOS de despliegue general, TDP fue reemplazado rápidamente por LDP. El resultado es que TDP se está volviendo obsoleto.

3.5.20. Distribución de etiquetas con LDP

Para cada prefijo IGP IP en su tabla de enrutamiento IP, cada LSR crea un enlace local, es decir, vincula una etiqueta al prefijo IPv4. El LSR luego distribuye está enlace a todos los vecinos LDP. Estas vinculaciones recibidas se convierten en enlaces remotos. Los vecinos almacenan estos enlaces remotos y locales en una tabla especial, la base de información de etiquetas (LIB). Cada LSR tiene solo un enlace local por prefijo, al menos cuando el espacio de etiqueta es por plataforma. Si el espacio de etiqueta es por interfaz, puede existir un enlace de etiqueta local por prefijo por interfaz. Por lo tanto, puede tener una etiqueta por prefijo o una etiqueta por prefijo por interfaz, pero el LSR obtiene más de un enlace remoto porque generalmente tiene más de un LSR adyacente.

De todos los enlaces remotos para un prefijo, el LSR necesita elegir solo uno y usar ese para determinar la etiqueta saliente para ese prefijo IP. La tabla de enrutamiento (a veces llamada la base de la instancia de enrutamiento o RIB) determina cual es el próximo salto del prefijo IPv4. El LSR elige la vinculación remota recibida del LSR indirecto, que es el siguiente salto en la tabla de enrutamiento para ese prefijo. utiliza esta información para configurar su base de información de reenvío de etiquetas (LFIB) donde la etiqueta del enlace local sirve como la etiqueta entrante y la etiqueta del único enlace remoto elegido a través de la tabla de enrutamiento sirve como etiqueta de salida. Por lo tanto, cuando un LSR recibe un paquete etiquetado, ahora es capaz de intercambiar la etiqueta entrante que le asignó, con la etiqueta de salida asignada por el LSR del siguiente salto adyacente. La figura 44 muestra el anuncio por LDP de las vinculaciones entre los LSR para el prefijo 10.0.0.0/8 de IPv4. Cada LSR asigna una etiqueta por prefijo IPv4. El enlace local es que está en el prefijo y su etiqueta asociada.

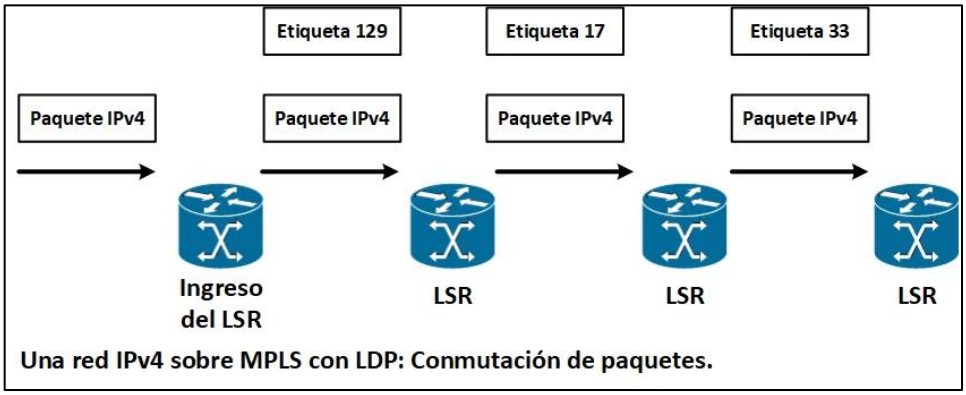
Figura 43. **Una red IPv4 sobre MPLS que ejecuta LDP**



Fuente: elaboración propia, empleando Visio 2013.

La figura 45 muestra el paquete IPv4 destinado a 10.0.0.0/8 que entra en la red MPLS en el ingreso LSR, donde se impone con la etiqueta 129 y se cambia al siguiente LSR. El segundo LSR intercambia la etiqueta saliente 17 y reenvía el paquete hacia el tercer LSR. El tercer LSR intercambia la etiqueta saliente 17 y reenvía el paquete hacia el tercer LSR. El tercer LSR intercambia la etiqueta entrante 17 con la etiqueta saliente 33 y reenvía el paquete al siguiente LSR y así sucesivamente.

Figura 44. **Red IPv4 sobre MPLS con LDP: conmutación de paquetes**



Fuente: elaboración propia, empleando Visio 2013.

3.5.21. Base de instancia de reenvío de etiquetas

El LFIB (*label forwarding instance base*) es la tabla utilizada para reenviar paquetes etiquetados. Se rellena con las etiquetas entrantes y salientes para los LSP. La etiqueta entrante es la etiqueta del enlace local en el LSR particular. La etiqueta de salida es la etiqueta de la vinculación remota elegida por el LSR de todos los enlaces remotos posibles. Todos estos enlaces remotos se encuentran en el LIB. El LFIB elige solo una de las posibles etiquetas de salida de todos los posibles enlaces remotos en el LIB y la instala en el LFIB. La

etiqueta remota elegida depende de que ruta es la mejor ruta encontrada en la tabla de enrutamiento.

En el ejemplo de MPLS sobre IPv4, la etiqueta está vinculada a un prefijo IPv4. Sin embargo, el LFIB puede completarse con etiquetas que LDP no asigna. En el caso de la ingeniería de tráfico MPLS, RSVP distribuye las etiquetas, el LFIB siempre se usa para reenviar un paquete de etiquetado entrante.

3.5.22. Carga útil de MPLS

La etiqueta MPLS no tiene un campo de identificador de protocolo de nivel de red. Está campo está presente en todos los cuadros de capa 2 para indicar que es el protocolo de capa 3. ¿Cómo sabe el LSR cuál es el protocolo detrás de la pila de etiquetas?, en otras palabras, ¿cómo sabe el LSR cuál es la carga útil de MPLS? la mayoría de los LSR no necesitan saber, porque recibirán un paquete etiquetado, cambiarán la etiqueta superior y enviarán el paquete en el enlace saliente. Este es el caso para LSR intermedios o routers P.

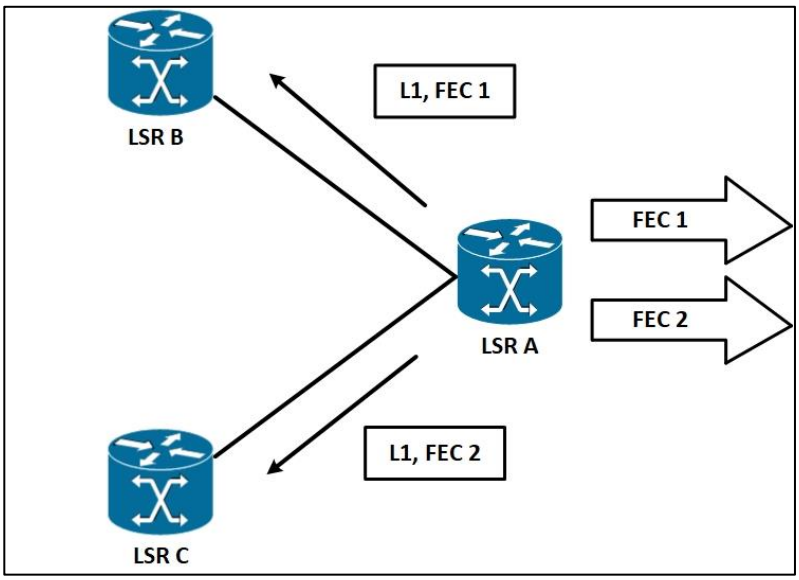
Los LSR intermedios no necesitan saber cuál es la carga útil de MPLS porque se conoce toda la información necesaria para cambiar el paquete mirando solo la etiqueta superior. Si la pila de etiquetas consta de más de una etiqueta, las etiquetas debajo de la etiqueta superior podrían no ser asignadas por el LSR y, por lo tanto, el LSR intermedio podría no tener conocimiento de lo que son. Además, es posible que el LSR no sepa cuál es la carga útil de MPLS transportada. Debido a que los LSR intermedios miran solo a la etiqueta superior para tomar una decisión de reenvío, esto no es un problema. Para que el reenvío basado en la etiqueta superior sea correcto, el LSR intermedio debe tener un enlace local y remoto para la etiqueta superior.

Un LSR de salida que está eliminando todas las etiquetas en la parte superior del paquete deben saber cuál es la carga útil MPLS, ya que debe reenviar la carga útil MPLS. El LSR de salida debe saber qué valor usar para el campo identificador del protocolo de nivel de red en el cuadro saliente. Ese LSR de salida es el que hizo el enlace local, lo que significa que ese LSR asignó una etiqueta local a ese FEC, y es esa etiqueta la que se usa como etiqueta entrante en el paquete. Por lo tanto, el LSR de egreso sabe cuál es la carga útil de MPLS mirando la etiqueta, porque es el LSR de egreso el que creó el enlace de la etiqueta para ese FEC, y sabe qué es ese FEC.

3.5.23. Espacios de etiquetas MPLS

En la figura 45, LSR A puede anunciar la etiqueta L1 para FEC 1 a LSR B y la etiqueta L1 para FEC 2 a LSR C, pero solo si LSR A puede distinguir posteriormente de que LSR se recibió el paquete con la etiqueta L1. En el caso de que LSR B y LSR C estén conectados directamente a LSR A a través de enlaces punto a punto, esto puede lograrse fácilmente mediante la implementación de MPLS en LSR. El hecho de que la etiqueta L1 sea única por interfaz da su nombre a está ámbito de etiqueta: espacio de etiqueta por interfaz. Si se utiliza el espacio de etiqueta por interfaz, el paquete no se reenvía únicamente en función de la etiqueta, sino que se basa tanto en la interfaz de entrada como en la etiqueta.

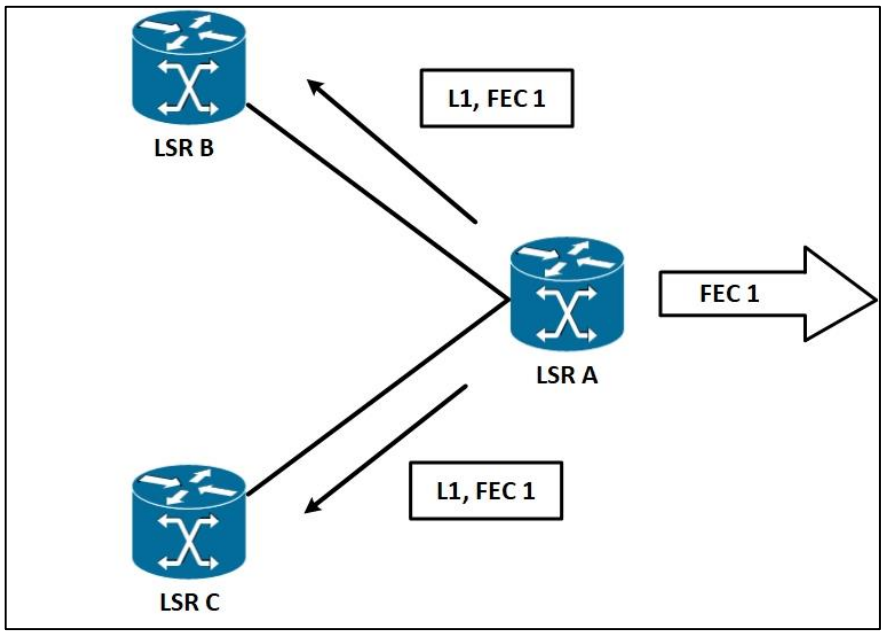
Figura 45. **Espacio de etiqueta por interfaz**



Fuente: elaboración propia, empleando Visio 2013.

La otra posibilidad es que la etiqueta no sea única por interfaz, sino por la LSR que asigna la etiqueta. esto se llama espacio de etiqueta por plataforma. En ese caso, LSR A distribuye FEC 1 con la etiqueta L1 a LSR B y C, como se observa en la figura 46. Cuando LSR A distribuye una etiqueta para FEC 2, está etiqueta debe ser una etiqueta diferente a la etiqueta L1. Si se usa el espacio de etiqueta por plataforma, el paquete se reenvía únicamente en función de la etiqueta, independientemente de la interfaz de entrada.

Figura 46. **Espacio de etiquetas por plataforma**



Fuente: elaboración propia, empleando Visio 2013.

3.5.24. Diferentes modos MPLS

Un LSR puede usar diferentes modos cuando distribuye etiquetas a otros LSR. Está sección cubre tres modos distintos, de la siguiente manera:

- Modo de distribución de etiquetas
- Modo de retención de etiqueta
- Modo de control LSP

3.5.25. Modo de distribución de etiquetas

La arquitectura MPLS tiene dos modos para distribuir enlaces de etiquetas:

- Modo de distribución de etiquetas *downstream-on-demand* (DoD).
- Modo de distribución de etiquetas decentes no solicitadas (UD, *unsolicited downstream*).

En el modo DoD, cada LSR solicita su LSR de siguiente salto (es decir, en sentido decente) en un LSP, un enlace de etiqueta para ese FEC. Cada LSR recibe un enlace por FEC solo desde su LSR en sentido descendente en ese FEC. El LSR indirecto es el router de siguiente salto indicando por la tabla de enrutamiento IP.

En el modo UD, cada LSR distribuye un enlace a sus LSR adyacentes, sin que esos LSR soliciten una etiqueta. En el modo UD, un LSR recibe un enlace de etiqueta remota de cada LSR adyacente.

En el caso de DoD, el LIB muestra solo un enlace remoto, mientras que en el caso de UD, es probable que vea más de uno. El modo de distribución de etiquetas utilizado depende de la interfaz y la implementación.

3.5.26. Modos de retención de etiquetas

Dos modos de retención de etiqueta son posibles:

- Modo de retención de etiqueta liberal (LLR)
- Modo *conservative label retention* (CLR)

En el modo LLR, un LSR mantiene todos los enlaces remotos recibidos en el LIB. Uno de estos enlaces es la vinculación remota recibida desde el flujo descendiente o siguiente salto para ese FEC. La etiqueta de ese enlace remoto se usa en el LFIB, pero ninguna de las etiquetas de los otros enlaces remotos

se coloca en el LFIB; por lo tanto, no todos se usan para reenviar paquetes. ¿Por qué mantener las etiquetas que no se utilizan? El enrutamiento es dinámico en una red. En cualquier momento, la topología de enrutamiento puede cambiar, por ejemplo, debido a que un enlace se desactiva o se elimina un router, por lo tanto, el router del siguiente salto para un FEC en particular puede cambiar. En ese momento, la etiqueta del nuevo router del siguiente salto ya está en el LIB y el LFIB se puede actualizar rápidamente con la nueva etiqueta saliente.

El segundo modo de retención de etiqueta es el modo CLR. Un LSR que ejecuta está modo no almacena todos los enlaces remotos en el LIB, pero almacena solo el enlace remoto asociado con el LSR del siguiente salto para un FEC en particular.

En resumen, el modo LLR le brinda una adaptación más rápida a los cambios de enrutamiento, mientras que el modo CLR le brinda menos etiquetas para almacenar y un mejor uso de la memoria disponible en el *router*.

3.5.27. Modos de control LSP

Los LSR pueden crear un enlace local para un FEC de dos maneras:

- Modo de control LSP independiente
- Modo de control LSP ordenado

El LSR puede crear un enlace local para un FEC independientemente de los otros LSR. Esto se llama modo de control LSP independiente. En está modo de control LSP independiente. En está modo de control, cada LSR crea un enlace local para un FEC en particular tan pronto como reconoce el FEC. Por lo

general, esto significa que el prefijo para el FEC está en su tabla de enrutamiento.

En el modo control LSP ordenado, un LSR solo crea un enlace local para un FEC sí reconoce que es el LSR de salida para el FEC o si el LSR ha recibido un enlace de etiqueta del siguiente salto para esta FEC.

La desventaja del control LSP independiente es que algunos LSR comienzan a etiquetar paquetes de conmutadores antes de que el LSP completo se configure de extremo a extremo; por lo tanto, el paquete no se reenvía de la manera que debería ser. Si el LSP no está completamente configurado, es posible que el paquete no reciba el tratamiento de reenvío correcto en todas partes o que incluso se elimine. Como ejemplo para ambos métodos de control, puede ver LSP como el método de distribución para los enlaces de etiquetas de los prefijos IGP. Si el LSR se estaba ejecutando en modo de control LSP independiente, asignará un enlace local para cada prefijo IGP en la tabla de enrutamiento. Si el LSR se ejecutaba en el modo control LSP ordenado, está LSR solo asignaría un enlace de etiqueta del router nexthop (como se indica en la tabla de enrutamiento).

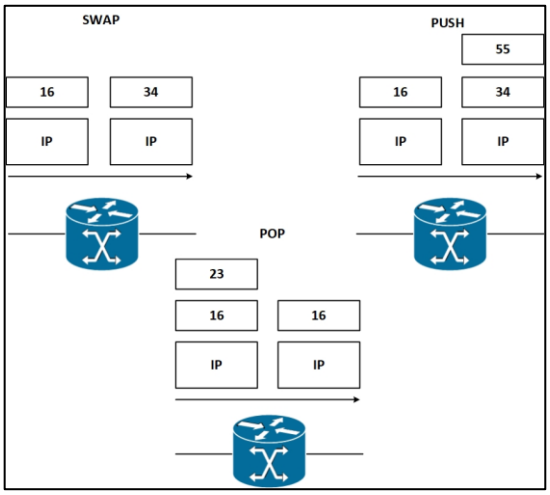
3.5.28. Reenvío de paquetes etiquetados

El reenvió de paquetes etiquetados es bastante diferente al reenvío de paquetes IP. No solo se reemplaza la búsqueda de IP con una búsqueda de la etiqueta en la base de información de reenvió de etiquetas (LFIB), sino que también son posibles diferentes operaciones de etiquetas. Estas operaciones se refieren a las operaciones *pop*, *push* y *swap* de etiquetas MPLS en la pila de etiquetas.

3.5.29. Operación de etiquetas

Las posibles operaciones de etiqueta son *swap*, *push* y *pop*. Observar la figura 47, para ver las posibles operaciones de etiquetas.

Figura 47. **Operación en etiquetas**



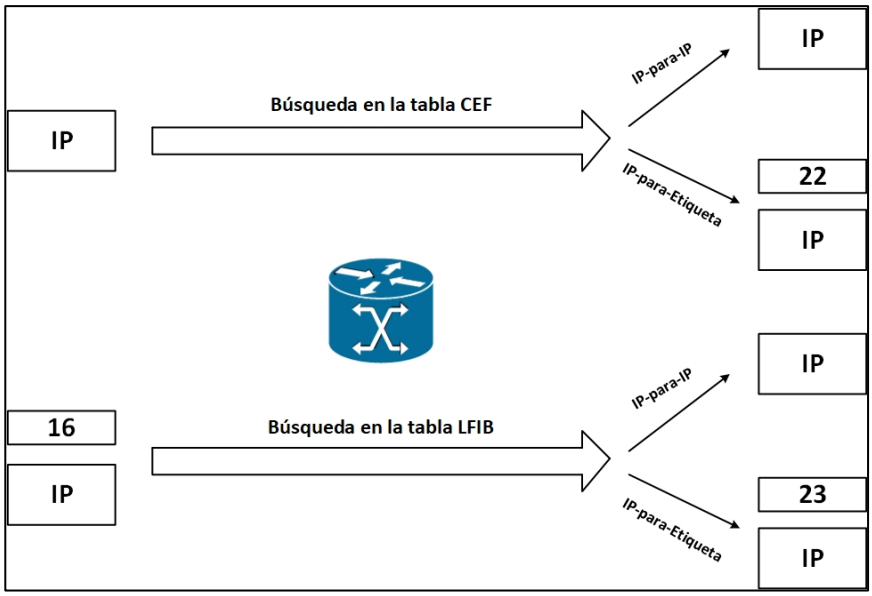
Fuente: elaboración propia, empleando Visio 2013.

Al mirar la etiqueta superior del paquete etiquetado recibido y la entrada correspondiente en el LFIB, el LSR sabe cómo reenviar el paquete. El LSR determina que operación de etiqueta debe realizarse (cambiar, pulsar o abrir) y cuál es el próximo salto al que se debe reenviar el paquete. La operación de intercambio significa que la etiqueta superior en la pila de etiquetas se reemplaza por otra, y la operación de inserción significa que la etiqueta superior se reemplaza por otra y luego una o más etiquetas adicionales se insertan en la pila de etiquetas. La operación pop significa que se quitó la etiqueta superior. El LSR ve el campo de 20 *bits* en la etiqueta superior, busca está valor en el LFIB e intenta hacer coincidir con un valor en la lista de etiquetas locales.

3.2.30. IP *lookup* versus *label lookup*

Cuando un router recibe un paquete IP, la búsqueda realizada es una búsqueda IP. En Cisco IOS, esto significa que el paquete se busca en la tabla CEF. Cuando un router recibe un paquete etiquetado, la búsqueda se realiza en el LFIB del router. El router sabe que recibe un paquete etiquetado o un paquete IP mirando el campo de protocolo en el encabezado de capa 2. Si un paquete es reenviado por Cisco Express Forwarding (CEF) (Búsqueda de IP) o por LFIB (búsqueda de etiqueta), el paquete puede dejar el router etiquetado o sin etiqueta. En la figura 48 se observa la diferencia entre una búsqueda en la tabla CEF y en la LFIB.

Figura 48. **Búsqueda en CEF o LFIB**



Fuente: elaboración propia, empleando Visio 2013.

Si un LSR de ingreso recibe un paquete de IP y lo reenvía como está etiquetado, reenvío de IP a etiqueta. Si un LSR recibe un paquete etiquetado, puede quitar las etiquetas y reenviarlo como un paquete IP, o puede reenviarlo como un paquete etiquetado. El primer caso se conoce como el caso de reenvío de etiqueta a IP; el segundo se conoce como el caso de reenvío de etiqueta a etiqueta.

En la figura 49 se muestra un caso de reenvío IP a etiqueta, es decir, el reenvío de paquete IP por tabla CEF.

Figura 49. **Ejemplo de una entrada en la tabla CEF**

```
Lactometer#show ip cef 10.200.254.4
10.200.254.4/32, version 44, epoch 0, cached adjacency 10.200.200.2
0 packets, 0 bytes
  tag information set, all rewrites owned
  local tag 20
  fast tag rewrite with Et0/0/0, 10.200.200.2, tags imposed {18}
Via 10.200.200.2, Ethernet0/0/0, 0 dependencies
  next hop 10.200.200.2, Ethernet0/0/0
  valid cached adjacency
  tag rewrite with Et0/0/0, 10.200.200.2, tags imposed {18}
```

Fuente: elaboración propia, empleando Visio 2013.

Los paquetes IP que entran en el LSR destinados a 10.200.254.4/32 salen en la interfaz Ethernet 0/0/0 después de ser impuestos con la etiqueta 18. El siguiente salto de está paquete es el 10.200.200.2. El reenvío de IP a etiqueta se realiza en el imponente LSR. En Cisco IOS, la conmutación CEF es el único modo de conmutación IP que puede usar para etiquetar paquetes. No se pueden usar otros modos de conmutación de IP, como la conmutación rápida, porque la memoria caché de conmutación rápida no contiene información en las

etiquetas. Debido a que la conmutación CEF es el único modo de conmutación IP compatible con MPLS, debe activar CEF cuando habilite MPLS en el *router*.

En la figura 50, se observa un extracto de LFIB, emitiendo el comando *show mpls forwarding-table*.

Figura 50. **Extracto de la LFIB**

```
Lactometer#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag     Outgoing     Next Hop
Tag    tag or VC   or Tunnel ID     switched      interface
16     Untagged    10.1.1.0/24      0             Et0/0/0      10.200.200.2
17     16          10.200.202.0/24  0             Et0/0/0      10.200.200.2
18     Pop tag     10.200.203.0/24  0             Et0/0/0      10.200.200.2
19     Pop tag     10.200.201.0/24  0             Et0/0/0      10.200.200.2
20     18          10.200.254.4/32  0             Et0/0/0      10.200.200.2
21     Pop tag     10.200.254.2/32  0             Et0/0/0      10.200.200.2
22     17          10.200.254.3/32  0             Et0/0/0      10.200.200.2
24     Untagged    12ckt(100)       4771050       Fa9/0/0      point2point
```

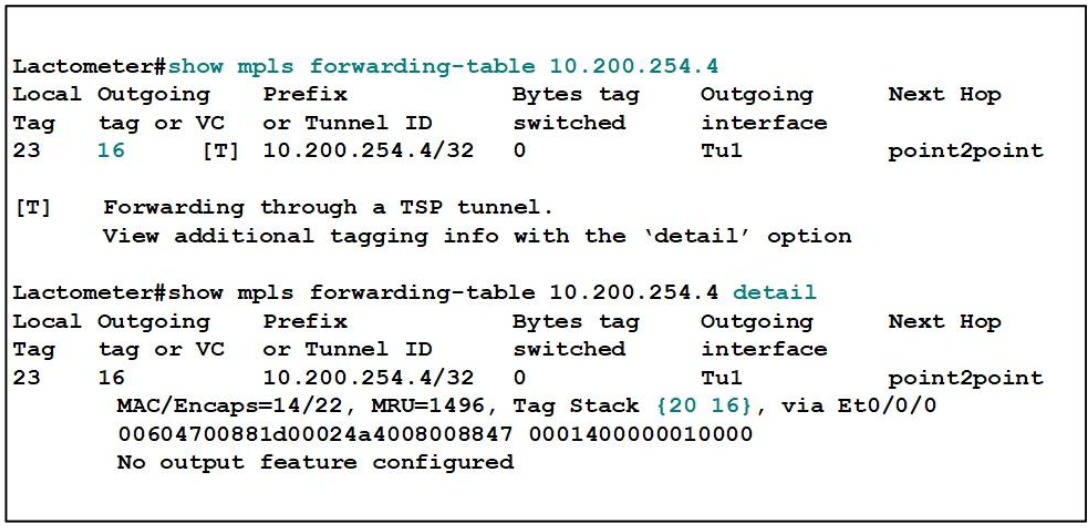
Fuente: elaboración propia, empleando Visio 2013.

La etiqueta local (o etiqueta) es la etiqueta que este LSR asigna y distribuye a los LSR. Como tal, está LSR espera que los paquetes etiquetados lleguen a él con estas etiquetas como las más altas en la pila de etiquetas. Si está LSR recibiera un paquete etiquetado en la etiqueta superior 22, cambiaría la etiqueta con la etiqueta 17 y luego la reenviaría a la interfaz Ethernet 0/0/0. Está es un ejemplo de *label-to-label forwarding case*.

Si está LSR recibe un paquete con la etiqueta superior 16, elimina todas las etiquetas y reenvía el paquete como un paquete IP, porque la etiqueta saliente (etiqueta) no está etiquetada. Este es un ejemplo del caso de etiqueta a IP. Si el LSR recibe un paquete con la etiqueta superior 18, elimina la etiqueta superior (abre una etiqueta) y reenvía el paquete como un paquete etiquetado o

como un paquete IP. Puede ver en este resultado algunos ejemplos de la operación de intercambio y *pop*. La figura 51 muestra una operación de inserción. La etiqueta entrante 23 se intercambia con la etiqueta 20, y la etiqueta 16 se empuja sobre la etiqueta 20.

Figura 51. **Ejemplo del comando MPLS *forwarding-table* (detallado)**



Fuente: elaboración propia, empleando Visio 2013.

Para ver todas las etiquetas que cambian en un paquete ya etiquetado, debe usar el comando *show mpls forwarding-table [network {mask | length}] [detail*]. En la figura 52, puede ver la diferencia entre la salida de está comando con y sin la palabra clave de detalle. si se especifica la palabra clave detail, puede ver todas las etiquetas que cambian en la pila de etiquetas. De izquierda a derecha entre {}, vera la primera etiqueta, que es la etiqueta intercambiada (20), y luego la etiqueta presionada (16) en la etiqueta intercambiada. Sin la palabra clave *detail*, solo vera la etiqueta presionada (16).

La operación agregada permanece. Cuando realiza una agregación (en resumen) en un LSR, anuncia una etiqueta específica para el prefijo agregado, pero la etiqueta saliente en el LFIB muestra 'agregado'. Debido a que este LSR está agregando un rango de prefijos, no puede reenviar un mensaje entrante. Paquete etiquetado mediante el intercambio de etiquetas de la etiqueta superior. La entrada de etiqueta saliente que muestra 'Agregado' significa que el LSR agregado necesita eliminar la etiqueta del paquete entrante y debe realizar una búsqueda IP para determinar el prefijo más específico que se utilizará para reenviar está paquete IP. El ejemplo muestra una entrada en el LFIB en un *router Provider Edge* (PE) de egreso en una red MPLS VPN. El LSR de salida que recibe un paquete con la etiqueta 23 eliminaría esa etiqueta y realizaría una búsqueda de IP en la dirección IP de destino en el encabezado IP.

Figura 52. **Ejemplo de una entrada en la LFIB para un prefijo VPN de MPLS**

```
singularity#show mpls forwarding-tablevrf cust-one
Local Outgoing    Prefix            Bytes tag     Outgoing      Next Hop
Tag   tag or VC   or Tunnel ID      switched      interface
23    Aggregate   10.10.1.0/24[V]   0
```

Fuente: elaboración propia, empleando Visio 2013.

Ahora sabe cómo el paquete etiquetado se reenvía a un siguiente salto específico después de una operación de etiqueta. Sin embargo, la tabla de adyacencia de CEF determina la encapsulación de enlace de datos salientes. La tabla de adyacencia proporciona la información necesaria de capa 2 para reenviar el paquete al LSR del siguiente salto.

La figura 53 muestra una tabla de adyacencia en un LSR. La tabla de adyacencia contiene la información de capa 2 necesaria para cambiar un marco en el alcance de datos de salida.

Figura 53. **Ejemplo de una tabla de adyacencia**

```
lactometer#show adjacency detail
Protocol      Interface                 Address
IP            Ethernet0/0/0             10.200.200.2(13)
                                        0 packets, 0 bytes
                                        epoch 0
                                        sourced in sev-epoch 4
                                        Encap length 14
                                        00604700881D00024A40080000800
                                        ARP
TAG           Ethernet0/0/0             10.200.200.2(9)
                                        231 packets, 22062 bytes
                                        epoch 0
                                        sourced in sev-epoch 4
                                        Encap length 14
                                        00604700881D00024A4008008847
                                        ARP
IP            Serial0/1/0               point2point (10)
                                        258 packets, 35612 bytes
                                        epoch 0
                                        sourced in sev-epoch 4
                                        Encap length 4
                                        0F00800
                                        P2P-ADJ
TAG           Serial0/1/0               point2point(5)
                                        0 packets, 0 bytes
                                        epoch 0
                                        sourced in sev-epoch 4
                                        Encap length 4
                                        0F008847
                                        P2P-ADJ
```

Fuente: elaboración propia, empleando Visio 2013.

Las operaciones de la etiqueta son:

- *Pop*: se quita la etiqueta superior. El paquete se reenvía con la pila de etiquetas restante o como un paquete sin etiqueta.

- *Swap*: la etiqueta superior se reemplaza con una nueva etiqueta (intercambiada) y se agregan (empujan) una o más etiquetas en la parte superior de la etiqueta intercambiada.

- *Untagged/No Label*: la pila se elimina y el paquete se reenvía sin etiqueta.

- *Aggregate*: la pila de etiquetas se elimina y se realiza una búsqueda de IP en el paquete de IP.

3.5.30. Paquetes etiquetados de balanceo de cargas

Si existen múltiples rutas de igual costo para un prefijo IPv4, Cisco IOS puede equilibrar la carga de los paquetes etiquetados, como se ilustra en la salida del IOS de Cisco de la figura 54. Puede ver la que las etiquetas entrantes/locales 17 y 18 tienen dos interfaces de salida, pero también pueden ser diferentes. Las etiquetas de salida son iguales si los dos enlaces están entre un par de routers y ambos pertenecen al espacio de etiqueta de la plataforma. Si existen varios LSR de siguiente salto, la etiqueta saliente para cada ruta suele ser diferente, porque los LSR del siguiente salto asignan etiquetas de forma independiente.

Figura 54. **Ejemplo de paquetes etiquetados con balanceo de cargas**

```
horizon#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag     Outgoing      Next Hop
Tag    tag or VC   or Tunnel ID     switched      interface
17     Pop tag     10.200.254.3/32  252           Et1/3         10.200.203.2
       Pop tag     10.200.254.3/32  0             Et1/2         10.200.201.2
18     16          10.200.254.4/32  10431273      Et1/2         10.200.201.2
       16          10.200.254.4/32  238           Et1/3         10.200.203.2
```

Fuente: elaboración propia, empleando Visio 2013.

Si se puede acceder a un prefijo a través de una combinación de rutas etiquetadas y no etiquetadas (IP), Cisco IOS no considera las rutas no etiquetadas para equilibrar la carga de los paquetes etiquetados. Esto se debe a que, en algunos casos, el tráfico que pasa por la ruta no etiquetada no llega a su destino. En el caso de MPLS sobre IPv4 simple (MPLS que se ejecuta en una red IPv4), los paquetes llegan al destino incluso si no están etiquetados. Los paquetes quedan sin etiqueta en el enlace donde MPLS no está habilitado y se convierten en etiquetado de nuevo en el siguiente enlace donde está habilitado MPLS. En el lugar donde los paquetes no están etiquetados, debe realizarse una búsqueda de IP. Debido a que la red ejecuta IPv4 en todas partes, debe poder entregar el paquete a su destino sin una etiqueta. Sin embargo, en algunos escenarios, como MPLS VPN o Any Transport over MPLS (AToM), un paquete que se convierte en no etiquetado en la red MPLS en un determinado enlace no llega a su destino final.

En el ejemplo de MPLS VPN, la carga útil MPLS es un paquete IPv4, pero los *routers* P normalmente no tienen las tablas de enrutamiento VPN, por lo que no pueden enrutar el paquete a su destino. En el caso de AToM, la carga útil de MPLS es un marco de capa 2; por lo tanto, si el paquete pierde su pila de etiquetas en un *router* P, el *router* P no tiene presentes las tablas de remisión de capa 2 para reenviar aún más el datagrama. Está es la razón por la cual en una red MPLS los paquetes etiquetados no tienen equilibrio de carga sobre IP y una ruta etiquetada. En general, la inteligencia para reenviar la carga útil de MPLS solo está en LSR (o PE) de borde. Por lo tanto, un *router* P no puede, en la mayoría de los casos, reenviar un paquete que se convierte en no etiquetado.

La figura 55 muestra el equilibrio de carga a través de dos rutas etiquetadas. Entonces el LDP se desactiva en uno de los dos enlaces salientes,

y ese enlace se elimina como un siguiente salto en el LFIB. El comando *no mpls ip* en una interfaz deshabilita LDP en esa interfaz.

Figura 55. **Cambiando un camino a un estado sin etiqueta**

```
horizon#show mpls forwarding-table 10.200.254.4
Local Outgoing    Prefix            Bytes tag     Outgoing      Next Hop
Tag   tag or VC   or Tunnel ID      switched      interface
18    18          10.200.254.4/32   56818         Et1/2         10.200.201.2
      18          10.200.254.4/32   160           Et1/3         10.200.203.2
horizon#conf t
Enter configuration commands, one per line. End with CNTL/Z.
horizon(config)#interface ethernet 1/3
horizon(config-if)#no mpls ip
horizon(config-if)#^Z
horizon#show mpls forwarding-table 10.200.254.4
Local Outgoing    Prefix            Bytes tag     Outgoing      Next Hop
tag   tag or VC   or Tunnel ID      switched      interface
18     16         10.200.254.4/32   10431273      Et1/2         10.200.201.2
       16         10.200.254.4/32   238           Et1/3         10.200.203.2
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.31. Etiqueta desconocida

En operación normal, un LSR debe recibir solo un paquete etiquetado con una etiqueta en la parte superior de la pila que el LSR conoce, porque el LSR debería haber anunciado previamente esa etiqueta. Sin embargo, es posible que algo salga mal en la red MPLS y el LSR no encuentra en su LFIB. El LSR teóricamente puede intentar dos cosas: quitar las etiquetas e intentar reenviar el paquete, o soltar el paquete. El Cisco LSR deja caer el paquete. Esto es lo correcto, porque está LSR no asignó la etiqueta superior, y no sabe qué tipo de paquete está detrás de la pila de etiquetas. ¿Es un paquete IPv4, IPv6, un marco de capa 2 o algo más? El LSR puede intentar resolverlo realizando una inspección de la carga útil de MPLS. Pero luego ocurre el mismo problema descrito anteriormente: es probable que el LSR en el que el paquete o el marco no está etiquetado no pueda buscar el destino del paquete o marco. Incluso si

el LSR intenta reenviar el paquete, no se garantiza que el paquete no se deje caer en un *router*. Lo único correcto es soltar un paquete entrante con una etiqueta superior desconocida.

3.5.32. Etiquetas reservadas

Las etiquetas 0 a 15 son etiquetas reservadas. Un LSR no puede usarlos en el caso normal para reenviar paquetes. Un LSR asigna una función específica a cada una de estas etiquetas. La etiqueta 0 es la etiqueta NULL explícita, mientras que la etiqueta 3 es la etiqueta NULL implícita. La etiqueta 1 es la etiqueta de alerta del router, mientras que la etiqueta 14 es la etiqueta de alerta OAM. Las otras reservadas entre 0 y 15 aún no se han asignado.

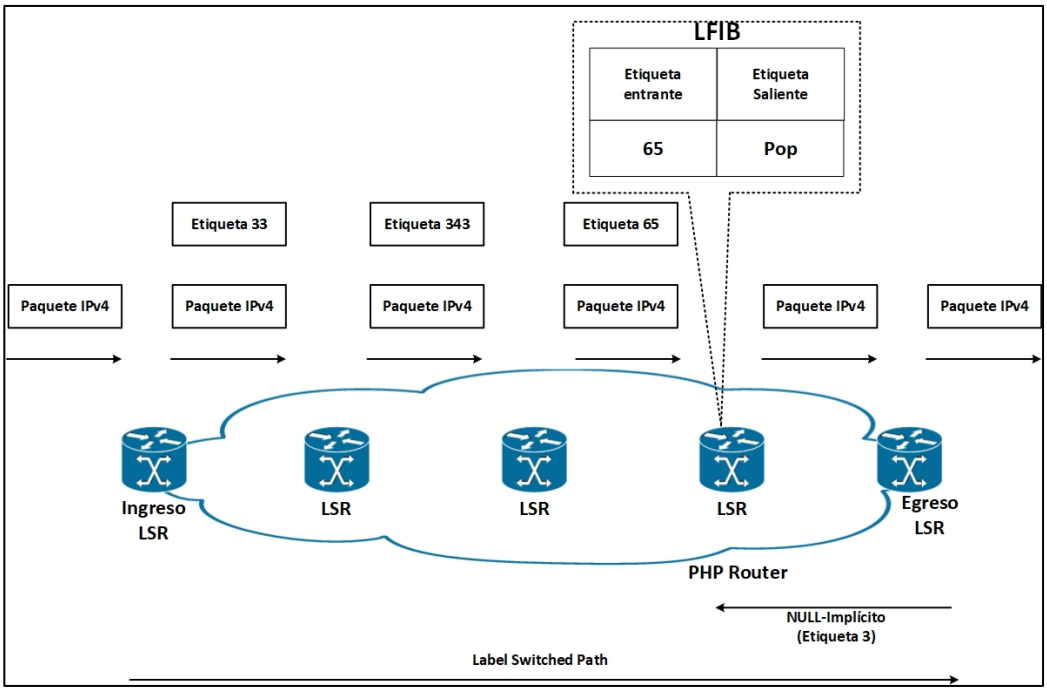
3.5.33. Etiqueta NULL implícita

La etiqueta NULL implícita es la etiqueta que tiene un valor de 3. Un LSR de salida asigna la etiqueta NULL implícita a un FEC si no desea asignar una etiqueta a ese FEC, solicitando de éste modo que el LSR en sentido ascendente realice una operación emergente. En el caso de una red simple IPv4 sobre MPLS, como una red IPv4 en la que LDP distribuye etiquetas entre los LSR, el IOS de Cisco que ejecuta LSR de salida asigna la etiqueta NULL implícita a sus prefijos conectados y resumidos. El beneficio de esto es que, si el LSR de egreso fuera a asignar una etiqueta para estos FEC, recibiría los paquetes con una etiqueta encima. Tendría que hacer dos búsquedas. Primero, tendría que buscar la etiqueta en el LFIB, solo para descubrir que la etiqueta debe ser eliminada; entonces tendría que realizar una búsqueda de IP. Estas son dos búsquedas, y la primera es innecesaria.

La solución para esta búsqueda doble es hacer que el LSR de salida señale el LSR último (o penúltimo) en la ruta de conmutación de etiqueta (LSP) para enviar los paquetes sin una etiqueta normal, sino al enviar la etiqueta especial con el valor 3. El resultado es que el LSR de salida recibe un paquete IP y solo necesita realizar una búsqueda IP para poder reenviar el paquete. Esto mejora el rendimiento en el LSR de salida.

El uso de NULL implícito al final de un LSP se llama *Penultimate Hop Popping* (PHP). La entrada LFIB para el LSP en el router PHP muestra una *Etiqueta Pop* como la etiqueta saliente. La figura 56 muestra el salto final.

Figura 56. **Penúltimo salto de popping**



Fuente: elaboración propia, empleando Visio 2013.

El uso de NULL implícito es generalizado y no se limita solo al ejemplo de la figura 50. Podría ser que los paquetes tengan dos o tres o más etiquetas en la pila de etiquetas. Luego, la etiqueta NULL implícita utilizada en el LSR de salida indicará al penúltimo router de salto que revele una etiqueta y envié el paquete etiquetado con una etiqueta menos al LSR de salida. Entonces, el LSR de salida no tiene que realizar dos búsquedas de etiquetas. El uso de la etiqueta NULL implícita no significa que todas las etiquetas de la pila de etiquetas deben eliminarse. Solo una etiqueta aparece. En cualquier caso, el uso de la etiqueta NULL implícita evita que el LSR de salida tenga que realizar dos búsquedas. Aunque el valor de etiqueta 3 señala el uso de la etiqueta NULL implícita, la etiqueta 3 nunca se verá como una etiqueta en la pila de etiquetas de un paquete MPLS. Es por eso que se llama etiqueta NULL implícita.

El uso de NULL implícito agrega inferencia al reenviar paquetes. Sin embargo, tiene una desventaja: el paquete se reenvía con una etiqueta menos de lo que fue recibido por el penúltimo LSR o sin etiqueta se se recibió con una sola etiqueta. Además del valor de la etiqueta, la etiqueta también contiene los bits experimentales (EXP). Cuando se elimina una etiqueta, los bits EXP también se elimina. Debido a que los bits EXP se usan exclusivamente para la calidad de servicio (QoS), la parte QoS del paquete se pierde cuando se elimina la etiqueta superior. En algunos casos, es posible que desee conservar está información de QoS y enviarla al LSR de salida. NULL implícito no se puede usar en ese caso.

La etiqueta NULL implícita es la solución a este problema, porque el LSR de salida señala la etiqueta NULL explícita de IPv4 (valor 0) al *router* de penúltimo salto. El LSR de salida recibe paquetes etiquetados con una etiqueta de valor 0 como etiqueta superior. El LSR no puede reenviar el paquete buscando el valor 0 en el LFIB porque puede asignarse a múltiples FEC. El LSR

simplemente elimina la etiqueta NULL explícita. Después de que el LSR elimine la etiqueta NULL explícita, debe realizarse otra búsqueda, pero la ventaja es que el router puede derivar la información de QoS del paquete recibido mirando los bits EXP de la etiqueta NULL explícita.

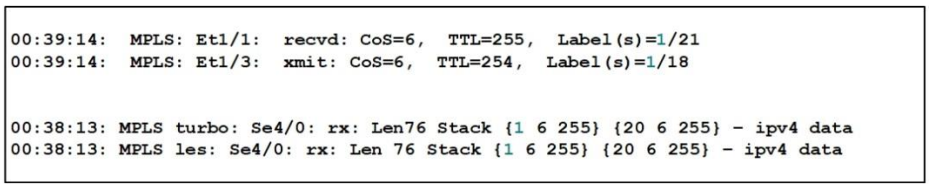
Puede copiar el valor de los bits EXP en los bits de presencia o DiffServ al realizar PHP y así conservar la información de QoS. O bien, si la pila de etiquetas tiene varias etiquetas y la etiqueta superior aparece, puede copiar el valor de los bits EXP en el campo EXP de la nueva etiqueta superior.

3.5.34. Etiqueta de alerta del *router*

La etiqueta de alerta del router es la que tiene el valor 1. Esta etiqueta puede estar presente en cualquier lugar de la pila de etiquetas, excepto en la parte inferior. Cuando la etiqueta del router alerta es la etiqueta superior, alerta al LSR que el paquete necesita una mirada más cercana. Por lo tanto, el paquete no se reenvía el hardware, sino que es examinado por un proceso de software. Cuando el paquete se reenvía, la etiqueta 1 se elimina. A continuación, se realiza una búsqueda de la siguiente etiqueta en la pila de etiquetas en el LFIB para decidir dónde debe cambiarse el paquete. A continuación, se realiza una acción de etiqueta (*pop, swap, push*), la etiqueta 1 se empuja hacia atrás en la parte superior de la pila de etiquetas y el paquete se reenvía.

La figura 57 muestra el resultado del paquete con el comando *debug mpls* en un *router* para un paquete etiquetado con la etiqueta de alerta del *router*.

Figura 57. **Depuración que muestra la etiqueta 1 en un paquete MPLS**



Fuente: elaboración propia, empleando Visio 2013.

La figura 57 muestra dos posibles formatos en la salida. Ambos formatos tienen las etiquetas ordenadas de izquierda a derecha o superior a la etiqueta inferior. El primer formato es el antiguo, con la barra que separa las etiquetas. El segundo formato es el nuevo formato {*label EXP TTL*}.

3.5.35. Etiquetas sin reserva

Excepto por las etiquetas reservadas de 0 a 15, puede usar todos los valores de etiqueta para el reenvío normal de paquetes. Como el valor de etiqueta tiene 20 *bits*, las etiquetas de 16 a 1 048 575 (220-1) se utilizan para el reenvío normal de paquetes. En Cisco IOS, el rango predeterminado es de 16 a 100 000. Esto es más que suficiente para etiquetar los prefijos IGP que tiene, pero si desea etiquetar los prefijos BGP, está número podría ser insuficiente. Puede cambiar el rango de etiqueta con el comando mpls label range min max. La figura 58 muestra cómo cambiar el rango de etiquetas mpls predeterminadas.

Figura 58. **Cambio de rango de etiquetas MPLS**

```
event#show mpls label range
Downstream Generic label region: Min/Max label: 16/100000

event#conf t
Enter configuration commands, one per line. End with CNTL/Z
event(config)#mpls label range ?
<16-1048575> Minimum label value
event(config)#mpls label range 16 ?
<16-1048575> Maximum label value
event(config)#mpls label range 16 1048575

event#show mpls label range
Downstream Generic label region: Min/Max label: 16/1048575
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.36. Comportamiento TTL de paquetes etiquetados

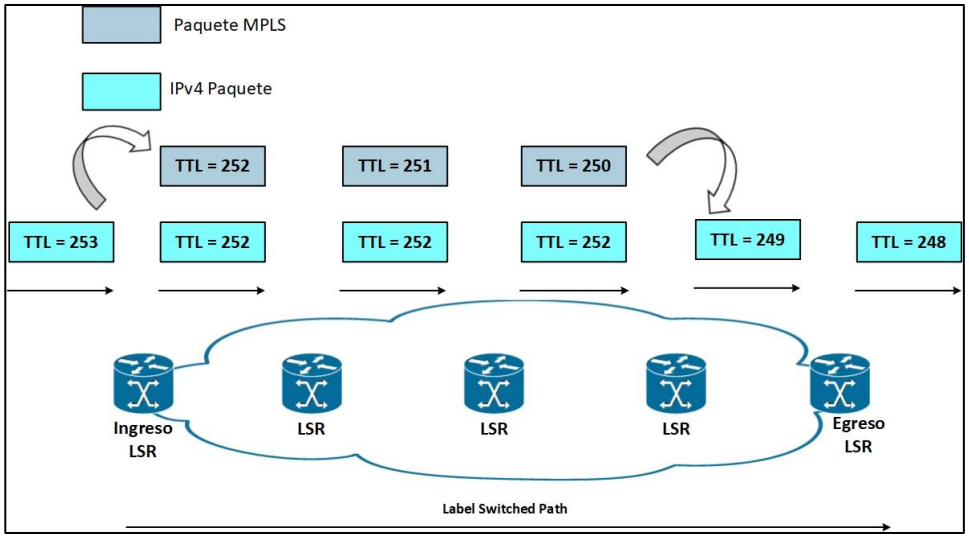
Time to live (TTL) es un mecanismo bien conocido gracias a IP. En el encabezado IP hay un campo de 8 bits que indica el tiempo que un paquete todavía tiene antes de que su vida termine y se descarta. Cuando se envía un paquete IP, su TTL suele ser de 255 y luego se disminuye en 1 en cada salto. Si el TTL llega a 0, el paquete se descarta. En tal caso, el *router* que eliminó el paquete IP para el cual el TTL alcanzó 0 envía un mensaje de tipo 11 del protocolo de mensajes de control de internet (ICMP) y un código 0 (tiempo excedido) al originador del paquete IP.

Con la introducción de MPLS, las etiquetas se agregan a los paquetes de IP. Esto requiere un mecanismo en el que el TTL se propaga desde el encabezado IP a la pila de etiquetas y viceversa. Esto garantiza que los paquetes no vivan para siempre al ingresar y salir de la nube de MPLS, si hay un bucle de enrutamiento.

3.5.37. Comportamiento TTL en el caso de IP a la etiqueta o etiqueta a la IP

En MPLS, el uso del campo TTL en la etiqueta es el mismo que el TTL en el encabezado IP. Cuando un paquete IP ingresa a la nube de MPLS, como en el LSR de ingreso, el valor IP TTL se copia (después de haber sido decrementado en 1) a los valores MPLS de la etiqueta presionada. En el LSR de salida, la etiqueta se elimina y el encabezado IP se expone nuevamente. El valor IP TTL se copia del valor MPLS TTL en la etiqueta superior recibida después de disminuirla en 1. Sin embargo, en Cisco IOS, una salvaguarda protege contra posibles bucles de enrutamiento al no copiar el TTL MPLS al TTL IP si el MPLS TTL es mayor que el IP TTL del paquete etiquetado recibido. Si el MPLS TTL se copiará en el encabezado IP, el menor valor de IP TTL se sobrescribirá con un valor más nuevo, pero más alto. Si el paquete IP se inyecta nuevamente en la nube MPLS, como el resultado de un bucle de enrutamiento, el paquete podría vivir para siempre porque el TTL nunca llegaría a 0. La figura 59 muestra el comportamiento predeterminado de copiar o propagar el TTL entre el encabezado IP y las etiquetas MPLS y viceversa.

Figura 59. **Comportamiento de propagación de TTL entre encabezado IP y etiquetas MPLS**

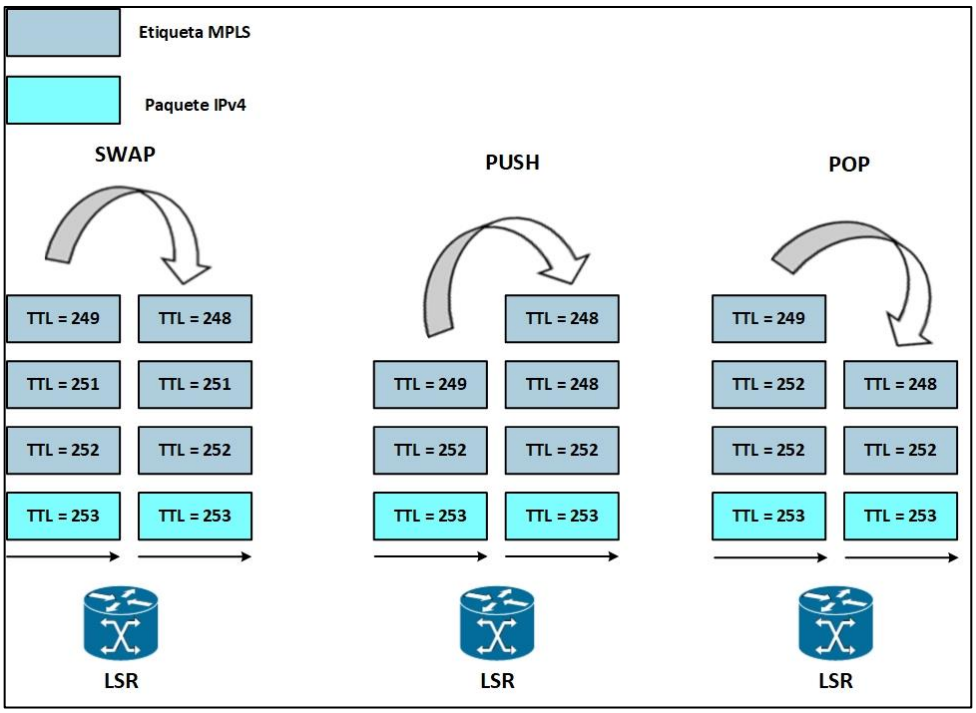


Fuente: elaboración propia, empleando Visio 2013.

3.5.38. Comportamiento TTL en el caso de etiqueta a etiqueta

Si la operación que se realiza en el paquete etiquetado es un intercambio, el TTL de la etiqueta entrante -1 se copia a la etiqueta intercambiada. Si la operación que se realiza en el paquete etiquetado es para empujar una o más etiquetas, el TTL MPLS recibido de la etiqueta superior -1 se copia a la etiqueta intercambiada y todas las etiquetas empujadas. Si la operación es emergente, el TTL de la etiqueta entrante -1 se copia a la etiqueta recién expuesta, a menos que ese valor sea mayor que el TTL de la etiqueta recién expuesta, en cuyo caso la copia no ocurre. La figura 60 muestra ejemplos de propagación TTL en el caso de la operación etiqueta a etiqueta para una operación de intercambio, inserción y *pop*.

Figura 60. **Propagacion TTL en la operación de etiqueta a etiqueta en el caso de una operación de intercambio, push y pop**



Fuente: elaboración propia, empleando Visio 2013.

El LSR intermedio no cambia el campo TTL en las etiquetas subyacentes o el campo TTL en el encabezado IP. Un LSR solo mira o solo cambia la etiqueta superior en la pila de etiquetas de un paquete. Este comportamiento TTL de los paquetes etiquetados que se describen aquí se refiere a la operación TTL en Cisco IOS.

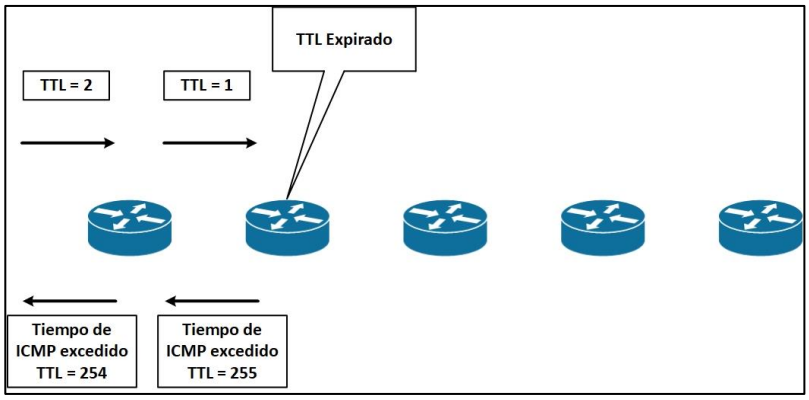
3.5.39. Expiración del TTL

Cuando se recibe un paquete etiquetado con un TTL de 1, el LSR receptor descarta el paquete y envía un mensaje ICMP 'tiempo excedido' (tipo 11, código

0) al que origina el paquete IP. está es el mismo comportamiento que un *router* exhibirá con un paquete IP que tenía un TTL que expira. Sin embargo, el mensaje ICMP no se envía de inmediato al originador del paquete porque un LSR provisional podría no tener una ruta IP hacia la fuente del paquete. El mensaje ICMP se reenvía a lo largo del LSP que estaba siguiendo el paquete original.

La figura 61 muestra un *router* que envía el mensaje ICMP 'tiempo excedido' al originador del paquete en el caso de una red IP.

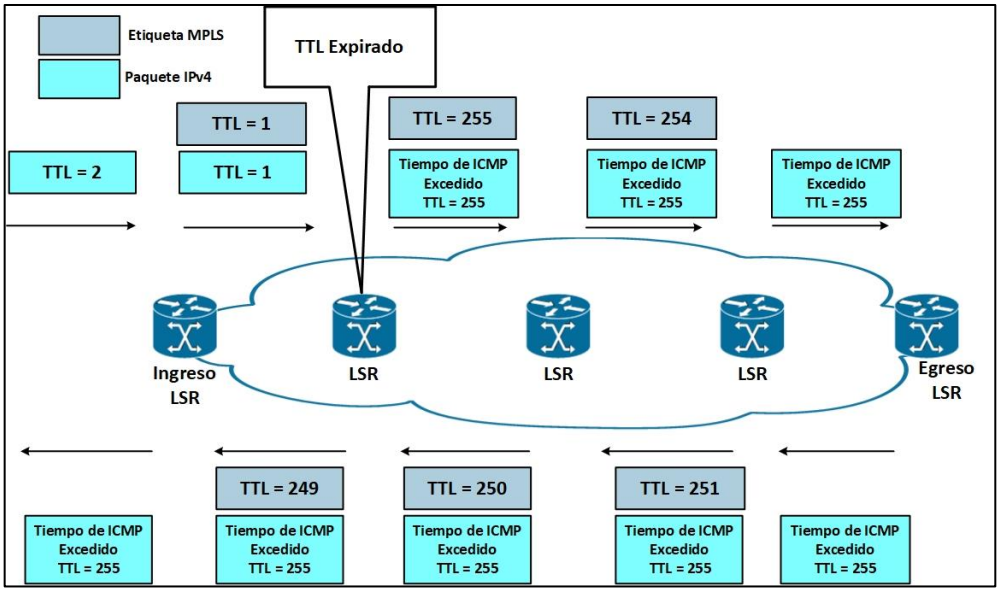
Figura 61. **ICMP, tiempo excedido devuelto por un enrutador en una red IP**



Fuente: elaboración propia, empleando Visio 2013.

En la figura 62, se muestra un LSR reenviando el mensaje ICMP 'tiempo excedido' a lo largo del LSP del paquete original.

Figura 62. **ICMP, tiempo excedido enviado por un router en una red MPLS**



Fuente: elaboración propia, empleando Visio 2013.

El motivo de este reenvío del mensaje ICMP a lo largo del LSP que estaba siguiendo el paquete original con el TTL que expira es que algunos casos el LSR que está generando el mensaje ICMP no tiene conocimiento como llegar al originador del paquete original. Igualmente, un LSR intermedio más cercano al originador del paquete podría no tener ese conocimiento. Uno de estos casos es una red con MPLS VPN. En este escenario el *router* P no tiene el conocimiento para devolver los mensajes ICMP al originador del paquete VPN, porque el router P no tiene una ruta para devolver directamente el mensaje ICMP. (En general, los *router* P no contienen las tablas de enrutamiento VPN). Por lo tanto, el *router* P crea el mensaje ICMP y reenvía el paquete a lo largo del LSP, con la esperanza de que el mensaje ICMP llegue a un router al final del LSP que puede devolver el paquete al enrutamiento de origen. En el caso

de MPLS VPN, el mensaje ICMP es devuelto por el PE de salida o el CE que está conectado a ese PE, porque estos routers ciertamente tienen la ruta para devolver el paquete correctamente.

Es importante que el *router* P, donde expira el TTL, observe cual es la carga útil de MPLS. El *router* P verifica si la carga útil es un paquete IPv4 (o IPv6). Si es así, puede generar el mensaje ICMP 'tiempo excedido' y reenviarlo a lo largo del LSP. Sin embargo, si la carga útil no es un paquete IPv4 (o IPv6), el *router* P no puede generar el mensaje ICMP. Por lo tanto, el *router* P descarta el paquete en todos los casos, excepto si se trata de un paquete IPv4 (o IPv6). Un caso en el que el LSR arroja un paquete con el vencimiento del TTL es AToM. La carga útil de MPLS en el caso de AToM es un marco de capa 2 y no un paquete de IP. Por lo tanto, si TTL en la etiqueta superior de un paquete AToM expira en un router P, la única acción que puede realizar el *router* P es soltar el paquete, porque no es posible realizar una búsqueda IP. El paquete también se descarta si la carga es útil es un paquete IPv6. Sin embargo, si el *router* P ejecuta un nuevo código Cisco IOS, que comprende el protocolo IPv6, ese *router* puede generar el paquete ICMP IPv6 tiempo excedido. Si el *router* P realmente tiene una ruta IPv6 apuntando al originador del paquete es irrelevante. Esto es así porque el mensaje ICMP siempre se reenvía a lo largo del LSP del paquete con el TTL que expira.

3.5.40. MPLS MTU

La unidad de transmisión máxima (MTU) es un parámetro bien conocido en el mundo de la IP. Indica el tamaño máximo del paquete IP que aún se puede enviar en un enlace de datos, sin fragmentar el paquete. Los enlaces de datos en redes MPLS también tienen una MTU específica, pero para paquetes etiquetados. Tomemos el caso de una red IPv4 que implementa MPLS. Todos

los paquetes IPv4 tienen una o más etiquetas. Esto implica que los paquetes etiquetados son ligeramente más grandes que los paquetes IP, porque para cada etiqueta, se agregan cuatro *bytes* al paquete. Entonces, si n es el número de etiquetas, se agregan n*4 bytes al tamaño del paquete cuando el paquete está etiquetado.

Esta sección explica que un parámetro MPLS MTU pertenece a los paquetes etiquetados. Además, explica que son los gigantes, los pequeños gigantes y los marcos gigantes de cómo garantizar los *switches* Ethernet puedan manejarlos. Finalmente, se introduce un nuevo parámetro: MPLS *maximum receive unit*. Este parámetro se utiliza en el LFIB para realizar un seguimiento de cómo los grandes paquetes etiquetados pueden ser reenviados sin necesidad de fragmentarlos.

3.5.41. Comando MPLS MTU

El comando de interfaz MTU en Cisco IOS especifica que tan grande puede ser un paquete de capa 3 sin tener que fragmentarlo al enviarlo en un enlace de datos. Para la encapsulación de Ethernet, por ejemplo, MTU se establece de forma predeterminada en 1 500. Sin embargo, cuando se agregan n etiquetas, se agregan n*4 bytes a un paquete IP de tamaño máximo ya de 1 500 *bytes*. Esto llevaría a la necesidad de fragmentar el paquete.

Cisco IOS tiene el comando mpls mtu que le permite especificar que tan grande puede ser un paquete etiquetado en un enlace de datos. Si, por ejemplo, sabe que todos los paquetes que se envían en el enlace tienen un máximo de dos etiquetas y la MTU es de 1 500 bytes, puede configurar MPLS MTU en 1 508 (1 500 + 2 * 4). Por lo tanto, todos los paquetes etiquetados de tamaño 1 508 bytes (etiquetas incluidas) puedan enviarse en el enlace sin

fragmentarlos. El valor MPLS MTU predeterminado de un enlace es igual al valor MTU. Observese la figura 63 para verificar cómo puede cambiar el MTU de MPLS en una interfaz en Cisco IOS.

Figura 63. **Cambio de MPLS MTU**

```
guatemala#show mpls interfaces fastEthernet 2/6 detail
Interface FastEthernet2/6:
         IP labeling enabled
         LSP Tunnel labeling not enabled
         BGP labeling not enabled
         MPLS not operational
         MTU = 1500
guatemala#conf t
Enter configuration commands, one per line. End with CNTL/Z
guatemala(config)#interface FastEthernet2/6
guatemala(config-if)#mpls mtu 1508
guatemala(config-if)#^Z
guatemala#
guatemala#show mpls interfaces fastEthernet 2/6 detail
Interface FastEthernet2/6:
         IP labeling enabled
         LSP Tunnel labeling not enabled
         BGP labeling not enabled
         MPLS not operational
         MTU = 1508
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.42. Unidad de recepción máxima MPLS

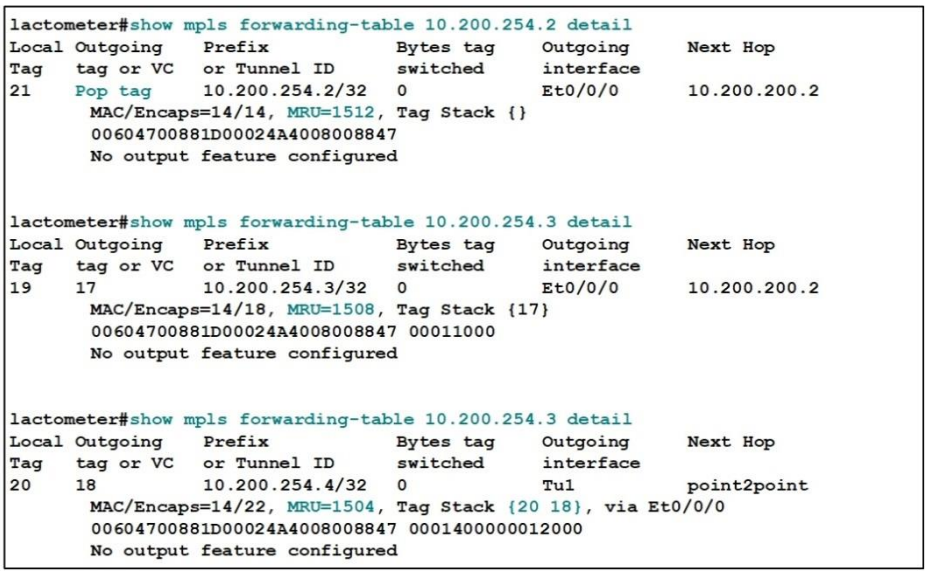
La unidad de recepción máxima (*maximum receive unit*, MRU) es un parámetro que utiliza Cisco IOS. Informa al LSR qué tan grande puede ser un paquete etiquetado recibido de un cierto FEC que todavía puede ser reenviado fuera de esta LSR sin fragmentarlo. Está valor es realmente un valor por FEC (o prefijo) y no solo por interfaz. La razón para esto es que la etiquetas se pueden agregar o eliminar de un paquete en un LSR.

Piense en el ejemplo de un router en el que todas las interfaces tienen una MTU de 1 500 *bytes*. Esto significa que el paquete IP más grande que se puede

recibir y transmitir en todas las interfaces es de 1 500 *bytes*. Imagine que los paquetes pueden etiquetarse agregando un máximo de dos etiquetas. (Típicamente, las redes MPLS VPN y AToM etiquetan los paquetes respectivamente las tramas con dos etiquetas). Suponga también que MPLS MTU se establece en 1 508 en todos los enlaces para acomodar los 8 *bytes* adicionales (2 veces 4 *bytes*) para las etiquetas. Un paquete etiquetado que se transmite en cualquiera de los enlaces ahora puede tener 1 508 *bytes*. Sin embargo, si la operación en el paquete entrante fuera POP, el paquete podría haber sido de 4 *bytes* o 1 etiqueta más grande (por lo tanto, 1 512 *bytes*) cuando se recibió, porque una etiqueta se habría quitado antes de transmitir el paquete. Sin embargo, si la operación de etiqueta fuera un impulso y se añadiera una etiqueta, el paquete entrante sólo podría haber tenido 1 504 *bytes*, porque se habrían agregado 4 *bytes* o una etiqueta, lo que haría que el paquete fuera de 1 508 *bytes*, antes de desconectar el paquete.

Como puede ver, la operación de etiqueta juega un papel en la determinación de la MRU. Debido a que la operación de etiqueta se determina por FEC o prefijo, la MRU puede cambiar por FEC o prefijo. Observe como en la figura 64 la MRU cambia por prefijo según la operación de etiqueta específica realizada en los paquetes. El LFIB le muestra el valor de la MRU por prefijo.

Figura 64. **Permitiendo tramas Jumbo en *switches* Ethernet**



Fuente: elaboración propia, empleando Visio 2013.

El MRU para el prefijo 10.200.254.2/32 es 1 512. El paquete recibido puede tener 1 512 *bytes*, porque una etiqueta se elimina antes de ser reenviada. El MRU para el prefijo 10.200.254.3/32 es 1 508. El tamaño del paquete no cambia, porque solo la etiqueta superior se intercambia. El MRU para el prefijo 10.200.254.4/32 es 1 504. El paquete recibido puede tener solo 1 504 *bytes* porque una etiqueta adicional se inserta en la pila de etiquetas antes de reenviar el paquete; por lo tanto, el tamaño del paquete aumenta en 4 bytes. La 'pila de etiquetas' muestra que una etiqueta se inserta en la pila de etiquetas después de la etiqueta entrante se intercambia.

3.5.43. Fragmentación de paquetes MPLS

Si un LSR recibe un paquete etiquetado que es demasiado grande para ser enviado en un enlace de datos, el paquete debe estar fragmentado. Esto

similar a fragmentar un paquete de IP. Si se escribe un paquete etiquetado y el LSR nota que la MTU saliente no es lo suficientemente grande para está paquete, el LSR elimina la pila de etiquetas, fragmenta el paquete IP, coloca la pila de etiquetas (después de la operación *pop, swap* o *push*) en todos los fragmentos, y reenvía los fragmentos. Solo si el encabezado IP tiene el conjunto de bits *do not fragment* (DF), el LSR no fragmenta el paquete IP, sino que descarta el paquete y devuelve un mensaje de error ICMP 'fragmentación necesaria y no fragmenta el conjunto de bits' (ICMP tipo 3, código 4) el creador del paquete IP. Al igual que con el mensaje ICMP 'tiempo excedido' (tipo 11, código 0), que se envía cuando el TTL caduca de un paquete etiquetado, se envía el mensaje ICMP 'fragmentación necesaria y no fragmenta el conjunto de *bits'*, utilizando una pila de etiquetas que es la pila de etiquetas salientes para el paquete que causó la creación del mensaje ICMP. Esto significa que el mensaje ICMP viaja más abajo que el LSP hasta que alcanza el LSR de salida de ese LSP. Luego se devuelve al creador del paquete con el *bit* DF configurado.

En general, la fragmentación causa un impacto en el rendimiento y debe evitarse. Un buen método para evitar la fragmentación es usar el método *Path MTU Discovery.*

3.5.44. Path MTU Discovery

Un método para evitar la fragmentación es *Path MTU Discovery*, que la mayoría de los host IP modernos realizan automáticamente. En ese caso, los paquetes IP enviados tienen el bit *do not fragment* (DF) fraguado. Cuando un paquete encuentra un router que no puede reenviar el paquete sin fragmentarlo, el router nota que el bit de DF está configurado, descarta el paquete y envía un mensaje de error de ICMP 'se necesita fragmentación y no se fragmenta el conjunto de *bits'* (ICMP tipo 3, código 4) al creador del paquete IP. El creador

del paquete IP luego reduce el tamaño del paquete y retransmite el paquete. Si aún existe un problema, el host puede reducir el tamaño del paquete nuevamente. Esto continúa hasta que no se recibe ningún mensaje ICMP para el paquete IP. El tamaño del último paquete IP enviado correctamente se utiliza como el tamaño máximo de paquete para todo el tráfico IP posterior entre el origen y el destino específicos; por lo tanto, es el MTU de la ruta.

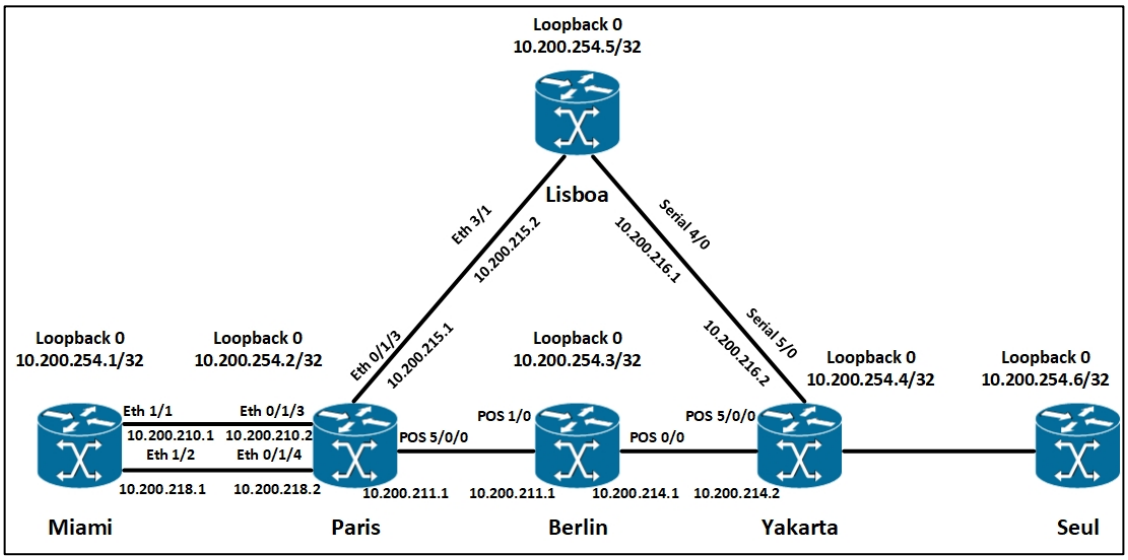
Path MTU Discovery no está garantizado para funcionar en todos los casos; a veces el mensaje ICMP no regresa al que lo origina. Las causas posibles para que el mensaje ICMP no llegue al creador del paquete son firewalls, listas de acceso y problemas de enrutamiento.

3.5.45. *Label distribution protocol*

La historia fundamental en MPLS es que los paquetes están etiquetados, y cada *router* de conmutación de etiquetas (LSR) debe realizar el intercambio de etiquetas para reenviar el paquete. Esto significa que, en todos los casos, las etiquetas deben distribuirse. Puede lograr esto de dos maneras: utilizar las etiquetas en un protocolo de enrutamiento existente o desarrollar un nuevo protocolo para hacer eso. Si desea ajustar el Protocolo de puesta de enlace interior (IGP), como *open shortest path first* (OSPF), *intermediate system-to-intermediate system* (IS-IS), *enhanced interior gateway routing protocol* (EIGRP), y *routing information protocol* (RIP): para llevar las etiquetas, debe hacerlo para todos los IGP, ya que todos ellos se utilizan como protocolos de enrutamiento en las redes actuales. Si escribe un protocolo nuevo desde cero, podría hacerlo independiente de la ruta y poder trabajar con cualquier IGP. Esa es exactamente la razón por la cual se inventó el protocolo de distribución de etiquetas (LDP): lleva los enlaces de etiquetas para las clases de equivalencia de reenvío (FEC) en la red MPLS. Una excepción es *border gateway protocol*

(BGP). Debido a que BGP lleva rutas exteriores, se considera más eficiente si también lleva las etiquetas junto a los prefijos. Debido a que BGP ya es multiprotocolo de todos modos, se puede hacer para llevar la información de la etiqueta con poco esfuerzo. Una segunda razón para elegir BGP para llevar la información de etiqueta es el hecho de que BGP es el único protocolo que distribuye prefijos entre sistemas autónomos; como tal, es un protocolo confiable para funcionar entre diferentes compañías.

Figura 65. **Red ejemplificada a nivel mundial MPLS**



Fuente: elaboración propia, empleando Visio 2013.

3.5.46. Descripción del LDP

Para obtener paquetes a través de una ruta conmutada de etiquetas (LSP) a través de la red MPLS, todos los LSR deben ejecutar un protocolo de distribución de etiquetas e intercambiar enlaces de etiquetas. Cuando todos los LSR tienen las etiquetas para una clase de equivalencia de reenvío (FEC)

particular, los paquetes se pueden reenviar en el LSP mediante la conmutación de etiqueta de los paquetes en cada LSR. La operación de etiquetado (intercambio, inserción, apertura) es conocida por cada LSR mirando en el LFIB. El LFIB, que es la tabla que reenvía los paquetes etiquetados, se alimenta de los enlaces de etiquetas recibidos por LDP, protocolo de reserva de recursos (*resource reservation protocol*, RSVP), MP-BGP o enlaces de etiquetas asignados estáticamente. Debido a que RSVP distribuye las etiquetas solo para las rutas BGP, esto etiqueta con LDP para distribuir todas las etiquetas para las rutas interiores. Por lo tanto, todos los LSR conectados directamente deben establecer una relación de pares LDP o una sesión LSP entre ellos. Los pares LDP intercambian los mensajes de asignación de etiquetas en esta sesión LDP. Una asignación o enlace de etiqueta es una etiqueta que está vinculada a un FEC. El FEC es un conjunto de paquetes que se asignan a un determinado LSP y se reenvían a través de ese LSP a través de la red MPLS. LDP tiene cuatro funciones principales:

- El descubrimiento de LSR que están ejecutando LDP
- Establecimiento y mantenimiento de la sesión
- Publicidad de mapeos de etiquetas
- Limpieza por medio de notificación

Cuando dos LSR ejecutan LDP y comparten uno o más enlaces entre ellos, deben descubrirse entre sí mediante mensajes de saludo. El segundo paso es que establezcan una sesión a través de una conexión TCP. A través de esta conexión TCP, LDP anuncia los mensajes de asignación de etiquetas entre los dos pares LDP. Estos mensajes de asignación de etiquetas se utilizan para publicitar, cambiar o retraer enlaces de etiquetas. LDP proporciona los medios para notificar al vecino LDP de algunos avisos y mensajes de error enviando mensajes de notificación.

3.5.47. El descubrimiento de LSR que están ejecutando LDP

Los LSR que ejecutan LDP envían mensajes de saludo LDP en todos los enlaces habilitados para LDP. Estas son todas las interfaces con mpls ip configuradas en ellas. Primero, sin embargo, debe habilitar CEF con el comando global ip cef. Luego de habilitar LDP globalmente con el comando *mpls ip*. En la figura 66 se muestran los comandos básicos globales y de interfaz para habilitar LDP.

Figura 66. **Configuración básica de MPLS LDP**

```
!

Hostname paris

!
Ip cef

!

Mpls ldp router-id loopback0 force
Mpls label protocol ldp
!

Interface Loopback0
  ip address 10.200.254.2 255.255.255.255

!

Interface Ethernet0/1/3
  ip address 10.200.210.2 255.255.255.0
Mpls ip

!
```

Fuente: elaboración propia, empleando Visio 2013.

Los mensajes LDP Hello son mensajes UDP que se envían en los enlaces a la dirección IP de multidifusión 'todos los routers en esta subred'; en otras

palabras, a la dirección de multidifusión IP de grupo 224.0.0.2. El puerto UDP utilizado para LDP es 646. El LSR que está recibiendo está recibiendo este mensaje de saludo LDP en una determinada interfaz es consciente de la presencia de está router LDP en esa interfaz. El mensaje *Hello* contiene un *hold time*. Si no se recibe ningún mensaje de Hello de ese LSR antes de que expire el tiempo de espera, el LSR elimina ese LSR de la lista de vecinos LDP descubiertos. Para descubrir si el LSR envía y recibe LDP *hello*, el intervalo de saludo y el tiempo de espera, use el comando *show mpls ldp discovery* [detail]. Si los mensajes LDP *Hello* se envían y reciben en una interfaz, existe una adyacencia LDP en el enlace entre dos LSR que ejecutan LDP. La figura 67 se muestra el descubrimiento de LDP en los enlaces.

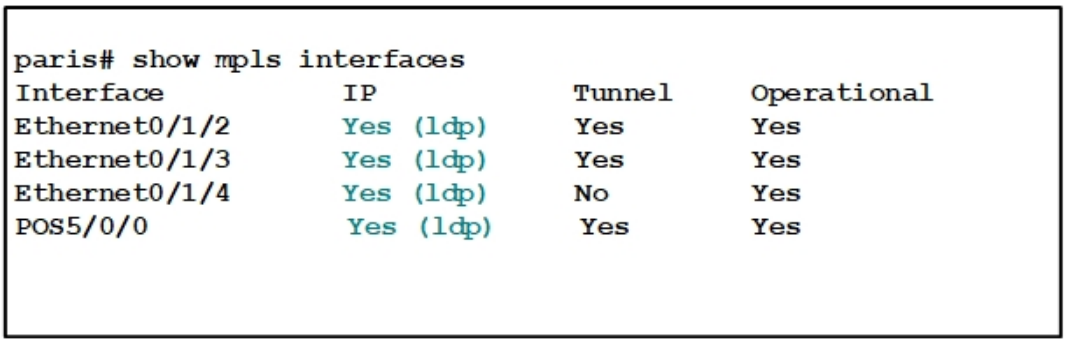
Figura 67. **Comando LDP Discovery**

```
paris# show mpls ldp discovery detail
  Local LDP Identifier:
    10.200.254.2:0
    Discovery Sources:
    Interfaces:
        Ethernet0/1/2 (ldp): xmit/recv
            Enabled: Interface config
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
            LDP Id: 10.200.254.5:0
                Src IP addr: 10.200.215.2; Transport IP addr: 10.200.254.5
                Hold time: 15 sec; Proposed local/peer: 15/15 sec
                Reachable via 10.200.254.5/32
        Ethernet0/1/3 (ldp): xmit/recv
            Enabled: Interface config
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
            LDP Id: 10.200.254.1:0
                Src IP addr: 10.200.210.1; Transport IP addr: 10.200.254.1
                Hold time: 15 sec; Proposed local/peer: 15/15 sec
                Reachable via 10.200.254.1/32
        Ethernet0/1/4 (ldp): xmit/recv
            Enabled: Interface config
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
            LDP Id: 10.200.254.1:0
                Src IP addr: 10.200.218.1; Transport IP addr: 10.200.254.1
                Hold time: 15 sec; Proposed local/peer: 15/15 sec
                Reachable via 10.200.254.1/32
        POS5/0/0 (ldp): xmit/recv
            Enabled: Interface config
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
            LDP Id: 10.200.254.3:0
                Src IP addr: 10.200.211.2; Transport IP addr: 10.200.254.3
                Hold time: 15 sec; Proposed local/peer: 15/15 sec
                Reachable via 10.200.254.3/32
```

Fuente: elaboración propia, empleando Visio 2013.

El comando *show mpls interfaces* le permite ver rápidamente qué interfaces ejecutan LDP. La figura 68 muestra el resultado del comando *show mpls interfaces*.

Figura 68. **Comando show MPLS interfaces**



Fuente: elaboración propia, empleando Visio 2013.

Para cambiar el intervalo entre el envío de mensajes de saludo o para cambiar el tiempo de espera de LDP, puede usar el comando *mpls ldp discovery {hello {holdtime | interval} seconds.*

El valor predeterminado para la palabra clave holdtime es de 15 segundos para los mensajes *Hello* de alcance, y el valor predeterminado para la palabra clave interval es de 5 segundos. La figura 69 tiene tres vecinos LDP descubiertos:10.200.254.1, 10.200.254.3 y 10.200.254.5. Como puede ver, el LSR 10.200.254.1 se descubre en dos interfaces: Ethernet 0/1/3 y Ethernet 0/1/4. El intervalo de saludo y el tiempo de espera se establecen en los valores predeterminados de 5 y 15 segundos. Si los dos pares LDP tienen diferentes tiempos de espera LDP configurados, el menor de los dos valores se usa como el tiempo de espera para esa fuente de descubrimiento LDP. Cisco IOS puede sobreescribir el intervalo de saludo LDP configurado. Elegirá un intervalo de

saludo LDP menor que el configurado para que pueda enviar al menos tres *Hellos* LDP antes de que expire el tiempo de espera. (Se envían al menos nueve hellos en el caso de una sesión LDP específica).

Si el tiempo de retención expira para un enlace, dicho enlace se elimina del descubrimiento LDP lista de fuentes si se elimina el último enlace de las fuentes de descubrimiento LDP para un vecino LDP, la sesión LDP se derriba. Si cambia el intervalo de saludo y el tiempo de espera para las fuentes de descubrimiento LDP, asegúrese de no configurar el tiempo de espera demasiado pequeño o demasiado grande. Si el tiempo de espera es demasiado pequeño, la sesión se puede perder inmediatamente incluso cuando solo se pierden unos pocos paquetes, por ejemplo, debido a la congestión en el enlace. Si el tiempo de espera se establece demasiado grande, la sesión LDP puede durar demasiado tiempo en el caso de un problema grave, y la reacción puede ser demasiado lenta. Como resultado, se pierden demasiados paquetes etiquetados.

Tenga en cuenta que los LSR que ejecutan LDP tiene un identificador LDP o ID LDP. Está ID de LDP es un campo de 6 *bytes* que consta de 4 *bytes* que identifican el LSR de forma exclusiva y 2 *bytes* que identifican el espacio de etiqueta que utiliza el LSR. Si los dos últimos bytes son 0, el espacio de etiqueta es el espacio de etiqueta de toda la plataforma o de la plataforma. Si no son cero, se utiliza un espacio de etiqueta por interfaz. Si es ese el caso, se usan múltiples ID de LDP, donde los primeros 4 *bytes* tienen el mismo valor, pero los últimos dos bytes indican un espacio de etiqueta diferente.

El espacio de etiqueta por interfaz se usa para enlaces LC-ATM. Los primeros 4 *bytes* de la ID LDP son una dirección IP tomada de una interfaz operativa en el router. Si existen interfaces de *loopback*, se toma la dirección IP

más alta de las interfaces de *loopback* para la ID del LDP o ID del *router* LDP. Si no existen interfaces de *loopback*, se toma la dirección IP más alta de una interfaz. En el ejemplo, la ID de LDP local o la ID del router LDP del router es 10.200.254.2:0, donde 10.200.254.2 es la dirección IP más alta de cualquier interfaz de *loopback* y: 0 hace referencia al espacio de etiqueta de toda la plataforma. Puede cambiar la ID del router LDP manualmente utilizando el comando *mpls ldp router-id interface* [*force*]. Si usa la palabra clave force, la ID del *router* LDP se cambia solo la próxima vez que sea necesario seleccionar la ID del *router* después de configurar está comando. Esto sucede cuando la interfaz que determina la ID del *router* LDP actual se apaga.

En Cisco IOS, la ID del router MPLS LDP debe estar presente en la tabla de enrutamiento de los routers vecinos LDP. Si no es así, la sesión LDP no se forma. Por lo tanto, la dirección IP que es la ID del router LDP en el router debe incluirse en el protocolo de enrutamiento del LSR. Si para esa dirección IP no hay ruta en la tabla de enrutamiento, la sesión LDP no está establecida. La figura 69 muestra la ruta a la dirección IP 10.200.254.3 no se encuentra en la tabla de enrutamiento del router de París. El resultado es que LSR París no forma una vecindad/sesión de LDP con LSR Roma, que tiene 10.200.254.3 como ID del router LDP.

Figura 69. **Problema 'no route'**

```
paris# show mpls ldp discovery
 Local LDP Identifier:
    10.200.254.2:0
    Discovery Sources:
    Interfaces:
       Ethernet0/1/2 (ldp): xmit/recv
           LDP Id: 10.200.254.5:0
       Ethernet0/1/3 (ldp): xmit/recv
           LDP Id: 10.200.254.1:0
       Ethernet0/1/4 (ldp): xmit/recv
           LDP Id: 10.200.254.1:0
       POS5/0/0 (ldp): xmit/recv
           LDP Id: 10.200.254.3:0; no route


paris# show mpls ldp discovery detail
 Local LDP Identifier:
    10.200.254.2:0
    Discovery Sources:
    Interfaces:
…
       POS5/0/0 (ldp): xmit/recv
           Enabled: Interface config
           Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
           LDP Id: 10.200.254.3.0; no route to transport addr
           Src IP addr: 10.200.211.2; Transport IP addr: 10.200.254.3
           Hold time: 15 sec; Proposed local/peer: 15/15 sec


paris# show ip route 10.200.254.3 255.255.255.255
% Subnet not in table
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.48. Establecimiento y mantenimiento de sesión LDP

Si dos LSR se han cubierto por medio de los Hellos de LDP, intentan establecer una sesión de LDP entre ellos. Un LSR intenta abrir una conexión TCP al puerto TCP 646 a la otra LSR. Si la conexión TCP está configurada, ambos LSR negocian los parámetros de sesión LDP intercambiando mensajes de inicialización LDP. Estos parámetros incluyen cosas como las siguientes.

- Valores del temporizador.

- Método de distribución de etiquetas.

- Rangos del identificador de ruta virtual (*virtual path identifier*, VPI) / identificador de canal virtual (*virtual channel identifier*, VCI) para ATM controlado por la etiqueta (LC-ATM).

- Rangos de identificador de conexión de enlace de datos (*data-link connection identifier* DLCI) para *LC-Frame Relay*.

Si los pares LDP acuerdan los parámetros de la sesión, mantienen la conexión TCP entre ellos. De lo contrario, intentan crear la sesión LDP entre ellos, pero a un ritmo acelerado. En Cisco IOS, el comando de finalización LDP controla está velocidad de aceleración:

mpls ldp backoff initial-backoff maximum-backoff

El parámetro inicial de retroceso es un valor entre 5 y 2. 147.483, con un valor predeterminado de 15 segundos. El máximo de retroceso es un valor entre 5 y 2, 147, 483, con un valor predeterminado de 120 segundos. Este comando ralentiza los intentos de configuración de la sesión LDP de dos LSR LDP, cuando los dos pares LDP vecinos son incompatibles en términos de los parámetros que intercambian. Si el intento de configuración de la sesión falla, los siguientes intentos se llevan a cabo en un tiempo exponencialmente incrementado, hasta que se alcanza el tiempo máximo de reducción. Un ejemplo en el que los dos pares LDP pueden estar en desacuerdo sobre los parámetros y no formar una sesión LDP es el caso de LC-ATM, donde los dos pares están usando diferentes rangos de valores VPI / VCI para las etiquetas.

Después de que se haya configurado la sesión LDP, se mantiene mediante la recepción de paquetes LDP o un mensaje *keepalive* periódico. Cada vez que el par LDP recibe un paquete LDP o un mensaje *keepalive*, el temporizador *keepalive* se restablece para ese par. El temporizador *keepalive* o el tiempo de espera para la sesión LDP se pueden configurar también. El comando para cambiar el temporizador *keepalive* de la sesión LDP es *mpls ldp holdtime seconds*. Puede configurar el valor del tiempo de espera para que este entre 15 y 2, 147, 483 segundos, con un valor predeterminado de 180 segundos.

La figura 70 muestra un par LDP con la ID del *router* LDP 10.200.254.2. El puerto TCP local utilizado es 646, y el puerto TCP remoto utilizado es 11537. El tiempo de espera de la sesión es de 180 segundos, y los mensajes *keepalive* (KA) se envían con un intervalo de 60 segundos.

Figura 70. **Tiempo de espera del vecino LDP e intervalo KA**

```
paris#show mpls ldp neighbor 10.200.254.5 detail
    Peer LDP Ident: 10.200.254.5:0; Local LDP Ident 10.200.254.2:0
        TCP connection: 10.200.254.5.11537 - 10.200.254.2.646
        State: Oper; Msgs sent/rcvd: 16/19; Downstream; Last TIB rev sent 50
        Up time: 00:00:36; UID: 9; Peer Id 1;
        LDP discovery sources:
         Ethernet0/1/2; Src IP addr: 10.200.215.2
           holdtime: 15000 ms, hello interval: 5000 ms
        Addresses bound to peer LDP Ident:
          10.200.254.5 10.200.215.2 10.200.216.1
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
```

Fuente: elaboración propia, empleando Visio 2013.

También puede ver los temporizadores de descubrimiento y sesión con el comando *show mpls ldp parameters*, como se ilustra en la figura 71.

Figura 71. **Comando show MPLS LDP parameters**

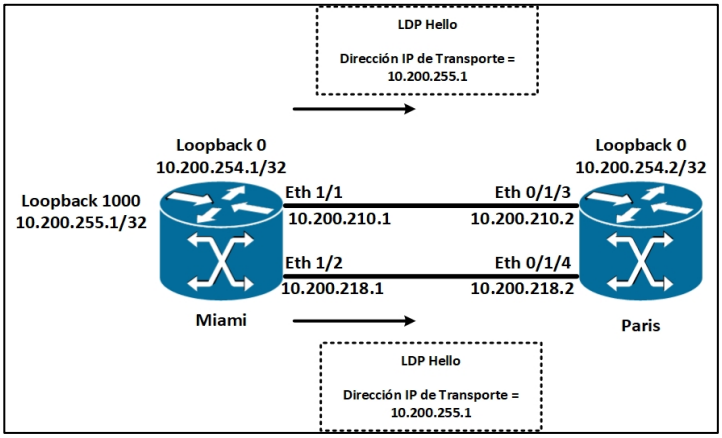
```
paris#show mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 100000
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

Fuente: elaboración propia, empleando Visio 2013.

La sesión LDP es una conexión TCP que se establece entre dos direcciones IP de los LSR. Por lo general, estas direcciones IP se utilizan para crear el identificador del *router* LDP en cada router. Sin embargo, si no desea utilizar está dirección IP para crear la sesión LDP, puede cambiarla. Para cambiar la dirección IP, configure el comando *mpls ldp discovery transport-address {interface* | *ipaddress*} en la interfaz del *router* y especificar una interfaz o dirección IP que se utilizará para crear la sesión LDP. Está dirección IP de transporte se anuncia en los hellos LDP que se envían en las interfaces habilitadas para LDP. Cuando un *router* tiene múltiples enlaces hacia otro *router* LDP, la misma dirección de transporte debe publicarse en todos los enlaces paralelos que usan el mismo espacio de etiqueta.

La figura 72 muestra dos *routers* están conectados a través de dos enlaces Ethernet. En el *router* Miami, la dirección de transporte se cambia a la dirección ip 1000 de *loopback*. Se observa en el ejemplo que la dirección utilizada para la conexión TCP ha cambiado de la dirección IP encontrada en la ID del *router* LDP a la dirección IP 10.200.255.1 de *loopback* 1000.

Figura 72. **Cambio de la dirección de transporte LDP predeterminada**



Fuente: elaboración propia, empleando Visio 2013.

Figura 73. **Cambio de la dirección de transporte LDP predeterminada a nivel de consola**

```
!
hostname miami
!
interface Ethernet1/1
  ip address 10.200.210.1 255.255.255.0
  mpls ldp discovery transport-address 10.200.255.1
  mpls ip
!
interface Ethernet1/2
  ip address 10.200.218.1 255.255.255.0
  mpls ldp discovery transport-address 10.200.255.1
  mpls ip
!
london#show mpls ldp discovery detail
  Local LDP Identifier:
     10.200.254.2:0
     Discovery Sources:
     Interfaces:
         Ethernet0/1/3 (ldp): xmit/recv
             Enabled: Interface config
             Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
             LDP Id: 10.200.254.1:0
               Src IP addr: 10.200.210.1; Transport IP addr: 10.200.255.1
               Hold time: 15 sec; Proposed local/peer: 15/15 sec
               Reachable via 10.200.255.1/32
         Ethernet0/1/4 (ldp): xmit/recv
             Enabled: Interface config
             Hello interval: 5000 ms; Transport IP addr: 10.200.254.2
             LDP Id: 10.200.254.1:0
               Src IP addr: 10.200.218.1; Transport IP addr: 10.200.255.1
               Hold time: 15 sec; Proposed local/peer: 15/15 sec
               Reachable via 10.200.255.1/32
```

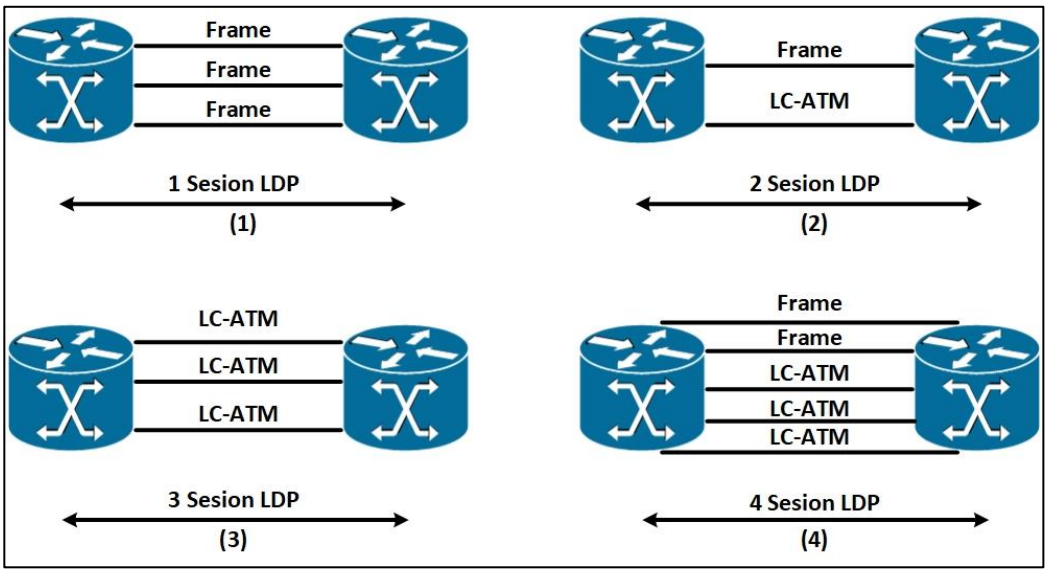
Fuente: elaboración propia, empleando Visio 2013.

Cuando un *router* tiene múltiples enlaces hacia otro router LDP y se anuncia una dirección de transporte diferente en esos enlaces, la sesión TCP aún está formada, pero falta un enlace de las 'fuentes de descubrimiento' del LDP en el otro *router*. En el ejemplo anterior, se forma la sesión LDP, pero Ethernet 0/1/3 o Ethernet 0/1/4 falta de las fuentes de descubrimiento LDP en la salida del router de París. Como tal, el tráfico desde el *router* Paris hacia el *router* Miami no tiene equilibrio de carga, pero usa solo un enlace de Ethernet saliente.

3.5.49. Número de sesiones LDP

Puede pensar que una sesión de LDP entre un par de LSR es suficiente para hacer el trabajo. Cuando el espacio de etiqueta por plataforma es el único espacio de etiqueta utilizado entre un par de LSR, una sesión de LDP es suficiente. esto es así porque solo se intercambia el conjunto de enlaces de etiquetas entre los dos LSR, sin importar cuantos enlaces haya entre ellos. Básicamente, las interfaces pueden compartir el mismo conjunto de etiquetas cuando se utiliza el espacio de etiquetas por plataforma. La razón de esto es que todos los enlaces de etiquetas son relevantes para todos los enlaces entre los dos LSR, ya que todos pertenecen al mismo espacio de etiqueta. Las interfaces pertenecen al espacio de etiqueta por plataforma cuando son interfaces de modo de marco. Las interfaces que no son interfaces en modo marco, como las interfaces LC-ATM, tienen un espacio de etiqueta por interfaz. Con espacio de etiqueta por interfaz, cada enlace de etiqueta tiene relevancia solo para esa interfaz, debe existir una sesión LDP entre el par de routers. En la figura 75 se observa algunos ejemplos del número de sesiones LDP entre un par de LSR.

Figura 74. **Ejemplos del número de sesiones LDP entre un par de LSR**



Fuente: elaboración propia, empleando Visio 2013.

Para todos los enlaces de modo de marco, solo una sesión de LDP debería intercambiar las etiquetas en el espacio de etiqueta por plataforma. para cada enlace LC-ATM, una sesión LDP debería intercambiar las etiquetas en el espacio de etiqueta por interfaz. En (1) la figura 74 se ven tres enlaces de cuadro, por lo que solo se requiere una sesión LDP entre los dos LSR; en (2) verá un enlace de cuadro y un enlace LC-ATM. Debido a que cada enlace LCATM requiere su propia sesión LDP, hay dos sesiones LDP; (3) muestra tres enlaces LC-ATM, por lo tanto, el número de sesiones LDP es tres; (4) muestra dos enlaces de cuadro y tres enlaces LCATM. Los dos enlaces de cuadro tienen una sesión LDP y los enlaces LC-ATM tienen tres sesiones LDP.

3.5.50. Publicidad de asignaciones de etiquetas

Las asignaciones de etiquetas publicitarias o enlaces de etiquetas es el objetivo principal de LDP. Cada uno de los tres modos tiene dos posibilidades, que conduce a los siguientes seis modos:

- Modo de publicidad no solicitada *Downstream* (UD) versus *Downstream-on-Demand* (DoD).

- Retención de etiqueta liberal (LLR) versus modo de conversación de etiqueta conservadora (CLR).

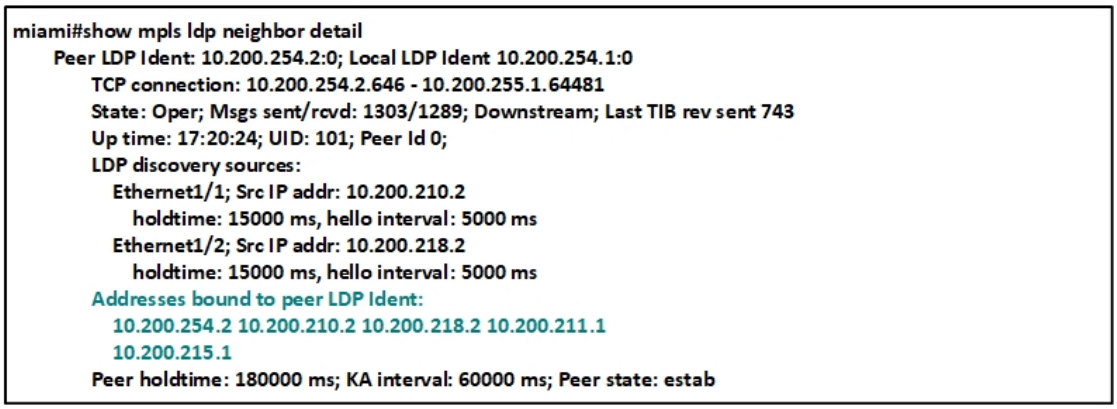
- Control LSP independiente versus modo de control LSP ordenado.

Independientemente del modo en que operen los pares LDP, el objetivo es anunciar enlaces de etiquetas. En el modo de publicidad UD, el par LDP distribuye los enlaces de etiquetas no solicitados a sus pares LDP. Sin embargo, los enlaces de etiqueta son un conjunto (identificador LDP, etiqueta) por prefijo. Un *router* LDP recibe múltiples enlaces de etiquetas para cada prefijo, es decir, uno por par LDP. Todos estos enlaces de etiquetas se almacenan en la LIB del *router*. Sin embargo, solo un par LDP es el LSR indirecto para ese prefijo en particular. Por supuesto, si existe equilibrio de carga, es posible tener más de un LSR en sentido descendente.

El LSR indirecto se encuentra buscando el siguiente salto para ese prefijo en la tabla de enrutamiento. Solo el enlace remoto asociado con ese LSR del próximo salto debe usarse para poblar el LFIB. Esto significa que solo una etiqueta de todos los enlaces de etiquetas publicitados de todos los vecinos LDP de este LSR debe usarse como etiqueta de salida en el LFIB para ese

prefijo. El problema es que los enlaces de etiquetas se anuncian como (identificador LDP, etiqueta) sin las direcciones IP de las interfaces. Esto significa que, para encontrar la etiqueta de salida para un prefijo en particular, debe asignar al identificador LDP la dirección IP de la interfaz, apuntando de nuevo a está LSR en el LSR indirecto. Solo puede hacer esto si cada par LDP anuncia todas sus direcciones IP. Estas direcciones IP son anunciadas por el par LDP con mensajes de dirección y se retiran con mensajes de retiro de direcciones. Puede encontrar estas direcciones cuando mira al par LDP. Se llaman las direcciones atadas para el par LDP. La figura 75 muestra las direcciones enlazadas a pares 10.200.254.2 (Paris) en LSR Miami.

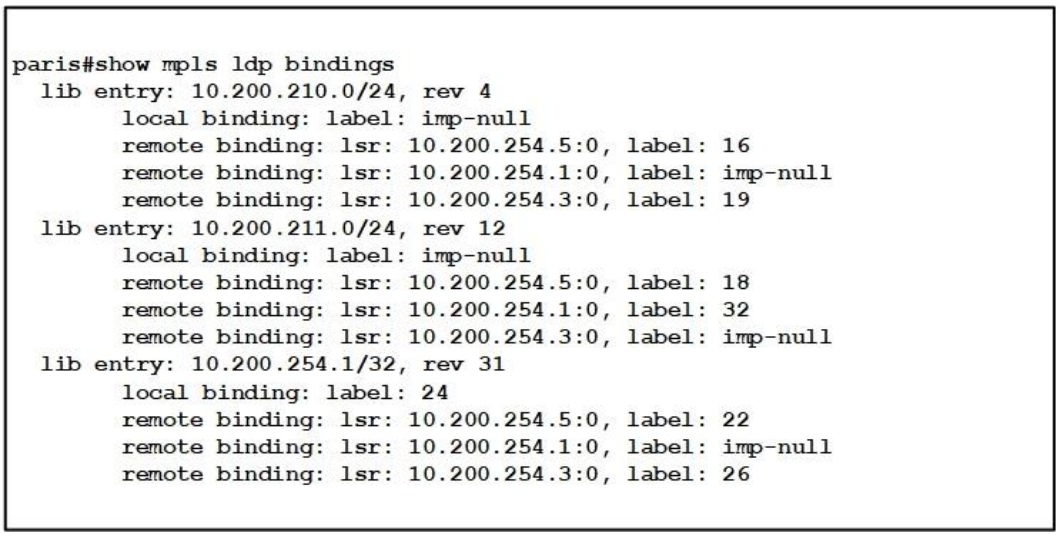
Figura 75. **Direcciones IP enlazadas LDP**



Fuente: elaboración propia, empleando Visio 2013.

Cada LSR asigna una etiqueta local a cada prefijo IGP en la tabla de enrutamiento. Está es el enlace de etiqueta local. Estos enlaces locales se almacenan en el LIB en el router. Cada una de estas etiquetas y los prefijos a los que están asignados se anuncian a través de LDP a todos los pares LDP. Estos enlaces de etiquetas son los enlaces remotos en los pares LDP y se almacenan en el LIB. En el ejemplo se muestra el LIB en un LSR.

Figura 76. **Ejemplo de un LIB, 1**



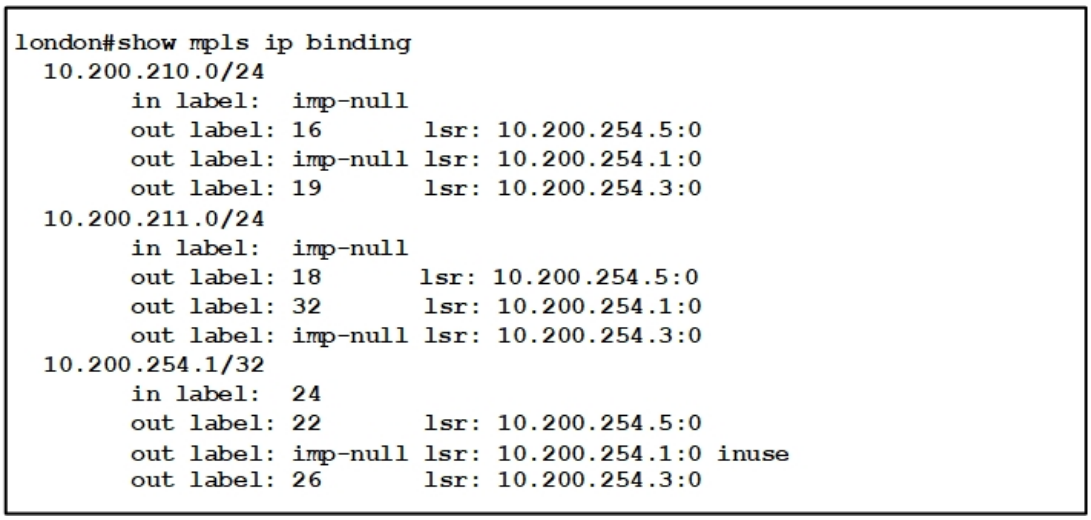
```
paris#show mpls ldp bindings
  lib entry: 10.200.210.0/24, rev 4
        local binding: label: imp-null
        remote binding: lsr: 10.200.254.5:0, label: 16
        remote binding: lsr: 10.200.254.1:0, label: imp-null
        remote binding: lsr: 10.200.254.3:0, label: 19
  lib entry: 10.200.211.0/24, rev 12
        local binding: label: imp-null
        remote binding: lsr: 10.200.254.5:0, label: 18
        remote binding: lsr: 10.200.254.1:0, label: 32
        remote binding: lsr: 10.200.254.3:0, label: imp-null
  lib entry: 10.200.254.1/32, rev 31
        local binding: label: 24
        remote binding: lsr: 10.200.254.5:0, label: 22
        remote binding: lsr: 10.200.254.1:0, label: imp-null
        remote binding: lsr: 10.200.254.3:0, label: 26
```

Fuente: elaboración propia, empleando Visio 2013.

Como se observa, para cada prefijo, el LSR siempre tiene un enlace local y un enlace remoto por par LDP.

En el ejemplo se muestra otro comando para echar un vistazo a la LIB en el LSR. La entrada 'en etiqueta' se refiere a la vinculación local. Las entradas de 'etiqueta de salida' se refieren a los enlaces remotos. Cada vez, puede ver la etiqueta y el identificador LDP del LSR que envió el enlace remoto.

Figura 77. **Ejemplo de un LIB, 2**



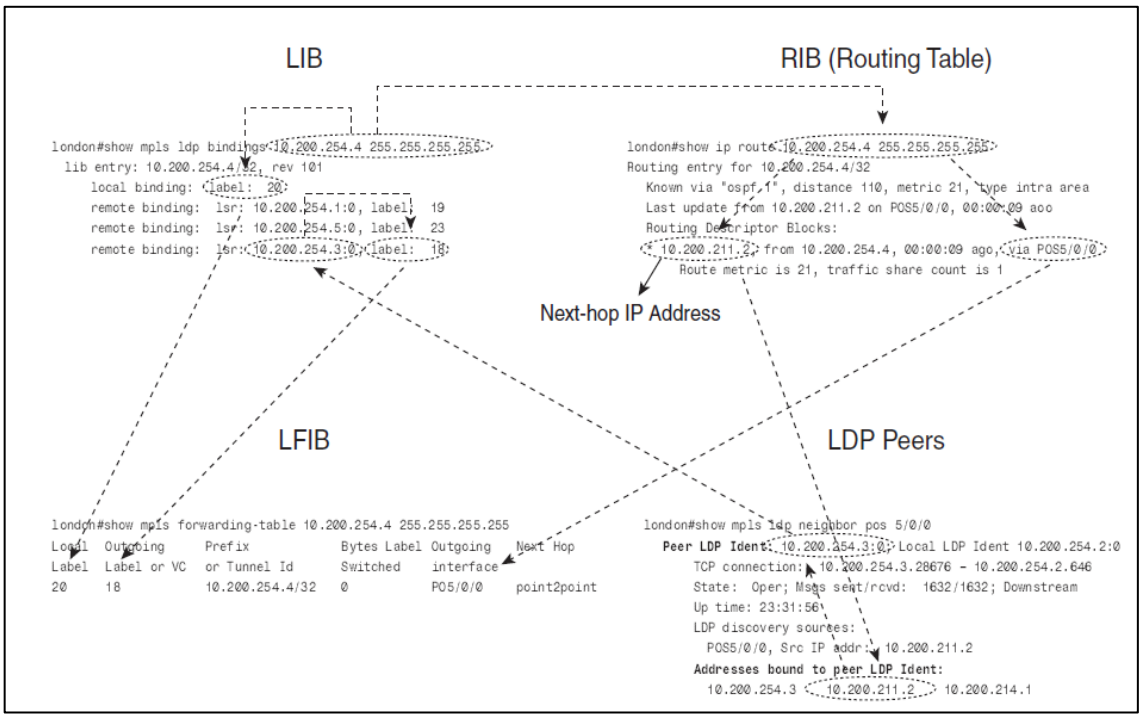
Fuente: elaboración propia, empleando Visio 2013.

La ventaja del comando *show mpls ip binding* es que también muestra que etiqueta de todos los posibles enlaces remotos se usa para reenviar el tráfico al indicar *inuse*. Inuse indica la etiqueta saliente en el LFIB para ese prefijo.

Se observa en la figura la asociación entre el RIB, las direcciones enlazadas de los pares del LDP, el LIB y el LFIB.

En la figura 78 se muestra el ejemplo de construir la entrada LFIB para un FEC vinculado al prefijo 10.200.254.4/32. La etiqueta entrante / local para el prefijo se encuentra directamente en el LIB, pero la etiqueta saliente se encuentra a través del RIB, las direcciones enlazadas de los pares LDP y el LIB.

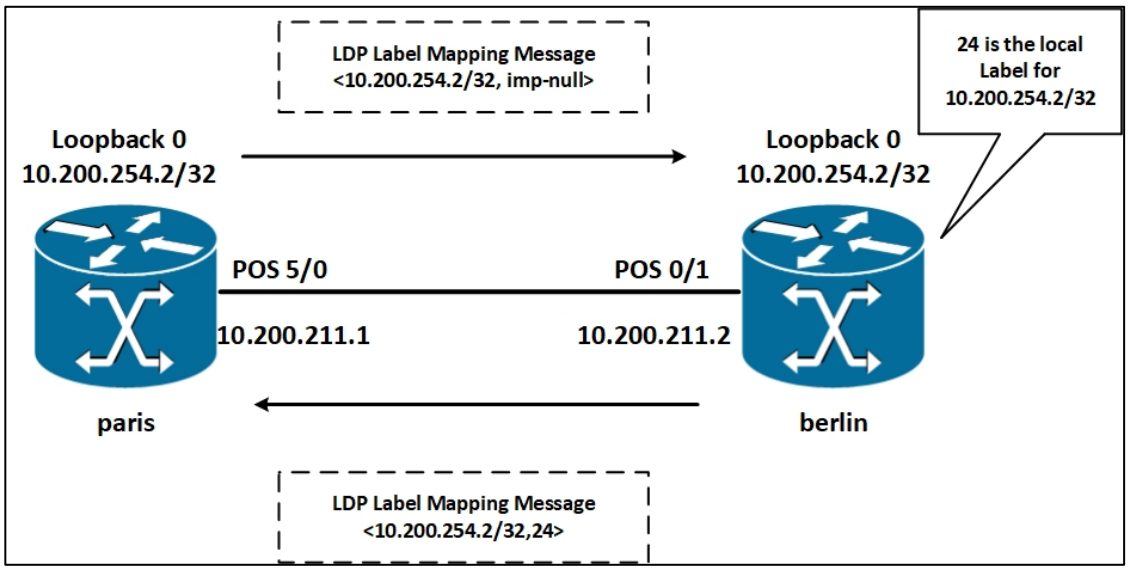
Figura 78. **Relación entre direcciones enlazadas, RIB,LIB y LFIB**



Fuente: elaboración propia, empleando Visio 2013.

Tener en cuenta que LDP asigna etiquetas locales a todos los prefijos IGP y anuncia los enlaces a todos los pares LDP. El concepto de horizonte dividido no existe; un par LDP asigna su propia etiqueta local a un prefijo y lo anuncia al otro par LDP, aunque ese otro par LDP posee el prefijo (es un prefijo conectado) o ese otro par LDP es el LSR indirecto. Obsérvese la figura, que muestra una red simple con dos LSR. El *router* Paris posee el prefijo 10.200.254.2/32 porque es el prefijo en la interfaz de *loopback* 0. Este *router* anuncia su enlace para el prefijo de Roma. La etiqueta anunciada es etiqueta implícita NULL. A su vez, el *router* París recibe el enlace remoto para el prefijo 10.200.254.2/32 del *router* Roma, aunque el *router* París posee el prefijo.

Figura 79. **Utilizando No LDP Split Horizon**



Fuente: elaboración propia, empleando Visio 2013.

Figura 80. **Fijado por 10.200.254.2/32 de la figura 79**

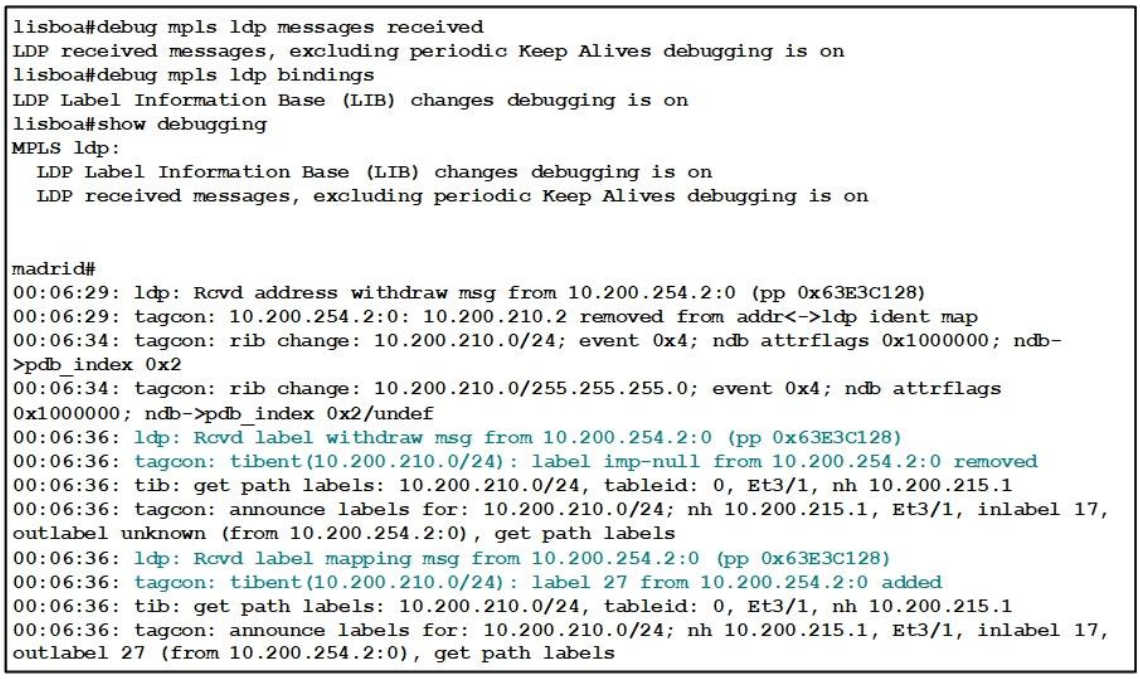
```
paris#show interfaces loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 10.200.254.2/32


paris#show mpls ldp bindings 10.200.254.2 255.255.255.255
  lib entry: 10.200.254.2/32, rev 8
        local binding: label: imp-null
        remote binding: lsr: 10.200.254.5:0, label: 21
        remote binding: lsr: 10.200.254.1:0, label: 25
        remote binding: lsr: 10.200.254.3:0, label: 24


berlin#show mpls ldp bindings 10.200.254.2 255.255.255.255
  lib entry: 10.200.254.2/32, rev 787
        local binding: label: 24
        Remote binding: lsr: 10.200.254.4:0, label: 23
        remote binding: lsr: 10.200.254.2:0, label: imp-null
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.51. Retiro de etiqueta

Cuando un par LDP anuncia un enlace de etiqueta, los pares LDP receptores lo conservan hasta que la sesión LDP se apaga o hasta que se retira la etiqueta. La etiqueta puede ser retirada si la etiqueta local cambia. La etiqueta local puede cambiar si, por ejemplo, la interfaz con un cierto prefijo se cae, pero otra LSR todavía anuncia el prefijo. Por lo tanto, la etiqueta local para ese prefijo cambia de implícita NULL a una etiqueta no reservada. Si esto sucede, la etiqueta NULL implícita se retira inmediatamente mediante el envío de un mensaje de retirada de etiqueta a los pares LDP. La nueva etiqueta se anuncia en un mensaje de asignación de etiquetas. En la figura 81 se muestra la interfaz Ethernet con el prefijo IP 10.200.210.0/24 desciende en el LSR Paris. Es por eso que París retira el prefijo con la etiqueta implícita NULL. Sin embargo, LSR Miami aún anuncia el prefijo, suponiendo que hay un *switch* de capa 2 entre Miami y París, de modo que el lado de Miami del enlace Ethernet permanezca activo. LSR París asigna una nueva etiqueta local (27) al prefijo y anuncia esa nueva etiqueta en un mensaje de asignación de etiquetas al LSR de Lisboa.

Figura 81. **Etiqueta retirada**

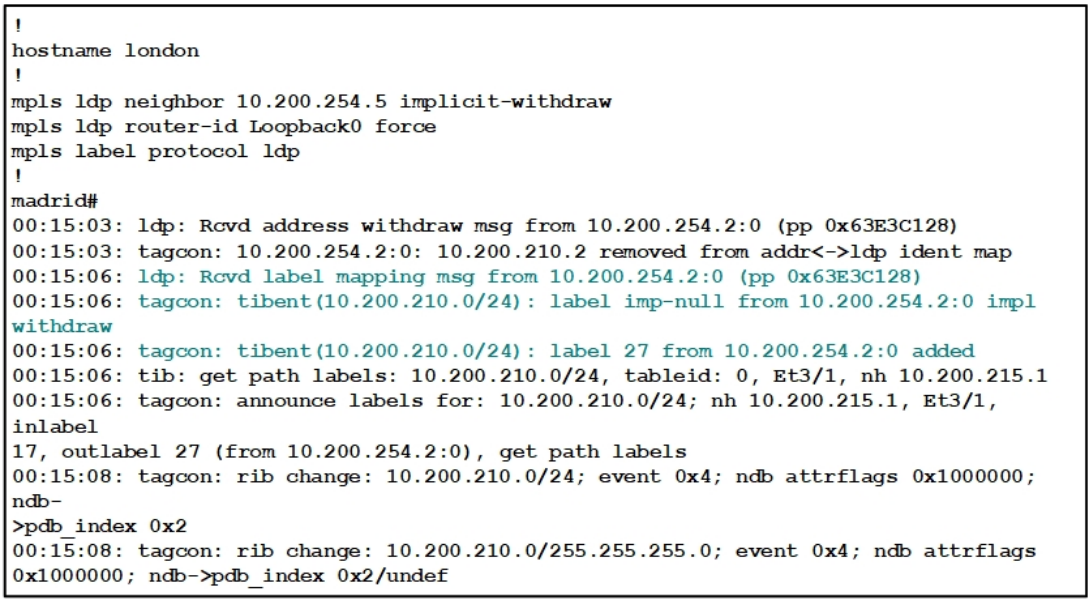


```
lisboa#debug mpls ldp messages received
LDP received messages, excluding periodic Keep Alives debugging is on
lisboa#debug mpls ldp bindings
LDP Label Information Base (LIB) changes debugging is on
lisboa#show debugging
MPLS ldp:
  LDP Label Information Base (LIB) changes debugging is on
  LDP received messages, excluding periodic Keep Alives debugging is on


madrid#
00:06:29: ldp: Rcvd address withdraw msg from 10.200.254.2:0 (pp 0x63E3C128)
00:06:29: tagcon: 10.200.254.2:0: 10.200.210.2 removed from addr<->ldp ident map
00:06:34: tagcon: rib change: 10.200.210.0/24; event 0x4; ndb attrflags 0x1000000; ndb-
>pdb_index 0x2
00:06:34: tagcon: rib change: 10.200.210.0/255.255.255.0; event 0x4; ndb attrflags
0x1000000; ndb->pdb_index 0x2/undef
00:06:36: ldp: Rcvd label withdraw msg from 10.200.254.2:0 (pp 0x63E3C128)
00:06:36: tagcon: tibent(10.200.210.0/24): label imp-null from 10.200.254.2:0 removed
00:06:36: tib: get path labels: 10.200.210.0/24, tableid: 0, Et3/1, nh 10.200.215.1
00:06:36: tagcon: announce labels for: 10.200.210.0/24; nh 10.200.215.1, Et3/1, inlabel 17,
outlabel unknown (from 10.200.254.2:0), get path labels
00:06:36: ldp: Rcvd label mapping msg from 10.200.254.2:0 (pp 0x63E3C128)
00:06:36: tagcon: tibent(10.200.210.0/24): label 27 from 10.200.254.2:0 added
00:06:36: tib: get path labels: 10.200.210.0/24, tableid: 0, Et3/1, nh 10.200.215.1
00:06:36: tagcon: announce labels for: 10.200.210.0/24; nh 10.200.215.1, Et3/1, inlabel 17,
outlabel 27 (from 10.200.254.2:0), get path labels
```

Fuente: elaboración propia, empleando Visio 2013.

En el antiguo software Cisco IOS (pre 12.0 (21) ST), el comportamiento predeterminado no era enviar un mensaje de retirada de etiqueta para retirar la etiqueta antes de anunciar la nueva etiqueta para el FEC. La nueva publicidad de etiqueta también era una retirada implícita de la etiqueta. Si desea mantener el comportamiento anterior, debe configurar el comando mpls ldp neighbor neighbor implicit-withdraw. La figura 82 muestra lo que sucede cuando una nueva etiqueta se anuncia para el prefijo 10.200.210.0/24 con extracción implícita configurada en el LDP vecino París. El mensaje de retiro de etiqueta falta en la salida de depuración. La ventaja de este comando es evitar enviar mensajes *Label Withdraw*, lo que equivale a una menor sobrecarga.

Figura 82. **Etiqueta implícita de retiro**



```
!
hostname london
!
mpls ldp neighbor 10.200.254.5 implicit-withdraw
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
madrid#
00:15:03: ldp: Rcvd address withdraw msg from 10.200.254.2:0 (pp 0x63E3C128)
00:15:03: tagcon: 10.200.254.2:0: 10.200.210.2 removed from addr<->ldp ident map
00:15:06: ldp: Rcvd label mapping msg from 10.200.254.2:0 (pp 0x63E3C128)
00:15:06: tagcon: tibent(10.200.210.0/24): label imp-null from 10.200.254.2:0 impl
withdraw
00:15:06: tagcon: tibent(10.200.210.0/24): label 27 from 10.200.254.2:0 added
00:15:06: tib: get path labels: 10.200.210.0/24, tableid: 0, Et3/1, nh 10.200.215.1
00:15:06: tagcon: announce labels for: 10.200.210.0/24; nh 10.200.215.1, Et3/1,
inlabel
17, outlabel 27 (from 10.200.254.2:0), get path labels
00:15:08: tagcon: rib change: 10.200.210.0/24; event 0x4; ndb attrflags 0x1000000;
ndb-
>pdb_index 0x2
00:15:08: tagcon: rib change: 10.200.210.0/255.255.255.0; event 0x4; ndb attrflags
0x1000000; ndb->pdb_index 0x2/undef
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.52. La limpieza por medio de la notificación

Los mensajes de notificación son necesarios para el mantenimiento de las sesiones de LDP. Los mensajes de notificación señalan eventos significativos al para LDP. Estos eventos pueden ser errores fatales (notificaciones de error) o información de asesoramiento simple (notificaciones de aviso). Si se produce un error fatal, el LSR emisor y el LSR receptor deben finalizar la sesión LDP inmediatamente. Las notificaciones de asesoramiento se utilizan para enviar información sobre la sesión LDP o un mensaje recibido del par. Los siguientes eventos se pueden señalar enviando mensajes de notificación:

- Unidad de datos de protocolo (*protocol data unit*, PDU) mal formada o mensaje.

- Valor de longitud de tipo desconocido (TLV) o mal formado.
- Sesión de expiración de temporizador keepalive.
- Cierre de sesión unilateral.
- Eventos de mensajes de inicialización.
- Eventos resultantes de otros mensajes.
- Errores internos.
- Detección de bucles.
- Eventos diversos.

3.5.53. Sesión dirigida LDP

Normalmente, las sesiones LDP se configuran entre LSR conectados directamente. En una red en la que las rutas IGP deben etiquetarse, esto es suficiente, ya que la conmutación de etiquetas de los paquetes es salto por salto. Por lo tanto, si los enlaces de etiqueta se anuncian salto por salto para las rutas IGP, se configuran los LSP. Sin embargo, en los casos, se necesita una sesión LDP remota o específica. Está es una sesión LDP entre LSR que están conectados directamente. Los ejemplos en lo que se necesita la sesión LDP específica son redes AToM y túneles TE en una red MPLS VPN. En el caso de AToM, debe existir una sesión LDP entre cada par de *routers* PE. La sesión LDP remota se configura al configurar el comando xconnect en los *routers* PE de la red AToM. En el caso de túneles TE en una red MPLS VPN, con los túneles TE que terminan en un router P, el LSR se cabecera y cola del túnel TE necesita una sesión LDP específica entre ellos para obtener el tráfico MPLS VPN correctamente etiquetado conmutado a través de la red MPLS VPN. Para vecinos directamente conectados, solo necesita habilitar mpls ip en la interfaz; los pares LDP se descubren entre si y crean la sesión LDP TCP entre ellos. Para los vecinos LDP que no están conectados directamente, el vecindario LDP

debe configurarse manualmente en ambos routers con el comando mpls ldp neighbor targeted.

La sintaxis del comando es el siguiente:

mpls ldp neighbor [vrf *vpn-name*] *ip - addr* targeted [ldp | tdp]

El vrf se refiere a los escenarios de *Carrier's Carrier* (CsC) en los que las sesiones LDP se establecen en las interfaces VRF.

Un vecino LDP específico puede mejorar el tiempo de convergencia de la etiqueta en comparación con el tiempo de convergencia con los pares LDP conectados directamente cuando hay enlaces de aleteo. Esto se debe a que cuando el enlace entre dos LSR disminuye, la sesión LDP TCP de un LSR al otro, la sesión LDP permanece activa cuando el enlace entre los dos LSR falla. Si la sesión LDP permanece activa, las etiquetas se conservan, mejorando la instalación de las etiquetas desde la LIB a la LFIB cuando el enlace vuelve a subir.

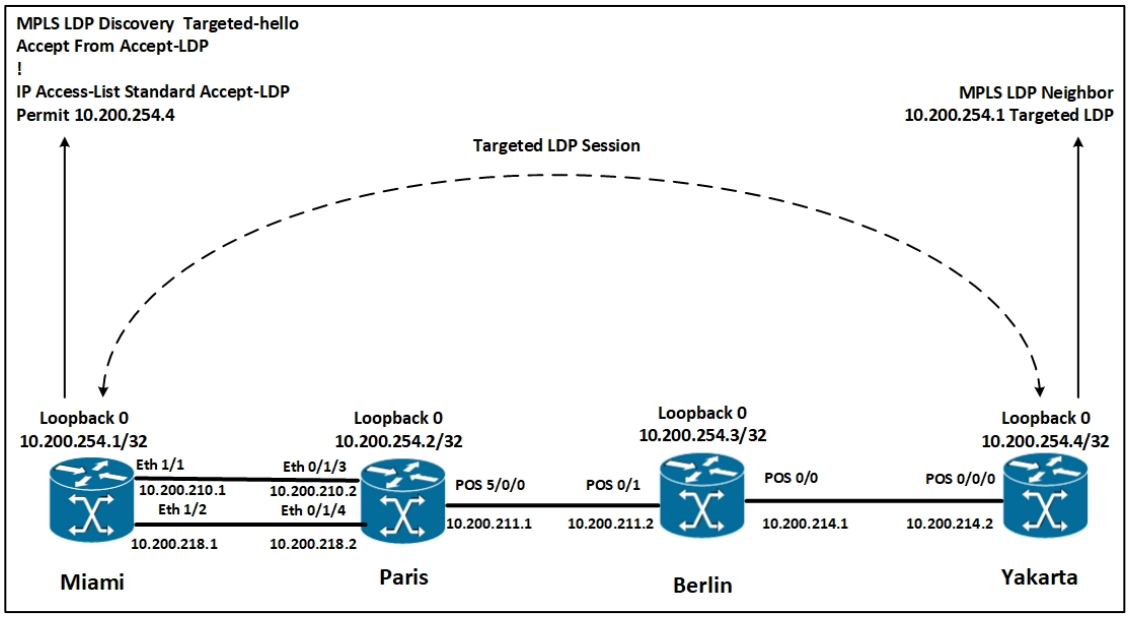
Para cambiar el intervalo de saludo LDP y el tiempo de espera para sesiones LDP específicas, puede usar el siguiente comando:

mpls ldp discovery { hello { holdtime | interval } *seconds* | targeted - hello { holdtime | interval } *seconds* | accept [from *acl*] }

Obsérvese la figura. Los *routers* Miami y Yakarta no están conectados directamente; sin embargo, desea que tengan una sesión LDP entre ellos. Puede configurar en ambos *routers* el vecino LDP como objetivo. Otra forma de lograr el mismo resultado es configurar el vecino LDP objetivo en un solo *router*

y configurar el otro *router* para aceptar sesiones LDP específicas de *routers* LDP específicos. Para ello, configure el comando *mpls ldp discovery targeted-hello accept [from acl*]. Para evitar que cualquier *router* configure una sesión LDP con está *router*, puede usar el comando con una lista de acceso para que pueda especificar qué *routers* pueden configurar una sesión LDP específica.

Figura 83. **Hello target aceptado en la red**



Fuente: elaboración propia, empleando Visio 2013.

En las figuras 84 y 85 se muestran la configuración necesaria en los *routers* Miami y Yakarta para configurar una sesión LDP específica entre ellos.

Figura 84. **Configuracion de Sydney para LDP dirigido (corregir)**

```
!
hostname yakarta
!
mpls label protocol ldp
mpls ldp neighbor 10.200.254.1 targeted ldp
mpls ldp router-id Loopback0 force
!
```

Fuente: elaboración propia, empleando Visio 2013.

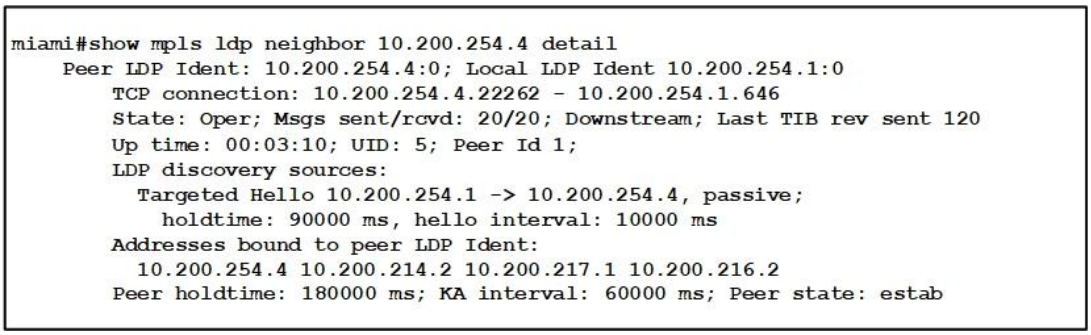
Figura 85. **Configuración de Miami para LDP dirigido**

```
miami#conf t
miami(config)#mpls ldp discovery targeted-hello accept from accept-ldp
miami(config)#ip access-list standard accept-ldp
miami(config-std-nacl)#permit host 10.200.254.4
miami(config-std-nacl)#^Z
miami#

!
mpls ldp discovery targeted-hello accept from accept-ldp
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!

!
ip access-list standard accept-ldp
permit 10.200.254.4
!
```

Fuente: elaboración propia, empleando Visio 2013.

En la figura 86 se muestra el resultado del comando *show mpls lsdp neighbor* para la sesión LDP específica.

Figura 86. **Sesión LDP dirigida en router Miami**



Fuente: elaboración propia, empleando Visio 2013.

3.5.54. Autenticación LDP

Las sesiones LDP son sesiones TCP. Las sesiones TCP pueden ser atacadas por segmentos TCP falsificados. Para proteger LDP contra tales ataques, puede usar la autenticación Message Digest 5 (MD5). MD5 agrega una firma, llamada el compendio MD5, a los segmentos TCP. El resumen de MD5 se calcula para el segmento TCP particular utilizando la contraseña configurada en ambos extremos de la conexión. La contraseña de MD5 configurada nunca se transmite. Esto dejaría a un pirata informático potencial teniendo que adivinar los números de secuencia TCP y la contraseña MD5. En Cisco IOS, puede configurar MD5 para el LDP configurado una contraseña para el par LDP con el siguiente comando:

mpls ldp neighbor [vrf *vpn - name*] *ip - addr* password [*0 - 7*] *pswd - string*

MD5 agrega un resumen a cada segmento TCP enviando. Este resumen se puede verificar sólo por los dos pares LDP que están configurados con la

contraseña correcta. Si un LSR tiene MD5 configurado para LDP y el otro no, se registra el mensaje:

%TCP -6- BADAUTH : No MD5 digest from 10.200.254.4 (11092) to 10.200.254.3 (646)

Si ambos pares LDP tienen una contraseña configurada para MD5 pero las contraseñas no coinciden, se registra el siguiente mensaje:

%TCP -6- BADAUTH : Invalid MD5 digest from 10.200.254.4 (11093) to 10.200.254.3 (646)

3.5.55. Control de anuncio de etiquetas a través de LDP

LDP le permite controlar el anuncio de las etiquetas. Puede configurar LDP para publicitar o anunciar ciertas etiquetas a ciertos pares LDP. A continuación, puede utilizar las etiquetas asignadas localmente que se anuncian a los pares LDP como etiqueta saliente en esos LSR. La sintaxis para este comando es la siguiente:

mpls ldp advertise - labels [vrf vpn - name] [interface interface] for prefix - access - list [to peer - access - list]

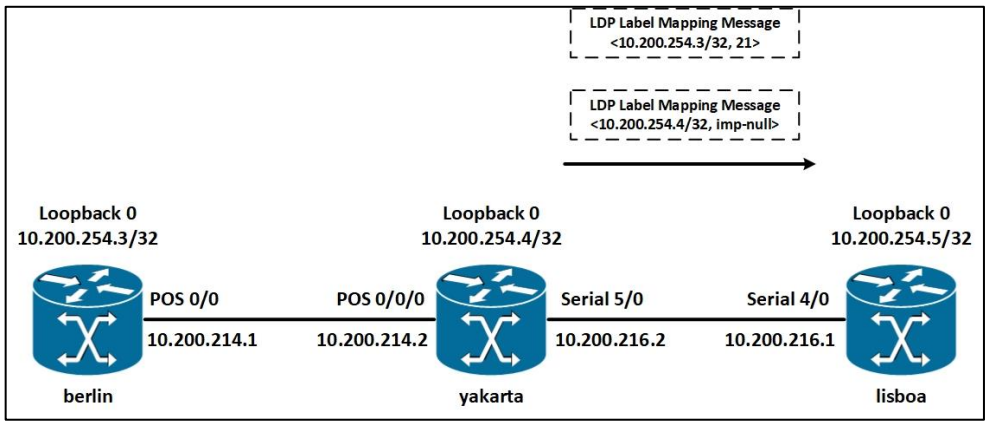
La lista de acceso a prefijo es una lista de acceso numerado estándar (1-99) o una lista de acceso con nombre que le permite especificar qué prefijos deben tener una etiqueta anunciada. La lista de acceso de pares es una lista de acceso numerado estándar (1-99) o una lista de acceso con nombre que le permite especificar qué compañeros LDP pueden recibir los anuncios de las etiquetas. Los pares LDP coinciden con esta lista de acceso si los primeros 4

bytes de la ID del router LDP están cubiertos por los prefijos enumerados en esa lista de acceso. El uso de éste comando es restringir en muchos casos el número de etiquetas anunciadas a los prefijos que realmente se usan para reenviar el tráfico a través de la red MPLS. Por ejemplo, en el caso de MPLS VPN, los prefijos importantes para obtener el tráfico VPN del cliente a través de la red MPLS son los prefijos BGP nexthop, que suelen ser las interfaces de bucle invertido en los *router* PE. En este caso, puede elegir no anunciar los enlaces de etiqueta para los prefijos que pertenecen a las otras interfaces en los routers PE o P. No es necesario borrar el vecino LDP al que aplica el comando mpls ldp public-labels para que este tenga efecto.

No puede controlar el anuncio LDP de etiquetas para redes LC-ATM con LDP implementado con el comando mpls ldp advertirse-labels. Esto se debe a que las redes LC-ATM usan DoD en lugar del modo de anuncio de etiqueta UD. DoD tiene su propio comando para limitar la publicidad de etiquetas LDP. El comando mpls ldp request-labels se usa en lugar de mpls ldp advertise labels para interfaces LC-ATM.

La figura 87 se ve la red que previamente se muestra. El *router* Yakarta solo anuncia su propio prefijo *loopback* 0 y el *router* Berlin (prefijos 10.200.254.4/32 y 10.200.254.3/32) hacia el LDP vecino Lisboa (ID del *router* LDP 10.200.254.5).

Figura 87. **Anuncio controlado LDP**



Fuente: elaboración propia, empleando Visio 2013.

La configuración necesaria para esto se puede observar en la figura 88. No se debe olvidar de configurar *no mpls ldp advertise-labels*, también. Si se olvida este comando y solo configura los *mpls ldp advertise labels* para *prefix-access-list*, el LSR Yakarta aún envía etiquetas para todos los prefijos a través de LDP.

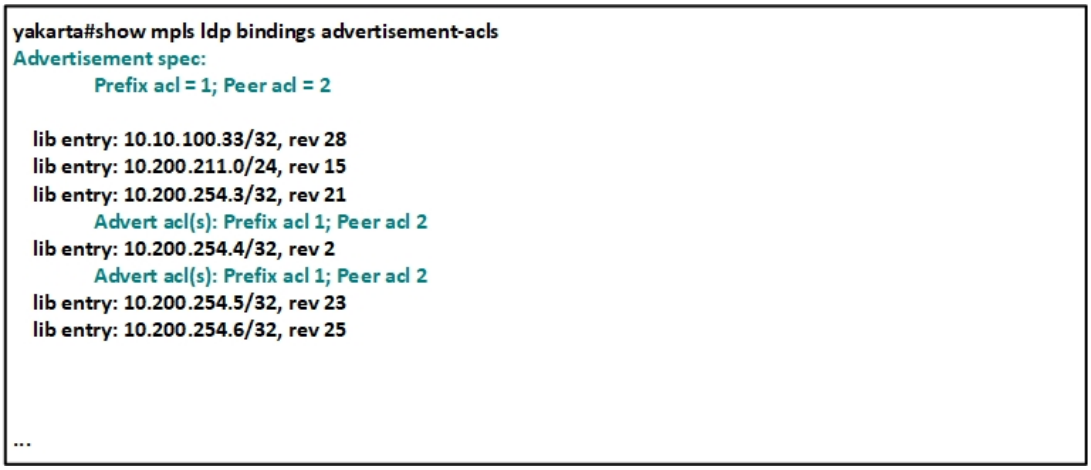
Figura 88. **Anuncio controlado LDP: configuración (arreglar Yakarta)**

```
!
hostname sydney
!
mpls ldp router-id Loopback0 force
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1 to 2
mpls label protocol ldp
!
access-list 1 permit 10.200.254.4
access-list 1 permit 10.200.254.3
access-list 1 deny any
access-list 2 permit 10.200.254.5
access-list 2 deny any
!
```

Fuente: elaboración propia, empleando Visio 2013.

Solo los prefijos 10.200.254.3/32 y 10.200.254.4/32 se anuncian a LDP vecino 10.200.254.5 (router Lisboa). La figura 89 muestra enlaces en el *router* de Yakarta como resultado de este filtrado en las etiquetas de enlaces.

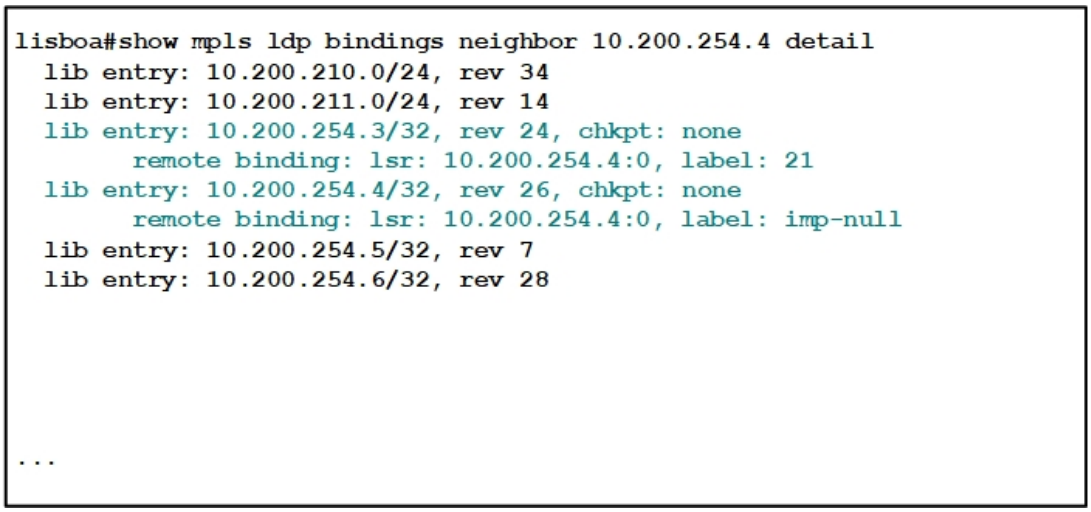
Figura 89. **Anuncio controlado LDP**



Fuente: elaboración propia, empleando Visio 2013.

Se observa en la figura 90 que todos los prefijos anunciados desde el *router* de Yakarta al *router* de Lisboa no tienen asociado más vinculaciones remotas.

Figura 90. **Listados de LSR Lisboa para el vecino 10.200.254.4**

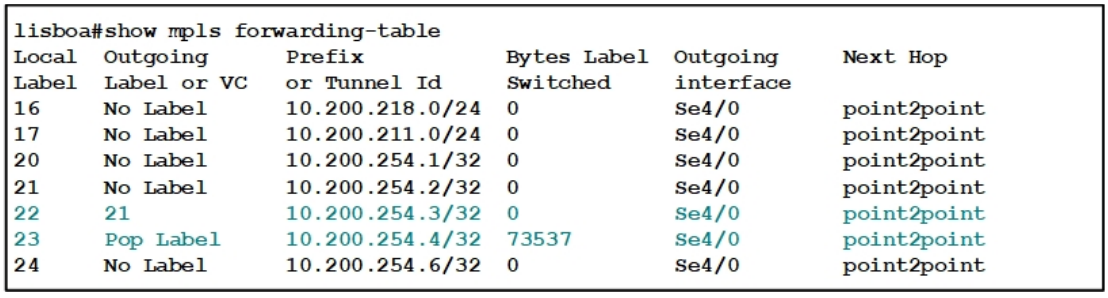


```
lisboa#show mpls ldp bindings neighbor 10.200.254.4 detail
  lib entry: 10.200.210.0/24, rev 34
  lib entry: 10.200.211.0/24, rev 14
  lib entry: 10.200.254.3/32, rev 24, chkpt: none
        remote binding: lsr: 10.200.254.4:0, label: 21
  lib entry: 10.200.254.4/32, rev 26, chkpt: none
        remote binding: lsr: 10.200.254.4:0, label: imp-null
  lib entry: 10.200.254.5/32, rev 7
  lib entry: 10.200.254.6/32, rev 28




...
```

Fuente: elaboración propia, empleando Visio 2013.

En el LFIB del *router* Lisboa, los dos prefijos 10.200.254.3/32 y 10.200.254.4/32 tienen una etiqueta de salida válida, mientras que los otros prefijos tienen *'no label'* asociada a ellos como etiquetas de salida. Puede ver el LFIB en el *router* Lisboa en la figura 91.

Figura 91. **LFIB en LSR Lisboa**



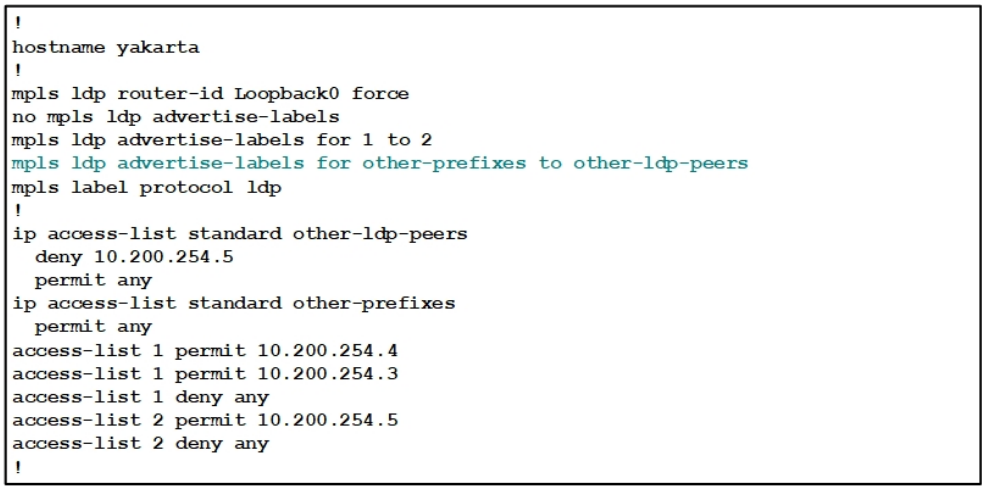
```
lisboa#show mpls forwarding-table
Local  Outgoing      Prefix           Bytes Label  Outgoing     Next Hop
Label  Label or VC   or Tunnel Id     Switched     interface
16     No Label      10.200.218.0/24  0            Se4/0        point2point
17     No Label      10.200.211.0/24  0            Se4/0        point2point
20     No Label      10.200.254.1/32  0            Se4/0        point2point
21     No Label      10.200.254.2/32  0            Se4/0        point2point
22     21            10.200.254.3/32  0            Se4/0        point2point
23     Pop Label     10.200.254.4/32  73537        Se4/0        point2point
24     No Label      10.200.254.6/32  0            Se4/0        point2point
```

Fuente: elaboración propia, empleando Visio 2013.

La implementación de Cisco IOS LDP le permite especificar más de un '*mpls ldp advertise labels*' para '*prefix-access-list*' al comando '*peer-access-list*'. Esto brinda mayor flexibilidad cuando se decide qué vinculaciones de etiqueta enviar a las que hace referencia LDP.

En la figura 92 es el mismo que el anterior, con la adición de otros *'mpls ldp advertise labels'* para *'prefix-access-list'* al comando *peer-access-list* en la configuración del *router*. Ahora, el *router* de Yakarta anuncia solo los enlaces de etiqueta para los dos prefijos a 10.200.254.4 y todas las vinculaciones de etiqueta para todos los prefijos a todos los demás pares de LDP.

Figura 92. **Anuncio controlado LDP**



Fuente: elaboración propia, empleando Visio 2013.

3.5.56. Filtrado de enlace de etiqueta de entrada MPLS LDP

Puede filtrar enlaces de etiquetas entrantes desde un vecino LDP. En efecto, esto es lo opuesto a la característica que impide la publicidad de las

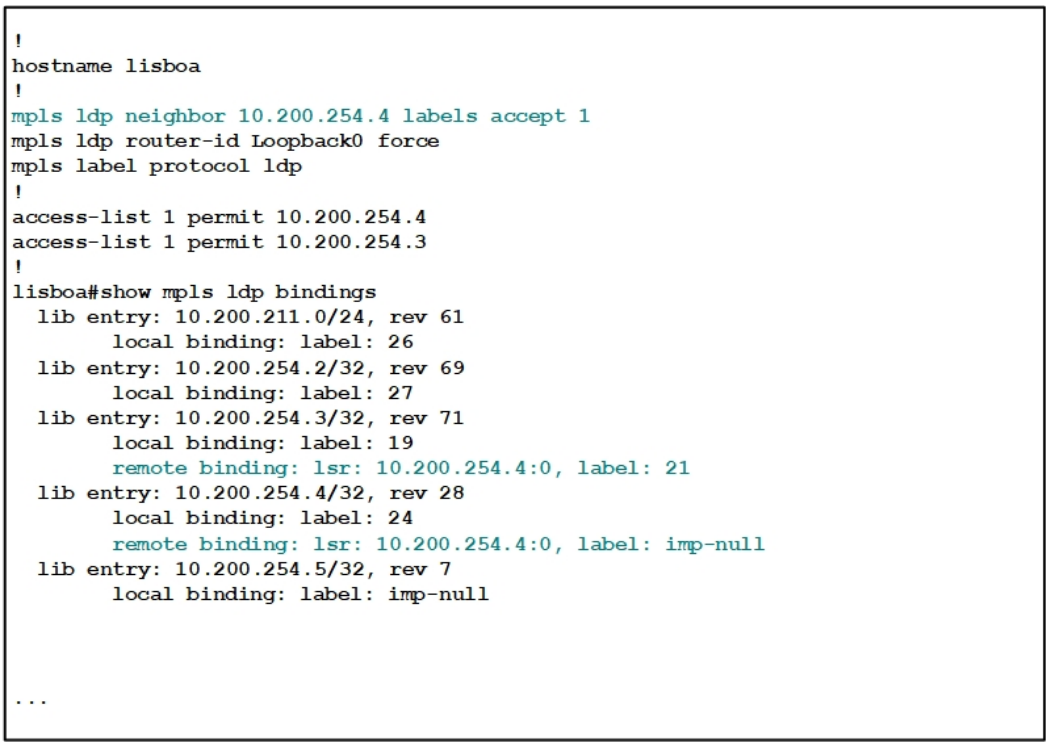
etiquetas. Puede utilizar filtrado de enlace de etiqueta entrante en el par LDP receptor si no puede aplicar el filtrado saliente de enlaces de etiqueta, como se describe en la sección anterior. Está característica puede limitar el número de enlaces de etiquetas almacenados en la LIB del *router*. Por ejemplo, puede filtrar todos los enlaces de etiquetas recibidos de los pares LDP, excepto los enlaces de etiquetas de las interfaces de bucle invertido de los *routers* PE en una red MPLS VPN. Por lo general, estas interfaces de bucle invertido tienen las direcciones IP de salto siguiente BGP, y los LSR pueden usar la etiqueta asociada con ese prefijo para reenviar el tráfico VPN del cliente etiquetado.

A continuación, se muestra el comando para habilitar el filtro de enlace entrante:

mpls ldp neighbor [vrf *vpn - name*] *nbr - address* labels accept *acl*

La figura 93 muestra el LSR de madrid que aplica está filtrado de enlace de etiqueta entrante de LDP al par LEP 10.200.254.4. Limita los enlaces de etiqueta aceptados a 10.200.254.3/32 y 10.200.254.4/32, los prefijos de bucle de los *routers* PE. Puede verificar que el LSR tenga enlaces de etiqueta remotos solo desde el par LDP especificado para los prefijos permitidos por la lista de acceso con el comando *show mpls ldp bindings*. El efecto de este filtrado de enlace de etiqueta entrante es el mismo que el filtrado de enlace de etiqueta saliente en el ejemplo.

Figura 93. **Ejemplo de filtrado de enlace de etiquetas entrantes LDP**



Fuente: elaboración propia, empleando Visio 2013.

3.5.57. Autoconfiguración LDP

LDP está habilitado en una interfaz configurando el comando de interfaz *mpls ip*. En un LSR, LDP generalmente está habilitado en todas las interfaces en las que está habilitado el IGP. Mucho más fácil que configurar *mpls ip* en cada interfaz por separado es habilitar la configuración automática de LDP para el IGP. Cada interfaz en la que se ejecuta el IGP tiene habilitado LDP. El comando del *router* OSPF para habilitar la autoconfiguración LDP es el siguiente:

- mpls ldp autoconfig [área *área - id*]

Como puede ver, se puede habilitar solo para un área OSPF específica. También puede desactivarlo desde interfaces específicas si lo desea. El comando de interfaz para deshabilitar la configuración automática de LDP en una interfaz es el siguiente:

- no mpls ldp igp autoconfig

Se observa en la figura 94, i'nterfaz de configuración' indica que LDP está habilitado a través del comando interface *mpls ip. 'IGP config'* indica que LDP está habilitado mediante el comando *mpls ldp autoconfig* del *router*.

Figura 94. **Ejemplo de configuración de autoconfiguración LDP**

```
!
hostname lisboa
!
router ospf 1
 mpls ldp autoconfig area 0
 router-id 10.200.254.5
 log-adjacency-changes
 network 10.200.254.0 0.0.0.255 area 0
 network 10.200.0.0 0.0.255.255 area 0
!
lisboa#show mpls interfaces detail
Interface Ethernet3/1:
        IP labeling enabled (ldp):
          Interface config
          IGP config
        LSP Tunnel labeling enabled
        BGP labeling not enabled
        MPLS operational

…

Interface Serial4/0:
        IP labeling enabled (ldp):
          Interface config
          IGP config
        LSP Tunnel labeling enabled
        BGP labeling not enabled
        MPLS operational
…

lisboa#show mpls ldp discovery detail
 Local LDP Identifier:
    10.200.254.5:0
    Discovery Sources:
    Interfaces:
        Ethernet3/1 (ldp): xmit/recv
            Enabled: Interface config, IGP config;
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.5
            LDP Id: 10.200.254.2:0
              Src IP addr: 10.200.215.1; Transport IP addr: 10.200.254.2
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 10.200.254.2/32
        Serial4/0 (ldp): xmit/recv
            Enabled: Interface config; IGP config;
            Hello interval: 5000 ms; Transport IP addr: 10.200.254.5
            LDP Id: 10.200.254.4:0
              Src IP addr: 10.200.216.2; Transport IP addr: 10.200.254.4
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Reachable via 10.200.254.4/32
```

Fuente: elaboración propia, empleando Visio 2013.

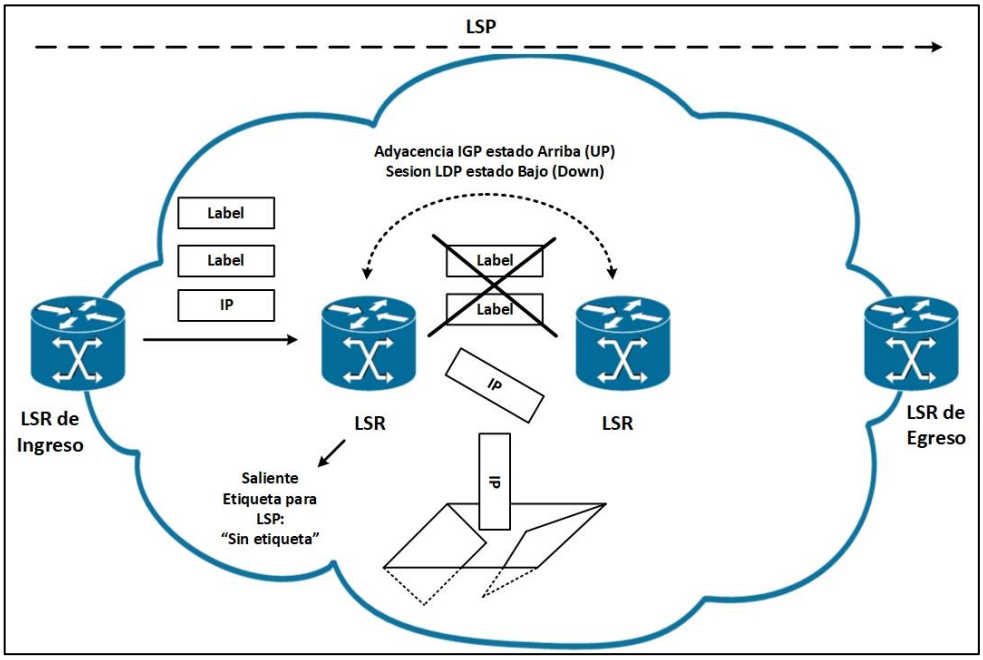
3.5.58. Sincronización MPLS LDP-IGP

Un problema con las redes MPLS es que LDP y el IGP de la red no están sincronizados. La sincronización significa que el reenvío de paquetes de una interfaz solo ocurre si tanto el IGP como el LDP acuerdan que está es el enlace de salida que se utilizará. Un problema común con las redes MPLS que ejecutan LDP es que cuando la sesión LDP se rompe en un enlace, el IGP todavía tiene ese enlace saliente; por lo tanto, los paquetes se siguen enviando fuera de ese enlace. Esto sucede porque el IGP instala la mejor ruta en la tabla de enrutamiento para cualquier prefijo. Por lo tanto, el tráfico para prefijos con un siguiente salto fuera de un enlace donde se rompe LDP se convierte en no etiquetado. Este no es un gran problema para las redes que solo ejecutan IPv4 sobre MPLS. En el punto donde LDP está roto, los paquetes se vuelven sin etiquetas. Los paquetes se envían como paquetes IPv4 hasta que se vuelven etiquetar en el siguiente LSR. Sin embargo, este es un problema para algo más que el caso de IPv4 sobre MPLS. Con MPLS VPN, AToM, *virtual private LAN switching* (VPLS) o IPv6 sobre MPLS, los paquetes no se deben etiquetar en la red MPLS. Si no se etiquetan, en el LSR no tiene la inteligencia para reenviar los paquetes y los descarta.

En el caso de MPLS VPN, los paquetes son paquetes IPv4, pero deben reenviarse según la tabla de enrutamiento VRF. Esta tabla es privada para un cliente y está presente solo en los *routers* LSR o PE de borde. Por lo tanto, cuando los paquetes MPLS VPN quedan sin etiquetar en los LSR centrales, los routers P, se descartan. Lo mismo es cierto para el tráfico AToM e IPv6. Los LSR centrales no pueden reenviarlos sin etiqueta. Una sesión de LDP baja mientras que la adyacencia de IGP entre dos LSR puede ocasionar problemas importantes porque puede perder mucho tráfico. En la figura 95 se muestra una

sesión LDP que está disminuyendo entre dos LSR en el núcleo de MPLS y se descartan los paquetes etiquetados.

Figura 95. **Sesión de LDP entre LSR**



Fuente: elaboración propia, empleando Visio 2013.

El mismo problema puede ocurrir cuando se reinician los LSR. El IGP puede ser más rápido en el establecimiento de las adyacencias que LDP puede establecer sus sesiones. Esto significa que el reenvío de IGP ya está sucediendo antes de que el LFIB tenga la información necesaria para iniciar el reenvío correcto de la etiqueta. Los paquetes se reenvían (sin etiquetar) incorrectamente o se descartan hasta que se establece la sesión LDP.

La solución es la sincronización MPLS LDP-IGP. Está característica asegura que el enlace no se use para reenviar tráfico (sin etiquetar) cuando la

sesión LDP a través del enlace está inactiva. Más bien, el tráfico se reenvía a otro enlace donde la sesión LDP aún está establecida.

El problema que resuelve LDP-IGP *Synchronization* no puede ocurrir con BGP y la distribución de etiquetas. Debido a que BGP se encarga de la publicidad vinculante y del plano de control para el enrutamiento IP, el problema antes mencionado no puede suceder. Aunque es posible que la adyacencia IGP está activa mientras el LDP está inactivo en un enlace, BGP está hacia arriba o hacia abajo, lo que significa que la instalación del prefijo IP en la tabla de enrutamiento por BGP está vinculada al anuncio de la etiqueta vinculante para ese prefijo por BGP.

3.5.59. Funcionamiento de la sincronización MPLS LDP-IGP

Cuando la sincronización MPLS LDP-IGP está activa para una interfaz, el IGP anuncia ese enlace con la métrica máxima hasta que se logre la sincronización, o hasta que la sesión LDP se ejecute a través de esa interfaz. La métrica de enlace máxima para OSPF es 65535 (hex 0xFFFF). No se utiliza ninguna ruta través de interfaz la donde LDP está inactivo a menos que sea la única ruta. (No hay otras rutas que tengan una mejor métrica). Después de que se establezca la sesión LDP y se hayan intercambiado los enlaces de etiqueta, el IGP anuncia el enlace con su métrica IGP normal. En ese punto, el tráfico se cambia de etiqueta a través de esa interfaz. Básicamente, OSPF no forma una adyacencia a través de un enlace si la sesión LDP no se establece primero a través de este enlace (OSPF no envía *Hellos* en el enlace).

Hasta que se establezca la sesión LDP o hasta que el temporizador de retención de sincronización haya expirado, la adyacencia OSPF no está establecida. Sincronizado aquí significa que los enlaces de etiquetas locales se

han enviado a través de la sesión LDP al vecino LDP. Sin embargo, cuando la sincronización se enciende en el *router* A y ese *router* tiene un solo enlace al *router* B y ninguna otra conectividad IP al *router* B a través de otra ruta (esto significa a través de otros *routers*), la adyacencia OSPF nunca parece. OSPF espera a que aparezca la sesión LDP, pero la sesión LDP no puede aparecer porque el router A no puede tener la ruta para el ID del router LDP del router B en su tabla de enrutamiento. La adyacencia de OSPF y LDP puede quedarse abajo para siempre en esta situación. Si el *router* A tiene solo un *router* B como vecino, la ID del *router* LDP del *router* B no es accesible; esto significa que no existe ninguna ruta para ello en la tabla de enrutamiento del *router* A. En ese caso, la sincronización LDP-IGP detecta que el par no es alcanzable y permite que OSPF muestre la adyacencia de todos los modos. En este caso, el enlace se anuncia con la métrica máxima hasta que se produce la sincronización. Esto hace que el camino a través de ese enlace sea un camino de último recurso.

En algunos casos, el problema con la sesión LDP puede ser persistente; por lo tanto, tal vez no sea deseable esperar a que se establezca la adyacencia IGP. La solución para esto es configurar un temporizador de retención para sincronización. Si el temporizador expira antes de que se establezca la sesión LDP, la adyacencia OSPF se construye de todos modos. Si todo está bien con LDP a través de ese enlace, LDP también forma una sesión a través del enlace. Mientras OSPF está esperando para mostrar su adyacencia hasta que se sincronice LDP, el estado de la interfaz OSPF está inactivo y OSPF no envía Hellos en ese enlace.

3.5.60. Configuración de sincronización MPLS LDP-IGP

La sincronización MPLS LDP-IGP está habilitada para el proceso IGP. Esto significa que está configurado para un IGP, y se aplica a todas las

interfaces en las que se ejecuta el IGP. El comando para habilitarlo para el IGP es *mpls ldp sync*, y está configurado bajo el proceso del *router*. Puede deshabilitar la sincronización MPLS LDP-IGP en una interfaz particular con el comando *no mpls ldp igp sync*. De manera predeterminada, si no se logra la sincronización, el IGP espera indefinidamente para que aparezca la adyacencia. Puede cambiar esto con el comando *global mpls ldp igp sync holddown msecs*, que indica al IGP que espere solo la hora configurada. Después de que finaliza la sincronización, el temporizador de retención expira, el IGP forma una adyacencia a través del enlace. Mientras la adyacencia IGP este activa, mientras que la sesión LDP no está sincronizada, el IGP anuncia el enlace con la métrica máxima.

Cuando OSPF está esperando a que LDP se sincronice, se da el siguiente mensaje 'la interfaz está inactiva y pendiente LDP'. En ese estado, OSPF no forma adyacencia. Cuando la adyacencia OSPF está activa pero la sesión LDP no, OSPF da el siguiente mensaje que es 'la interfaz está activa y envía una métrica máxima'. La interfaz no se usa para reenviar el tráfico en este caso, a menos que sea la única ruta de salida del LSR. En la figura 96 se muestra la configuración para la sincronización MPLS LDP-IGP.

Figura 96. **Ejemplo de configuración de la sincronización de MPLS LDP-IGP**

```
!
hostname lisboa
!
router ospf 1
 mpls ldp sync
 router-id 10.200.254.5
 log-adjacency-changes
 network 10.200.254.0 0.0.0.255 area 0
 network 10.200.0.0 0.0.255.255 area 0
!
lisboa#show ip ospf mpls ldp interface serial 4/0
Serial4/0
  Process ID 1, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is not configured
  Interface is up
```

Fuente: elaboración propia, empleando Visio 2013.

En la figura 97 se muestra el resultado del comando *show ip ospf mpls ldp interface* cuando la interfaz está respaldada después de que se apagó, pero LDP tiene un problema y la sesión LDP no aparece. Como resultado, OSPF no forma una adyacencia. De hecho, el estado OSPF de la interfaz es ABAJO.

Figura 97. **Sincronización MPLS LDP-IGP**

```
lisboa#show ip ospf mpls ldp interface serial 4/0
Serial4/0
  Process ID 1, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is not configured
  Interface is down and pending LDP
lisboa#show ip ospf interface serial 4/0
Serial4/0 is up, line protocol is up
  Internet Address 10.200.216.1/24, Area 0
  Process ID 1, Router ID 10.200.254.5, Network Type POINT_TO_POINT, Cost: 10
  Transmit Delay is 1 sec, State DOWN,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
lisboa#show interfaces serial 4/0
Serial4/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 10.200.216.1/24
```

Fuente: elaboración propia, empleando Visio 2013.

Para evitar que OSPF espere indefinidamente a que aparezca LDP, puede configurar un temporizador de retención como en la figura 98. Después de que expira el temporizador de retención, OSPF forma una adyacencia, incluso cuando LDP aún no está sincronizado.

Figura 98. **Ejemplo de sincronización MPLS LDP-IGP con temporizador de espera**

```
!
hostname lisboa
!
mpls label protocol ldp
mpls ldp igp sync holddown 30000
!
lisboa#show ip ospf mpls ldp interface serial 4/0
Serial4/0
  Process ID 1, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is configured : 30000 msecs
  Holddown timer is running and is expiring in 1708 msecs
  Interface is down and pending LDP

lisboa#
22:21:00: %OSPF-5-ADJCHG: Process 1, Nbr 10.200.254.4 on Serial4/0 from LOADING to FULL,
 Loading Done
```

Fuente: elaboración propia, empleando Visio 2013.

Después de que expira el temporizador de retención, se forma la adyacencia OSPF, pero la sesión LDP aún está inactiva debido a un problema persistente de LDP en el enlace. Mientras está estado permanezca, OSPF anunciara el enlace con la métrica OSPF máxima de 65535. Observese las figuras 99 y 100 para la corroboración de datos. El estado de sincronización es 'sincronización no lograda'.

Figura 99. **Sincronización MPLS LDP-IGP: métrica máxima de publicidad**

```
lisboa#show ip ospf mpls ldp interface serial 4/0
Serial4/0
  Process ID 1, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is configured : 30000 msecs
  Holddown timer is not running
  Interface is up and sending maximum metric

lisboa#show ip ospf database router 10.200.254.5

            OSPF Router with ID (10.200.254.5) (Process ID 1)

                Router Link States (Area 0)

  LS age: 276
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 10.200.254.5
  Advertising Router: 10.200.254.5
  LS Seq Number: 800000CA
  Checksum: 0x43D7
  Length: 72
  Number of Links: 4

   Link connected to: another Router (point-to-point)
    (Link ID) Neighboring Router ID: 10.200.254.4
    (Link Data) Router Interface address: 10.200.216.1
     Number of TOS metrics: 0
      TOS 0 Metrics: 65535

   Link connected to: a Stub Network
    (Link ID) Network/subnet number: 10.200.216.0
    (Link Data) Network Mask: 255.255.255.0
     Number of TOS metrics: 0
      TOS 0 Metrics: 10
...

lisboa#show mpls ldp igp sync serial 4/0
    Serial4/0:
        LDP configured; LDP-IGP Synchronization enabled.
        Sync status: sync not achieved; peer reachable.
        IGP holddown time: 30000 milliseconds.
        IGP enabled: OSPF 1
```

Fuente: elaboración propia, empleando Visio 2013.

El resultado de anunciar el enlace con una métrica máxima es que el LSR no puede usar el enlace para reenviar paquetes. Si un paquete MPLS AToM, IPv6, VPLS o cualquier paquete etiquetado con dos más o más etiquetas debían llegar al *router* Lisboa y deben ser reenviados en la interfaz serial 4/0, mientras que LDP está inactivo y la sincronización LDP-IGP no existe, aquellos los paquetes se descartarían con la sincronización LDP-IGP, estos paquetes se enlutarían la otra interfaz, donde se establece la sesión LDP.

El siguiente comando de depuración proporciona información de depuración en la sincronización LDP.

debug mpls ldp sync [interface < name >] [peer - acl < acl >]

En la figura 100 se muestra el resultado del comando *debug mpls ldp igp sync*.

Figura 100. **Información de depuración MPLS LDP-IGP**

```
lisboa#debug mpls ldp igp sync interface serial 4/0
LDP-IGP Synchronization debugging is on for interface Serial4/0
lisboa#
22:42:34: %LINK-3-UPDOWN: Interface Serial4/0, changed state to up
22:42:34: LDP-SYNC: Se4/0: queue swif_updown, set INTFADDR_PENDING.
22:42:34: LDP-SYNC: Se4/0: process swif_updown, clear INTFADDR_PENDING.
22:42:35: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/0, changed state to up
22:43:14: %OSPF-5-ADJCHG: Process 1, Nbr 10.200.254.4 on Serial4/0 from LOADING to FULL,
 Loading Done
lisboa#
22:44:31: LDP-SYNC: Se4/0: No session or session has not send initial update, ignore adj
joining event.
22:44:31: %LDP-5-NBRCHG: LDP Neighbor 10.200.254.4:0 is UP
22:44:31: LDP-SYNC: Se4/0: session 10.200.254.4:0 came up, sync_achieved up
22:44:31: LDP-SYNC: Se4/0, OSPF 1: notify status (required, achieved, no delay, holddown
 30000)
22:44:31: OSPF: schedule to build router LSA after notification from LDP
```

Fuente: elaboración propia, empleando Visio 2013.

Si el par no es alcanzable, como en la figura 102, el IGP forma una adyacencia de todos modos para darle al LDP la oportunidad de construir una sesión LDP a través de ese enlace. Esto sucede cuando este enlace es la única ruta (sigue funcionando) para el router vecino.

Figura 101. **Peer no alcanzable**

```
lisboa#show mpls ldp igp sync interface serial 4/0
    Serial4/0:
        LDP configured; LDP-IGP Synchronization enabled.
        Sync status: sync not achieved; peer not reachable.
        IGP holddown time: infinite.
        IGP enabled: OSPF 1
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.61. Protección de sesión MPLS LDP

Un problema común en las redes es el flapping de enlaces. El aleteo de enlaces puede tener varias causas, pero el objetivo de este libro no es profundizar en esto. Los enlaces de aleteo tienen un impacto importante en la convergencia de la red. Debido a que la adyacencia IGP y la sesión LDP se ejecutan a través del enlace, se desactivan cuando el enlace se desactiva. Esto es desafortunado, especialmente porque el enlace generalmente no está fuera de servicio por mucho tiempo. Sin embargo, el impacto es bastante severo, porque el protocolo de enrutamiento y el LDP pueden demorar un tiempo en reconstruir el vecindario. LDP tiene que reconstruir la sesión LDP y debe intercambiar los enlaces de etiquetas nuevamente. Para evitar tener que reconstruir la sesión LDP por completo, puede protegerla. Cuando la sesión LDP entre dos LSR conectados directamente está protegida, se genera una sesión LDP específica entre los dos LSR. Cuando el enlace directamente

conectado baja entre los dos LSR, la sesión LDP objetivo se mantiene activa mientras exista una ruta alternativa entre los dos LSR. La adyacencia del enlace LDP se elimina cuando el enlace se desactiva, pero la adyacencia dirigida mantiene la sesión LDP activa. Cuando el enlace vuelve a aparecer, el LSR no necesita restablecer la sesión LDP; por lo tanto, la convergencia es mejor. El comando global para habilitar la protección de sesión LDP es esta:

mpls ldp session protection [vrf *vpn - name*] [for *acl*] [duration *seconds*]

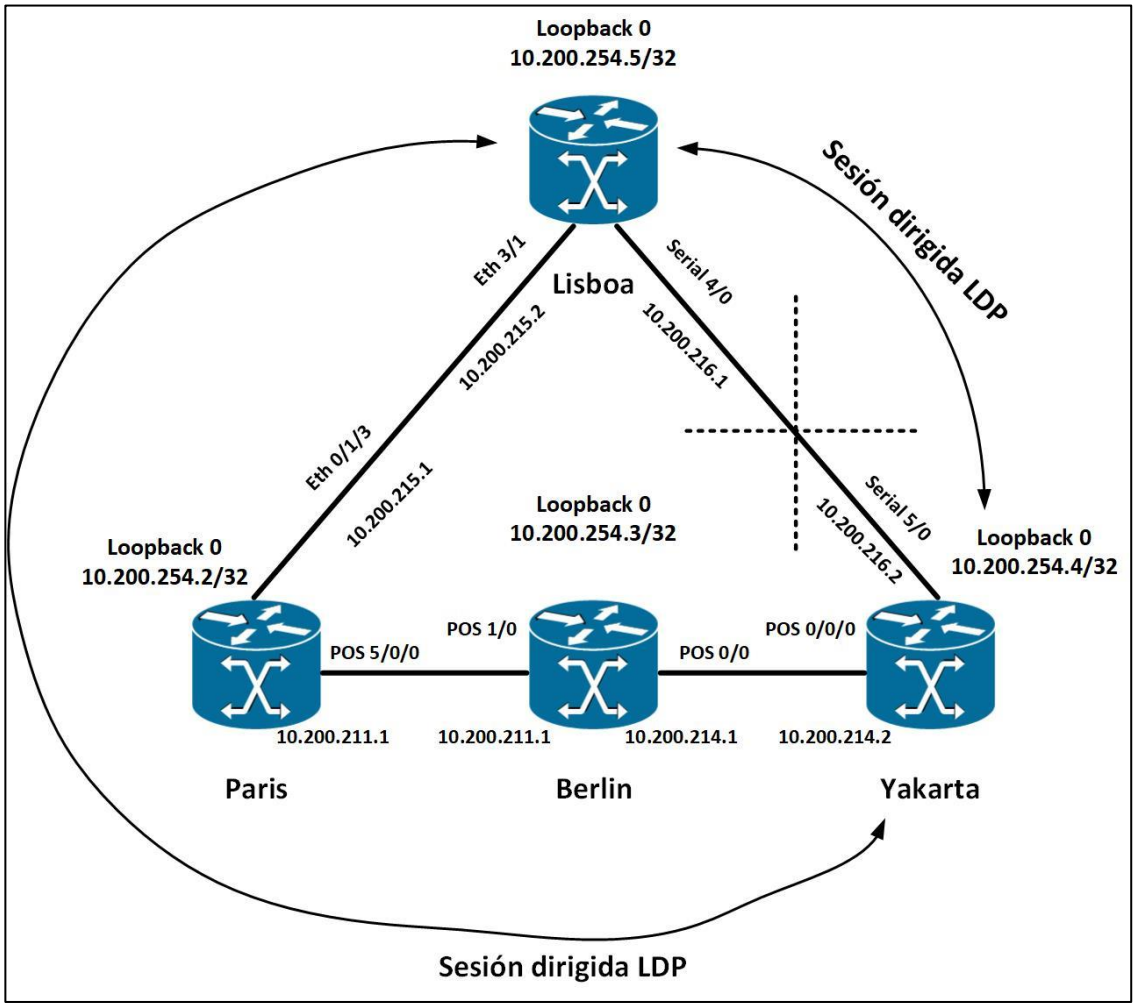
La lista de acceso (ACL) se puede configurar y le permite especificar los pares LDP que deben protegerse. Debe contener el identificador del *router* LDP de los vecinos LDP que necesitan protección. La duración es el tiempo que la protección (la sesión LDP específica) debe permanecer en su lugar después de que la adyacencia del enlace LDP haya disminuido. El valor predeterminado es infinito.

Para que la protección funcione, debe habilitarla en ambos LSR. Si esto no es posible, puede habilitarlo en un LSR, y el otro LSR puede aceptar los Hellos LDP específicos configurando el comando mpls ldp discovery targeted-hello accept.

En la figura 102 se observa un ejemplo. La protección de sesión LDP está habilitada en los cuatro routers. El LSR Lisboa tiene dos sesiones LDP: una con París y otra con Yakarta. Cuando el enlace de Lisboa-Yakarta falla, la sesión de LDP seleccionada se mantiene mientras se reencamina en la ruta Lisboa-Paris-Berlin-Yakarta. El ejemplo muestra la sesión LDP en Lisboa al *router* Yakarta antes de que el enlace se cayera. El enlace Lisboa-Yakarta luego baja. Puede ver un mensaje de registro para la sesión LDP cuando el enlace se desactiva y

cuando vuelve a aparecer el enlace. El primer mensaje de registro indica que la sesión LDP entró en estado de protección; el segundo indica que la sesión LDP se ha recuperado con éxito.

Figura 102. **Sesión de protección LDP**



Fuente: elaboración propia, empleando Visio 2013.

Figura 103. **Ejemplo de sesión de protección LDP**

```
madrid#show mpls ldp neighbor serial 4/0 detail
    Peer LDP Ident: 10.200.254.4:0; Local LDP Ident 10.200.254.5:0
        TCP connection: 10.200.254.4.646 - 10.200.254.5.21396
        State: Oper; Msgs sent/rcvd: 43/42; Downstream; Last TIB rev sent 63
        Up time: 00:15:32; UID: 18; Peer Id 0;
        LDP discovery sources:
          Targeted Hello 10.200.254.5 -> 10.200.254.4, active, passive;
            holdtime: infinite, hello interval: 10000 ms
          Serial4/0; Src IP addr: 10.200.216.2
            holdtime: 15000 ms, hello interval: 5000 ms
          Addresses bound to peer LDP Ident:
            10.200.254.4 10.200.214.2 10.200.217.1 10.200.216.2
          Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
          Clients: Dir Adj Client
          LDP Session Protection enabled, state: Ready
              duration: infinite

madrid#show mpls ldp discovery
 Local LDP Identifier:
    10.200.254.5:0
    Discovery Sources:
    Interfaces:
        Ethernet3/1 (ldp): xmit/recv
            LDP Id: 10.200.254.2:0
        Serial4/0 (ldp): xmit/recv
            LDP Id: 10.200.254.4:0
    Targeted Hellos:
        10.200.254.5 -> 10.200.254.4 (ldp): active/passive, xmit/recv
            LDP Id: 10.200.254.4:0
        10.200.254.5 -> 10.200.254.2 (ldp): active/passive, xmit/recv
            LDP Id: 10.200.254.2:0

madrid#
02:48:38: %OSPF-5-ADJCHG: Process 1, Nbr 10.200.254.4 on Serial4/0 from FULL to DOWN,
Neighbor Down: Interface down or detached
02:48:39: %LINK-3-UPDOWN: Interface Serial4/0, changed state to down
02:48:39: %LDP-5-SP: 10.200.254.4:0: session hold up initiated
02:48:40: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/0, changed state to down
madrid#show mpls ldp neighbor 10.200.254.4 detail
    Peer LDP Ident: 10.200.254.4:0; Local LDP Ident 10.200.254.5:0
        TCP connection: 10.200.254.4.646 - 10.200.254.5.21396
        State: Oper; Msgs sent/rcvd: 55/51; Downstream; Last TIB rev sent 69
        Up time: 00:17:18; UID: 18; Peer Id 0;
        LDP discovery sources:
          Targeted Hello 10.200.254.5 -> 10.200.254.4, active, passive;
            holdtime: infinite, hello interval: 10000 ms
        Addresses bound to peer LDP Ident:
          10.200.254.4 10.200.214.2 10.200.217.1 10.200.216.2
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
        Clients: Dir Adj Client
        LDP Session Protection enabled, state: Protecting
            duration: infinite

madrid#
02:49:10: %LINK-3-UPDOWN: Interface Serial4/0, changed state to up
02:49:11: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/0, changed state to up
02:49:15: %LDP-5-SP: 10.200.254.4:0: session recovery succeeded
```

Fuente: elaboración propia, empleando Visio 2013.

Finalmente, una característica útil de LDP es LDP *Graceful Restart*. Especifica un mecanismo para que los pares LDP conserven el estado de reenvío MPLS cuando la sesión LDP falla. Como tal, el tráfico puede seguir siendo reenviado sin interrupción, incluso cuando se reinicie la sesión LDP.

3.5.62. MPLS VPN

MPLS VPN, o MPLS Virtual Private Networks, es la implementación más popular y extendida de la tecnología MPLS. Su popularidad ha crecido exponencialmente desde que fue inventada, y sigue creciendo constantemente. Aunque la mayoría de los proveedores de servicios lo han implementado como reemplazo de los servicios Frame Relay y ATM que eran populares antes, MPLS VPN ahora está viendo un interés creciente de las grandes empresas que lo ven como el siguiente paso en el diseño de su red. MPLS VPN puede proporcionar escalabilidad y dividir la red en redes más pequeñas separadas, lo que a menudo es necesario en las redes empresariales más grandes, donde la infraestructura TI común tiene que ofrecer redes aisladas a departamentos individuales. Muchos proveedores de servicios que han ejecutado MPLS VPN durante años ahora están buscando interconectar su red a las redes MPLS VPN de otros proveedores de servicios para mejorar la escalabilidad y la facilidad de operación de su red. Aquí es donde entra en escena la VPN MPLS inter-autómata y el operador del operador (CsC).

3.5.63. Definición de una VPN

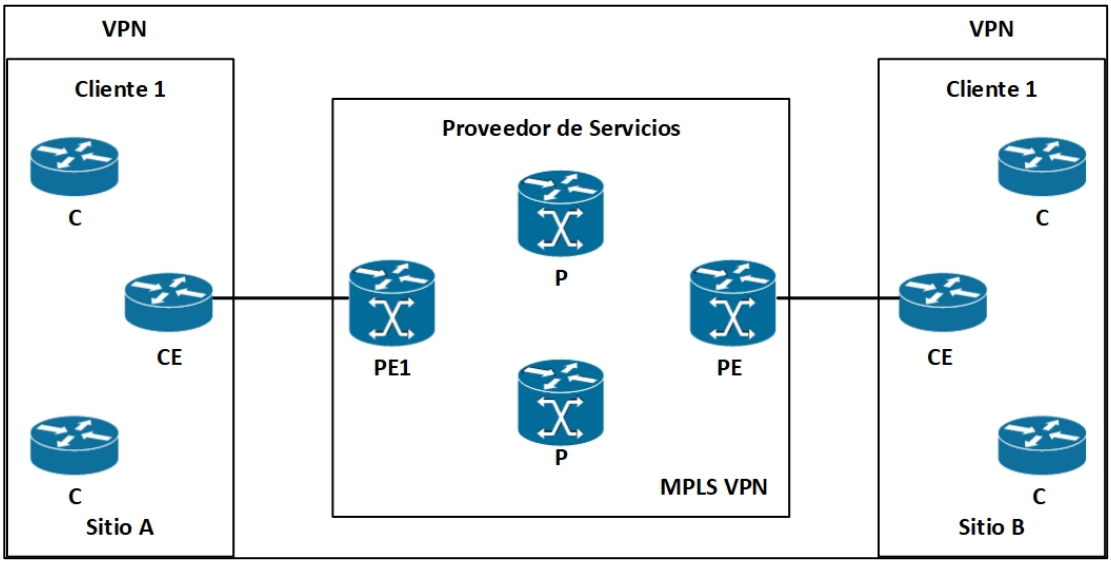
Una VPN es una red que emula una red privada a través de una infraestructura común. La VPN podría proporcionar comunicación en la capa 2 o 3 del modelo OSI. La VPN generalmente pertenece a una compañía y tiene varios sitios interconectados a través de la infraestructura del proveedor de

servicios común. La red privada requiere que todos los sitios de los clientes se puedan interconectar y estén completamente separados de otras VPN. Ese es el requisito de conectividad mínimo. Sin embargo, los modelos VPN en la capa IP pueden requerir más que eso. Pueden proporcionar conectividad a internet. MPLS VPN ofrece todo esto. Las VPN MPLS son posibles porque el proveedor de servicios ejecuta MPLS en la red troncal, que suministra un desacoplamiento del plano de reenvío y el plano de control que IP no tiene.

3.5.64. Modelo MPLS VPN

Es importante familiarizarse con la terminología relativa a MPLS VPN. Mire la figura para ver una descripción esquemática del modelo MPLS VPN. Un proveedor de servicios proporciona la infraestructura pública común que usan los clientes.

Figura 104. **Vista esquemática de una red MPLS VPN**



Fuente: elaboración propia, empleando Visio 2013.

Un *router* PE es un *router* del proveedor (PE). Tiene una conexión directa con el router del cliente (CE) en la capa 3. Un router del proveedor (P) es un router sin la conexión directa a los routers del cliente. En la implementación MPLS VPN, los *routers* P y PE ejecutan MPLS. Esto significa que deben distribuir etiquetas entre ellos y reenviar los paquetes etiquetados.

Un *router* CE tiene una conexión directa de capa 3 con el *router* PE. Un *router* de cliente (C) es un *router* sin una conexión directa con el *router* PE. Un *router* CE no necesita ejecutar MPLS.

Debido a que los *routers* CE y PE interactúan en la capa 3, deben ejecutar un protocolo de enrutamiento (o enrutamiento estático) entre ellos. El *router* CE tiene un solo para fuera de su propio sitio: el *router* PE. Si el *router* CE tiene múltiples domicilios, puede analizar con múltiples *routers* PE. El *router* CE no es compatible con ninguno de los *routers* CE de otros sitios en la red del proveedor de servicios, como ocurre con el modelo de superposición. El nombre modelo de punto a punto se deriva del hecho de que el CE y PE forman un igual en la capa 3.

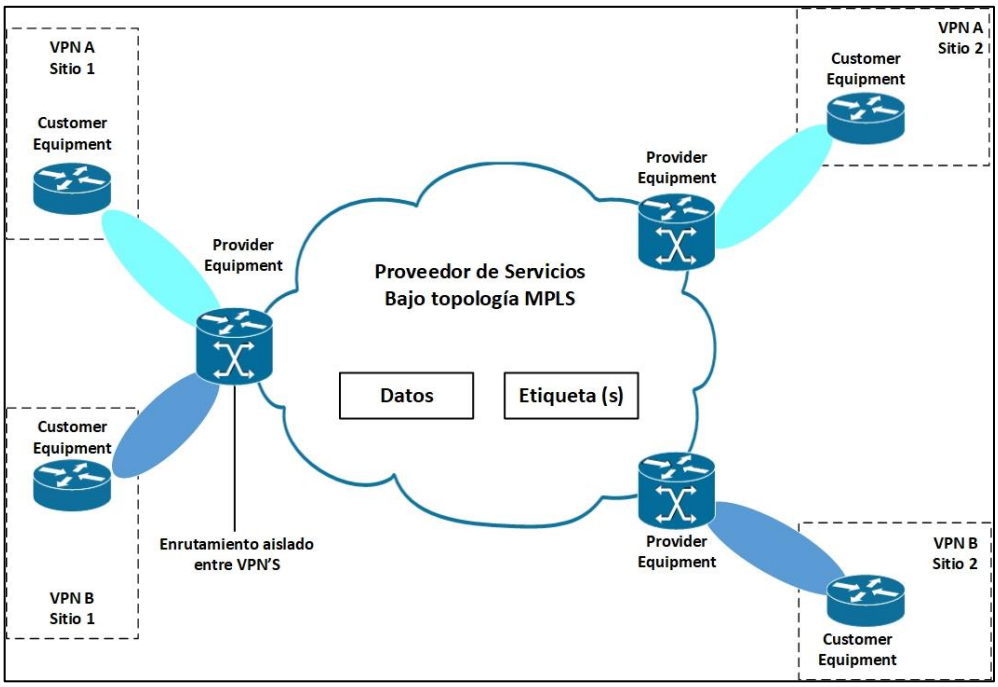
El P en VPN significa privado. Como tal, los clientes del proveedor de servicios pueden tener su propio esquema de direccionamiento IP. Esto significa que pueden usar direcciones IP registradas, pero también direcciones IP privadas (consultar RFC 1918) o incluso direcciones IP que también utilizan otros clientes que se conectan al mismo proveedor del servicio, esto causaría problemas, porque los routers P se confundirían. Si el esquema de direccionamiento IP privado y superpuesto no está permitido, entonces cada cliente debe usar un rango de direcciones único. En ese caso, los paquetes pueden reenviarse buscando la dirección IP de destino en cada *router* de la red del proveedor de servicios. Esto significa que todos los *routers* P y PE deben

tener la tabla de enrutamiento completa de cada cliente. Está sería una gran tabla de enrutamiento. El único protocolo de enrutamiento que es capaz de transportar una gran cantidad de rutas es *border gateway protocol* (BGP). Esto significa que todos los routers P y PE tendrían que ejecutar BGP interno (iBGP) entre ellos. Sin embargo, esto no es una VPN, porque no es privado para los clientes.

Una solución escalable sería tener los routers P completamente ajenos a las VPN. Entonces los routers P no estarían agobiados por tener información de enrutamiento para rutas VPN. Los paquetes IP del cliente están etiquetados en la red del proveedor del servicio para lograr una VPN privada para cada cliente. Además, los routers P ya no necesitan tener la tabla de enrutamiento de los clientes mediante el uso de dos etiquetas MPLS. Por lo tanto, BGP no es necesario en los routers P. Las rutas VPN solo se conocen en los routers PE. Como tal, el conocimiento de VPN solo está presente en los routers de borde de la red MPLS VPN, lo que hace que la solución MPLS VPN sea escalable.

En la figura 105 se muestra el modelo MPLS VPN: paquetes de conmutación de etiquetas en la red del proveedor de servicios y *routers* PE que son conscientes de VPN.

Figura 105. **Modelo MPLS VPN**



Fuente: elaboración propia, empleando Visio 2013.

3.5.65. Descripción arquitectónica de MPLS VPN

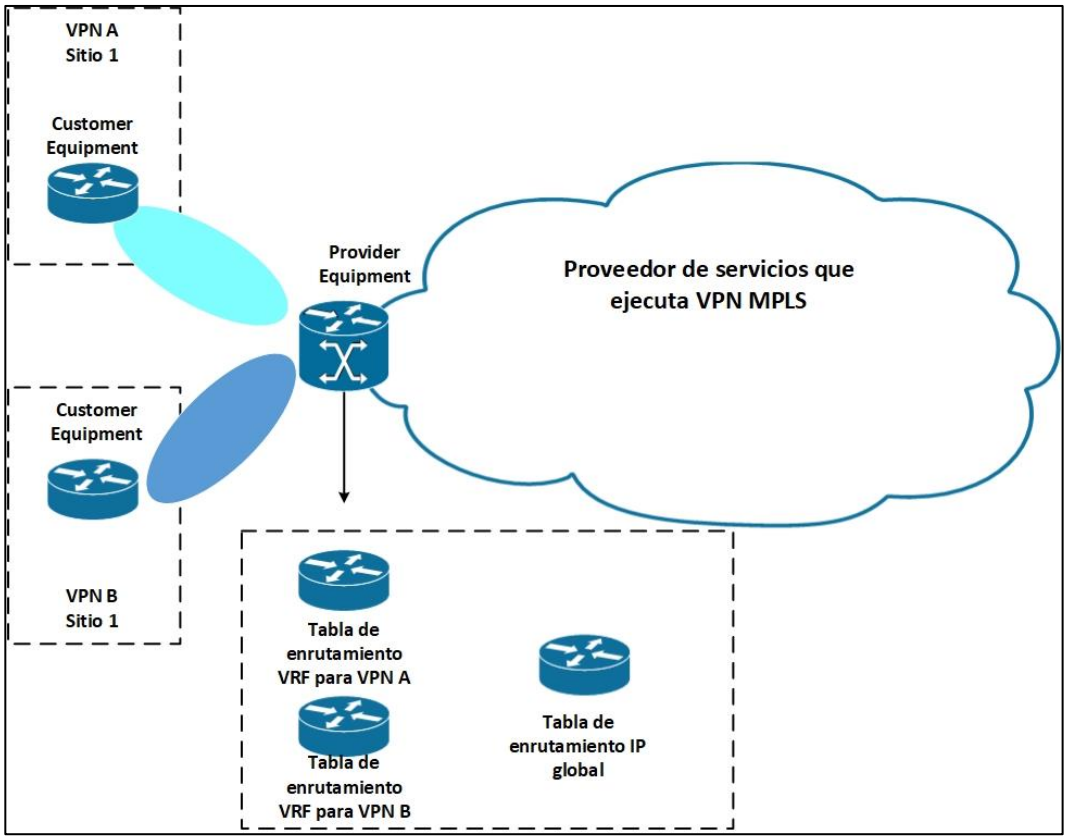
Para lograr MPLS VPN, se necesita algunos bloques de construcción básicos en los routers PE. Estos bloques de construcción son los siguientes: VRF, distintivo de ruta (RD, *route distinguisher*), objetivos de ruta (RT, *route targets*), propagación de ruta a través de MP-BGP y reenvío de paquetes etiquetados.

3.5.66. *Virtual routing forwarding*

Un enrutamiento / reenvío virtual (VRF) es una instancia de enrutamiento y reenvío VPN. Es el nombre para la combinación de la tabla de enrutamiento

VPN, la tabla VRF Cisco Express Forwarding (CEF) y los protocolos de enrutamiento IP asociados en el *router* PE. Un *router* PE tiene una instancia VRF para cada VPN conectada. Mire la figura para ver la que un router PE contiene la tabla de enrutamiento IP global, pero también una tabla de enrutamiento VRF por VPN conectada al PE.

Figura 106. **VRF's en un router PE**



Fuente: elaboración propia, empleando Visio 2013.

Debido a que el enrutamiento debe ser independiente y privado para cada cliente (VPN) en un router PE, cada VPN debe tener su propia tabla de enrutamiento. Esta tabla de enrutamiento privada se denomina tabla de

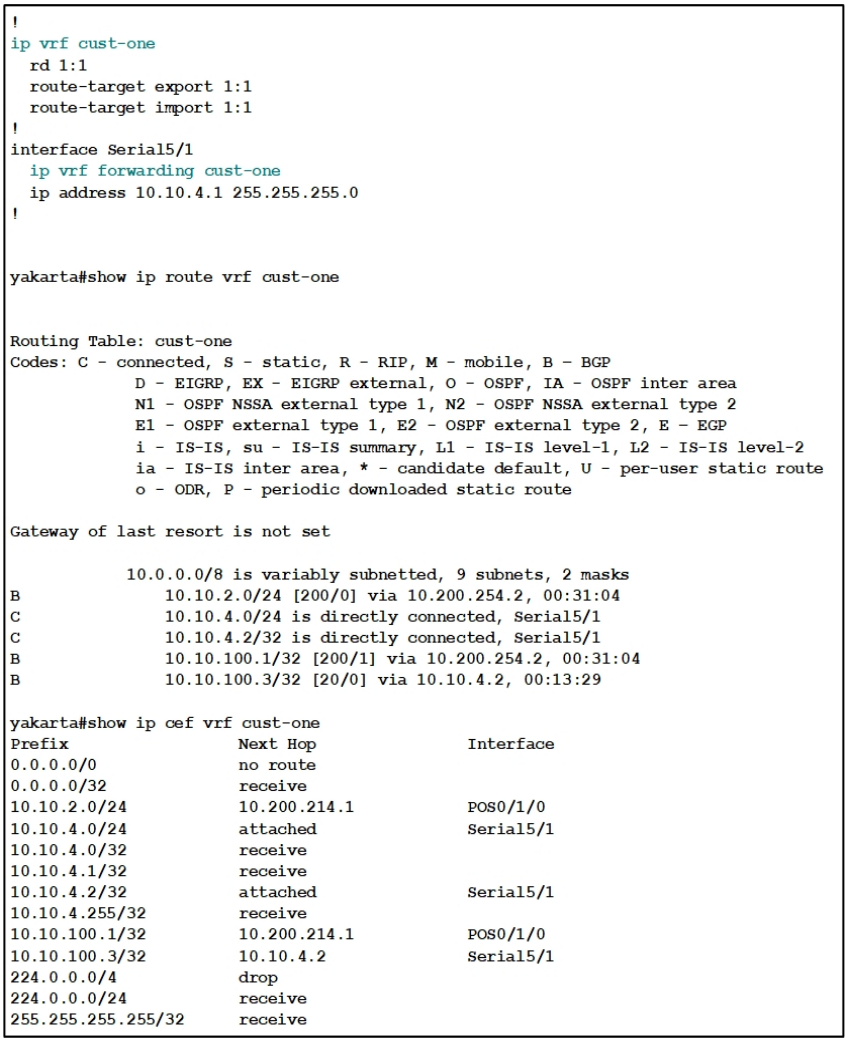
enrutamiento VRF. La interfaz en el router PE hacia el *router* CE puede pertenecer a un solo VRF. Como tal, todos los paquetes IP recibidos en la interfaz VRF se identifican sin ambigüedad como pertenecientes a ese VRF. Debido a que hay una tabla de enrutamiento separada por VPN, hay una tabla CEF separada por VPN para reenviar estos paquetes en el router PE. Esta es la tabla VRF CEF. Al igual que con la tabla de enrutamiento global y la tabla CEF global, la tabla VRF CEF se deriva de la tabla de enrutamiento VRF.

Usted crea el VRF en el *router* PE con el comando ip vrf. Utiliza el comando de reenvío ip vrf para asignar interfaces PE-CE en el *router* PE a un VRF. Puede asignar una interfaz a un solo VRF, pero puede asignar varias interfaces al mismo VRF. El *router* PE luego crea automáticamente una tabla de enrutamiento VRF y una tabla CEF. La tabla de enrutamiento VRF no difiere de una tabla enrutamiento normal en Cisco IOS aparte de que se usa solo para un conjunto de sitios VPN y está completamente separada de todas las demás tablas de enrutamiento. La tabla de enrutamiento tal como la conoce hasta ahora se denominará tabla de enrutamiento global o predeterminada. Observe el ejemplo donde el VRF configurado es el VRF *cust-one*.

La tabla de enrutamiento VRF *cust-one* tiene prefijos que están llenos de protocolos de enrutamiento dinámico y enrutamiento estático, al igual que la tabla de enrutamiento global. El concepto de métricas, distancia, siguiente salto, etc. no cambia. Debido a que la instancia VRF está asociada a interfaces, solo los paquetes IP que ingresan al router PE a través de esas interfaces VRF se envían de acuerdo con esa tabla VRF CEF.

En Cisco IOS, CEF es el único método de conmutación compatible para reenviar paquetes IP desde la interfaz VRF. Como tal, CEF debe estar habilitado globalmente en todos los routers PE y todas las interfaces VRF.

Figura 107. **Configurando VRF**



```
!
ip vrf cust-one
  rd 1:1
  route-target export 1:1
  route-target import 1:1
!
interface Serial5/1
  ip vrf forwarding cust-one
  ip address 10.10.4.1 255.255.255.0
!


yakarta#show ip route vrf cust-one


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

Gateway of last resort is not set

            10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
B               10.10.2.0/24 [200/0] via 10.200.254.2, 00:31:04
C               10.10.4.0/24 is directly connected, Serial5/1
C               10.10.4.2/32 is directly connected, Serial5/1
B               10.10.100.1/32 [200/1] via 10.200.254.2, 00:31:04
B               10.10.100.3/32 [20/0] via 10.10.4.2, 00:13:29

yakarta#show ip cef vrf cust-one
Prefix                  Next Hop                Interface
0.0.0.0/0               no route
0.0.0.0/32              receive
10.10.2.0/24            10.200.214.1            POS0/1/0
10.10.4.0/24            attached                Serial5/1
10.10.4.0/32            receive
10.10.4.1/32            receive
10.10.4.2/32            attached                Serial5/1
10.10.4.255/32          receive
10.10.100.1/32          10.200.214.1            POS0/1/0
10.10.100.3/32          10.10.4.2               Serial5/1
224.0.0.0/4             drop
224.0.0.0/24            receive
255.255.255.255/32      receive
```

Fuente: elaboración propia, empleando Visio 2013.

La tabla de enrutamiento VRF *cust-one* tiene prefijos que están llenos de protocolos de enrutamiento dinámico y enrutamiento estático, al igual que la tabla de enrutamiento global. El concepto de métricas, distancia, siguiente salto, entre otros. no cambia. Debido a que la instancia VRF está asociada a

interfaces, solo los paquetes IP que ingresan al *router* PE a través de esas interfaces VRF se envían de acuerdo con esa tabla VRF CEF.

En Cisco IOS, CEF es el único método de conmutación compatible para reenviar paquetes IP desde la interfaz VRF. Como tal, CEF debe estar habilitado globalmente en todos los routers PE y todas las interfaces VRF.

3.5.67. *Route distinguisher* (*RD*)

Los prefijos VPN se propagan a través de la red MPLS VPN por *Multiprotocol* BGP (MPBGP). El problema es cuando BGP lleva estos prefijos IPv4 a través de la red del proveedor de servicios, deben ser únicos. Si los clientes tenían direcciones IP superpuestas, el enrutamiento sería incorrecto. Para resolver este problema, el concepto de RD se concibió para hacer que los prefijos IPv4 fueran únicos. La idea básica es que cada prefijo de cada cliente recibe un identificador único (el RD) para distinguir el mismo prefijo de diferentes clientes. Un prefijo derivado de la combinación del prefijo IPv4 y el RD se denomina prefijo vpnv4. MP-BGP necesita llevar estos prefijos vpnv4 entre los routers PE.

Un RD es un campo de 64 bits utilizado para hacer que los prefijos VRF sean únicos cuando MP-BGP los transporta. El RD no indica a que VRF pertenece el prefijo. La función del RD no es la de un identificador de VPN, porque algunos escenarios de VPN más complejos pueden requerir más de un RD por VPN. Cada instancia de VRF en el enrutador PE debe tener un RD asignado. Está valor de 64 *bits* puede tener dos formatos: ASN: nn o dirección IP: nn, donde nn representa un número. El formato más comúnmente usado es ASN: nn, donde ASN representa el número de sistema autónomo. Por lo general, el proveedor del servicio utiliza ASN: nn, donde ASN es el número de

sistema autónomo que la autoridad de números asignados de Internet (IANA) asigna al proveedor del servicio y nn es el número que el proveedor de servicios asigna de forma exclusiva al VRF. El RD no impone semántica; sólo se usa para identificar de manera única las rutas VPN. Esto es necesario porque las rutas IPv4 de un cliente pueden superponerse con las rutas IPv4 de otra. La combinación del RD con el prefijo IPv4 proporciona un prefijo vpnv4, cuya dirección tiene una longitud de 96 *bits*. La máscara tiene 32 *bits* de largo, al igual que para un prefijo IPv4. Si toma un prefijo IPv4 10.1.1.0/24 y un RD 1:1, el prefijo vpnv4 se convierte en 1:1:10.1.1.0/24.

Un cliente puede usar diferentes RD para la misma ruta IPv4. Cuando un sitio VPN está conectado a dos *routers* PE, las rutas desde el sitio VPN pueden obtener dos RD diferentes, dependiendo de en qué router PE se reciban las rutas. Cada ruta IPv4 obtendría dos RD diferentes asignados y tendría dos rutas vpnv4 completamente diferentes. Esto permitiría a BGP verlos como rutas diferentes y aplicar una política diferente a las rutas. En el ejemplo se muestra como configurar el RD en Cisco IOS.

Figura 108. **Configurando un RD**

```
yakarta#conf t
Enter configuration commands, one per line. End with CNTL/Z.
yakarta(config)#ip vrf ?
  WORD VPN Routing/Forwarding instance name
yakarta(config)#ip vrf cust-one
yakarta(config-vrf)#rd ?
  ASN:nn or IP-address:nn VPN Route Distinguisher
yakarta(config-vrf)#rd 1:1
```

Fuente: elaboración propia, empleando Visio 2013.

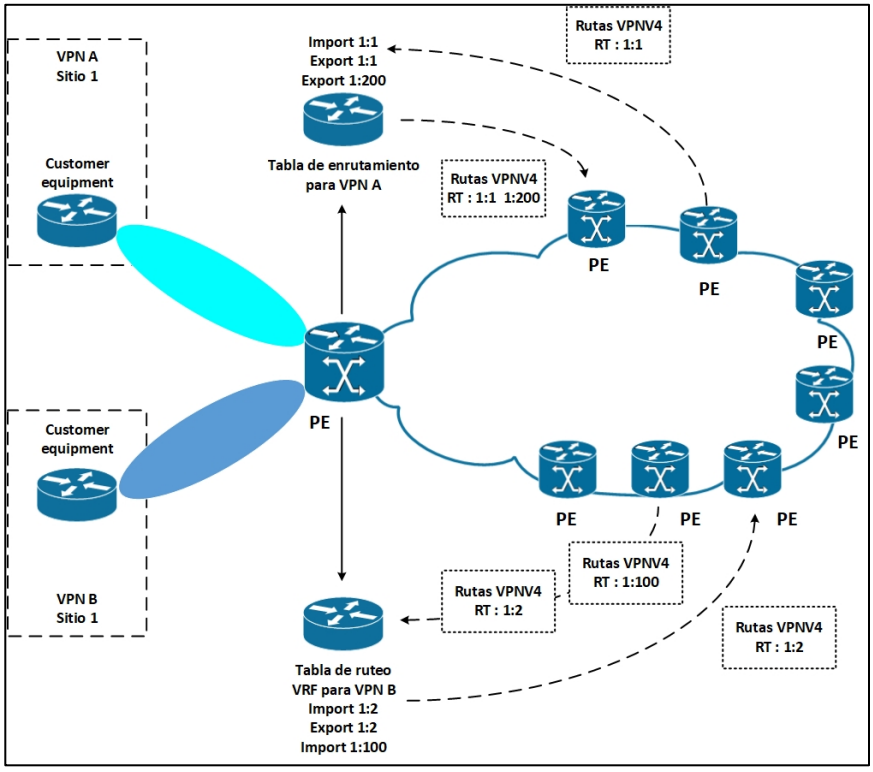
3.5.68. *Route Targets* RTs

Si los RD solo se usarán para indicar la VPN, la comunicación entre los sitios de diferentes VPN sería problemática. Un sitio de la compañía A no podría hablar con un sitio de la compañía B porque los RD no coincidían. El concepto de que los sitios de la compañía A puedan comunicarse con los sitios de la compañía B se denomina VPN extranet. El caso simple de comunicación entre sitios de la misma compañía, la misma VPN, se llama intranet. La comunicación entre los sitios está controlada por otra función MPLS VPN llamada RTs.

Un RT es una comunidad extendida de BGP que indica que rutas se deben importar desde MPBGP al VRF. La exportación de una RT significa que la ruta vpnv4 exportada recibe una comunidad extendida BGP adicional -está es el RT- como se configura bajo *ip vrf* en el *router* PE, cuando la ruta se redistribuye desde la tabla de enrutamiento VRF a MP-BGP. La importación de una RT significa que la ruta vpnv4 recibida desde MP-BGP se verifica para una comunidad extendida coincidente, este es el destino de la ruta, con los que están en la configuración. Si el resultado es una coincidencia, el prefijo se coloca en la tabla de enrutamiento VRF como una ruta IPv4. Si no ocurre una coincidencia, el prefijo es rechazado. El comando para configurar RTs para un VRF es *route-target {import | export | both} route-target-extcommunity*. La palabra clave indica tanto la importación como la exportación.

En la figura 109 se muestra que los RT controlan que las rutas se importan a que VRF de los *routers* PE remotos y con qué RT se exportan las rutas vpnv4 hacia los *routers* PE remotos. Se puede unir más de una RT a la ruta vpnv4. Para que se permita la importación en el VRF, solo se debe coincidir un RT de la ruta vpnv4 con la configuración de los RT importados en la sección *ip vrf del router PE.*

Figura 109. **Ilustración de RTs**



Fuente: elaboración propia, empleando Visio 2013.

En la figura 110 se muestra como configurar los RT en Cisco IOS.

Figura 110. **Configuración de RTs**

```
yakarta#conf t
Enter configuration commands, one per line. End with CNTL/Z.
yakarta(config)#ip vrf cust-one
yakarta(config-vrf)#route-target ?
  ASN:nn or IP-address:nn     Target VPN Extended Community
  both Both                    import and export Target-VPN community
  export                       Export Target-VPN community
  import                       Import Target-VPN community
yakarta(config-vrf)#route-target both 1:1
```

Fuente: elaboración propia, empleando Visio 2013.

El RD y los RT definen entonces el VRF cust-one, como se 'puede ver en la figura 111.

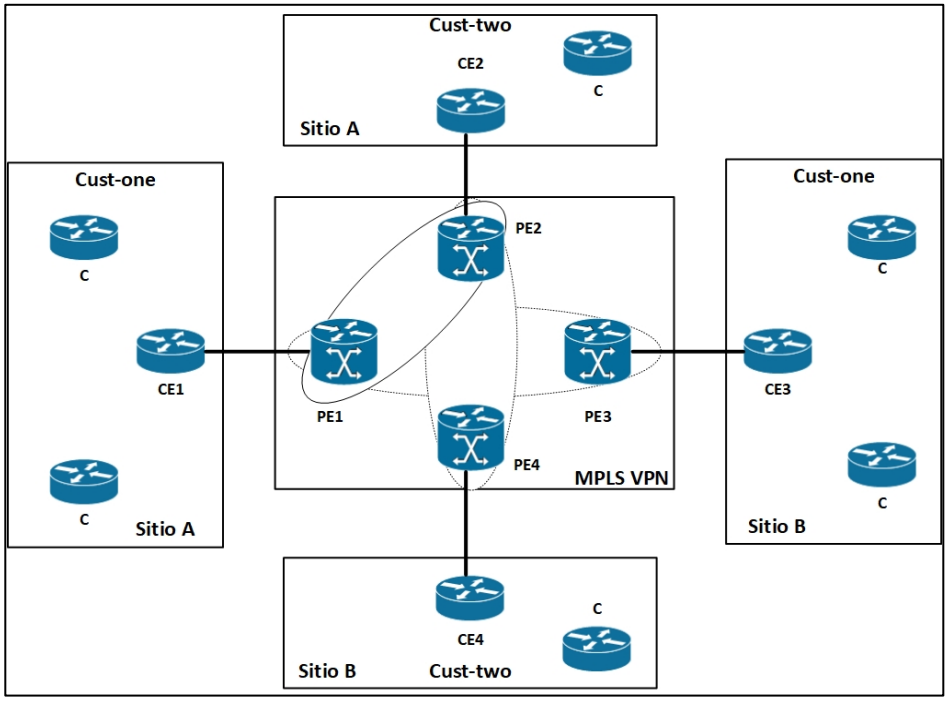
Figura 111. **Configuración VRF**

```
!
ip vrf cust-one
rd 1:1
route-target export 1:1
route-target import 1:1
!
```

Fuente: elaboración propia, empleando Visio 2013.

Al configurar un VRF con varios sitios que pertenecen a una VPN, sin tener que comunicarse con sitios pertenecientes a otra VPN, solo se necesita configurar un RT para importar y exportar en todos los routers PE con un sitio perteneciente a ese VRF. Este es el caso simple de una intranet. Cuando tiene sitios pertenecientes a una VPN que necesitan poder comunicarse con sitios de otra VPN (el caso de extranet), se presta atención a la forma de configurar los RTs correctamente. En la figura 112 se muestra un ejemplo de extranet.

Figura 112. **Ejemplo de una extranet**

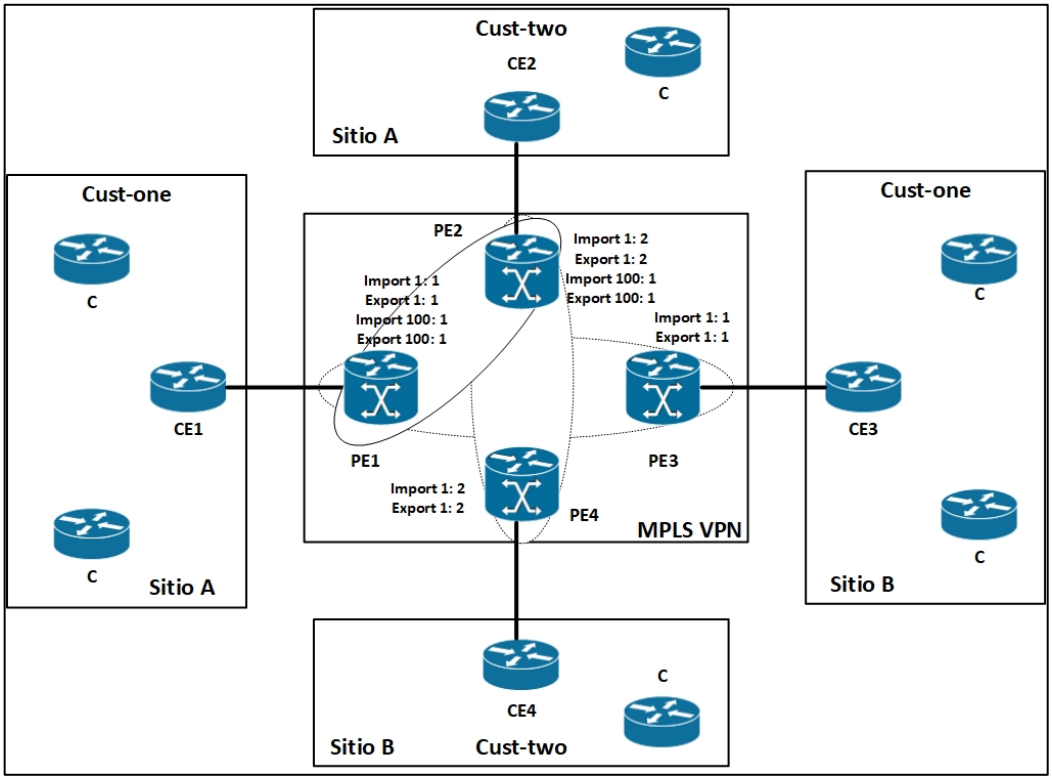


Fuente: elaboración propia, empleando Visio 2013.

Obviamente, los sitios A y B del cliente de VRF deberían poder comunicarse entre ellos. Lo mismo es cierto para los sitios A y B de los dos primeros VRF. El RT que VPN *cust-one* usa es 1: 1. El RT que utiliza VPN cust-two es 1: 2. Ahora imagínese que el sitio A solo de VRF necesita hablar con el sitio A solo de VRF *cust-two*. Esto es perfectamente posible y se determina configurando

Los RTs en consecuencia. El RT 100: 1 se importa y exporta para el sitio A de *vrf cust-one* y *cust-two* en PE1 y PE2 para lograr esto. Esto se llama una extranet. La figura 113 muestra la misma red que en la figura 112, pero con los RT.

Figura 113. **Ejemplo de una extranet con RTs**



Fuente: elaboración propia, empleando Visio 2013.

La figura 114 muestra la configuración necesaria en los *routers* PE.

Figura 114. **Configurando RTs para una extranet**

```
PE1:
!
ip vrf cust-one
rd 1:1
route-target export 1:1
route-target export 100:1
route-target import 1:1
route-target import 100:1
!
PE2:
!
ip vrf cust-two
rd 1:2
route-target export 1:2
route-target export 100:1
route-target import 1:2
route-target import 100:1
!
```

Fuente: elaboración propia, empleando Visio 2013.

En la figura 115 se muestra una ruta 10.10.100.1/32. Es una ruta con RD 1:1 (VRF *cust-one*) que se importa en el VRF *cust-one* y se convierte en una ruta vpnv4 con RD 1:2.

Figura 115. **Ruta extranet**

```
PE1#show ip bgp vpnv4 all 10.10.100.1
BGP routing table entry for 1:1:10.10.100.1/32, version 40
Paths: (1 available, best #1, table cust-one)
    Advertised to update-groups:
       3
    65001
      10.10.2.1 from 10.10.2.1 (192.168.1.1)
        Origin IGP, metric 0, localpref 100, valid, external, best
        Extended Community: RT:1:1 RT:100:1,
        mpls labels in/out 45/nolabel
BGP routing table entry for 1:2:10.10.100.1/32, version 41
Paths: (1 available, best #1, table cust-two)
    Not advertised to any peer
    65001, imported path from 1:1:10.10.100.1/32
      10.10.2.1 from 10.10.2.1 (192.168.1.1)
        Origin IGP, metric 0, localpref 100, valid, external, best
        Extended Community: RT:1:1 RT:100:1
```

Fuente: elaboración propia, empleando Visio 2013.

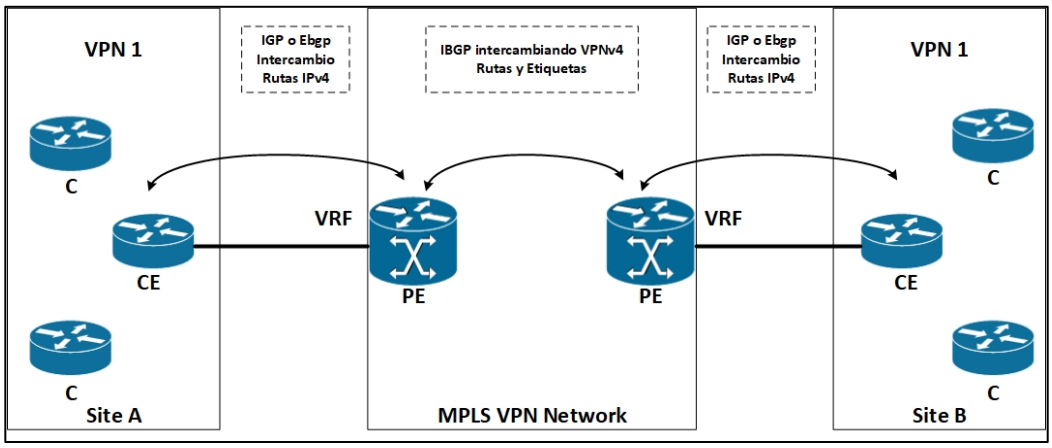
Es posible que no desee que dos VRF intercambien todas las rutas. El número de rutas filtradas de un VRF a otro puede limitarse configurando un mapa de importación o exportación bajo ip vrf, que utiliza un mapa de una ruta para filtrar las rutas adicionales.

3.5.69. Propagación de una ruta VPNv4 en la red MPLS VPN

El VRF separa las rutas de los clientes en los routers PE, pero la siguiente pregunta que se debe plantear es: ¿cómo se transportan los prefijos a través de la red del proveedor de servicios? Debido a que, potencialmente, se podrían transportar numerosas rutas, tal vez cientos de miles, BGP es el candidato ideal porque es un protocolo de enrutamiento probado y estable para llevar a cabo tantas rutas. Simplemente téngase en cuenta que BGP es el protocolo estándar para llevar la tabla completa de enrutamiento de internet. Debido a que las rutas VPN del cliente se vuelvan únicas al agregar el RD a cada ruta IPv4, convirtiéndolas en rutas vpnv4, todas las rutas de los clientes se pueden transportar de manera segura a través de la red MPLS VPN. En la figura se muestra la descripción general de la propagación de la ruta en una red MPLS VPN.

En la figura 116 se muestra una descripción general de la propagación de la ruta en una red MPLS VPN.

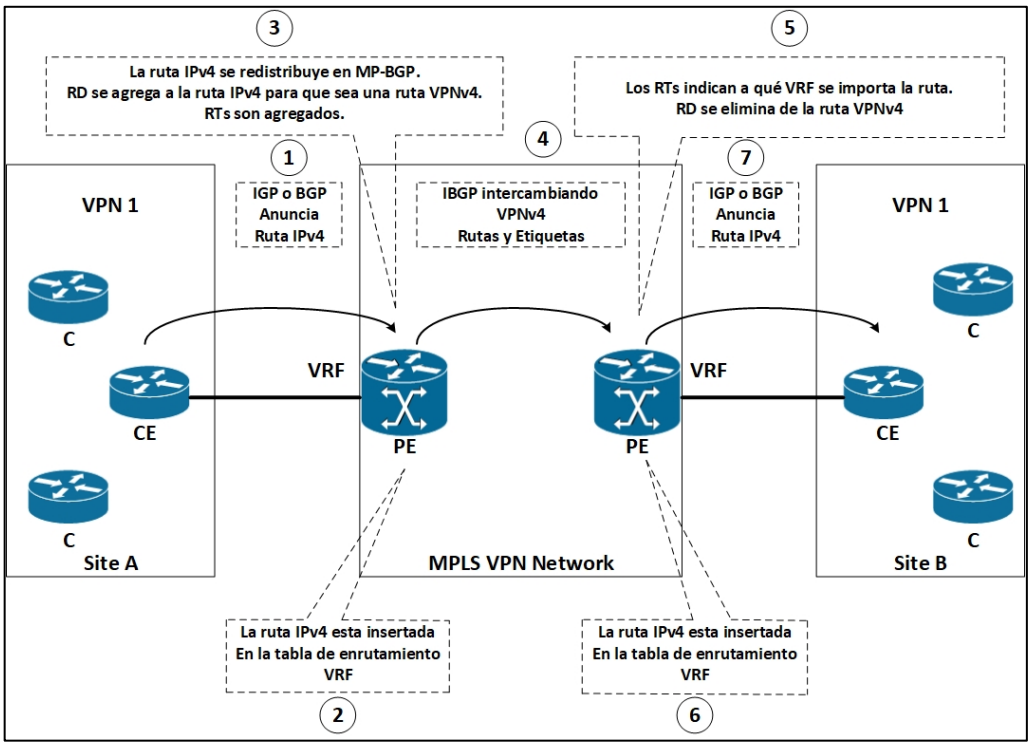
Figura 116. **Propagación de ruta en una red VPN MPLS**



Fuente: elaboración propia, empleando Visio 2013.

El *router* PE recibe rutas IPv4 desde el *router* CE a través de un protocolo de puerta de enlace interior (IGP) o un BGP externo (eBGP), está rutas IPv4 desde el destino VPN se ponen en la tabla de enrutamiento VRF. El VRF que se usa depende del VRF configurado en la interfaz del *router* PE hacia el *router* CE. Estas rutas se anexan con el RD que está asignado a ese VRF. Por lo tanto, se convierten en rutas vpnv4, que luego se ponen en MP-BGP. BGP se encarga de distribuir estas rutas vpnv4 a todos los *routers* PE en la red MPLS VPN. En los *routers* PE, las rutas vpnv4 se eliminan de los RD y se colocan en la tabla de enrutamiento VRF como rutas IPv4. Si la ruta vpnv4, después de quitar el RD, se coloca en el VRF depende si los RT permiten la importación al VRF. Estas rutas IPv4 se anuncian al router CE a través de un IGP o eBGP que se ejecuta entre el router PE y CE. La figura 117 muestra los pasos en la propagación de la ruta de CE a CE a través de la red MPLS VPN.

Figura 117. **Propagación de ruta paso a paso en una red VPN MPLS**



Fuente: elaboración propia, empleando Visio 2013.

Como el proveedor de servicios que ejecuta la red MPLS VPN ejecuta BGP en un sistema autónomo, iBGP se ejecuta entre los *routers* PE.

La propagación de eBGP, que se ejecuta entre el *router* PE y CE, a MP-iBGP en la red MPLS VPN y viceversa es automática y no necesita configuración adicional. Sin embargo, la redistribución de MP-iBGP en el IGP que se ejecuta entre el router PE y CE no es automática. Necesita configurar la redistribución mutua entre MP-iBGP y el IGP.

3.5.70. Reenvío de paquetes en una red MPLS VPN

Como se observó anteriormente, los paquetes no se pueden reenviar como paquetes IP puros entre sitios. Los routers P no pueden reenviarlos porque no tienen la información VRF de cada sitio. MPLS puede resolver este problema de etiquetando los paquetes. Los *routers* P solo deben tener la información de reenvío correcta para que la etiqueta reenvíe los paquetes. La forma más común es configurar el *label distribution protocol* (LDP) entre todos los *routers* P y PE para que todo el tráfico IP cambie de etiqueta entre ellos. También, puede usar RSVP con extensiones para ingeniería de tráfico (TE) al implementar MPLS TE, pero LDP es el más común para MPLS VPN. Los paquetes IP son luego reenviados con etiqueta con una etiqueta desde el *router* PE de ingreso al router PE de salida. Un *router* P nunca tiene que realizar una búsqueda de la dirección IP de destino. Esta es la forma en que se cambian los paquetes entre el router PE de entrada y el *router* PE de salida. Está etiqueta se denomina etiqueta IGP, porque es la etiqueta que está vinculada a un prefijo IPv4 en la tabla de enrutamiento global del *router* P y PE, y el IGP de la red del proveedor del servicio lo anuncia.

¿Cómo sabe el *router* PE de egreso a que VRF pertenece el paquete? Esta información no está en el encabezado IP, y no puede derivarse de la etiqueta IGP, ya que se usa únicamente para reenviar el paquete a través de la red del proveedor del servicio. La solución es agregar otra etiqueta en la pila de etiquetas MPLS. Está etiqueta indica a que VRF pertenece el paquete. Por lo tanto, todos los paquetes de clientes se envían con dos etiquetas: la etiqueta IGP como la etiqueta superior y la etiqueta VPN como la etiqueta inferior. La etiqueta VPN debe ser activada por el router PE de ingreso para indicar al *router* PE de egreso al que pertenece el paquete VRF. ¿Cómo señala el *router* PE de egreso al *router* PE de entrada que etiqueta usar para un prefijo VRF?

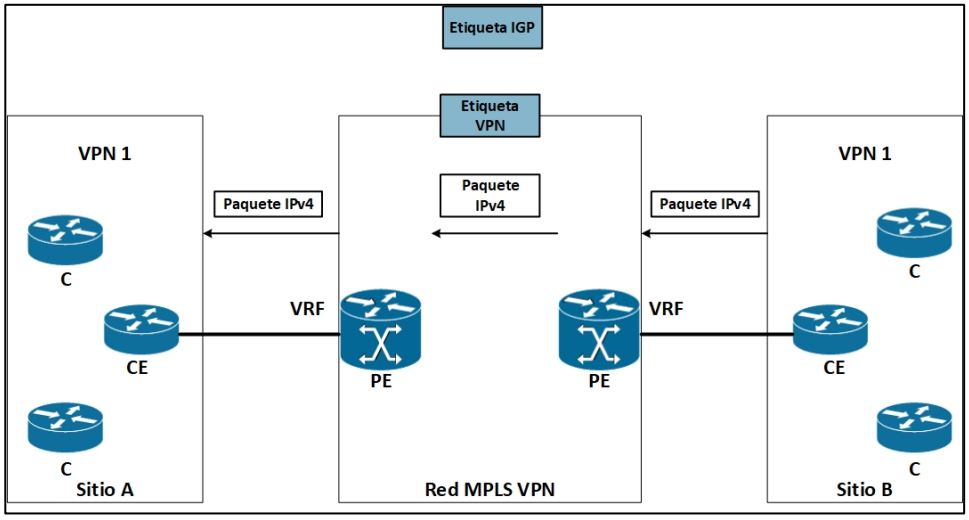
Debido a que MP-BGP ya se usa para publicar el prefijo vpnv4, también señala la etiqueta VPN (también conocida como la etiqueta BGP) que está asociada con el prefijo vpnv4.

En realidad, el concepto de tener una etiqueta de VPN que indique el VRF al que pertenece el paquete no es del todo correcto. Esto podría ser cierto en algunos casos, pero la mayoría de las veces no lo es. Una etiqueta de VPN generalmente indica el próximo salto al que el paquete debe reenviarse en el *router* PE de egreso. Por lo tanto, la mayor parte del tiempo, su propósito es indicar el *router* CE correcto como el siguiente salto del paquete.

Para recapitular, el tráfico de VRF a VRF tiene dos etiquetas en la red MPLS VPN. La etiqueta superior es la etiqueta IGP y se distribuye por LDP o RSVP para TE entre todos los *routers* P y PE salto por salto. La etiqueta inferior es la etiqueta VPN que MP-iBGP anuncia de PE a PE. Los *routers* P usan la etiqueta IGP para reenviar el paquete al *router* PE de salida correcto. El *router* PE de egreso utiliza la etiqueta VPN para reenviar el paquete IP al *router* CE correcto.

La figura 118 muestra el reenvío de paquetes en una red MPLS VPN. El paquete ingresa al *router* PE en la interfaz VRF como el paquete IPv4. Se reenvía a través de la red MPLS VPN con dos etiquetas. Los *routers* P remiten el paquete mirando la etiqueta superior. La etiqueta superior se intercambia en cada *router* P. Las etiquetas se eliminan en el *router* PE de salida y el paquete se reenvía como un paquete IPv4 a la interfaz VRF hacia el router CE. El *router* CE correcto se encuentra observando la etiqueta VPN.

Figura 118. **Reenvío de paquetes en una red VPN MPLS**



Fuente: elaboración propia, empleando Visio 2013.

La sección posterior 'reenvío de paquetes' tiene un ejemplo más detallado de está reenvío de paquetes a través de la red MPLS VPN. En primer lugar, sin embargo, debe conocer más detalles sobre la función de BGP en la red MPLS VPN.

3.5.71. BGP

BGP ha existido por muchos años y es el protocolo estándar para el enrutamiento entre dominios. BGP es el protocolo que hace que Internet funcione tan bien hoy. Los proveedores de servicios que componen Internet y ejecutan BGP entre ellos. Están homologados con otros proveedores de servicios a través de eBGP y ejecutan iBGP en sus propias redes. BGP es un protocolo de enrutamiento que es adecuado para llevar cientos de miles de rutas y tiene un historial comprobado para respaldar esto. BGP es también un protocolo de enrutamiento que permite implementar políticas flexibles y

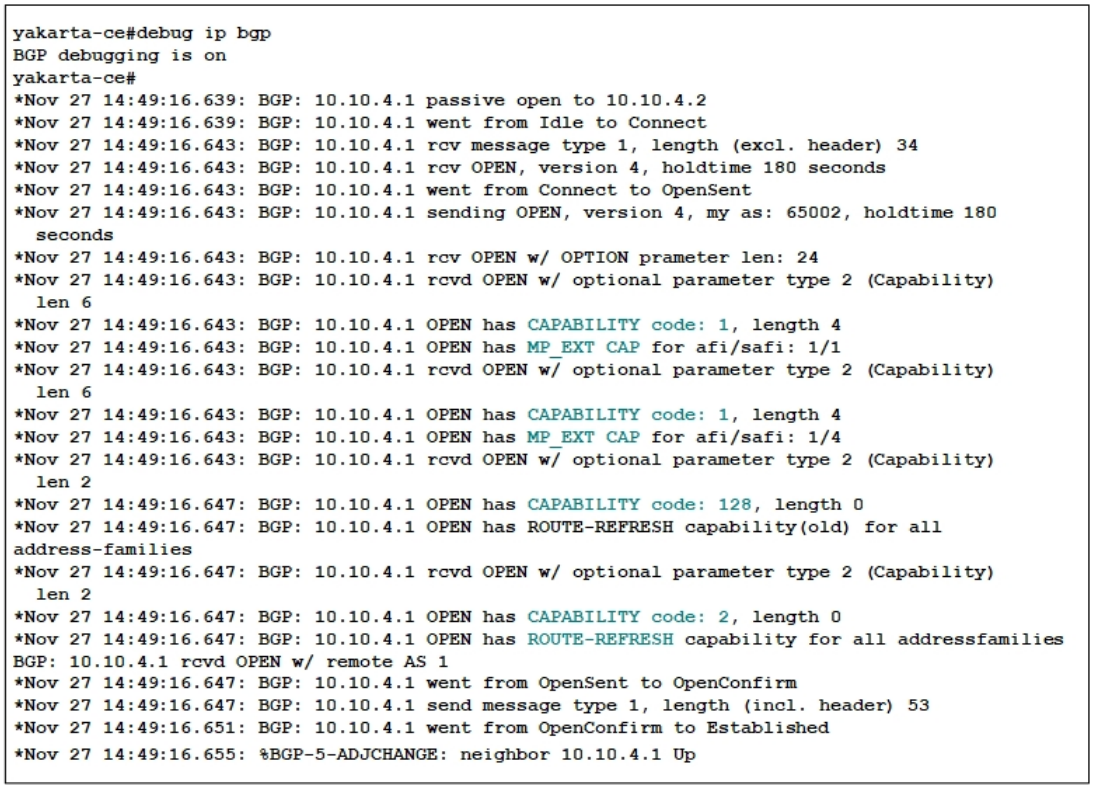
extendidas. Es por eso que un buen candidato para llevar rutas MPLS VPN. Como se mencionó anteriormente, la combinación del RD con el prefijo IPv4 constituye el prefijo vpnv4. Es este prefijo vpnv4 que iBGP necesita transportar entre los routers PE.

3.5.72. Extensiones y capacidades multiprotocolo BGP

BGP-4 se describe en RFC 1771, pero ese RFC describe solo el uso de BGP para llevar los prefijos IPv4. BGP puede hacer mucho más para llevar prefijos IPv4. RFC 2858, 'extensiones multiprotocolo para BGP-4', se escribió para extender BGP como capaz de transportar otra información de enrutamiento que IPv4. Por ejemplo, BGP-4 puede llevar prefijos IPv6 y así proporcionar el enrutamiento entre dominios para IPv6. Cada parlante BGP les permite a sus compañeros saber que las extensiones multiprotocolo para BGP-4 son compatibles con el use de la publicidad de capacidades. Los pares BGP se envían mutuamente las capacidades que admiten. Las capacidades que comparten los compañeros pueden usarse. Algunos ejemplos de capacidades son el filtrado de rutas salientes (ORF), la capacidad de actualización de rutas y las extensiones multiprotocolo. RFC 3392 (publicidad de capacidades con BGP-4) describe el funcionamiento del anuncio de capacidades.

Cuando un anunciante BGP envía un mensaje de apertura a su vecino, puede incluir el parámetro opcional de capacidad, enumerando todas las capacidades de está anunciante BGP. El vecino BGP puede hacer lo mismo. O las capacidades coinciden en ambos pares, o se recibe una notificación BGP de otro altavoz BGP que indica que capacidades no admite. En la figura 119 se muestra un intercambio en capacidades BGP entre dos pares BGP.

Figura 119. **Intercambio de capacidades BGP**



```
yakarta-ce#debug ip bgp
BGP debugging is on
yakarta-ce#
*Nov 27 14:49:16.639: BGP: 10.10.4.1 passive open to 10.10.4.2
*Nov 27 14:49:16.639: BGP: 10.10.4.1 went from Idle to Connect
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcv message type 1, length (excl. header) 34
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcv OPEN, version 4, holdtime 180 seconds
*Nov 27 14:49:16.643: BGP: 10.10.4.1 went from Connect to OpenSent
*Nov 27 14:49:16.643: BGP: 10.10.4.1 sending OPEN, version 4, my as: 65002, holdtime 180
  seconds
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcv OPEN w/ OPTION prameter len: 24
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcvd OPEN w/ optional parameter type 2 (Capability)
  len 6
*Nov 27 14:49:16.643: BGP: 10.10.4.1 OPEN has CAPABILITY code: 1, length 4
*Nov 27 14:49:16.643: BGP: 10.10.4.1 OPEN has MP_EXT CAP for afi/safi: 1/1
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcvd OPEN w/ optional parameter type 2 (Capability)
  len 6
*Nov 27 14:49:16.643: BGP: 10.10.4.1 OPEN has CAPABILITY code: 1, length 4
*Nov 27 14:49:16.643: BGP: 10.10.4.1 OPEN has MP_EXT CAP for afi/safi: 1/4
*Nov 27 14:49:16.643: BGP: 10.10.4.1 rcvd OPEN w/ optional parameter type 2 (Capability)
  len 2
*Nov 27 14:49:16.647: BGP: 10.10.4.1 OPEN has CAPABILITY code: 128, length 0
*Nov 27 14:49:16.647: BGP: 10.10.4.1 OPEN has ROUTE-REFRESH capability(old) for all
address-families
*Nov 27 14:49:16.647: BGP: 10.10.4.1 rcvd OPEN w/ optional parameter type 2 (Capability)
  len 2
*Nov 27 14:49:16.647: BGP: 10.10.4.1 OPEN has CAPABILITY code: 2, length 0
*Nov 27 14:49:16.647: BGP: 10.10.4.1 OPEN has ROUTE-REFRESH capability for all addressfamilies
BGP: 10.10.4.1 rcvd OPEN w/ remote AS 1
*Nov 27 14:49:16.647: BGP: 10.10.4.1 went from OpenSent to OpenConfirm
*Nov 27 14:49:16.647: BGP: 10.10.4.1 send message type 1, length (incl. header) 53
*Nov 27 14:49:16.651: BGP: 10.10.4.1 went from OpenConfirm to Established
*Nov 27 14:49:16.655: %BGP-5-ADJCHANGE: neighbor 10.10.4.1 Up
```

Fuente: elaboración propia, empleando Visio 2013.

En el ejemplo se muestra que puede verificar las capacidades BGP del vecino BGP con el comando *show ip bgp neighbors*.

Las extensiones multiprotocolo para BGP-4 definen dos nuevos atributos BGP: NLRI multiprotocolo accesible y multiprotocolo inalcanzable NLRI. Estos atributos anuncian o retiran rutas. Ambos tienen dos campos: el identificador de familia de direcciones (*address family identifier*, AFI) y el identificador de familia de direcciones subsiguientes (*subsequent address family identifier*, SAFI). Juntos describen exactamente qué tipos de rutas está llevando BGP.

Obsérvese en la figura el formato de está tupla.

Tabla I. **Formato de tupla**

Identificador de familia de dirección (2 octetos)
Identificador de familia de dirección posterior (1 octeto)

Fuente: elaboración propia.

La tabla proporciona algunos de los números de AFI y sus descripciones.

Tabla II. **Números de AFI y sus descripciones**

Número	Descripción
0	Reservada
1	IP (IP versión 4)
2	IP6 (IP versión 6)
11	IPX
12	*Apple talk*

Fuente: elaboración propia.

La tabla enumera los números SAFI y sus descripciones para la familia de direcciones IP.

Tabla III. **Números SAFI y sus descripciones para la familia de direcciones IP**

Número	Descripción
1	NLRI para el envío *unicast*
2	NLRI para el envío *multicast*
3	NLRI para el envío *unicast* y *multicast*
4	NLRI para IPv4 y reenvío de etiquetas
128	NLRI para el reenvío de etiquetado VPN

Fuente: elaboración propia.

Para soportar el comportamiento multiprotocolo de BGP en Cisco IOS, el proceso de enrutamiento BGP tiene el concepto de familias de direcciones. Las cuatro familias de direcciones actualmente admitidas son IPv4, IPv6, vpnv4 (VPN-IPv4) y vpnv6 (VPN-IPv6). Las siguientes familias de direcciones que pueden especificar son *unicast*, *multicast* y VRF. La figura 120 muestra la configuración de familias de direcciones BGP.

Figura 120. **Configurando las familias de direcciones BGP**

```
yakarta#conf t
Enter configuration commands, one per line. End with CNTL/Z.
yakarta(config)#router bgp 1
yakarta(config-router)#address-family ?
  ipv4 Address family
  ipv6 Address family
  vpnv4 Address family
  vpnv6 Address family
yakarta(config-router)#address-family ipv4 ?
  multicast Address Family modifier
  unicast Address Family modifier
  vrf Specify parameters for a VPN Routing/Forwarding instance
  <cr>
```

Fuente: elaboración propia, empleando Visio 2013.

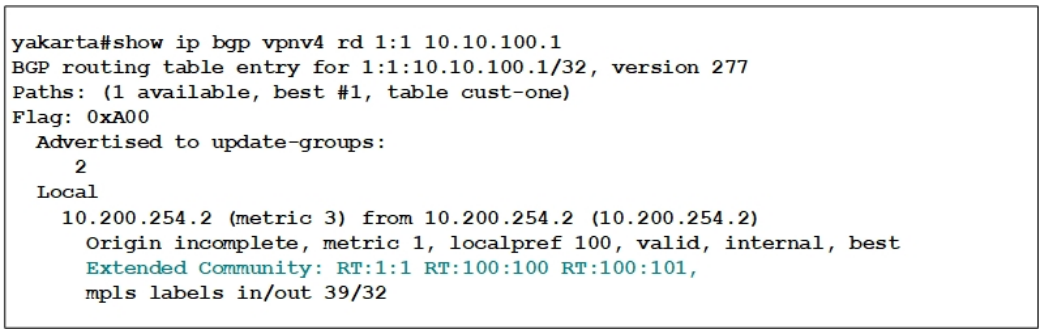
Utiliza la familia de direcciones vpnv4 en el proceso bgp del *router* para configurar las sesiones BGP vpnv4 y los parámetros que necesitan los *routers* PE.

Utiliza la dirección de la familia ipv4 vrf vrf-0 name en el proceso bgp del *router* en los *routers* PE para configurar las sesiones BGP y los parámetros hacia los routers CE, a través de las interfaces VRF.

3.5.73. Comunidad extendida BGP: RT

El borrador de *ietf-idr-bgp-ext-communities* define el atributo de comunidad extendida. El atributo de comunidad es un atributo transitivo opcional que se describe en el RFC 1997. La comunidad extendida es también un atributo transitivo opcional BGP. Surgió para extender el rango de comunidades y tiene una estructura mejorada sobre el atributo de la comunidad BGP. Se definen varios atributos de comunidades extendidas BGP, pero solo se necesita uno para VPN MPLS: la comunidad RT extendida. Indica a los anunciantes BGP (los routers PE) si la ruta debe importarse a un VRF. En la figura 121 se muestra la ruta vpnv4 1:1: 10.10.100.1/32 con los RTs 1: 1, 100: 100: 101. Solo los VRF que están configurados para importar al menos uno de estos RT insertan la ruta de IPv4 10.10.100.1/32 en la tabla de enrutamiento de VRF.

Figura 121. **Atributo BGP RT**



Fuente: elaboración propia, empleando Visio 2013.

3.5.74. Rutas VPNv4

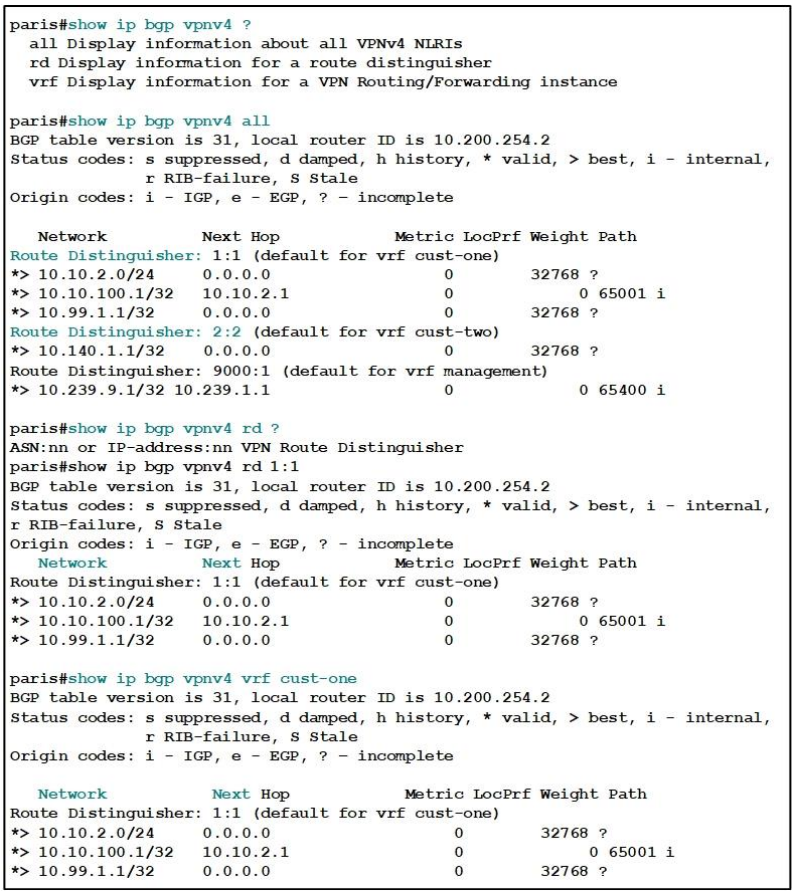
El campo de 64 *bits* del RD y el prefijo IPv4 de 32 *bits* conforman el prefijo vpnv4, que tiene 96 *bits* de longitud. MP-iBGP anuncia estos prefijos entre los enrutadores PE. Puede ver los prefijos vpnv4 que BGP lleva con el siguiente comando:

show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [rib-failure] [ip-prefix/length [longer-prefixes] [output-modifiers]] [network-address [mask] [longer-prefixes] [labels]]

La palabra clave all para este comando muestra todas las rutas vpnv4, o todas las rutas para todos los RD. Con la palabra clave rd, puede ver sólo las rutas vpnv4 con ese cierto RD. Puede hacer lo mismo usando la palabra clave vrf en un router PE. Sin embargo, si usa el comando con la palabra clave vrf en un reflector de ruta (RR), es posible que no le muestre las rutas. Es posible que RR no tenga VRF configurados, ya que probablemente solo se use para reflejar las rutas vpnv4. En ese caso, debe usar el comando con la palabra clave rd

para ver rutas vpnv4 específicas. La figura 122 se muestra el comando *show ip bgp vpnv4* con las palabras clave.

Figura 122. **Rutas VPNv4**



```
paris#show ip bgp vpnv4 ?
  all Display information about all VPNv4 NLRIs
  rd Display information for a route distinguisher
  vrf Display information for a VPN Routing/Forwarding instance

paris#show ip bgp vpnv4 all
BGP table version is 31, local router ID is 10.200.254.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust-one)
*> 10.10.2.0/24     0.0.0.0                  0         32768 ?
*> 10.10.100.1/32   10.10.2.1                0             0 65001 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 2:2 (default for vrf cust-two)
*> 10.140.1.1/32    0.0.0.0                  0         32768 ?
Route Distinguisher: 9000:1 (default for vrf management)
*> 10.239.9.1/32 10.239.1.1                  0             0 65400 i

paris#show ip bgp vpnv4 rd ?
ASN:nn or IP-address:nn VPN Route Distinguisher
paris#show ip bgp vpnv4 rd 1:1
BGP table version is 31, local router ID is 10.200.254.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust-one)
*> 10.10.2.0/24     0.0.0.0                  0         32768 ?
*> 10.10.100.1/32   10.10.2.1                0             0 65001 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?

paris#show ip bgp vpnv4 vrf cust-one
BGP table version is 31, local router ID is 10.200.254.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust-one)
*> 10.10.2.0/24     0.0.0.0                  0         32768 ?
*> 10.10.100.1/32   10.10.2.1                0             0 65001 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
```

Fuente: elaboración propia, empleando Visio 2013.

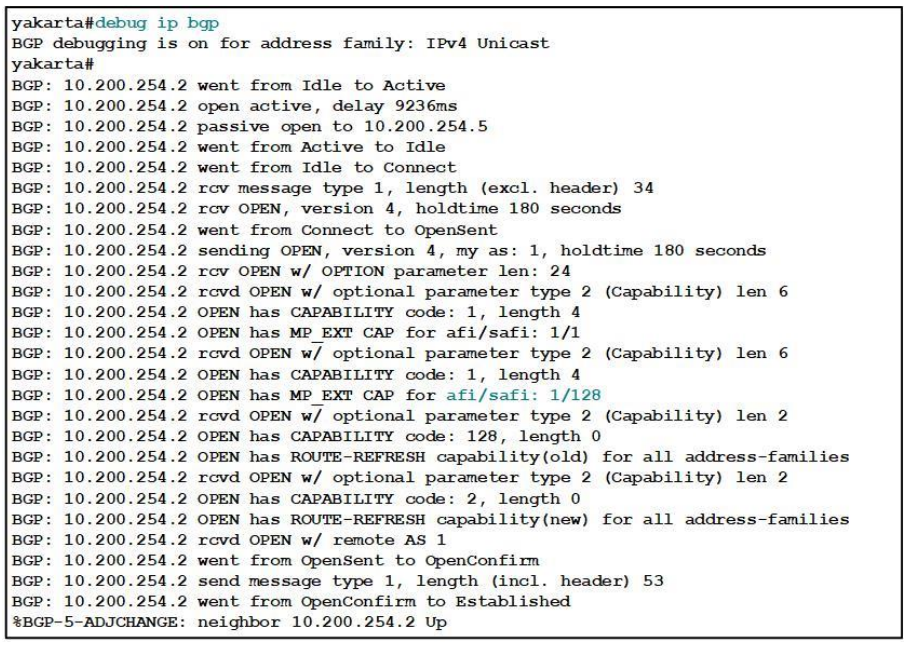
3.5.75. BGP llevando la etiqueta

BGP anuncia los prefijos vpnv4 en la red MPLS VPN. Esto no es suficiente para poder reenviar correctamente el tráfico VPN. Para que el *router* PE egreso pueda reenviar el tráfico VPN correctamente al router CE, debe reenviar el

paquete según la etiqueta. El *router* PE de salida puede asignar una etiqueta de este tipo al prefijo vpnv4, que se denomina etiqueta VPN. El *router* PE de egreso debe anunciar la etiqueta junto con el prefijo vpnv4 a los posibles routers PE de ingreso. La codificación de la etiqueta con el prefijo se describe en RFC 3107, 'llevando la información de la etiqueta en BGP-4.' La etiqueta simplemente se incorpora junto con el prefijo vpnv4 y se anuncia mediante BGP usando el atributo de extensiones multiprotocolo. La etiqueta está contenida en el campo NLRI. El AFI es 1 y el SAFI 128 en el caso de MPLS VPN para IPv4.

Es posible ver que el *router* PE es capaz de anunciar etiquetas para los prefijos vpnv4 cuando se intercambian las capacidades BGP, como se observa en la figura 123.

Figura 123. **Capacidad de anuncio de etiqueta BGP**



Fuente: elaboración propia, empleando Visio 2013.

Obsérvese la figura para ver la codificación del campo NLRI para MPLS VPN. Obsérvese que la etiqueta está codificada como tres octetos, no cuatro. Los tres octetos contienen los 20 *bits* del valor de etiqueta codificado en los bits de orden superior y el *bit* de fondo de pila como el *bit* de orden inferior. Puede existir más de una etiqueta, cada una codificada como tres octetos. Sin embargo, para MPLS VPN descrito aquí, MP-BGP anuncia solo una etiqueta para cada prefijo vpnv4.

Tabla IV. **Codificación del campo NLRI para MPLS VPN**

Longitud (1 octeto)
Etiqueta (3 octetos)
Prefijo (variable)

Fuente: elaboración propia.

Un anunciador BGP solo asigna una etiqueta a un prefijo para el que es siguiente salto. Está es una regla importante para recordar cuando se observa el comportamiento de un BGP RR para las rutas vpnv4. La figura 124 muestra la configuración de la familia de direcciones vpnv4. Primero, necesita definir el vecino BGP en la parte global de la configuración BGP. Luego debe activar el vecino BGP en la familia de direcciones vpnv4 especificando la palabra clave activate.

Solo las comunidades extendidas BGP se envían por defecto al vecino vpnv4. Si desea usar comunidades estándar, también, especifique *send-community* para el vecino BGP.

Figura 124. **Configuración de VPNv4 de la familia de direcciones BGP**

```
router bgp 1
bgp log-neighbor-changes
neighbor 10.200.254.2 remote-as 1
neighbor 10.200.254.2 update-source Loopback0
!
address-family ipv4
redistribute rip
neighbor 10.200.254.2 activate
exit-address-family
!
address-family vpnv4
neighbor 10.200.254.2 activate
neighbor 10.200.254.2 send-community both
exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

Debug ip bgp vpnv4 unicast updates habilita la depuración de las actualizaciones de vpnv4 en BGP. La figura 125 muestra esta depuración cuando se recibe un prefijo vpnv4.

Figura 125. **Actualizaciones Unicast para el comando *debug* ip bgp vpnv4**

```
yakarta#debug ip bgp vpnv4 unicast updates
BGP updates debugging is on for address family: VPNv4 Unicast
yakarta#
BGP(2): 10.200.254.2 rcvd UPDATE w/ attr: nexthop 10.200.254.2, origin ?, localpref 100,
metric
1, extended community RT:1:1
BGP(2): 10.200.254.2 rcvd 1:1:10.10.100.1/32
BGP(2): Revise route installing 1 of 1 routes for 10.10.100.1/32 -> 10.200.254.2(main) to
custone
IP table
```

Fuente: elaboración propia, empleando Visio 2013.

Puede ver las etiquetas que BGP recibe y anuncia para las rutas vpnv4 en el ejemplo. La etiqueta de entrada se utiliza como una etiqueta entrante en la base de información de reenvío de etiquetas (LFIB) para está prefijo vpnv4. Es la etiqueta que se anuncia a los otros enrutadores PE es la que se adjunta al prefijo vpnv4. Es la etiqueta de VPN que utiliza está PE al reenviar tráfico a través de la red MPLS VPN. Para los prefijos con 0.0.0.0 como el siguiente salto en la figura 126, no se ha recibido ninguna etiqueta saliente. Esto se debe a que los prefijos se aprenden de las interfaces VRF y los paquetes se deben reenviar sin etiquetar hacia el router CE.

Figura 126. **Publicidad BGP y etiquetas MPLS**

```
yakarta#show ip bgp vpnv4 rd 1:1 labels
   Network          Next Hop        In label/Out label
Route Distinguisher: 1:1 (cust-one)
   10.10.2.0/24     10.200.254.2    29/36
   10.10.4.0/24     0.0.0.0         26/nolabel
   10.10.4.2/32     0.0.0.0         37/nolabel
   10.10.100.1/32   10.200.254.2    32/35
   10.10.100.3/32   10.10.4.2       38/exp-null
   10.88.1.1/32     10.200.254.2    34/34
   10.99.1.1/32     10.200.254.2    28/33
   10.99.1.2/32     0.0.0.0         27/nolabel
   10.200.200.1/32  10.200.254.2    30/32
```

Fuente: elaboración propia, empleando Visio 2013.

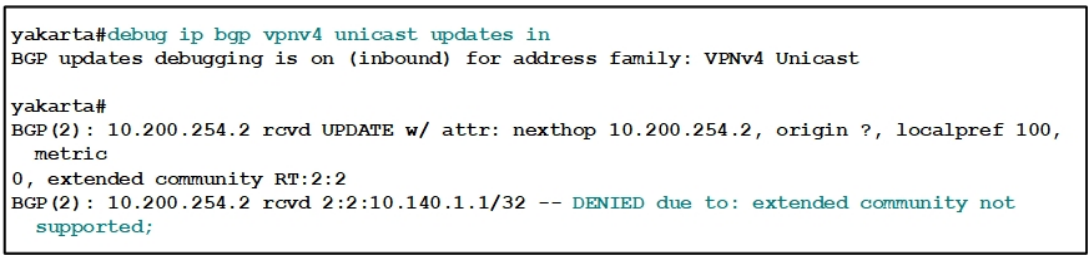
3.5.76. RR'S

Un RR es un anunciador BGP que refleja las rutas de otros altavoces BGP. Los RR se inventaron cuando las redes crecieron. El BGP interno requiere que todos los anunciantes BGP estén en una malla completa entre si. Esto está bien para un bajo número de anunciantes BGP, pero le da al operador problemas de red cuando la red se vuelve más grande que un cierto tamaño.

Cuando tiene una red con n anunciadores BGP internos, cada altavoz BGP tiene n-1 pares, y n*(n-1)/2 sesiones BGP en total. las confederaciones RR y BGP se inventaron para aliviar este problema. Los RR coinciden con los anunciantes de BGP en un cluster, pero los anunciantes de BGP en el cluster ya no se necesitan mirar más si miran con los RR. Los RR solo se adelantan o reflejan todas las rutas BGP que reciben. Si desea usar RR con MPLS VPN, los RR deben reflejar prefijos vpnv4, que llevan etiquetas. Los RR solo cambian la etiqueta si se convierten en el siguiente salto para las rutas, que generalmente no lo hacen. Los RR que se convierten en el próximo salto para la ruta iBGP están en la ruta de reenvío. esto significa que tiene que reenviar el tráfico para esas rutas. Esto podría generar una gran cantidad de tráfico que fluye a través de algunos RR, lo que definitivamente no es una buena idea. Los RR no deberían cargarse con el tráfico de reenvío. Además, la ruta del tráfico a través de los RR desde la entrada PE a la salida de PE generalmente no es la ruta más óptima, porque los RR pueden estar en cualquier parte de la red. Los RR no deben reenviar tráfico, sólo reflejan las rutas BGP.

Los RR se diferencian de otra manera de los otros anunciantes BGP (los routers PE) en la red MPLS VPN. No rechazan las rutas vpnv4 cuando el RT no está configurado para aceptación en los RR. Un *router* PE que recibe una ruta vpnv4 para la cual cualquiera de los RT no se importa a un VRF rechaza la ruta. En la figura 127 muestra una ruta que se rechaza porque el RT 2: 2 no está configurado para ser importado por un VRF en el PE.

Figura 127. **Ruta vpnv4 rechazada**



Fuente: elaboración propia, empleando Visio 2013.

El router PE tiene está comportamiento predeterminado para guardar memoria. ¿Por qué necesita almacenar las rutas vpnv4 en la tabla BGP si no hay VRF conectado a este router PE que importa la ruta? Los RR no muestran este comportamiento porque no saben qué RT's permiten o niegan los PE. Los RR aceptan y almacenan todas las rutas BGP. Para aliviar la carga, puede implementar grupos de RR para dividir la carga de las rutas vpnv4 que reflejan entre varios RR o grupos de RR. Cada RR o grupo de RR refleja un subconjunto de todas las rutas vpnv4.

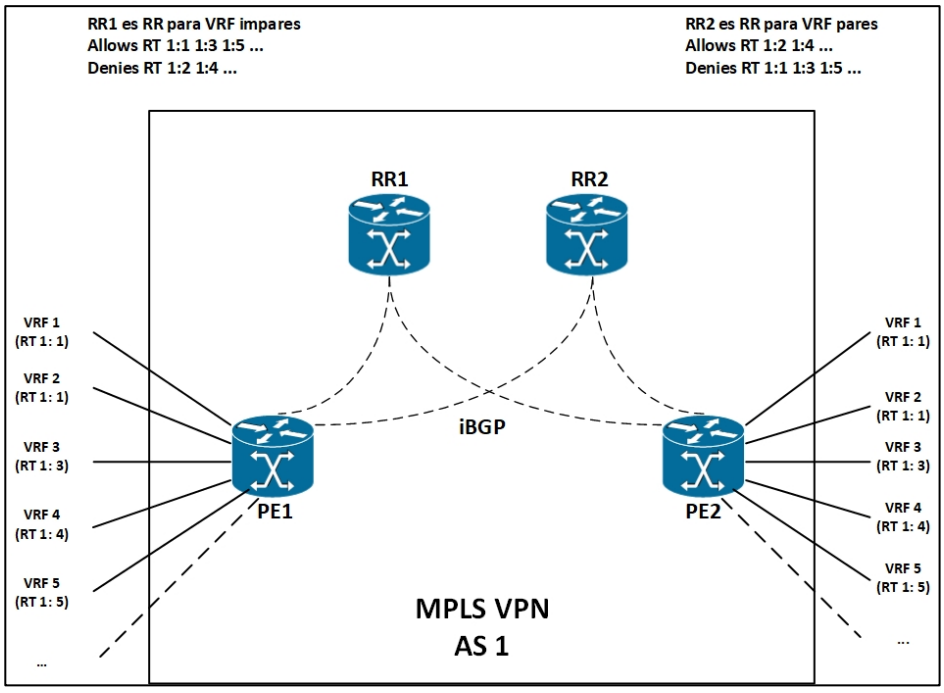
Es una buena idea tener al menos dos RR para un subconjunto de los prefijos vpnv4 por razones de redundancia.

3.5.77. GRUPO RR

No es necesario que un RR o un grupo de RR tenga todas las rutas vpnv4 en la tabla BGP. Puede subdividir las rutas vpnv4 en grupos y permitir que varios RR o varios grupos de RR lleven uno de esos subconjuntos de rutas. Divide y conquistaras. Esto aumenta la escalabilidad de su red. El comando necesario para implementar el grupo RR en los RR es *bgp rr-group {extcom-listnumber}* en la familia de direcciones vpnv4. Debe especificar una lista de

comunidad extendida para el grupo RR. Esta lista extendida de la comunidad específica los RT que desea que está RR permitida o niegue. En la figura 128 se muestra un ejemplo del uso de un grupo RR. Hay dos RR disponibles, con el grupo RR configurado para un conjunto diferente de rutas vpnv4. Un RR filtra las actualizaciones vpnv4 con los pares RTs pares; el otro RR filtra las actualizaciones de vpnv4 con los RTs impares.

Figura 128. **Ejemplo de una red MPLS VPN con grupos RR**



Fuente: elaboración propia, empleando Visio 2013.

La figura 129 muestra el uso de grupos RR. Se crea una lista de comunidad extendida 1 en RR1 que permite la RT 1: 1 y se niega la RT 1: 2 a los clientes de RR. En el segundo RR, intercambia los RT permitidos y denegados.

Figura 129. **Ejemplo de grupos RR**

```
RR1(config-router)#address-family vpnv4
RR1(config-router-af)#bgp rr-group ?
<1-500> Extended-Community list number
!
router bgp 1
neighbor 10.200.254.2 remote-as 1
neighbor 10.200.254.2 update-source Loopback0
neighbor 10.200.254.5 remote-as 1
neighbor 10.200.254.5 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 10.200.254.2 activate
neighbor 10.200.254.2 route-reflector-client
neighbor 10.200.254.2 send-community extended
neighbor 10.200.254.5 activate
neighbor 10.200.254.5 route-reflector-client
neighbor 10.200.254.5 send-community extended
bgp rr-group 1
exit-address-family
!
ip extcommunity-list 1 permit rt 1:1
ip extcommunity-list 1 deny rt 1:2
ip extcommunity-list 1 permit rt 1:3
ip extcommunity-list 1 deny rt 1:4
…
!
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.78. Ruta de selección BGP

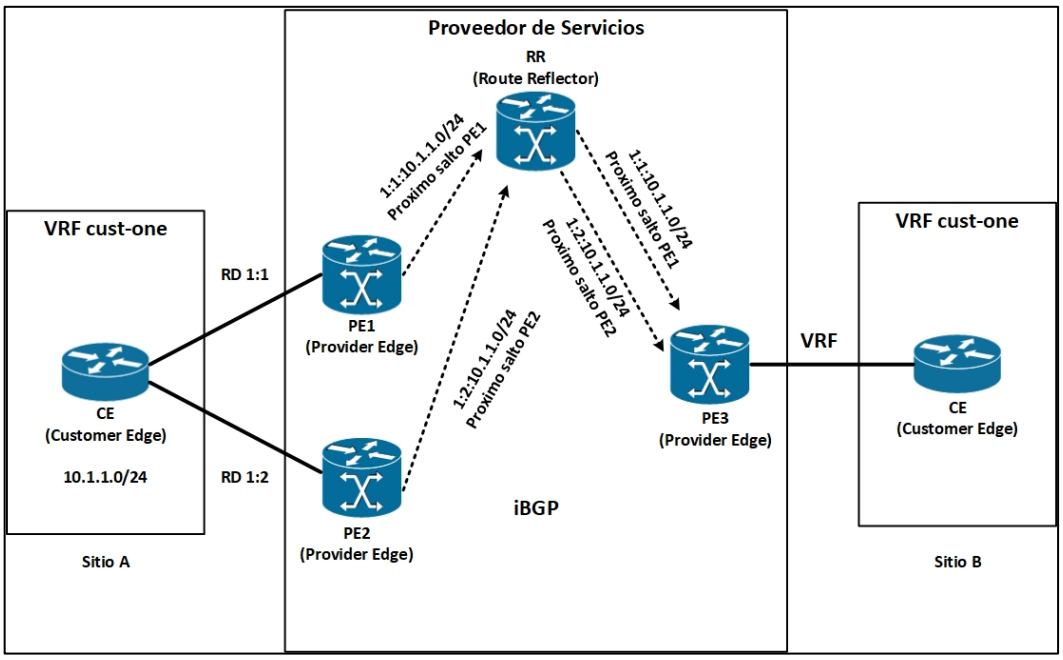
Distintos anunciadores de BGP pueden publicitar la ruta vpnv4 cuando, por ejemplo, un sitio de un cliente que está conectado a dos *routers* PE. El anunciador BGP receptor debe elegir una ruta BGP como la mejor. El proceso para seleccionar la mejor ruta vpnv4 es el mismo que para las rutas regulares BGP IPv4. La única diferencia es que ahora las rutas BGP nos son prefijos IPv4 de 32 bits, sino 96 bits en prefijos vpnv4. Por lo tanto, si el sitio de un cliente

está conectado a dos *routers* de PE, el *router* PE de ingreso recibe la ruta vpnv4 con dos saltos de BGP diferentes, es decir, los dos *routers* PE de egreso. El *router* de PE de ingreso aplica el mejor proceso de selección de una ruta BGP e instala una de las dos rutas BGP en la tabla de enrutamiento VRF.

3.5.79. Usando múltiples RD

Cuando un router CE tiene doble conexión a dos *routers* PE y los RR se utilizan para las rutas vpnv4, tendrá un problema al intentar utilizar BGP multipath en los routers PE. BGP *multipath* debe instalar múltiples rutas para el mismo destino en la tabla de enrutamiento. Sin embargo, cuando se utilizan RR, utilizan el mejor proceso de selección de una ruta BGP para elegir la mejor ruta vpnv4. Los RR anuncian o reflejan solo el mejor camino. El *router* PE de ingreso solo obtiene una ruta de instalación en lugar de dos. Obsérvese la figura 130. El prefijo 10.1.1.0/24 se anuncia desde dos routers PE. En cada *router* PE, el mismo RD 1: 1 está conectado al prefijo IP. El RR recibe dos anuncios de prefijo BGP para está prefijo vpnv4. El RR elige la mejor ruta como la que tiene PE1 como el próximo salto. Esta es la única ruta anunciada desde el RR a sus clientes (los *routers* PE).

Figura 130. **RR anuncia solo la mejor ruta BGP**



Fuente: elaboración propia, empleando Visio 2013.

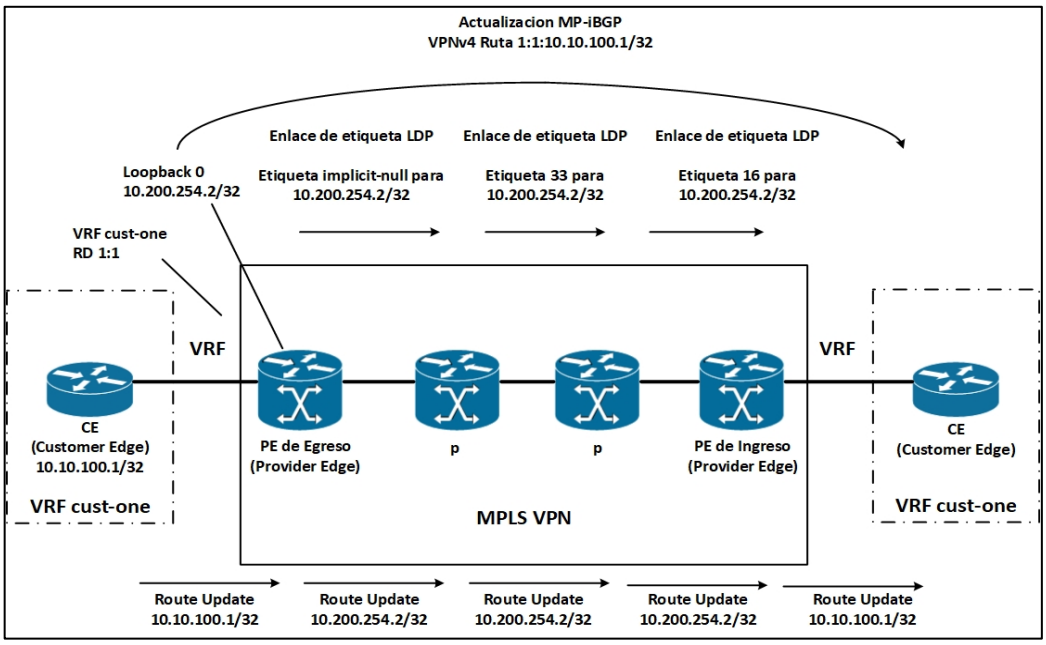
Una solución simple para este problema es hacer que los prefijos vpnv4 anunciados por ambos *routers* PE de egreso sean diferentes. Si las dos rutas son diferentes, los RR anunciarán ambas. Las rutas vpnv4 pueden hacerse diferentes entre sí asignando diferentes RD a los VRF en ambos *routers* PE de egreso. Al tener un RD diferente bajo la configuración VRF en los *routers* PE de egreso. Al tener un RD diferente bajo la configuración VRF en los *routers* PE de egreso, las rutas se vuelven únicas. En ese caso, los RR anuncian ambas rutas, y los *routers* PE de ingreso pueden instalar ambas rutas en la tabla de enrutamiento VRF si se usa BGP *multipath*. Observe la figura para el mismo ejemplo que en la figura, pero con dos RD usados para la misma ruta IPv4 desde el VRF en los routers PE de egreso. El resultado es que el RR puede

reenviar dos rutas vpnv4 a los routers PE. PE3 puede instalar ambas rutas si tiene configurado iBGP *multipath*.

3.5.80. Reenvío de paquetes

Está sección, ilustrada con un ejemplo específico, analiza la vida de un paquete IP a medida que atraviesa la red troncal MPLS VPN de un sitio de cliente a otro. Los bloques de construcción básicos de MPLS VPN deben estar en primer lugar. El iBGP multiprotocolo debe ejecutarse entre los *routers* PE y P. En la siguiente figura 131 supone que el protocolo de distribución de etiquetas es LDP. Entre los *routers* PE y CE, se debe ejecutar un protocolo de enrutamiento y colocar las rutas del cliente en la tabla de enrutamiento VRF en los *routers* PE. Finalmente, esas rutas deben distribuirse en MP-iBGP y viceversa. Ver las figuras para una comprensión. En la figura se muestra el anuncio de ruta de la ruta vpnv4 y la etiqueta desde la salida PE a la entrada PE y el anuncio de la ruta IGP que representa el siguiente salto BGP del PE de salida y la etiqueta al PE entrante. La dirección BGP del siguiente salto en la salida PE es 10.200.254.2/32, que un IGP anuncia a la entrada PE. La etiqueta de esa ruta IGP es 10.200.254.2/32, que un IGP anuncia a la entrada PE. La etiqueta de esa ruta IGP se anuncia salto por salto por LDP. La ruta del cliente IPv4 10.10.100.1/32 se anuncia mediante un protocolo de enrutamiento PE-CE desde CE al PE de salida. El PE de egreso agrega el RD 1: 1, lo convierte en la ruta vpnv4 1: 1 10.10.100.1/32, y lo envía al PE de ingreso con la etiqueta 30, a través de multiprotocolo iBGP.

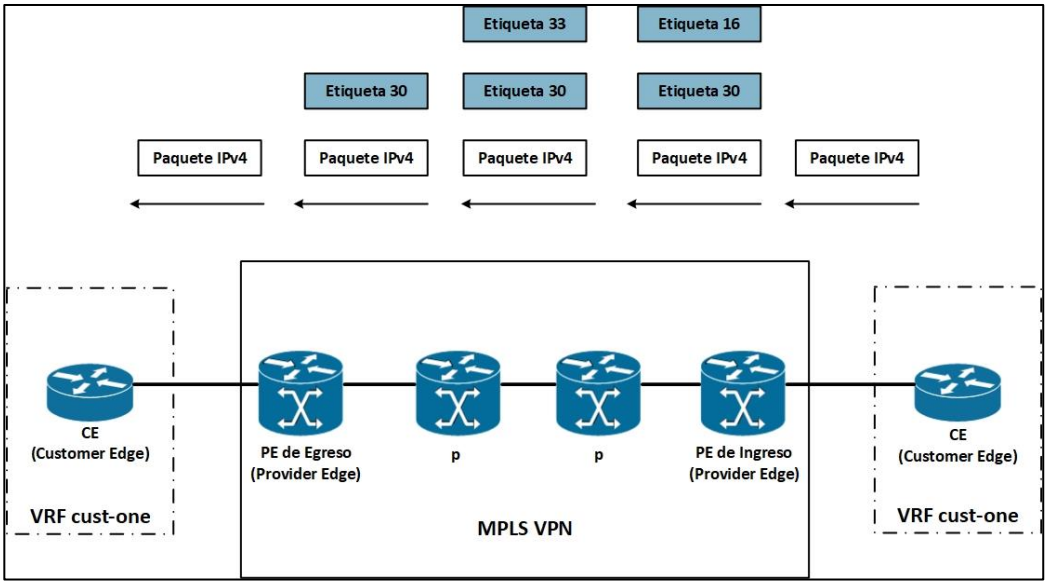
Figura 131. **Uso de multiples RD's**



Fuente: elaboración propia, empleando Visio 2013.

En la figura 132 se muestra un paquete con la dirección IP de destino 10.10.100.1 siendo reenviada con las dos etiquetas como se anuncia en la figura 133.

Figura 132. **Vida de un paquete IPv4 a través de una red *backbone* VPN de MPLS: reenvio de paquetes**



Fuente: elaboración propia, empleando Visio 2013.

Cuando un paquete IP ingresa al *router* PE de ingreso desde el CE, el *router* PE de ingreso busca la dirección IP de destino en la tabla CEF de VRF. El router PE de ingreso encuentra el VRF correcto al observar en qué interfaz entró el paquete al router PE y con qué tabla VRF está asociada está interfaz. La entrada específica en la tabla VRF CEF generalmente indica que se deben agregar dos etiquetas.

Cuando los routers PE de entrada y salida están conectados directamente, los paquetes solo tendrán una etiqueta: la etiqueta VPN. Esto es cierto debido al salto del penúltimo salto (*Penultimate hop popping,* PHP).

En primer lugar, el router PE de ingreso empuja la etiqueta VPN 30, según lo anunciado por BGP para la ruta vpnv4. Esto se convierte en la etiqueta

inferior. Luego, el *router* PE de ingreso empuja la etiqueta IGP como la etiqueta superior. Esta etiqueta es la etiqueta que está asociada con la ruta / 32 IGP para la dirección IP del siguiente salto de BGP. Está suele ser la dirección IP de interfaz de bucle invertido en la salida PE. Está etiqueta se anuncia salto por salto entre los *routers* P hasta que llega al *router* PE de ingreso. Cada salto cambia el valor de la etiqueta. La etiqueta IGP que empuja el ingreso PE es la etiqueta 16.

El paquete IPv4 sale del router PE de ingreso con dos etiquetas encima. La etiqueta superior, la etiqueta IGP para el router PE de egreso, se intercambia en cada salto en la ruta. Esta etiqueta obtiene el paquete VPN IPv4 al router PE de egreso correcto. Normalmente, debido a que es el comportamiento predeterminado en Cisco IOS, el comportamiento de PHP (*Penultimate hop popping*) tiene lugar entre la última P y el router PE de egreso. Por lo tanto, la etiqueta IGP se abre en el último *router* P y el router PE de egreso. Por lo tanto, la etiqueta IGP se abre en el último *router* P y el paquete ingresa al *router* PE de salida con solo la etiqueta VPN en la pila de etiquetas. El *router* PE de salida busca esta etiqueta VPN en el LFIB y toma una decisión de reenvío. Como la etiqueta saliente es sin etiqueta, la pila de etiquetas restante se elimina y el paquete se reenvía como un paquete IP al *router* CE. El *router* PE de salida no tiene que realizar una búsqueda IP de dirección IP de destino en el encabezado IP si la etiqueta saliente es sin etiqueta. La información correcta del siguiente salto se encuentra buscando la etiqueta VPN en el LFIB. Solo cuando la etiqueta saliente es agregada, el router PE de salida debe realizar una búsqueda IP en la tabla CEF de VRF después de la búsqueda de etiqueta en el LFIB.

Observe en los ejemplos para ver las etiquetas anunciadas por LDP y MP-iBGP y su uso en la tabla VFF CEF y LFIB. Estas etiquetas se corresponden con las etiquetas de las figuras 133, 134 y 135.

Figura 133. **VRF CEF cust one en ingreso PE**

```
Ingreso-PE#show ip cef vrf cust-one 10.10.100.1 255.255.255.255 detail
10.10.100.1/32, epoch 0
  recursive via 10.200.254.2 label 30
    nexthop 10.200.214.1 POS0/1/0 label 16
```

Fuente: elaboración propia, empleando Visio 2013.

Figura 134. **Ruta VPNv4 en ingreso PE**

```
Ingreso-PE#show ip bgp vpnv4 rd 1:1 10.10.100.1
BGP routing table entry for 1:1:10.10.100.1/32, version 81
Paths: (1 available, best #1, table cust-one)
  Not advertised to any peer
  Local
    10.200.254.2 (metric 3) from 10.200.254.2 (10.200.254.2)
      Origin incomplete, metric 1, localpref 100, valid, internal, best
      Extended Community: RT:1:1,
      mpls labels in/out nolabel/30
```

Fuente: elaboración propia, empleando Visio 2013.

Figura 135. **Entrada de LFIB en ingreso PE**

```
Egreso-PE#show mpls forwarding-table labels 30
Local      Outgoing       Prefix              Bytes Label     Outgoing       Next Hop
Label      Label or VC    or Tunnel Id        Switched        interface
30         No Label       10.10.100.1/32[V]   0               Et0/1/2        10.10.2.1
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.81. Protocolos de enrutamiento PE-CE

El enrutamiento debe ocurrir entre los *routers* PE y CE. Los protocolos de enrutamiento PE-CE que admite Cisco IOS son enrutamiento estático, RIPv2, *open shortest path first* (OSPF), *enhanced interior gateway routing protocol* (EIGRP), *intermediate system-to-intermediate system* (IS-IS) y eBGP.

3.5.82. Rutas conectadas

Estrictamente hablando, las rutas conectadas no son un protocolo de enrutamiento. Sin embargo, para garantizar la conectividad, es una buena práctica redistribuir las rutas conectadas en el router PE a BGP. De esta forma, cuando el usuario inicia un *ping* desde un *router* CE al *router* CE remoto, el paquete de retorno se reencamina. De forma predeterminada, si el usuario envía un ping y no especifica la dirección IP de origen, toma como dirección IP de origen la dirección IP de la interfaz de salida, que en el caso de un *router* CE es una dirección IP de la subred en el enlace PE-CE. Como tal, el paquete de devolución tiene esta dirección IP como la dirección IP de destino. Por lo tanto, está prefijo debe conocerse en los sitios remotos para que el *ping* tenga éxito. Pueda optar por no distribuir las subredes conectadas en BGP, pero luego debe iniciar un *ping* de CE a CE especificando una dirección IP de origen diferente en el *router* CE. Luego debe incluir esta dirección IP en el protocolo de enrutamiento PE-CE específico. Lo mismo aplica para otras aplicaciones, como telnet. En el ejemplo se muestra el comando redistribuido conectado bajo la familia de direcciones ipv4 para el VRF. Como se mencionó anteriormente, se usa la familia de direcciones ipv4 vrf vrf-name en el proceso bgp del *router* en los *routers* PE para configurar las secciones BGP y los parámetros hacia los routers CE, a través de las interfaces VRF. Está es también el lugar donde otros protocolos de enrutamiento VRF se redistribuyen en BGP.

Figura 136. **Redistribución de rutas conectadas en BGP**

```
router bgp 1
…
 !
 address-family ipv4 vrf cust-one
 redistribute connected
 neighbor 10.10.2.1 remote-as 65001
 neighbor 10.10.2.1 activate
 exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.83. Enrutamiento estático

El enrutamiento estático es el más simple de todos los enrutamientos para configurar. Sin embargo, puede ser tedioso cuando manualmente necesita configurar muchas rutas estáticas. Para admitir VRF, las rutas estáticas se han hecho conscientes de VRF para que se pueden configurar en el *router* PE para enrutar el tráfico en los VRF. En la figura 137 se muestra una ruta estática para el prefijo 10.88.1.1/32 que apunta a un próximo salto 10.10.2.1, qué es la dirección IP de la interfaz del enlace PE-CE en el *router* CE. Puede ver que la ruta estática se aplica al VRF *cust-one* y que la ruta se instala en la tabla de enrutamiento VRF asociada con VRF *cust-one*.

Figura 137. **Configuración VRF OSPF**

```
!
ip route vrf cust-one 10.88.1.1 255.255.255.255 10.10.2.1
!

paris#show ip route vrf cust-one static
     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
S       10.88.1.1/32 [1/0] via 10.10.2.1
```

Fuente: elaboración propia, empleando Visio 2013.

Para asegurarse de que la ruta estática se aprenda en los otros *routers* PE como una ruta vpnv4, debe distribuir las rutas estáticas en BGP bajo la familia de direcciones para el VRF específico. La figura 138 muestra el comando redistribuir para rutas estáticas.

Figura 138. **Distribución de rutas estáticas en BGP**

```
router bgp 1
…
 !
 address-family ipv4 vrf cust-one
 redistribute connected
 redistribute static
 exit-address-family
 !
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.84. RIP versión 2

El protocolo de información de enrutamiento (*routing infomation protocol*, RIP) es un protocolo simple de enrutamiento de vector distancia. Su uso es limitado y no es un protocolo de enrutamiento adecuado para grandes redes debido a su lentitud de convergencia. Sin embargo, todavía se utiliza a menudo en redes pequeñas como un protocolo de enrutamiento rápido y sucio que hace el trabajo con respecto a la funcionalidad básica de enrutamiento. La versión 2 de RIP (RIPv2) ha experimentado algunas mejoras con respecto a la primera especificación de RIP, pero sigue siendo un protocolo de enrutamiento limitado. Las siguientes son algunas mejoras:

- Incluir una máscara de subred con los prefijos.

- Usar la dirección de Multicast 224.0.0.9 en lugar de la dirección de difusión 255.255.255.255.
- Incluyendo una dirección de siguiente salto.
- Incluir una etiqueta de ruta.
- Usar autenticación (opcional).

En Cisco IOS, RIPv2 es compatible como un protocolo de enrutamiento PE-CE, pero la versión RIPv1 no lo es. Puede ver la configuración RIPv2 VRF básica en un *router* PE en el ejemplo. Solo existe un proceso RIPv2 en el *router* PE. La configuración específica necesaria por VRF se configura en la familia de direcciones específicas. Asegúrese de que el comando *default-metric* está configurado para RIP. De lo contrario, no se distribuyen rutas de BGP a RIP.

Figura 139. **Configuración RIPv2 VRF**

```
!
ip vrf cust-one
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
router rip
 no auto-summary
 !
 address-family ipv4 vrf cust-one
 redistribute bgp 1
 network 10.0.0.0
 default-metric 2
 version 2
 exit-address-family
!
router bgp 1
 …
 !
 address-family ipv4 vrf cust-one
 redistribute connected
 redistribute rip
 exit-address-family
!
```

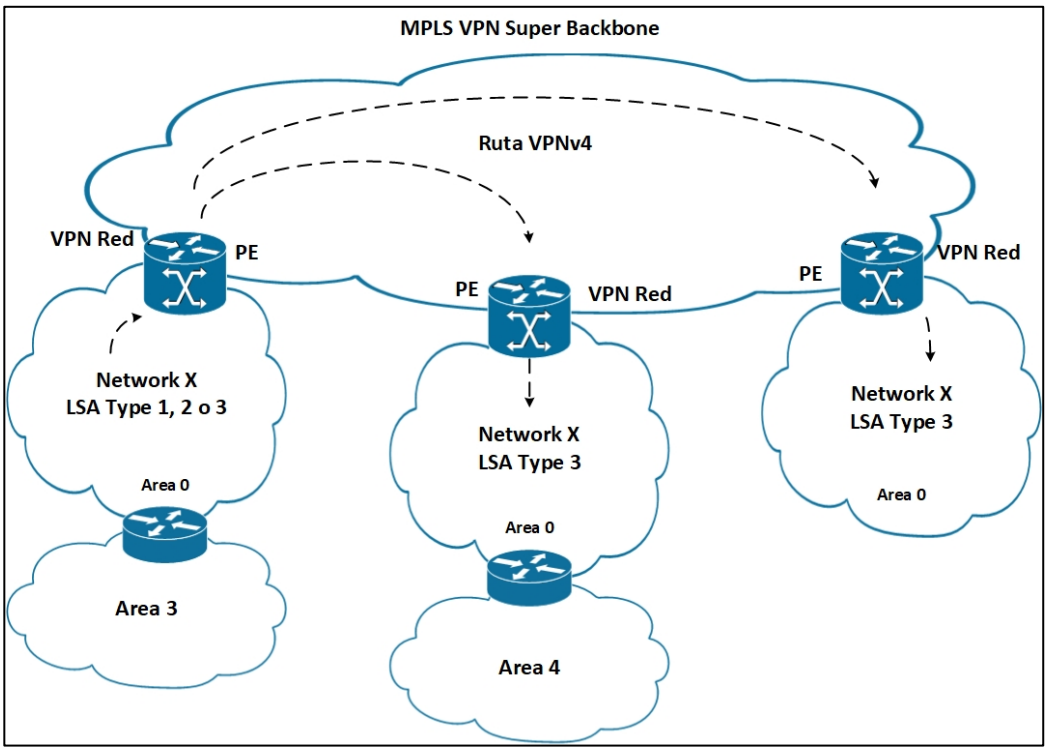
Fuente: elaboración propia, empleando Visio 2013.

3.5.85. OSPF

OSPF puede ser el protocolo de enrutamiento en el enlace PE-CE. Para propagar las rutas del cliente de PE a PE, OSPF se redistribuye en iBGP y viceversa en los *routers* PE. El inconveniente de esto es que todas las rutas OSPF se convierten en rutas externas en el PE remoto cuando las rutas se redistribuyen nuevamente a OSPF. El resultado de esto sería que todas las rutas OSPF que atraviesan la red troncal MPLS VPN serían menos preferibles que las rutas que no atravesaban la red troncal, sino que se enviaron a través de un enlace interno (enlace de puerta trasera) de un sitio OSPF a otro.

Para evitar que todas las rutas redistribuidas se conviertan en prefijos externos OSPF, las rutas OSPF internas se anuncian como rutas de resumen (anuncio de estado de enlace [LSA] tipo 3), que son rutas inter área, en el PE cuando se redistribuyen desde BGP de nuevo a OSPF. Este no es el comportamiento normal, porque los *routers* PE redistribuyen las rutas BGP en OSPF y son *routers* límite de un sistema autónomo (ASBR) que deben anunciar las rutas como rutas OSPF externas (LSA tipo 5). En efecto, es como si los routers PE fueran *routers* de borde de área (ABR) que anuncian rutas de resumen en otra área. Sin embargo, todas las rutas internas de OSPF (rutas inter áreas e interareas) se convierten en rutas interareas (LSA tipo 3) después de que BGP las propaga, incluso si los números de área coinciden en diferentes routers PE. La figura 140 muestra la propagación de las rutas internas de OSPF a través de la red troncal MPLS VPN.

Figura 140. **Rutas OSPF internas a través de la red *backbone* MPLS VPN**

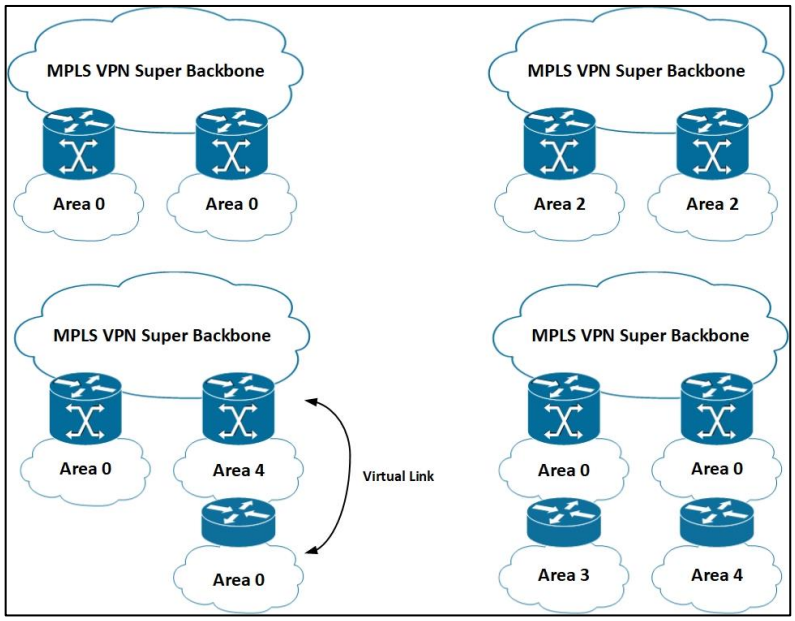


Fuente: elaboración propia, empleando Visio 2013.

La preferencia de ruta OSPF normal dicta que las rutas intra-área son más preferidas que las rutas OSPF interarea. Debido a que todas las rutas OSPF internas se convierten en rutas interarea en los sitios remotos, las rutas intraarea aún pueden causar un problema al convertirse en rutas interarea cuando existe un enlace de puerta trasera entre sitios. Las rutas dentro del área siguen siendo rutas dentro del área a través del enlace de la puerta trasera, pero se convierten en rutas interareas a través de la red troncal MPLS VPN. Por lo tanto, las rutas intra- área que se anuncian a través del enlace de puerta trasera siempre son preferidas. Para evitar esto, debe configurar un enlace especial, llamado enlace simulado, entre los *routers* PE.

Los *routers* PE tienen áreas OSPF conectadas a ellos. Estas áreas pueden ser el área del *backbone* 0 o cualquier otra área. La red troncal MPLS VPN se puede considerar una jerarquía adicional que es más alta que el área de la red troncal OSPF: la red troncal súper VPLS MPLS. En la figura 141 se muestra este concepto.

Figura 141. **Posibles escenarios de VPN MPLS de OSPF**



Fuente: elaboración propia, empleando Visio 2013.

3.5.86. Configuración OSPF VRF

Para ejecutar OSPF para un VRF, configure el comando de proceso OSPF con la palabra clave VRF. La sintaxis es *router ospf process-id vrf vrf-name*. Tenga en cuenta que RIPv2 y EIGRP solo tienen un proceso de enrutamiento con una familia de direcciones por VRF configurada. OSPF tiene un proceso OSPF separado por VRF.

En la figura 142 se muestra la configuración OSPF VRF básica. Obviamente, el proceso OSPF VRF necesita ser redistribuido en BGP y viceversa. Puede configurar todos los comandos reguladores OSPF para el proceso OSPF VRF. Asegúrese de tener la palabra clave subredes en el comando redistribute bgp en el proceso ospf del *router*. De lo contrario, solo las rutas con clase se redistribuyen. Cuando está redistribuyendo OSPF en BGP, asegúrese de configurar los parámetros de coincidencia apropiados en el comando redistribuir para que pueda redistribuir el tipo apropiado de rutas OSPF.

Figura 142. **Configuración básica de OSPF VRF**

```
!
ip vrf cust-one
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
interface Loopback1
 ip vrf forwarding cust-one
 ip address 10.99.1.1 255.255.255.255
!
router ospf 42 vrf cust-one
 router-id 10.99.1.1
 log-adjacency-changes
 redistribute bgp 1 metric 10 subnets
 network 10.10.2.0 0.0.0.255 area 0
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 10.200.254.5 remote-as 1
 neighbor 10.200.254.5 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.200.254.5 activate
 neighbor 10.200.254.5 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf cust-one
 redistribute connected
 redistribute ospf 42 vrf cust-one metric 10 match internal external 1 external 2
 exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

La figura 143 muestra el proceso OSPF VRF 42 ejecutándose en el *router* PE.

Figura 143. **Comando *show* IP OSPF**

```
paris#show ip ospf 42
 Routing Process "ospf 42" with ID 10.99.1.1
   Domain ID type 0x0005, value 0.0.0.42
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Connected to MPLS VPN Superbackbone, VRF cust-one
 It is an area border and autonomous system boundary router
 Redistributing External Routes from,
    bgp 1 with metric mapped to 10, includes subnets in redistribution
...
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:04:35.120 ago
        SPF algorithm executed 25 times
        Area ranges are
        Number of LSA 17. Checksum Sum 0x0E27D6
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 13
        Flood list length 0
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.87. Propagación de métrica OSPF

Cuando redistribuye las rutas OSPF internas y externas desde OSPF a MP-BGP en el *router* PE, el *router* PE usa la métrica OSPF para establecer el BGP MED. El MED a menudo se conoce como la métrica externa de una ruta BGP. Es parte del mejor proceso de selección de ruta BGP. BGP puede usarlo para seleccionar la mejor ruta cuando dos o más hablantes de BGP anuncian la

misma ruta en BGP. El MED se muestra como 'métrica' en la tabla BGP en Cisco IOS. Observe el ejemplo, donde los prefijos 1: 1: 10.10.100.1/32 y 1: 1: 10.200.200.1/32 se anuncian con un MED (métrica externa) de 10. Cuando el PE redistribuye la ruta desde BGP en OSPF, el PE utiliza el MED para establecer la métrica OSPF de la ruta interna o externa de OSPF. Cuando se utiliza el comando *default-metric metric value* o la opción de métrica en el comando *redistribute*, anula está comportamiento porque establece directamente el indicador en el valor configurado.

3.5.88. Comunidades extendidas BGP para OSPF

Para llevar las características de las rutas OSPF a través de la red troncal MPLS VPN, se definieron varias comunidades extendidas BGP adicionales. Las características de OSPF que se transportan con MPBGP incluyen lo siguiente:

- Tipo de ruta
- Número de área
- ID del *router* OSPF
- ID de dominio
- Métrica tipo 1 o 2 para rutas externas de OSPF

Las comunidades extendidas de BGP específicas de OSPF permiten que la ruta de OSPF se reconstruya completamente en el *router* de PE remoto. El tipo de ruta permite al *router* de PE remoto descubrir qué tipo de ruta anunciar en OSPF. Si el tipo de ruta es 1, 2 o 3 (correspondiente a los tipos 1, 2 o 3 de LSA), el *router* de PE remoto anuncia una ruta de resumen interarea (LSA tipo 3) en el área OSPF.

El ID de dominio indica al PE remoto si se anunciara una ruta OSPF externa. Si el ID de dominio (establecido de manera predeterminada igual al identificador de proceso del router OSPF) de la ruta recibida por un router PE no coincide con el ID del proceso OSPF del VRF particular, la ruta se anuncia como una ruta externa OSPF (tipo 5 de LSA) tipo 2 para proporcionar soporte para redes que redistribuyen rutas IP entre diferentes procesos OSPF. Si la ID del dominio coincide con la ID del proceso OSPF, la ruta se anuncia como una ruta interna. Puede cambiar la ID de dominio en el router PE con el comando *domain-id ospf domain* ID.

Un tipo de rua de 5 indica que la ruta vpnv4 que se recibe es una ruta externa OSPF. Como tal, el PE inunda una LSA de tipo 5 en el sitio OSPF; de forma predeterminada, el tipo de indicador es el tipo 2. El indicador que se utiliza para el tipo 5 LSA es el MED de la ruta BGP vpnv4 si hay uno presente. Si no hay ninguno presente, se usa la métrica OSPF predeterminada.

La figura 144 se muestra las comunidades extendidas de BGP para OSPF. El prefijo 10.10.100.1/32 es una ruta interna de OSPF, y el prefijo 10.200.200.1/32 es una ruta externa de OSPF.

Figura 144. **Comunidad extendida BGP para OSPF**

```
yakarta#show ip bgp vpnv4 rd 1:1 10.10.100.1
BGP routing table entry for 1:1:10.10.100.1/32, version 2045
Paths: (2 available, best #2, table cust-one)
  Advertised to update-groups:
     1
  Local
  10.200.254.2 (metric 3) from 10.200.254.2 (10.200.254.2)
    Origin incomplete, metric 10, localpref 100, valid, internal
    Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x0000002A0200
      OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.10.2.2:512,
    mpls labels in/out 18/28

yakarta#show ip bgp vpnv4 rd 1:1 10.200.200.1
BGP routing table entry for 1:1:10.200.200.1/32, version 5649
Paths: (1 available, best #1, table cust-one)
  Not advertised to any peer
  Local
    10.200.254.2 (metric 3) from 10.200.254.2 (10.200.254.2)
      Origin incomplete, metric 10, localpref 100, valid, internal, best
      Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x0000002A0200
        OSPF RT:0.0.0.0:5:1 OSPF ROUTER ID:10.99.1.1:1281,
      mpls labels in/out nolabel/18
```

Fuente: elaboración propia, empleando Visio 2013.

Puede ver que el tipo de ruta OSPF está codificado como área: *route-type*: *option*. El área está codificado en 4 bytes; el tipo de ruta y la opción son de 1 byte cada uno. Si la ruta es externa, el área es 0.0.0.0. Si se establece el bit menos significativo del campo de opción, indica una métrica externa tipo 2. Si el bit menos significativo no está establecido, indica una métrica externa tipo 1.

3.5.89. Diseño de red OSPF MPLS VPN

OSPF está diseñado para trabajar con áreas. El área especial 0 es el área de la red troncal que conecta todas las demás áreas. MPLS VPN tiene otra 'área' entre sitios OSPF: el *backbone* MPLS VPN. Esto no es un área OSPF, por supuesto, porque ejecuta iBGP. Sin embargo, actúa como un área ya que los *routers* PE actúan como ABR. Por lo tanto, puede considerar la red troncal MPLS VPN como la red troncal súper en la jerarquía OSPF. Si varios sitios de

un VRF tienen un *router* PE con área 0, el área de la red troncal se divide en más de una parte. Normalmente, Un área de *backbone* dividida requiere un enlace virtual para interconectar las partes. Sin embargo, esto no es necesario para MPLS VPN, porque iBGP lleva las rutas OSPF, las rutas OSPF se vuelven a crear en los routers PE, y la red troncal MPLS VPN cuando se utiliza un enlace simulado.

Los routers PE funcionan como ABR ya que anuncian LSA de tipo 3 a los *routers* CE. Los *routers* CE pueden estar en el área 0 o cualquier otra área. Sin embargo, si un sitio tiene más de un área, los *routers* PE deben estar en el área 0 porque son ABR. Si no lo son, un enlace virtual entre el *router* PE y el ABR más cercano en el sitio del cliente debe elevar el área 0 al *router* PE. Obsérvese la figura nuevamente para ver los posibles escenarios con respecto a las áreas OSPF y las conexiones a la red troncal MPLS VPN.

3.5.90. Enlace simulado

Si dos sitios pertenecen a la misma área y están interconectados con un enlace de puerta trasera, parecen como un área para OSPF. A través del enlace de puerta trasera, todos los LSA se inundan sin modificaciones de un sitio a otro. Esto significa que las rutas dentro del área se mantienen dentro de las rutas del área. Las rutas intra-área (tipo 1 y 2 LSA) se inundan a través del enlace de puerta trasera. Se transforman en rutas interrarea a través de la red troncal MPLS VPN. Eso significa que la ruta preferida entre los dos sitios es siempre el enlace de puerta trasera porque OSPF siempre prefiere las rutas intra-área sobre las rutas interrarea. Esto reduce el servicio MPLS VPN a una mera solución de respaldo en caso de que el enlace de puerta trasera se caiga.

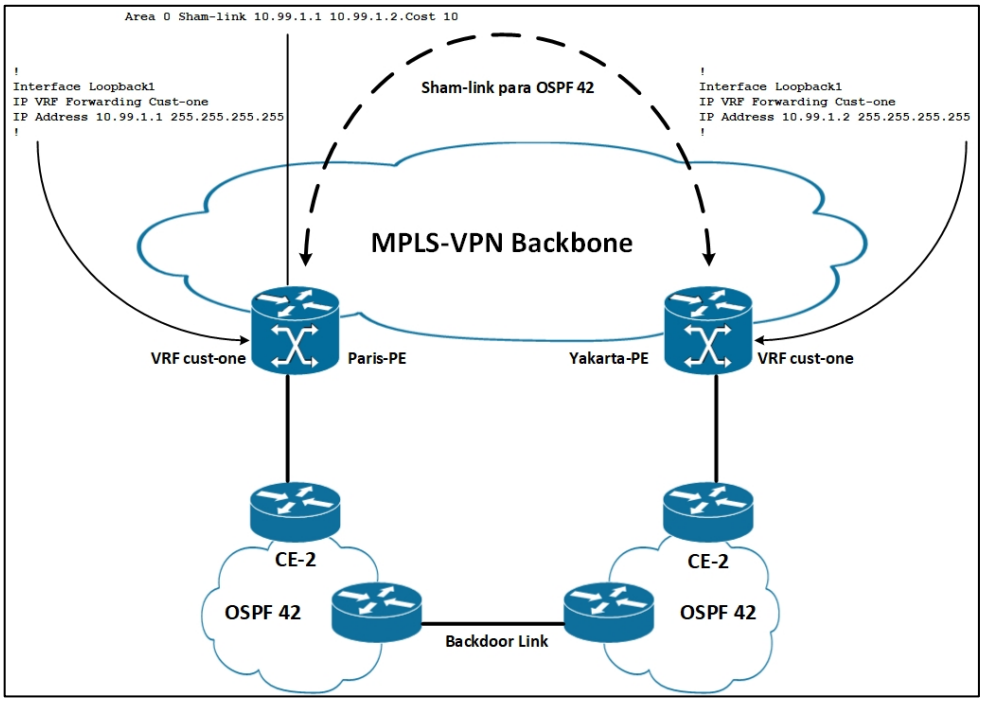
El concepto de enlace simulado fue inventado para resolver este problema. El enlace simulado no es un enlace real sino uno falso entre dos *routers* PE. Es un enlace intra-área OSPF creado entre los dos routers PE para que puedan inundar está enlace en el área conectada a ambos routers PE. El enlace simulado tiene dos puntos finales. El punto final del enlace simulado en cada router PE es una /32 dirección IPv4 del VRF específico. iBGP debe anunciar está /32 dirección IPv4 de un PE a otro como un prefijo vpnv4. El enlace simulado es un enlace intraarea punto a punto no numerado que se trata como un enlace de circuito de demanda. Esto significa que los LSA se inundan a través del enlace simulado, pero no se producen inundaciones periódicas de actualización a través del enlace ficticio.

El enlace simulado se incluye en el cómputo de ruta más corta primero (SPF), al igual que cualquier enlace en OSPF. Como los LSA se inundan a través del enlace simulado, todos los tipos de ruta OSPF se pueden conservar y no tienen que convertirse en tipos 3 o 5 de LSA. Si el enlace simulado falla, se produce el mecanismo predeterminado para enviar solo LSA de tipo 3 y tipo 5 al sitio.

El enrutamiento a través de la red troncal MPLS VPN sigue siendo posible si el enlace simulado falla, pero el enlace de puerta trasera es la ruta preferida para las rutas dentro del área porque las rutas intra-área todavía se aprenden de esa manera. Usted configura el enlace simulado especificando una dirección IP de origen en el VRF en el PE local y una dirección IP de destino en el VRF en el PE remoto. Además, puede especificar un costo para el enlace simulado para que sea más o menos preferido que el enlace de puerta trasera. La sintaxis del comando *sham link* es area *area-id-sham-link* source-address destination-address cost *number*.

La figura 145 se muestra un ejemplo de un enlace simulado configurado para el proceso OSPF 42 en VRF *cust-one*. Ambos sitios que están conectados a los *routers* PE están en el área OSPF 0 de VRF *cust-one*. Un enlace simulado se configura entre los *routers* PE con puntos finales en el VRF *cust-one*.

Figura 145. **Ejemplo de un enlace simulado**



Fuente: elaboración propia, empleando Visio 2013.

Puede ver la configuración para el enlace simulado en la figura 146.

Figura 146. **Enlace simulado OSPF**

```
!
router ospf 42 vrf cust-one
 router-id 10.99.1.1
 log-adjacency-changes
 area 0 sham-link 10.99.1.1 10.99.1.2 cost 10
 redistribute bgp 1 metric 10 subnets
 network 10.10.2.0 0.0.0.255 area 0
!
paris#show ip ospf 42 neighbor


Neighbor ID       Pri State         Dead Time         Address           Interface
10.200.200.1        1 FULL/DR       00:00:35          10.10.2.1         Ethernet0/1/2
10.99.1.2           0 FULL/ -          -              10.99.1.2         OSPF_SL2

paris#show ip ospf 42 sham-links
Sham Link OSPF_SL2 to address 10.99.1.2 is up
Area 0 source address 10.99.1.1
  Run as demand circuit
  DoNotAge LSA allowed. Cost of using 10 State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Hello due in 00:00:03
    Adjacency State FULL (Hello suppressed)
    Index 2/2, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

Fuente: elaboración propia, empleando Visio 2013.

iBGP, no OSPF, siempre debe anunciar los puntos finales del enlace simulado. De lo contrario, en enlace de simulación intermitente. Primero, iBGP aprende los puntos finales del enlace simulado y se crea el enlace simulado. Cuando se crea el enlace falso, OSPF anuncia los puntos finales del enlace simulado, si el comando de red los incluye. La distancia de OSPF es 110, frente a 200 para las rutas iBGP. Por lo tanto, las rutas del punto final del enlace simulado están en la tabla de enrutamiento como rutas OSPF, porque la distancia para las rutas OSPF es menor que la distancia para las rutas iBGP. Tan pronto como los puntos finales ya no se aprenden en la tabla de enrutamiento a través de iBGP, en enlace simulado baja y el proceso comienza de nuevo. El resultado es un flapping continuo del enlace simulado.

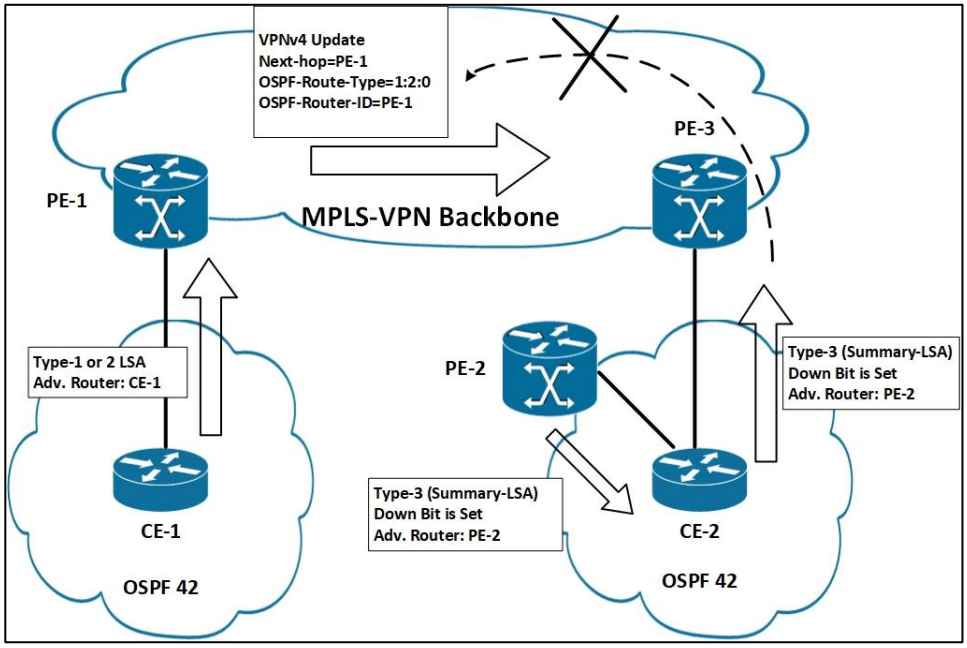
Incluso si existe un enlace simulado y las rutas OSPF están inundadas a través de el, iBGP aún necesita anunciar las rutas OSPF como rutas vpnv4 desde el *router* PE al *router* PE. La razón de esto es que iBGP aún necesita llevar la etiqueta MPLS VPN para cada ruta OSPF para que los paquetes se puedan reenviar correctamente a través de la red troncal MPLS VPN.

Si se aprende un prefijo a través del enlace simulado y se selecciona la ruta a través del enlace ficticio como la mejor, el *router* PE no se genera una actualización MP-BGP para el prefijo. Esto significa que las rutas OSPF aprendidas a través del enlace simulado no se redistribuyen en BGP. El *router* PE en el otro lado del enlace simulado ya ha redistribuido las rutas OSPF en BGP, por lo que no es necesario hacerlo una segunda vez.

3.5.91. *Bit Down* y etiqueta de dominio

El bit de reducción es un poco establecido en el campo de opciones de un OSPF LSA tipo 3. Indica la dirección en que se ha anunciado la ruta. Si la ruta OSPF se ha anunciado desde un *router* PE en un área OSPF, se establece el bit hacia abajo. Otro *router* PE en la misma área no redistribuye está ruta en iBGP de la red MPLS VPN si está bit está configurado. El *router* PE ni siquiera incluye la ruta en el cálculo SPF. Como tal, puede evitar un posible bucle de enrutamiento si el sitio es multitarjeta o si existe un enlace de puerta trasera entre los sitios OSPF. La figura 147 demuestra que una ruta OSPF con el *bit down* configurado no se visualiza en iBGP.

Figura 147. ***Down Bit***

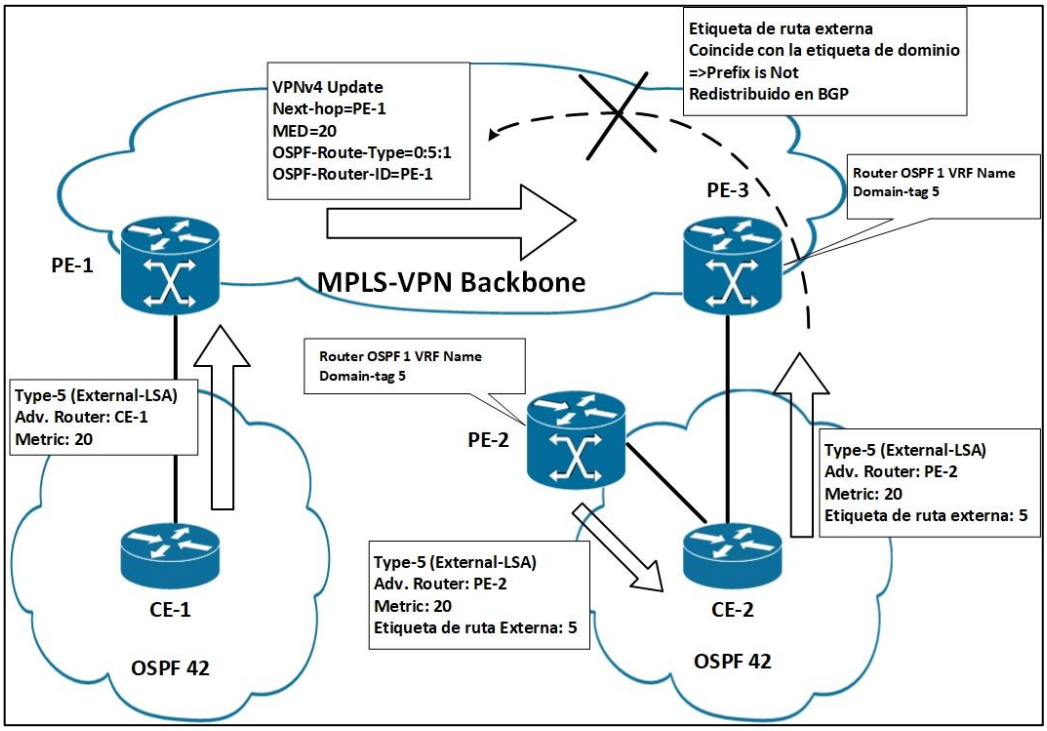


Fuente: elaboración propia, empleando Visio 2013.

La etiqueta de dominio (también conocida como etiqueta de ruta VPN) cumple la misma función que el *down bit*, pero para las rutas externas de OSPF. Puede configurarlo manualmente en los *routers* PE con el comando *domaintag tag-value*. Si configura la etiqueta de dominio con un valor particular en un router PE, el valor de la etiqueta de la ruta OSPF externa se establece en ese valor. Si otro *router* PE que está conectado al mismo sitio u otro sitio que está conectado a través de un enlace de puerta trasera recibe está ruta y coincide con la etiqueta de dominio configurada, la ruta no se redistribuye en iBGP. De forma predeterminada, la etiqueta de dominio se establece en un valor como se determina en RFC 1745. Este RFC especifica la interacción entre OSPF y BGP y determina cómo se debe establecer automáticamente la etiqueta. El número de sistema autónomo de BGP está codificado en la etiqueta

de las rutas externas de OSPF en los 16 *bits* menos significativos. En la figura 148 se muestra que una ruta OSPF no se vuelve a leer en iBGP si la etiqueta de dominio de la ruta coincide con la etiqueta de dominio configurada en el router PE.

Figura 148. **Etiqueta de dominio**



Fuente: elaboración propia, empleando Visio 2013.

3.5.92. EIGRP MPLS

EIGRP puede ser el protocolo de enrutamiento PE-CE. La desventaja habitual de la redistribución entre iBGP y el protocolo de enrutamiento entre el router PE y CE también está presente aquí. Esto significa que redistribuir las rutas de BGP a EIGRP hace que todas las rutas sean rutas EIGRP externas.

Sin embargo, la mayor calidad posible de información de EIGRP está codificada en nuevas comunidades extendidas de BGP para aliviar el problema esto permite que el *router* PE remoto reconstruya la ruta EIGRP con todas sus características, incluidos los componentes métricos, AS, TAG y, para las rutas externas, el número AS remoto, el ID remoto, el protocolo remoto y la métrica remota. Estas son las características EIGRP de un prefijo que puede encontrar en la tabla de topología. Si la ruta anunciada por EIGRP es interna, la ruta se anuncia como una ruta interna en el sitio remoto si el AS de destino coincide con el AS fuente de la comunidad extendida de BGP. Si los números AS no coinciden, la ruta se reconstruye como una ruta EIGRP externa. La tabla muestra las comunidades extendidas BGP que transmiten la información EIGRP.

Tabla V. **Topología**

Tipo	Uso	Valor
0x8800	Información general de ruta	Banderas + etiquetas
0x8801	Enrutar la información métrica y sistema autónomo	Sistema Autónomo + Demora
0x8802	Información de ruta externa	Confiabilidad + contador de salto + BW
0x8803	Información de ruta externa	Campo reservado + Carga + MTU
0x8804	Información de ruta externa	Sistema remoto autónomo + ID remoto
0x8805	Información de ruta externa	Protocolo remoto + métrica remota

Fuente: elaboración propia.

La figura 149 demuestra cómo BGP lleva estas comunidades extendidas. El prefijo 10.10.100.1/32 es un prefijo EIGRP interno, y el prefijo 10.200.200.1/32 es un prefijo EIGRP externo.

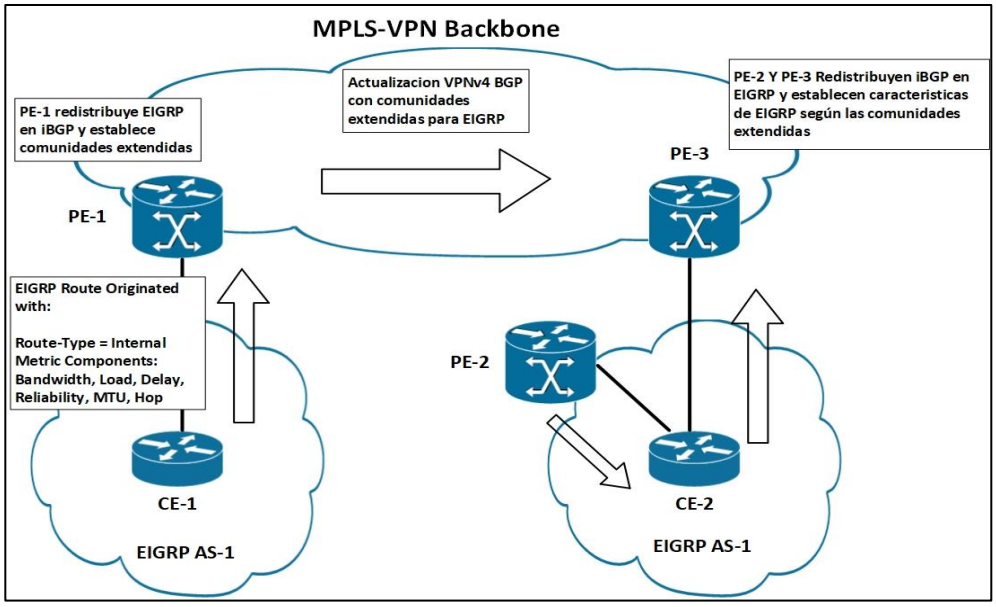
Figura 149. **Comunidades extendidas BGP para EIGRP**

```
amsterdam#show ip bgp vpnv4 all 10.10.100.1
BGP routing table entry for 1:1:10.10.100.1/32, version 28
Paths: (1 available, best #1, table cust-one)
  Advertised to update-groups:
     1
  Local
    10.10.2.1 from 0.0.0.0 (10.200.254.2)
      Origin incomplete, metric 409600, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:1:1 Cost:pre-bestpath:128:409600 0x8800:32768:0
        0x8801:42:153600 0x8802:65281:256000 0x8803:65281:1500,
      mpls labels in/out 22/nolabel

yakarta#show ip bgp vpnv4 all 10.200.200.1
BGP routing table entry for 1:1:10.200.200.1/32, version 91
Paths: (1 available, best #1, table cust-one)
Flag: 0x820
  Not advertised to any peer
  Local
    10.200.254.2 (metric 4) from 10.200.254.2 (10.200.254.2)
      Origin incomplete, metric 409600, localpref 100, valid, internal, best
      Extended Community: RT:1:1 Cost:pre-bestpath:129:409600 0x8800:0:0
        0x8801:42:153600 0x8802:65281:256000 0x8803:65281:1500
        0x8804:0:168453121 0x8805:11:0,
      mpls labels in/out nolabel/31
```

Fuente: elaboración propia, empleando Visio 2013.

La figura 150 muestra cómo se propaga una ruta EIGRP a través de la red troncal MPLS VPN desde un sitio EIGRP a otro.

Figura 150. **Propagación de una ruta EIGRP a través de la red troncal MPLS VPN**



Fuente: elaboración propia, empleando Visio 2013.

En el lado derecho, PE-2 y PE-3 redistribuyen la ruta vpnv4 de iBGP a EIGRP. Sin embargo, la misma ruta puede recibirse como una ruta EIGRP desde el otro *router* PE en el mismo sitio. Sin embargo, la ruta vpnv4 que se aprende de PE-1 siempre se prefiere sobre la ruta EIGRP que se aprende de la otra PE en el mismo sitio. Esto se debe a que se compara la métrica de las rutas recibidas y la métrica más baja siempre gana. Está es siempre la ruta vpnv4 desde el *router* PE remoto, si el costo de la ruta EIGRP se calcula reconstruyendo los componentes métricos de las comunidades extendidas. Esta es la razón porque EIGRP no necesita un mínimo, como lo hace OSPF. El costo de atravesar el backbone MPLS VPN es 0 para las rutas EIGRP.

3.5.93. Configuración

De forma similar a RIPv2, EIGRP se ha ampliado con soporte para familias de direcciones para que pueda admitir MPLS VPN. Por los tanto, la configuración de VRF en los *routers* PE para EIGRP se encuentra en la familia de direcciones IPv4 vrf. Los comandos regulares de EIGRP están disponibles para cada VRF configurado. Debido a que los clientes VPN usualmente utilizan diferentes números EIGRP AS (y el número AS de coincidir entre los vecinos EIGRP), el nuevo comando EIGRP de sistema autónomo como número le permite especificar el número de sistema autónomo para el VRF especificado. La figura 151 muestra la configuración de un *router* PE que está configurado para dos VRF que ejecutan EIGRP. Como con todos los demás protocolos de enrutamiento PE-CE, debe configurar la redistribución de BGP en EIGRP y viceversa.

Figura 151. **Ejemplo de configuración EIGRP VRF**

```
!
router eigrp 1
 no auto-summary
 !
 address-family ipv4 vrf cust-two
 redistribute static metric 64 2000 255 1 1500
 redistribute bgp 1 metric 300 40000 255 1 1500
 network 10.10.0.0 0.0.255.255
 no auto-summary
 autonomous-system 33
 exit-address-family
 !
 address-family ipv4 vrf cust-one
 redistribute bgp 1 metric 300 40000 255 1 1500
 network 10.0.0.0
 no auto-summary
 autonomous-system 42
 exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.94. POI *pre-bestpath*

La comunidad de costos en BGP es una comunidad no transitiva que se transfiere a iBGP y pares de la confederación, pero no más allá. Influye en el mejor proceso de selección de ruta de BGP al asignar valores de costo a rutas específicas. La comunidad de costo se establece con el comando *set extcommunity cost* en un mapa de ruta. Puede establecer un ID de comunidad de costo (0-255) y un valor de costo (0 - 4,294,967,295). El ID de comunidad de costo indica la preferencia de este camino BGP frente a los demás. Cuanto menor sea la identificación del costo, más preferido es.

El punto de inserción (POI) es el lugar en el mejor proceso de selección de ruta de BGP donde BGP considera la comunidad de costos. El POI *pre-bestpath* indica que BGP debe considerar la comunidad de costos antes de cualquiera de las comparaciones regulares de BGP pasos en el bien conocido proceso de selección BGP *best path*. Puede configurar una comunidad de costo *pre-bestpath* configurado la comunidad de costo con la palabra clave *pre-bestpath* en un mapa de ruta. La comunidad de costo tiene la forma Costo: PDI: ID: valor.

Es la comunidad de costo con *pre-bestpath* que se establece cuando EIGRP se redistribuye en BGP. Sin la comunidad de costos para EIGRP en el router PE, el router PE siempre prefiere la ruta BGP de origen local por encima de la ruta aprendida por un par BGP. En el caso de tener un enlace de backdoor es la puerta preferida. Con la comunidad de costos para EIGRP, se comparan el enlace de backdoor y el camino aprendido de iBGP a través de la red troncal MPLS VPN. La ruta con el costo EIGRP más bajo es la ruta preferida. La comunidad de costos para EIGRP sobre MPLS VPN se enciende automáticamente en el caso de EIGRP como el protocolo de enrutamiento PE-

CE, por lo que no es necesario configurarlo. El POI es *pre-bestpath*. El ID de la comunidad de costo es 128 o 129 para rutas internas EIGRP y 129 para rutas externas EIGRP. El valor de este es el valor métrico compuesto EIGRP establecido en el *router* PE que redistribuye la ruta BGP. Las rutas que tienen un valor inferior son preferibles a las rutas que tienen un valor mayor. Si el ID de comunidad de costos de la ruta y el valor son los mismos, Cisco IOS prefiere la ruta EIGRP sobre la ruta BGP en el router PE.

En la figura 152 se muestra la comunidad de costos que utiliza EIGRP en un escenario MPLS VPN. Dos *routers* PE anuncian el prefijo vpnv4 10.10.100.1/32, y cada uno establece la comunidad de costo en un ID de 128 y un valor que representa la métrica EIGRP compuesta como se ve en un *router* Yakarta PE. El Yakarta PE puede elegir la mejor ruta en función del valor de la comunidad de costo que anuncian los router PE, ignorando los otros atributos de BGP.

Figura 152. **Comunidad de costos para EIGRP sobre MPLS VPN**

```
yakarta#show ip bgp vpnv4 all 10.10.100.1
BGP routing table entry for 1:1:10.10.100.1/32, version 1259
Paths: (2 available, best #2, table cust-one)
  Advertised to update-groups:
     1
  Local
    10.200.254.2 (metric 3) from 10.200.254.2 (10.200.254.2)
      Origin incomplete, metric 256384000, localpref 100, valid, internal
      Extended Community: RT:1:1
        Cost:pre-bestpath:128:256384000 (default-1891099647) 0x8800:32768:0
        0x8801:42:256128000 0x8802:65281:256000 0x8803:65281:1500,
      mpls labels in/out 16/16
  Local
    10.10.4.2 from 0.0.0.0 (10.200.254.5)
      Origin incomplete, metric 2323456, localpref 100, weight 32768, valid, sourced, best
      Extended Community: SoO:10:10 RT:1:1
        Cost:pre-bestpath:128:2323456 (default-2145160191) 0x8800:32768:0
        0x8801:42:665600 0x8802:65282:1657856 0x8803:65281:1500,
      mpls labels in/out 16/nolabel
```

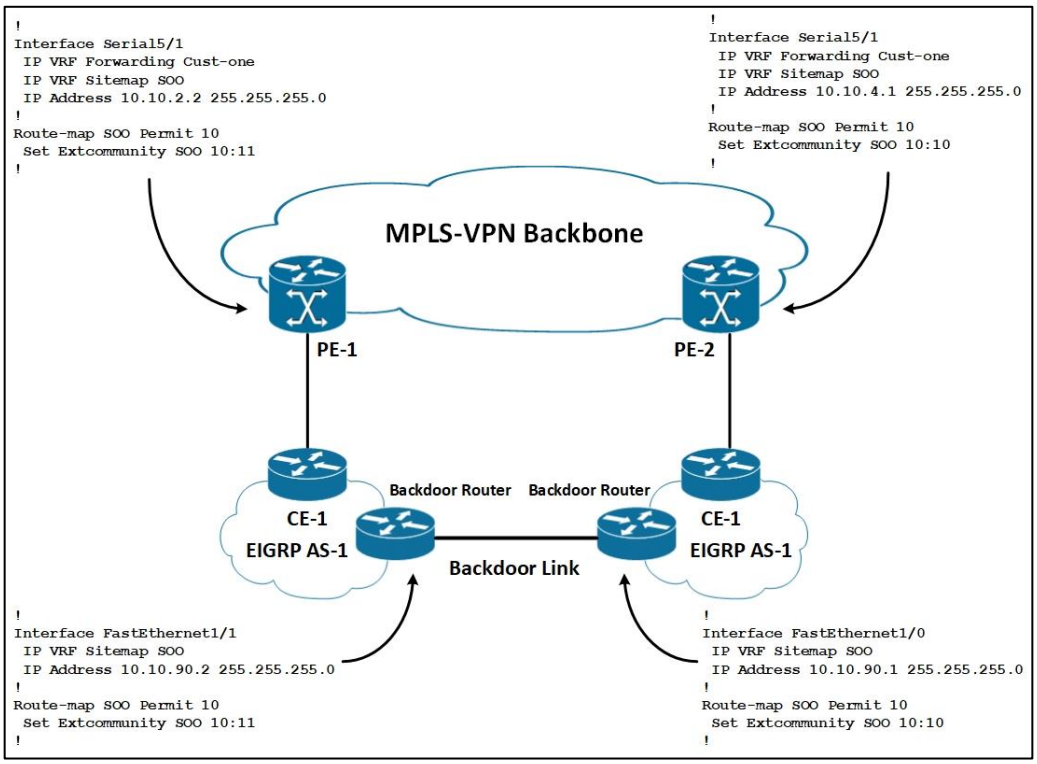
Fuente: elaboración propia, empleando Visio 2013.

3.5.95. EIGRP pe-ce con enlaces *backdoor*

Los enlaces backdoor son compatibles entre los sitios EIGRP que están conectados a la red troncal MPLS VPN. Sin embargo, cuando una ruta desaparece, el enrutamiento puede demorar más en *reconverger*, lo que es típico en el caso de la redistribución entre protocolos de enrutamiento. La causa de la convergencia más larga en la redistribución entre EIGRP y BGP. Para ayudar a acelerar la reconvergencia, puede usar el sitio de origen (*Site-of-origin*, SOO) para EIGRP. Se puede definir en los routers PE en las interfaces VRF hacia los routers CE y en los routers con un enlace de *backdoor*. Necesita configurar ip vfr sitemap en la interfaz, configurando la comunidad extendida SOO. Este mapa de ruta establece el SOO en la ruta EIGRP, ya sea en el PE o en el router de enlace de *backdoor*. Cuando el router recibe una ruta a través de la interfaz con está mapa de ruta configurado y el SOO de la ruta coincide con el SOO configurado, el *router* rechaza la ruta. Cuando el *router* PE recibe una actualización vpnv4 con el conjunto SOO, extrae el SOO y lo agrega a la ruta EIGRP cuando se reconstruye.

La figura 153 se muestra una red con un enlace de *backdoor* entre los sitios EIGRP y EIGRP SOO configurado en los *routers* PE y *backdoor*.

Figura 153. **Enlace *Backdoor* entre sitios EIGRP**



Fuente: elaboración propia, empleando Visio 2013.

Cuando no se usa SOO para EIGRP en ninguna parte, puede existir un problema de conteo infinito en todos los sitios de EIGRP y en toda la red troncal MPLS VPN. Esto significa que cuando una ruta desaparece, los *routers* EIGRP ven el conteo de saltos que aumenta lentamente hasta el infinito. Con EIGRP ven que el conteo de saltos aumenta lentamente hasta el infinito. Con EIGRP, el infinito es un conteo de saltos de 100 por defecto. Eso significa que puede tomar bastante tiempo para que la ruta desaparezca, mientras tanto, el tráfico se enlaza. Puede reducir el conteo de saltos máximo predeterminado de EIGRP configurado el comando métrico de saltos llamado *maximum-hops*. Sin embargo, debe tener cuidado de no configurar este valor demasiado bajo. El

valor debe ser suficientemente grande para la operación normal, pero también en el caso de la ruta más corta no está disponible y una ruta más larga que enrute el tráfico.

La desventaja de usar SOO para EIGRP en los **routers** PE y *backdoor* es que una parte del sitio no puede llegar a la otra parte del sitio a través del enlace de *backdoor* y la red troncal MPLS VPN si el sitio está dividido. El router de *backdoor* o el router PE bloquean la ruta necesaria para llegar a la otra parte del sitio. Para evitar este problema, se puede configurar el mapa del sitio para SOO solo en los routers PE y no en los *routers* de *backdoor*. El problema de contar hasta el infinito no ocurre en este caso, pero el router puede demorar un poco más en reconvocar. En la figura 154 se muestra SOO para una ruta EIGRP.

Figura 154. **Set de SOO para una ruta EIGRP**

```
PE-1#show ip eigrp vrf cust-one topology 10.10.100.3 255.255.255.255
IP-EIGRP (AS 42): Topology entry for 10.10.100.3/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
  10.200.254.5, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2297856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      Extended Community: SoO:10:10
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.96. eBGP

eBGP puede ser el protocolo de enrutamiento PE-CE. En la dirección de la familia ipv4 vrf del proceso bgp del router en el PE, debe configurar el router CE

como el vecino eBGP y activarlo. En la figura 155 se configura el vecino eBGP 10.20.2.1 (el router CE) en el sistema autónomo 65001 en VRF cust-one.

Figura 155. **Configuración básica de BGP como protocolo de enrutamiento PE-CE**

```
!
router bgp 1
 neighbor 10.200.254.5 remote-as 1
 neighbor 10.200.254.5 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.200.254.5 activate
 neighbor 10.200.254.5 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf cust-one
 redistribute connected
 neighbor 10.10.2.1 remote-as 65001
 neighbor 10.10.2.1 activate
 exit-address-family
!
```

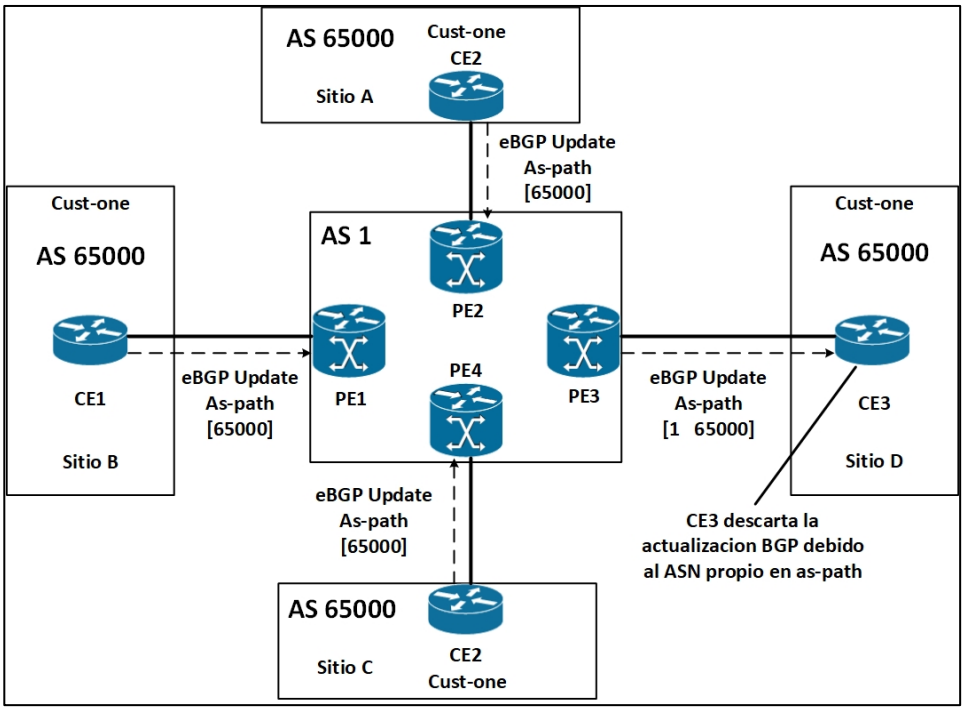
Fuente: elaboración propia, empleando Visio 2013.

Si los sitios del cliente tienen diferentes números de sistema autónomo para cada sitio, BGP puede operar con el comportamiento predeterminado con respecto al *as-path*. Sin embargo, a veces el comportamiento predeterminado no es suficiente para tener el enrutamiento correcto para el cliente en el VRF. En dos escenarios, BGP debe adaptarse para obtener la ruta correcta: clientes que tienen el mismo número de sistema autónomo (ASN) en más de un sitio y una situación de *hub-and-spoke*.

3.5.97. Anulación del sistema autónomo

Si el cliente tiene el mismo ASN en diferentes sitios, los *routers* CE eliminan las rutas BGP. Obsérvese la figura 156, donde todos los sitios de clientes de VPN *cust-one* tienen Sistema Autónomo 65000.

Figura 156. **Uso de AS-Override**



Fuente: elaboración propia, empleando Visio 2013.

Cada *router* PE enviar las rutas VRF *cust-one* con un *as-path* de [65000] en la actualización vpnv4 a los otros routers PE. La ruta BGP que se envía al router CE remoto es la ruta vpnv4 que se convierte en una ruta IPv4 cuando se extrae el RD. La ruta se envía al *router* CE a través de eBGP con un *as-path* de [1 65000]. El *router* CE descarta la actualización de BGP porque ve que su

propio ASN 65000 está en la actualización. Este comportamiento es el comportamiento predeterminado de BGP y es un mecanismo de prevención contra los bucles en BGP. Esto significa que si el cliente tenía su propia red privada (con solo 1 número de sistema autónomo) antes de usar el servicio MPLS VPN del proveedor del servicio, ahora se tendría que utilizar diferentes números de sistemas autónomo para cada sitio. Esto se vuelve una tarea tediosa y es casi imposible obtener nuevos números de sistemas autónomos. El cliente puede usar ASN del rango ASN privado [64512-65535]. Sin embargo, hay una solución más fácil disponible e implica que el router PE envía el prefijo BGP al *router* CE remoto con el *as-path* [1 1] en lugar de [1 65000]. El *router* PE simplemente verifica el ASN del *router* CE con respecto a los ASN en el as-path se reemplazan con el ASN del proveedor del servicio. El CE remoto acepta está ruta porque ya no ve su propio ASN en el *as-path* de la ruta BGP.

El comando que necesita configurar en el router PE para anular en el ASN es *ipaddress as-override*. La protección contra posibles bucles de enrutamiento y enrutamiento subóptimo que proviene de la verificación de camino se ha ido. Por lo tanto, cuando se utiliza la funcionalidad de anulación, es aconsejable implementar la función 'SOO' para BGP.

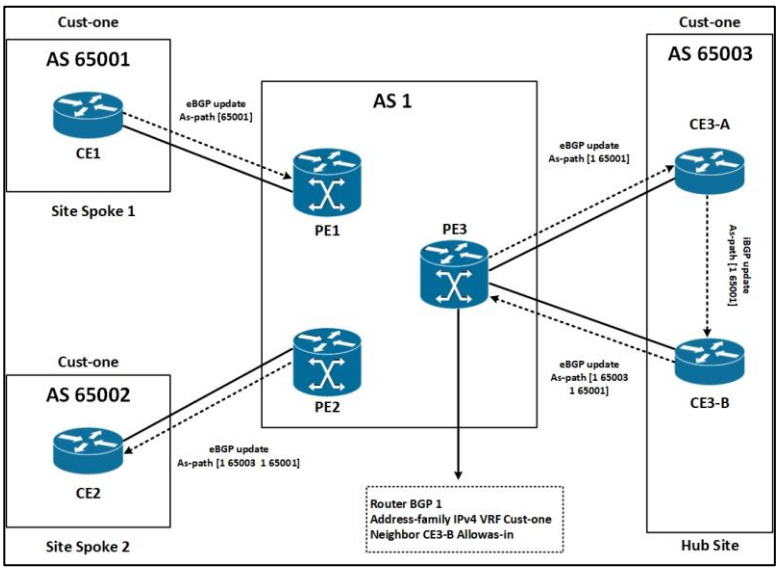
3.5.98. *Allowas-in*

Puede realizar otro truco con el *as-path* en BGP. En lugar de anular los números de sistema autónomo en el *as-path*, puede indicar al router PE que afloje la comprobación del *as-path*. En la siguiente sección, puede leer sobre el escenario de *hub-and-spoke*. Está escenario debe permitir que las rutas que provienen del sitio del hub VRF vuelvan a ingresar al sistema autónomo del proveedor de servicios. Esto asegura que la comunicación de *spoke-to-spoke* ocurra a través del sitio del hub de VRF. Para BGP, esto implica que una ruta

atraviesa el proveedor de servicios AS desde un sitio de radio VRF al sitio del centro de VRF y lo cruza de nuevo en el camino a otro sitio de radio VRF. El *router* PE que se conecta al sitio del hub VRF ve su propio ASN en el *as-path*, por lo que se rechaza la ruta BGP. Para evitar este problema, puede configurar el número de permisos de acceso a vecinos en el router PE que se conecta al sitio del hub VRF. El comando *allow-in* permite múltiples apariciones del mismo ASN (en este caso, el ASN del proveedor del servicio) en el *as-path* como el ASN del anunciante BGP sin que BGP niegue la ruta. El número que puede configurar es de 1 a 10, especificando la cantidad de veces que permite el ASN en el as-path.

En la figura 157 se muestra un escenario de *hub-and-spoke* con BGP como el protocolo de enrutamiento PE-CE en los sitios del cliente.

Figura 157. ***Hub-and-spoke* con BGP como protocolo de enrutamiento PE-CE**



Fuente: elaboración propia, empleando Visio 2013.

Cuando necesite anunciar una ruta de *spoke to spoke*, se anuncia primero en el sitio del hub. Cuando la ruta llegue al sitio de radio 2, habrá atravesado la red troncal MPLS VPN dos veces. Esto es solo posible si el neighbor *allowas-in* está configurado en el router PE3 hacia el router CE3-B. PE3 simplemente ignora ver su propio ASN en el as-path y acepta la ruta BGP desde el router CE CE3-B.

3.5.99. *Hub-and-spoke*

A menudo, los clientes no quieren que sus sitios tengan una interconectividad total. Esto significa que no quieren o no necesitan que los sitios estén completamente conectados. Un escenario típico involucra un sitio principal en una compañía con muchos sitios remotos. Un escenario típico involucra un sitio principal en una compañía con muchos sitios remotos. Los sitios o radios remotos necesitan conectividad con el sitio principal o concentrador, pero no necesitan comunicarse directamente entre ellos. Tal vez la conectividad sea posible pero no deseada por razones de seguridad. Este escenario se conoce comúnmente como el escenario hub-and-spoke. También se puede lograr a través de MPLS VPN, pero se debe tener cuidado. Lo siguiente es necesario:

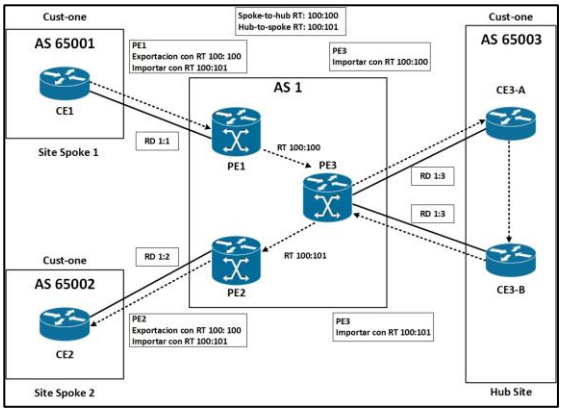
- Los *spoke site* pueden comunicarse solo con el sitio del hub
- El tráfico de *spoke-to-spoke* debe enviarse primero al sitio del *hub*
- Dos RTs diferentes
- RD diferentes

Primero, se eligen los RTs cuidadosamente. Necesita dos RT diferentes: una está conectada a las rutas de radios cuando el sitio de radios las envía al sitio del *hub*, y una está conectada a las rutas del concentrador cuando el sitio

de radio envía las rutas de radio a los otros sitios de radio, las rutas deben rechazarse porque el VRF en los sitios de radio no importa los RT. Sin embargo, el mismo *router* PE podría usarlos para otro sitio, o el filtrado de entrada automático podría desactivarse si se trata de un *router* ASBR en el caso de VPN MPLS inter-autónoma. En esos casos, el *router* PE acepta la ruta, pero el VRF no importa la ruta porque el RT no está configurado para la importación.

Sin embargo, podría seleccionarse como la mejor ruta vpnv4, lo que significa que el PE no importará la otra ruta vpnv4 con la RT correcta. Por lo tanto, se recomienda encarecidamente que utilice un RD único por sitio radial. Puede salirse con la suya usando el mismo RD para todos los *spoke site* la mayor parte del tiempo, pero esto no funciona en todos los escenarios. Es posible que tenga dos routers CE de radios conectados al mismo router PE. En ese caso, lo único que impide que las rutas de los radios se envíen directamente al otro router de radios de radio de frecuencia que está conectado al mismo *router* de PE es la RD diferente para cada *router* de radios con radios.

Figura 158. **Escenario *hub-and-spoke***



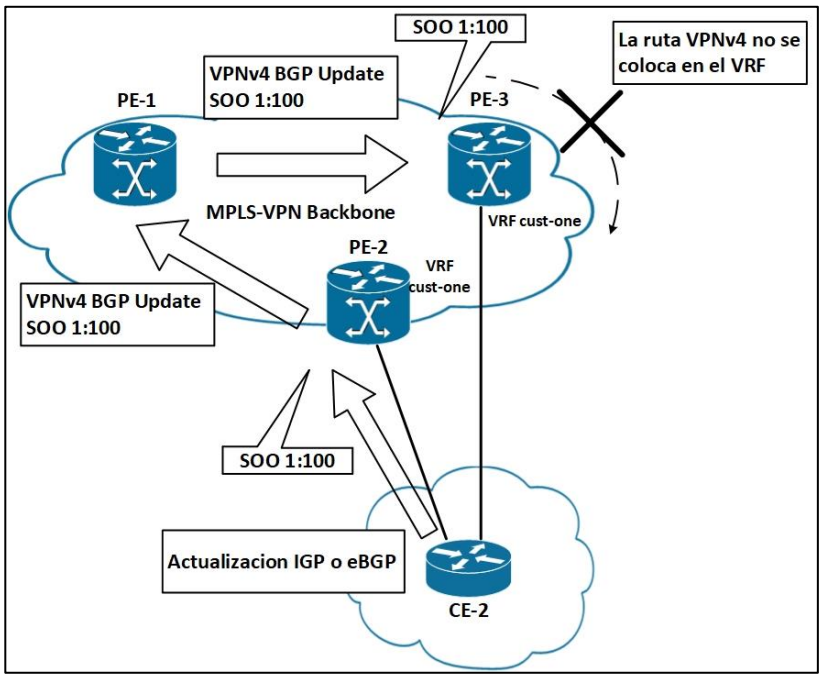
Fuente: elaboración propia, empleando Visio 2013.

Los *routers* PE de *spoke* anuncian las rutas con RT = 100: 100, mientras que el router PE del centro importa está RT. El *router* PE del hub anuncia las rutas desde el router CE central con RT = 100: 101. Los *routers* de PE de *spoke* importan RT 100: 101, lo que garantiza que el tráfico de habla a voz pase por el sitio del hub.

3.5.100. SOO

SOO identifica de manera única el sitio que origina una ruta. Es una comunidad extendida de BGP que evita los bucles de enrutamiento o el enrutamiento subóptimo, específicamente cuando hay una *backdoor* entre los sitios de VPN. SOO proporciona prevención de bucles en redes con sitios de doble conexión (sitios que están conectados a dos o más *routers* PE). Puede usarlo cuando un IGP es el protocolo de enrutamiento PE-CE. También, puede usarlo cuando se utiliza BGP entre PE y CE, cuando ya no se puede confiar en la prevención de bucle as-path. Esto sucede cuando BGP usa *as-override* o *allowas-in*. Si el SOO está configurado para un *router* CE y se aprende una ruta vpnv4 con el mismo SOO, la ruta no debe colocarse en la tabla de enrutamiento VRF en el PE y anunciarse al CE. En la figura 159, el prefijo vpnv4 se anuncia desde PE-2 hacia los otros *routers* PE con SOO 1: 100. Está ruta vpnv4 se puede publicitar en el mismo sitio y se puede recibir en PE-3 a través de MP-BGP. Cuando PE-3 nota el mismo SOO en la ruta vpnv4 que el SOO en la configuración, no instala el prefijo en la tabla de enrutamiento VRF.

Figura 159. **SOO previniendo bucles de enrutamiento**



Fuente: elaboración propia, empleando Visio 2013.

Esto evita posibles bucles de enrutamiento, pero también evita el enrutamiento subóptimo. El caso subóptimo que implica que para las rutas locales a los sitios de doble origen, la ruta a través de la red troncal MPLS VPN es preferible a la ruta local está bloqueada.

SOO se establece en un mapa de ruta, como se muestra en la figura 160.

Figura 160. **Configuración SOO de mapas de ruta**

```
!
route-map cust-one-soo permit 10
 set extcommunity soo 1:100
!
```

Fuente: elaboración propia, empleando Visio 2013.

Si se aplica SOO para BGP, el mapa de ruta se configura en el comando vecino BGP, como en la figura 161.

Figura 161. **Aplicando mapas de ruta SOO para BGP**

```
!
router bgp 1
…
!
 address-family ipv4 vrf cust-one
 redistribute connected
 neighbor 10.10.2.1 remote-as 65001
 neighbor 10.10.2.1 activate
 neighbor 10.10.2.1 route-map cust-one-soo in
 exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

Si el SOO se aplica a cualquier protocolo de enrutamiento que no sea BGP, el mapa de ruta se configura con el comando ip vrf *sitemap* en la interfaz VRF apropiada, como en la figura 162.

Figura 162. **Aplicando mapas de ruta SOO sobre las interfaces VRF**

```
!
interface Ethernet0/1/2
 ip vrf forwarding cust-one
 ip vrf sitemap cust-one-soo
 ip address 10.10.2.2 255.255.255.0
!
```

Fuente: elaboración propia, empleando Visio 2013.

También puede configurar el SOO para las rutas estáticas y conectadas cuando se redistribuyen en el IGP. La figura 163 muestra el comando redistribuir con un mapa de ruta SOO.

Aplicación del mapa de ruta de SOO para rutas estáticas (continuación).

Figura 163. **Aplicando mapas de ruta SOO en rutas estáticas**

```
!
router bgp 1
…
!
 address-family ipv4 vrf cust-one
 redistribute static route-map cust-one-soo
 neighbor 10.10.2.1 remote-as 65001
 neighbor 10.10.2.1 activate
 exit-address-family
!
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.101. Acceso VRF

En el *router* PE, los comandos de Cisco IOS se hicieron conscientes de VRF para que el usuario pudiera comunicarse con los dispositivos CE o las direcciones IP en el router PE en el contexto VRF. Los comandos *ping*,

traceroute y telnet se han hecho conscientes de VRF para la resolución de problemas y para el acceso a los routers CE y otros dispositivos en los sitios VRF desde el router PE. En la figura 164 muestra cómo se extendieron estos tres comandos para el contexto VRF.

Figura 164. **Comandos VRF *ping*, *traceroute* y *telnet***

```
paris#ping vrf cust-one 10.10.100.1

paris#traceroute vrf cust-one 10.10.100.1

paris#telnet 10.10.100.1 /vrf cust-one
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.102. Acceso a internet

El enrutamiento de internet generalmente se realiza a través de la tabla BGP de la red MPLS VPN del proveedor de servicio. Está tabla BGP se encuentra en el espacio de enrutamiento global, no en el contexto VRF. Por defecto, los sitios VRF pueden comunicarse solo con otros sitios VRF en la misma VPN, no con nada en el espacio de enrutamiento global. Por lo tanto, se debe hacer algo para proporcionar acceso a internet (contexto global) a los *routers* CE (contexto VRF). Las siguientes secciones detallan como proporcionar acceso a internet a los sitios de VRF. Obviamente, el acceso a internet solo es posible para las subredes IP del cliente que no provienen del espacio de direcciones IP privadas (RFC 1918).

Tan pronto como la VPN tenga conectividad a internet, existe el riesgo de seguridad potencial. Es importante tomar las medidas adecuadas, como el filtrado y el uso de firewall, para garantizar el más alto nivel de seguridad.

3.5.103. Internet en una VPN

Una solución que podría parecer la más simple es la peor. El proveedor del servicio podría colocar la tabla completa de enrutamiento de internet en el VRF. Sin embargo, eso significaba que se colocaría una enorme cantidad de rutas en la VPN. El proveedor podría hacer él esto una vez y poner a todos los clientes que requieran acceso a internet en está VRF. Sin embargo, entonces el punto de cada cliente que tiene su propia red privada se perdería por completo. Otra solución podría implicar que el proveedor del servicio ponga la tabla de enrutamiento de Internet en cada VRF de un cliente que requiera acceso a internet. Sin embargo, eso sería aún peor. La enorme cantidad de rutas de internet se replicará varias veces y causaría problemas de escala en los routers PE. Por lo tanto, se debe evitar esta solución.

3.5.104. Acceso a internet a través de la tabla de enrutamiento global

Una manera fácil de proporcionar acceso a internet a *routers* CE es tener una interfaz desde el *router* PE al *router* CE que se encuentra en el espacio de enrutamiento global. El *router* PE tiene una interfaz VRF hacia el *router* CE, pero puede tener una segunda interfaz que no está un VRF hacia el *router* CE. El enrutamiento en el *router* CE debe encargarse de enviar el tráfico VPN a la interfaz VRF y el tráfico de internet a la interfaz en el espacio de enrutamiento global en el router PE. La desventaja obvia es que necesita un segundo enlace entre los routers PE y CE, utilizando una interfaz adicional en ambos routers. Para resolver esto, puede usar subinterfaces cuando la encapsulación de capa 2 es Frame Relay o encapsulación 802.1Q. Sin embargo, si la encapsulación de capa 2 no permite subinterfaces, puede utilizar una solución alternativa. Una solución posible podría ser quedarse con solo la interfaz VRF en el router PE y

la creación de un túnel GRE en el espacio de enrutamiento global a través de esa interfaz VRF.

En la figura 165 se muestra una configuración de muestra para esta solución. La ruta predeterminada en el *router* CE apunta a la interfaz del túnel para el acceso a internet. Por lo tanto, todo el tráfico que no tiene una ruta específica se envía a la interfaz del túnel de acuerdo con la ruta predeterminada. Está tráfico termina en el contexto de enrutamiento global del router PE. Todo el tráfico que tiene una ruta específica en la tabla de enrutamiento del CE se envía a la interfaz física y termina dentro del VRF en el router PE. El tráfico de internet a los *routers* del cliente se reenvía según una ruta estática en el *router* PE que apunta a la interfaz de túnel. En el ejemplo, la ruta estática para 192.168.1.0/24 se encarga de devolver el tráfico de internet en la red troncal hacia y desde las puertas de enlace de internet con BGP. La interfaz del túnel en el router PE necesita el comando *tunnel vrf vrf-name* porque el punto final del túnel no está en el espacio de enrutamiento global, pero en el VRF especificando. Como el túnel no tiene el comando *ip vrf forwarding vrf-name*, se encuentra en el espacio de enrutamiento global.

Figura 165. **Configuración del túnel GRE en el espacio de enrutamiento global en el PE**

```
paris#
!
interface Tunnel1
 ip address 10.10.20.1 255.255.255.0
 tunnel source 10.10.2.2
 tunnel destination 10.10.2.1
 tunnel vrf cust-one
!
interface Ethernet0/1/2
 ip vrf forwarding cust-one
 ip address 10.10.2.2 255.255.255.0
!
ip route 192.168.1.0 255.255.255.0 Tunnel1


paris-ce#
!
interface Tunnel1
 ip address 10.10.20.2 255.255.255.0
 tunnel source 10.10.2.1
 tunnel destination 10.10.2.2
!
ip route 0.0.0.0 0.0.0.0 Tunnel1
```

Fuente: elaboración propia, empleando Visio 2013.

3.5.105. Acceso a internet a través de la tabla de enrutamiento global con rutas estáticas

Puede proporcionar acceso a internet a los clientes VPN reenviando su tráfico a la puerta de enlace de Internet del proveedor de servicio. Todos los routers P de la red VPN MPLS conocen la puerta de enlace de internet porque la dirección IP de la puerta de enlace se conoce en la tabla de enrutamiento global del proveedor de servicios. Seguramente está ejecutando eBGP con un router de un proveedor de internet. Los *routers* PE ya están ejecutando BGP, por lo que pueden proporcionar servicios MPLS VPN. Los *routers* PE también pueden ejecutar una sesión de interconexión iBGP para IPv4 en el *router* de puerta de enlace de internet. Para proporcionar acceso a internet a un VRF, la tabla de enrutamiento global debe reenviar el tráfico. Esto ocurre al crear

estática en la tabla VRF en el *router* PE y el especificar un próximo salto que se encuentra en la tabla VRF en el *router* PE y al especificar un próximo salto que se encuentra en la tabla de enrutamiento global. Para hacer esto, use la palabra clave global en la ruta VRF estática. Esto garantiza que el tráfico que fluye desde el *router* CE al *router* PE a través de la interfaz VRF y que se reenvía según la ruta estática se reenvía al siguiente salto debe estar en el router de la puerta de enlace de internet. Necesita reenviar al VRF el tráfico que fluye de internet. Configurar una ruta estática en el *router* PE y especificar que el siguiente salto sea el *router* CE lo logra. Para garantizar que la puerta de enlace de internet conozca esta ruta, distribuya la ruta estática en BGP o IGP del proveedor de servicios. Como el tráfico ya no es de VPN a VPN, sino que se reenvía a la tabla de enrutamiento global, solo tiene una etiqueta en la red MPLS VPN.

Se observa en la figura 166 para la configuración en el *router* Paris PE donde la ruta estática se distribuye en BGP. El *router* de puerta de enlace de internet es 10.200.254.5 y 192.168.1.0/24 es la subred del cliente que necesita acceso a internet. Todo el tráfico que no tiene una ruta específica en la tabla de enrutamiento de VRF *cust-one* se reenvía de acuerdo con la ruta predeterminada en el VRF con el 10.200.254.5 del próximo salto en la tabla de enrutamiento global. El tráfico de internet hacia el router Paris-CE se reenvía según la ruta estática para 192.168.1.0/24 que apunta a la interfaz Ethernet 0/1/2 en el router PE hacia el *router* CE.

Figura 166. **Acceso a internet a través de la tabla de enrutamiento global con rutas estáticas**

```
paris#
!
interface Ethernet0/1/2
 ip vrf forwarding cust-one
 ip address 10.10.2.2 255.255.255.0
!
router bgp 1
 bgp log-neighbor-changes
 redistribute static
 neighbor 10.200.254.3 remote-as 1
 no auto-summary
 !
ip route vrf cust-one 0.0.0.0 0.0.0.0 10.200.254.5 global
ip route 192.168.1.0 255.255.255.0 Ethernet0/1/2 10.10.2.1
!
paris-ce#show ip route 0.0.0.0 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "rip", distance 120, metric 2, candidate default path
  Redistributing via rip
  Last update from 10.10.2.2 on Ethernet1/1, 00:00:14 ago
  Routing Descriptor Blocks:
  * 10.10.2.2, from 10.10.2.2, 00:00:14 ago, via Ethernet1/1
      Route metric is 2, traffic share count is 1
```

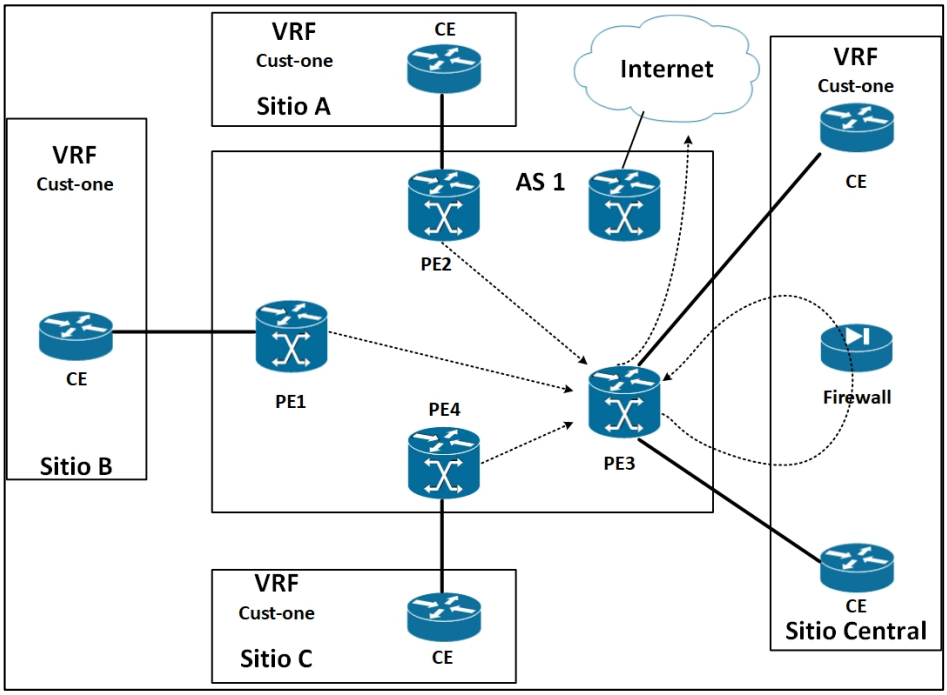
Fuente: elaboración propia, empleando Visio 2013.

3.5.106. Acceso a internet a través de un sitio central VRF

En lugar de reenviar el tráfico de cada sitio VPN directamente al router de la puerta de enlace de internet, es posible reenviar todo el tráfico de internet de los sitios VRF a los routers CE de un sitio VRF central en una VPN. La ventaja es que las funciones de seguridad, como los servicios de *firewall* u otros servicios, como la traducción de direcciones de red (NAT), se implementan solo una vez y centralmente en el sitio central de VRF. El tráfico de internet entre los sitios de VRF y el sitio central de VRF se reenvía a través de las interfaces de VRF normales de la manera normal para MPLS VPN. Se observa la figura 167 para la red en está escenario. Es muy probable que está sea el escenario preferido para redes VPN *hub-and-spoke* de todos modos. Tenga en cuenta

que el sitio VRF central, puede implementar un *firewall* para verificar todo el tráfico de internet.

Figura 167. **Acceso a internet a través de un sitio VRF central**



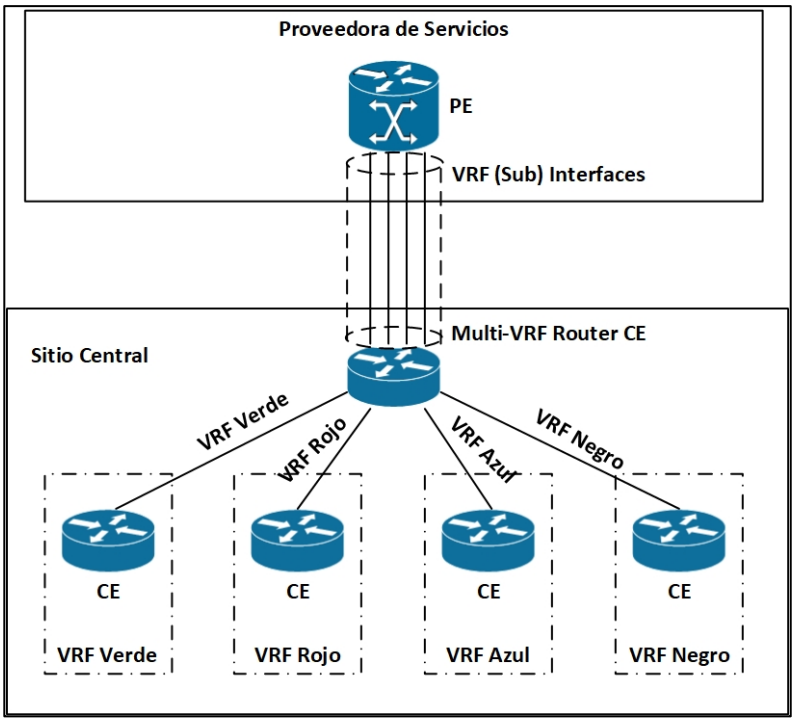
Fuente: elaboración propia, empleando Visio 2013.

3.5.107. Multi-VRF CE

La función Multi-VRF CE, también conocida como VRF-Lite, es una función mediante la cual la funcionalidad de VPN se extiende al *router* CE de forma económica. Supongamos que tiene una empresa con un sitio principal grande y algunos sitios más pequeños que están interconectados a través de una red MPLS VPN. El sitio principal de la empresa es bastante grande y tiene varios departamentos que deben separarse unos de otros por razones de

privacidad. Estos departamentos (finanzas, recursos humanos, ingeniería, entre otros.) se conectan con los sitios remotos de sus respectivos departamentos a través de la red MPLS VPN. Puede separar los departamentos implementando VLAN en los conmutadores en el sitio principal y asignando cada VLAN a una interfaz VRF (sub) en el *router* PE. En lugar de usar un conmutador de capa 2 o un router CE por departamento, puede llevar la funcionalidad de VPN al *router* CE. Para Multi-VRF CE, la separación en VRF se usa en el *router* CE, ya que se usa en el *router* PE. Sin embargo, el CE no necesita la otra funcionalidad MPLS VPN, como los paquetes de etiquetado. Multiprotocol iBGP y LDP. Las interfaces hacia el *router* PE son interfaces VRF. Debe configurar los VRF y los protocolos de enrutamiento VRF apropiados en el router Multi-VRF CE.

Figura 168. **Ejemplo de un CE Multi-VRF**



Fuente: elaboración propia, empleando Visio 2013.

El *router* Multi-VRF CE tiene una interfaz VRF para cada *router* CE que está conectado a él. Cada VRF hacia un router CE también debe tener una interfaz VRF hacia el *router* PE. Por supuesto, si usa una interfaz para cada VRF en el router PE y Multi-VRF CE, se vuelve costoso. Una solución mucho más barata es usar un enlace FastEthernet y GigabitEthernet o un enlace de serie canalizado entre el PE y el Multi-VRF CE y usar una subinterfaz para cada VRF.

No necesita ninguna funcionalidad MPLS VPN en el router Multi-VRF CE. La característica Multi-VRF CE está definida en el *router* CE y no en el router PE. El resto de la red VPN MPLS funciona de la misma manera que la VPN MPLS normal.

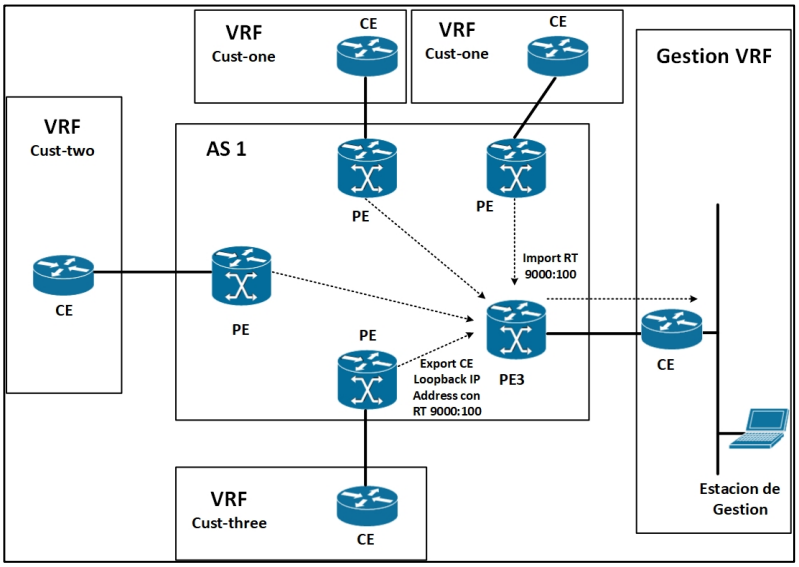
3.5.108. Gestión CE

A menudo, el proveedor del servicio, no el cliente, posee y administra el router CE. En esta situación, el proveedor del servicio desea acceso de gestión al router CE desde un servidor de administración central. Puede hacerlo haciendo que el router PE anuncie un prefijo del router CE gestionando con RT que se importa en el VRF de gestión mediante el router PE conectado al VRF de gestión.

Puede limitar el número de prefijos anunciados con esta RT de gestión configurando un mapa de exportación de cada VRF que asigna RT de gestión a solo un prefijo en el *router* CE. También puede anunciar los RT VRF habituales utilizados por la VPN con este prefijo si los otros *routers* CE necesitan poder alcanzarlo. En la figura 169 se muestra una descripción general de la configuración de gestión. La administración VRF tiene una estación de administración. El router PE con el VRF de administración está importando

todas las rutas con el RT 9000: 100. El *router* PE Sydney establece el RT de un prefijo en el router CE (aquí en el prefijo 10.10.100.3/32; el prefijo de *loopback* en el *router* CE) en 9000: 100.

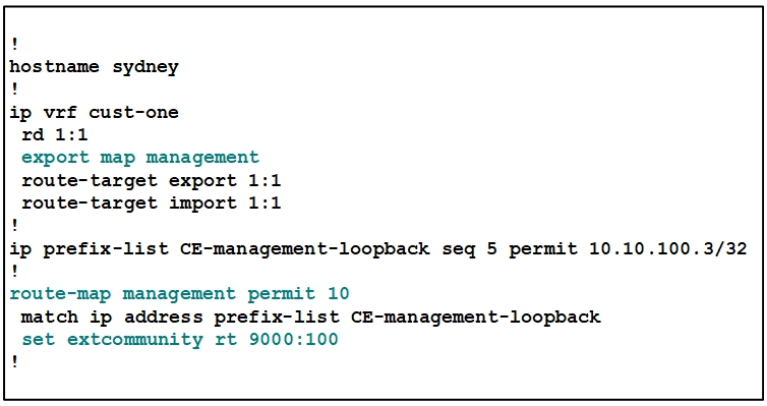
Figura 169. **Ejemplo de acceso de gestión**



Fuente: elaboración propia, empleando Visio 2013.

La configuración de un *router* PE que proporciona acceso de administración al CE se muestra en la figura 170.

Figura 170. **Configuración de un *router* PE que proporciona acceso de administración**



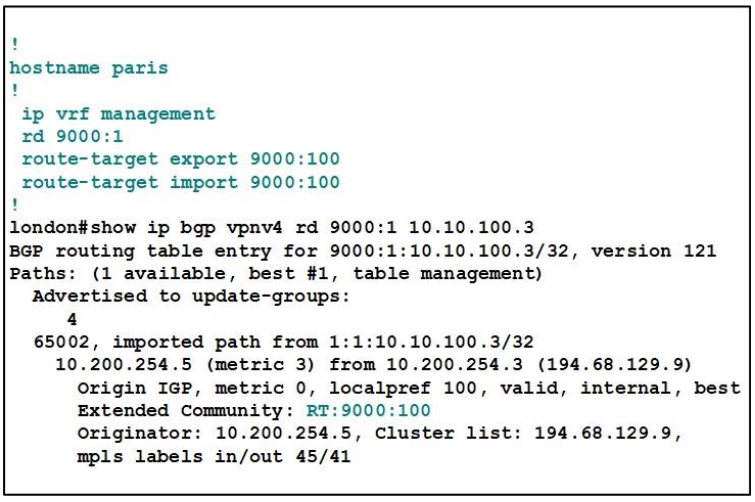
```
!
hostname sydney
!
ip vrf cust-one
 rd 1:1
 export map management
 route-target export 1:1
 route-target import 1:1
!
ip prefix-list CE-management-loopback seq 5 permit 10.10.100.3/32
!
route-map management permit 10
 match ip address prefix-list CE-management-loopback
 set extcommunity rt 9000:100
!
```

Fuente: elaboración propia, empleando Visio 2013

.

La configuración del *router* PE con el VRF de administración adjunto se muestra en la figura 171.

Figura 171. **Gestión de la configuración del *router* PE**



```
!
hostname paris
!
 ip vrf management
 rd 9000:1
 route-target export 9000:100
 route-target import 9000:100
!
london#show ip bgp vpnv4 rd 9000:1 10.10.100.3
BGP routing table entry for 9000:1:10.10.100.3/32, version 121
Paths: (1 available, best #1, table management)
  Advertised to update-groups:
     4
  65002, imported path from 1:1:10.10.100.3/32
    10.200.254.5 (metric 3) from 10.200.254.3 (194.68.129.9)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:9000:100
      Originator: 10.200.254.5, Cluster list: 194.68.129.9,
      mpls labels in/out 45/41
```

Fuente: elaboración propia, empleando Visio 2013.

Cada vez más proveedores de servicios interconectan sus *backbones* MPLS VPN. Se puede hacer esto de dos maneras:

- VPN MPLS inter-autónoma
- CsC

Con MPLS VPN inter-autónoma, las redes MPLS VPN se relacionan entre sí e intercambian los prefijos de los clientes que tienen sitios conectados a cada uno de los proveedores de servicios. Los proveedores de servicios deben ofrecer la conectividad entre los sitios de los clientes, incluso cuando no están conectados a una sola red troncal MPLS VPN.

CsC es una solución mediante la cual un operador más grande proporciona servicios MPLS VPN a otros operadores o proveedores de servicios. El servicio es de naturaleza jerárquica, mientras que la VPN MPLS inter-autónoma es simplemente una interconexión entre redes troncales MPLS VPN que intercambian prefijos de clientes.

3.6. VPN

Una red vitual privada o por sus siglas en ingles virtual private network, es una red en la cual se pueden realizar interconexiones seguras en redes publicas. La característica mas importante que posee este tipo de conexión es que realiza un cifrado de extremo a extremo, por lo que es una conexión muy segura.

3.6.1. Fundamentos

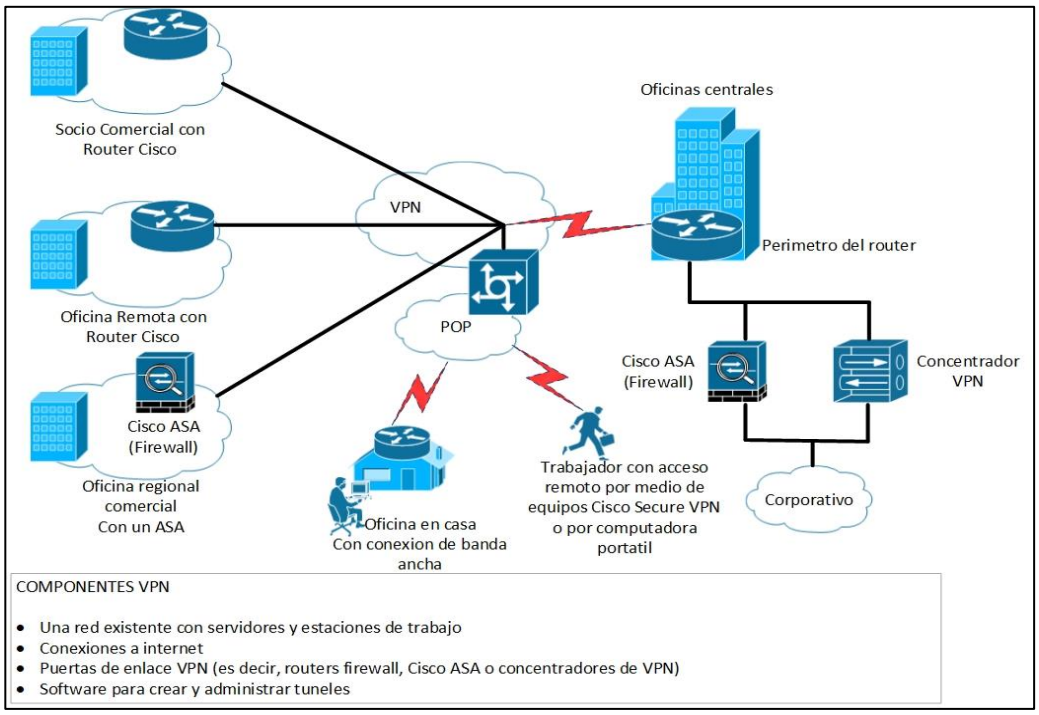
Internet es una red de IP accesible a nivel mundial. Debido a su gran proliferación global, se ha convertido en un método viable de interconexión de sitios remotos. Sin embargo, el hecho de que sea una infraestructura pública ha disuadido a la mayoría de las empresas de adoptarlo como un método viable de acceso remoto para sucursales.

Una red privada virtual (VPN) es un concepto que describe cómo crear una red privada a través de una infraestructura de red pública mientras se mantiene la confidencialidad y la seguridad. Las VPN utilizan protocolos de túnel criptográfico para proporcionar autenticación del remitente, integridad del mensaje y confidencialidad mediante la protección contra el rastreo de paquetes. La VPN se pueden implementar en las capas 2, 3 y 4 del modelo de interconexión de sistemas abiertos (modelo OSI)

En la figura 172 se ilustra una topología de VPN típica. Los componentes necesarios para establecer una VPN incluyen:

- Una red existente con servidores y estaciones de trabajo.
- Conexión a internet.
- Puertas de enlace VPN (es decir, *routers*, ASA, concentradores VPN que actúan como puntos finales para establecer, administrar y controlar conexiones VPN.)
- Software para crear y gestionar túneles.

Figura 172. **Topología básica VPN**



Fuente: elaboración propia, empleando Visio 2013.

La clave de tecnología VPN es la seguridad. Las VPN protegen los datos encapsulando los datos, encriptando los datos, o encapsulando los datos y luego encriptando:

- La encapsulación también se conoce como *tunneling* porque la encapsulación transmite datos de forma transparente desde la red a la red a través de una infraestructura de red compartida.

- Cifrado de datos de códigos de un formato diferente. El descifrado decodifica los datos encriptados en el formato original descifrado de los datos.

3.6.2. Arquitectura VPN de superposición y de punto a punto

En términos de evolución, hay dos modelos principales de VPN: VPN superpuesto y VPN punto a punto.

3.6.3. VPN superpuestas

Los proveedores de servicios (*service provider*, SP) son los usuarios más comunes del modelo VPN superpuesto. El diseño y el aprovisionamiento de circuitos virtuales (*virtual circuits*, VC) en la red troncal se completan antes de cualquier flujo de tráfico. En caso de una red IP, esto significa que, aunque la tecnología subyacente no tiene conexión, requiere un enfoque orientado a la conexión para proporcionar el servicio.

Los problemas de escalado de las VPN supuestas presentan un desafío para los proveedores de servicios (*service provider*) cuando tienen que administrar y aprovisionar una gran cantidad de circuitos y túneles entre los dispositivos de los clientes. Desde el punto de vista del cliente, el diseño del protocolo interior también es complejo y difícil de administrar.

3.6.4. EL modelo de superposición incluye VPN L2 y L3

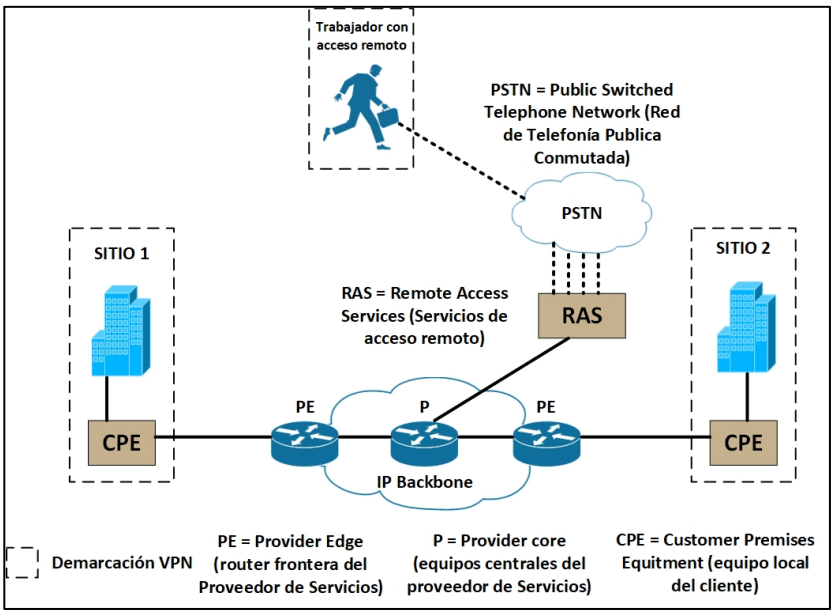
VPN superposición L2: las VPN superpuestas L2 son independientes del protocolo de red utilizado por el cliente, lo que significa que la VPN no se limita a transportar tráfico IP. Si el operador ofrece el servicio ATM apropiado, la VPN superpuesta llevará cualquier tipo de información. *Frame Relay* las VPN normalmente se limitan a aplicaciones de datos, aunque los dispositivos con equipos locales de voz (CPE) de voz sobre *Frame Relay* pueden ser utilizables en algunos servicios.

VPN de superposición L3: las VPN de superposición L3 suelen utilizar un esquema de túnel “IP en IP” utilizando protocolo de túnel de punto a punto (PPTP), el protocolo de túnel de capa 2 (L2TP) y la seguridad de (IPsec). La figura resume las propiedades básicas de estas tecnologías.

- VPN basada en CPE (punto a punto).

- VPN basada en CPE es otro nombre para un VPN superpuesta L3. La VPN se implementa utilizando CPE, como se muestra en la figura 173. De esta forma, un cliente crea una VPN a través de una conexión a internet sin ningún conocimiento específico o cooperación del proveedor del servicio. El cliente obtiene la ventaja de una mayor privacidad utilizando una conexión a internet económica.

Figura 173. **VPN de superposición basada en CPE**



Fuente: elaboración propia, empleando Visio 2013.

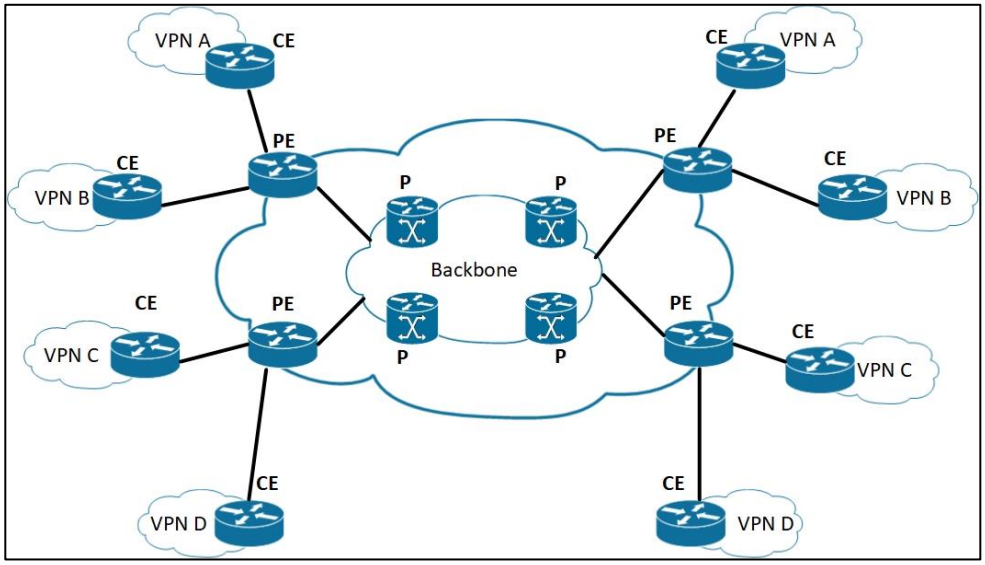
Este enfoque no es ventajoso para el proveedor de servicio porque hay pocas oportunidades para los ingresos del servicio VPN. Sin embargo, los SP cobran una tarifa más alta por los servicios de internet de 'clase ejecutiva' aplicables a las medianas y grandes empresas. Además, algunos proveedores de servicio ofrecen servicios de “VPN administrada” donde la configuración de CPE y la gestión de direcciones de traducción de direcciones de red (NAT) son realizadas por el proveedor de servicio en lugar de por el cliente.

3.6.5. VPN con aprovisionamiento del proveedor de servicio

La introducción de la conmutación de etiquetas multiprotocolo (MPLS) combina los beneficios de la superposición de VPN (seguridad y aislamiento entre los clientes) con los beneficios de enrutamiento simplificado de una VPN punto a punto. MPLS VPN proporciona un enrutamiento más simple para el cliente, un aprovisionamiento más simple del proveedor de servicios y una cantidad de topologías posibles que son difíciles de implementar en los modelos VPN de igual o de superposición. MPLS también agrega los beneficios de un enfoque orientado a la conexión al paradigma de enrutamiento IP, a través del establecimiento de rutas de conmutación de etiquetas que se crean en función de la información de topología en lugar del flujo de tráfico.

Este modelo usa tres tipos de routers como se muestra en la figura 174:

Figura 174. **Modelo provisional VPN de un proveedor de servicios**



Fuente: elaboración propia, empleando Visio 2013.

Se supone que los routers del proveedor (P) y el *router* frontera (*Client edge*, CE) desconocen los protocolos o procedimientos de VPN.

Solo los *routers* frontera del proveedor (PE) deben aprovisionarse para admitir las VPN.

Tomar en cuenta que las VPN MPLS no pueden reemplazar todas las implementaciones de VPN porque MPLS solo admite IP como el protocolo de capa 3.

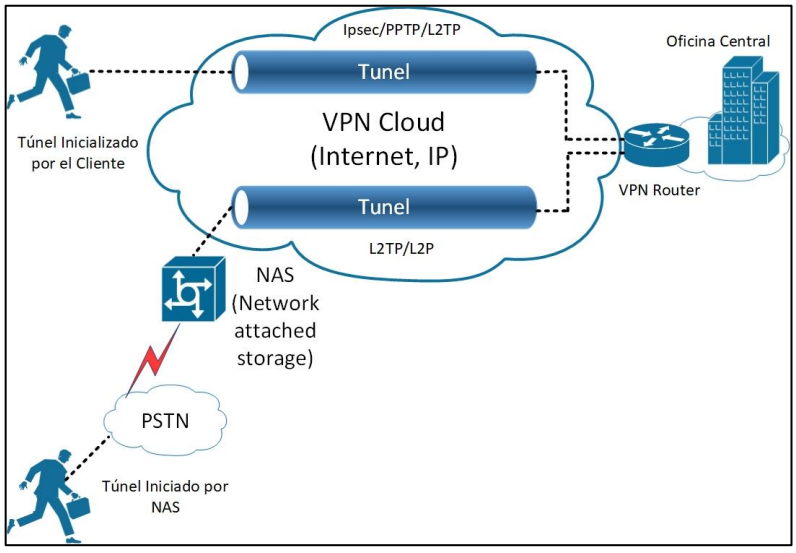
3.6.6. Topologías de VPN

Hay tres topologías VPN a considerar:

3.6.6.1. VPN de acceso remoto

Las VPN de acceso remoto brindan a los usuarios remotos acceso a una intranet o extranet a través de una infraestructura compartida. Los usuarios de dispositivos móviles, tele trabajadores y sucursales se pueden conectar de una forma segura mediante acceso telefónico, red digital de servicios integrados (RDSI), línea de suscriptor digital (DSL), IP móvil y tecnologías de cable. Las VPN de acceso remoto utilizan solo una única puerta de enlace VPN. La parte que negocia una conexión segura con VPN *gateway* utiliza el software de cliente VPN. El software VPN Client permite a los tele trabajadores y usuarios que viajan comunicarse en la red central y acceder a los servidores desde diferentes ubicaciones. Los túneles se crean usando IPsec, protocolo de túnel punto a punto (PPTP), protocolo de túnel de capa 2 (L2TP) o protocolo de envío de nivel 2 (L2F).

Figura 175. **Cliente inicializado en acceso remoto VPN**



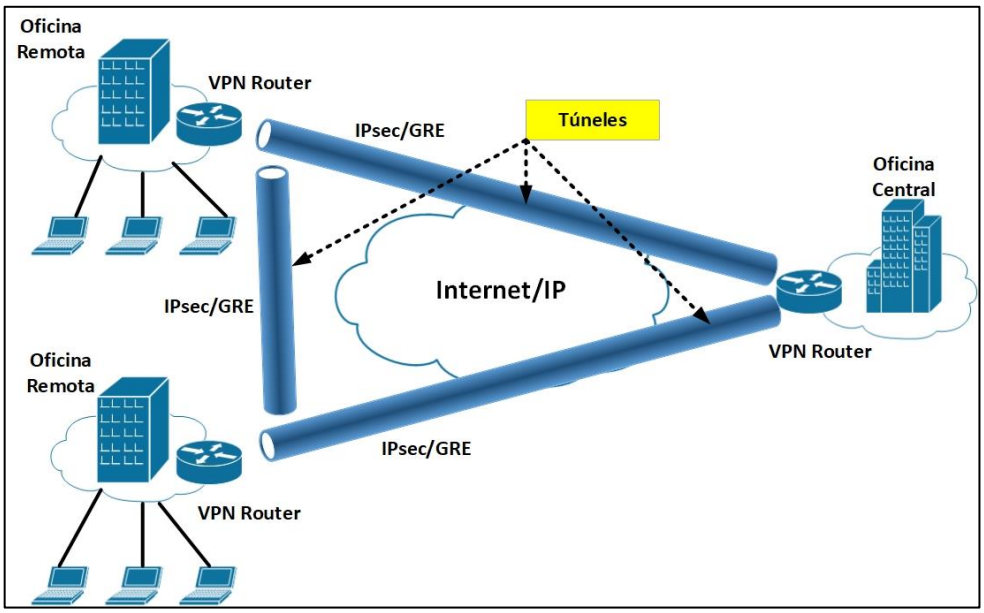
Fuente: elaboración propia, empleando Visio 2013.

- Beneficios: las VPN de acceso remoto reducen los cargos de larga distancia asociados con el acceso telefónico. Las VPN de acceso remoto también ayudan a aumentar la productividad y la confianza al garantizar el acceso seguro a la red independiente de la ubicación del empleado.

3.6.6.2. VPN de intranet de punto a punto

Las VPN de intranet de punto a punto enlazan la sede central, las oficinas remotas y las sucursales a una red interna a través de una infraestructura compartida mediante conexiones dedicadas. Las VPN de intranet difieren de las VPN de extranet en que las VPN de intranet solo permiten el acceso a empleados de confianza. Con una VPN de intranet, las puertas de enlace en varias ubicaciones físicas dentro de la misma empresa negocian túneles seguros a través de internet. Un ejemplo de este tipo, se observa en la figura 176, donde la VPN es una red que existe en varias ubicaciones geográficas, que se conecta a un centro de datos o mainframe que tiene acceso seguro a través de internet. Los usuarios de las redes en cualquier lado del túnel pueden comunicarse entre sí como si las redes fueran una sola red. Estas redes pueden necesitar cifrado fuerte y requisitos estrictos de rendimiento y ancho de banda. Los túneles se crean usando IPsec o IPsec/GRE.

Figura 176. **Intranet VPN de sitio a sitio**

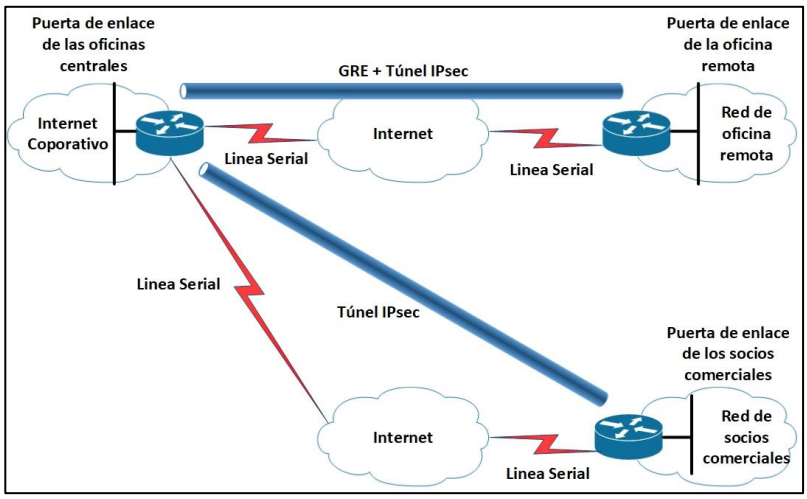


Fuente: elaboración propia, empleando Visio 2013.

- Beneficios: las VPN de intranet de sitio a sitio ofrecen ahorros de costos sobre las tecnologías tradicionales de línea arrendada o *Frame Relay*.

VPN extranet de punto a punto: una extranet VPN de punto a punto vincula a clientes, proveedores, socios o comunidades de interés para la red de un cliente empresarial a través de una infraestructura compartida mediante conexiones dedicadas. Las VPN de extranet difieren de las VPN de intranet en que las VPN de extranet permiten el acceso a los usuarios que se encuentran fuera de la empresa. Las VPN de extranet usan *firewalls* junto con túneles VPN para que los socios comerciales solo puedan obtener acceso seguro a datos y recursos específicos sin obtener acceso a información corporativa privada.

Figura 177. **Extranet VPN de sitio a sitio**



Fuente: elaboración propia, empleando Visio 2013.

Beneficios: las empresas disfrutan de las mismas políticas que una red privada, que incluye seguridad, calidad de servicio (QoS), capacidad de administración y confiabilidad.

3.6.6.3. Características de una VPN segura

La seguridad es el foco de cualquier diseño de VPN. Las VPN pueden utilizar técnicas avanzadas de cifrado y tunelización para establecer conexiones de red privadas, seguras y de extremo a extremo, a través de redes de terceros, como internet o extranets. La base de las VPN seguras se basa en la autenticación, la encapsulación y el cifrado.

Tabla VI. **Características de una vpn segura**

Característica	Propósito
Autenticación	Asegura que solo los remitentes y dispositivos autorizados ingresen a la red.
Confidencialidad en los datos	Protege los datos de los espías (*Spoofing*, suplantación de identidad).
Integridad de los datos	Garantiza que no suceda ningún tipo de manipulación o que ocurran alteraciones en los datos.

Fuente: elaboración propia.

Implementando adecuadamente la seguridad, las implementaciones exitosas de VPN cumplen tres objetivos:

- Autenticación: la autenticación asegura que un mensaje proviene de una fuente auténtica y va a un destino auténtico. La identificación del usuario le da la confianza al usuario de que la parte con la que el usuario establece comunicación es quién cree que es el usuario. Las tecnologías VPN están haciendo uso de varios métodos confiables para establecer la identidad de la parte en el otro extremo de la red. estos incluyen contraseñas, certificados digitales, tarjetas inteligentes y datos biométricos.

- Confidencialidad de los datos: una de las preocupaciones de seguridad tradicionales es proteger los datos de los espías. Como característica de diseño, la confidencialidad de los datos tiene como objetivo proteger los contenidos del mensaje de ser interpretados por fuentes no autenticadas o no autorizadas. Las VPN logran confidencialidad utilizando mecanismos de encapsulación y encriptación.

- Integridad de datos: ya no tiene control sobre dónde han viajado los datos y quien ha visto o manejado los datos que envía o recibe mientras los datos viajan a través de internet, siempre existe la posibilidad de que los datos se hayan modificado. La integridad de los datos garantiza que no se produzcan alteraciones o alteraciones en los datos mientras viaja entre el origen y el destino. Las VPN suelen utilizar una de las tres tecnologías para garantizar la integridad de los datos: funciones *hash* unidireccionales, códigos de autenticación de mensajes (MAC) o firmas digitales.

3.6.7. Seguridad de VPN: encapsulación

la incorporación de las capacidades de confidencialidad de datos adecuadas en una VPN garantiza que solo las fuentes y destinos previstos sean capaces de interpretar los contenidos del mensaje original. La encapsulación es uno de los principales componentes de la confidencialidad. La encriptación es la otra. El túnel es la transmisión de datos a través de una red pública para que los nodos de enrutamiento es la red pública no sepan que la transmisión forma parte de una red privada. El túnel permite el uso de redes públicas (por ejemplo, internet) para transportar datos en nombre de los usuarios como si los usuarios tuvieran accesos a una red privada. De ahí viene el nombre de VPN.

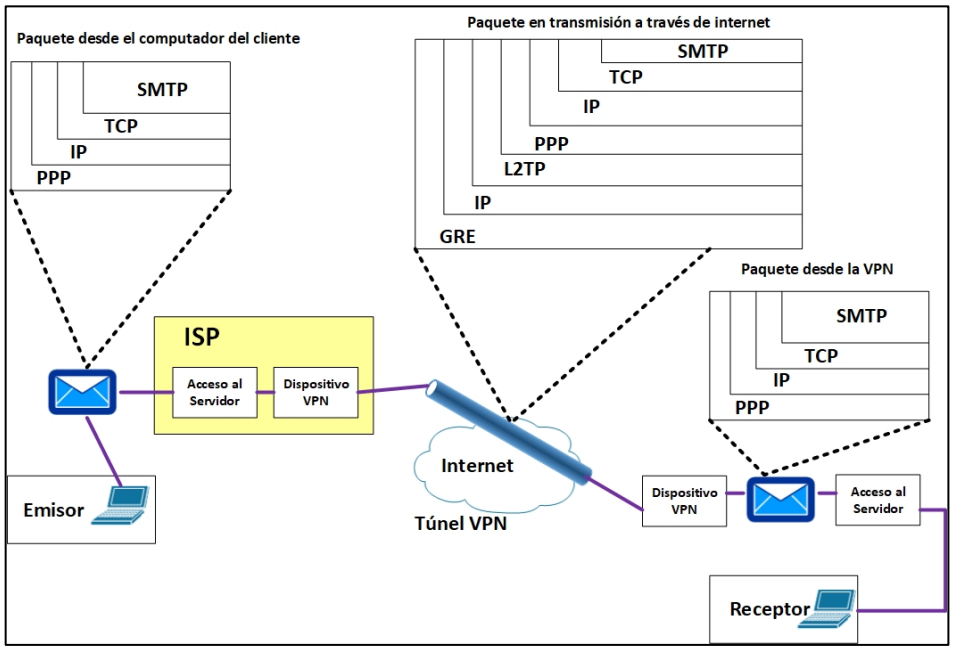
Las VPN construyen túneles encapsulando los datos de la red privada y la información del protocolo dentro de los datos del protocolo de red pública para que los datos del túnel no estén disponibles para cualquiera que examine los marcos de datos transmitidos.

El túnel es el proceso de colocar un paquete completo dentro de otro paquete y enviar el nuevo paquete compuesto a través de una red. En la figura,

el origen externo del paquete y direccionamiento de destino se asignan a “interfaces de túnel” y se pueden enrutar a través de la red. Una vez que un paquete compuesto llega a la interfaz del túnel de destino, se extrae el paquete interno.

En la figura 178 se muestran los tres protocolos diferentes que utiliza el túnel:

Figura 178. **Seguridad VPN: encapsulación de paquetes**



Fuente: elaboración propia, empleando Visio 2013.

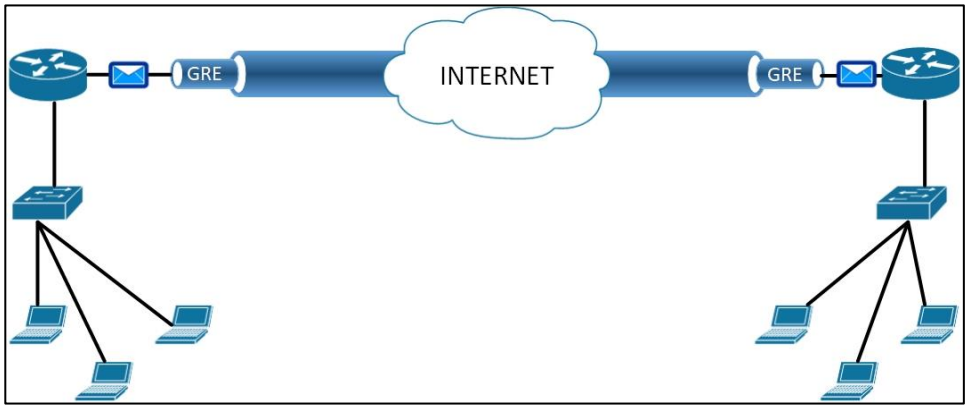
- Protocolo de operador: el protocolo por el que viaja la información.

- Protocolo encapsulante: el protocolo (GRE, IPsec, L2F, PPTP, L2TP) que está envuelto alrededor de los datos originales. No todos los protocolos ofrecen el mismo nivel de seguridad.

- Protocolo de pasajero: los datos originales (IPX, IPv4, IPv6).

En la figura 179 se ilustra un correo electrónico que viaja a través de internet a través de una conexión VPN. PPP lleva el mensaje al dispositivo VPN donde el mensaje se encapsula dentro de un paquete genérico de encapsulado de enrutamiento (GRE).

Figura 179. **Seguridad VPN: ejemplo de encapsulación y proceso de túnel**



Fuente: elaboración propia, empleando Visio 2013.

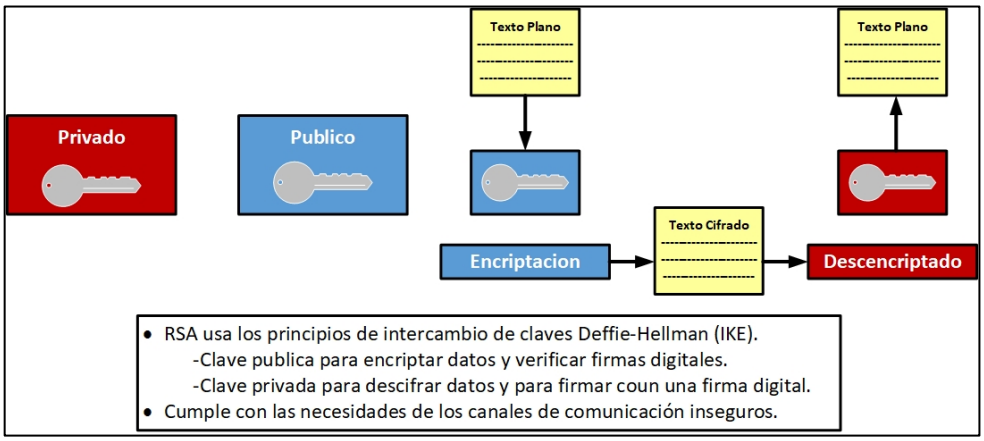
Para reforzar los conceptos de tunelización, considere un ejemplo de envío de una tarjeta navideña a través del correo tradicional. La tarjeta navideña tiene un mensaje adentro y es el protocolo del pasajero. La tarjeta se coloca dentro de un sobre (protocolo de encapsulado) con el direccionamiento adecuado aplicado. El sobre se coloca dentro de un buzón para la entrega. El

sistema postal (protocolo de operador) recoge y entrega el sobre a su buzón. Los dos puntos finales en el sistema de operador son las 'interfaces de túnel'. Quita la tarjeta de vacaciones (extrae el protocolo de pasajero) y lee el mensaje.

3.6.8. Cifrado asimétrico

Dos algoritmos asimétricos utilizados para IPsec son *Diffie-Hellman* (DH) y RSA. los dispositivos de distintas marcas utilizan RSA y *Diffie-Hellman* cada vez que se establece un nuevo túnel IPsec. RSA auténtica el dispositivo remoto mientras *Diffie-Hellman* intercambian las claves que se utilizan para el cifrado. La Asociación de Seguridad de Internet (ISA, *internet security association*) implementa estos protocolos en hardware especializado para garantizar una configuración rápida de túnel y un alto rendimiento de cifrado general.

Figura 180. **Encriptación asimétrica: Deffie-Hellman y RSA**



Fuente: elaboración propia, empleando Visio 2013.

RSA (llamado así por los diseñadores Rivest, Shamir y Adelman) es un algoritmo para el cifrado de clave pública y fue el primer algoritmo conocido

tanto para firmar como para encriptar. RSA fue uno de los primeros grandes avances en la criptografía de clave pública.

La seguridad del criptosistema RSA se basa en dos problemas matemáticos: el problema de factorizar números muy grandes y el algoritmo RSA mismo. Se piensa que el descifrado completo de un texto de cifrado RSA es imposible porque ambos problemas son difíciles y no existe un algoritmo eficiente para resolverlos. Todavía no se ha encontrado ningún método de tiempo polinomial para factorizar enteros grandes en una computadora clásica, pero no se ha demostrado que no exista un método. A partir de 2005, el mayor número que fue factorizado por métodos de propósito general fue 663 *bits* de largo utilizando métodos distribuidos de última generación. Las claves RSA suelen tener una longitud de 1 024 a 2 048 bits.

La criptografía de clave pública es computacionalmente intensiva. Para lograr la mejor combinación de rendimiento y funcionalidad, DH combinó criptografía de clave pública con clave secreta de criptografía. El acuerdo clave DH se inventó en 1976 durante la colaboración entre Whitfield Diffie y Martin Hellman, y fue el primer método práctico para establecer un secreto compartido sobre el canal de comunicaciones sin protección.

Como algoritmos simétricos, DES, 3DES, Message Digest 5 (MD5) y SHA requieren una clave secreta compartida para realizar cifrado y descifrado. La pregunta es: ¿cómo los dispositivos de cifrado y descifrado tienen la clave secreta compartida? Las posibles soluciones son que las claves se pueden enviar por correo electrónico, servicio de mensajería instantánea o cambio de clave pública. Otro método más fácil y seguro es el intercambio de claves públicas DH. El acuerdo de clave DH es un método de cifrado de clave pública que proporciona una forma para que dos pares establezcan una clave secreta

compartida que solo los pares conocen, aunque los pares se comuniquen a través de un canal inseguro.

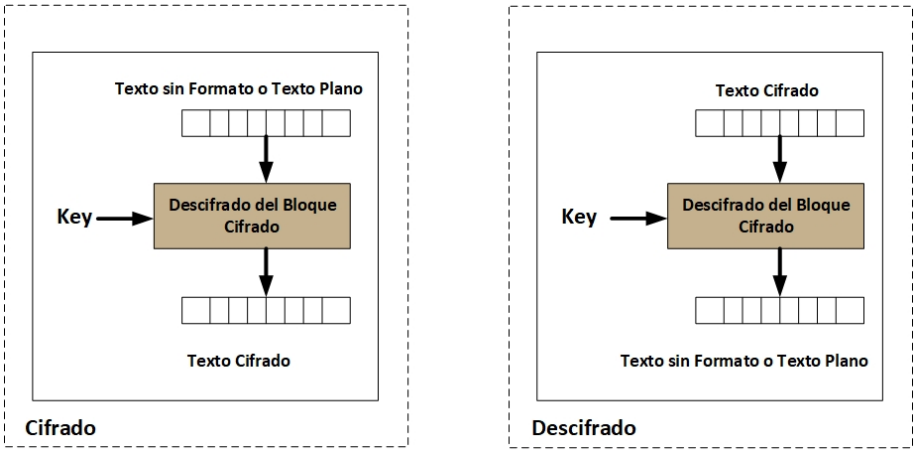
Los criptosistemas de clave pública se basan en un sistema de dos claves:

- Clave pública: intercambiada entre usuarios finales
- Clave privada: secreto guardado por los propietarios originales

3.6.9. Algoritmos de cifrado simétrico

Los algoritmos de clave simétrica se pueden dividir en cifrado de flujo y cifrado de bloque. Los cifrados de flujo encriptan los *bits* del mensaje, uno a la vez, y los cifrados de bloque toman una cantidad de *bits* y los encriptan como una sola unidad. Un cifrado de bloques opera en grupos de bits de longitud fija, denominados bloques, con una transformación invariable. Al cifrar, un cifrado de bloques podría tomar, por ejemplo, un bloque de texto plano de 128 *bits* como entrada, y generar un bloque correspondiente de texto cifrado de 128 *bits*. La transformación exacta se controla con una segunda entrada: la clave secreta. El descifrado es similar: el algoritmo de descifrada toma, en este ejemplo, un bloque de texto cifrado de 128 *bits* junto con la clave secreta, y cede el bloque original de texto claro de 128 *bits*. Por otro lado, los sistemas de cifrado de flujo operan en dígitos individuales, uno a la vez, y la transformación varía durante el cifrado. La figura 181 muestra el proceso básico de cifrado y descifrado de un cifrado de bloques.

Figura 181. **Encriptación simétrica: 3DES**



Fuente: elaboración propia, empleando Visio 2013.

El estándar de cifrado de datos (DES), desarrollado en IBM y publicado como estándar en 1977, ha influido en el desarrollo de los diseños de cifrado en bloque actualmente en uso. Un sucesor de DES, el algoritmo de estándar de cifrado avanzado (AES) aprobado por el Instituto Nacional de Estándares y Tecnología (NIST) en diciembre de 2001 utiliza bloques de 128 bits.

La tabla se enumera los algoritmos de encriptación simétrica comunes y muestra las diferencias en los niveles de seguridad que ofrecen varios algoritmos. El factor de trabajo (O) representa la fuerza del algoritmo.

Tabla VII. **Algoritmos de encriptación simétrica comunes y niveles de seguridad**

Algoritmos	Nivel de seguridad	Factor de trabajo
DES, MD5	Débiles	O(2^40)
RC4, SHA-1	Legado	O(2^64)
3DES	Línea base	O(2^80)
AES-128, SHA-256	Estándar	O(2^128)
AES-192, SHA-384	Alta	O(2^192)
AES-256, SHA-512	Ultra alta	O(2^256)

Fuente: elaboración propia.

Las siguientes tres descripciones muestran como se ha desarrollado el cifrado simétrico desde una solución relativamente débil hasta el algoritmo actual y más ampliamente aceptando.

3.6.10. Cifrado simétrico: DES

El DES es un cifrado que fue seleccionado como un estándar oficial de procesamiento de información federal (FIPS) para los Estados Unidos en 1976. Por esta razón, el DES se implementó ampliamente a nivel internacional. El algoritmo fue inicialmente controvertido, con elementos de diseño clasificados y una longitud de clave relativamente corta. En consecuencia, DES fue sometido a un intenso escrutinio académico y motivó la comprensión moderna de las cifras cifradas y su criptoanálisis. Alguna documentación se refiere a DES como el algoritmo de encriptación de datos (DEA).

- Algoritmo de cifrado de clave simétrica.

- Cifrado de bloque: funciona en un bloque de datos de 64 *bits*, utiliza una clave de 56 bits (el último *bit* de cada *byte* es utilizado para la paridad).
- Modo de operación: aplicar DES para cifrar bloques de datos.

DES es un algoritmo de encriptación de cifrado de bloques. El algoritmo DES toma una cadena de longitud fija de bits de texto sin formato y la transforma a través de una serie de operaciones complicadas en otra cadena de bits de texto cifrado de la misma longitud y devuelve bloques de textos cifrado del mismo tamaño. Dado a que tiene 64 *bits*, tiene 2^64 combinaciones posibles. DES simplemente reorganiza los bits en combinaciones que requieren el procedimiento inverso para decodificar el texto sin formato.

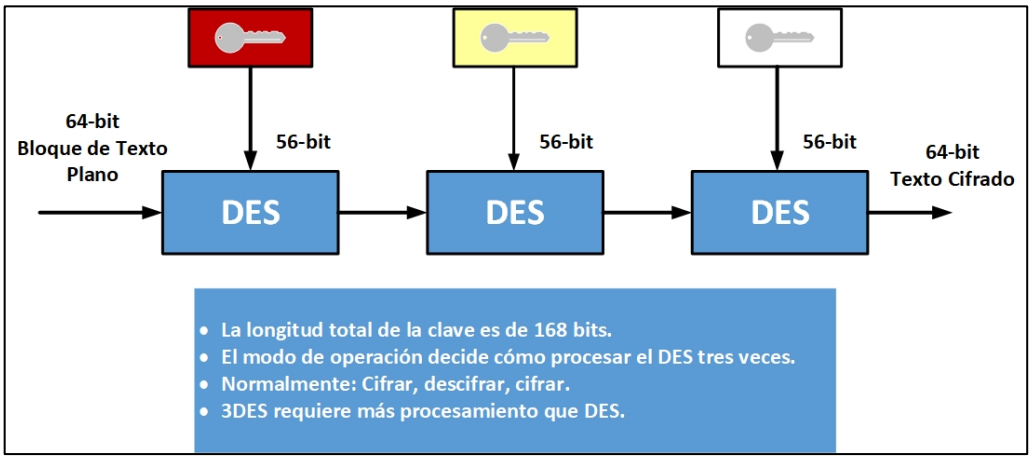
DES usa una clave para personalizar la transformación, de modo que el descifrado solo puede realizarlo quien conoce la clave particular utilizada para encriptar. La clave consiste ostensiblemente en 64 bits: sin embargo, solo 56 de estos son realmente utilizados por el algoritmo. Ocho bits se usan únicamente para verificar la paridad, y luego se descartan. Por lo tanto, la longitud efectiva de la clave es de 56 *bits*, y generalmente se cita como tal.

En la actualidad, DES se considera inseguro para muchas aplicaciones, principalmente debido a que el tamaño de la clave DES de 56 *bits* es demasiado pequeño. Las claves DES se han roto en menos de 24 horas. También hay algunos resultados analíticos que demuestran debilidades teóricas en el cifrado. Se cree que el algoritmo es seguro en la forma de triple DES (3DES), aunque existen ataques teóricos que rompen 3DES. En los últimos años, el cifrado ha sido reemplazado por el AES.

3.6.11. Cifrado simétrico: 3DES

3DES, o Triple DES, es un cifrado de bloques que se formó a partir del cifrado DES. 3DES fue desarrollado por Walter Tuchman (el líder del equipo de desarrollo de DES en IBM) en 1978 y está especificado en FIPS Pub 46-3. Hay varias formas de usar DES tres veces; no todas las formas son 3DES y no todas las formas son tan seguras como 3DES.

Figura 182. **Intercambio de llaves *Deffie-Hellman***



Fuente: elaboración propia, empleando Visio 2013.

3DES se define como la realización de un cifrado DES, luego un descifrado DES y luego un cifrado DES nuevamente. En la evolución de DES a 3DES, es necesario explicar la obvia omisión del paso intermedio de doble DES.

2DES se volvió ineficaz debido a un tipo de ataque conocido como *meet-in-the-middle attack*. Este ataque es una búsqueda de fuerza bruta realizada desde ambos extremos de 2DES. La primera operación cifra el texto

sin formato con todas las claves DES posibles y, en segundo lugar, descifra el texto del cifrado del producto con todas las claves DES posibles mientras busca coincidencias. Cuando se encuentra una coincidencia, el atacante tiene ambas claves en Double DES. Para superar este ataque, se agregó una tercera operación DES. Por lo tanto, mientras que 3DES tiene una longitud de clave de 168 *bits* (tres claves DES de 56 *bits*), su longitud de clave efectiva desde un punto de vista de seguridad es de solo 112 *bits*.

3.6.12. Cifrado simétrico: AES

AES, a menudo referido como el cifrado de *Rijndael* (Pronunciado "*Rhine dahl*"), es un cifrado de bloque que fue adoptado como un estándar de cifrado por el gobierno de E.E.U.U. Se espera que AES se use en todo el mundo y se analice exhaustivamente, como fue el caso del predecesor de AES, DES. Además, AES es más seguro y más rápido que 3DES.

- Anteriormente conocido como *Rijndael*.
- Sucesor de DES y 3DES.
- Clave simétrica del bloque de claves.
- El tamaño de bloque fijo es de 128 bits.
- Cifrado fuerte con una larga expectativa de vida.
- AES puede admitir claves de 128, 192 y 256 *bits*; la clave de 128 *bits* se considera segura.

El cifrado fue desarrollado por dos criptógrafos belgas, Joan Daemen y Vincent Rijmen, y se sometió al proceso de selección de AES bajo el nombre *Rijndael* AES no es exactamente igual al *Rijndael* original porque *Rijndael* admite una mayor gama de tamaños de bloque y clave. AES tiene un tamaño de bloque fijo de 128 *bits* y un tamaño de clave de 128, 192 o 256 *bits*, mientras

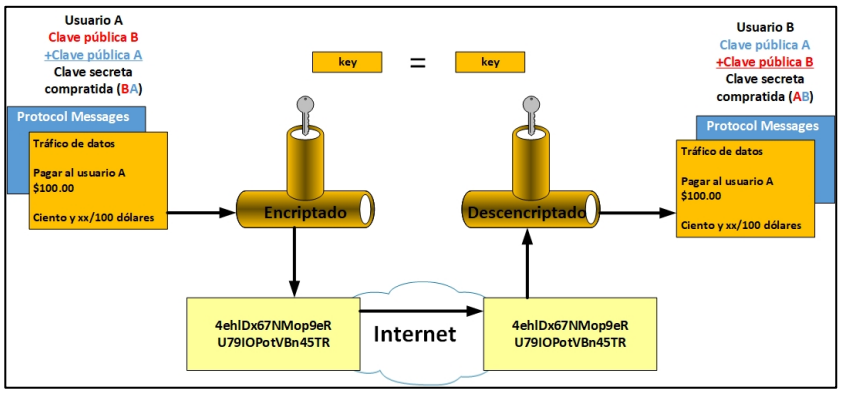
Rijndael se puede especificar con tamaños de clave y bloque en cualquier múltiplo de 32 *bits*, con un mínimo de 128 bits y un máximo de 256 *bits*.

La Agencia de Seguridad Nacional del gobierno de los EE.UU. (NSA) revisó todas las cifras enviadas como finalistas al proceso de selección de AES, incluida Rijndael, declaró que todos los finalistas eran suficientemente seguros como para usar datos gubernamentales no clasificados del gobierno de EE.UU. En junio de 2003, el gobierno de los EE.UU. Anunció que AES puede utilizarse para información clasificada. Esta es la primera vez que el público tiene acceso a un sistema de cifrado aprobado por la NSA para obtener información de alto secreto. Es interesante observar que muchos productos públicos usan claves secretas de 128 *bits* por defecto.

3.6.13. Intercambio de llaves *Diffie-Hellman*

El algoritmo de clave pública *Diffie-Hellman* establece que si el usuario A y el usuario B intercambian claves públicas y se realiza un cálculo en su clave privada individual y en la clave pública del otro interlocutor, el resultado final del proceso es una clave compartida idéntica. La clave compartida se usa para cifrar y descifrar los datos.

Figura 183. **Seguridad VPN: acceso remoto VPN usando IPsec**



Fuente: elaboración propia, empleando Visio 2013.

La seguridad no es un problema con el intercambio de claves *Diffie-Hellman*. Aunque alguien puede conocer la clave pública de un usuario, el secreto compartido no se puede generar porque la clave privada nunca se convierte en conocimiento público.

Con *Diffie-Hellman*, cada par genera un par de claves públicas y privadas. La clave privada que genera cada par se mantiene en secreto y nunca se comparte. La clave pública se calcula a partir de la clave privada de cada par y se intercambia a través del canal inseguro. cada par combina la clave pública del otro par con su propia clave privada y calcula el mismo número secreto compartido. el número secreto compartido luego se convierte en una clave secreta compartida. La clave secreta compartida nunca se intercambia en el canal inseguro.

3.6.14. Ejemplo clásico de *Diffie-Hellman*: Alice y Bob

Muchas veces se ha utilizado este ejemplo de Alice y Bob para explicar el proceso de *Diffie-Hellman*. Este permite a Alice y Bob acordar una clave que pueden usar para cifrar los mensajes que desean enviarse entre ellos. Pueden hacerlo incluso cuando un espía (Eve) escucha toda la conversación. Diffie-Hellman se basa en la superposición de que es fácil elevar un número a cierta potencia, pero es difícil calcular qué potencia se utilizó dado el número y el resultado.

- Paso 1, *Alice y Bob* necesitan acordar un número primo p, lo cual pueden hacer simplemente enviándolo entre ellos. En este caso, el número primo acordado p=23. Eve puede aprender el número p porque en la práctica el número p a menudo simplemente se anuncia en algun lugar póblico.
- Paso 2, dado un número p, es posible obtener un número g (el llamado generador) con una propiedad interesante. Cada número entre 1 y p-1 puede escribirse como una potencia de g al calcular el módulo p. (El siguiente subtema describe brevemente el concepto de aritmética modular.) Por ahora, se acepta que g=5 por ejemplo, usando p=5 el generador es 2, porque:

    - En informática, la operación de módulo se encuentra el recordatorio de división de un número por otro.

    - dados dos números, Y y X, un módulo N (abreviado como mod N) es el recordatorio, en la división de A por N.

    - Por ejemplo, se desea dormir 8 horas y quiere acostarse a las 10 P.M.

- Contar hasta las 12 A.M.
- Alrededor a cero.
- Contar hasta las 6 A.M.
- (10 + 8) mod 12.

- Otros ejemplos:

  - “8 mod 3” se evaluará a 2.
  - “9 mod 3” se evaluará a 0.

2^0=1
2^1=2
2^2=4
2^3=3 (porque 8=3 mod 5)

Alice y Bob acuerdan de la misma manera en un generador g para los números entre 1 y p-1. En este punto, los números p y g sirven como clave pública.

- Paso 3 *Alice y Bob* eligen números aleatorios, a y b respectivamente. Ambos números permanecen en secreto porque solo Alice conoce su número y solo Bob conoce su número. En el ejemplo, Alice eligió a=6 y Bob eligio b=16.

- Paso 4 Alice luego calcula ga mod p y Bob calcula gb mod p. Ellos intercambian sus resultados.

- Paso 5 la clave con la que Alice y Bob ahora están de acuerdo es simplemente ga*b. Esto es bastante fácil de calcular:

- Alice sabe a y gb
- Bob sabe b y ga, y
- (ga) b = (gb) a = ga * b

- Paso 6 Alice y Bob pueden usar la clave ga * b para encriptar mensajes con cualquier algoritmo de clave secreta.

La seguridad del sistema *Diffie-Hellman* depende de la suposición de que es fácil elevar un número a cierta potencia, pero es difícil calcular qué potencia se utilizó dado el número y el resultado. Por ejemplo, es fácil calcular 210 = 1024, pero es más difícil determinar qué 1024 es la décima potencia de 2.

Eve conoce a ga y a gb, pero como ella no sabe ni a ni a sí misma, no puede calcular la clave en un tiempo razonable.

3.6.15. Primeros números y aritmética modular

El algoritmo de intercambio de claves *Diffie-Hellman* utiliza una serie de cálculos basados en números primos y aritmética modular. Recuerde que un número primo es un número entero (un número entero) que tienen como únicos factores 1 y el mismo (por ejemplo, 2, 17, 23 y 127 son primos). El algoritmo *Diffie-Hellman* usa las propiedades especiales asociadas con los números primos.

La aritmética modular se basa en el concepto de hacer las operaciones de suma y otras en un círculo en lugar de una línea. Los valores en cualquier operación aritmética 'se vuelven' y siempre son menores que un número fijo llamado módulo.

Por ejemplo, para encontrar 39 módulo 7, simplemente calcula 39/7 (= 5 4/7) y toma el resto. En este caso, 7 se divide en 39 con un resto de 4. Por lo tanto, 39 módulo 7 = 4. Tenga en cuenta que el resto (cuando se divide por 7) siempre es menor que 7. Por lo tanto, los valores 'se envuelven', como se muestra en el siguiente ejemplo:

0 mod 7 = 0
1 mod 7 = 1
2 mod 7 = 2
3 mod 7 = 3
4 mod 7 = 4
5 mod 7 = 5
6 mod 7 = 6
7 mod 7 = 0
8 mod 7 = 1
9 mod 7 = 2
10 mod 7 = 3
y así

En adición modular, primero agrega los dos números normalmente, luego divida por el módulo y tome el resto. Por lo tanto, (17+20) mod 7 = (37) mod 7 = 2.

Una simple analogía ayudará a aclarar este concepto, pero es probable que lo haya utilizado antes cuando calcula cuando tendría que levantarse por la mañana si quiere dormir cierto número de horas. Por ejemplo, suponga que planea acostarse a las 10 PM y quiere dormir 8 horas. Para saber a qué hora configurar su alarma, cuente, a partir de las 10, las horas hasta la medianoche (en este caso, dos). A la media noche (12), restablece a cero (se 'ajusta' a 0) y

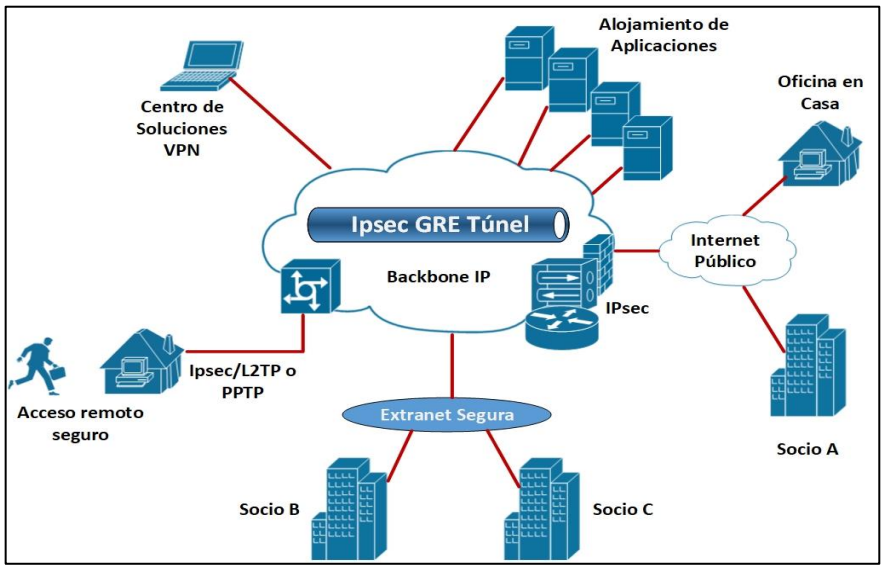
continúa contando hasta que su total sea 8. El resultado es 6 AM. por lo que el resultado es (10 + 8) mod 12. Siempre y cuando no quieras dormir más de 12 horas, obtendrás la respuesta correcta con esta técnica.

3.6.16. Seguridad VPN: IPSEC Y GRE

Los protocolos de túnel varían en las características que admiten, los problemas que intentan resolver y la cantidad de seguridad que proporcionan a los datos que transportan. Este caso se enfoca en usar IPsec e IPsec con GRE.

Cuando se usa solo, IPsec proporciona una red privada y resistente solo para unidifusión IP. Use IPsec junto con GRE cuando se requiera soporte para multidifusión IP, protocolos dinámicos de enrutamiento IGP o protocolos que no sean IP. La figura 184 muestra un ejemplo de VPN de acceso remoto seguro.

Figura 184. **Características de seguridad IPsec**



Fuente: elaboración propia, empleando Visio 2013.

IPsec tiene dos modos de encriptación:

- Modo túnel
- Modo transporte

El modo túnel encripta el encabezado y la carga útil de cada paquete mientras que el modo de transporte solo cifra la carga útil. Solo los sistemas que son compatibles con IPsec pueden aprovechar el modo de transporte. Además, todos los dispositivos deben usar una clave común y los *firewalls* de cada red deben configurarse con políticas de seguridad similares. IPsec puede cifrar datos entre varios dispositivos, incluidos *router* a *router*, *firewall* en *router*, PC en router y PC en servidor.

GRE incluye el encabezado IP y la carga de paquetes con un encabezado de encapsulado GRE. Los diseñadores de red usan este método de encapsulación para ocultar el encabezado IP de los paquetes como parte de la carga útil encapsulada GRE. Al ocultar información, los diseñadores separan o 'canalizan' datos de una red a otra sin realizar cambios en la infraestructura de red común subyacente.

3.6.17. Túneles VPN de punto a punto

En una VPN de punto a punto, GRE proporciona el marco para empaquetar el protocolo de pasajeros para el transporte a través del protocolo de operador (generalmente basado en IP). Este transporte incluye información sobre qué tipo de paquete está encapsulado e información sobre la conexión entre el cliente y el servidor.

Las VPN de punto a punto también pueden usar IPsec en modo túnel como el protocolo de encapsulado. IPsec funciona bien en VPN de acceso remoto y de punto a punto. Para usar IPsec, ambas interfaces de túnel deben ser compatibles con IPsec.

3.6.18. Túneles: acceso remoto

En una VPN de acceso remoto, el *tunneling* a menudo usa PPP y protocolos asociados. Cuando la comunicación se establece a través de la red entre la computadora host y un sistema de acceso remoto, PPP es el protocolo del operador.

Las VPN de acceso remoto también pueden usar los protocolos enumerados a continuación. Cada protocolo usa la estructura básica de PPP:

- Reenvío de capa 2 (L2L): desarrollado por *Cisco Systems*, L2F usa cualquier esquema de autenticación que sea compatible con PPP. Sin embargo, L2F no es compatible con el cifrado.

- Protocolo de túnel punto a punto (PPTP): el PPTP *Forum*, un consorcio que incluye US *Robotics*, Microsoft, 3COM, *Ascend* y ECI *Telematics*, creó PPTP. PPTP admite el cifrado de 40 y 128 bits y utiliza cualquier esquema de autenticación compatible con PPP.

- Protocolo de túnel de capa 2 (L2TP): L2TP es un producto de una asociación entre miembros del foro PPTP, *Cisco Systems* y el grupo de trabajo de ingeniería de internet (IETF). Es una combinación de los protocolos PPTP y L2F. Tanto las VPN de punto a punto como las VPN de acceso remoto pueden usar L2TP como protocolo de túnel. Sin

embargo, debido a la falta de confidencialidad inherente al protocolo L2TP, a menudo se implementa junto con IPsec y se denomina L2TP/IPsec.

3.6.19. Características de seguridad de IPSEC

IPsec proporciona un mecanismo para la transmisión segura de datos a través de redes IP, garantizando la confidencialidad, integridad y autenticidad de las comunicaciones de datos a través de redes no protegidas como el internet.

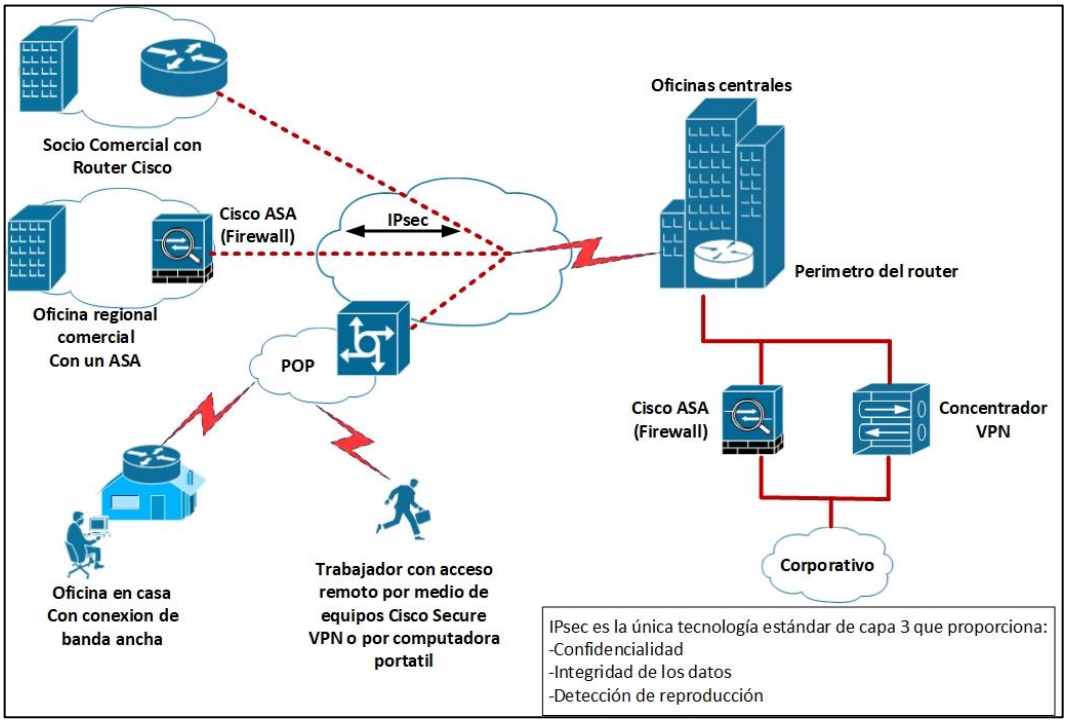
- Un estándar IETF que se emplea mecanismos criptográficos en la capa de red:
    - Autenticación de cada paquete de IP.
    - Verificación de integridad de datos para cada paquete.
    - Confidencialidad de la carga útil del paquete.
- Consiste en estándares abiertos para asegurar las comunicaciones privadas.
- Escalable desde redes muy pequeñas a redes muy grandes.

IPsec abarca un conjunto de protocolos y no está vinculado a ningún algoritmo de cifrado o autenticación específica, técnica de generación de clave o asociación de seguridad (*Security Association*, SA). IPsec proporciona las reglas mientras que los algoritmos existentes proporcionan encriptación, autenticación, administración de claves, entre otros.

IPsec actúa en la capa de red, protegiendo y autenticando paquetes IP entre dispositivos IPsec (emparejados), como *Adaptive Security Appliances* (ASA), *routers Cisco*, *Cisco Secure VPN* Client y otros productos IPsec.

IPsec es un estándar llamado Internet Engineering Task Force (IETF) y también se encuentra dentro de los estándares RFC 2401-2412 que define cómo se puede crear una VPN a través de redes IP. IPsec proporciona las siguientes funciones de seguridad esenciales:

Figura 185. **Encabezados IPsec**



Fuente: elaboración propia, empleando Visio 2013.

- Confidencialidad de los datos: IPsec garantiza la confidencialidad mediante el uso de cifrado. El cifrado de datos evita que terceros lean

datos, especialmente los datos que se transmiten a través de redes públicas o redes inalámbricas. El remitente IPsec puede encriptar paquetes antes de transmitir los paquetes a través de una red y evitar que cualquiera oiga lo vea la comunicación (escuchas). Si se intercepta, los datos no pueden decodificarse. La encriptación se proporciona utilizando algoritmos de encriptación que incluyen DES, 3DES y AES.

- Integridad de los datos: IPsec garantiza que los datos lleguen sin cambios en el destino; es decir, que los datos no se manipulan en ningún punto a lo largo de la ruta de comunicación. IPsec garantiza la integridad de los datos mediante hashes. Un *hash* es una simple verificación de redundancia. El protocolo IPsec suma los componentes básicos de un mensaje (generalmente el número de *bytes*) y almacena el valor total. IPsec realiza una operación de suma (*checksum*) de comprobación de los datos recibidos y compara el resultado con la suma de comprobación (*checksum*). Si las sumas coinciden, los datos se consideran que no han sido manipulados. La integridad de los datos se proporciona a través de la función *Hash-Based Message Authentication Code* (HMAC). Las funciones soportadas de HMAC incluyen Message Digest 5 (MD5) y *Secure Hash Algorithm* 1 (SHA-1).

- Autenticación de origen de datos: el receptor IPsec puede autenticar la fuente de los paquetes IPsec. La autenticación garantiza que la conexión se realice realmente con el socio de comunicación deseado. IPsec auténtica usuarios (personas) y dispositivos que pueden llevar a cabo la comunicación de forma independiente. la calidad de la autenticación de origen de datos depende del servicio de integridad de datos que se proporciona.

- *Anti-Replay*: la protección *Anti-Replay* verifica que cada paquete sea único, no duplicado. Los paquetes de IPsec están protegidos al comparar el número de secuencia de los paquetes recibidos y una ventana deslizante en el host de destino o puerta de enlace de seguridad. Un paquete cuyo número de secuencia es anterior a la ventana deslizante se considera tarde o duplicado. Los paquetes retrasados y duplicados se eliminan.

3.6.20. Protocolos y encabezados IPsec

El estándar IPsec proporciona un método para administrar la autenticación y la protección de datos entre múltiples pares que participan en la transferencia segura de datos. IPsec incluye un protocolo para intercambiar claves denominado intercambio de claves de internet (IKE, Internet Key Exchange) y dos protocolos IP de IPsec, *encapsulating security payload* (ESP) y *authentication header* (AH).

En términos simples, IPsec proporciona túneles seguros entre dos pares, como dos routers. El remitente define qué paquetes necesitan protección y se enviaran a través de estos túneles seguros y luego define los parámetros necesarios para proteger estos paquetes sensibles al especificar las características de estos túneles. Luego, cuando el par IPsec se ve un paquete tan sensible, el par IPsec configura el túnel seguro apropiado y envía el paquete a través del túnel al par remoto.

Más exactamente, estos túneles son conjunto de Asociaciones de Seguridad (SA, *security association*). Establecido entre dos vecinos remotos de IPsec. Las asociaciones de seguridad definen qué protocolos y algoritmos deben aplicarse a los paquetes sensibles y especifican el material de claves

que utilizarán los dos pares. Las *Security Association* (SA) son unidireccionales y están establecidas por el protocolo de seguridad que se usa (AH o ESP).

IPsec utiliza tres protocolos principales para crear un marco de seguridad:

- IKE: proporciona un marco para la negociación de parámetros de seguridad y establece claves autenticadas. IPsec utiliza algoritmos de cifrado simétricos para la protección de datos, que son más eficientes y más fáciles de implementar en el hardware que otros tipos de algoritmos. Estos algoritmos necesitan un método seguro de intercambio de claves para garantizar la protección de datos. Los protocolos IKE proporcionan la capacidad de intercambio seguro de claves.

- AH: el encabezado de autenticación (AH, *authentication header*) proporciona integridad sin conexión y autenticación de origen de datos para datagramas IP y protección opcional contra repeticiones. AH está incrustado en los datos que deben protegerse. ESP ha reemplazado el protocolo AH, y AH ya no se utiliza con mucha frecuencia en IPsec.

- ESP: *encapsulating security payload* (ESP) proporciona un marco para datos, autenticación de datos opcional y servicios antireproducción. ESP encapsula los datos que necesitan protección. La mayoría de las implementaciones de IPsec usan el protocolo ESP.

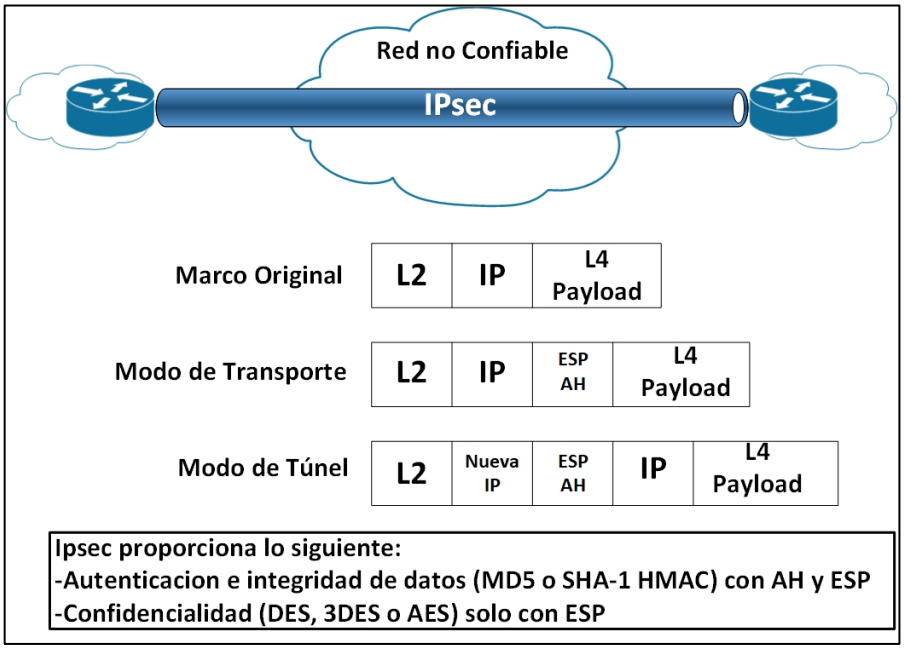
RFC 2401 define la arquitectura para IPsec, incluido en el marco y los servicios que proporcionan. RFC 2401 también define como los servicios trabajan juntos y como y donde usar los servicios. Otros RFC definen los protocolos individuales. Más allá de estos protocolos, el marco consiste en los

detalles de la implementación, como el algoritmo de cifrado exacto y la longitud de clave que se utiliza para ESP.

- Encabezados IPsec

IPsec proporciona autenticación, integridad y cifrado mediante la inserción de uno o ambos encabezados específicos, AH o ESP, en el datagrama IP.

Figura 186. **IKE**



Fuente: elaboración propia, empleando Visio 2013.

AH proporciona verificación de autenticación e integridad en el datagrama IP. La autenticación exitosa significa que el paquete fue, de hecho, enviado por el aparente remitente. Integridad significa que el paquete no fue cambiado durante el transporte.

El encabezado ESP proporciona información que indica el cifrado del contenido de la carga útil del datagrama. El encabezado ESP también proporciona verificación de autenticación e integridad. AH y ESP se usan entre dos hosts. Estos *hosts* pueden ser estaciones finales o puertas de enlace.

AH y ESP proporcionan servicios para transportar protocolos de capa como TCP y *user datagram protocol* (UDP). AH y ESP son protocolos de internet y la Autoridad de Números Asignados de Internet (IANA) les asigna los números 51 (AH) Y 50 (ESP).

Las soluciones AH y ESP requieren una forma basada en estándares para asegurar que los datos sean modificados y leídos por un tercero. IPsec tiene la opción de diferentes encriptaciones (estándar de cifrado de datos [DES, *data encryption standard*], estándar de cifrado de datos triple [3DES, *triple data encryption standard*], y estándar de cifrado avanzado [AES, *advanced encryption*]) para que los usuarios puedan elegir la fortaleza de su protección de datos.

IPsec también tiene varios métodos hash para elegir (código de autenticación basado en hash [HMAC, H*ash-based message authentication code]*, *message digest* 5 [MD5] y *secure hash algorithm* 1 [SHA-1]), cada uno brinda diferentes niveles de protección.

3.6.21. Intercambio de claves de internet

Para implementar una solución VPN con cifrado, es necesario cambiar periódicamente las claves de cifrado. Si no se cambian estas llaves, la red es susceptible a los ataques de fuerza bruta. IPsec resuelve el problema de la capacidad de sustentarse con el protocolo de intercambio de claves de internet

(IKE), que utiliza otros dos protocolos para autenticar un par y generar claves. El protocolo IKE utiliza el intercambio de claves DH para generar claves simétricas para ser utilizadas por dos pares IPsec. IKE también gestiona la negociación de otros parámetros de seguridad, como los datos a proteger, la fuerza de las claves, los métodos *hash* utilizados y si los paquetes estén protegidos contra la reproducción. IKE utiliza el puerto UDP 500.

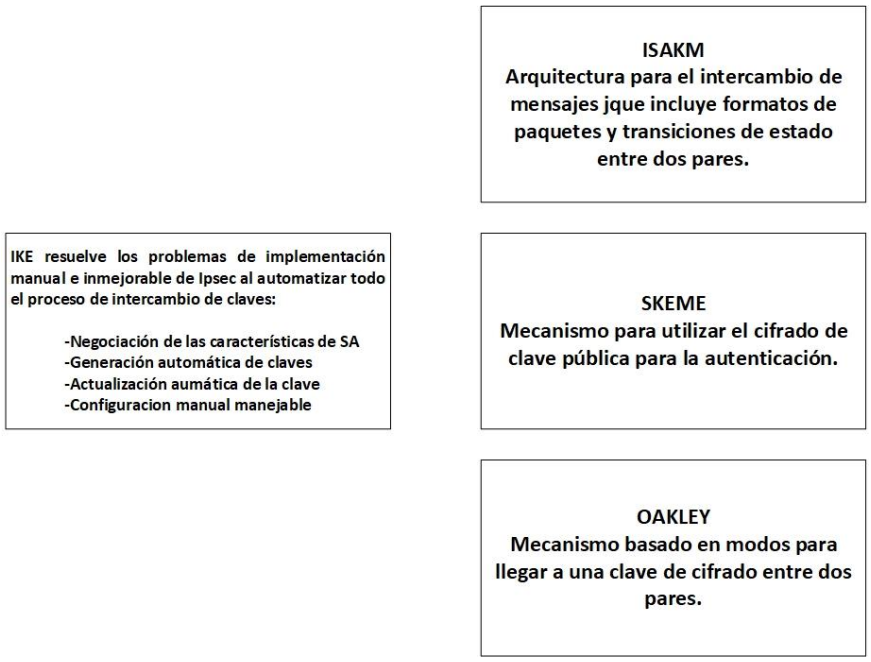
IKE negocia una asociación de seguridad (SA), que es un acuerdo entre dos pares que participan en un intercambio de IPsec, y consta de todos los parámetros necesarios para establecer una comunicación exitosa.

IPsec usa el protocolo IKE para proporcionar estas funciones:

- Negociación de las características de SA
- Generación automática de claves
- Actualización automática de la clave
- Configuración manual manejable

Una Asociación de Seguridad (SA) requiere lo siguiente:

Figura 187. **Modos IKE**



Fuente: elaboración propia, empleando Visio 2013.

- *Internet security association y key management protocol* (ISAKMP): ISAKMP es un marco de protocolo que define los mecanismos para implementar un protocolo que define los mecanismos para implementar un protocolo de intercambio de claves y negociar una política de seguridad. ISAKMP puede implementarse sobre cualquier protocolo de transporte. El documento de referencia para ISAKMP es RFC 2408.

- SKEME: un protocolo de intercambio de claves que define cómo derivar el material de claves autenticado con un refrescado de claves rápido.

- OAKLEY: un protocolo de intercambio de claves que define cómo derivar el material de claves autenticado. El mecanismo básico para OAKLEY es

el algoritmo de intercambio de claves DH. El documento de referencia es RFC 2412: el protocolo de determinación de claves OAKLEY.

IKE negocia automáticamente las SA de IPsec y habilita las comunicaciones seguras de IPsec sin una pre configuración manual costosa. IKE incluye estas características:

- Elimina la necesidad de especificar manualmente todos los parámetros de seguridad IPsec en ambos pares.
- Permite la especificación de por vida para IPsec SA.
- Permite que la claves de cifrado cambien durante las sesiones de IPsec.
- Permite IPsec para proporcionar servicios antireproducción.
- Permite el soporte de la autoridad de certificación (CA) para una implementación de IPsec escalable y manejable.
- Permite la autenticación dinámica de pares.

3.6.22. Fases y modos IKE

IKE se ejecuta en dos fases para establecer un canal de comunicación seguro entre dos vecinos:

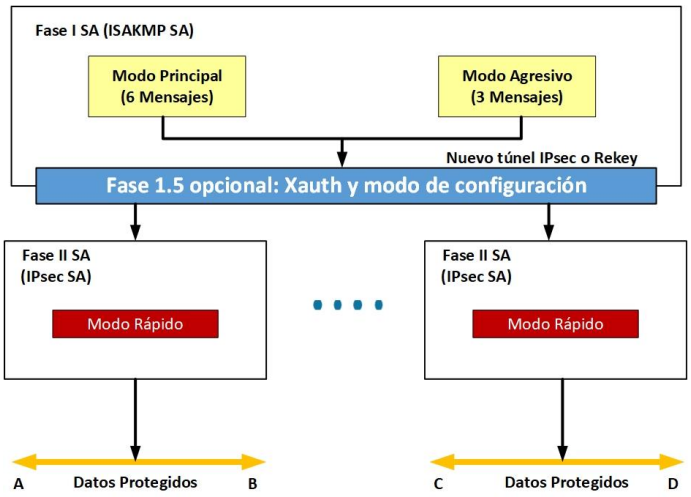
- IKE fase 1: fase 1 es la negociación inicial de SA entre dos pares de IPsec. Opcionalmente, la fase 1 también puede incluir autenticación en la que cada par puede verificar la identidad del otro. Esta conversación entre dos pares IPsec puede estar sujeta a escuchas sin que el tercero descubra la vulnerabilidad pueden enviar y recibir usando el mismo material clave que se genera. IKE fase 1 ocurre en dos modos: modo principal o modo agresivo.

- IKE fase 1.5 (opcional): para autenticar aún más a los participantes de VPN (clientes), puede usar un protocolo llamado *Extended Authentication* (Xauth) que proporciona autenticación de usuario de túneles IPsec dentro del protocolo IKE. Además, puede intercambiar otros parámetros entre los vecinos. La configuración de modo se usa para entregar parámetros tales como la dirección IP y la dirección del sistema de nombres de dominio (DNS) al cliente.

- IKE fase 2: las SA de fase 2 son negociadas por el proceso IKE (ISAKMP) en nombre de otros servicios como IPsec que necesitan material clave para su funcionamiento. Debido a que las SA que utiliza IPsec son unidireccionales, se necesitan intercambios de claves por separado para los datos que fluyen hacia adelante y hacia atrás. Los dos pares ya han acordado los conjuntos de transformación, los métodos hash y otros parámetros durante la negociación de la fase 1. El modo rápido es el método utilizado para las negociaciones de SA de la fase 2.

Para establecer un canal de comunicación seguro entre dos pares, el protocolo IKE utiliza estos tres modos de operación:

Figura 188. **Operación IKE**



Fuente: elaboración propia, empleando Visio 2013.

- Modo principal: en el modo principal, una sección IKE comienza cuando el iniciador envía una propuesta o propuestas al respondedor. Estas propuestas definen qué protocolos de encriptación y autenticación son aceptables, cuanto tiempo las claves deben permanecer activas y se debe cumplir el secretismo (PFS, *Perfect Forward Secrecy*) perfecto. Se pueden enviar múltiples propuestas en una sola oferta. El primer intercambio entre nodos establece la política de seguridad básica. El respondedor elige la propuesta adecuada y envía la respuesta al iniciador. El siguiente intercambio pasa claves públicas DH y otros datos. Toda la negociación adicional está encriptada dentro de IKE SA. El tercer intercambio auténtica la sesión ISAKMP. Una vez que se establece IKE SA, comienza la negociación de IPsec (modo rápido).

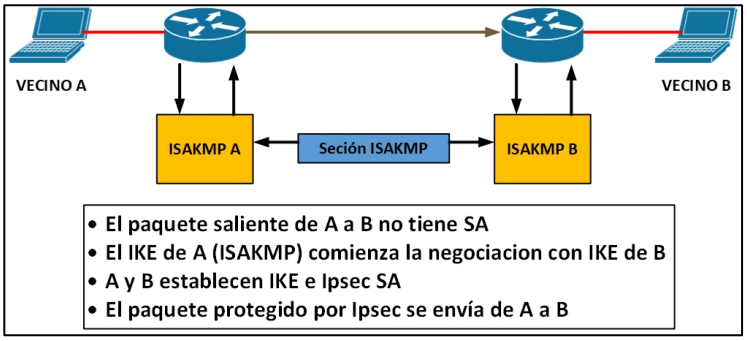
- Modo agresivo: el modo agresivo comprime la negociación IKE SA en tres paquetes, con todos los datos necesarios para que el iniciador pase

la SA. El respondedor envía la propuesta, el material de claves e identificación y autentica la sesión en el siguiente paquete. El iniciador responde autenticando la sesión. La negociación es más rápida que en el modo principal, y la ID del iniciador y el responder pasan en texto sin formato. El modo agresivo es apropiado y debe usarse siempre que los dispositivos sean capaces de manejar el modo principal sin dificultad.

- Modo rápido: la negociación IPsec en modo rápido es similar a una negociación IKE en modo agresivo, excepto que la negociación debe estar protegida dentro de una SA IKE. El modo rápido negocia el SA para el cifrado de datos y gestiona el intercambio de claves para ese IPsec SA. Si el encuestado da una respuesta negativa, el iniciador enviará la solicitud en modo principal.

En la figura 189 se ilustra cómo una negociación IKE da como resultado comunicaciones seguras entre dos SA.

Figura 189. **El problema: IPsec: IPsec y NAT**



Fuente: elaboración propia, empleando Visio 2013.

3.6.23. Otras funciones IKE

IKE puede entregar funciones adicionales que se utilizan para verificar si el dispositivo vecino aun este activo, para pasar IPsec a través de dispositivos de traducción de direcciones de red (NAT, *network address translation*) o para intercambiar parámetros de configuración adicionales.

- DPD:
    - Bidireccional
    - Enviado a intervalos periódicos
    - El que envía debe de recibir una respuesta o desconectarse
- Los keepalives IKE son unidireccionales y se envían cada 10 segundos
- NAT-T
    - Definido en las RFC 3947 y 3948
    - Se encapsula en un paquete IPsec en un paquete UDP
- Modo configuración (configuración de inserción) y Xauth (autenticación de usuario).

3.6.24. *Dead Peer Protection* (*DPD*) y Cisco IOS *Keepalives*

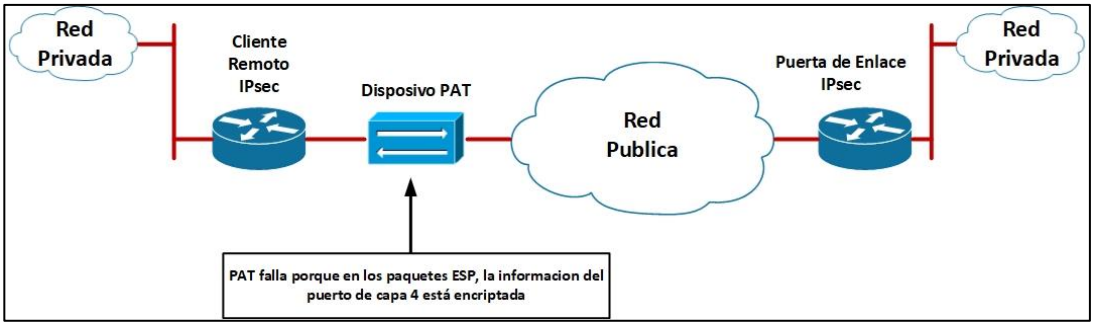
Los keepalives DPD y Cisco IOS funcionan sobre la base de un temporizador, si el temporizador está configurado durante 10 segundos, el *router* enviará un mensaje de 'saludo' cada 10 segundos. El beneficio de los *keepalives* de Cisco IOS y el DPD periódico es la detección temprana de pares

muertos. Sin embargo, los *keepalives* de Cisco IOS y el DPD periódico es la detección temprana de pares muertos. Sin embargo, los *keepalives* de Cisco IOS y el DPD periódico dependen de mensajes periódicos que deben enviarse con frecuencia. El resultado de enviar mensajes frecuentes es que los pares que se comunican deben encriptar y desencriptar más paquetes.

La operación predeterminada de DPD es la mensajería bajo demanda. Con DPD bajo demanda, los mensajes se envían en función de los patrones de tráfico. Si un *router* no tiene tráfico para enviar, el *router* nunca envía un mensaje DPD. Si un par está muerto y el *router* nunca tiene tráfico para enviar al par, el *router* no descubrirá que el par está muerto hasta que IKE o el IPsec SA tengan que volver a seleccionar (la animación del par no es importante si el router no está tratando de comunicarse con el vecino).

El problema con NAT

Figura 190. **La solución: IPsec NAT-T**



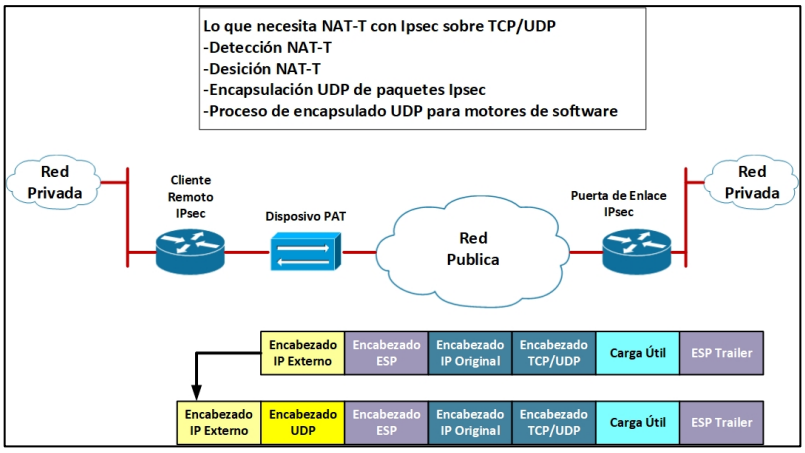
Fuente: elaboración propia, empleando Visio 2013.

Los vecinos de IPsec no pueden ubicarse detrás de un dispositivo habilitado para NAT. Un túnel IPsec VPN estandar no funciona si hay uno o

más puntos NAT (o PAT) en la ruta de entrega del paquete de IPsec. Sin embargo, los proveedores de servicios de Internet (ISP) y las oficinas pequeñas u oficinas domésticas (SOHO, *small office/home office*) a menudo usan NAT para compartir direcciones individuales. Aunque NAT ayuda a conservar el espacio de direcciones IP, NAT también presenta problemas para IPsec. De hecho, un túnel IPsec VPN no funciona si no hay números de puerto en los encabezados IPsec que se pueden usar para crear y mantener tablas de traducción. La información del puerto de capa 4 esta encriptada y, por lo tanto, no se puede leer.

La solución: IPsec NAT transversal (NAT-T)

Figura 191. **Modo de configuración**



Fuente: elaboración propia, empleando Visio 2013.

Para resolver el problema de NAT, el IETF, a través de las RFC 3947 y 3948, ha definido IPsec NAT Transversal (IPsec NAT-T). IPsec NAT-T define los cambios en el proceso de negociación y los diferentes métodos para enviar datos protegidos con IPsec. El IPsec NAT-T, que se introdujo a partir de la

versión 12.2 (13) T de Cisco IOS, permite que el tráfico IPsec viaje a través de dispositivos NAT o PAT en la red al encapsular paquetes IPsec en un contenedor UDP.

NAT-T se negocia con estos factores:

- Detección NAT-T
    - Durante la negociación IKE fase 1, se producen dos tipos de detección NAT antes de que comience el modo rápido IKE: compatibilidad con NAT y existencia de NAT a lo largo de la ruta de la red. Para detectar el soporte de NAT, la cadena de identificación del proveedor se intercambia con el par remoto. El vecino remoto envía una carga útil de cadena ID de proveedor a su par para indicar que esta versión es compatible con NAT-T. A partir de entonces, se puede determinar la existencia de NAT a lo largo de la ruta de red
    - NAT-T habilita un dispositivo IPsec para encontrar cualquier dispositivo NAT entre dos pares IPsec. Para detectar si existe un dispositivo NAT a lo largo de la ruta de la red, los pares envían una carga útil con hash de la dirección IP y el puerto de la dirección de origen y destino de cada extremo. Los hash se envían como una serie de cargas útiles NAT *discovery* (NAT-D). Si, al recibirla, ambos extremos vuelven a calcular los valores hash y los valores hash coinciden con el hash de carga útil, cada vecino sabe que no existe ningún dispositivo NAT en la ruta de red entre ellos. Si el *hash* de la carga útil y los *hashes* recalculados no coinciden (es decir, si alguien ha traducido la dirección o el

puerto), entonces cada par debe realizar NAT-T para transportar el paquete IPsec a través de la red.

- Decisión de NAT-T: mientras que la fase 1 de IKE detecta el soporte de NAT y la existencia de NAT a lo largo de la ruta de la red, la fase 2 de IKE decide si los pares en ambos extremos usarán NAT-T. La carga útil SA de modo rápido se usa para la negociación NAT-T.

- Encapsulación UDP de paquetes IPsec: además de permitir que los paquetes IPsec atraviesen los dispositivos NAT, la encapsulación UDP también soluciona muchos problemas de incompatibilidad entre IPsec, NAT y PAT. Los problemas resueltos son los siguientes:

    - Incompatibilidad entre IPsec ESP y PAT: Si PAT encuentra una dirección IP y un puerto, PAT descarta el paquete ESP. Para evitar que un paquete ESP caiga, la encapsulación UDP se usa para ocultar el paquete ESP como un paquete UDP, procesando el paquete ESP como un paquete UDP normal.

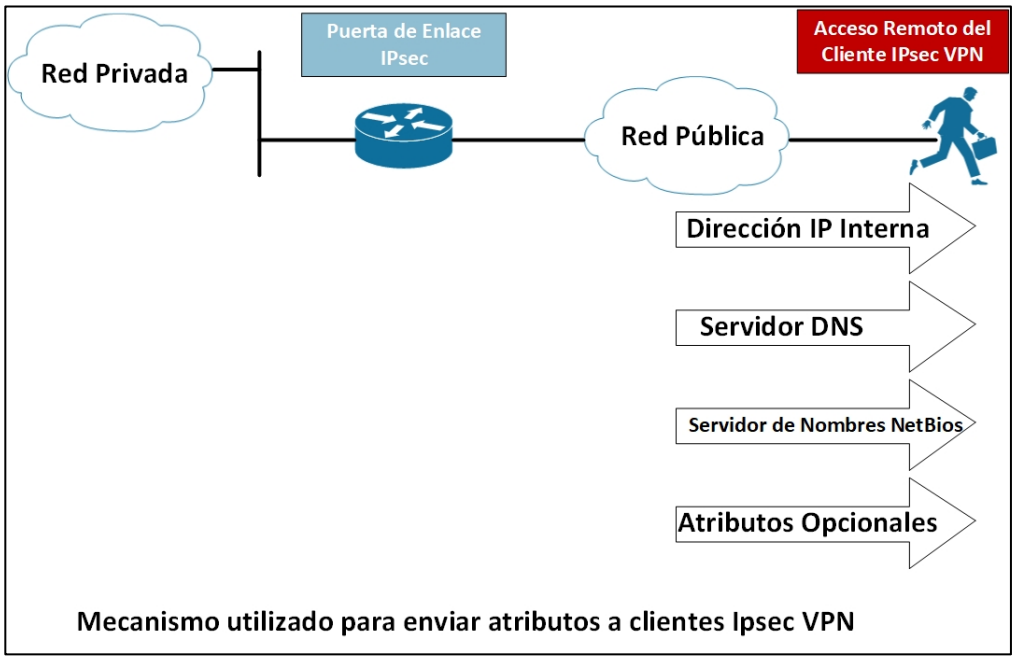
    - Incompatibilidad entre checksums y NAT: en el nuevo encabezado UDP, el valor de la suma de comprobación es siempre 0. Este valor evita que un dispositivo intermedio valide la suma de comprobación contra la suma de comprobación del paquete. El problema de suma de comprobación se resuelve porque NAT cambia las direcciones de origen y destino de IP.

    - Incompatibilidad entre los puertos de destino IKE fijos y PAT: PAT cambia la dirección del puerto en el nuevo encabezado UDP para la traducción y no modifica la carga original.

- Proceso encapsulado UDP para motores de *software*: modo de transporte y encapsulación ESP en modo túnel. Después de que el paquete de IPsec esté encriptado por un acelerador de *hardware* o un motor criptográfico de *software*, se insertan un encabezado UDP y un marcador no IKE (que tiene una longitud de 8 bytes) entre el encabezado IP original y el encabezado ESP. Los campos longitud total, protocolo y suma de comprobación se cambian para coincidir con esta modificación.

Los *keepalives* NAT se pueden usar para mantener vivo el mapeo NAT dinámico durante una conexión entre dos pares. Los keepalives NAT son paquetes UDP con una carga útil no encriptada de 1 byte. Por defecto, no se envían keepalives NAT.

Figura 192. **Opción modo de configuración**



Fuente: elaboración propia, empleando Visio 2013.

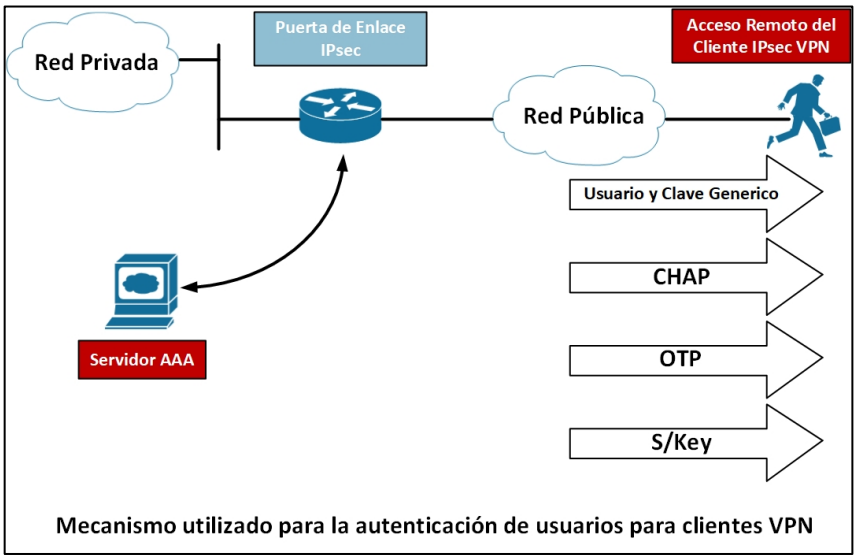
La configuración de modo es una opción para impulsar los parámetros del sistema (por ejemplo, la dirección IP y los atributos de DNS) al par, que generalmente es el cliente en una VPN de acceso remoto.

La opción de configuración de modo se usa ampliamente para Easy VPN. Easy VPN permite a los clientes remotos recibir políticas de seguridad de un servidor Cisco *Easy* VPN, lo que minimiza los requisitos de configuración en el cliente.

3.6.25. Autenticación extendida

En la autenticacion extendida podemos observar el funcionamiento como Xauth funciona mediante la ilustracion que se presenta a continuacion.

Figura 193. **Xauth**



Fuente: elaboración propia, empleando Visio 2013.

Xauth se basa en el protocolo IKE. Xauth permite que los métodos de autenticación, autorizacion y contabilidad (AAA) realicen la autenticación de usuario en una frase separada después del intercambio de fase 1 de autenticación IKE.

Xauth no reemplaza IKE. IKE permite la autenticación del dispositivo, mientras que *Xauth* permite la autenticación del usuario, que ocurre después de la autenticación del dispositivo IKE. Una opción de autenticación de usuario puede ser un nombre de usuario y una contraseña genéricos, el protocolo de autenticación por desafío mutuo (CHAP, *Challenge Handshake Authentication Protocol*), las contraseñas de un solo uso (OTP, One-time password) o la clave segura (S/Key, *Secure Key*).

3.6.26. Protocolos de ESP y AH, transporte y modos de túnel

Estos dos protocolos IP se usan en el estándar IPsec:

- ESP: el encabezado ESP (protocolo IP 50) forma el núcleo del protocolo IPsec. Este protocolo, junto con un método de cifrado acordado o conjunto de transformación, protege los datos al hacer que los datos sean indescifrables. Este protocolo protege solo la porción de datos del paquete. Este protocolo también puede proporcionar, opcionalmente, la autenticación de los datos protegidos.

- AH: la otra parte de IPsec está formada por el protocolo AH (protocolo IP 51). El AH no protege los datos en el sentido habitual al ocultar los datos, sino al agregar un sello de inviolabilidad a los datos. Este protocolo también protege los campos en el encabezado IP que contiene los datos,

incluidos los campos de dirección del encabezado IP. El protocolo AH no debe usarse solo cuando se requiere confidencialidad de datos.

IPsec tiene dos métodos para reenviar datos a través de una red, modo de túnel y modo de transporte, que difieren en su aplicación y en la cantidad de sobrecarga añadida al paquete de pasajeros:

- Modo de túnel: el modo túnel funciona encapsulando y protegiendo un paquete IP completo. Como el modo túnel encapsula u oculta el encabezado IP del paquete, se debe agregar un nuevo encabezado IP para que el paquete se reenvie correctamente. Los dispositivos de cifrado poseen las direcciones IP que se usan en este nuevo encabezado. Estas direcciones se pueden especificar en la configuración de los *routers*. El modo túnel se puede usar con ESP o AH o con ambos. El modo de túnel da como resultado una expansión de paquete adicional de aproximadamente 20 *bytes* debido al nuevo encabezado de IP.

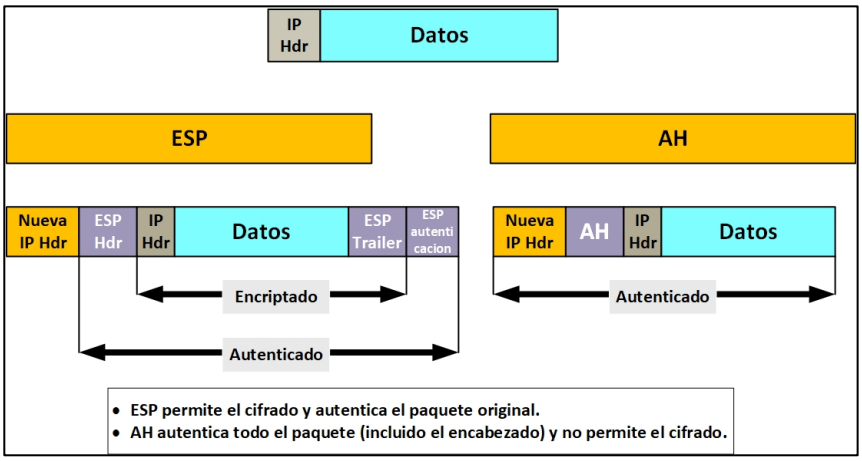
- Modo transporte: debido a que la expansión de paquetes puede ser una preocupación durante el reenvío de paquetes pequeños, también es posible un segundo método de reenvío. El modo de transporte IPsec funciona insertando el encabezado ESP entre el encabezado IP y el próximo protocolo o la capa de transporte del paquete. Ambas direcciones IP de los dos nodos de red cuyo tráfico está siendo protegido por IPsec son visibles. Este modo de IPsec a veces puede ser susceptible de análisis de tráfico. Sin embargo, dado que no se agrega un encabezado IP adicional, el resultado es una menor expansión de paquetes. El modo de transporte se puede implementar con ESP o AH o con ambos. Este modo funciona bien con *Generic Routing Encapsulation*

(GRE) porque GRE ya oculta las direcciones de las estaciones finales al agregar un encabezado IP.

3.6.27. Encabezados ESP y AH

En la siguiente ilustracion se puede apreciar las etiquetas que se agregan al realizar encriptado como autenticacion.

Figura 194. **Encabezado ESP y AH**



Fuente: elaboración propia, empleando Visio 2013.

Puede lograr la autenticación AH aplicando una función de hash unidireccional al paquete, creando un *hash* o resumen de mensaje. El hash se combina con el texto y luego se transmite. Los cambios en cualquier parte del paquete que ocurra durante el tránsito son detectados por el receptor cuando el receptor realiza la misma función hash unidireccional en el paquete recibido y compara el valor del resumen del mensaje que el emisor ha proporcionado. El hash unidireccional también también implica el uso de una clave simétrica entre los dos sistemas, lo que significa que la autenticidad está garantizada.

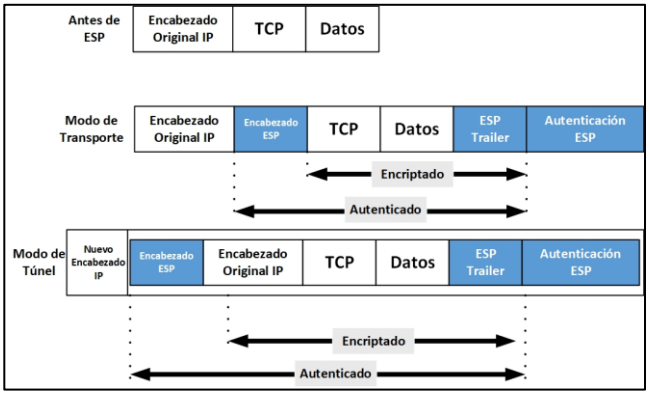
ESP proporciona confidencialidad al encriptar la carga útil. El algoritmo predeterminado para IPsec es DES de 56 *bits*. La mayoría de marcas admite el uso de 3DES para un cifrado más sólido.

Los algoritmos de cifrado ESP por sí mismos no proporcionan autenticación ni garantizan la integridad de los datos. El cifrado ESP con un servicio de autenticación e integridad de datos se puede lograr de dos maneras:

- Formato ESP autenticado
- ESP anidado dentro de AH

Con el ESP autenticado, IPsec cifra la carga usando una clave simétrica, luego calcula un valor de autenticación para los datos cifrados utilizando una segunda clave simétrica y el algoritmo HMAC-SHA1 o HMAC-MD5. El valor de autenticación ESP se adjunta al final del paquete. El destinatario calcula su propio valor de autenticación para los datos cifrados utilizando la segunda clave simétrica y el mismo algoritmo. El destinatario compara el resultado con el valor de autenticación transmitido. Si los valores coinciden, el destinatario descifra la porción cifrada del paquete con la primera clave simétrica y extrae los datos originales. La figura 195 muestra el ESP autenticado en modos de transporte y túnel.

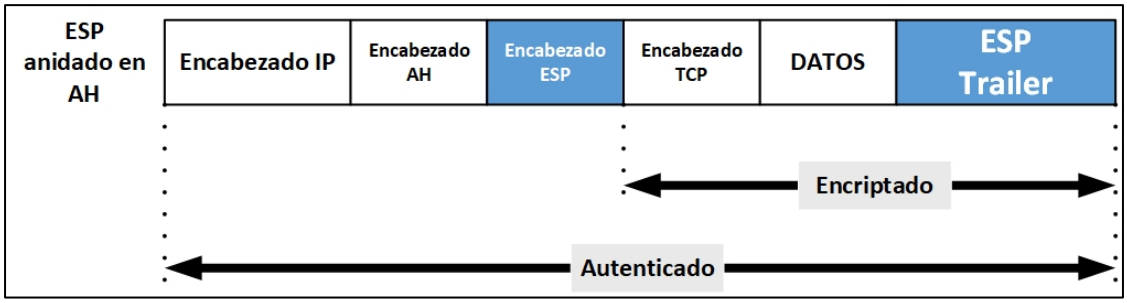
Figura 195. **Encriptación ESP**



Fuente: elaboración propia, empleando Visio 2013.

Un paquete ESP se puede anidar dentro de un paquete AH. Primero, la carga útil está encriptada. A continuación, la carga útil encriptada se envía a traves de un algoritmo *hash*: MD5 o SHA-1. El *hash* proporciona autenticación de origen e integridad de datos para la carga de datos. La figura 196 muestra el ESP anidado en AH utilizando el modo de transporte.

Figura 196. **ESP anidado en AH**

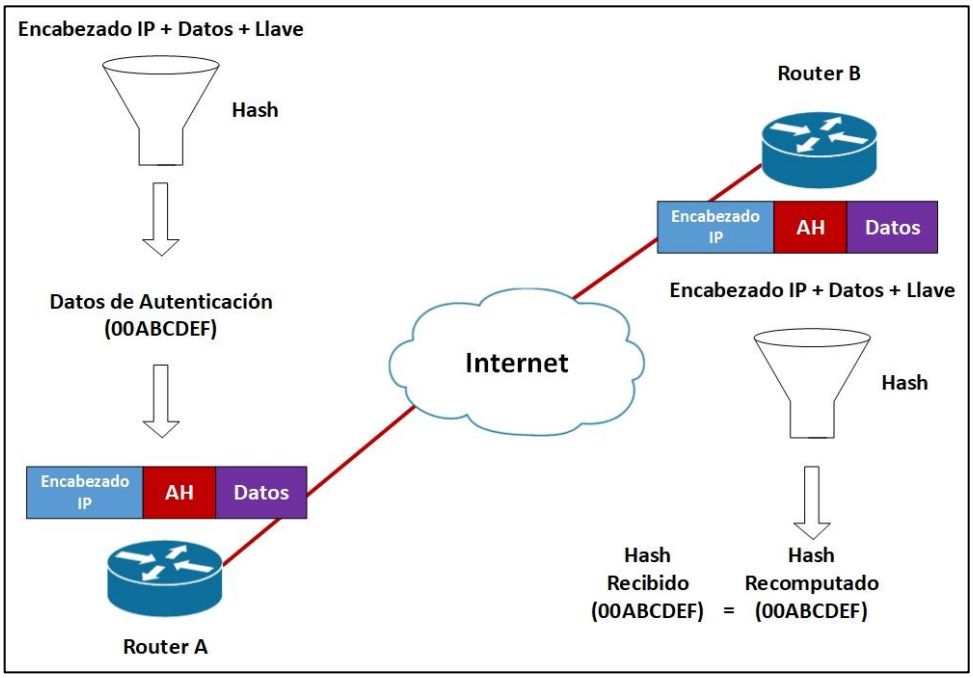


Fuente: elaboración propia, empleando Visio 2013.

3.6.28. Autenticación e integridad AH

La función AH se aplica a todo el datagrama, a excepción de cualquier campo de cabecera IP mutable que cambie en tránsito, como los campos *time to live* (TTL) que son modificados por los routers a lo largo de la ruta de transmisión. AH funciona de la siguiente manera y como se muestra en la figura 197.

Figura 197. **Autenticación e integridad AH**



Fuente: elaboración propia, empleando Visio 2013.

- Paso 1 el encabezado IP y la carga útil de datos son *hash*.

- Paso 2 el *hash* se usa para construir un encabezado AH, que se encarga al paquete original.

- Paso 3 el nuevo paquete se transmite al *router* para IPsec.

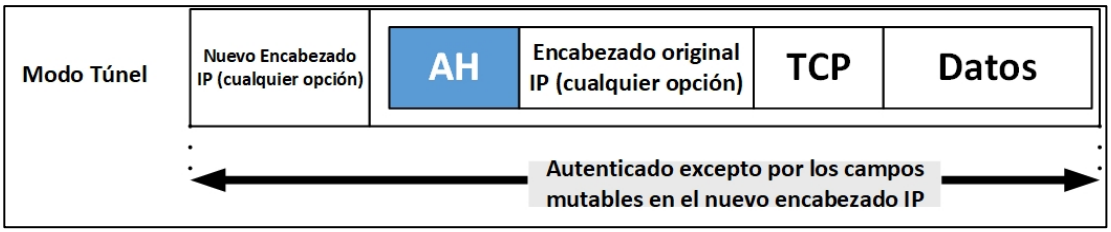
- Paso 4 el *router* par codifica el encabezado IP y la carga de datos.

- Paso 5 el *router* par extrae el *hash* transmitido del encabezado AH.

- Paso 6 el *router* par compara los dos *hashes*. Los *hashes* deben coincidir exactamente. Incluso si se cambia un *bit* en el paquete transmitido, la salida *hash* en el paquete recibido cambiará y el encabezado AH no coincidirá.

AH es compatible con los algoritmos MD5 y SHA-1. La figura 198 muestra un formato de cuadro del encabezado de autenticación (AH) en modo túnel.

Figura 198. **Formato de trama AH en modo túnel**



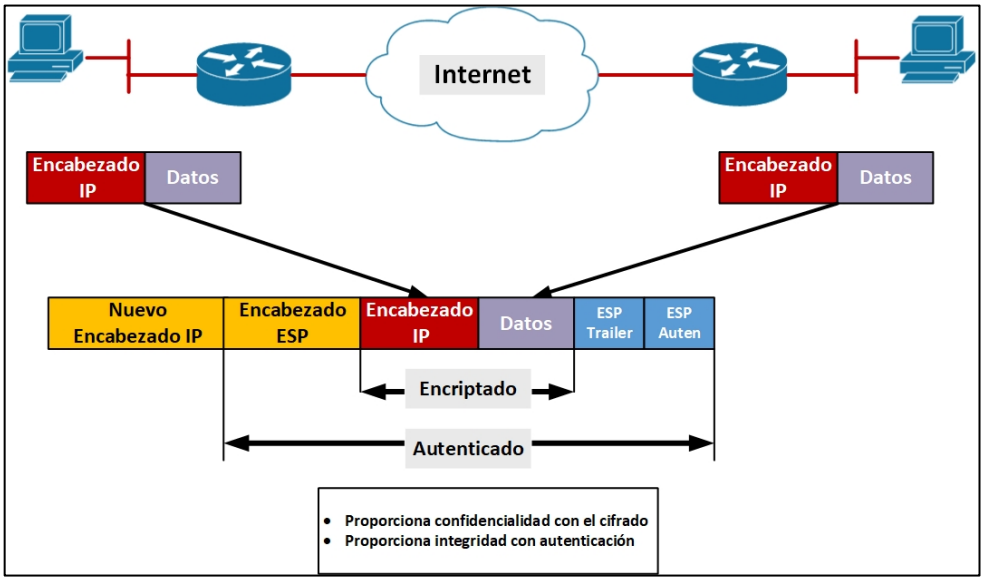
Fuente: elaboración propia, empleando Visio 2013.

3.6.29. Protocolo ESP

Entre dos puertas de enlace de seguridad, la carga útil original está bien protegida porque todo el datagrama de IP original está encriptado. Un encabezado ESP y un avance se agregan a la carga cifrada. Con autenticación ESP, el datagrama IP encriptado y el encabezado o el trailer ESP están

incluidos en el proceso hash. Por último, se encarga un nuevo encabezado IP al frente de la carga autenticada. La nueva dirección IP se usa para enrutar el paquete a través de internet.

Figura 199. **Protocolo ESP**



Fuente: elaboración propia, empleando Visio 2013.

Cuando se seleccionan la autenticación ESP y el cifrado, el cifrado se realiza primero antes de la autenticación. La base para este orden de procesamiento es facilitar la detección rápida y el rechazo de paquetes reprogramados o falsos por parte del nodo receptor. Antes de descifrar el paquete, el receptor puede autenticar los paquetes entrantes. Al autenticar primero los paquetes, el receptor puede detectar y potencialmente reducir el impacto de los ataques de denegación de servicio (DoS, *denial of service*).

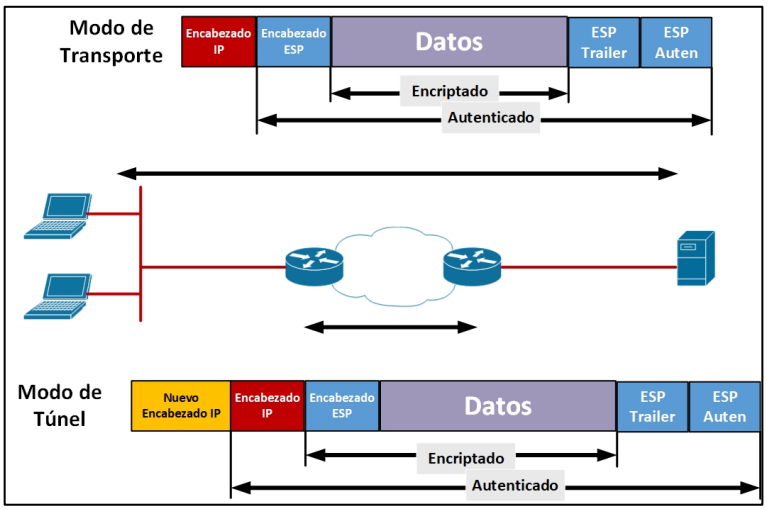
El modo de transporte es el modo predeterminado para IPsec. El modo de transporte solo protege la carga útil del paquete y los protocolos de capa

superior, pero deja desprotegida la dirección IP original. La dirección IP original se usa para enrutar el paquete a través de internet. El modo de transporte ESP se usa entre dos *host*.

Cuando se utiliza el modo de túnel IPsec, IPsec cifra el encabezado IP y la carga útil. El modo túnel proporciona la protección de un paquete IP completo al tratar el paquete como una carga útil AH o ESP. Con el modo túnel, todo un paquete IP se encapsula con un encabezado AH o ESP y un encabezado IP adicional. El modo de túnel ESP se utiliza entre un *host* y una puerta de enlace de seguridad o entre dos puertas de enlace de seguridad. Para las aplicaciones de puerta de enlace a puerta de enlace, en lugar de cargar IPsec en todas las computadoras de las oficinas remotas y corporativas, es más fácil que las pasarelas de seguridad realicen la encriptación de IP en IP.

En la aplicación de acceso remoto IPsec, se usa el modo de túnel ESP. En una oficina hogareña, puede que no haya un *router* o un *firewall* para realizar el encapsulado y el cifrado de IPsec. En el ejemplo de la figura 200, el cliente IPsec que se ejecuta en la PC realiza la encriptación y encriptación IP-in-IP de IPsec. En la oficina corporativa, el *router* desencapsula y descifra el paquete.

Figura 200. **Tunel ESP y modos de transporte**



Fuente: elaboración propia, empleando Visio 2013.

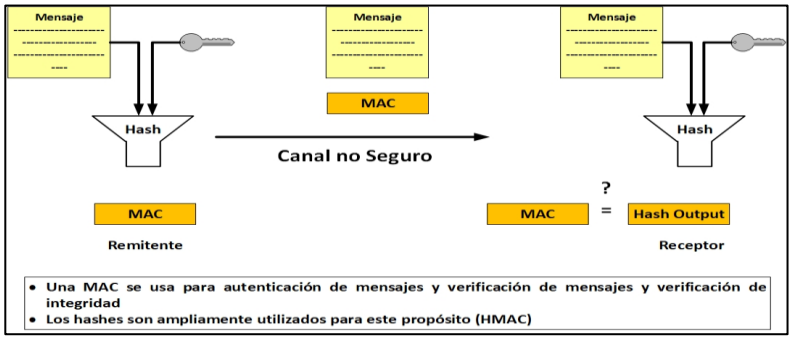
3.6.30. Autenticación de mensajes y verificación de integridad

Los datos VPN se pueden transportar a través de internet público. Potencialmente, esta información podría ser interceptada y modificada. Para protegerse contra la posibilidad de interceptación, cada mensaje tiene un *hash* adjunto al mensaje. Un *hash* proporciona una forma de garantizar la integridad del mensaje original: si el *hash* transmitido coincide con el hash recibido, el mensaje no se ha alterado. Sin embargo, si los valores *hash* no coinciden, el mensaje se modificó.

El HMAC se usa para autenticación de mensajes y verificación de integridad. HMAC se puede usar con cualquier función *hash* criptográfica iterativa, como MD5 o SHA-1, en combinación con una clave secreta compartida. La fuerza criptográfica de HMAC depende de las propiedades de las propiedades de la función *hash* subyacente. HMAC también usa una clave

secreta para calcular y verificar los valores de autenticación del mensaje. MD5 y SHA-1 son ejemplos de tales funciones *hash*.

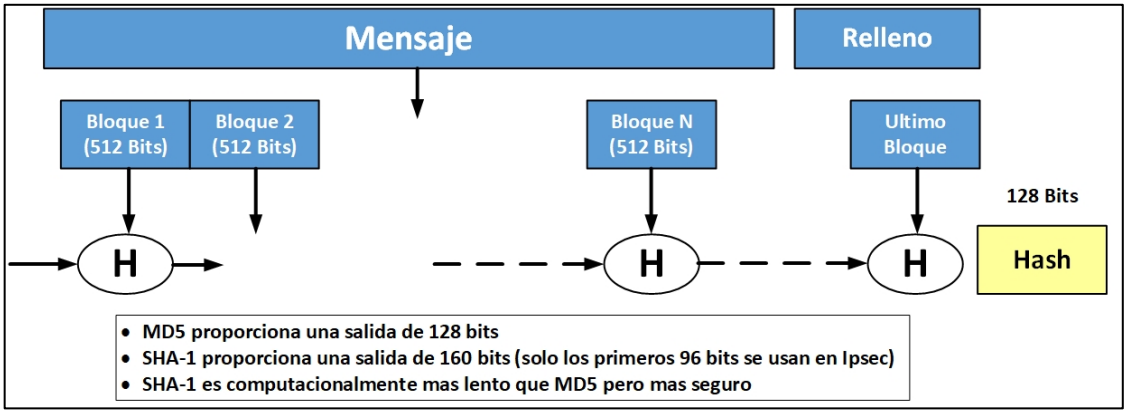
Figura 201. **Mensaje de autenticación y chequeo de integridad usando hash**



Fuente: elaboración propia, empleando Visio 2013.

Las dos funciones hash IPsec comúnmente usadas se muestran en la figura 202. MD5 es bien conocido por diversos usos en los componentes de distintas marcas, como contraseñas hash en el software de distintas marcas. Ambas funciones *hash* toman un mensaje de entrada de longitud variable y crean un *hash* de longitud fija.

Figura 202. **Funciones Hash utilizadas comúnmente**



Fuente: elaboración propia, empleando Visio 2013.

MD5 crea un *hash* de 128 *bits*, mientras que SHA-1 crea un *hash* de 160 *bits*. En el caso de SHA-1, solo 96 *bits* de este hash se utilizan para IPsec.

El vector de inicialización (IV) se usa como valor inicial para comenzar a crear un hash.

3.6.31. Entorno PKI

Una infraestructura de clave pública (PKI) permite a los usuarios de internet intercambiar de forma segura y privada datos y dinero utilizado un par de claves criptográficas públicas y privadas que se obtienen y comparten a través de una autoridad de confianza. Una PKI proporciona un marco jerárquico para administrar los atributos de seguridad digital de las entidades que participaran en comunicaciones seguras. Además de los usuarios humanos, existen *gateways* de encriptación, servidores web seguros y otros recursos que requieren un control cercano de la identidad y el cifrado.

El entorno de PKI a veces puede aparecer como un rompecabezas. Una PKI se compone de estas entidades:

- Pares que se comunican en una red segura.

- Al menos una autoridad de certificación (CA) para otorgar y mantener certificados.

- Certificados digitales, que contienen información como el periodo de validez del certificado, información de identidad de pares, claves de cifrado que se usan para comunicación seguras y la firma de la CA emisora.

- Una autoridad de registro (RA) opcional para descargar la CA procesando las solicitudes de inscripción (la inscripción del certificado es el proceso de obtener un certificado de una CA).

- Un mecanismo de distribución, como el protocolo ligero de acceso a directorios (LDAP, *lightweight directory access protocol*) o HTTP, para listas de revocación de certificados (CRL, *certificate revocation list*).

PKI proporciona a los clientes un mecanismo escalable y seguro para distribuir, administrar y revocar el cifrado y la información de identidad en una red de datos segura. Cada entidad (una persona o dispositivo) que participa en las comunicaciones seguras se inscribe en la PKI en un proceso en el que la entidad genera un par de claves RSA (una clave privada y una clave pública) y su identidad validada por una entidad confiable (también conocido como CA o punto de confianza).

Después de inscribirse en una PKI, a cada par (también conocido como *host* final) en una PKI se le otorga un certificado digital que ha sido emitido por una CA. Cuando los pares deben de negociar una sesión de comunicación segura, intercambian certificados digitales. De acuerdo con la información en el certificado, un par puede validar la identidad de otro par y establecer una sesión encriptada con las claves públicas que están contenidas en el certificado.

3.6.32. Autoridad certificadora

Una CA, o punto de confianza, administra las solicitudes de certificados y emite certificados a los dispositivos de red participantes. La gestión de solicitudes de certificados y la emisión de certificados proporcionan una gestión centralizada de claves para los dispositivos participantes. La CA cuenta con la confianza explícita del receptor para validar identidades y crear certificados digitales. Antes de que las operaciones PKI puedan comenzar, la CA autofirmado; a partir de entonces, la CA puede firmar solicitudes de certificados y comenzar la inscripción de pares para la PKI.

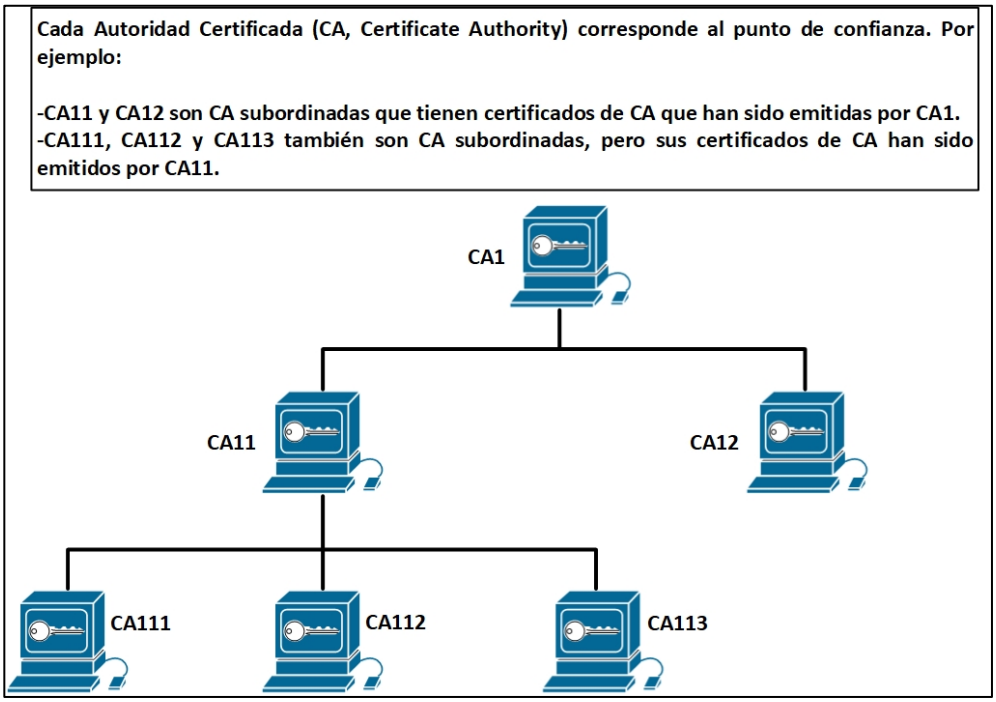
Puede usar una CA proporcionada por un proveedor de CA externo, o puede usar una CA interna, que es el servidor de certificados de distintas marcas.

3.6.33. PKI jerárquica: múltiples CA

Puede configurar una PKI en un marco jerárquico para admitir múltiples CA. En la parte superior de la jerarquía hay una CA raíz, que contiene un certificado autofirmado. La confianza dentro de toda la jerarquía se deriva del par de claves RSA de la CA raíz. Las CA subordinadas dentro de la jerarquía pueden inscribirse con la CA raíz o con otra CA subordinada. Estas opciones de

inscripción permiten configurar varios niveles de CA. Dentro de una PKI jerárquica, todos los pares inscritos pueden validar el certificado de los demás si los pares comparten un certificado de CA raíz confiable o una CA subordinada común.

Figura 203. **Topología de jerarquía de tres niveles**



Fuente: elaboración propia, empleando Visio 2013.

Múltiples CA proporcionan a los usuarios mayor flexibilidad y confiabilidad. Por ejemplo, las entidades emisoras subordinadas pueden ubicarse en sucursales mientras la entidad emisora de certificados raíz se encuentra en la oficina central. Además, se pueden implementar diferentes políticas de concesión por CA, por lo que puede configurar una CA para otorgar

automáticamente solicitudes de certificados, mientras que otra CA dentro de la jerarquía requiere que cada solicitud de certificado se otorgue manualmente.

Hay dos escenarios en los que se recomienda al menos una CA de dos niveles:

- Redes grandes y muy activas en las que se revocan y vuelven a emitir una gran cantidad de certificados. Una CA de múltiples niveles ayuda a controlar el tamaño de las CRL (*certificate revocation list*).

- Cuando se utilizan protocolos de inscripción en línea, la CA raíz puede mantener fuera de línea con la excepción de emitir certificados de CA subordinados. Este escenario proporciona seguridad adicional para la CA raíz.

3.6.34. Certificado X.509 v3

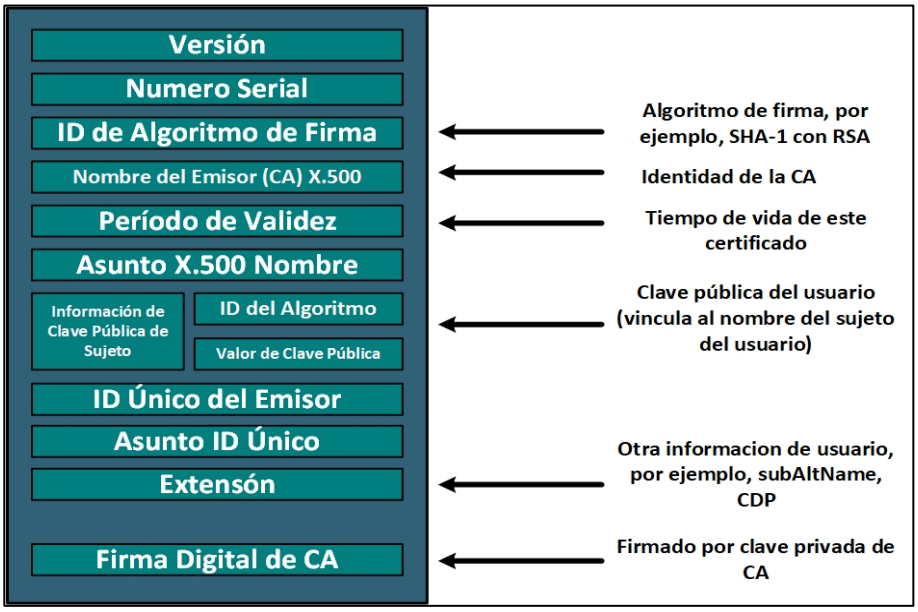
Los certificados se pueden usar para el uso a gran escala de la criptografía de clave pública. El intercambio seguro de claves secretas entre los usuarios resulta poco práctico para redes grandes.

Un certificado puede revocarse si se descubre que la clave privada relacionada con el certificado está comprometida o si se descubre que la relación entre una entidad y una clave pública, incrustada en el certificado, es incorrecta o ha cambiado. Cualquiera de estas corrupciones puede ocurrir, por ejemplo, si una persona cambia de trabajo o nombre. Una revocación es una ocurrencia rara, pero la posibilidad de revocación significa que incluso cuando se confía en un certificado, en usuario siempre debe verificar su validez. Puede verificar la validez del certificado comparando el certificado con una CRL, una

lista de certificados revocados o cancelados. Garantizar que dicha lista está actualizada y sea precisa es una función central en una PKI centralizada. Para ser efectivo, el certificado debe estar disponible para todos y debe actualizarse con frecuencia. La otra forma de comprobar la validez del certificado es consultar a la CA utilizando el protocolo de estado de certificado en línea (OCSP) para conocer el estado de un certificado específico.

La estructura de un certificado digital X.509 v3 se muestra en:

Figura 204. **Certificado X.509 V3**



Fuente: elaboración propia, empleando Visio 2013.

- Certificado
    - Versión
    - Número Serial

- ID de algoritmo
- Issuer
- Validez
    - No antes
    - No después
- Subject Public Key Info
    - Algoritmo de clave pública
    - Clave pública de sujeto
- Identificar individual del identificador (opcional)
- Identificar único de sujeto (opcional)
- Extensiones (opcional)

- Algoritmo de firma de certificado
- Firma del certificado

3.6.35. Intercambio de mensajes PKI

La inscripción del certificado es el proceso de obtener un certificado de una CA. Cada *host* final que se desee participar en la PKI debe obtener un certificado.

La inscripción de certificados se produce con estos pasos entre el *host* final que solicita el certificado y la CA, y como se muestra en:

- Paso 1: el host final genera un par de claves RSA y solicita la clave pública de la CA.

- Paso 2: la CA envía la clave pública de CA al host final

- Paso 3: el host final genera una solicitud de certificado y reenvía la solicitud a la CA (o a la RA, si corresponde). La CA recibe la solicitud de inscripción del certificado y, dependiendo de su configuración de red, se produce una de las siguientes opciones:

    - Se requiere la intervención manual para aprobar la solicitud.

    - El host final está configurado para solicitar automáticamente un certificado de la CA. Por lo tanto, la intervención del operador ya no es necesaria en el momento en el que se envía la solicitud de inscripción al servidor de CA.

- Paso 4: después de que se aprueba la solicitud, la CA firma la solicitud con la clave privada de la CA.

- Paso 5: la CA devuelve el cifrado completo al *host* final. El *host* final escribe el certificado en un área de almacenamiento NVRAM.

- Paso 6: el *host* final usa el certificado para comunicarse con otros socios de comunicación.

3.6.36. Credenciales de PKI

En la siguiente descripción se muestra cómo las credenciales de PKI, cómo las claves RSA y los certificados, se pueden almacenar en una ubicación distinta de NVRAM, la ubicación predeterminada en el *router*.

La mayoría de las plataformas de distintas marcas ahora son compatibles con la tecnología de tarjetas inteligentes en forma de clave de bus serial universal (USB, *universal serial bus*) (también conocida como clave *eToken* USB de Aladdin). Un *eToken* proporciona una distribución de configuración segura y permite a los usuarios almacenar credenciales de VPN para utilizarlas en el despliegue.

Un eToken es una tarjeta inteligente con una interfaz USB. El *eToken* puede almacenar de forma segura cualquier tipo de archivo dentro del espacio de almacenamiento de 32 KB disponible. Los archivos de configuración que están almacenados en el *eToken* se pueden encriptar y acceder solo a través de un número de identificación personal (PIN, *personal identification number*) del usuario. El *router* no cargará el archivo de configuración a menos que se haya configurado el PIN apropiado para la implementación segura de los archivos de configuración del *router*.

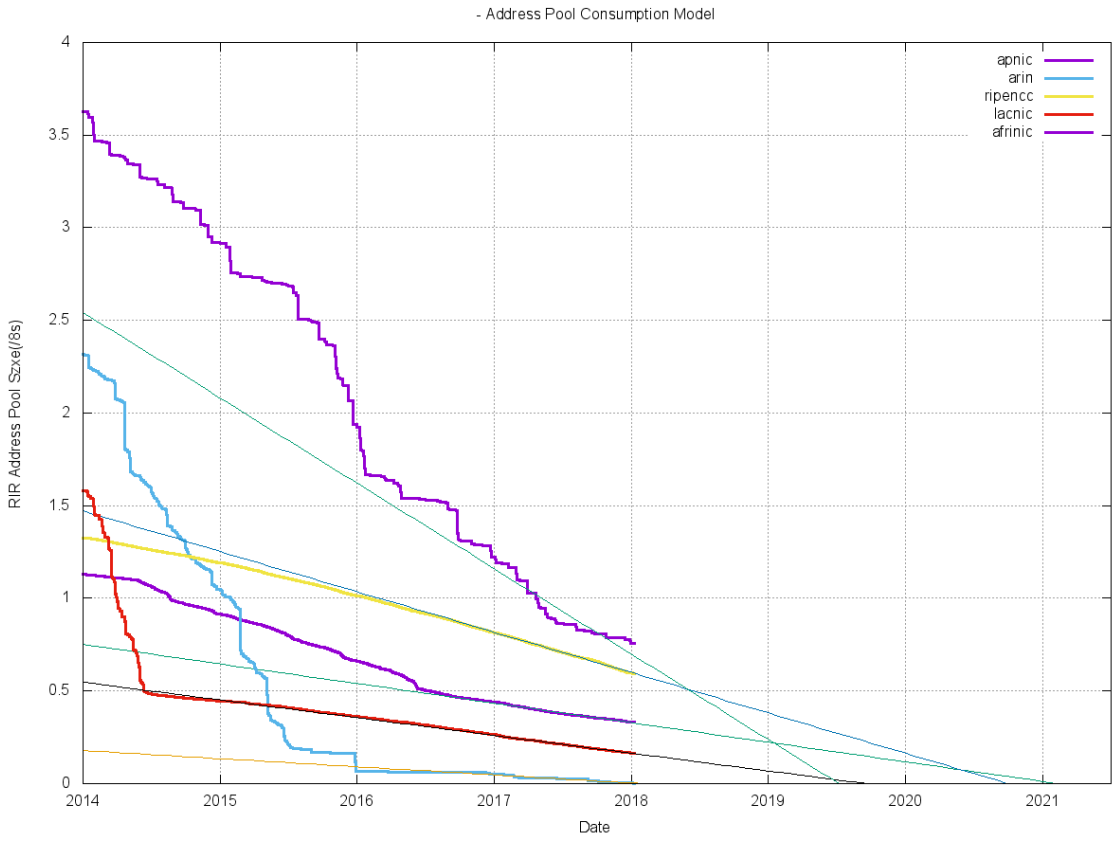
Después de enchufar el *eToken* en el router, debe iniciar sesión en el eToken. Una vez que haya iniciado sesión, puede cambiar la configuración predeterminada, como el PIN de usuario (el valor predeterminado es 1234567890) y la cantidad permitida de intentos de inicio de sesión fallidos antes de que se rechacen los inicios de sesión futuros (el valor predeterminado es 15 intentos).

Después de que haya iniciado sesión correctamente en *eToken*, puede copiar archivos desde el router al *eToken* a través del comando de copia. De forma predeterminada, después de que *eToken* se elimina del *router*, se eliminan todas las claves RSA asociadas; Los túneles IPsec no se destruyen hasta que se inicia el siguiente periodo de negociación IKE.

3.7. IPv6

Es una version de protocolo de internet (Internet Protocol IP), que a partir de 2016 se ha estado implementando en varios dispositivos a nivel mundial, lo cual se dejara de utilizar IPv4. Este protocolo fue diseñado por Steve Deering de Xerox y Craig Mudge. Este protocolo tiene la caracteristica principal que utiliza parte del identificador MAC y es una direccion en forma hexadecimal.

Figura 205. **Modelo de consumo de direcciones IPv4 por regiones a nivel mundial**



Fuente: *Modelo de consumo de direcciones IPv4*.
https://www.potaroo.net/ispcol/201801/addrfig1.png. Consulta: 29 de octubre de 2018.

El 31 de enero del 2011, el IANA asignó los últimos dos bloques de direcciones sin reservas. Como puede ver en la figura # 205, con respecto a todos los continentes, ahora se está experimentando una escasez de direcciones IPv4.

Esta es la razón por la cual necesitamos IPv6. Durante los últimos diez años, hemos visto suficientes mensajes en las noticias de los que nos estamos

quedando sin espacio IPv4 y se necesita cambiar a IPv6. Hoy en día se tienen muchos dispositivos móviles como teléfonos, tabletas y todos los que necesitan conexión a internet. Se necesitará más espacio si se necesita un crecimiento futuro en internet.

IPv6 es completamente diferente a IPv4 y los protocolos no son compatibles entre si. Esto significa que necesitaremos un plan de migración para pasar de IPv4 a IPv6.

Además de más espacio de direcciones, IPv6 cubre una serie de características nuevas:

- Funciones de asignación de direcciones: hay nuevas formas de asignar direcciones IPv6 a dispositivos de red. DHCP se ha actualizado, pero también hay algo nuevo llamado autoconfiguración sin estado.

- Remuneración de direcciones: Renumerar una red IPv4 no es tan divertido volver a numerar una red IPv6 es más fácil porque hay un método para cambiar su prefijo IPv6 con bastante facilidad.

- Compatibilidad con movilidad: IPv6 tiene funciones para que un dispositivo móvil pueda moverse por la red y mantener la misma dirección IPv6 donde sea que se encuentre.

- No es necesario NAT/PAT: se tiene tanto espacio de direcciones que ya no se necesita ninguna traducción de direcciones de red o puertos.

- IPSEC está integrado en el protocolo.

- IPv6 tiene encabezados mucho más pequeños de IPv4.

- Sin tráfico de difusión: IPv6 ya no usa ninguna transmisión de capa 3.

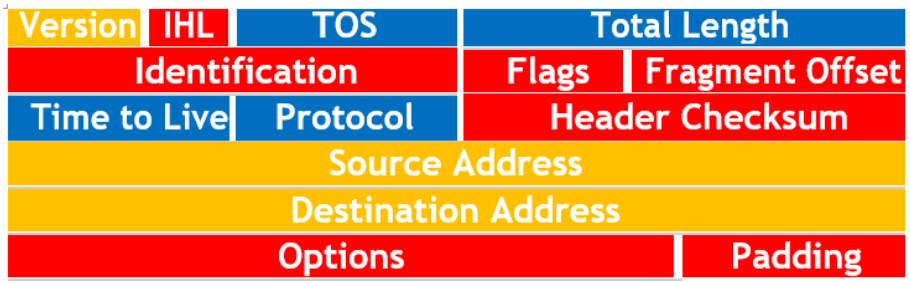
- Herramientas de migración: IPv4 e IPv6 no son compatibles entre sí, por lo que necesitamos herramientas para que la migración sea agradable y fácil.

Existió el protocolo IPv5, lo cual fue un proyecto experimental llamado *Internet Stream Protocol*. Está definido en el RFC 1819, IPv6 tiene direcciones de 128 *bits* en comparación con nuestras direcciones IPv4 de 32 bits. Por lo que cada *bit* adicional duplica la cantidad de direcciones IP, así que de tener 4 mil millones a 8 mil millones, 16, 32, 64, etc. se sigue duplicando hasta llegar a 128 bits. Lo cual IPv6 nos brindará:

- 340-undecillión
- 282 decillones
- 366 nonillón
- 920 octillión
- 938 septillon
- 463 sextillones
- 463 quintillones
- 374 cuatrillones
- 607 billones
- 431 mil millones
- 768 millones
- 211 mil
- 456

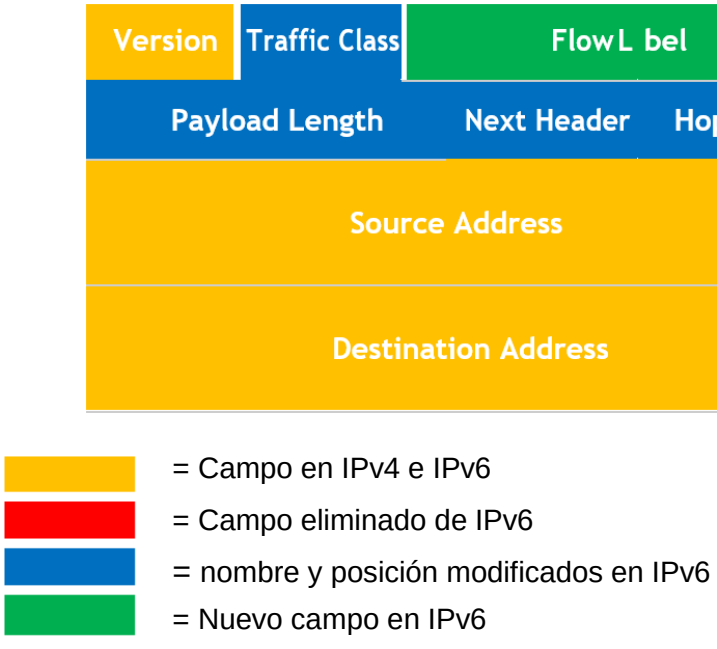
Lo cual nos deja con muchas direcciones IP. Parece que tomará un tiempo antes de que se quede sin espacio de direcciones.

Figura 206. **Encabezado IPv4**



Fuente: elaboración propia, empleando Visio 2013.

Figura 207. **Encabezado IPv6**



Fuente: elaboración propia, empleando Visio 2013.

Se puede observar que existe una gran diferencia entre los encabezados IPv4 e IPv6. El encabezado IPv6 no tiene tantos campos, pero en el campo *next header* (siguiente encabezado). IPv6 funciona un tanto diferente, en lugar de colocar todo en un encabezado se puede trabajar con encabezados de extensión. Esto mantiene el encabezado limpio y simple. La notación IPv6 tiene las siguientes características:

- X: X: X: X: X: X: X: X: donde X es un campo hexadecimal de 16 bits
- Insensible a mayúsculas y minúsculas

Lo cual tendrá una similitud como en se observa a continuación:

2041:0000:140F:0000:0000:0000:875B:131B

Pero en el caso en el cual se desea realizar pruebas como lo es un *ping* o proporcionar la dirección como dato a algún usuario, existe el poder acortar las direcciones IPv6.

- Original: 2041:0000:140F:0000:0000:0000:875B:131B
- Corto: 2041:0000:140F: :875B:131B

Si existiera una cadena de ceros, puede eliminarse de tal forma que pueden ser reemplazados con dos puntos (:). En la dirección IPv6 anterior se eliminaron los ceros haciendo que la dirección sea un poco más corta.

- Corto: 2041:0000:140F: :875B:131B
- Más corto: 2041: 0 :140F: :875B: 131B

Lo cual nos deja con estas reglas para poder representar de una forma más corta la dirección IPv6:

- Se puede eliminar una cadena de ceros dejando solo dos puntos (:).
- 4 ceros se pueden eliminar dejando solo un cero.

Una de las grandes diferencias entre IPv4 e IPv6 es que ya no tenemos las mismas transmisiones. Lo cual son las siguientes:

- *Unicast*: lo mismo que IPv4. Puede tener múltiples direcciones IPv6 en una interfaz.

- *Multicast*: se utiliza *multicast* incluso más para IPv6 ya que ya no tenemos tráfico de Broadcast.

- *Anycast*: este es nuevo para IPv6. Con *anycast* puedes tener la misma dirección IPv6 en múltiples *host* y los routers se aseguran de que se enrutado al destino más cercano. Se puede utilizar este mismo para equilibrar la carga.

IPv4 se tienen direcciones IP privadas y públicas y para IPv6 se tiene algo similar pero que trabaja un poco diferente:

- Local único
- Enlace local
- Unicast global

Local Único: es equivalente a las direcciones privadas IPv4. Estas son las direcciones que se deben usar para redes LAN y no en Internet. Se pueden reconocer porque el espacio de direcciones FD00 :: /8 está reservado para esto. Sin embargo, el espacio de direcciones IPv6 es tan grande que las organizaciones probablemente usarán direcciones IPv6 globales para sus LAN.

Enlace Local: las direcciones son algo nuevo. Cada dispositivo IPv6 tendrá una dirección local de enlace en la interfaz y tiene un ámbito de enlace local. Los paquetes enviados entre las direcciones locales del enlace permanecerán en el enlace y los *routers* no los reenviarán a otras subredes. Lo cual se utilizan para:

- Se usa como la dirección de origen para RS (*router solicitation*) y RA (*router advertisement*).
- Usado para el descubrimiento de vecinos (equivalente a ARP para IPv6)
- Se usa como la dirección IPv6 del siguiente salto para rutas de IP.

Las direcciones IPv6 de enlace local se generan automáticamente por su dispositivo IPv6 para cada interfaz. Puede reconocerlos ya que usan el rango FE80 :: /10.

Unicast Global: estas direcciones se usan en internet y, dado que el espacio de direcciones es tan grande, es probable que esto sea lo que utilizaremos para todo, incluso para redes LAN. En lugar de utilizar NAT/PAT, tendrá su propio espacio de direcciones IPv6 para usar. El espacio de direcciones que estamos utilizando para unicast global es 2000 :: /3.

Se encuentran dos tipos de direcciones IPv6 más que se pueden encontrar, las cuales son:

- No especificado
- *Loopback*

La dirección no especificada aparecerá como :: /128 y se usa cuando su *host* no tiene una dirección IPv6 utilizable. La de Loopback es la misma que 127.0.0.1 de IPv4, pero para IPv6 usamos :: 1/128.

El multicast se usa aún más en IPv6 y el espacio de direcciones reservado para esto es FF :: /8. Para hacer conocer de una forma más sencilla, algunas de las direcciones de multicast IPv6 se ven un poco similares a su equivalente IPv4.

Las direcciones multicast utilizadas en IPv6 son las siguientes:

Tabla VIII. **Direcciones multicast utilizadas en IPv6**

Propósito	Direcciones IPv6	Equivalente IPv4
Todos los nodos IPv6 en el enlace	FF02 :: 1	Direcciones de Broadcast
Todos los *routers* IPv6 en el enlace	FF02 :: 2	-
OSPF	FF02 :: 5 y FF02 :: 6	224.0.0.5 + 224.0.0.6
EIGRP	FF02 :: A	224.0.0.10
RIP-NG	FF02 :: 9	224.0.0.9
Agentes de retransmisión DHCP	FF02 : :1 : 2	-
Servidores DHCP	FF05 :: 1 : 3	-
Todos los servidores NTP	FF05 :: 101	-

Fuente: elaboración propia, empleando Visio 2013.

Se observa que OSPF, EIGRP y RIP-NG (Rip próxima generación para IPv6) se parecen un poco a su equivalente IPv4. EIGRP tiene FF02 :: A y se traduce A (hex) a decimal lo cual es 10.

Al configurar IPv6 se tienen más opciones que IPv4, donde se puede elegir entre manual o dinámico (DHCP).

Tabla IX. **Comparación de configuración IPv6**

Estatico	Dinámico
Manual	Autoconfiguración sin estado
EUI-64	DHCPv6

Fuente: elaboración propia, empleando Visio 2013.

Veanse primero los métodos estáticos. Se puede elegir en el manual o EUI-64

Figura 208. **Configuración IPv6**

```
Guatemala(config)#interface fastEthernet 0/0
Guatemala(config-if)#ipv6 address 2001:1234:5678:abcd:124:5678:abcd:1234/128
```

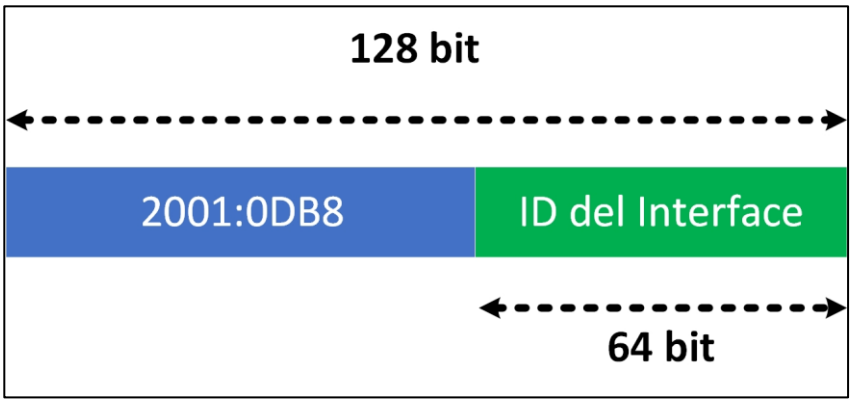
```
Guatemala#show ipv6 interface brief
FastEthernet0/0    [up/up] FE80::CE09:18FF:FE0E:0
   2001:1234:5678:ABCD:124:5678:ABCD:1234
```

Fuente: elaboración propia, empleando Visio 2013.

Si se utiliza el método del manual, simplemente se escribe la dirección IPv6 con el comando de dirección IPv6. Como se puede observar, ya no existe el colocar de una vez la máscara de subred, pero solo se utiliza la notación CIDR. Se puede utilizar el comando IPv6 para verificar sus direcciones. La mayoría de los comandos son los mismos para IPv6, solo se necesita colocar ipv6 en lugar de ip. Además de la dirección IPv6 unicast global, también se puede observar que el router genera una dirección IPv6 de enlace local para la interfaz.

En lugar de escribir 128 *bits* para la dirección IPv6, también se puede usar EUI-64 que también se puede utilizar para configurar manualmente las direcciones IPv6.

Figura 209. **Composición IPv6**



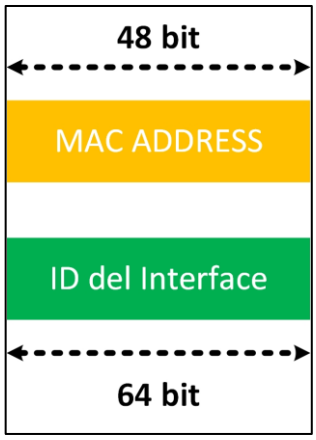
Fuente: elaboración propia, empleando Visio 2013.

Una dirección IPv6 es de 128 *bits*, pero consta de dos partes:

- El prefijo (equivalente a una dirección de red IPv4)
- ID de la interfaz (que identifica de forma única el *host*)

Una de las mejores características de IPv6 es que se puede tener la información de capa de enlace de datos (capa 2) en su dirección IPv6. Esto significa que se puede utilizar la dirección MAC como identificación de la interfaz cuando se utiliza *ethernet*. Las direcciones MAC son únicas, por lo que si se utiliza para la ID de la interfaz, esto nos dará una dirección IPv6 única.

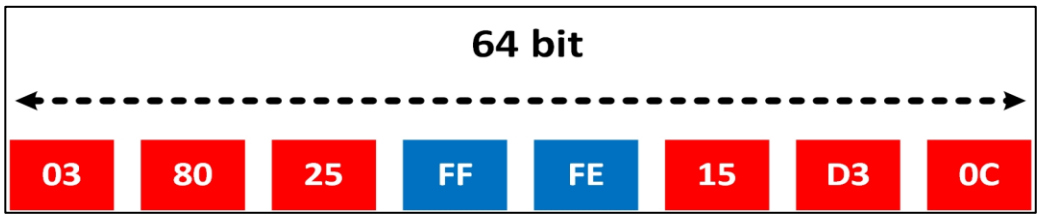
Figura 210. **Tamaño de dirección IPv6**



Fuente: elaboración propia.

Se observa que la dirección MAC es de 48 *bits* y la ID de la interfaz es de 64 *bits*, por lo que los *bits* que hacen falta para completar se especifican en la siguiente figura 210.

Figura 211. **Ejemplo de dirección IPv6**



Fuente: elaboración propia.

Solo se toma tu dirección MAC y se parte en dos. Se coloca FF: FE en el medio y se obtiene 64 bits. Así es como se crea una ID de interfaz y se obtiene una dirección IPv6 única.

Figura 212. **Configuración IPv6 en una interfaz de Router**

```
Miami(config)# interface fastEthernet 0/0
Miami(config-if)# ipv6 address 2001:1234: :/64 eui-64
```

Fuente: elaboración propia.

En esto se configura el router Miami con un prefijo IPv6 y se utilizó EUI-64 al final. Así es como se puede generar automáticamente la ID de la interfaz usando la dirección MAC.

Figura 213. **Colocación de dirección IPv6 con autocompletado**

```
miami#show interfaces fastEthernet 0/0 | include Hardware
  Hardware is AmdFE, address is cc0a.180e.0000 (bia cc0a.180e.0000)
```

```
miami#show interfaces fastEthernet 0/0 | include Hardware
miami#show ipv6 interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::CE0A:18FF:FE0E:0
  Global unicast address(es):
    2001:1234::CE0A:18FF:FE0E:0, subnet is 2001:1234::/64 [EUI]
```

Fuente: elaboración propia.

Se puede observar la dirección MAC de la interfaz FastEthernet 0/0 del *router* Miami. Si se utiliza el comando show ipv6 interface, se puede ver la dirección IPv6 completa con la dirección MAC como el ID de la interfaz. También se puede apreciar el FF: FE en el medio. EUI-64 siempre se utiliza para crear la dirección local de enlace.

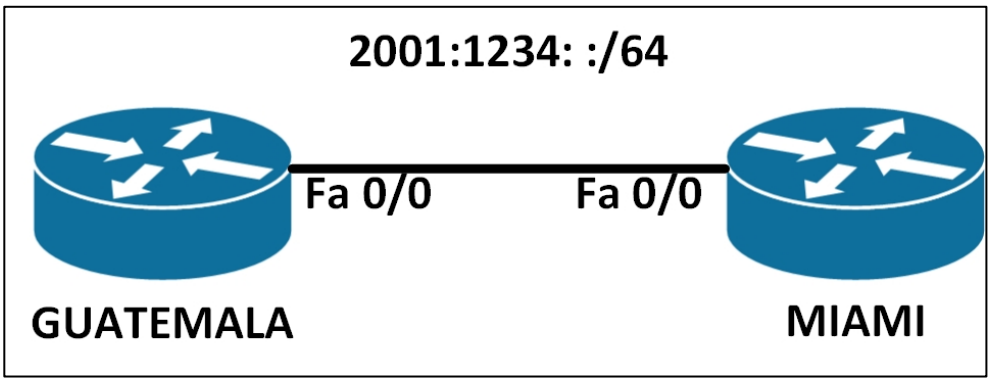
También se tienen métodos automáticos de configuración de direcciones IPv6, así que se puede apreciar la autoconfiguración sin estado y DHCPv6.

La autoconfiguración sin estado es nueva para IPv6 y otro método para proporcionar dispositivos de red.

Una dirección IPv6 automáticamente. Por lo cual se le llama a esto 'mini-dhcp'

IPv6 utiliza NDP (Neighbor *Discovery Protocol*) y una de las cosas que este protocolo ofrece es RS (*Router Solicitation*) y (RA) *Router advertisement* que son mensajes que ayudan al dispositivo IPv6 a configurar automáticamente una dirección IPv6.

Figura 214. **Utilización de NDP**



Fuente: elaboración propia.

Además de configurar una dirección IPv6, se debe de utilizar el comando ipv6 *unicast-routing* para hacer que el router de Miami actúe como un *router*. Este comando también lo necesitan los protocolos de enrutamiento.

Figura 215. **Comando Unicast-Routing**

```
miami(config)#ipv6 unicast-routing
miami(config)#interface fastEthernet 0/0
miami(config-if)#ipv6 address 2001:1234::/64 eui-64
```

```
Guatemala(config)#interface fastEthernet 0/0
Guatemala(config-if)#ipv6 address autoconfig
```

Fuente: elaboración propia.

Se necesita habilitar la configuración automática de direcciones IPv6 en el *router* de Guatemala para garantizar que genere su propia dirección IPv6.

Figura 216. **Verificación de procesos IPv6**

```
Guatemala#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
```

```
miami#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
```

Fuente: elaboración propia.

Se puede utilizar *debug* ipv6 nd para observar todo el proceso.

Figura 217. **Anuncio de verificación IPv6**

```
miami#
ICMPv6-ND: Sending RA to FF02::1 on FastEthernet0/0
ICMPv6-ND: MTU = 1500
ICMPv6-ND: prefix = 2001:1234::/64 onlink autoconfig
```

Fuente: elaboración propia.

Aquí se puede verificar el *router* Miami enviando el anuncio del *router* con el prefijo.

Figura 218. **Autoconfiguración IPv6**

```
Guatemala#
ICMPv6-ND: Received RA from FE80::CE0A:18FF:FE0E:0 on FastEthernet0/0
ICMPv6-ND: Autoconfiguring 2001:1234::CE09:18FF:FE0E:0 on FastEthernet0/0
```

Fuente: elaboración propia.

Está es el *router* Guatemala que recibe los avisos del *router* y configura su propia dirección IPv6.

Figura 219. **Resultado de la autoconfiguración IPv6**

```
Guatemala#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::CE09:18FF:FE0E:0
    2001:1234::CE09:18FF:FE0E:0
```

Fuente: elaboración propia.

En la configuración se puede apreciar una nueva dirección IPv6 en el router Guatemala.

Figura 220. **Comando Show ipv6 en la verificación de direcciones**

```
Guatemala#show ipv6 routers
Router FE80::CE0A:18FF:FE0E:0 on FastEthernet0/0, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  HomeAgentFlag=0, Preference=Medium
  Reachable time 0 msec, Retransmit time 0 msec
  Prefix 2001:1234::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
```

Fuente: elaboración propia.

También se puede utilizar el comando show ipv6 *routers* para verificar los anuncios del *router* en caché. Está es un buen ejemplo en el que verá la dirección de enlace local del *router* Miami en lugar de la dirección global unicast.

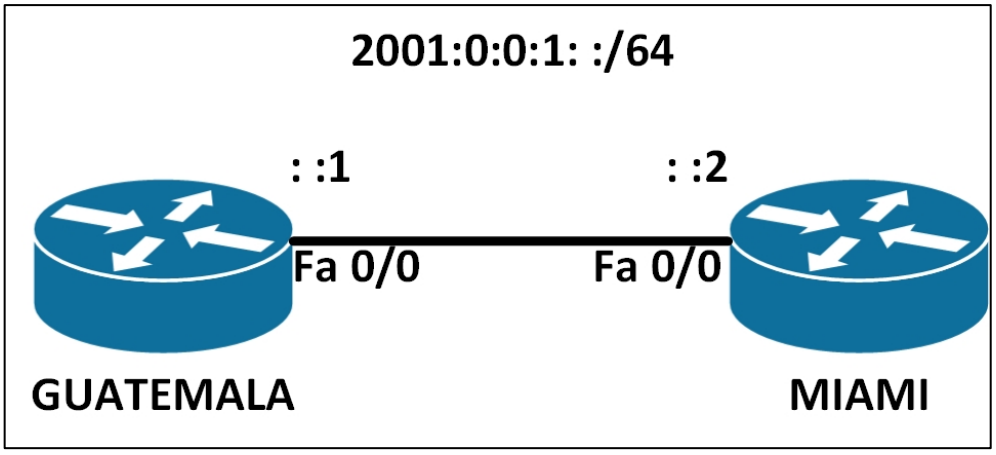
DHCP también está disponible para IPv6 y se llama DHCPv6. La gran diferencia entre DHCP para IPv4 y DHCPv6 es que ya no usamos el tráfico de

difusión. Cuando un dispositivo IPv6 está buscando un servidor DHCPv6, enviará paquetes de multidifusión a FF02 :: 1: 2. Los *routers* reenviarán estos paquetes a servidores DHCP.

Al igual que en IPv4, necesitamos asignar la información de la capa 2 a la capa 3. para IPv4 se tiene ARP, pero como no podemos transmitir más con IPv6, necesitamos algo más. No usamos ARP para IPv6, pero nuestro NDP (protocolo de descubrimiento de vecino) del que acaba de ser testigo también ha reemplazado al ARP.

Cuando un dispositivo IPv6 desea enviar un paquete IPv6 a un *host* en la misma LAN, comenzará revisando su base de datos de vecinos. En esta base de datos encontrará la lista con todos los vecinos y sus direcciones MAC. Si su dispositivo IPv6 no reconoce la dirección MAC de un vecino, usará NDP para descubrir automáticamente la dirección MAC. NDP es similar a ARP pero funciona ligeramente diferente.

Figura 221. **Verificación NDP**



Fuente: elaboración propia.

Figura 222. **Proceso EUI-64**

```
Guatemala(config)#interface fastEthernet 0/0
Guatemala(config-if)#ipv6 address 2001:0:0:1::1/64
```

```
miami(config)#interface fastEthernet 0/0
miami(config-if)#ipv6 address 2001:0:0:1::2/64
```

Fuente: elaboración propia.

Se utilizará el *router* Guatemala y Miami nuevamente para mostrar el proceso. Se configurarán manualmente sin EUI-64.

Figura 223. **Habilitación del comando neighbor Discovery debugging**

```
Guatemala#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
```

Fuente: elaboración propia, empleando Visio 2013.

Se habilitará el comando *neighbor discovery debugging* para observar qué sucederá.

Figura 224. **Comando Ping de verificación**

```
Guatemala#ping 2001:0:0:1::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0:0:1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/12 ms
```

Fuente: elaboración propia.

Se enviará un *ping* para despertar el descubrimiento vecino.

Figura 225. **Mensajes NS y NA**

```
Guatemala#
ICMPv6-ND: Sending NS for 2001:0:0:1::2 on FastEthernet0/0
ICMPv6-ND: Received NA for 2001:0:0:1::2 on FastEthernet0/0 from
2001:0:0:1::2
ICMPv6-ND: INCMP -> REACH: 2001:0:0:1::2
Guatemala#
ICMPv6-ND: Received NS for 2001:0:0:1::1 on FastEthernet0/0 from
FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: DELETE -> INCMP: FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: INCMP -> STALE: FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: Sending NA for 2001:0:0:1::1 on FastEthernet0/0
ICMPv6-ND: STALE -> DELAY: FE80::CE0A:18FF:FE0E:0
Guatemala#
ICMPv6-ND: DELAY -> PROBE: FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: Sending NS for FE80::CE0A:18FF:FE0E:0 on FastEthernet0/0
ICMPv6-ND: Received NA for FE80::CE0A:18FF:FE0E:0 on FastEthernet0/0 from
FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: PROBE -> REACH: FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: Received NS for FE80::CE09:18FF:FE0E:0 on FastEthernet0/0 from
FE80::CE0A:18FF:FE0E:0
ICMPv6-ND: Sending NA for FE80::CE09:18FF:FE0E:0 on FastEthernet0/0
```

Fuente: elaboración propia.

En la depuración, puede ver los mensajes NS (*Neighbor Solicitation*) y NA (*Neighbor Advertisement*) entrantes y salientes.

Figura 226. **Comando show neighbors**

```
Futura#show ipv6 neighbors
IPv6  Address                 Age Link-layer Addr State
Interface
2001:0:0:1::2                 7 cc0a.180e.0000     STALE Fa0/0
FE80::CE0A:18FF:FE0E:0        7 cc0a.180e.0000     STALE Fa0/0
```

Fuente: elaboración propia.

Se puede utilizar el comando show ipv6 neighbors para verificar la asignación entre la dirección MAC y las direcciones IPv6.

En IPv4 se pueden utilizar ARP gratuitos, pero en IPv6 tiene algo más. Cuando una interfaz IPv6 utiliza un mecanismo llamado DAD (*Duplicate Address Detection*). Para asegurarse de que la dirección IPv6 no se encuentre en uso, el dispositivo IPv6 enviará una solicitud de vecino a la dirección de multidifusión del nodo solicitado, pero en función de su propia dirección IPv6. Si responde cualquier otro dispositivo, se sabe que se encuentra otra dirección IPv6 duplicada.

3.7.1. Multicast

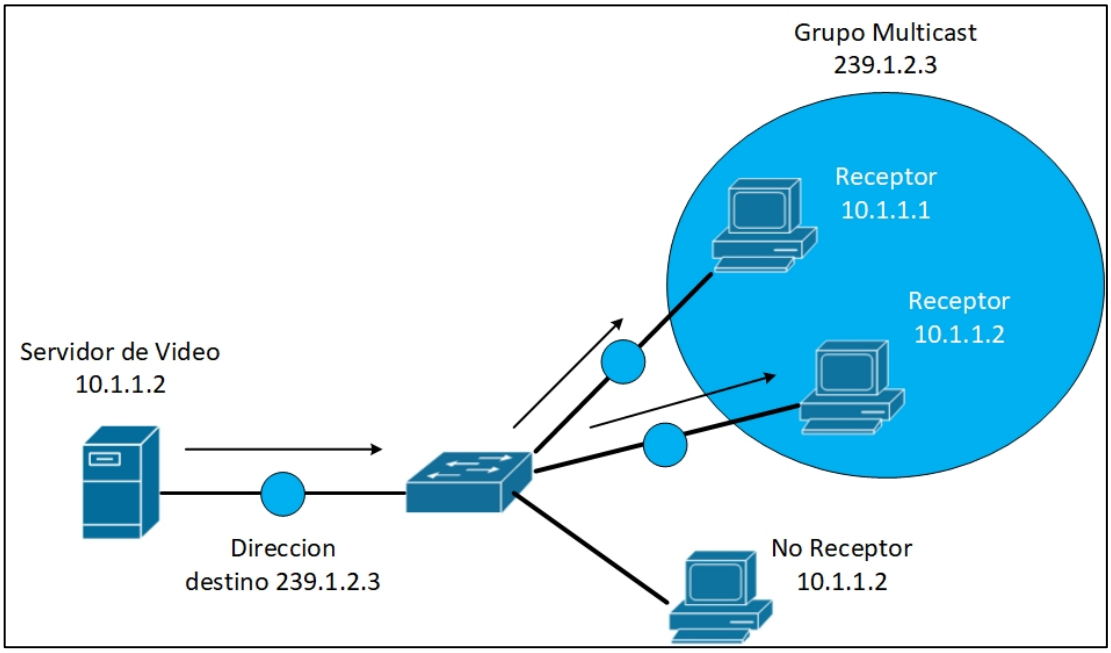
La tecnología de multicast proporciona un mecanismo eficiente para que un solo *host* envíe tráfico a destinos múltiples, pero específicos. Por ejemplo, se tiene una red con 100 usuarios. Veinte de esos usuarios desean recibir una transmisión de video desde un servidor de video. Con una solución de unidifusión, el servidor de video debería enviar 20 transmisiones individuales, una transmisión por cada destinatario. Tal solución podría consumir una cantidad significativa de ancho de banda de red y poner una pesada carga de procesador en el servidor de video.

Con una solución de transmisión, el servidor de video solo tendría que enviar la transmisión de video una vez; sin embargo, todos los dispositivos que no desean recibirla. A pesar de esos dispositivos no desean recibirla. A pesar de que son dispositivos no desean recibir la transmisión de video, aún tiene que detener lo que están haciendo y tomarse un tiempo para revisar cada uno de estos paquetes no deseados.

Como se muestra en la figura, el *multicast* ofrece un compromiso, permitiendo que el servidor de video envíe la transmisión de video solo una vez

y solo envíe la transmisión de video a los dispositivos en la red que desean recibir la transmisión.

Figura 227. **Representación de una topología multicast**



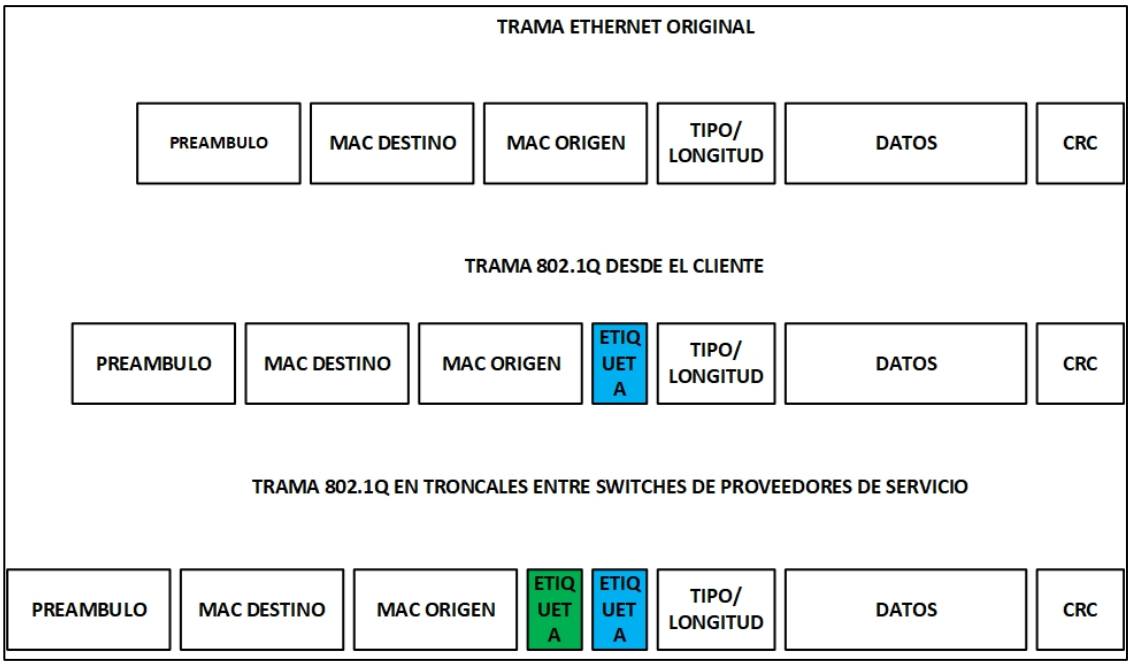
Fuente: elaboración propia, empleando Visio 2013.

Lo que hace esto posible en redes IPv4 es el uso de una dirección Clase D. Una dirección de Clase D, como 239.1.2.3, representa la dirección de un grupo de multicast. El servidor de video podría, en éste ejemplo, enviar una sola copia de cada paquete de flujo de vídeo destinado a 239.1.2.3. Los dispositivos que desean recibir la transmisión de video pueden unirse al grupo de multicast. En función de la solicitud del dispositivo, los switches y los *routers* de la topología pueden determinar de forma dinámica a partir de que puertos debe reenviarse la transmisión de video.

3.7.2. Tunelización Q-IN-Q

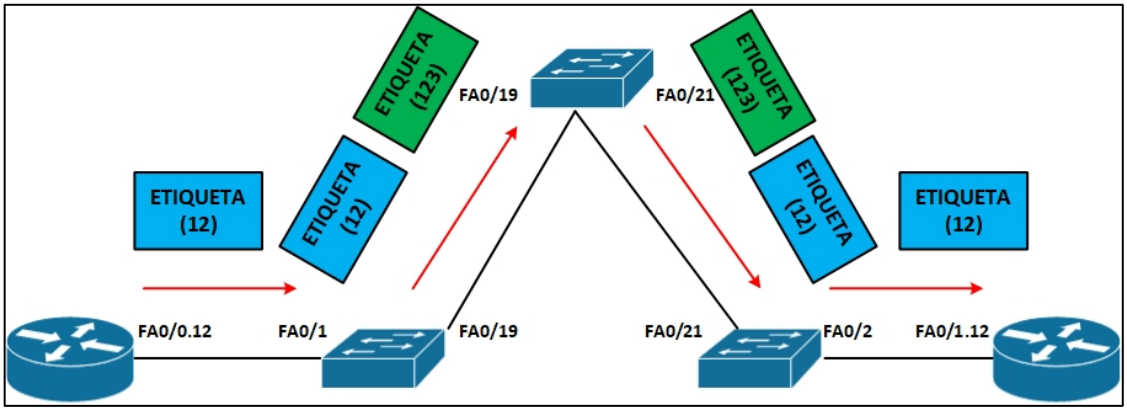
La tunelización Q-in-Q y la traducción de vlan permiten a los proveedores de servicios crear una conexión Ethernet de capa 2 entre dos sitios de clientes. Los proveedores pueden segregar el tráfico de VLAN se diferentes clientes en un enlace (por ejemplo, si los clientes usan ID de VLAN superpuestas) o agrupar diferentes VLAN de clientes en una sola VLAN de servicio. Los centros de datos pueden usar la tunelización Q-in-Q y la traducción de la VLAN para aislar el tráfico de clientes dentro de un solo sitio o para permitir el flujo de tráfico de clientes entre centros en la nube en diferentes ubicaciones geográficas.

Figura 228. **Representación de las distintas tramas en Q-in-Q**



Fuente: elaboración propia, empleando Visio 2013.

Figura 229. **Representación del envío de etiquetas Q-in-Q**



Fuente: elaboración propia, empleando Visio 2013.

Al utilizar la tunelización Q-in-Q, los proveedores pueden segregar o agrupar el tráfico del cliente en menos VLAN o diferentes VLAN agregando otra capa de etiquetas 802.1Q. La tunelización Q-in-Q es útil cuando los clientes tienen identificadores de VLAN supuestas, ya que las etiquetas VLAN 802.1Q (dot1Q) del cliente están precedidas por la etiqueta VLAN de servicio (S-VLAN).

En la tunelización Q-in-Q, cuando un paquete viaja desde una VLAN de un cliente (C-VLAN) a una VLAN de un proveedor de servicios, se agrega una etiqueta 802.1Q que especifica el paquete del cliente. Está etiqueta adicional se usa para segregar el tráfico hacia las VLAN del servicio (S-VLAN) definidas por el proveedor del servicio. La etiqueta 802.Q del cliente original del paquete permanece y se transmite de forma transparente, pasando a través de la red del proveedor del servicio. A medida que el paquete abandona la S-VLAN en la dirección descendente, se elimina la etiqueta adicional 802.1Q. Nota importante, todas las VLAN en una implementación pueden ser VLAN de servicio. Es decir, si la cantidad total de VLAN soportadas es 4090, todas pueden ser VLAN de servicio.

Cuando la tunelización Q-in-Q está habilitada en los *switches ethernet*, se supone que las interfaces troncales son parte de la red del proveedor de servicios y se supone que las interfaces de acceso están orientadas al cliente. Una interfaz de acceso puede recibir marcos etiquetados y no etiquetados en este caso.

Una interfaz puede ser miembro de múltiples S-VLAN. Puede asignar una C-VLAN a una S-VLAN (1: 1) o múltiples C-VLAN a una S-VLAN (N: 1). Los paquetes tienen una etiqueta doble para una capa adicional de segregación o agrupación de C-VLAN. Las etiquetas C-VLAN y S-VLAN son únicas; para que pueda tener una C-VLAN 101 y una S-VLAN 101, por ejemplo. Puede limitar el conjunto de etiquetas de clientes aceptadas a un rango de etiquetas o valores discretos. Los valores de clase de servicio (CoS) de C-VLAN no se modifican en la dirección descendente. Puede, opcionalmente, copiar la prioridad de ingreso y las configuraciones de CoS en la S-VLAN. En los *switches* que no son ELS, puede usar VLAN privadas para aislar a los usuarios a fin de evitar el reenvío del tráfico entre las interfaces de usuario, incluso si las interfaces están en la misma VLAN.

Cuando el túnel Q-in-Q está habilitado, se supone que las interfaces troncales son parte del proveedor del servicio o de la red del centro de datos. Se supone que las interfaces de acceso están orientadas al cliente y aceptan marcos etiquetados y no etiquetados. Cuando se utiliza la agrupación de muchos a uno o el mapeo de una interfaz específica, debe usar la opción nativa para especificar una S-VLAN para los paquetes sin etiquetar y etiquetados con prioridad si desea aceptar estos paquetes. (Los paquetes etiquetados con prioridad tienen su ID de VLAN configurada en 0, y sus bits de punto de código de prioridad pueden configurarse con un valor CoS).

Si no especifica una S-VLAN para ellos, los paquetes no etiquetados se descartan. La opción nativa no está disponible para la agrupación todo en uno porque no es necesario especificar paquetes sin etiquetar y etiquetados con prioridad cuando todos los paquetes se asignan a una S-VLAN.

Puede usar la opción nativa para especificar una S-VLAN para paquetes sin etiquetar y etiquetados con prioridad cuando se usa la agrupación de muchos a uno y la asignación de enfoques de una interfaz específica para asignar C-VLAN a S-VLAN. De lo contrario, los paquetes se descartan. La opción nativa no está disponible para la agrupación todo en uno porque no es necesario especificar paquetes sin etiquetar con prioridad etiquetada cuando todos los paquetes se asignan a la S-VLAN.

La tunelización Q-in-Q no afecta ningún valor de clase de servicio (CoS) que está configurado en una C-VLAN. Estas configuraciones se conservan en la etiqueta C-VLAN y se pueden usar después de que un paquete abandona una S-VLAN. Los valores CoS no se copian de las etiquetas C-VLAN a las etiquetas S-VLAN.

Dependiendo de la configuración de su interfaz, es posible que deba ajustar el valor MTU en su troncal o puertos de acceso para acomodar los 4 bytes utilizados para la etiqueta agregada por el túnel Q-in-Q. Por ejemplo, si usa el valor MTU predeterminado de 1514 bytes en su acceso y puertos troncales, necesita hacer uno de los siguientes ajustes:

- Reduzca la MTU en los enlaces de acceso en al menos 4 *bytes* para que los marcos no excedan la MTU del enlace troncal cuando se agregan las etiquetas S-VLAN.

- Aumente la MTU en el enlace troncal para que el enlace pueda manejar el tamaño de datagrama más grande.

La traducción VLAN reemplaza una etiqueta entrante C-VLAN con una etiqueta S-VLAN en lugar de agregar una etiqueta adicional. Por lo tanto, se pierde la etiqueta C-VLAN, por lo que un paquete con una solo etiqueta normalmente no tiene etiqueta cuando sale la S-VLAN (en el otro extremo del enlace). Si un paquete entrante ha tenido túneles Q-in-Q aplicados por adelantado, la traducción de VLAN reemplaza a la etiqueta externa y la etiqueta interna se conserva cuando el paquete abandona la S-VLAN en el otro extremo del enlace. Los paquetes entrantes cuyas etiquetas no coinciden con la etiqueta C-VLAN se descartan, a menos que exista una configuración de traducción de VLAN adicional para esas etiquetas.

Para configurar la traducción de VLAN, use la declaración de intercambio de mapas en el nivel de jerarquía [editar interfaz vlans]. Siempre que las etiquetas C-VLAN y S-VLAN sean únicas, puede configurar más que una traducción C-VLAN a VLAN en un puerto de acceso. Si está traduciendo solo una VLAN en una interfaz, no necesita incluir la introducción do1q en la configuración de S-VLAN. Si está traduciendo más de una VLAN, debe usar la instrucción dot1q.

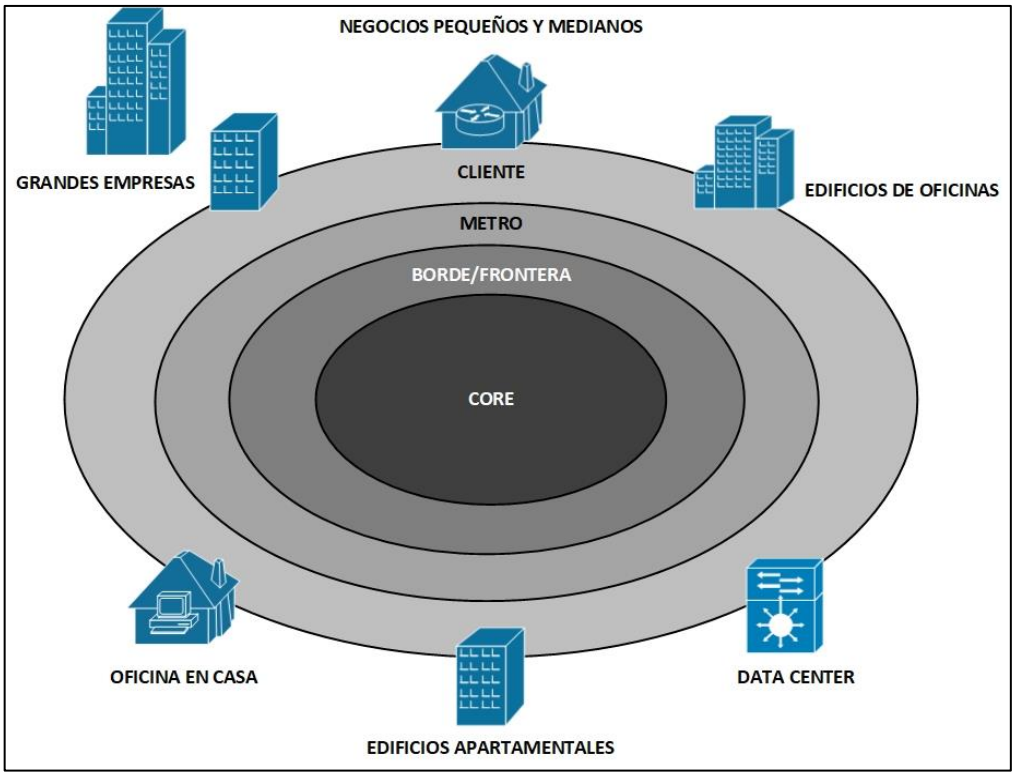
3.7.3. Red metro

El metro es simplemente el primer tramo de la red que conecta a los suscriptores y las empresas con la WAN. Las diferentes entidades atendidas por el metro incluyen clientes residenciales y comerciales, ejemplos de los cuales son grandes empresas (LE, Large Enterprises), oficina pequeña, oficina hogar (SOHO, *Small Office/home office*), pequeñas y medianas empresas

(SMB, *Small and medium-sized businesses*), unidades multiusuario (MTU, *multitenant units*). Vea la figura.

La parte del metro que toca al cliente se llama la última milla para indicar el último tramo de red del operador. En un mundo donde el cliente que paga se encuentra en el centro del universo, la industria también llama a está tramo la primera milla para reconocer que el cliente es el primero. Un término adecuado sería probablemente “la frontera final” porque el último tramo de la red suele ser el más desafiante y el más costoso de construir y ese la barrera final para acelerar la transformación del metro en una red de datos de alta velocidad.

Figura 230. **Representación de una topología de red metro**



Fuente: elaboración propia, empleando Visio 2013.

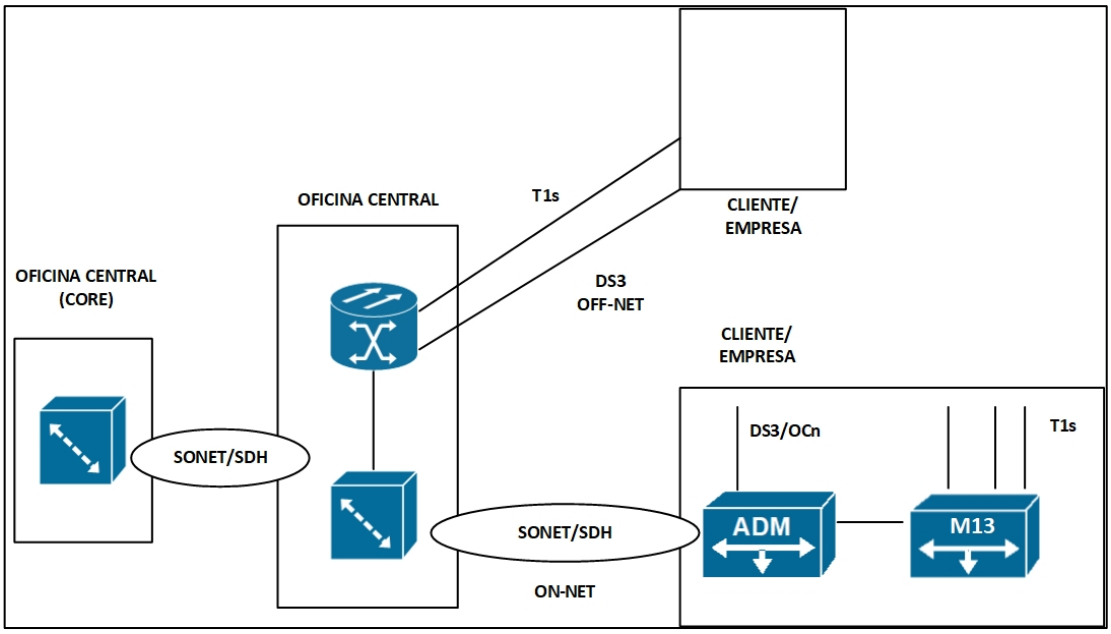
El metro heredado consiste principalmente en la tecnología de multiplexación por división de tiempo (TDM, *time division multiplexing*), que está muy optimizada para la entrega de servicios de voz, una red de metro típica consiste en equipos TDM colocados en el sótano de los edificios de los clientes y las oficinas centrales del operador de intercambio local (ILEC, *icumbent local exchange carrier*). El equipo TDM consta de multiplexores digitales, conexiones cruzadas de acceso digital (DAC, *digital access cross-connects*, a menudo denominadas conexiones cruzadas digitales), multiplexores de adición extracción SONET / SDH (ADM), conexiones cruzadas SONET / SDH, y más.

En la figura se muestra una vista TDM de una implementación de red metro heredada. Está escenario muestra la conectividad con clientes empresariales para redes dentro y fuera de la red. Una red *on-net* es una red en la que la fibra llega al edificio y el transportista instala un ADM en el sótano del edificio y ofrece circuitos T1 o DS3 / OCn a diferentes clientes en el edificio. En este caso, los multiplexores digitales como M13S multiplexan múltiples T1s por un DS3 o múltiples DS3 por un circuito OCn que se transporta a través del anillo de fibra SONET/SDH a la oficina central (CO, *Central Office*). En una red fuera de la red, en la que la fibra no llega al edificio, la conectividad se realiza a través de circuitos T1 o DS3 de cobre que se agregan en el CO mediante DACS. Los circuitos agregados están conectados de forma cruzada en el CO a otros CO centrales, donde los circuitos se terminan o se transportan a través de la WAN según el servicio que ofrece.

La operación e instalación de una red TDM pura es muy tediosa y extremadamente costosa de implementar, porque TDM en si misma es una tecnología muy rígida y no tiene la flexibilidad o la economía para adaptarse a las necesidades del cliente. El costo de desplegar redes de metro es la suma del gasto de capital en equipo y el gasto operacional. El gasto operacional

incluye el costo de la planificación, instalación, operación y administración de la red, mantenimiento y resolución de problemas, entre otros. Lo importante es darse cuenta de que estos gastos operativos podrían alcanzar alrededor del 70 por ciento del gasto total del operador, lo que podría tener un gran peso en la decisión del operador respecto a que productos del operador y tecnologías instalar en la red.

Figura 231. **Representación de una red TDM**



Fuente: elaboración propia, empleando Visio 2013.

El costo de llevar el servicio a un cliente tiene un gran efecto en el éxito de la entrega de ese servicio. Cuanto menos el transportista tenga que tocar las instalaciones del cliente y el equipo de CO para ofrecer un servicio inicial e incremental, mayor será el retorno de la inversión del transportista para ese cliente. El término rollos de camiones se refiere a los camiones que se envían a las instalaciones del cliente para activar o modificar un servicio en particular.

Cuántas más vueltas de camión requiera un cliente, más dinero gastará el operador en ese cliente.

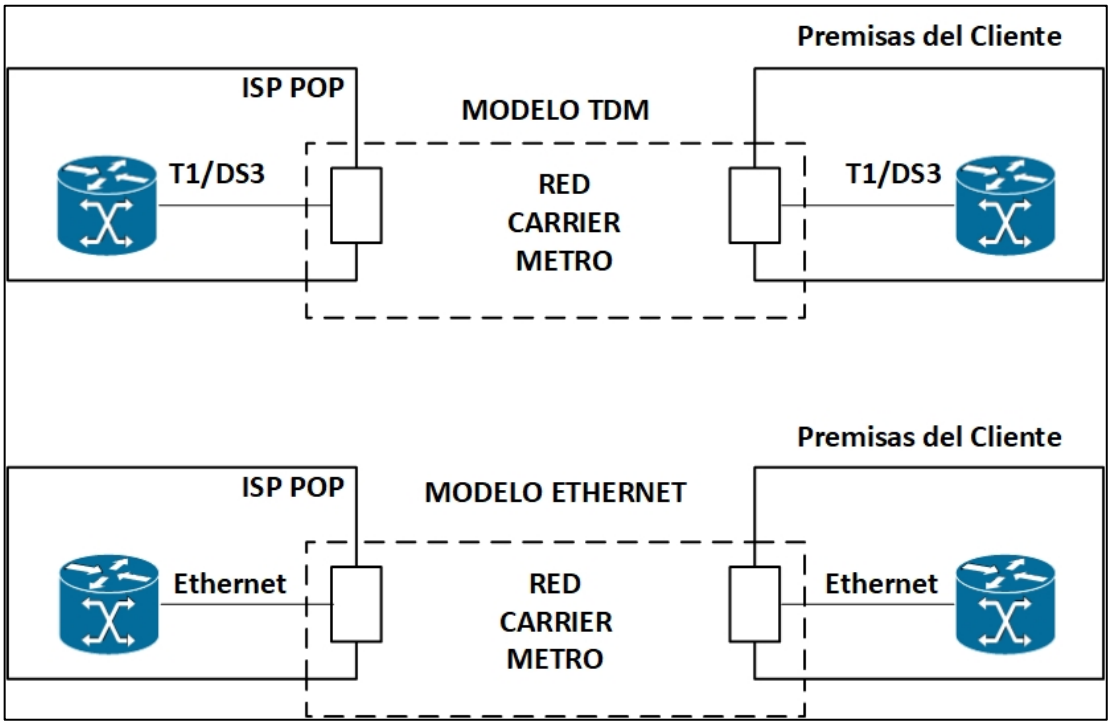
El desafío que tienen las interfaces TDM es que el ancho de banda que ofrecen no crece linealmente con las demandas de los clientes, sino que crece en las funciones paso a paso. Una interfaz DS3 a 45 *Mbps*; la siguiente función de paso es una interfaz OC3 a 155 Mbps; y así. Entonces, cuando las necesidades de ancho de banda de un cliente exceden la tasa de 1.5 *Mbps*, el operador se ve obligado a ofrecer al cliente múltiples circuitos T1 (nXT1) o moverse a un circuito DS3 y darle al cliente una porción del DS3. El efecto final es que la interfaz física vendida al cliente ha cambiado, y el costo del cambio tiene un gran impacto tanto en el operador como en el cliente.

Pasar de una interfaz T1 a un nXT1 o DS3/OCn requiere cambios en el equipo de las instalaciones del cliente (CPE, *customer premises equipment*) para admitir una interfaz y también requiere cambios en el equipo de CO para acomodar los nuevos circuitos desplegados. Esto ocurrirá cada vez que un cliente solicite un cambio de ancho de banda durante la vida de la conexión del cliente. Los servicios como *Channelized* DS1, *Channelized* DS3 y *Channelized* OCN pueden ofrecer más flexibilidad en la implementación de incrementos de ancho de banda. Sin embargo, estos servicios tienen un costo mucho mayor para la interfaz física y los routers y tienen una granularidad limitada. Está es uno de los principales impulsores de la proliferación de *Ethernet* en el metro como interfaz de acceso. Una interfaz *Ethernet* 10/100/1000 escala mucho mejor desde velocidades de *submegabit* hasta *megabit*, a una fracción del costo de una interfaz TDM.

En la figura se muestra la diferencia entre el modelo TDM y el modelo *Ethernet* para brindar conectividad a internet. En el modelo TDM, el operador de

metro, como ICEL o RBOC, ofrece el circuito T1 punto a punto, mientras que el ISP administra la entrega de servicios de internet, que incluye la administración de las direcciones IP del cliente y la conectividad del *router* en el punto de presencia (POP, *Point of Presence*). Está normalmente ha sido el modelo preferido para los ILEC que no desean involucrarse en el direccionamiento IP y en el enrutamiento del tráfico de IP. En algunos casos, los ILEC pueden subcontratar el servicio o administrar toda la conexión IP si lo desean. Sin embargo, esta modelo mantiene una línea de demarcación entre la entrega de servicios IP y la entrega de servicios de conectividad.

Figura 232. **Comparación de red ethernet y TDM**



Fuente: elaboración propia, empleando Visio 2013.

En el modelo Ethernet, ambas interfaces de red en el lado del cliente y el lado ISP son interfaces *Ethernet*. El ILEC gestiona la conexión (L2), mientras que el ISP gestiona los servicios de IP, desde una perspectiva operacional, esta disposición mantiene al ILEC en un modelo similar al servicio de línea privada T1; sin embargo, abre la oportunidad para ILEC de vender un servicio adicional sobre la misma conexión *Ethernet* sin ningún cambio en el CPE y la red.

3.7.4. Ethernet en metro

La tecnología *Ethernet* hasta ahora ha sido ampliamente aceptada en las implementaciones empresariales, y millones de puertos *Ethernet* ya se han implementado. La simplicidad de esta tecnología le permite escalar la interfaz de *Ethernet* a un gran ancho de banda sin dejar de ser rentable. El costo de una interfaz de 100 Mbps para *switches* LAN de grupo de trabajo empresarial L2 inferior a 400 quetzales en los próximos años.

Estos costos y métricas de rendimiento y la facilidad de uso de *Ethernet* motivan a las redes de operadores a utilizar *Ethernet* como una tecnología de acceso. En está nuevo modelo, al cliente se le otorga una interfaz *Ethernet* en un lugar de una interfaz TDM.

3.7.5. Una vista de datos de la red metro

Una vista de datos del metro pone en perspectiva los diferentes servicios del metro y como los ofrecen los diferentes proveedores.

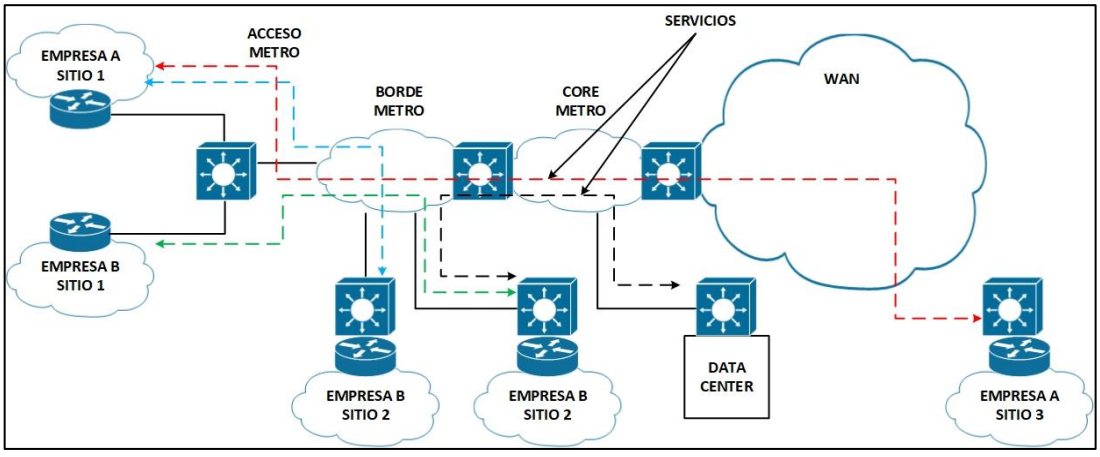
En la figura muestra una vista de la red metro con énfasis en el acceso de datos, la agregación de datos y la entrega de servicios. Como se puede observar, la red metro está dividida en tres segmentos:

- Acceso a metro: este segmento constituye el primer nivel de agregación metropolitana. Las conexiones que salen de los edificios se agregan en esta ubicación de CO en tuberías más grandes que a su vez se transportan dentro del metro a través de la WAN.

- Metro: este segmento constituye el primer nivel de agregación metropolitana. Las conexiones que salen de los edificios se agregan en esta ubicación de CO en tuberías más grandes que a su vez se transportan dentro del metro o a través de la WAN.

- Núcleo metro: este segmento constituye un segundo nivel de agregación en el que se agregan OC fronterizos en un CO central. A su vez, los CO centrales están conectados entre sí para formar un núcleo metropolitano desde el cual se revisa el tráfico en la WAN.

Figura 233. **Representación de una red metro operativa**



Fuente: elaboración propia, empleando Visio 2013.

La terminología y mucha variación del metro puede ser confusas. En algunos casos, solo hay un nivel de agregación; por lo tanto, las conexiones de construcción se agregan en un lugar y luego se conectan directamente en un router central. En otros escenarios, el centro de operaciones metropolitanas CO, a veces llamado centro metropolitano CO, a veces llamado centro metropolitano, se ubica junto con el área amplia POP.

- Servicios metro

Los servicios de metro varían según el mercado objetivo, comercial o residencial, y si se trata de un servicio minorista o un servicio mayorista. La siguiente lista ofrece un resumen de algunos de los servicios metro que se promocionan:

  - Conectividad a internet
  - Servicio de LAN transparente (LAN de punto a punto)
  - L2VPN (LAN punto a punto o multipunto a multipunto a LAN)
  - LAN a recursos de red (centro de datos remoto)
  - Extranet
  - LAN a *frame relay*/ATM VPN
  - Redes de área de almacenamiento (SAN)
  - Transporte de metro (*backhaul*)
  - VoIP

Algunos de estos servicios, como la conectividad a internet y TLS, se ofrecen desde hace muchos años. La diferencia ahora es que estos servicios cuentan con conectividad *Ethernet*, y los operadores se están moviendo hacia un modelo en el que todos los servicios se pueden ofrecer en la misma infraestructura y se pueden vender al mismo cliente sin mayores gastos

operacionales. Esto introduce una excelente propuesta de valor tanto para el cliente como el operador. Los servicios se provisionan mediante el transporte de la aplicación a través de conexiones L2 punto a punto o multipunto. Las siguientes secciones discuten algunos de estos servicios en mayor detalle.

Un ejemplo de dicho servicio es aquel que permite a una empresa hacer una copia de seguridad de sus datos en una ubicación remota y segura para la recuperación de desastres. En la figura se muestra que, además del servicio de internet, el cliente puede tener un servicio de copia de seguridad de datos y recuperación de desastres que respalda constantemente los datos en la red metro.

Para las nuevas redes de datos en las que la conectividad se realiza a través de *gigabit* y tuberías de 10 *gigabits*, la red metro puede transformarse en una LAN de alta velocidad que ofrece aplicaciones de ancho de banda que normalmente no serían factibles de desplegar sobre la infraestructura TDM heredada.

Como se mencionó anteriormente, el servicio en la red metro tomará muchas formas dependiendo del cliente objetivo. El mismo modelo de recursos LAN a la red podría aplicarse a aplicaciones residenciales, lo que permite a los ILEC comenzar a competir con las compañías de cable en la distribución de servicios multimedia. En una aplicación residencial, los servidores de video se ubicarían en una red metro POP y los clientes residenciales de MDU podrían acceder a video digital de alta velocidad a pedido a través de una conexión *Ethernet*. Si bien estos servicios parecen futuristas en otros países, el panorama internacional pronto podría ser diferente al resto del mundo, donde el rápido despliegue de redes *Ethernet* ya está haciendo que estas aplicaciones sean una realidad.

CONCLUSIONES

1. Se ha presentado una propuesta para el laboratorio de telecomunicaciones y redes locales II, un contenido el cual el estudiante puede adquirir y para complementar el conocimiento acerca del funcionamiento de protocolos que se utilizan en la actualidad.

2. El auxiliar a cargo podrá seguir este manual para la realización de prácticas así como contribuir con su metodología a impartir el laboratorio de telecomunicaciones y redes locales II.

3. Se presenta un contenido de fundamentos teóricos para comprender el funcionamiento de un proveedor de servicios de internet, el cual se complementa con ejemplos y material complementario.

4. Este material posee contenido totalmente en español para facilitar su análisis y comprensión, hacia un público de habla hispana.

RECOMENDACIONES

1. Explicar el funcionamiento de los protocolos y tecnicas distintas que existen para el fortalecimiento del laboratorio de electronica con conceptos a nivel de proveedores de servicios.

2. La constante actualizacion de los conceptos impartidos en esta tesis para que el estudiante pueda obtener un punto de partida a realizar una investigacion mas profunda asi mismo con los equipos que se puedan configurar.

3. La capacitacion de la configuracion de los equipos de comunicacion en todos los niveles que existen, el cual es desde nucleo, distribucion y acceso en niveles de proveedores de servicio.

4. Ejemplos en los cuales el alumno pueda ejecutar los ejemplos para realizar las pruebas y comprobar la teoria con la practica, tomando en cuenta que puede realizar distintas configuraciones de los ejemplos dados en este trabajo de graduación.

BIBLIOGRAFÍA

1. GHEIN, Luc. *MPLS Fundamentals*. Estados Unidos: Cisco Press, 2007. 172 p.

2. MOLENAAR, R. *How to Master CCNA*. [en línea]. <http://gns3vault.com/product/how-to-master-ccna-rs/>. [Consulta: 9 de agosto de 2015].

3. PAQUET, Catherine. *Building scalable*. Estados Unidos: Academia Cisco, 2010. 291 p.

4. _____. *Implementing Secure converged wide area networks* Estados Unidos: Cisco Press, 2007. 223 p.

5. WALLACE, Kevin. *CCNP Routing and Switching Route 300-101*. Estados Unidos: Cisco Press, 2015. 182 p.