_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE UPV INGENIEROS DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISEÑO DE REDES CON BGP

Víctor Sánchez García

Tutor: Jose Oscar Romero Martínez

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2016-17

Valencia, 4 de julio de 2017

Resumen

El objetivo de este trabajo será el de realizar un profundo análisis del protocolo de enrutamiento BGP, no solo a nivel teórico en lo que respecta a los fundamentos y algoritmos utilizados, sino que para poder adquirir un alto grado de conocimiento acerca del mismo es necesario realizar diferentes simulaciones en un entorno de pruebas, para el cual utilizaremos equipos Cisco Systems.

De esta forma, se pretende presentar una visión clara y concisa de los entornos de red que utilizan los ISPs, así como de los métodos que emplean para el intercambio de información de enrutamiento entre diferentes AS. Mediante la realización del proyecto se incidirá sobretodo en esta última parte, ya que constituye la base sobre la que se rige la comunicación entre los grandes nodos de Internet. Este protocolo es el encargado de encontrar el camino más eficiente entre los mismos.

Resum

L'objectiu d'aquest treball será el de realizar un profund análisis del protocol d'enrutament BGP, no soles a nivell teóric pel que fa als fonaments i algoritmes utilitzats, sino que per a poder adquirir un alt grau de coneixement sobre el mateix, és necessari realizar diferents simulacions en un entorn de proves, per al qual utilitzarem equips Cisco Systems.

D'aquesta forma, es pretén presentar una visió clara y concisa dels entorns de xarxa que utilitzen els ISPs, aixi com dels mètodes que empren per a l'intercanvi d'informació d'enrutament entre diferents sistemes autónoms. Mitjançant la realització del projecte s'incidirà sobretot en aquesta última part, ja que constituïx la base sobre la qual es regix la comunicació entre els grans nodes d'Internet. Aquest protocol és el que s'encarrega de trobar el camí més eficient entre els mateixos.

Abstract

The aim of this work is to realize a deep analysis of the routing protocol BGP, not only to theoretical level regarding the foundations and used algorithms, but also to be able to acquire a high degree of knowledge of the same one it is necessary to realize different simulations in a test environment, for which we will use Cisco Systems equipment.

This research tries to present a clear and concise vision of the network environments which are used by the ISPs, as well as of the methods used for the exchange of information of routing between different autonomous systems. This project will focus on the latest part, since it constitutes the base in which the big nodes of Internet are regulated.

Índice

Capítulo 1. Introducción

En el presente capítulo introduciremos el proyecto de diseño de redes con el protocolo BGP. Para el desarrollo del mismo se realizará una integración teórica con los conceptos elementales para poder desarrollar satisfactoriamente el diseño de la red y su posterior puesta en servicio.

1.1 Objetivos

El objetivo de este trabajo será el de realizar un profundo análisis del protocolo de enrutamiento BGP, no solo a nivel teórico en lo que respecta a los fundamentos y algoritmos utilizados, sino que para poder adquirir un alto grado de conocimiento acerca del mismo será necesario realizar diferentes simulaciones en un entorno de pruebas, para el cual utilizaremos equipos Cisco Systems. Una vez tengamos claras las bases del protocolo, nos centraremos en realizar y verificar el diseño de una red y su posterior puesta en marcha mediante el protocolo BGP.

1.2 Metodología

La metodología empleada para el desarrollo del proyecto ha sido el estudio teórico de las bases del protocolo BGP, así como de algunas de las configuraciones permitidas por el mismo. Tras ello, se ha realizado un diseño de red a nivel esquemático e implementado en el programa *Cisco Packet Tracer* para explorar todos los aspectos posibles de configuración de la red y simular su puesta en marcha.

1.2.1 Gestión del proyecto

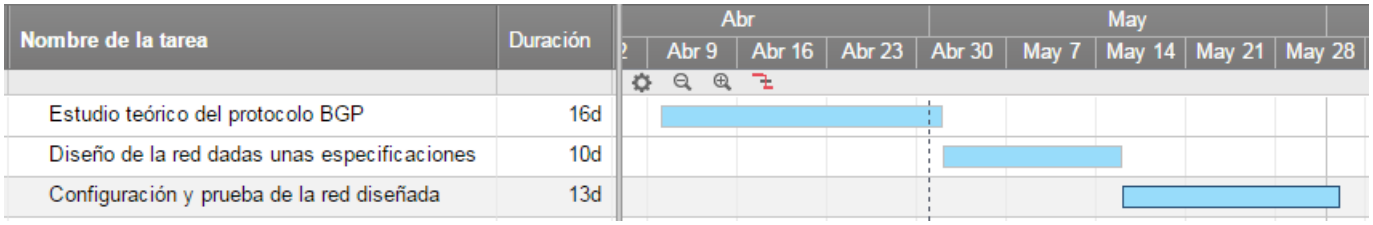
La gestión del proyecto se ha realizado planteando un marco de estudio con el que sería posible implementar el diseño, tras lo cual, asumiendo unas condiciones cerradas, realizar las configuraciones y probar que es completamente funcional.

1.2.1.1 Distribución en tareas

Las tareas elementales en las cuales se ha estructurado el proyecto son las siguientes:

- Estudio teórico del protocolo BGP
- Diseño de la red dadas unas especificaciones
- Configuración y prueba de la red diseñada

1.2.1.2 Diagrama Temporal



Capítulo 2. Introducción a BGP

El Border Gateway Protocol (BGP) es el sistema que utilizan los grandes nodos de Internet para comunicarse entre ellos y transferir una gran cantidad de información entre dos puntos de la red. Su misión es encontrar el camino más eficiente entre los nodos para propiciar una correcta circulación de la información en Internet. Facilita el intercambio de información sobre redes IP y la comunicación entre AS (AS). Por tanto, BGP es un protocolo interdominio (entre AS) e intradominio (dentro del mismo sistema autónomo).

El protocolo BGP se utiliza para intercambiar información. El intercambio de información en la red se realiza mediante el establecimiento de una sesión de comunicación entre los routers ABR (area border router) de los AS, estos routers son los que proporcionan la conexión entre diferentes AS. Para conseguir una entrega fiable de la información, se hace uso de una sesión de comunicación basada en TCP en el puerto número 179. Esta sesión debe mantenerse conectada debido a que ambos extremos de la comunicación periódicamente se intercambian y actualizan información. De modo que, al principio, cada router envía al vecino toda su información de encaminamiento y después únicamente se enviarán las nuevas rutas, las actualizaciones o la eliminación de rutas transmitidas con anterioridad. Además, periódicamente se envían mensajes para garantizar la conectividad. Cuando una conexión TCP se interrumpe por alguna razón, cada extremo de la comunicación está obligado a dejar de utilizar la información que ha aprendido por el otro lado. En otras palabras, la sesión TCP sirve como un enlace virtual entre dos AS vecinos, y la falta de medios de comunicación indica que el enlace virtual se ha caído. Cabe destacar que esa unión virtual tendrá más de un enlace físico que conecte a los dos routers frontera, pero si una conexión virtual se cae no indica necesariamente que la conexión física se haya caído.

Desde este punto de vista, la topología de Internet se puede considerar como un gráfico de conexión de AS conectados mediante enlaces virtuales. Para la puesta en funcionamiento de una red como la mencionada anteriormente, se debe proveer de un mecanismo de intercambio de rutas que permita comunicar correctamente los diferentes AS. El protocolo BGP utiliza el protocolo de vector distancia (path vector) para el intercambio de información de enrutamiento en la red. Se transmite una lista con identificación de los ASs por los que pasa el anuncio. De esa manera se conseguirá saber cómo llegar a cualquier dirección del prefijo de red propagado, del mismo modo, se dispondrá para cursar tráfico hacia cualquier dirección del prefijo de red correspondiente.

Capítulo 3. Características BGP

BGP, como se menciona en el apartado anterior, es categorizado como un protocolo de vector distancia avanzado. Se diferencia del resto de protocolos de vector distancia en varios aspectos claves, el más importante es que utiliza el protocolo TCP como protocolo de transporte, lo que proporciona un método de conexión fiable entre los diferentes hosts que forman la red BGP.

De esta forma, se asume pues que al existir una conexión fiable no es necesario implementar ningún mecanismo de reenvío o de recuperación de errores en la conexión. La información de los paquetes BGP se encapsula en paquetes TCP que usan el puerto 179, a su vez estos paquetes se encapsulan dentro de paquetes IP. Tal y como se puede ver en la siguiente imagen:

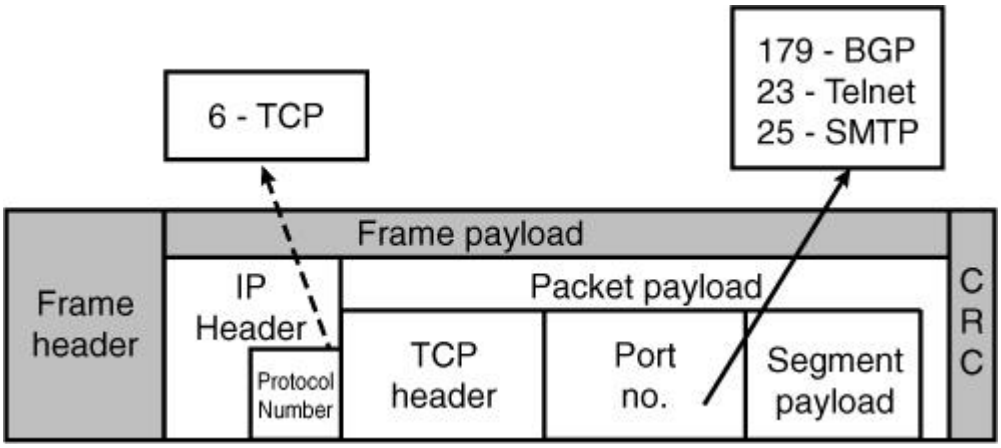


Figura 1. Estructura paquete BGP

Dos routers configurados con BGP establecen una conexión TCP el uno con el otro y se intercambian mensajes para iniciar dicha conexión y confirmar los parámetros de la misma. Estos dos routers se denominan "BGP peer routers" o "vecinos BGP". Una vez la conexión esté establecida, los routers intercambian al completo sus tablas BGP. Sin embargo, al ser una conexión fiable (ya que utilizamos el protocolo TCP), los routers BGP solo deberán intercambiar, a partir de ese momento, los cambios que se produzcan en la red. Los mensajes de actualización periódicos que se utilizan en otros protocolos aquí dejan de ser necesarios, lo que hace BGP es enviar mensajes "keepalive" cada 60 segundos, para indicar que la conexión sigue establecida.

Sin embargo, hay un aspecto importante que se debe tener en cuenta a la hora de utilizar TCP y es que, por propia definición del protocolo, utiliza mensajes ACK "acknowledgment". Estos mensajes son los que necesita el host que inicia la conexión para poder seguir con la misma, de esta forma pueden ocurrir problemas de latencia ya que de demorarse esos paquetes el resto de la conexión también tendría una demora importante. Además, hay que tener en cuenta que BGP transporta una gran cantidad de información en sus paquetes puesto que hay alrededor de unas 650000 redes (según el CIDR-Report) que se deben de publicar a lo largo de todo Internet.

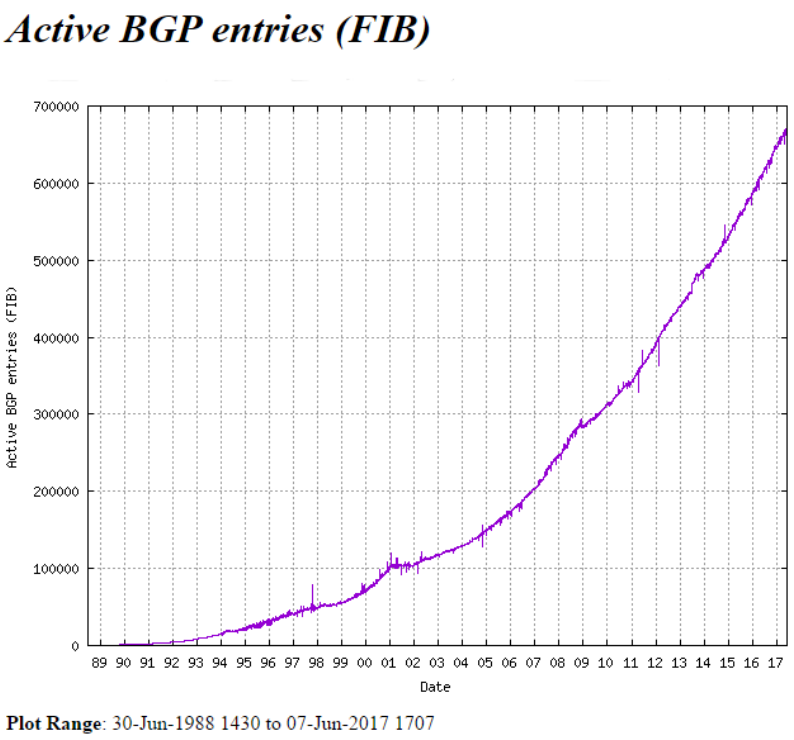


Figura 2. Evolución redes anunciadas por BGP

Para evitar que existan colapsos en la red, el propio TCP es un protocolo de ventana deslizante que permite al receptor enviar el ACK antes de haber recibido por completo los octetos especificados por la ventana. Este método permite que cualquier aplicación que utilice TCP, como en este caso BGP, seguir enviando paquetes sin necesidad de hacer un "stop and wait".

3.1 Vector Distancia BGP

Los protocolos de enrutamiento internos consisten en anunciar un listado de redes alcanzables con una determinada métrica para poder llegar a cada una de ellas. Sin embargo, como se menciona anteriormente, el BGP es un protocolo basado en vector distancia que a diferencia de los protocolos tradicionales intercambia información de alcanzabilidad de las diferentes redes basándose en "path attributes". Estos atributos son los utilizados por BGP para seleccionar la mejor ruta para alcanzar una red, aunque más adelante los analizaremos en profundidad, estos son algunos de ellos:

- ***AS-path***: listado completo del camino para alcanzar los diferentes AS BGP.
- ***next-hop***: incluye las direcciones IP necesarias para llegar a los AS.
- ***origin code***: explicación de cómo se introducen a BGP las redes que se encuentran al final del camino (path).

La información del camino entre AS se usa para crear una gráfica de la red BGP libre de bucles, ya que uno de los puntos fuertes en los que se basa BGP es en disponer de un esquema de red en la que los paquetes lleguen a su destino evitando cualquier problema que pudiese haber, tanto físico (bucles en la red) como físico (un enlace caído o un router averiado). Por esto mismo, un

router BGP no acepta actualizaciones de routing que incluyan el sistema autónomo en el que se encuentra, ya que eso significaría que esa actualización ha pasado por ese mismo AS, y aceptar ese "routing update" podría resultar en un bucle de enrutamiento.

BGP permite que se apliquen políticas de enrutamiento en algunos de los AS, de manera que el tráfico que fluye dentro del propio sistema autónomo puede estar condicionado a ciertas políticas que se consideren necesarias por parte del administrador de dicha red.

El protocolo BGP especifica que un router BGP puede publicar a los routers de sus AS vecinos solo aquellas rutas que usa. Esta regla refleja el "hop-by-hop paradigm" que es el usado en el internet actual, hay algunas reglas que no están soportadas por este paradigma de enrutamiento. Por ejemplo, BGP no permite que un sistema autónomo envié tráfico a otro sistema autónomo vecino y que pretenda que ese tráfico coja una ruta diferente de la que el sistema autónomo establece para todas las conexiones que pasen a través de él. Básicamente lo que dice es que no se puede influenciar en como un sistema autónomo va a enrutar tu tráfico, pero sí que puedes influir en cómo le llega ese tráfico desde tu sistema autónomo al suyo.

BGP es un protocolo que puede soportar cualquier política que rige el paradigma de conexión de Internet en la actualidad, por lo que es altamente aplicable como protocolo de routing interdominio para los diferentes AS de Internet.

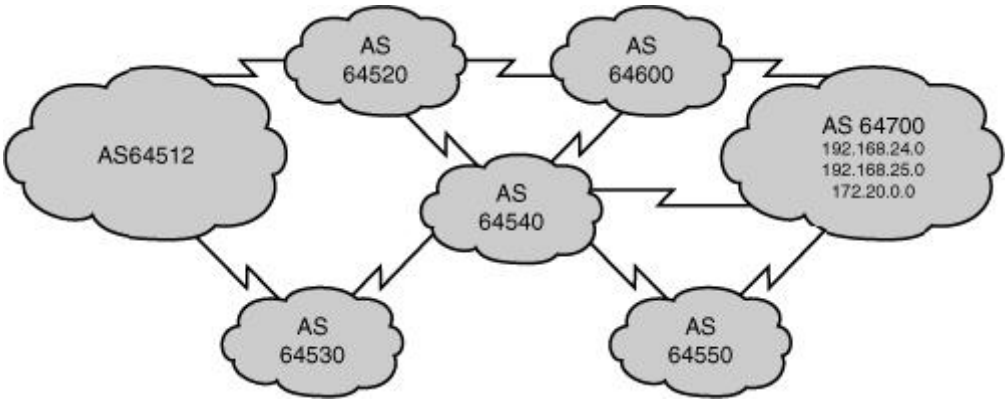


Figura 3. Ejemplo vector distancia BGP

Por ejemplo, en la siguiente figura, hay diferentes caminos posibles para que el sistema autónomo 64512 alcance las redes del sistema autónomo 64700 a través del sistema 64520:

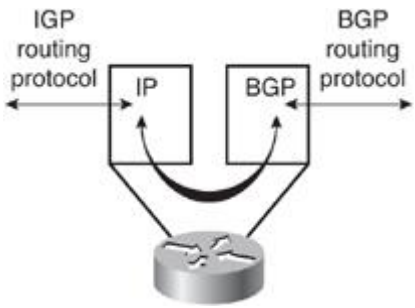
- 64520-64600-64700
- 64520-64600-64540-64550-64700
- 64520-64540-64600-64700
- 64250-64540-64550-64700

Pero realmente, el 64512 no ve todas esas posibilidades; ya que el 64520 solo publica al sistema autónomo 64512 la mejor opción (64520-64600-64700), al igual que ocurre con otros protocolos de enrutamiento interno (RIP, OSPF…). De esta forma, todos los paquetes que salgan del sistema autónomo 64512 con destino el 64700 cogerán esta ruta; esta elección de destino es la que mencionamos anteriormente como "hop-by-hop paradigm" que es la que se utiliza actualmente en Internet. Este es un buen ejemplo de cómo BGP se adapta a las necesidades de Internet a la

hora de elección de destino, ya que el propio protocolo no permite que se publiquen aquellas rutas que no son las seleccionadas como mejores.

Aunque le fuese enviada otra ruta hacia el AS 64700. el sistema autónomo 64512 enviará el tráfico a través del 64520 y este será el que determine hacia donde enviar el tráfico, que será a través de su mejor ruta que es la mencionada anteriormente. Lo que se si que podría hacer el AS 64512 para modificar el camino hacia el 64700 sería enrutar su tráfico hacia otro sistema autónomo vecino como el 64530, esta decisión de "next-hop" la tomará basándose en sus propias políticas BGP internas.

3.2 Tablas BGP



BGP almacena en una tabla la lista de vecinos con los que tiene una conexión BGP. Como se puede observar en la figura, el router tiene su propia tabla para almacenar toda la información, enviada como recibida, mediante el protocolo BGP. El propio router es el que se encarga de ofrecer la mejor ruta de su tabla BGP a la IP, y además se puede configurar para que se intercambien información constantemente.

Figura 4. Tablas de enrutamiento

Para establecer una relación de adyacencia entre dos routers, hay que configurarlo explícitamente en cada uno de los routers vecinos. Una vez se haya configurado, BGP crea una conexión TCP entre ambos que se mantiene activa mediante los mensajes de "keepalive" que se envían periódicamente.

Después de que se haya generado esa adyacencia, los vecinos BGP intercambian sus mejores rutas. Cada router almacena las rutas, que adquiere de cada vecino que tiene conectado, en su tabla BGP. La ruta que cada router propaga como la mejor opción es aquella que cumple con unos atributos determinados, tal y como se vio en el anterior apartado, esa ruta es comparada con las que el router tiene en su propia tabla BGP. Después de otro proceso de selección, ese router decidirá que ruta envía a su tabla IP.

Cada router debe de tener una ruta BGP óptima a cada destino, pero esa ruta también es posible que no esté instalada en la tabla IP porque puede tener una distancia administrativa mayor que otra ruta. Sin embargo, esa ruta sí que será la que se propague por BGP al resto de routers adyacentes.

3.3 Tipos de Mensaje BGP

Existen cuatro tipos de mensajes BGP que son los siguientes:

3.3.1 OPEN y KEEPALIVE

Después de que la conexión TCP sea establecida, el primer mensaje que se envía desde cada lado es un mensaje de ***"open".*** Si este mensaje es aceptado, se envía un mensaje, por parte del receptor

del mensaje ***"open",*** de ***"keepalive"*** confirmando que se ha recibido el paquete ***"open".*** De esta forma, la conexión BGP se establece; y a partir de este momento, se puede comenzar a enviar mensaje ***"update", "keepalive"*** y ***"notification"***.

Inicialmente, los nodos BGP intercambian su tabla de rutas BGP al completo. A partir de entonces, solo se enviarán mensajes de ***"update"*** incrementales. Los ***"keepalive"*** se envían para verificar que la conexión entre ambos nodos BGP sigue establecida, y los mensajes de ***"notification"*** se envían para notificar errores o eventos especiales.

El formato de los mensajes ***"open"*** BGP tiene el siguiente formato:

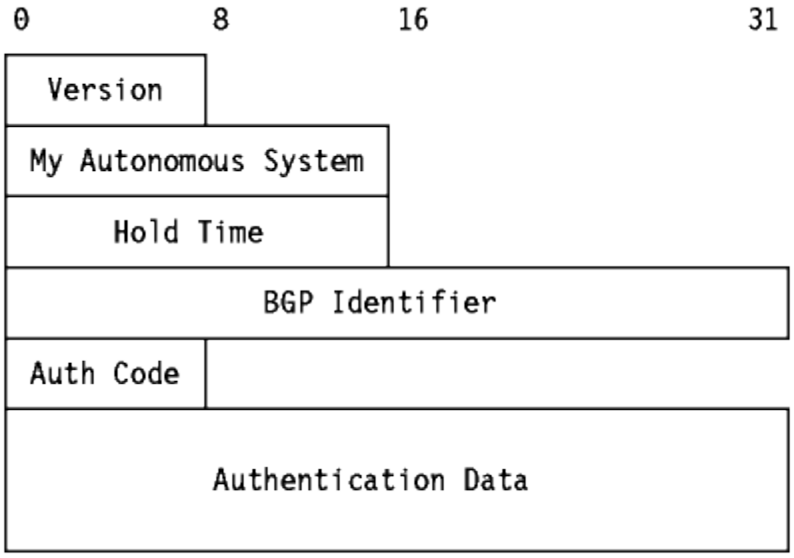


Figura 5. Formato mensajes *OPEN*

- **Version**: Es un campo de 8 bits que indica el número de versión del mensaje BGP. se utiliza para el establecimiento de una sesión BGP una vez haya sido establecida la conexión TCP. Se suelen negociar ciertos parámetros que caractericen a esa sesión. Por ejemplo, es muy posible que los miembros de la sesión no tengan la misma versión de BGP por lo que es importante indicar el número de versión en este campo.
- **My Autonomous System**: Es un campo de 16 bits que indica el número de AS (sistema autónomo) del remitente del mensaje. Los nodos verifican esta información, y si no se recibe el número de AS que se esperaba, entonces la sesión BGP finaliza.
- **Hold time**: Es un campo de 16 bits que indica el máximo número de segundos que pueden pasar entre varios mensajes sucesivos de ***"keepalive"*** del remitente. En cuanto se recibe un mensaje ***"open",*** el router calcula el valor de este campo para usarlo conjuntamente con su vecino; la decisión de cual usar se toma conjuntamente y se decide utilizar el valor más bajo (que tiene un valor por defecto de 180 segundos).
- **BGP Identifier**: Es un campo de 32 bits indica el identificador BGP del remitente, también es conocido como BGP router ID. Se trata de una dirección IP asignada a ese router que se configura en su inicio. El proceso de selección es similar al del protocolo OSPF, donde se escoge la dirección IP activa del router más elevada, a no ser que haya configurado un interfaz de "loopback". Sin embargo, también es posible configurarlo manualmente y sobrescribir la selección automática.
- **Parametros Opcionales** (Auth Code, Authentication Data, …etc)

BGP no utiliza ningún protocolo de transporte basado en ***"keepalives"*** para determinar si los diferentes nodos son alcanzables por el resto de la red. Sin embargo, los mensajes ***"keepalive"*** propios de BGP son intercambiados entre los nodos constantemente para evitar que el parámetro que hemos visto anteriormente de "Hold Time" expire. Estos mensajes de ***"keepalive"*** solamente

constan de una cabecera y tienen un tamaño de 19 bytes; por defecto están configurados para enviarse cada 60 segundos.

3.3.2 UPDATE

Un mensaje de ***"update"*** tiene información acerca de una sola ruta, por lo que, una red que conste de múltiples rutas requiere de múltiples mensajes. Todos los atributos de un mensaje de ***"update"*** se refieren a la ruta, y a las redes que son alcanzables mediante esa ruta.

El formato de los mensajes ***"update"*** BGP tiene el siguiente formato:

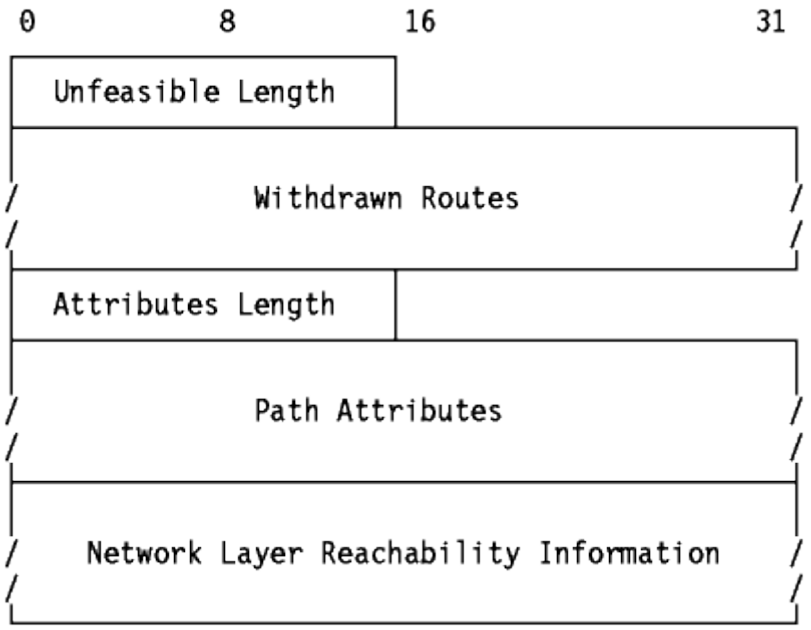


Figura 6. Formato mensajes *UPDATE*

- **Withdrawn Routes**: es un campo de 32 bytes, que contiene un listado con los prefijos de las direcciones IP pertenecientes a las rutas que se están "retirando", si hubiera alguna. Con retirando se refiere a rutas que por cualquier motivo dejen de ser utilizadas.
- **Path Attributes**: es un campo de 32 bytes que incluye todos los parámetros que el protocolo BGP utiliza más adelante para tomar una decisión de que rutas ha de propagar. Cada atributo incluye los siguientes campos: tipo, longitud y valor (TLV). El tipo de atributo consiste en un "flag", seguido del código del tipo de atributo.ç
- **Network Layer Reachability Information (NLRI)**: campo de 32 bytes, que contiene un listado de las redes que pueden ser alcanzadas a través de esta ruta.

3.3.3 NOTIFICATION

Un router BGP envía un mensaje de ***"notification"*** cuando se detecta un error o algún evento especial. Cuando esto ocurre, el propio router finaliza la conexión BGP inmediatamente después de enviar el mensaje de notificación. Este mensaje incluye un código de error, un subcódigo e información relativa al error que puede servir para analizar lo que ha sucedido.

BGP puede llevar al router a través de cualquiera de los estados siguientes, según la conexión que tenga en cada momento con cada uno de sus vecinos BGP:

State	Listen for TCP?	Initiate TCP?	TCP Up?	Open Sent?	Open Received?	Neighbor Up?
Idle	No					
Connect	Yes					
Active	Yes	Yes				
Open sent	Yes	Yes	Yes	Yes		
Open confirm	Yes	Yes	Yes	Yes	Yes	
Established	Yes	Yes	Yes	Yes	Yes	Yes

Figura 7. Diferencias entre distintos estados del protocolo BGP

Gracias a ello, se puede conocer el estado de las conexiones en todo momento, ya que como se puede ver en la tabla, solo cuando una conexión está establecida se pueden enviar mensajes de ***"update", "keepalive"*** y ***"notification".***

3.4 Atributos BGP y proceso de selección de ruta

El protocolo BGP puede ser utilizado para ejecutar enrutamiento basado en políticas, lo que en términos de seguridad se conoce como PBR (policy-based routing). Para poder llevarlo a cabo se debe manipular los mejores caminos que son elegidos por BGP para propagarse a sus routers vecinos. En este apartado se procederá a explicar cómo BGP selecciona su mejor ruta, y los atributos que utiliza para tomar esta decisión y como configurarlos.

3.4.1 Selección de ruta BGP

Un router que ejecute BGP puede recibir actualizaciones sobre las posibles rutas desde múltiples vecinos, algunos pueden ser del mismo AS, y por lo tanto pueden existir múltiples rutas para alcanzar una única red; estas rutas son almacenadas en la tabla BGP del router. El proceso de selección de la mejor ruta consiste en evaluar estas rutas que llegan al router, de manera que aquellas que no son la mejor opción son eliminadas del proceso, aunque si que se quedan almacenadas en la tabla del router por si en un futuro se diera el caso de que la mejor ruta no se encuentra disponible.

BGP no está configurado para implementar balanceo de carga, las rutas son escogidas basándose en una determinada política de selección; la elección no está basada en parámetros como el ancho de banda. Una vez que la mejor ruta se determine, esta se envía a la tabla de enrutamiento IP y se evalua conjuntamente con otras rutas proporcionadas por otros protocolos de enrutamiento como OSPF, RIP… , que también proporcionen alcanzabilidad de la red en cuestión. Finalmente, la ruta con la menor distancia administrativa se instala en la tabla de enrutamiento del router.

3.4.1.1 Proceso de selección

El proceso de selección de ruta BGP está basado, como hemos visto en puntos anteriores, en los denominados como atributos BGP. En el caso de que existan multiples rutas para alcanzar un mismo destino, BGP escogerá la mejor ruta para poder enrutar tráfico a través de ese destino. Para escoger la mejor ruta, se consideran únicamente rutas en las que no haya AS-loops (bucles dentro del propio sistema autonómo) y una dirección de siguiente salto válida. El proceso de selección de la mejor ruta sigue un esquema como el siguiente:

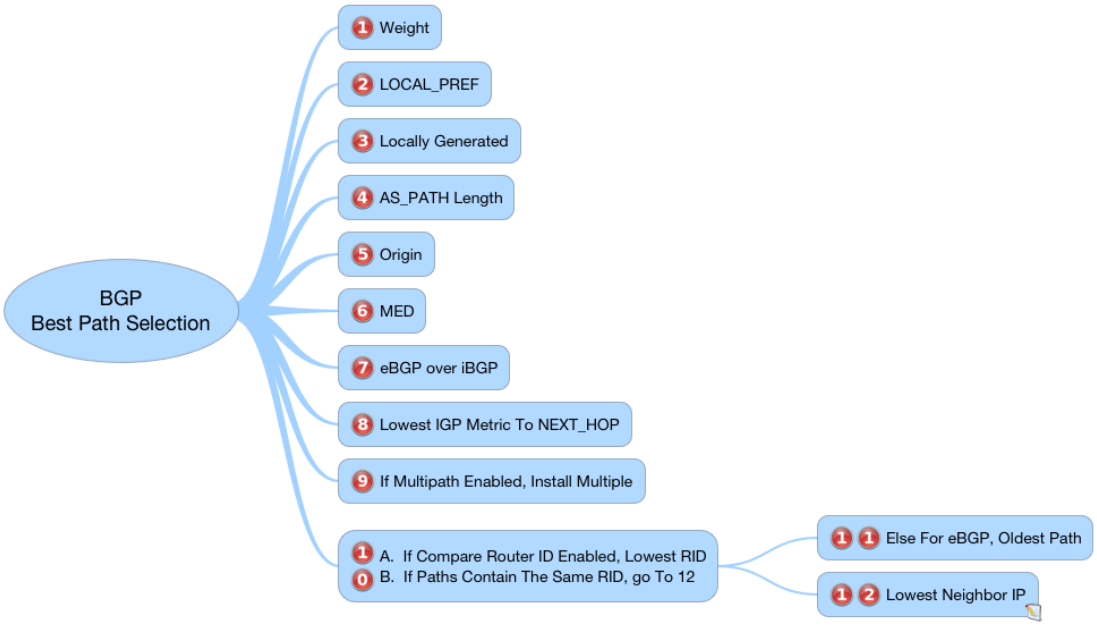


Figura 8. Proceso de selección de ruta

1. Se escoge la ruta con el mayor *"peso" (weight)*, es una forma de expresar un parámetro utilizado por los routers para definir preferencias entre rutas.
2. Si múltiples rutas tienen el mismo *"peso"*, se escoge la ruta con el valor de *"local preference"* más alto (*"local preference"* es un valor utilizado dentro de un mismo AS)
3. Si múltiples rutas tienen el mismo valor de *"local preference"*, se escoge la ruta originada por el router local, es decir, que el siguiente salto sea el 0.0.0.0
4. Si ninguna de las rutas se origina en el router local, es escoge la ruta con el camino más corto entre AS.
5. Si la distancia entre AS es la misma, se escoge la ruta con el menor código de origen (IGP>EGP>incompleto)
6. Si los códigos de origen son todos iguales, se escoge aquella con el menor valor de MED (el MED es un valor que se intercambian varios AS). La comparación del MED se realiza solo si los AS vecinos son iguales para todas las rutas que se estén considerando en este punto del proceso.
7. Si las rutas tienen el mismo MED, tienen preferencia las rutas externas (eBGP) sobre las internas (iBGP).
8. Si solo hay rutas internas, se escoge la ruta a través del vecino IGP más cercano. Esto significa que el router busca el camino más corto a través de un mismo AS para llegar al siguiente vecino BGP.

9. Para los protocolos de rutas externas (eBGP), se selecciona la ruta más antigua para minimizar los efectos que podrían tener las rutas inconsistentes.
10. Se escoge la ruta con el menor router ID del vecino BGP
11. Si los router ID de los diferentes vecinos BGP son iguales, se escoge aquella con la "menor" dirección IP del vecino.

Finalmente, únicamente se ofrecerá la mejor ruta a la tabla IP del router y se propagará a todos los vecinos BGP.

3.4.1.1.1 Proceso de selección de major ruta en un entorno multihomed

Un AS raramente implementa el protocolo BGP con una sola conexión eBGP (external BGP), por lo que generalmente, siempre nos encontraremos el caso de que existen múltiples rutas para alcanzar una misma red basándose en las decisiones de la tabla de enrutamiento BGP. Para analizar el proceso expuesto anteriormente vamos a exponer un caso de estas características para ver cómo sería el proceso de selección de la mejor ruta.

1. Se analiza el *"peso"*, que por defecto se establece a 0 para todas las rutas que no se originen en nuestro router.
2. Se compara el valor de *"local preference"*, por defecto se establece a 100 para todas las redes. Por lo que observando el paso 1 y 2 se puede concluir que estos últimos solo serán motivo de decisión si el administrador de la red configura estos parámetros manualmente.
3. Se observan las redes publicadas en el propio AS. Si una de las rutas es introducida en la tabla BGP por el router local, el router preferirá esa ruta por delante de cualquier otra recibida por uno de sus vecinos BGP.
4. Se selecciona la ruta que pase por menos AS antes de llegar a su destino. Este paso es el que normalmente se utiliza para la elección de rutas mediante el protocolo BGP. Si el administrador de la red no quiere que esta decisión sea automática tendrá que manipular los valores de *"local preference"* o *"peso"* como hemos visto anteriormente.
5. Se analiza cómo se ha introducido la red en BGP. Esta inserción se hace habitualmente a través del comando *"network"* (como veremos más adelante) o mediante redistribución de rutas.
6. Se analiza el MED para decidir dónde debe enviar el tráfico el AS basándonse en la configuración de los AS vecinos. Sin embargo, en pocas ocasiones veremos que este punto sea el clave para la decisión de la mejor ruta; ya que a no ser que el administrador de la red modifique este parámetro, por defecto estará siempre a 0.
7. Si existen múltiples rutas con el mismo número de AS que deben atravesar se tomará la decisión basándose en si esas rutas han sido aprendidas mediante un vecino eBGP (externo) o iBGP (interno); teniendo preferencia el externo. La explicación es sencilla, y es que un router prefiere utilizar el ancho de banda de un ISP antes que el ancho de banda interno para llegar a un destino.
8. Si el número de AS hasta el destino es el mismo, y no existen vecinos eBGP para esa red, se escogerá el camino más rápido (menos saltos) hasta el siguiente vecino BGP.
9. Si todas las rutas tienen el mismo número de saltos, se escogerá el camino que más tiempo lleve establecido en la tabla de rutas. Este es un aspecto obvio, puesto que la permanencia de una ruta a lo largo del tiempo indicará la estabilidad de la misma.
10. Si con ninguno de los anteriores pasos se puede determinar la mejor ruta hacia un destino, la decisión sería la de seleccionar la ruta que vaya a través del router vecino BGP con el menor ID.
11. En el caso de que el ID de los routers fuera el mismo, se utilizaría la ruta en la cual se encuentre el vecino BGP con una dirección IP menor que el resto.

3.4.2 Atributos BGP

Los routers BGP envían mensajes de *"update"* sobre las redes alcanzables al resto de vecinos BGP. Estos mensajes contienen un parámetro denominado NLRI (lista de una o más redes junto con sus direcciones y prefijos IP) y los conocidos atributos BGP, estos atributos son utilizados para determinar la ruta óptima hacia una determinada red. Para definir estos atributos hay que tener claros ciertos términos:

- Un atributo se categoriza como *"well-known"* u *"optional"*, *"mandatory"* o *"discretionary"* y "*transitive"* o *"nontransitive"*. También pueden ser *"partial"*
- Sin embargo, no todos las combinaciones de estas características son válidas, las combinaciones aceptadas son:
    - Well-known mandatory
    - Well-known discretionary
    - Optional transitive
    - Optional nontransitive
- Solo los atributos "optional transitive" pueden ser marcados como *"partial"*

3.4.2.1 Well-Known Attributes

Un atributo *"Well-Known"* es aquel que todas las implementaciones de BGP deben reconocer y propagar al resto de vecinos BGP. Como hemos visto anteriormente, hay dos tipos:

- ***Well-known mandatory attribute***: debe aparecer en todos los mensajes de actualización BGP. Si un atributo de este tipo no aparece en un mensaje de actualización, se genera una notificación de error. Esta medida garantiza que todas las implentaciones de BGP cumplan un estándar a la hora de establecer atributos.
- ***Well-Know discretionary attribute***: en este caso, no es necesario que aparezca en el mensaje de actualización BGP.

3.4.2.2 Optional Attributes

Los routers BGP que implementen cualquier *"Optional Attribute"* propagaran o no estos al resto de los vecinos BGP, dependiendo del contenido de los mismos. También tenemos dos tipos:

- ***Optional transitive***: los routers BGP que no implementen un "*optional transitive attribute"* deberán enviar ese atributo al resto de routers BGP sin modificarlo y marcar el atributo como "*partial"* (el otro tipo de atributos que hemos definido anteriormente).
- ***Optional nontransitive***: los routers BGP que no implementen un atributo de este tipo, deberán eliminar el atributo y no enviarlo al resto de routers BGP.

3.4.2.3 Defined BGP Attributes

Como se ha definido en los capítulos anteriores, los atributos BGP incluyen los siguientes:

- **Well-known mandatory attributes**
  - AS-path
  - Next-hop
  - Origin

- **Well-known discretionary attributes**
  - Local preference
  - Atomic aggregate

- **Optional transitive attributes**
  - Aggregator
  - Community

- **Optional nontransitive attribute**
  - MED

3.4.2.3.1 AS-Path

Este atributo se corresponde con la lista de AS que una ruta tiene que atravesar para alcanzar un determinado destino, además se incorpora la información del AS que originó la ruta. Como bien se ha explicado en capítulos anteriores, es un *"Well-known mandatory attribute"*. Por lo que siempre que un mensaje de actualización de una ruta pase a través de un AS, el número de AS se adjunta al mensaje de actualización.

Por ejemplo, en la siguiente figura, el router A que se encuentra en el AS 64520 propaga la red 192.168.1.0. Cuando esa ruta atraviesa el AS 65500, el router C adjunta su número de AS al mensaje. Por lo tanto, cuando la ruta llega al router B, tiene dos AS adjuntos; y desde el router B la ruta para alcanzar la red 192.168.1.0 será a través de los AS (65500, 64520).

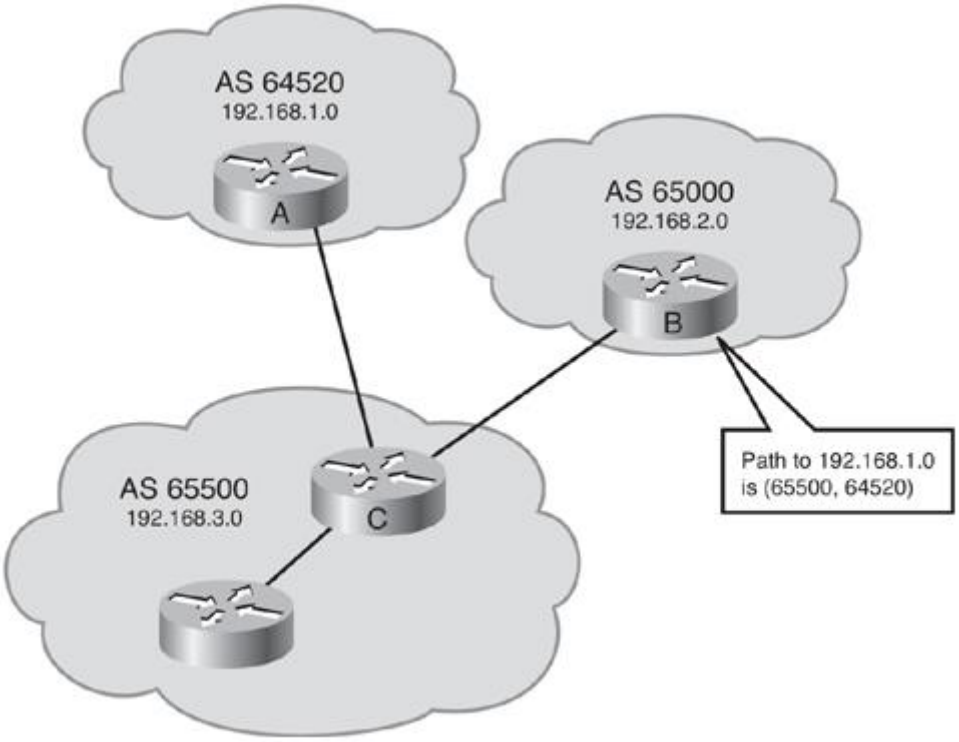


Figura 9. Ejemplo utilización *AS-PATH*

El mismo caso sería para la red 192.168.2.0 y la 192.168.3.0. El camino del router A para llegar a la red 192.168.2.0 es a través de los AS (65500 65000). En el caso del router C solo tendría que atravesar el AS 65000 para alcanzar la red 192.168.2.0 y el AS 64520 para alcanzar la 192.168.1.0.

Los routers BGP usan el atributo ***"AS-Path"*** para verificar que se trata de una red libre de bucles. Si un router BGP recibe una ruta en la cual su propio AS es parte del atributo ***"AS-Path"***, no se acepta. Los números de AS solamente son adjuntados a la ruta por los routers que propagan sus rutas a los vecinos eBGP. Los routers que propagan rutas a los vecinos iBGP no modifican el ***AS-Path***.

3.4.2.3.2 Next-Hop

Este atributo como se ha visto antes es un *"Well-known mandatory attribute"* que indica la dirección IP del siguiente salto que se va a utilizar para llegar a un determinado destino. En capítulos anteriores se explicó que para un router eBGP la dirección IP de siguiente salto es la dirección IP del vecino que ha enviado el mensaje de actualización.

Aunque esto puede modificarse si se configura manualmente un router para que se publique a si mismo como la dirección de next-hop para las rutas que envíe a sus vecinos.

3.4.2.3.3 Origin

Este atributo es el último perteneciente al grupo de *"Well-known mandatory attribute"*. En este caso, este atributo define el origen de información de la ruta. Dependiendo de sus características puede ser cualquiera de los siguientes valores:

- IGP: la ruta está dentro del AS que la origina. Esto como veremos más adelante, sucede cuando se utiliza el comando *"network"* para propagar una determinada ruta. Un origen del tipo *IGP* se indica en la tabla BGP con una ***"i".***
- EGP: significa que la ruta se ha aprendido mediante EGP, se trata de un protocolo de red antiguo que no está soportado por el Internet, tal y como lo conocemos actualmente; ya que no soporta CIDR. Se indica en la tabla BGP con una ***"e".***
- Incomplete: se indica cuando se desconoce el origen de la ruta o se aprende mediante otros métodos que no son los anteriores. Esto puede ocurrir cuando una ruta es redistribuida en BGP. Se indica en la tabla BGP con un símbolo de interrogación ***"?".***

3.4.2.3.4 Local-Preference

Se trata de un atributo *"Well-known discretionary attribute"* e indica los routers que hay que seguir dentro de un mismo AS para salir del AS hacia un destino determinado. Por lo tanto, un camino con un atributo de ***"local-preference"*** alto será siempre seleccionado frente a otros que lo tengan más bajo.

Este atributo es enviado solo por los vecinos iBGP, no entre los nodos eBGP. Así pues, se trata de un atributo que es configurado en un router y solo se intercambia con aquellos routers que formen también parte del mismo AS. En los routers Cisco el valor por defecto es de 100. Ç

3.4.2.3.5 Community

Las comunidades o grupos BGP son una forma de filtrar las rutas que enviamos o recibimos. Además, permiten a los routers taggear rutas con un indicador *("community")* habilitando de esta manera que otros routers tomen decisiones basadas en este tag o etiqueta. Cualquier router BGP puede taggear rutas tanto en mensajes de actualización entrantes como salientes.

Estas comunidades BGP son también utilizadas para rutas que comparten algunas propiedades y, que, por lo tanto, comparten también unas políticas comunes. Las comunidades no se restringen a una sola red o a un solo AS.

Se trata de un *"Optional transitive attribute"*. Si un router no entiende el concepto de comunidad, se desvía al siguiente router. Sin embargo, si el router puede gestionar y comprender el concepto, se debe configurar para propagar esa comunidad determinada; de lo contrario, se droppearian todas las comunidades por defecto.

3.4.2.3.6 MED

También se denomina como métrica, y se trata de un *"Optional nontransitive attribute"*. El MED indica a los vecinos externos la ruta preferida dentro de un mismo AS. Se trata por tanto, de una forma de influenciar por parte de un AS sobre otro, en la forma de la que este último escogerá una cierta ruta para alcanzar un destino habiendo múltiples opciones para llegar a la misma.

Como ocurre con otros protocolos de enrutamiento, es preferible un valor de MED lo más bajo posible. A diferencia del atributo de "local-preference", el valor del MED se intercambia entre AS, se envía a los nodos eBGP, estos routers propagan el MED en su propio AS, pero no lo envían al siguiente AS. Cuando se envía el mismo mensaje de actualización en otro AS, la métrica se vuelve a poner en su valor por defecto, 0. Por tanto, la principal diferencia con el atributo de "local-preference", es que el MED tiene una influencia directa sobre el tráfico entrante en un AS y el "local-preference" lo tiene sobre el tráfico saliente.

Por defecto, el router compara el atributo MED solo con las rutas de los vecinos que tiene en un mismo AS.

Mediante el uso del MED, BGP es el único protocolo que puede verse influenciado en la decisión de las rutas a través de las cuales va a enviarse el tráfico. Sin embargo, hay que tener en cuenta que cuando se envié el atributo MED el siguiente AS puede decidir sus rutas basándose en otros atributos.

Por ejemplo, en la siguiente figura, tanto el router B como el C, se han configurado con los comandos que se pueden ver en la parte inferior. Más adelante se profundizará en este tipo de configuraciones, pero a nivel básico simplemente comentar que el comando "**ip prefix-list** [*list-name* | *list-number*] [**seq** *seq-value*] [**deny | permit**] *network/length*", se usa para crear la lista de prefijos. Para establecer un mapa de rutas para BGP, se utiliza el comando "**neighbor** *ip address* **route-map** *name* [**in | out**]"

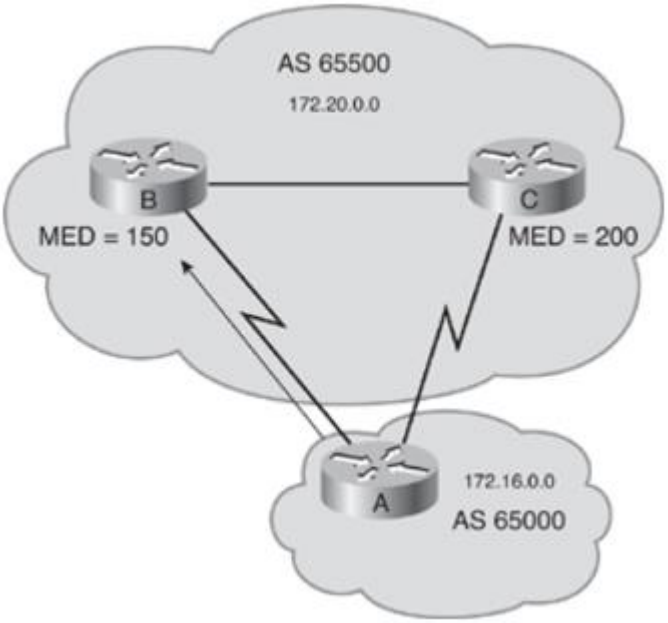


Figura 10. Ejemplo utilización *MED*

Configuración router B

>>>RB (config)# ip prefix-list PF1 permit 172.20.0.0

>>>RB (config)# route-map SET-MED permit 10

>>>RB (config-route-map) # match ip address prefix-list PF1

>>>RB (config-route-map) # set metric 150

>>>RB (config-route-map) # route-map SET-MED permit 20

>>>RB (config-route-map) # exit

>>>RB (config)# router bgp 65550

>>>RB (config-router) # neighbor 172.20.0.2 route-map SET-MED out

Configuración router C

>>>RC (config)# ip prefix-list PF1 permit 172.20.0.0

>>>RC (config)# route-map SET-MED permit 10

>>>RC (config-route-map) # match ip address prefix-list PF1

>>>RC (config-route-map) # set metric 200

>>>RC (config-route-map) # route-map SET-MED permit 20

>>>RC (config-route-map) # exit

>>>RC (config)# router bgp 65550

>>>RC (config-router) # neighbor 172.20.0.3 route-map SET-MED out

En el ejemplo, cuando se envía la ruta 172.20.0.0 al router A en el AS 65000, el router B establece el valor del MED a 150 y el router C lo establece a 200. Cuando el router A recibe el mensaje de actualización de los routers B y C (que incluyen los *"path attributes"*), selecciona el router B como el mejor siguiente salto para alcanzar la red 172.20.0.0 en el AS 65500 porque el MED del router B es menor que el del C.

Hay que tener en cuenta que en un proceso normal de actualización de BGP la comparación del MED no sería el primer paso. Para que un router compare las métricas que le lleguen de los vecinos de los distintos AS, hay que utilizar el comando ***"bgp always-compare-med".***

3.4.3 Modificando el proceso de selección de ruta

A lo largo de este capítulo nos basaremos en la siguiente imagen para explicar la influencia que podemos tener a la hora de selección de rutas en una infraestructura de red. Partiremos de la base de que las conexiones BGP entre los diferentes nodos ya están establecidas, GW1 y GW2 están propagando las redes configuradas en su interfaz de loopback 1 (Lo1) en BGP, y el ISP3 esta propagando las dos redes en su interfaz loopback 0 (Lo0) también en BGP; para poder ver el ejemplo de una redistribución de rutas utilizaremos el protocolo OSPF en el AS 65000, es decir entre GW1 y GW2.

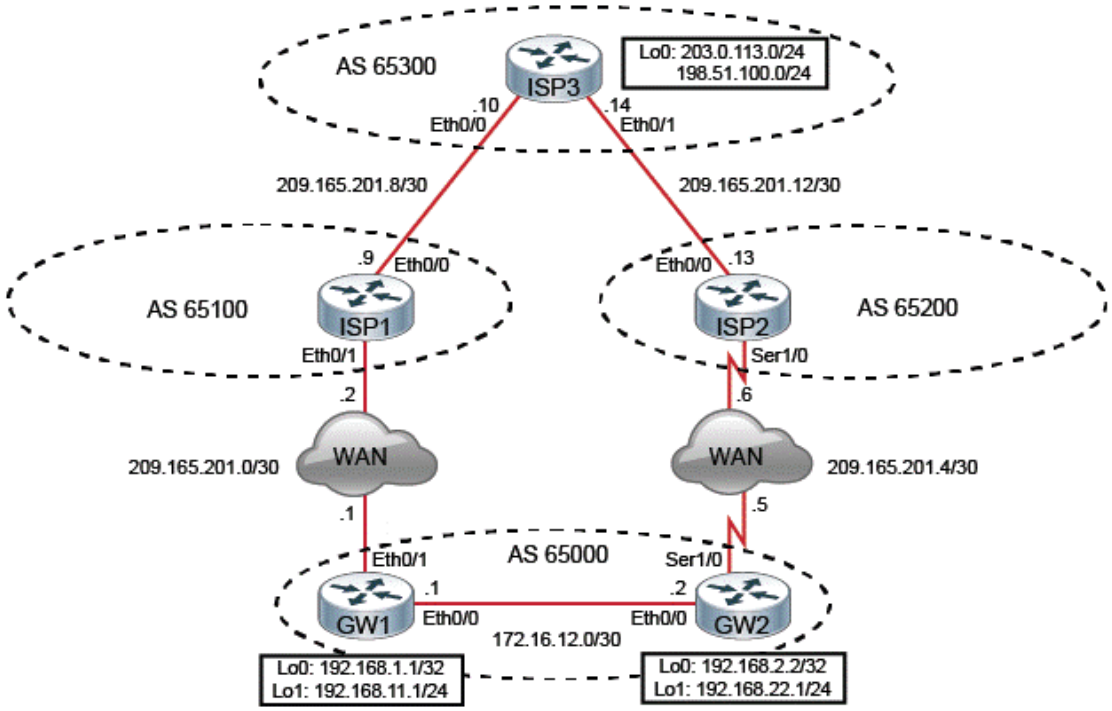


Figura 11. Esquema de red utilizado para la modificación del proceso de selección de ruta

Después de un previo estudio, se modificarán algunos de los atributos BGP para modificar el camino del tráfico. En primer lugar, asignaremos un peso más alto a los mensajes de actualización que se reciben en GW2 procedentes del GW1, para elegir GW1 como punto de salida del AS 65000. Una vez que se compruebe que se ha aplicado correctamente ese cambio, cambiaremos a una preferencia de *"local preference"* en GW1 para los mensajes de actualización que se reciben del ISP1. Finalmente, provocaremos que todo el tráfico de entrada al AS 65000 escoge el camino más rápido, configurando el *"AS-path"* mediante las actualizaciones recibidas por GW2 desde ISP2.

Por lo tanto, en primer lugar, comprobaremos la configuración inicial de BGP en GW1 examinando tanto la tabla BGP como la IP; también verificaremos la conectividad a las redes externas propagadas por el ISP3. Para verificar la conectividad entre los distintos AS (65000 65300) desde GW1, necesitamos enviar tráfico con origen en una dirección IP que sea alcanzable desde el ISP3.

Verificación BGP en GW1

>>>GW1#show ip bgp

BGP table version is 20, local router ID is 209.165.201.1

*Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,*

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? – incomplete

RPKI validation codes: V valid, I invalid, N Not found

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.168.11.0	0.0.0.0	0	0	32768	i
*>i	192.168.22.0	192.168.2.2	0	100	0	i
*>	198.51.100.0	209.165.201.2			0	65100 65300 i
* i		209.165.201.6	0	100	0	65200 65300 i
*>	203.0.113.0	209.165.201.2			0	65100 65300 i
* i		209.165.201.6	0	100	0	65200 65300 i

>>>GW1#show ip route bgp

Gateway of last resort is not set

B 192.168.22.0/24 [200/0] via 192.168.2.2

B 198.51.100.0/24 [20/0] via 209.165.201.2

B 203.0.113.0/24 [20/0] via 209.165.201.2

>>>GW1#ping 198.51.100.1 source loopback 1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 198.51.100.1, timeout is 2 seconds:

Packet sent with a source address of 192.168.11.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms

>>>GW1#traceroute 198.51.100.1 source loopback 1

Type escape sequence to abort

Tracing the route to 198.51.100.1

VRF info: (vrf in name/id, vrf out name/id)

1 209.165.201.2 0 msec 0 msec 1 msec

*2 209.165.201.10 4 msec * 4 msec*

>>>GW1#traceroute 203.0.113.1 source loopback 1

Type escape sequence to abort

Tracing the route to 203.0.113.1

VRF info: (vrf in name/id, vrf out name/id)

1 209.165.201.2 1 msec 0 msec 1 msec

*2 209.165.201.10 0 msec * 1 msec*

La table BGP e IP del GW1 muestra que los prefijos proporcionados por el ISP3 198.51.100.0/24 y 203.0.113.0/24 son recibidos mediante las rutas de ambos ISPs. GW1 prefiere las rutas externas via ISP1 porque, como hemos visto anteriormente, las rutas externas se escogen antes que las internas. Las direcciones de loopback del ISP3 son alcanzables desde el GW1. El mismo

procedimiento de comprobación que hemos hecho para el GW1, lo hacemos también para el GW2. Mediante el cual obtendremos unos resultados similares al anterior.

Verificación BGP en GW2

```
>>>GW2#show ip bgp
BGP table version is 15, local router ID is 192.168.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? – incomplete
RPKI validation codes: V valid, I invalid, N Not found
     Network          Next Hop            Metric  LocPrf  Weight  Path
*>i  192.168.11.0     192.168.1.1              0     100      0     i
*>   192.168.22.0     0.0.0.0                  0              32768 i
* i  198.51.100.0     209.165.201.2            0     100      0     65100 65300 i
*>                    209.165.201.6                           0     65200 65300 i
* i  203.0.113.0      209.165.201.2            0     100      0     65100 65300 i
*>                    209.165.201.6                           0     65200 65300 i

>>>GW2#show ip route bgp
Gateway of last resort is not set
B       192.168.11.0/24     [200/0]    via 192.168.1.1
B       198.51.100.0/24     [20/0]     via 209.165.201.6
B       203.0.113.0/24      [20/0]     via 209.165.201.6

>>>GW2#ping 198.51.100.1 source loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.51.100.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.22.1
!!!!!
Success rate is 100 percent   (5/5), round-trip min/avg/max  =  7/8/9 ms

>>>GW2#traceroute 198.51.100.1 source loopback 1
Type escape sequence to abort
Tracing the route to 198.51.100.1
VRF info:   (vrf in name/id, vrf out name/id)
1  209.165.201.6   8 msec   8  msec  8 msec
2  209.165.201.14   8 msec  *  6 msec

>>>GW2#traceroute 203.0.113.1  source loopback 1
Type escape sequence to abort
Tracing the route to 203.0.113.1
VRF info:   (vrf in name/id, vrf out name/id)
1  209.165.201.6   7 msec   9  msec  9 msec
2  209.165.201.14   9 msec  *  9 msec
```

Por último, verificaremos que estado inicial y la conectividad del ISP3.

Verificación BGP en ISP3

```
>>>ISP3#show ip bgp
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? – incomplete
RPKI validation codes: V valid, I invalid, N Not found
     Network          Next Hop            Metric  LocPrf  Weight  Path
*    192.168.11.0     209.165.201.13                       0       65200 65000 i
*>                    209.165.201.9                        0       65100 65000 i
*    192.168.22.0     209.165.201.9                        0       65100 65000 i
*>                    209.165.201.13                       0       65200 65000 i
*> 198.51.100.0       0.0.0.0                  0           32768   i
*> 203.0.113.0        0.0.0.0                  0           32768   i

>>>ISP3#show ip route bgp
Gateway of last resort is not set
B       192.168.11.0/24     [20/0]    via 209.165.201.9
B       192.168.22.0/24     [20/0]    via 209.165.201.13

>>>ISP3#ping 192.168.11.1  source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.11.1, timeout is 2 seconds:
Packet sent with a source address of 198.51.100.1
!!!!!
Success rate is 100 percent   (5/5), round-trip min/avg/max  =  1/1/1 ms

>>>ISP3#traceroute 192.168.22.1 source loopback 0
Type escape sequence to abort
Tracing the route to 192.168.22.1
VRF info:   (vrf in name/id, vrf out name/id)
1  209.165.201.13   1 msec   0  msec  1 msec
2  209.165.201.5   9 msec  *  9 msec
```

Tanto el interfaz de loopback del GW1 como el del GW2 son alcanzables desde el ISP3. La tabla BGP del ISP3 muestra que los prefijos del AS 65000 192.168.11.0/24 y 192.168.22.0/24 son recibidos vía los dos ISP. Ambas actualizaciones tienen los mismos atributos, por lo que el ISP3 selecciona la ruta más antigua.

Por lo cual, el ISP3 elige un camino distinto para cada uno de los prefijos del mismo AS; este es un caso que no podría producirse en un entorno de producción puesto que podría ocasionar una serie de problemas muy importantes ya que al elegir la ruta con mayor antigüedad pudiera ser que se esté utilizando el interfaz WAN que va desde el ISP2 al GW2, y que este tenga un menor ancho de banda del que disponemos con el ISP1. Para que esto no se produzca vamos a simular otras configuraciones en las que el camino se escoja según nuestra preferencia.

3.4.3.1 Cambio del atributo "Weight"

A partir de este punto, como hemos mencionado anteriormente, vamos a explorar opciones para alterar la ruta escogida por los diferentes nodos para alcanzar una red determinada. En primer lugar, cambiaremos el *"peso"* por defecto en GW2 para todos los mensajes de actualización recibidos desde GW1 a un valor distinto de 0; de esta forma, provocaremos que los prefijos de red recibidos vía iBGP sean escogidos por delante del resto.

Modificación del atributo "weight" en GW2

```
>>>GW2(config)# router bgp 65000
>>>GW2(config-router)# neighbor 192.168.1.1 weight 10

>>>GW2#show ip bgp
      Network          Next Hop            Metric  LocPrf  Weight  Path
*>i   192.168.11.0     192.168.1.1              0     100      0      i
*>    192.168.22.0     0.0.0.0                  0              32768  i
* i   198.51.100.0     209.165.201.2            0     100      0      65100 65300 i
*>                     209.165.201.6                           0      65200 65300 i
* i   203.0.113.0      209.165.201.2            0     100      0      65100 65300 i
*>                     209.165.201.6                           0      65200 65300 i
```

En la tabla se comprueba, que a pesar de haber modificado el *"peso"* por defecto para el GW1 en el GW2, la tabla BGP del GW2 no refleja ese cambio de políticas inmediatamente. El protocolo BGP solo aplica las nuevas políticas cuando se intercambian los mensajes de actualización, por lo que para ver esos cambios tenemos que provocar un reseteo de manera que tanto el GW1 como el GW2 se intercambien mensajes de actualización BGP.

Reseteo BGP en GW2 y verificación de los resultados

```
>>>GW2# clear ip bgp 192.168.1.1 in
>>>GW2#show ip bgp
      Network          Next Hop            Metric  LocPrf  Weight  Path
*>i   192.168.11.0     192.168.1.1              0     100      10     i
*>    192.168.22.0     0.0.0.0                  0              32768  i
*>i   198.51.100.0     209.165.201.2            0     100      10     65100 65300 i
*                      209.165.201.6                           0      65200 65300 i
*> i  203.0.113.0      209.165.201.2            0     100      10     65100 65300 i
*                      209.165.201.6                           0      65200 65300 i
>>>GW2#traceroute 198.51.100.1 source loopback 1
Type escape sequence to abort
Tracing the route to 198.51.100.1
VRF info:  (vrf in name/id, vrf out name/id)
1  172.16.12.1  1 msec  1 msec  0 msec
2  209.165.201.2  1 msec  0 msec  1 msec
3  209.165.201.10  3 msec *  5 msec
```

Se puede observar, que después de haber hecho un reseteo en el GW2, GW1 vuelve a enviar los mensajes de actualización BGP al GW2 y este último aplica la nueva política que habíamos configurado previamente. Ahora, GW2 prefiere los mensajes recibidos por GW1 antes que aquellos que recibe del ISP2. El tráfico encaminado hacia las redes externas (como puede ser la 198.51.100.1) va a través del GW1 y el ISP1. Esta política que hemos aplicado solo tendrá efecto sobre GW2, el resto de rutas no se verán modificadas.

3.4.3.2 Cambio del atributo "Local Preference"

El atributo *"local preference"* se comparte con todos los nodos de un mismo AS, por lo que, a diferencia del anterior, los cambios que realicemos aquí sí que afectarán tanto al GW1 como al GW2. Para realizar el cambio del valor de este atributo tenemos varias opciones, en primer lugar, podemos cambiar el valor por defecto (100), mediante el protocolo ***bgp default local-preference,*** de esta forma todas las rutas propagadas contendrán el nuevo valor de *"local preference"*. Otro método, es mediante un mapa de rutas; este será el método utilizado para hacer el cambio en nuestro esquema de red. Para ello, estableceremos un valor de *"local preference"* más alto en GW1 para los mensajes de actualización que se reciban del ISP1, hay que recordar desactivar el *"peso"* que hemos modificado en el anterior capítulo.

Modificación del atributo "local-preference" en GW1

```
>>>GW2(config)# router bgp 65000
>>>GW2(config-router)# no neighbor 192.168.1.1 weight 10
>>>GW1(config)# route-map prefer_isp1 permit 10
>>>GW1(config-route-map)# set local-preference 150
>>>GW1(config-route-map)# router bgp 65000
>>>GW1(config-router)# neighbor 209.165.201.2 route-map prefer_isp1 in
```

Al igual que en el anterior capítulo para que los cambios surtan efecto hay que hacer un reset de las conexiones, para que los nodos BGP vuelvan a intercambiarse los mensajes de actualización.

Reseteo BGP en GW1

```
>>>GW1# clear ip bgp 209.165.201.2 in
>>>GW1#show ip bgp
     Network          Next Hop            Metric  LocPrf  Weight  Path
*>   192.168.11.0     0.0.0.0                  0                32768  i
*>i  192.168.22.0     192.168.2.2              0     100        0  i
*>   198.51.100.0     209.165.201.2                 150        0  65100 65300 i
*>   203.0.113.0      209.165.201.2                 150        0  65100 65300 i
```

Después del reseteo se comprueba que efectivamente el valor de "local-preference" se ha establecido a 150 para las redes externas 198.51.100.0/24 y 203.0.113.0/24; ambas redes se alcanzan a través del ISP1. Las rutas sobre esas mismas redes externas recibidas del GW2 han desaparecido de la tabla, esto ocurre porque ahora el GW2 prefiere el camino a través del GW1 y solo se intercambian las mejores rutas al resto de vecinos BGP.

Verificación de que el tráfico desde el GW2 al ISP3 se enruta via el GW1

>>>GW2#show ip bgp

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	192.168.11.0	192.168.1.1	0	100	10	i
*>	192.168.22.0	0.0.0.0	0		32768	i
*>i	198.51.100.0	209.165.201.2	0	150	0	65100 65300 i
*		209.165.201.6			0	65200 65300 i
*>i	203.0.113.0	209.165.201.2	0	150	0	65100 65300 i
*		209.165.201.6			0	65200 65300 i

>>>GW2#traceroute 198.51.100.1 source loopback 1

Type escape sequence to abort

Tracing the route to 198.51.100.1

VRF info: (vrf in name/id, vrf out name/id)

1 172.16.12.1 1 msec 1 msec 0 msec

2 209.165.201.2 1 msec 0 msec 1 msec

*3 209.165.201.10 5 msec * 5 msec*

>>>GW2#show ip route

172.16.0.0/15 is variably subnetted, 2 subnets, 2 masks

C 172.16.12.0/30 is directly connected, Ethernet0/0

L 172.16.12.2/32 is directly connected, Ethernet0/0

192.168.1.0/32 is subnetted, 1 subnets

O 192.168.1.1 [110/11] via 172.16.12.1, 00:03:20, Ethernet0/0

192.168.2.0/32 is subnetted, 1 subnets

C 192.168.2.2 is directly connected, Loopback0

B 192.168.11.0/24 [200/0] via 192.168.1.1, 00:02:50

192.168.22.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.22.0/24 is directly connected, Loopback1

L 192.168.22.1/32 is directly connected, Loopback1

B 198.51.100.0/24 [200/0] via 209.165.201.2, 00:01:01

B 203.0.113.0/24 [200/0] via 209.165.201.2, 00:01:01

209.165.201.0/24 is variably subnetted, 3 subnets, 2 masks

O 209.165.201.0/30 [110/21] via 172.16.12.1, 00:03:20, Ethernet0/0

C 209.165.201.4/30 is directly connected, Serial1/0

L 209.165.201.5/32 is directly connected, Serial1/0

Capítulo 4. ¿Cuándo usar o no usar BGP?

El uso del protocolo BGP dentro de un AS, puede ser mucho más útil cuando los efectos del BGP son bien entendidos y al menos una de las siguientes condiciones existe:

- El AS permite que los paquetes transiten a través de él para alcanzar otros AS (por ejemplo, en un proveedor de servicios ISP).
- El AS tiene múltiples conexiones con otros AS.
- La política de enrutamiento y la selección de rutas para el tráfico que entra y sale del AS debe ser manipulado.

Si una empresa quiere que su tráfico se diferencie del tráfico Internet de su ISP, se debe conectar a ese ISP usando BGP. Si, en lugar de eso, una empresa es conectada a su ISP con una ruta estática, el tráfico de esa empresa en Internet es indistinguible del tráfico del/de los ISP.

BGP fue diseñado para permitir la comunicación y el intercambio de paquetes por parte de los ISPs. Estos, tienen múltiples conexiones a otros proveedores de Internet y tienen acuerdos con ellos para el intercambio de actualizaciones. BGP es el protocolo usado para implementar estos acuerdos entre dos o más AS. Si BGP no está bien controlado y filtrado, tiene el potencial de permitir que un AS exterior afecte al flujo de tráfico que circula en nuestro propio AS. Por ejemplo, si eres un cliente conectado a un ISP A e ISP B (para obtener redundancia en la red), lo lógico es implementar una política de enrutamiento para asegurar que el ISP A no envía tráfico al ISP B mediante el propio AS del cliente. BGP se implementa para ser capaz de recibir tráfico destinado a tu AS a través del ISP, sin malgastar recursos valiosos, como el ancho de banda, del propio AS para encaminar el tráfico hacia los ISPs.

Sin embargo, BGP no siempre es la solución apropiada para interconectar AS. Por ejemplo, si sólo hay una ruta de salida desde el AS, lo más apropiado sería una ruta. En un caso como el anterior, el uso del BGP no logrará nada excepto usar los recursos y la memoria de la CPU del router. Si la política de enrutamiento que será implementada en el AS se ajusta a la política implementada en el ISP del AS, no es necesario configurar BGP en ese AS. La única vez en la que BGP será requerido es cuando la política local difiere de la política del ISP.

No debe usarse BGP si una o más de las siguientes condiciones existe:

- Una sola conexión a Internet o a otro AS.
- Falta de memoria o de potencia del procesador en los routers edge para manejar las actualizaciones constantes BGP.
- Entendimiento limitado del filtrado de rutas y el proceso de selección de rutas BGP.
- Si la política de enrutamiento que será implementada en el AS es consistente con la política de implementación en el ISP del AS.

En estos casos, debe hacerse uso de rutas estáticas.

Capítulo 5. Implementación básica BGP

En este capítulo introduciremos las relaciones entre vecinos BGP y como se establecen. Describiremos también los estados a través de los cuales avanza el protocolo BGP para establecer una sesión entre ambos vecinos. De esta forma podremos explorar los parámetros básicos de una red BGP y los comandos que necesitamos para programarla, el objetivo es sentar las bases para el diseño final de nuestra red BGP.

5.1 BGP Neighbour Relationships

Actualmente, sería imposible pensar en un único router capaz de manejar las comunicaciones con las decenas de miles de routers que ejecutan BGP y están conectados a Internet, representando más de 48000 AS.

Un router BGP forma una relación vecina directa con un número limitado de routers BGP. Mediante estos vecinos BGP, un router BGP aprende los caminos a través de Internet para llegar a cualquier red anunciada.

Para comprender esta sección hay que recordar que cualquier router que ejecute BGP se denomina speaker BGP. Un router vecino BGP, es un speaker BGP que está configurado para formar una relación vecina con otro speaker BGP con el fin de intercambiar directamente información de enrutamiento BGP con otro.

Un speaker BGP tiene un número limitado de vecinos BGP con el que se asocia y forma una relación basada en TCP, como se puede observar en la siguiente figura. Los nodos BGP pueden ser internos o externos a los AS. Ambos tipos requieren de una conexión TCP para el correcto funcionamiento del protocolo.

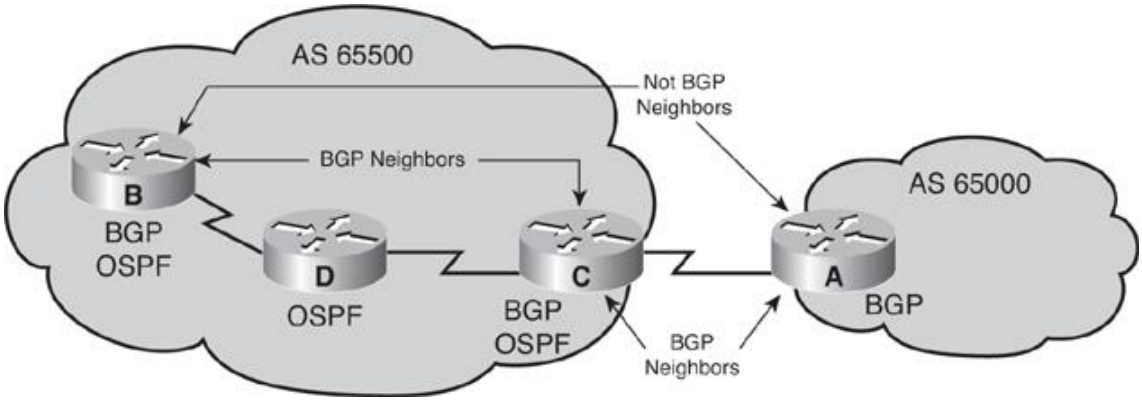


Figura 12. Distintos tipos de vecinos BGP

5.1.1 Vecinos externos BGP

Cuando BGP se está ejecutando entre routers en AS diferentes, se denomina eBGP (*external BGP)*. Los routers eBGP están conectados directamente entre sí, como se muestra en la siguiente figura.

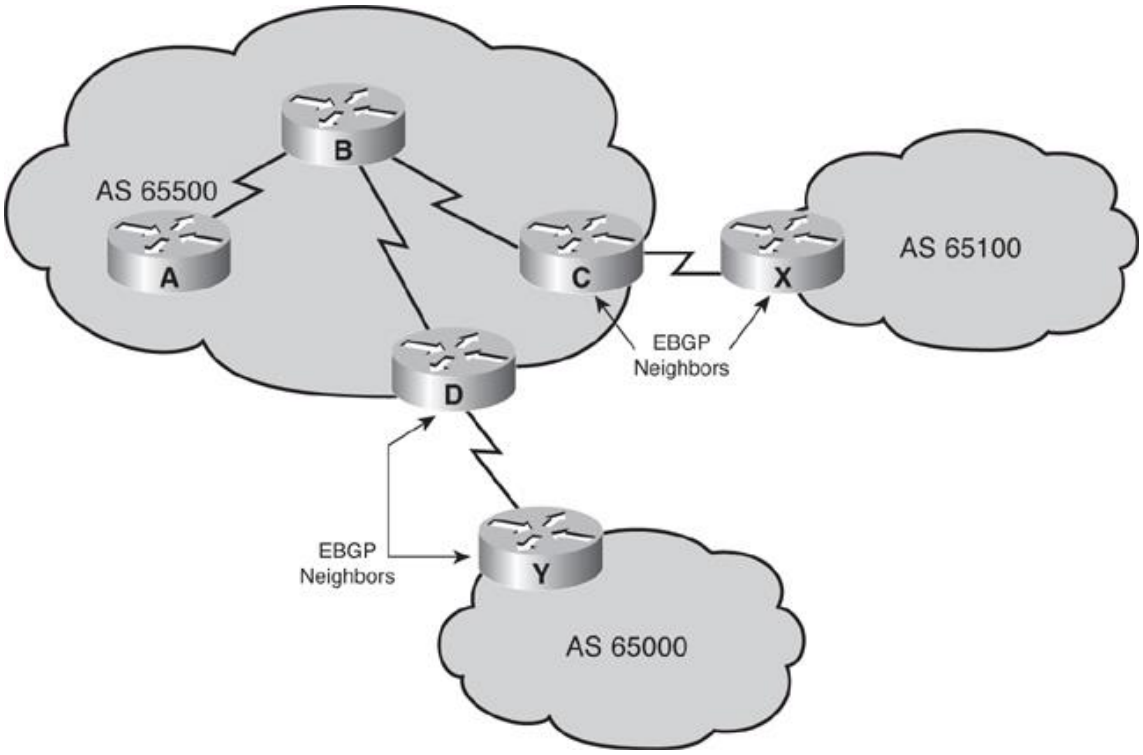


Figura 13. Vecinos eBGP

Un vecino eBGP es un router que se ejecuta en un AS diferente. Un IGP no se ejecuta entre eBGP vecinos. Para que dos routers intercambien actualizaciones de enrutamiento BGP, la capa de transporte TCP en cada lado debe pasar satisfactoriamente el handshaking TCP de tres vías antes de que se pueda establecer la sesión BGP. Por lo tanto, la dirección IP usada en el comando *neighbor* debe ser accesible sin usar un IGP. Esto puede lograrse señalando una dirección a la que se puede llegar a través de una conexión directa o configurando una ruta estática a esa dirección IP. Generalmente, la dirección vecina usada es la de la red conectada directamente.

Una red puede tener una conexión a uno o varios ISPs, y los ISPs en sí mismos pueden estar conectados a varios ISPs. Para cada una de estas conexiones entre AS diferentes, se requiere una sesión eBGP entre todos los routers eBGP vecinos. En la figura anterior, se establece una relación eBGP entre los routers D e Y, y otra relación eBGP entre los routers C y X. Los vecinos intercambiarán los mensajes de actualización de enrutamiento BGP con el resto. Por lo tanto, los routers del AS 65500 aprenden las rutas a los AS externos de sus respectivos vecinos eBGP.

Hay varios requerimientos que deben cumplirse para que se establezca correctamente una relación vecina eBGP:

- **Diferente número de AS**: Los vecinos eBGP deben residir en diferentes AS para ser capaces de formar una relación eBGP.
- **Definir vecinos**: Una sesión TCP se debe establecer antes de que empiece el intercambio de mensajes de actualización de enrutamiento BGP.
- **Alcanzabilidad**: Las direcciones IP usadas en el comando *neighbor* deben ser accesibles; los vecinos eBGP suelen estar directamente conectados.

5.1.2 Vecinos internos BGP

Cuando BGP se ejecuta entre routers dentro del mismo AS, se denomina iBGP (*internal BGP*). iBGP se ejecuta dentro de un AS para intercambiar información BGP de manera que todos los *speakers BGP* internos tienen la misma información de enrutamiento BGP sobre los AS exteriores y por lo tanto esta información puede ser transmitida a otros AS.

Hay varios requerimientos para poder establecer una relación de vecindad iBGP:

- **Mismo número de AS**: Los vecinos iBGP deben residir en el mismo AS para ser capaces de formar una relación iBGP.
- **Definir vecinos**: Una sesión TCP debe ser establecida entre vecinos antes de que empiecen a intercambiar mensajes de actualización de enrutamiento BGP.
- **Alcanzabilidad**: Los vecinos iBGP deben ser accesibles. Un IGP como puede ser RIP u OSPF se ejecuta dentro del AS, y proporciona esta accesibilidad.

Los routers que ejecutan iBGP no tienen por qué contar con una conexión directa entre ellos, basta con que se puedan alcanzar entre sí de modo que el handshaking TCP se pueda realizar para configurar las relaciones vecinas BGP. Se puede acceder a los vecinos iBGP mediante una conexión directa a la red, rutas estáticas, o un protocolo de enrutamiento interno. Generalmente, existen múltiples rutas dentro de un mismo AS para alcanzar otros routers, una dirección loopback suele ser usada en el comando *neighbor* BGP para establecer las sesiones iBGP.

Por ejemplo, en la siguiente figura, los routers A, D y C aprenden las rutas a los AS externos desde sus respectivos vecinos eBGP (routers Z, Y y X). Si el enlace entre los routers D e Y cae, el router D debe aprender nuevas rutas para los AS externos. Otros routers dentro del AS 65500 que estaban usando al router D para obtener rutas hacia redes externas deben también ser informados de que esas rutas a través del router D son inviables. Estos routers BGP dentro del AS 65500 necesitan tener rutas alternativas a través de los routers A y C en su tabla de enrutamiento BGP.

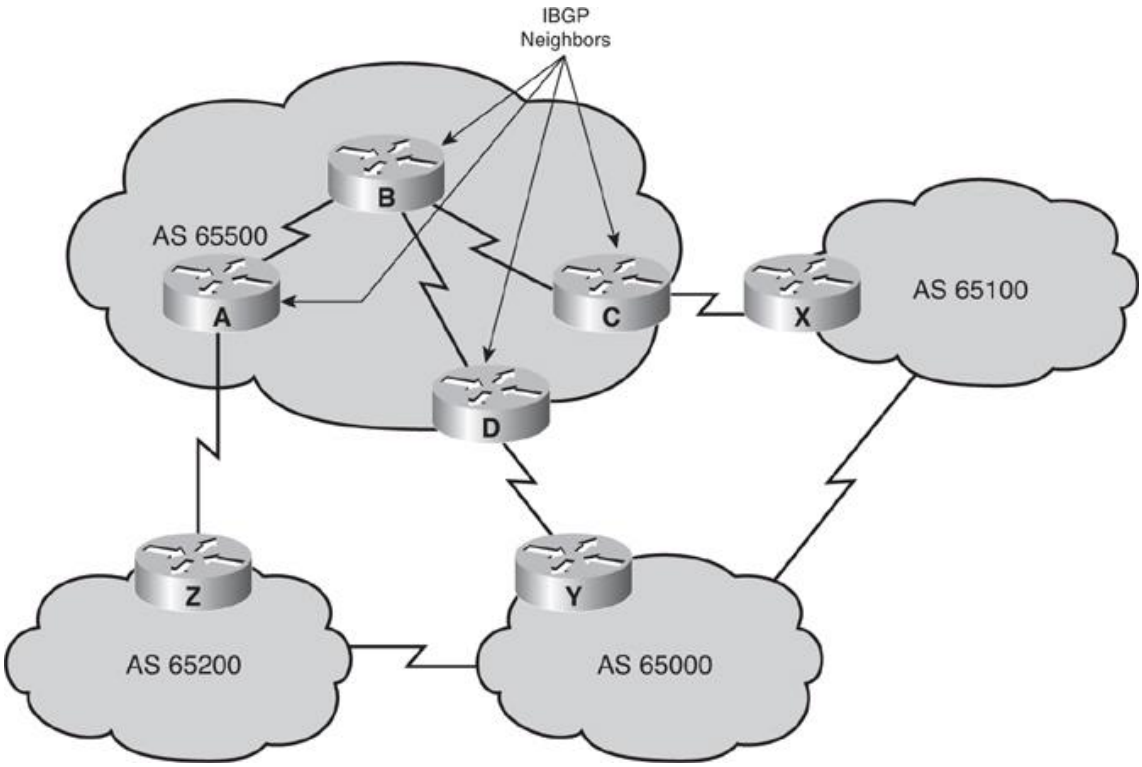


Figura 13. Vecinos iBGP

En el siguiente capítulo se explica la necesidad de configurar una malla completa de sesiones iBGP entre todos los routers en la trayectoria del AS 65500 de manera que cada router en el trayecto dentro del AS aprenda los caminos a las redes externas a través de iBGP.

5.1.2.1 iBGP en todos los routers de tránsito

En este capítulo expondremos los argumentos por los cuales se argumenta por qué la propagación de rutas iBGP requiere que todos los routers en el trayecto de un mismo AS se ejecuten en una malla completa iBGP.

5.1.2.1.1 iBGP en un AS de tránsito

BGP fue originalmente destinado a ejecutarse a lo largo de las fronteras de un AS, con los routers en el medio de un AS ignorando los detalles de BGP – por eso el nombre Border Gateway Protocol. Un ***transit AS***, como el AS 65102 de la siguiente imagen, es un AS que enruta el tráfico desde un AS externo a otro. Como hemos mencionado anteriormente, los ***transit AS*** normalmente son ISPs. Todos los routers en el AS de tránsito deben tener conocimiento completo de los routers externos. Teóricamente, un camino para lograr este objetivo es redistribuir los routers BGP dentro de un IGP en los edge routers; sin embargo, este enfoque tiene problemas.

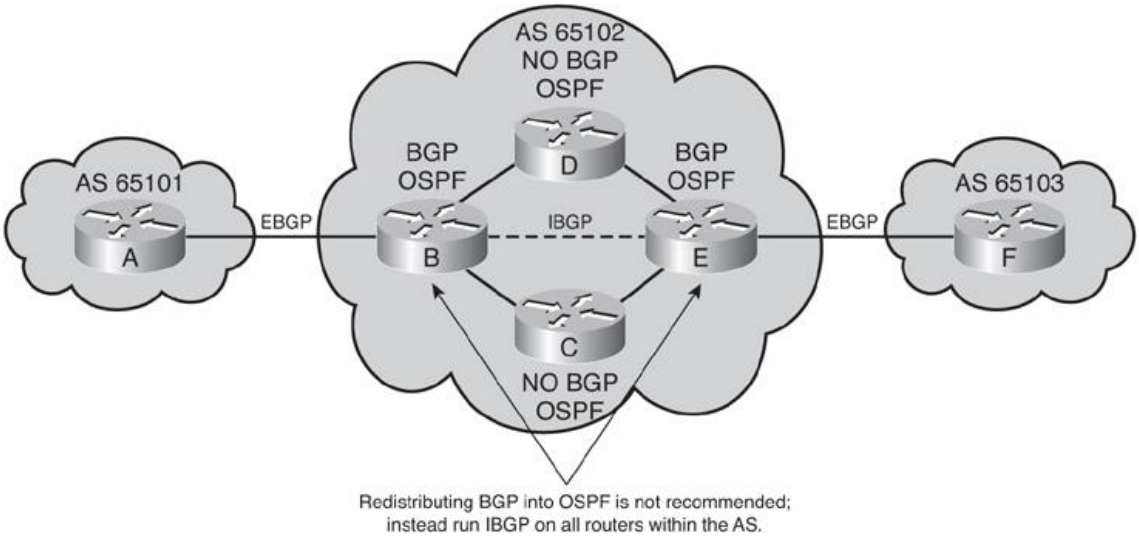


Figura 14. Transit AS

Estos problemas vienen, porque la tabla de enrutamiento del Internet actual es muy grande, la redistribución de todos los routers BGP dentro de un IGP no es la forma óptima para los routers interiores dentro de un AS para aprender el camino hacia las redes externas. Otro método que se puede usar es ejecutar iBGP en todos los routers dentro del AS.

5.1.2.1.2 iBGP en un AS non-transit

Un ***nontransit AS***, tal como una organización multihoming con dos ISPs, no pasa rutas entre los ISPs. Para tomar decisiones de enrutamiento apropiadas los routers BGP dentro del AS requieren conocimiento de todas las rutas BGP enviadas al AS.

BGP no trabaja de la misma manera que los IGPs. Esto es debido a que los diseñadores de BGP no podían no garantizar que el protocolo se ejecutaría en todos los routers de un mismo, por lo que hubo que definir un método para asegurar que todos los speakers iBGP pudieran enviar los mensajes de actualización de uno a otro asegurándose de que no existieran bucles de enrutamiento.

5.1.3 TCP y la estructura de malla completa

Como hemos mencionado anteriormente, TCP fue seleccionado como la capa transporte para BGP porque TCP puede mover un gran volumen de datos de manera fiable. Con la tabla de enrutamiento de Internet tan grande y cambiando constantemente, usar TCP con su método de ventana deslizante se decidió como la mejor solución. Las sesiones TCP no pueden ser multicast o broadcast porque TCP tiene que asegurar la entrega de paquetes a cada recipiente. Como TCP no puede usar el broadcasting o el multicasting, BGP tampoco puede usarlo.

Para evitar los bucles de enrutamiento dentro de un mismo AS, BGP especifica que las rutas aprendidas a través de iBGP nunca se propaguen a otros usuarios iBGP; esto algunas veces se conoce como la regla de split-horizon BGP. Así, cada router iBGP necesita enviar rutas a todos los vecinos iBGP en el mismo AS (para que todos tengan una imagen completa de las rutas enviadas al AS. Porque ellos no pueden usar broadcast o multicast, una relación vecina iBGP debe ser configurada entre cada par de routers). Por defecto, se supone que cada speaker BGP tiene una declaración de vecindad para el resto de speakers iBGP en el mismo AS; esto es lo que se conoce como iBGP full-mesh.

Si el vecino iBGP que envía no está completamente en la estructura de malla con cada router iBGP, los routers que no establezcan una relación de vecindad con este router tendrán diferentes tablas de enrutamiento IP que los routers que sí que la tengan. Las tablas de enrutamiento inconsistentes pueden causar bucles de enrutamiento o agujeros negros de enrutamiento, porque la asunción por defecto de que todos los routers que ejecutan BGP dentro del AS es que cada router BGP intercambie información iBGP directamente con todos los demás routers BGP en el AS.

Cuando todos los vecinos iBGP están completamente en la malla y se recibe un cambio desde un AS externo, el router eBGP del AS local es responsable de informar a todos los vecinos iBGP del cambio. Los vecinos iBGP que reciben esta actualización no la envían a ningún otro vecino iBGP porque ellos asumen que el vecino BGP que envía está completamente en la malla con todos los demás iBGP speakers ya ha enviado la actualización a cada vecino iBGP.

5.1.4 Ejemplos de estructuras BGP parcial y completamente malladas

La primera red que se muestra en la siguiente imagen ilustra el comportamiento de la actualización iBGP en un entorno vecino parcialmente mallado. El router B recibe una actualización eBGP del router A. El router B tiene dos vecinos iBGP, los routers C y D, pero no tiene una relación vecina iBGP con el router E. Por lo tanto, los routers C y D aprenden acerca de las redes que se agregaron o retiraron detrás del router B. Incluso si los routers C y D tienen sesiones vecinas iBGP con el router E, ellos asumen que el AS está completamente en la malla por el iBGP y que no reproduce exactamente las actualizaciones y las envía al router E. El iBGP que envía las actualizaciones al router E es responsabilidad del router B porque es el router con conocimiento de primera mano de las redes y más allá del AS 65101. Así que, el router E no aprende de ninguna red a través del router B y no lo usa para alcanzar ninguna red en el AS 65101 o en otro AS detrás del AS 65101.

Sin embargo, en la otra red el iBGP está completamente mallado. Cuando el router B recibe la actualización eBGP del router R1, ésta actualiza a los tres de sus nodos iBGP, el router C, el

router D y el router E. OSPF, el IGP, es usado para enrutar el segmento TCP que contiene la actualización BGP del router B al router E, porque estos dos routers no están directamente conectados. La actualización es enviada una vez a cada vecino y no es duplicada por otro vecino iBGP (que también reduce el tráfico innecesario). En el iBGP que está completamente en la malla, cada router asume que todos los demás routers internos tienen una declaración de vecino que señala a cada vecino iBGP.

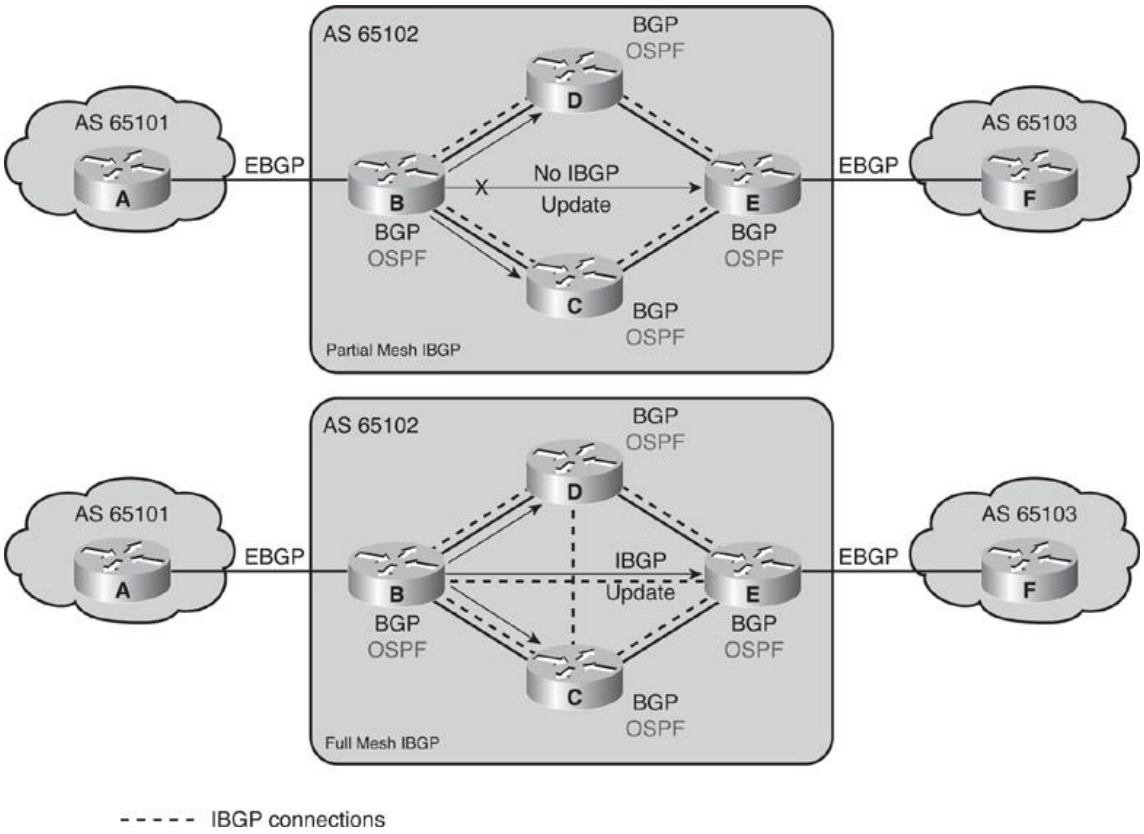


Figura 15 Estructura completamente mallada BGP

Cuando todos los routers BGP que se ejecutan en el AS están completamente engranados y tienen la misma tabla de enrutamiento como resultado de una política de enrutamiento coherente, ellos pueden aplicar la misma fórmula de selección de ruta. Los resultados de la selección de ruta serán por lo tanto uniformes a lo largo del AS. La trayectoria uniforme a través del AS significa que no hay bucles de enrutamiento y una política coherente para el AS existente y entrante.

5.2 Requerimientos para una configuración BGP básica

Antes de configurar el BGP, un administrador de red debe definir los requerimientos de la red, incluyendo la conectividad interna (por iBGP) y la externa al ISP (por eBGP). El siguiente paso es reunir los parámetros necesarios para proporcionar los detalles de la configuración BGP. Para una configuración BGP básica, estos detalles incluyen lo siguiente:

- Número de AS (en nuestra propia red y en todos los AS remotos).
- Las direcciones IP de todos los vecinos (nodo) involucrados.

- Las redes que se anunciarán dentro de BGP.

La configuración básica de BGP requiere los siguientes pasos:

Paso 1. Definir el proceso BGP.

Paso 2. Establecer las relaciones vecinas.

Paso 3. Anunciar las redes dentro de BGP.

5.3 Modo de configuración BGP

Para entrar al modo de configuración BGP de un router se utiliza el comando de configuración global **router bgp** *autonomous-system* para entrar en el modo de configuración e identificar el AS local al que pertenece ese router. En el comando, *autonomous-system* hay que identificar el AS local. El proceso BGP necesita ser informado de su AS de modo que cuando los vecinos BGP se configuren puedan determinar si son vecinos iBGP o eBGP.

El comando **router bgp** sólo, no activa el BGP en un router. Hay que introducir al menos un subcomando debajo del comando **router bgp** para activar el proceso BGP en el router.

Una sola instancia de BGP puede ser configurado en un router a la vez. Por ejemplo, si tu configuras tu router en un AS 65000 y luego intentas configurar el comando **router bgp 65100**, el router te informa de que está configurado actualmente por el AS 65000.

5.4 Definiendo vecinos BGP y activando las sesiones BGP

Hay que utilizar el comando de configuración del router **neighbor** *ip-address* **remote-as** *autonomous-system* para activar una sesión BGP para vecinos externos e internos e identificar un nodo con el cual el router local establecerá una sesión, como se ve en la siguiente tabla.

La dirección IP usada en el comando **neighbor remote-as** es la dirección de destino para todos los paquetes BGP que van a este router vecino. Para establecer una relación BGP, ésta dirección tiene que ser accesible, porque BGP intenta establecer una sesión TCP e intercambiar actualizaciones BGP con el dispositivo de ésta dirección IP.

El valor asignado en el campo *autonomous-system* del comando **neighbor remote-as** determina si la comunicación con el vecino es una sesión iBGP o eBGP. Si el campo *autonomous-system* configurado en el comando **router bgp** es idéntico al campo en el **neighbor remote-as**, BGP inicia una sesión interna, y la dirección IP especifica no tiene una conexión directa. Si en el campo los valores difieren, BGP inicia una sesión externa, y la dirección IP especifica tiene que tener una conexión directa.

La red mostrada en la siguiente imagen, usa los comandos vecinos BGP. El router R1 en el AS 65101 tiene dos estados vecinos. En el primer estado, **neighbor 10.2.2.2** (R2) esta en el mismo AS que el router R1 (65101); este estado vecino define R2 como un vecino iBGP. AS 65101

ejecuta EIGRP entre todos los routers internos. El router R1 tiene un camino EIGRP para alcanzar la dirección 10.2.2.2. Como un vecino iBGP, R2 puede estar a varios saltos del R1.

Parámetro	Descripción
ip-address	Identifica el router compañero
autonomous-system	Identifica el router compañero del AS

Tabla 1. Definición de parámetros para vecinos BGP

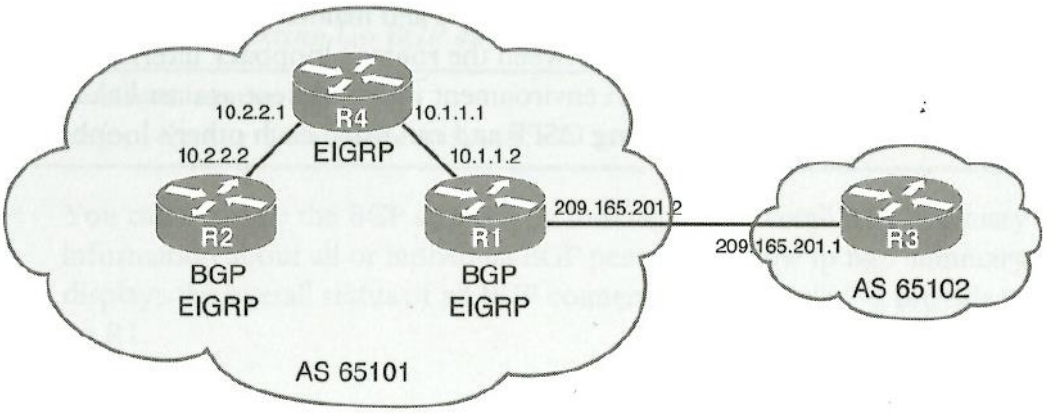


Figura 16. Establecimiento relaciones vecindad BGP

Configuration of Router R1

>>>router bgp 65101

>>neighbor 10.2.2.2 remote-as 65101

>>neighbor 209.165.201.1 remote-as 65102

Configuration of Router R2

>>>router bgp 65101

>>neighbor 10.1.1.2 remote-as 65101

Configuration of Router R3

>>>router bgp 65102

>>neighbor 209.165.201.2 remote-as 65101

El router R1 sabe que el router R3 es un vecino externo porque el **neighbor** statement para R3 usa AS 65102, el cual difiere del número del AS del R1, AS 65101. El router R1 puede alcanzar el AS 65102 vía 209.165.201.2, el cual está directamente conectado al R1.

5.5 Configuración y verificación BGP básico

En este capítulo analizaremos un ejemplo para configurar y verificar una red con el protocolo BGP básico. En la siguiente imagen vemos el diagrama de la red para este ejemplo. Las sesiones BGP internas y externas serán establecidas primero, y los prefijos de red serán anunciados vía BGP. Los comandos **show** serán usados para observar cómo se propaga BGP y como mantiene la información de enrutamiento. Las sesiones BGP serán entonces establecidas entre los bucles de enrutamiento de la interfaz de las direcciones IP; esto es una técnica para hacer el entorno más resistente contra los fallos de enlace, que puede haberlos por causas no controladas. Los routers R2 y R3 ya están ejecutando OSPF y pueden alcanzar cada dirección de loopback 0.

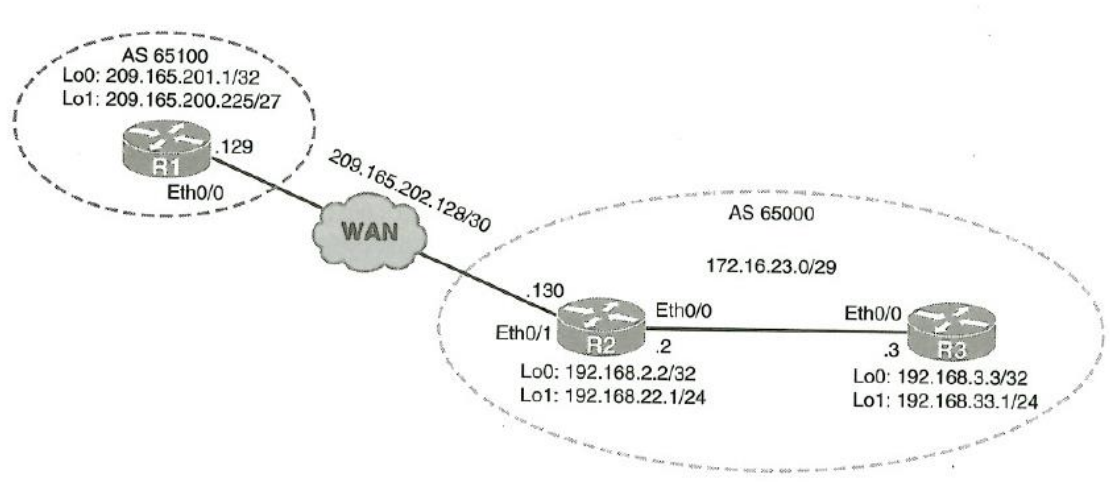


Figura 17. Ejemplo configuración BGP

5.5.1 Configuración y verificación sesión eBGP

Vamos a empezar con el router R1 y su configuración por BGP y con una sesión eBGP con R2. Debemos recordar que hay que entrar en la configuración global con el comando **router bgp** *autonomous-system* para identificar el router del AS local; para R1, este es el 65100. El comando de configuración del router **neighbor** *ip-address* **remote-as** *autonomous-system* identifica la dirección IP y el AS del vecino. El vecino de R1 es R2 en el AS 65000. Una relación eBGP tiene que abarcar un máximo de un salto por defecto, por tanto, las direcciones IP para la sesión eBGP tienen que estar conectadas directamente con un vecino.

Inicio configuración BGP y estableciendo sesión en R1

>>>R1(config) # router bgp 65100

>>>R1(config-router) # neighbor 209.165.202.130 remote-as 65000

Inicio configuración BGP y estableciendo sesión en R2

>>>R2(config) # router bgp 65000

>>>R2(config) # neighbor 209.165.202.129 remote-as 65100

Se pueden examinar las sesiones BGP buscando un resumen general de BGP o información detallada acerca de todos los nodos BGP individuales. El comando **show ip bgp summary** muestra el estado general de todas las conexiones BGP.

show ip bgp summary en R1

```
>>>R1# show ip bgp summary
    >>BGP router identifier 209.165.201.1, local AS number 65100
    >>BGP table versión is 1, main routing table version 1
    >>Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
209.165.202.130     4 65000      91      93        1    0    0 01:20:28        0
```

La primera parte de la salida de este comando describe el router local:

- **BGP router identifier**: La dirección IP de todos los demás speakers BGP con la que reconocen este router.
- **Local AS number**: El número de AS del router local.

La siguiente parte de la salida describe la tabla BGP:

- **BGP table version**: número de versión de la tabla local BGP; aumenta cuando la tabla BGP cambia.
- **Main routing table version**: última versión de la base de datos BGP que fue introducida en la tabla de enrutamiento principal.

El resto describe el estado actual del vecino, uno por cada vecino configurado:

- **Neighbor**: La dirección IP, usada en la declaración del vecino, con la que este router está estableciendo una relación.
- **Version (V)**: La version de BGP de este router que se está ejecutando con los vecinos listados.
- **AS**: El número del AS del vecino listado.
- **Messages received (MsgRcvd)**: El número de mensajes BGP recibidos desde este vecino.
- **Messages sent (MsgSent)**: El número de mensajes BGP enviados a este vecino.
- **TblVer**: La última versión de la tabla BGP que se envió a este vecino.
- **In queue (InQ):** El número de mensajes desde este vecino que están esperando a ser procesados.
- **Out queue (OutQ)**: El número de mensajes en cola y esperando a ser enviados a este vecino. El control de flujo TCP impide que este router abrume a un vecino con una gran actualización.
- **Up/down**: La duración de tiempo que este vecino ha estado en el estado BGP actual (establecido, activo, o libre).
- **State**: El estado actual de la sesión BGP: activo, libre, abierto enviado, abierto confirmado, o libre (admin). El estado admin indica que vecino esta administrativamente cerrado; este estado es creado por el comando de configuración de router **neighbor** *ip-address* **shutdown**. El estado activo significa que el router está intentando crear una conexión TCP con ese vecino. Hay que tener en cuenta que, si la sesión está establecida,

el estado no se muestra. En su lugar, se muestra un número que representa el PfxRcd, como se describe a continuación.

- **Prefix received (PfxRcd)**: Cuando la sesión está en el estado establecida, este valor representa el número de entradas de red BGP recibidas desde el vecino.

En la tabla se puede observar que hay un cero en la columna de PfxRcd; esto indica que el estado es establecido pero los prefijos de la red aún no han sido recibidos. Se puedes usar la información del comando **show ip bgp summary** para verificar que las sesiones BGP están establecidas. Si no lo están, se puede investigar la configuración BGP para localizar el problema. También se puede verificar la dirección IP y el número del AS de los vecinos BGP configurados con este comando. Si la sesión está establecida, el número de mensajes que hayan sido enviados y recibidos, como se muestra en la salida de este comando, indican la estabilidad de la sesión BGP.

El comando **show ip bgp neighbors** proporciona información adicional, como la capacidad de negociación, la familia de direcciones soportadas, y otros.

show ip bgp neighbors en R1

>>>R1# show ip bgp neighbors

BGP neighbor is 209.165.202.130, remote AS 65000, external link

>>BGP version 4, remote router ID 192.168.22.1

>>BGP state = Established, up for 01:21:17

>>Last read 00:00:25, last write 00:00:00, hold time is 180, keepalive interval is 60 seconds

>>Neighbor sessions:

>1 active, is not multisession capable (disabled)

>>Neighbor capabilities:

>Route refresh: advertised and received (new)

>Four-octets ASN Capability: advertised and received

>Address family IPv4 Unicast: advertised and received

>Enhanced Refresh Capability: advertised and received

>Multisession Capability:

>Stateful switchover support enabled: NO for session 1

>>Message statistics:

>InQ depth is 0

>OutQ depth is 0

	Sent	*Rcvd*
>>Opens:	*1*	*1*
>>Notifications:	*0*	*0*
>>Updates:	*1*	*1*
>>Keepalives:	*92*	*90*
>>Route Refresh:	*0*	*0*
>>Total:	*94*	*92*

>>Default mínimum time between adverisement runs in 30 seconds

>>For address family: IPv4 Unicast

>Session: 209.165.202.130

El comando **show ip bgp neighbors** es útil para obtener información acerca de las sesiones TCP y de los parámetros BGP de las sesiones incluyendo temporizadores y contadores TCP. Se puede también examinar los detalles de una sesión específica si se añade la dirección IP del vecino al comando. Este comando también tiene parámetros opcionales que pueden ser incluidos para un vecino específico, como se muestra en la siguiente configuración. Se pueden usar estos parámetros para examinar información de enrutamiento BGP específica que fue enviada o recibida desde el vecino, la cual puede ser útil cuando estamos realizando troubleshooting acerca del enrutamiento.

show ip bgp neighbors Opciones del Comando

>>>R1# show ip bgp neighbors 209.165.202.130 ?

>>advertised-routes Display the routes advertised to a BGP neighbor

>>dampened-routes Display the dampened routes received from neighbor (eBGP nodos only)

>>flap-statistics Display flap statistics of the routes learned from neighbor (eBGP nodos only)

>>paths Display AS paths learned from neighbor

>>policy Display neighbor pólices per address-family

>>received Display information received from a BGP neighbor

>>received-routes Display the received routes from neighbor

>>routes Display routes learned from neighbor

>>| Output modifiers

show ip bgp summary and show ip bgp neighbors en R2

>>>R2# show ip bgp summary

>>BGP router identifier 192.168.22.1, local AS number 65000

>>BGP table version is 1, main routing table version 1

>>Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
209.165.202.129	4	65100	116	114	1	0	0	01:41:20	0

>>>R2# show ip bgp neghbors

>>BGP neighbor is 209.165.202.129, remote AS 65100, external link

>BGP version 4, remote router ID 209.165.201.1

>BGP state = Established, up for 01:41:32

>Last read 00:00:41, last write 00:00:47, hold time is 180, keepalive interval is 60 seconds

>>Neighbor sessions:

>1 active, is not multisession capable (disabled)

>>Neighbor capabilities:

>Route refresh: advertised and received (new)

>>>R2# show ip bgp neighbors

>>BGP neighbor is 209.165.202.129, remote AS 65100, external link

>BGP version 4, remote router ID 209.165.201.1

>BGP state = Established, up fpr 01:41:32

>Last read 00:00:41, last write 00:00:47, hold time is 180, keepalive interval is 60 seconds

>Neighbor sessions:

>1 active, is not multisession capable (disabled)

>Neighbor capabilities:

>Route refresh: advertised and received (new)

5.5.2 Configuración y verificación sesión iBGP

Ahora que las sesiones eBGP están establecidas, a continuación, vamos a configurar la sesión iBGP entre R2 y R3, usando las direcciones en la conexión entre los routers. Una sesión iBGP se configura usando el comando **neighbor** *ip-address* **remote-as** *autonomous-system*, de la misma forma que se establecen las sesiones externas. Hay que recordar que el router identifica automáticamente una sesión interna examinando el de número AS y comparándolo con el número de AS local. Para una sesión iBGP, las direcciones IP del vecino no tienen que estar conectadas directamente (a pesar de que en este ejemplo lo están).

Estableciendo relaciones entre el R2 y R3

>>>R2 (config)# router bgp 65000

>>>R2 (config-router)# neighbor 172.16.23.3 remote-as 65000

>>>R3 (config)# router bgp 65000

>>>R3 (config-router)# neighbor 172.16.23.2 remote-as 65000

Se pueden verificar las sesiones iBGP de la misma manera que se monitorizan las sesiones BGP externas. En los siguientes cuadros veremos la salida del comando en los routers R2 y R3. Se observa que la conexión iBGP se identifica como un *internal link* en la salida del comando **show ip bgp neighbors**. De nuevo, la salida del comando en el router R3 es un reflejo de la información de la conexión iBGP en R2.

Examinando sesiones BGP en R2

>>>R2# show ip bgp summary

>>>BGP router identifier 192.168.22.1, local AS number 65000

>>>BGP table version is 1, main routing table version 1

>>> Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd

172.16.23.3 4 65000 13 13 1 0 0 00:08:23 0 209.165.202.129 4 65100 287 284 1 0 0 04:16:06 0

>>>R2# show ip bgp neighbors

>>>BGP neighbor is 172.16.23.3, remore AS 65000, internal link

>>BGP version 4, remote router ID 192.168.33.1

>>BGP state = Established, up for 00:08:38

Examinando sesiones BGP en R3

>>>R3# show ip bgp summary

>>>BGP router identifier 192.168.33.1, local AS number 65000

>>>BGP table version is 1, main routing table version 1

>>> Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd

172.16.23.1 4 65000 109 110 1 0 0 01:36:17 0

>>>R3# show ip bgp neighbors

>>BGP neighbor is 172.16.23.2, remote AS 65100, internal link

>BGP version 4, remote router ID 192.168.22.1

>BGP state = Established, up fpr 01:37:20

>Last read 00:00:06, last write 00:00:03, hold time is 180, keepalive interval is 60 seconds

>Neighbor sessions:

>1 active, is not multisession capable (disabled)

>Neighbor capabilities:

>Route refresh: advertised and received (new)

5.6 Propagación de redes en BGP

Ahora que las sesiones están establecidas, hay que configurar los routers para anunciar redes. Se utiliza el comando de configuración **network** *network-number* [**mask** network-mask] para introducir rutas que están en la tabla de enrutamiento IPv4 dentro de la tabla BGP de manera que ellas puedan ser anunciadas por BGP. La Tabla 7-3 describe este comando.

Parameter	Description
network-number	Identifica una red IPv4 para ser anunciada por BGP
mask network-mask	(Opcional) Identifica la máscara de la subred para ser anunciada por BGP. Si la máscara de red no está especificada, la máscara por defecto es la máscara clásica

Tabla 2. Definición de parámetros para configuración

Es importante tener en cuenta que el comando BGP **network** determina que redes anuncia este router. Este es un concepto diferente del que se usa para configurar IGPs. A diferencia de los IGPs, el comando **network** no empieza el protocolo BGP en una interfaz específica. Esto indica a BGP que redes deberían de originarse desde este router. La lista de los comandos **network** debe incluir todas las redes del AS que se deseen anunciar, no solo aquellas que estén conectadoas localmente a nuestro router. El parámetro **mask** indica que BGP-4 permite clases sin prefijos; esto nos permite anunciar subredes y superredes.

Hay que tener en cuenta la diferencia entre el comando **neighbor** y el comando **network**: el comando **neighbor** dice a BGP donde anunciar; el comando **network** dice a BGP que anunciar. El único propósito del comando **network** es notificar a BGP que redes hay para anunciar. Si el parámetro **mask** no se especifica, este comando anuncia solo el número de red con clase; al menos

una subred de la red principal especificada debe estar presente en la tabla de enrutamiento IP para permitir que BGP empiece anunciando la red de clase como una ruta BGP.

Sin embargo, si especificamos **mask** *network-mask,* una coincidencia exacta con la red (tanto dirección como máscara) debe existir en la tabla de enrutamiento para que la red sea anunciada. Antes de que BGP anuncie una ruta, comprueba si puede alcanzarlo. Por ejemplo, si tenemos que anunciar la ruta 192.168.0.0/24, y por error configuramos la red 192.168.0.0 con máscara 255.255.0.0 en vez de la red 192.168.0.0 con máscara 255.255.255.0, BGP busca 192.168.0.0/16 en la tabla de enrutamiento. En este caso, podría encontrar 192.168.0.0/24 pero no encontrará 192.168.0.0/16. Esto es debido a que la tabla de enrutamiento no contiene una coincidencia específica con la red, BGP no anuncia la red 192.168.0.0/24 a ningún vecino.

Si tenemos que anunciar el bloque CIDR 192.168.0.0/16, hay que configurar la red 192.168.0.0 con máscara 255.255.0.0. De nuevo, BGP busca 192.168.0.0/16 en la tabla de enrutamiento, y si nunca encuentra 192.168.0.0/16, BGP no anuncia la red 192.168.0.0/16 a ningún vecino. En este caso, puedes configurar una ruta estática del bloque CIDR hacia la interfaz null, con el comando **ip route 192.168.0.0 255.255.0.0 null0**, de forma que BGP puede encontrar una coincidencia exacta en la tabla de enrutamiento. Después de encontrar una coincidencia exacta en la tabla de enrutamiento, BGP anuncia la red 192.168.0.0/16 a los vecinos.

En R3, vamos a anunciar el prefijo de red que está configurado en la interfaz loopback 1 (192.168.33.0/24) en BGP. En R3, para examinar la tabla BGP, usamos el comando **show ip bgp**, para ver el prefijo anunciado.

Propagando la red del interfaz loopback 1 en R3

```
>>>R3(config)# router bgp 65000
>>>R3(config-router)# network 192.168.33.0 mask 255.255.255.0
```

Examinar tabla BGP en R3

```
>>>R3# show ip bgp
>>>BGP table version is 2, local router ID is 192.168.33.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,
                r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
                x best-external, a additional-path, c RIB-compresses,
>>>Origin codes: i – IGP, e – EGP, ? – incomplete
>>>RPKI validation codes: V valid, I invalid, N Not found

>>>Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.33.0     0.0.0.0                  0         32768  i
```

Cuando se usa el comando **show ip bgp** sin los parámetros opcionales, se muestra toda la tabla BGP. Se muestra también una lista de información abreviada sobre cada ruta, un prefijo por línea. La salida se clasifica en orden de número de red; si la tabla BGP contiene más de una entrada por la misma red, las rutas alternativas se muestran en líneas sucesivas. El número de red se muestra sólo en la primera de éstas líneas.

Los *status codes* se muestran al principio de cada línea de la salida, y los *origin codes* son mostrados al final de cada línea. Una fila con un asterisco (*) en la primera columna significa que toda la tabla es válida. Algunas de las otras opciones para la primera columna son las siguientes:

- Una ***s,*** indica que las rutas específicas están suprimidas
- Una ***d***, indica que la ruta está siendo amortiguada (penalizada) por caerse con demasiada frecuencia. A pesar de que la ruta podría levantarse ahora mismo, ésta no es anunciada hasta que la penalización haya expirado.
- Una ***h***, indica que la ruta es inviable y probablemente está caida. Existe información histórica sobre las rutas, pero la mejor ruta no existe.
- Una ***r***, por fallo de la Tabla de Información de Enrutamiento (RIB), indica que la ruta no fue instalada en el RIB; el RIB es otro nombre para la tabla de enrutamiento IP. La razón de que la ruta no esté instalada puede ser mostrada usando el comando **show ip bgp rib-failure**
- Una ***S***, indica que ruta está viciada. (Esto es usado en un router directo sin envío).

Un signo mayor que (>) en la segunda columna indica el mejor camino para la ruta seleccionada por BGP. Esta ruta es ofrecida por la tabla de enrutamiento IP. La tercera columna está en blanco o tiene una *i* en ella. Si está en blanco, BGP aprendió que esa ruta es de un compañero externo. Si tiene una *i*, un vecino iBGP anunció ésta ruta a este router. La cuarta columna lista las redes que el router aprendió.

Algunos, pero no todos, de los atributos BGP que están asociados con la ruta son mostrados. La quinta columna lista todas las direcciones next-hop de cada ruta. Si la columna contiene 0.0.0.0, este router originó la ruta. (Para BGP la dirección next-hop no siempre está en un router que está directamente conectado con éste router).

Las siguientes tres columnas listan tres atributos BGP asociados con la ruta: metric, que también se denomina discriminador de salida múltiple (MED); local preference y weight.

La última columna significa que esta ruta fue introducida en el router original BGP (el atributo original). Si la última columna tiene una *i* en ella, el router original probablemente usa un comando **network** para introducir esta red en BGP. La letra ***e*** significa que el router original aprendió ésta red de EGP, el cual es el predecesor histórico de BGP. Un signo de interrogación (?) significa que el proceso BGP original no puede verificar absolutamente la viabilidad de ésta red porque está redistribuido desde un IGP dentro del proceso BGP.

En R2, se puede examinar la tabla BGP y la parte BGP de la tabla de enrutamiento; tal y como se ve en el siguiente cuadro.

Examinando la tabla BGP y de routing en R2

```
>>>R2# show ip bgp
>>>BGP table version is 4, local router ID is 192.168.22.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,
     >> Network                Next Hop            Metric LocPrf Weight Path
  *i>192.168.33.0              172.16.23.3              0    100      0  i

>>>R2# show ip route bgp
>>>B        192.168.33.0/24   [200/0]  via 172.16.23.3,  01:20:57
```

La tabla BGP de R2 indica que el prefijo 192.168.33.0/24 es una ruta interna (tiene la *i* en la tercera columna) y su atributo next-hop es la dirección IP del vecino que origina la ruta. El atributo next-hop es también visible en la tabla de enrutamiento. El next-hop identifica el camino hacia la red de destino. Sin un AS, el next-hop no cambia; apunta al router que anuncia la ruta.

Examinando la tabla BGP y de routing en R1

```
>>>R1# show ip bgp
>>>BGP table version is 4, local router ID is 209.165.201.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,
     >> Network                Next Hop            Metric LocPrf Weight Path
*>    192.168.33.0             209.165.202.130                          0 6500 i

>>>R1# show ip route bgp
>>>B          192.168.33.0/24     [20/0]    via 209.165.202.130,  01:16:42
```

La tabla BGP de R1 muestra el prefijo 192.168.33.0/24; sin embargo, esta vez no está marcado como una ruta interna. Por lo tanto, ésta es una ruta externa. Hay darse cuenta de que el atributo *next-hop* es la dirección IP del vecino en el AS adyacente. El atributo *next-hop* informa al router dónde enviar el tráfico hacia la red dada. BGP, como los IGPs, es un protocolo de enrutamiento hop-by-hop. Sin embargo, a diferencia de los IGPs, las rutas BGP son AS por AS, no router por router, y el next-hop por defecto es el siguiente AS. La dirección *next-hop* para la red desde otro AS es una dirección IP del punto de entrada del siguiente AS a lo largo de la ruta de la red de destino. Por lo tanto, para eBGP, la dirección *next-hop* es la dirección IP del vecino que envía la actualización.

Para R1, hay que tener en cuenta que el atributo *AS-path* lista el AS 65000 como el único AS en la ruta del destino anunciado. Ahora vamos a configurar R2 para anunciar en BGP el prefijo de la red que está configurada en su interfaz loopback 1 (192.168.22.0/24) y verificar que se propaga a R1.

Propagando la red del interfaz loopback 1 en R2

```
>>>R2(config)# router bgp 65000
>>>R2(config-router)# network 192.168.22.0 mask 255.255.255.0
```

Confirmando que R1 llega a la interfaz de loopback R2

```
>>>R1# show ip bgp
>>>BGP table version is 5, local router ID is 209.165.201.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,

>> Network                Next Hop            Metric LocPrf Weight Path
*>192.168.22.0           209.165.202.130          0             0 6500 i
*>192.168.33.0           209.165.202.130                        0 6500 i

>>>R1# show ip route bgp
>>>B    192.168.22.0/24  [20/0]  via 209.165.202.130,   00:01:15
>>>B    192.168.33.0/24  [20/0]  via 209.165.202.130,   13:48:43
```

Hay que tener cuenta que el atributo *next-hop* en R1 es la dirección de R2 y la distancia administrativa de las rutas eBGP aprendidas a 20. Porque éste valor es menor que el valor de todos los IGPs, una ruta eBGP es preferida por defecto. Por lo tanto, los routers dirigen el tráfico a dominios externos en lugar de entregarlo localmente sin un dominio IGP; este comportamiento ayuda a evitar bucles. Ahora vamos a verificar que R3 puede ver la red loopback de R2.

Confirmando que R3 llega a la interfaz de loopback R1

```
>>>R3# show ip bgp
>>>BGP table version is 3, local router ID is 192.168.33.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,
     >> Network          Next Hop            Metric LocPrf Weight Path
   *>i 192.168.22.0      172.16.23.2              0    100      0 i
   *>  192.168.33.0      0.0.0.0                  0             32768 i

>>>R3# show ip route bgp
>>>B    192.168.22.0/24 [200/0] via 172.16.23.2,  05:56:53
```

El atributo *next-hop* en R3 es la dirección de R2, y la ruta está instalada en la tabla de enrutamiento de R3. Porque la distancia administrativa por defecto de las rutas iBGP es 200, la cual es más alta que el valor de todos los IGPs, si un router recibe anuncios sobre el mismo prefijo de red via iBGP y un IGP, la ruta IGP será preferida. Los routers, por lo tanto, dirigirán el tráfico a través de un dominio interno conforme la información de IGP en lugar de via iBGP. Este comportamiento ayuda a prevenir *black-holing* que puede producirse en los routers que no ejecutan BGP sin un dominio local. Ahora vamos a anunciar el prefijo de interfaz de loopback de R1 (209.165.200.224/27) y a verificar que se propaga a R2 y a R3.

Propagando la red del interfaz loopback 1 en R1

```
>>>R1(config)# router bgp 65100
>>>R1(config-router)# network 209.165.200.224 mask 255.255.255.224
```

Confirmando que R3 y R2 llega a la interfaz de loopback R1

```
>>>R2# show ip bgp
>>>BGP table version is 6, local router ID is 192.168.22.1
>>>Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,

     >> Network          Next Hop            Metric LocPrf Weight Path
*>   192.168.22.0        0.0.0.0                  0             32768 i
*>i  192.168.33.0        172.16.23.3              0    100      0 i
*>   209.165.200.224/27
                         209.165.202.129          0             0 65100 i

>>>R2# show ip route bgp
```

```
>>>B    192.168.33.0/24    [200/0]  via 172.16.23.3,  20:15:50
         209.165.200.0/7 is subnetted, 1 subnets
>>>B    209.165.200.224   [20/0] via 209.165.202.129,  00:00:56

>>>R3# show ip bgp
>>>BGP table version is 3, local router ID is 192.168.33.1
>>>Status codes: s suppressed, d damped, h history, *valid, > best, i – internal,
                    r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
                    x best-external, a additional-path, c RIB-compressed,
>>>Origin codes:  i – IGP, e – EGP, ? – incomplete
>>>RPKI validation codes: V valid, I invalid, N Not found

     >> Network                Next Hop            Metric LocPrf Weight Path
*>i    192.168.22.0             172.16.23.2              0    100      0  i
*>     192.168.33.0             0.0.0.0                  0         32768  i
* i    209.165.200.224/27
                                209.165.202.129          0    100      0  65100 i

>>>R3# show ip bgp
>>>B      192.168.22.0/24   [200/0] via 172.16.23.2,  05:56:53

>>>R3# show ip route 209.165.202.129
>>>% Network not in table
```

R2 recibe la información de 209.165.200.224/27 via BGP, con un atributo *next-hop* de R1; la ruta está instalada en la tabla de enrutamiento. R3 recibe el prefijo externo 209.165.200.224/24 en BGP y lo almacena en su tabla BGP. Hay que tener en cuenta que la entrada no está designada como la mejor ruta; el carácter *">"* no está. Esto es porque R3 no tiene una ruta para la dirección next-hop (209.165.202.129), y por lo tanto la ruta no está instalada en la ruta de enrutamiento.

5.7 Next-Hop-Self

El cómo BGP establece una relación iBGP, difiere significativamente de la manera en la que se comportan IGPs. Un protocolo interno, como RIP, EIGRP, o OSPF, siempre usa la dirección IP de origen de una actualización de enrutamiento como una dirección *next-hop* para cada red desde que la actualización se coloca en la tabla de enrutamiento. Hay que recordar que las rutas BGP son AS por AS, no router por router, y el *next-hop* por defecto es el siguiente AS. Como tal, para BGP el *next-hop* es la dirección IP que es usada para alcanzar el siguiente AS.

Como resultado, para eBGP, la dirección *next-hop* es la dirección IP del vecino que envía la actualización. Para iBGP, sin embargo, el *next-hop* anunciado por eBGP es almacenado dentro de iBGP, por defecto.

A veces es necesario anular el comportamiento predeterminado de un router y forzarlo a anunciarse como una dirección next-hop para las rutas enviadas al vecino. El comando de configuración del router **neighbor** *ip-address* **next-hop-self** te permite forzar BGP para usar la dirección IP de origen de la actualización como el *next-hop* para cada red que anuncia al vecino, en lugar de dejar que el protocolo elija la dirección *next-hop* para usar.

En nuestra red de ejemplo, vamos a configurar R2 para establecer la dirección *next-hop* como sí misma cuando los prefijos de publicidad de R3 y verificar la tabla de enrutamiento de R3.

Configurar R2 para que se propague a el mismo como siguiente salto

```
>>>R2(config)# router bgp 65000
>>>R2(config-router)# neighbor 172.16.23.3 next-hop-self
```

Confirmando que R3 alcanza el interfaz de loopback de R1

```
>>>R3# show ip bgp
     >> Network                Next Hop            Metric LocPrf Weight Path
*>i  192.168.22.0              172.16.23.2              0    100      0 i
*>   192.168.33.0              0.0.0.0                  0             32768 i
*>i  209.165.200.224/27
                               172.16.23.2              0    100      0 65100 i

>>>R3# show ip route bgp

>>>B      192.168.22.0/24 [200/0] via 172.16.23.2, 06:57:03
         209.165.200.0/27 is subnetted, 1 subnets
>>>B      209.165.200.224 [200/0] via 172.16.23.2, 00:02:51
```

Ahora R3 tiene una ruta a la subred loopback de R1 via la dirección next-hop de R2 (172.16.23.2). Ésta dirección está directamente conectada a R3, por lo que es accesible.

5.8 Troubleshooting estados de vecinos BGP

Después de que el handshake TCP esté completo, el protocolo BGP intenta establecer una sesión con el vecino. BGP se comporta como una máquina de estado que lleva un router a través de los siguientes estados con sus vecinos:

- **Idle**: El router está buscando la ruta de enrutamiento para ver si una ruta existe para alcanzar al vecino.

- **Connect**: El router encuentra una ruta para el vecino y ha completado el proceso de tres vías handshake TCP.
- **Open sent**: Un mensaje abierto se envió con los parámetros de la sesión BGP.
- **Open confirm**: El router recibió un acuerdo sobre los parámetros para establecer una sesión. Alternativamente, el router va al estado activo si no hay respuesta del mensaje abierto.
- **Established**: La relación de vecindad se establece y comienza el enrutamiento.

Después de que se introduzca el comando **neighbor** *remote-as*, BGP empieza en el estado idle, y el proceso BGP comprueba que tiene una ruta a la dirección IP listada. BGP debería estar en el estado ***idle*** únicamente unos pocos segundos. Sin embargo, si BGP no encuentra una ruta a la dirección IP vecina, se queda en el estado idle. Si encuentra una ruta, va al estado ***connect*** cuando el handshaking TCP synchronize acknowledge (SYN ACK) devuelve paquetes (cuando el proceso de tres vías TCP handshake está completo). Después de que la conexión TCP está establecida, el proceso BGP crea un mensaje abierto BGP y lo envía al vecino. Después de que TCP envíe este mensaje abierto, la sesión BGP de vecindad cambia el estado a *open sent.* Si no hay respuesta durante 5 segundos, el estado cambia a activo. Si una respuesta se vuelve a tiempo, BGP va al estado de *open confirm* y empieza a escanear (evaluar) la tabla de enrutamiento para las rutas enviadas al vecino. Cuando éstas rutas se han encontrado, BGP entonces va al estado *established* y empieza el enrutamiento entre los vecinos. El estado BGP es mostrado en la última columna en la salida del **comando show ip bgp summary**.

5.8.1 Idle State Troubleshooting

El estado **idle** indica que el router no sabe cómo alcanzar la dirección IP listada en la declaración del vecino. La razón más común para el estado idle es que el vecino no está anunciando la dirección IP o la red. Hay que comprobar las dos siguientes condiciones para solucionar éste problema:

- Asegurar que el vecino anuncia la ruta en su protocolo de enrutamiento local (IGP) (para vecinos iBGP).
- Verificar que no has introducido una dirección IP incorrecta en la declaración del vecino.

5.8.2 Active State Troubleshooting

Si el router está en el estado activo, significa que ha encontrado la dirección IP en la declaración del vecino y ha creado y enviado un paquete BGP, pero no ha recibido una respuesta (un paquete open confirm) desde el vecino.

Una causa común de esto es cuando el vecino no tiene una ruta de retorno a la dirección IP de origen. Por lo tanto, hay que asegurar que la dirección IP de origen o la red de los paquetes es anunciada dentro del protocolo de enrutamiento local (IGP) en el router vecino.

Otro problema común asociado con el estado activo es cuando un router BGP intenta establecer una relación de vecindad con otro router BGP que no tiene un vecino apuntando de nuevo al primer router, o el otro router está apuntando a la dirección IP érronea del primer router. Si el estado cambia entre el *idle* y el activo, los números de los distintos AS pueden estar mal configurados. Se vería un mensaje similar al siguiente con el número de AS incorrecto configurado en la declaración del vecino:

%BGP-3-NOTIFICATION: sent to neighbor 172.31.1.3 2/2 (peer in wrong AS) 2 bytes FDE6

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D 0104 FDE6 00B4 AC1F 0203 1002 0601 0400

0100 0102 0280 0002 0202 00

En el router remoto, sería un mensaje similar al siguiente:

%BGP-3-NOTIFICATION: received from neighbor 172.31.1.1 2/2 (peer in wrong AS) 2 bytes FDE6

5.9 BGP Session Resilience

En la red de ejemplo que estamos configurando, R2 y R3 tienen sólo una conexión entre ellos. Si la interfaz a la que pertenece la dirección IP utilizada en el comando **neighbor** se cayera (perdiera la conexión), la relación de vecindad BGP se perdería.

En casos donde existen múltiples rutas para alcanzar unos router vecinos iBGP, los routers podrían parearse también con la dirección del interfaz loopback y la sesión BGP no se perdería porque las interfaces loopback siempre están disponible, siempre y cuando el propio router no falle. Esta relación de vecindad añade un componente de resistencia a las sesiones BGP porque no estarán vinculados solo a una interfaz física, que podría caerse por cualquier razón.

Para establecer una relación de vecindad con el interfaz loopback de un vecino iBGP, hay que configurar cada router con el comando **neighbor** usando la dirección loopback de los routers vecinos. Ambos routers deben tener una ruta hacia la dirección loopback de otro vecino en su tabla de enrutamiento; hay que comprobar que ambos routers estén anunciado sus direcciones loopback en el IGP. En nuestra red de ejemplo, las rutas están ejecutando OSPF y tienen una ruta para las demás direcciones loopback 0.

En R2 y R3, podemos cambiar las direcciones de los pares iBGP a las respectivas direcciones de los interfaces loopback 0 (192.168.2.2 y 192.168.3.3); cómo podemos ver en los siguientes recuadros. Como éste ejemplo muestra, podemos cambiar la dirección del vecino borrando la dirección IP previa de éste vecino y realizando de nueva la configuración usando la dirección del interfaz loopback del vecino. Cada router enviará ahora paquetes BGP a la dirección loopback 0 de los otros routers.

Configurando R2 y R3 para establecer la relación de vecindad a través de los interfaces loopback

```
>>>R2(config)# router bgp 65000
>>>R2(config-router)# no neighbor 172.16.23.3
>>>R2(config-router)# neighbor 192.168.3.3 remote-as 65000
>>>R2(config-router)# neighbor 192.168.3.3 next-hop-self

>>>R3(config)# router bgp 65000
>>>R3(config-router)# no neighbor 172.16.23.2
>>>R3(config-router)# neighbor 192.168.2.2 remote-as 65000
```

Verificando que R2 y R3 han establecido la relación de vecindad

```
>>>R2# show ip bgp summary
>>>Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.3.3      4 65000       0       0        1    0    0 00:14:59        Idle
209.165.202.129  4 65100    2980    2981        9    0    0 1d21h           1

>>>R3# show ip bgp summary

>>>Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.2      4 65000       0       0        1    0    0 never         Idle
```

Como se observa, el estado de la sesión es *idle*. La declaración de vecino BGP le dice al proceso BGP la dirección IP de destino de cada paquete de actualización. El router debe decidir qué dirección IP usa la dirección IP de origen en la actualización de enrutamiento BGP. Cuando un router genera un paquete, si está en la actualización de enrutamiento, un ping, u otro tipo de paquete IP, el router hace una búsqueda en la tabla de enrutamiento para la dirección de destino. La tabla de enrutamiento lista la interfaz apropiada para obtener la dirección de destino. La dirección de ésta interfaz saliente se utiliza como dirección de origen de ese paquete.

Para los paquetes BGP, ésta dirección IP de origen debe coincidir con la dirección correspondiente en la declaración de vecindad del otro router. (En otras palabras, el otro router debe tener una relación BGP con la dirección de origen del paquete). Dicho de otra manera, los routers no serán capaces de establecer la sesión BGP, y el paquete será ignorado si no se cumplen las condiciones anteriores. BGP no acepta actualizaciones no solicitadas; debe ser consciente de todos los routers vecinos y tener una declaración de vecindad para ello.

5.10 Enviando tráfico desde la interfaz Loopback

En éste caso, R2 y R3 no establecen la sesión BGP porque, a pesar de que las direcciones IP de los vecinos son correctas, cada router espera que los paquetes BGP se originen desde la dirección loopback 0 de otro nodo. Tenemos que decirle a BGP que use la dirección de interfaz loopback en lugar de una dirección de interfaz física como la dirección de origen de todos los paquetes BGP, incluyendo esos que inician la conexión TCP de los vecinos BGP. Hay que el comando de configuración del router **neighbor** *ip-address* **update-source loopback** *interface-number* para provocar que el router use la dirección de la interfaz loopback específica como la dirección de origen para las conexiones BGP a éste vecino. El comando **neighbor update-source** es necesario para ambos routers.

En R2 y R3, vamos a la fuente de los paquetes iBGP desde las direcciones loopback 0 y verificamos los distintos nodos.

Configuración y verificación de que R2 y R3 han establecido una relación de vecindad a través de los interfaces loopback

```
>>>R2(config)# router bgp 65000
>>>R2(config-router)# neighbor 192.168.3.3 update-source Loopback 0

>>>R3(config)# router bgp 65000
>>>R3(config-router)# neighbor 192.168.2.2 update-source Loopback 0

>>>R3# show ip bgp summary
>>>Neighbor        V    AS   MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down   State/PfxRcd
192.168.2.2       4  65000       8        8       12    0     0  00:02:38        2
```

La salida del comando **show ip bgp summary** tiene un 2 en la columna State/PfxRcd; esto indica que la sesión BGP entre los dos routers está en el estado *established* y R3 ha recibido dos prefijos desde su vecino, R3.

Estableciendo la sesión BGP entre las direcciones IP loopback puede ayudar también a incrementar la resistencia de una conexión eBGP. Si existen múltiples rutas entre dos vecinos eBGP, la sesión sobrevivirá sin importar que rutas permanezcan disponibles. Las direcciones loopback deber ser alcanzadas desde ambos sitios respectivamente. A diferencia de otras redes empresariales internas, donde un IGP proporciona accesibilidad para las direcciones loopback usadas por los nodos iBGP, donde normalmente hay que configurar las rutas estáticas a las respectivas direcciones IP loopback remotas.

En R1 y R2, vamos a configurar una ruta estática para alcanzar la dirección de loopback 0 de otro router, porque estos routers no están ejecutando un IGP, y luego tenemos que configurar los nodos eBGP entre las direcciones loopback 0 y verificar las relaciones de vecindad de los diferentes nodos.

Configuración y verificación de que R1 y R2 han establecido una relación de vecindad a través de los interfaces loopback

```
>>>R1(config)# ip route 192.168.2.2 255.255.255.255 209.165.202.130
>>>R1(config)# router bgp 65100
>>>R1(config-router)# no neighbor 209.165.202.130
>>>R1(config-router)# neighbor 192.168.2.2 remote-as 65000
>>>R1(config-router)# neighbor 192.168.2.2 update-source Loopback 0

>>>R2(config)# ip route 209.165.201.1 255.255.255.255 209.165.202.129
>>>R2(config)# router bgp 65000
>>>R2(config-router)# no neighbor 209.165.202.129
>>>R2(config-router)# neighbor 209.165.201.1 remote-as 65100
>>>R2(config-router)# neighbor 209.165.201.1 update-source Loopback 0

>>>R1# show ip bgp summary
>>>Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.2      4 65000       0       0        1    0    0 never    Idle
```

Como se observa en la salida de los comandos la conexión eBGP entre las direcciones loopback permanece en *idle*, a pesar de que la configuración de enrutamiento y la configuración del vecino BGP sería suficiente para nodos iBGP. La sesión no se convierte en *established* porque las direcciones de los vecinos eBGP deben ser por defecto directamente adyacentes.

5.11 eBGP Multihop

Para solucionar el problema expuesto anteriormente, se debe habilitar también el eBGP multihop, con el comando de configuración de router **neighbor** *ip-addresses* **ebgp-multihop**[*ttl*].

Éste comando permite al router aceptar e intentar conexiones BGP con nodos externos residentes en las redes que no están directamente conectadas. Éste comando, aumenta el valor predeterminado de un salto para nodos eBGP cambiando el valor por defecto del Time To Live (TTL) de 1 (con el parámetro *ttl*) por lo tanto, permitimos rutas a direcciones loopback eBGP. Por defecto, el TTL se establece a 255 con éste comando. Éste comando es muy usado cuando existen rutas redundantes entre vecinos eBGP. Algunos escenarios donde no se pueden usar las direcciones IP adyacentes, son aquellas con nodos que cuenten con routers third-party, conexiones sobre una capa 3 hop, y conexiones avanzadas como Multiprotocol Label Switching(MPLS).

Vamos a configurar un multihop eBGP entre los routers R1 y R2 y verificar que los nodos establecen conexón correctamente.

Configurando y verificando R1 y R2 usando eBGP Multihop

>>>R1(config)# router bgp 65100

>>>R1(config-router)# neighbor 192.168.2.2 ebgp-multihop

>>>R2(config)# router bgp 65000

>>>R2(config-router)# neighbor 209.165.201.1 ebgp-multihop

>>>R1# show ip bgp summary

>>>Neighbor	*V*	*AS*	*MsgRcvd*	*MsgSent*	*TblVer*	*InQ*	*OutQ*	*Up/Down*	*State/PfxRcd*
192.168.2.2	*4*	*65000*	*6*	*5*	*12*	*0*	*0*	*00:00:30*	*2*

Ésta vez la conexión eBGP se establece correctamente, como indica el 2 en la columna State/PfxRcd en el router R1; R1 ha recibido dos prefijos desde su vecino R2.

5.12 Ressetting BGP Sessions

BGP puede manejar potencialmente grandes volúmenes de información de enrutamiento. Cuando se produce un cambio de configuración de política BGP (como cuando accede a listas, temporizadores, o los atributos se cambian), el router no puede ir a través de una gran tabla de información BGP y recalcular que entrada ya no es válida en la tabla local. Ni tampoco el router determina que ruta o rutas, ya anunciadas, deberían ser retiradas desde un vecino. Hay un riesgo obvio de que el primer cambio de la configuración seguido inmediatamente de un segundo, que causaría que todo el proceso empezara otra vez. Para evitar un problema, el software CISCO IOS aplica cambios sólo en esas actualizaciones recibidas o enviadas después del cambio de configuración de política BGP que se ha realizado. La nueva política, reforzada por nuevos filtros, es aplicada sólo en las rutas recibidas o enviadas después del cambio.

Si el administrador de la red quiere que un cambio de política se aplique en todas las rutas, se debe activar una actualización para forzar al router dejar pasar todas las rutas a través del nuevo filtro. Si el filtro es aplicado a la información saliente, el router tiene que volver a enviar la tabla BGP a través del nuevo filtro. Si el filtro es aplicado a la información entrante, el router necesita que sus vecinos reenvíen su tabla BGP que pasará a través del nuevo filtro.

Hay dos formas de activar una actualización: un hard reset, y un soft reset, el cual también es llamado *"route refresh"*.

5.12.1 Hard Reset de sesiones BGP

El restablecimiento de una sesión es un método de informar al vecino o vecinos de un cambio de política. Si las sesiones BGP son restablecidas, toda la información recibida en estas sesiones es invalidada y borrada de la tabla BGP. El vecino remoto detecta una sesión BGP en estado ***down***

e, igualmente, invalida las rutas recibidas. Después de un periodo de 30 a 60 segundos, las sesiones BGP son reestablecidas automáticamente, y la tabla BGP es intercambiada de nuevo, pero a través de los nuevos filtros. Sin embargo, el restablecimiento de la sesión BGP interrumpe el envío de paquetes.

Hay que usar el comando privilegiado EXEC **clear ip bgp *** o **clear ip bgp** [*neighbor-address*] para causar un hard reset a los vecinos BGP involucrados, donde ***** indica todas las sesiones y *neighbor-address* identifica la dirección del vecino específico para el que las sesiones BGP serán restablecidas. Un hard reset significa que el router que está emitiendo cualquiera de estos comandos cerrará adecuadamente las conexiones TCP, reestableciendo éstas sesiones TCP según sea apropiado, y reenviará toda la información a cada vecino afectado por el comando utilizado.

El comando **clear ip bgp *** causa la tabla de reenvío BGP en el router en el que éste comando se emite para ser eliminado completamente; todas las redes deben ser reaprendidas desde cada vecino. Si un router tiene múltiples vecinos, ésta acción es un evento muy peligroso ya que fuerza a todos los vecinos a reenviar sus tablas enteras simultáneamente.

Si, en lugar de eso, se usa el comando **clear ip bgp** *neighbor-address*, se reinician los vecinos uno a uno. El impacto es menos severo en el router que emite éste comando. Sin embargo, tarda más en cambiar la política de todos los vecinos porque debera hacerse individualmente en lugar de todos a la vez como con el comando **clear ip bgp *.** El comando **clear ip bgp** *neighbor-address* realiza un hard reset y debe reestablecer la sesión TCP con la dirección específica usada en el comando, pero este comando afecta únicamente a un solo vecino, no a todos los vecinos simultáneamente.

5.12.2 Soft Reset

Tenemos que usar el comando privilegiado EXEC **clear ip bgp** {* | *neighbor-address*} para provocar que BGP haga un ***soft reset*** de las actualizaciones salientes. El router en el que se emite éste comando no reestablece la sesión BGP. En lugar de eso, el router crea una nueva actualización y envía la tabla completa a los vecinos específicos. Ésta actualización incluye los comandos retirados por las redes que el vecino no verá más, basándose en las nuevas políticas salientes.

La configuración saliente soft BGP no tiene ninguna sobrecarga de memoria. Éste comando es altamente recomendado cuando estamos cambiando la política saliente, pero no ayuda si cambiamos la política entrante.

En R2, del ejemplo anterior, vamos a habilitar la actualización BGP de depuración con el comando **debug ip bgp updates**, y luego en R1 realizaremos un soft reset saliente para su relación de vecindad con R2.

Soft reset de las actualizaciones salientes BGP

>>>R2# debug ip bgp updates

>>>BGP updates debugging i on for address family: IPv4 Unicast

>>>R1# clear ip bgp 192.168.2.2 out

>>>R2#

>>>BGP: nbr_topp global 209.165.201.1 IPv4 Unicast: base (0xEC245CF8:1) rcvd Refresh

>>>Start-of-RIB

>>>BGP: nbr_topo global 209.165.201.1 IPv4 Unicast: base (0xEC245CF8:1) refresh_epoch is 3

>>>BGP(0): 209.165.201.1 rcvd UPDATE w/ attr: nexthop 209.165.201.1, origin i, metric 0, merged path 65100, AS_PATH

>>>BGP(0): 209.165.201.1 rcvd 209.165.200.224/27… duplicate ignored

>>>BGP: nbr_topo global 209.165.201.1 IPv4 Unicast: base (0xEC245CF8:1) rcvd Refresh

>>>End-of-RIB

>>>R2# no debug all

>>>All possible debugging has been turned off

En la configuración anterior, hay que darse cuenta de que cuando se activa el ***soft reset*** saliente de R1 hacia R2, todos los prefijos existentes en la tabla BGP que no han sido recibidos desde R2 son reenviados a R2. En éste caso, la información recibida es un duplicado de la entrada previa y R2 lo ignora.

No hay que olvidar desactivar la depuración con el comando **no debug all**, como se hace al final.

Cuando una sesión BGP es reestablecida usando la reconfiguración soft, los siguientes comandos pueden ser usados para la monitorización de las rutas BGP recibidas, enviadas o filtradas:

- **show ip bgp neighbors** {*address*} **received-routes**: Muestra todas las rutas recibidas (tanto las aceptadas como las rechazadas) desde un vecino específico.
- **show ip bgo neighbors** {*address*} **routes:** Muestra todas las rutas que son recibidas y aceptadas desde un vecino específico. Esta salida es un subconjunto de la salida mostrada por la palabra clave **received-routes**.
- **show ip bgp**: Muestra las entradas en la tabla BGP.
- **show ip bgp neighbors** {*address*} **advertised-routes**: Muestra todas las rutas BGP que han sido anunciadas por los vecinos.

Capítulo 6. Controlando las actualizaciones BGP

Si hay múltiples rutas entre nuestra red y el ISP, se puede necesitar filtrar cierta información durante el intercambio de actualizaciones BGP para influenciar la selección de ruta o reforzar una política administrativa. A lo largo de este capítulo, se profundizará en estas opciones para modificar el envío de los mensajes de actualización. Los grupos de nodos BGP son usados para agrupar nodos con políticas similares en conjunto para una configuración más sencilla.

6.1 Filtrando actualizaciones BGP

BGP permite filtrar las actualizaciones que son recibidas desde un vecino o que son enviadas a un vecino. Las herramientas principales de filtrado de rutas incluyen mapas de rutas y listas de prefijos. Un escenario común donde el filtro de actualización es usado son las redes dual-homed. En este tipo de redes, una empresa debería anunciar solo su propio espacio de direcciones IP a los diferentes ISPs. Si la empresa anuncia bloques de direcciones recibidas desde un ISP, otros ISPs pueden usar el AS empresarial para el tránsito de tráfico y la empresa se convertirá en un AS de tránsito. Para ello nos serviremos de los siguientes filtros para evitar estas situaciones que son tan perjudiciales para la red de una empresa.

6.1.1 Filtrado BGP utilizando listas de prefijos

En este capítulo explicaremos el uso de las listas de prefijos para los filtros de ruta BGP.

El comando de configuración del router **neighbor** *ip-address* **prefix-list** *prefix-list-name* {**in** | **out**} es usado para aplicar una lista de prefijos a las rutas desde o hacia un vecino; estos parámetros podemos verlos en la siguiente tabla

Parámetro	Descripción
ip-address	Dirección IP del vecino BGP
prefix-list-name	Nombre de la lista de prefijos
in	La lista de prefijos es aplicada a los anuncios de entrada
on	La lista de prefijos es aplicada a los anuncios de salida

Tabla 3. Definición de parámetros para configuración

Por ejemplo, en la siguiente imagen, el R2 está configurado como se indica en el recuadro más abajo. La lista de prefijos ANY-8to24-NET es configurada para combinar rutas desde cualquier red que tuviera una longitud de máscara de 8 a 24 bits. La combinación 0.0.0.0/0 red/longitud no coincide con una red específica; en lugar de eso, define cualquier red. Los parámetros **ge 8 and le 24** especifican que cualquier red con la longitud de máscara entre 8 y 24 coincida con la lista de prefijos.

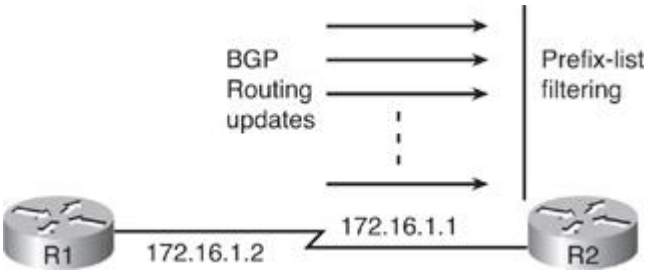


Figura 18. Intermcambio de mensajes de actualización BGP

Configuración R2

>>>router bgp 65001

>>neighbor 172.16.1.2 remote-as 65002

>>neighbor 172.16.1.21 prefix-list ANY-8to24-NET in

>>>|

>>>ip prefix-list ANY-8to24-NET permit 0.0.0.0/0 ge 8 le 24

La lista de prefijos ANY-8to24-NET se aplica a los anuncios entrantes del vecino BGP 172.16.1.2. Esto permite rutas desde cualquier red con una longitud de máscara de 8 a 24 bits.
Se puedes usar el comando **show ip prefix-list detail** para mostrar información detallada sobre la configuración de las listas de prefijos. Usando el comando **clear ip prefix-list** prefix-list-name [*network*/*length*] restablecemos el recuento de las entradas de la lista de prefijos, que se obtienen mediante el comando **show ip prefix-list detail**.

6.1.2 Filtrado BGP usando listas de acceso con AS-Path

Varios escenarios requieren filtrado y selección de información de enrutamiento basándose en el contenido del atributo *AS-path* incluido en ruta BGP.

Hay que recordar que el atributo *AS-path* se refiere a la lista de AS que una ruta ha atravesado para alcanzar un determinado destino, con el número del AS que originó la ruta al final de la lista. Cuando una ruta BGP es originada como resultado del comando **network** en un proceso BGP o de redistribución dentro de un proceso BGP, se crea el atributo *AS-path* y está vacío. Cada vez que una ruta es anunciada por un router de salida a otro AS, el atributo *AS-path* es modificado por el router de salida, que propone su número AS al atributo *AS-path.*

Los routers pueden filtrar las rutas entrantes basadas en sus atributos *AS-path*. Por ejemplo, un AS que quiere filtrar todas las rutas que son locales a sí mismo antes de enviarlas a un AS vecino puede permitir que solo se envíen rutas con el *AS-path* vacío, y negar el envío de todas las demás. Otro ejemplo, un AS puede no querer que las rutas desde un AS específico sean recibidas desde un cierto vecino; en este caso, las rutas con ese AS en el *AS-path* pueden ser fitradas en el router receptor.

Cuando los routers filtran actualizaciones BGP basados en el contenido del atributo AS-path, usan expresiones regulares. Las expresiones regulares son comúnmente encontradas en el ambiente UNIX y también en algunas aplicaciones basadas en Microsoft Windows. Las expresiones regulares son herramientas de concordancia de cadenas y consisten en cadenas de caracteres. Algunos de estos caracteres tienen un significado especial, como el funcionamiento con comodines y operadores. La siguiente tabla proporciona algunos ejemplos de los caracteres especiales usados en expresiones regulares. Algunos caracteres simplemente significan ellos mismos (por ejemplo, A a Z, a a z, o 0 a 9). Una expresión regular se dice que coincide con una cadena si los caracteres ordinarios y el significado aplicado de los caracteres del operador especial puede ser traducido dentro de la cadena adaptada. Cuando una expresión regular coincide, el test de selección se da como correcto. Si no coincide, el test es falso. Por ejemplo, la combinación ^$ significa una cadena vacía; esta podría ser usada cuando está buscando todas las rutas que son originadas en el AS local.

Parámetro	Descripción
.	Coincide con cualquier carácter
*	Coincidentes 0 o más secuencias de un patrón
^	Coincide con el principio de la cadena
$	Coincide con el final de la cadena
_ *(underscore)*	Coincide con una coma, la llave izquierda, la llave derecha, el paréntesis izquierdo, el paréntesis derecho, el principio de la cadena, el final de la cadena, o un espacio

Tabla 4. Distintas expresiones regulares

La lista de acceso a las rutas del AS es definida por el comando de configuración global **ip as-path** access-list *access-list-number* {**permit** | **deny**} *regexp*. Los parámetros de este comando son descritos en la siguiente tabla.

Parámetro	Descripción
access-list-number	Número de 1 a 500 que especifica el número de la lista de acceso AS-path
permit | *deny*	Indica si esta entrada permite o bloquea si la expresión regular es cierta
regexp	La expresión regular que define el filtro AS-path. El número AS esta expresado en el rango de 1 a 65535

Tabla 5. Definición de parámetros para configuración

El comando de configuración de router **neighbor** *ip-address* **filter-list** *access-list-number* {**in** | **out**} se usa para aplicar la lista de acceso AS-path para las rutas desde o hacia los diferentes vecinos. Los parámetros de este comando son descritos en la tabla a continuación.

Parámetro	Descripción
ip-address	La dirección IP del vecino BGP
access-list-number	El número de la lista de acceso del AS-path

in	Lista de acceso es aplicada a las rutas entrantes
out	Lista de acceso es aplicada a las rutas de salida

Tabla 6. Definición de parámetros para configuración

Las rutas que están permitidas por la lista de acceso AS-path son permitidas para ser recibidas desde o enviadas al vecino, y las que se deniegan no son incluidas. Como en todas las listas de acceso, el candidato a ser permitido o denegado es testeado contra las líneas en la lista de acceso en el orden en el que la lista está configurada. La primera coincidencia indica *"permit"* o *"deny"*. Si el final de la lista de acceso es alcanzado sin ninguna coincidencia explícita, el candidato es implícitamente denegado.

En el caso de un cliente ISP multihomed como en de la imagen. El AS 65000 no debe actuar como un AS de tránsito entre sus proveedores de servicios. Esta situación se puede evitar asegurándose de que las únicas rutas de origen son enviadas al ISP, y luego el cliente evite recibir paquetes IP desde los ISPs para destinos fuera de su propio AS. La configuración para los routers GW1 y GW2 se muestra también en los cuadros de configuración siguientes. La lista de acceso AS-path permite sólo la cadena vacía, que coincide con la expresión regular ^$, que representa las rutas de origen locales. Para aplicar esta lista de filtros en la información de salida de todos los vecinos, el vecino anuncia solo sus rutas locales.

Configuración routers GW1 y GW2

>>>GW1(config)# ip as-path access-list 1 permit ^$

>>>GW1(config)# router bgp 65000

>>>GW1(config-router)# neighbor 209.165.201.1 filter-list 1 out

>>>GW2(config)# ip as-path access-list 1 permit ^$

>>>GW2(config)# router bgp 65000

>>>GW2(config-router)# neighbor 209.165.201.5 filter-list 1 out

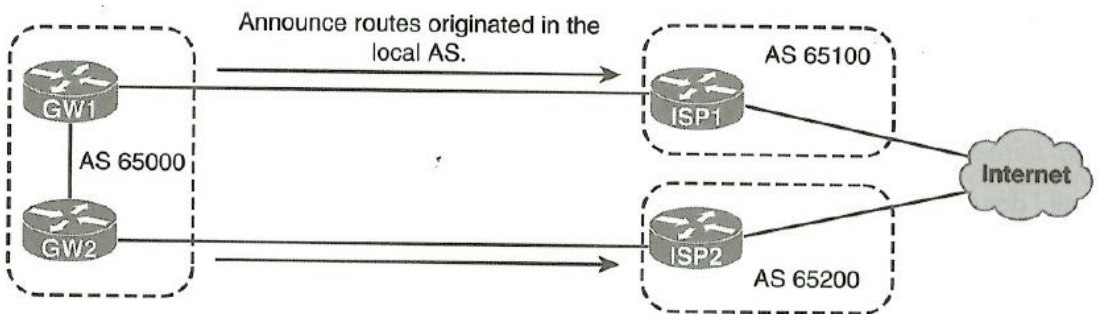


Figura 19. Filtrado de rutas mediante el parámetro *AS-PATH*

6.1.3 Filtrado BGP usando mapas de rutas

Los mapas de rutas ofrecen una gran flexibilidad para manipular las actualizaciones BGP. Un mapa de rutas puede combinar y establecer varios atributos BGP diferentes, incluyendo los siguientes:

- **Origin**
- **Next hop**
- **Community**
- **Local preference**
- **MED**

Los mapas de rutas pueden coincidir con otros ítems, incluyendo los siguientes:

- Número de red y máscara de subred (con la lista de prefijos IP)
- El originador de ruta
- La etiqueta adjunta a una ruta IGP
- AS-path
- Tipo de ruta (interna o externa)

Si todas las cláusulas de concordancia dentro de una declaración de mapa de rutas coinciden, la declaración es considerada una coincidencia, y la declaración es ejecutada (permitida o denegada, como se especifique). Cuando se utiliza para el filtrado BGP, un *deny* significa que la ruta es ignorada y un *permit* significa que la ruta es procesada y las cláusulas establecidas son aplicadas. Las cláusulas establecidas permiten que uno o más atributos se cambien o se establezcan para valores específicos antes de que la ruta pase al mapa de rutas.

Para aplicar un mapa de rutas para filtrar las rutas entrantes o salientes BGP, hay que usar el comando de configuración del router **neighbor route-map**. Las rutas que están permitidas pueden tener sus atributos establecidos o cambiados, usando comandos **set** en el mapa de rutas. Este es útil cuando se intenta influir en la selección de ruta.

En la red de ejemplo de la siguiente imagen, el cliente acepta únicamente una ruta por defecto desde dos ISPs y usa el link del AS 65100 como su link primario para el tráfico saliente. Para ello hacemos uso de la configuración del GW que se en el recuadro de configuración.

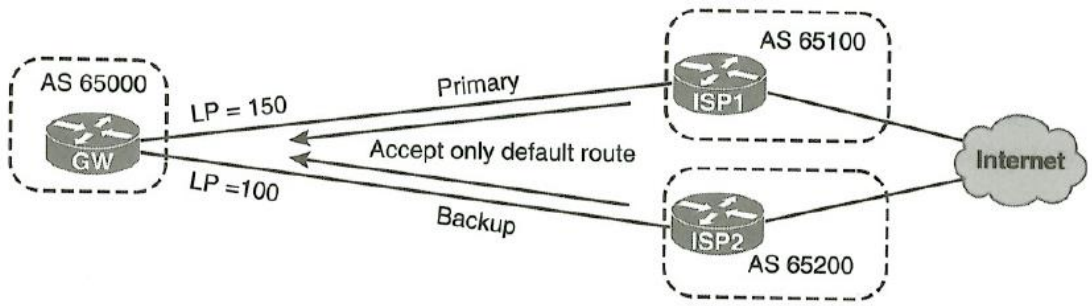


Figura 20. Filtrado de rutas mediante mapas de rutas

Configuración GW

>>>router bgp 65000

>>neighbor 209.165.201.1 remote-as 65100

>>neighbor 209.165.201.1 route-map FILTER in

>>neighbor 209.165.201.5 remote-as 65200

```
    >>neighbor 209.165.201.5 route-map FILTER in
>>>|

>>>route-map FILTER permit 10
    >>match ip address prefix-list default-only
    >>match as-path 10
    >>set local-preference 150
>>>|

>>>route-map FILTER permit 20
    >>match ip address prefix-list default-only
>>>|

>>>ip as-path access-list 10 permit ^65100$
>>>ip prefix-list default-only permit 0.0.0.0/0
```

El router GW es configurado para BGP con dos vecinos usando los comandos **neighbor remote-as**. Ambos vecinos están configurados con el comando **neighbor route-map** por lo que filtran el tráfico de actualización de enrutamiento entrante de acuerdo con el mapa de rutas llamado *FILTER*. El mapa de rutas *FILTER* permite solo una ruta por defecto dentro de la red cliente, como se define por la lista de prefijos *default-only*. La ruta por defecto viene desde el ISP1 en el AS 65100, como se define en la lista de acceso 10 *AS-path*, que se asignada al valor *local-preference* de 150 y todas las demás rutas (en este caso, la que viene del ISP2 en el AS 65200) tendrán el valor *local-preference* por defecto a 100. Debido a que el valor *local-preference* alto se prefiere por defecto, el link de ISP1 del AS 65100 es el escogido.

6.1.3.1 Orden de filtrado

Las listas de filtros (para filtros *AS-path*), listas de prefijos, y mapas de rutas pueden ser aplicados ya sea información BGP entrante o saliente, o cualquier otra combinación.

La lista de filtros entrante, la lista de prefijos (o lista de distribución) y el mapa de rutas (en este orden) deben permitir las rutas que son recibidas desde un vecino antes de ser aceptadas dentro de la tabla BGP. Similarmente, las rutas salientes deben pasar la lista de filtros saliente, el mapa de rutas, y la lista de prefijos (o lista de distribución) (en este orden) antes de ser enviadas al vecino.

6.1.3.2 Clearing BGP sesión

Si queremos que una política de cambio, como un filtro BGP, sea aplicada, se debe activar una actualización para forzar al router que deje las rutas apropiadas pasar a través del filtro. Se puede hacer tanto un hard reset como una route refresh (soft reset); tal y como vimos en capítulos anteriores.

6.2 Grupos de nodos BGP

En BGP, los vecinos son a menudo configurados con la misma política de actualización. En los router con Cisco IOS, los vecinos con las mismas políticas de actualización pueden ser agrupados en nodos grupales para simplificar la configuración y, más importante, hacer la actualización más eficiente y mejorar el rendimiento. Cuando un router BGP tiene muchos nodos, este enfoque es altamente recomendado.

6.2.1 Peer group operation

Un grupo de nodos BGP es un grupo de vecinos BGP del router que se han configurado con las mismas políticas de actualización.

En vez de definir por separado las mismas políticas para cada vecino, un grupo de nodos puede ser definido como con estas políticas asignadas al grupo de nodos. Los vecinos individuales se hacen entonces miembros del grupo de nodos. Las políticas del grupo de nodos son similares a una plantilla; la plantilla es entonces aplicada a los miembros individuales del grupo de nodos.

Los miembros del grupo de nodos heredan todas las opciones de configuración del grupo de nodos. El router puede también ser configurado para anular estas opciones para algunos miembros del grupo de nodos si estas opciones no afectan a las actualizaciones salientes. En otras palabras, solo las opciones que afectan a las actualizaciones entrantes pueden ser anuladas.

Los vecinos BGP de un solo router pueden ser divididos en varios grupos, cada grupo teniendo sus propios parámetros BGP.

La configuración de un grupo de nodos puede tener muchas características, incluyendo las siguientes:

- Update-source
- Next-hop-self
- Ebgp-multihop
- Autenticación de las sesiones BGP
- Cambiar el peso de las rutas recibidas.
- Los filtros de las rutas entrantes o salientes usando la lista de prefijos, la lista de filtros, y el mapa de rutas.

Cuando los routers vecinos son asignados a un grupo de nodos en un router, todos los atributos que están configurados para el grupo de nodos son aplicados a todos los miembros del grupo de nodos. El Cisco IOS Software optimiza las rutas salientes ejecutando a través de los filtros salientes y los mapas de ruta solo una vez y luego replicando los resultados en cada miembro del grupo de nodos. El Cisco IOS Software asigna un líder al grupo de nodos, para que el software genere una actualización, y esta actualización es replicada por el líder en todos los otros miembros del grupo.

6.2.2 Peer group configuration

El comando de configuración del router **neighbor** *peer-group-name* **peer-group** es usado para crear un grupo de nodos BGP. El *peer-group-name* es el nombre del grupo de nodos BGP que va a ser creado. El *peer-group-name* es local al router en el que está configurado; no se pasa a ningún otro router.

Otra opción del comando **neighbor peer-group**, el comando de configuración del router **neighbor** *ip-address* **peer-group** *peer-group-name*, es usado para asignar vecinos como parte del grupo después de que el grupo haya sido creado. La siguiente tabla proporciona los detalles de los parámetros de este comando. Utilizando este comando permitimos escribir el nombre del grupo de nodos en lugar de escribir las direcciones IP de los vecinos individuales con otros comandos (por ejemplo, para vincular una política al grupo de routers vecinos).

Parámetro	Descripción
ip-address	La dirección IP del vecino que está asignado como miembro del grupo de nodos
peer-group-name	El nombre del grupo de nodos BGP

Tabla 7. Definición de parámetros para configuración

Un router vecino puede ser parte únicamente de un grupo de nodos. El comando EXEC **clear ip bgp peer-group** *peer-group-name* es usado para reestablecer las conexiones BGP de todos los miembros del grupo de nodos BGP. El peer-group-name es el nombre del grupo de nodos BGP para que las conexiones sean claras.

6.2.3 Peer group configuration example

Hay varios escenarios donde los grupos de nodos son aplicables. Por ejemplo, las sesiones iBGP casi siempre se configuran de forma idéntica. Si una malla completa es desplegada sin un AS, pueden existir un gran número de configuraciones vecinas; configurando estos resultados por separado puede derivar en una tremenda cantidad de configuración redundante.

Otro caso de uso es la configuración de un router ISP con múltiples nodos BGP propiedad del cliente. Los AS del cliente son todos anunciados únicamente en redes locales. Todos los AS de clientes deberían recibir actualizaciones BGP con el mismo conjunto de rutas de Internet, y los AS de clientes son asumidos para generar únicamente unos pocos prefijos. Esta situación hace que la configuración del vecino sea casi idéntica para cada cliente, con únicamente unos pocos cambios que son específicos para cada vecino.

La siguiente imagen ilustra un router de frontera de la empresa que mantiene múltiples sesiones con diferentes ISPs BGP. Estas sesiones externas comparten una serie de parámetros comunes que los hacen muy adecuados para una configuración grupal.

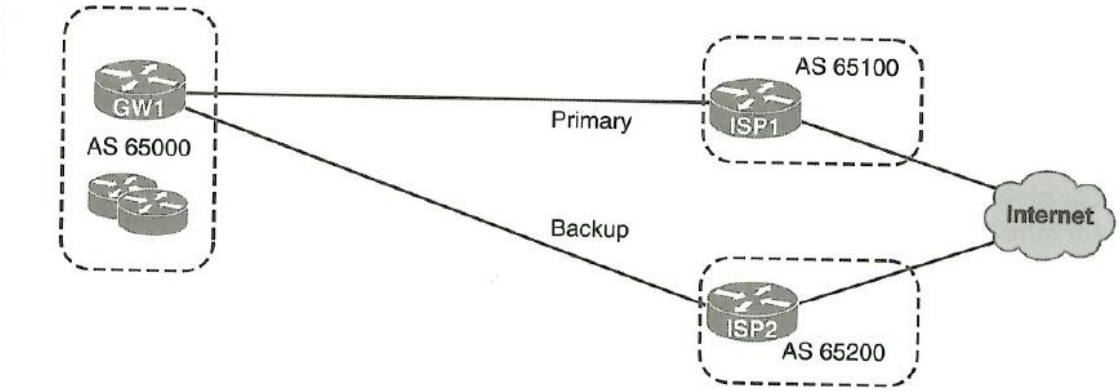


Figura 22. Ejemplo para configuración de un modo de nodos agrupados

Configuración del GW1

>>>router bgp 65000

>>neighbor ISP peer-group

>>neighbor ISP filter-list 10 out

>>neighbor ISP prefix-list desired-subnets in

>>neighbor ISP route-map FILTER in

>>>|

>>>neighbor 209.165.201.1 remote-as 65100

>>>neighbor 209.165.201.1 peer-group ISP

>>>neighbor 209.165.201.5 remote-as 65200

>>>negihbor 209.165.201.5 peer-group ISP

>>>|

>>>route-map FILTER permit 10

>>match as-patch 20

>>set local-preference 150

>>>|

>>>route-map FILTER permit 20

>>>|

>>>ip as-path access-list 10 permit ^$

*>>>ip as-path access-list 20 permit ^65100_*

>>>|

>>>ip prefix-list desired-subnets permit 0.0.0.0/0

>>>ip prefix-list desired-subnets permit 0.0.0.0/0 ge 8 le 24

El grupo de nodos llamado ISP comparte múltiples parámetros comunes; una lista de filtros salientes, una lista de prefijos entrante, y un mapa de rutas entrante. Los vecinos individuales, definidos mediante sus direcciones IP y números AS, han sido asignados al grupo de nodos. La

lista de filtros hace referencia a la lista de acceso IP *AS-path 10*, que permite anunciar solo las redes de origen en el AS local. La lista de prefijos entrante es usada para aceptar la ruta por defecto y subredes cuyas máscaras están en el rango entre 8 y 24. El mapa de rutas *FILTER* establece una alta preferencia local para redes recibidas desde el ISP primario (AS 65100).

Capítulo 7. Vulnerabilidades BGP

Sin embargo, como hemos visto anteriormente, BGP es un protocolo muy potente que se utiliza actualmente para soportar la infraestructura del Internet actual tal y como lo conocemos; pero tiene un gran problema y es que cuando se creó se hizo pensando en que todo nodo que formará parte de la red iba a ser *"bueno"*. Sin embargo, esto no es así, ya que a lo largo de la historia el fenómeno "hacker" se ha ido extendiendo hasta hacerse presente en nuestra vida cotidiana como paso hace algunas semanas con el virus WannaCry que dejó sin servicio a organismos públicos como hospitales. Esto representa un punto débil en su estructura que será lo que mostremos a lo largo de este capítulo, mediante ejemplos que pueden encontrarse fácilmente en la red.

7.1 Robo de información

Que Internet sigue teniendo agujeros abiertos es un secreto a voces conocido por todas aquellas personas dedicadas al mundo de la ciberseguridad y el pentesting. Uno de ellos, descubierto por dos expertos en seguridad informática, permite "secuestrar" cantidades nunca antes imaginadas de información. El punto débil tiene que ver con el protocolo BGP. El resultado, interceptar datos no encriptados que estén siendo transmitidos a cualquier parte del mundo, e incluso modificarlos antes de que lleguen a su destino. Internet se diseñó para facilitar la comunicación, no tanto para ser segura. Este punto de partida, que ha facilitado un grandioso crecimiento de la red, es también su parte más endeble, pues permite que ciertas personas exploten sus vulnerabilidades para sus propios intereses.

Este fallo tiene su principal foco sobre los grandes nodos, que permiten que la información fluya por Internet, que suelen ser routers que funcionan bajo el protocolo BGP. Cuando una persona envía un correo electrónico desde Madrid a, por ejemplo, Montevideo, los sistemas de su proveedor de Internet buscarán el camino más rápido para que dicho e-mail llegue a su destinatario.

Alguien con el equipo necesario podría engañar al router del proveedor haciéndole creer que ese camino más corto pasa por su sistema. La información seguiría llegando a su destinatario, pero el nuevo intermediario podría monitorizar los datos.

Al tratarse de un problema tan grande, los organismos ya están trabajando en un desarrollo del protocolo que impida malas actuaciones. Sin embargo, es necesario que los usuarios de Internet tengan cuidado cuando navegan, de la misma manera que si alguien visita por primera vez una ciudad no entra en según qué barrios sin saber dónde se mete o sin tomar ciertas precauciones.

7.2 Ciberespionaje

Los routers BGP son comunes en los proveedores de Internet (como Telefónica, Ono, Tele2). Éstos utilizan el citado protocolo para compartir la información de ruta, esto es, para localizar ordenadores en la red (que se identifican individualmente mediante direcciones IP). La puerta

entreabierta en este sistema, es un mecanismo pensado para que las agencias de inteligencia pudieran intervenir determinadas comunicaciones.

El fallo está en el propio funcionamiento del BGP, que se basa en la confianza. Sirviéndonos del anterior ejemplo, cuando alguien envía un correo de un país a otro, las diferentes compañías de Internet se comunican entre ellas con un router que les indica cuál es la ruta más eficiente para enviar la información a su destino; y BGP confía ciegamente en el veredicto. La tarea de los piratas informáticos es, pues, evidente: engañar a los routers para que les envíen a ellos la información. Este ataque se conoce como secuestro de IPs y, además de ser un negocio ilícito, no es la primera vez que da problemas.

Otro caso relacionado con el espionaje fue en 2013 cuando se descubrió una redirección del tráfico con origen en EE. UU hacia países como Islandia o Bielorrusia antes de llegar a su destino. El reporte de Renesys, empresa que descubrió este problema, indica que el tráfico afectado pertenecía a toda clase de remitentes, incluidos aquellos pertenecientes al gobierno, instituciones financieras y proveedores de servicios de Internet. No se trata de un ataque con víctimas definidas, pues la lista de afectados cambia diariamente.

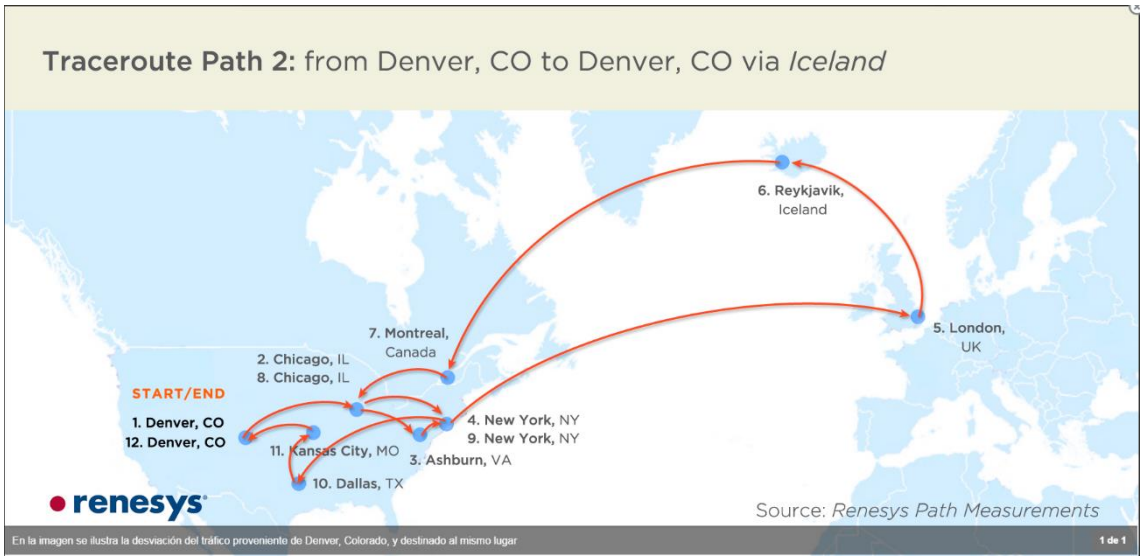


Figura 23. Desvío de tráfico procedente de EE.UU

La redirección del tráfico se efectuó mediante una vulnerabilidad en el protocolo de enrrutamiento Border Gaateway Protocol (BGP. La vulnerabilidad permite que los atacantes modifiquen las tablas de enrutamiento de los diferentes routers de tal manera que el tráfico de Internet se envíe a donde los hackers deseen. Posteriormente, la información es enviada a su destino original, por lo que la anomalía pasa desapercibida para las víctimas.

7.3 Bloqueo servicios WEB

Estos problemas que presenta BGP llevan a situaciones curiosas. Por ejemplo, en 2008 Pakistán quiso bloquear YouTube. Para ello, uno de los ISP del país, Pakistan Telecom, cambió una entrada en sus routers, que venía a decir "Para ir a las IPs de YouTube 21.3.4.5 o 21.5.4.3, mandad los

datos a la basura". Dicho de otra forma, ordenaron a sus routers que todo el tráfico destinado a YouTube se descartase para que no llegue a su destino.

El problema es que se equivocaron al hacer la configuración, y esa ruta se anunció a las redes conectadas con Pakistan Telecom, y en pocos minutos se extendieron por todo el mundo. Resultado: YouTube inaccesible por completo.

Este problema surgió del propio funcionamiento del protocolo, ya que suponiendo que quieres ir a una red determinada como 3.3.3.3, si tienes dos posibles rutas, una hacia una red que dice poder acceder a las IPs 1.1.1.1 a 5.5.5.5 y otra hacia otra red que accede a las IPs 3.3.3.0 a 3.3.3.5; BGP se decantará por la segunda porque el rango de IPs que anuncia es menor, más específico, y por lo tanto se supone que esa ruta es más eficiente.

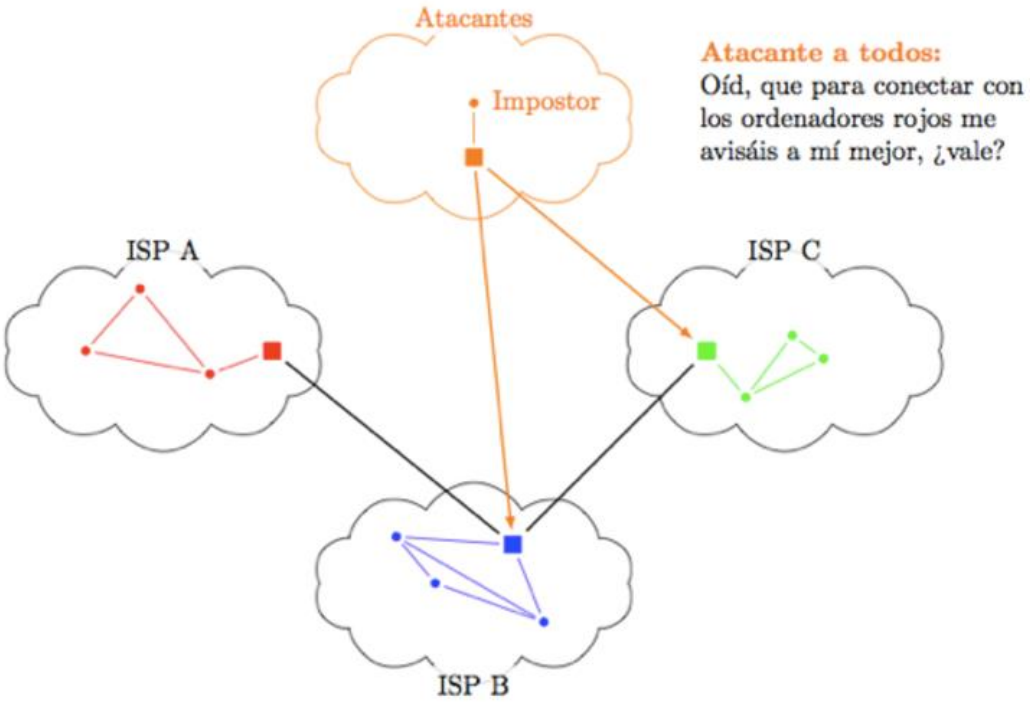


Figura 24. Modelo de ataque BGP

Lo que pasó es que todas las redes recibieron datos que decían que Pakistan Telecom sabía llegar a dos IPs (como puede verse en la imagen superior). Tal y como está diseñado BGP, se hace caso a la ruta que mejor acota la IP de destino1, que en este caso era la de Pakistan Telecom. Google lo resolvió en poco tiempo anunciando rangos de IP más pequeños todavía que sobreescribiesen los de Pakistan Telecom, y todo volvió a la normalidad.

Esto fue un accidente, pero en todo momento los protocolos funcionaron como tenían que funcionar. Ese es el problema de BGP: cualquiera puede anunciar rutas alternativas y desviar ingentes cantidades de tráfico. Es muy difícil distinguir cuándo es un cambio legítimo (si aparece una red nueva o un enlace deja de funcionar hay que cambiar las rutas, por ejemplo) de cuándo es malicioso.

Los expertos aseguran que los proveedores de Internet pueden evitar este tipo de ataque "al cien por cien", utilizando filtros potentes, pero que son bastante costosos.

7.4 Denegar acceso Internet

Durante las protestas de Egipto de 2011 el gobierno de Hosni Mubarak ordenó a todos los proveedores de acceso que operan en el país árabe el corte de las conexiones internacionales. Como consecuencia de los cortes y bloqueos en la noche del 27 al 28 de enero los routers egipcios dejaron de anunciar hasta 3.500 rutas de BGP, dejando al resto de routers sin la información necesaria para intercambiar tráfico con los servidores egipcios.

La caída súbita del tráfico en internet en el país africano puede apreciarse en una gráfica de firmas de monitoreo de la actividad en internet como Renesys y BGP Mon. Renesys publicó un reporte titulado "Egipto abandonó internet" que las solicitudes para enrutar el tráfico web a redes egipcias fueron rechazadas a partir de la media noche de forma masiva y súbita. Según una gráfica de BGP Mon la actividad de los routers egipcios se desplomó del jueves 27 de enero al viernes 28.

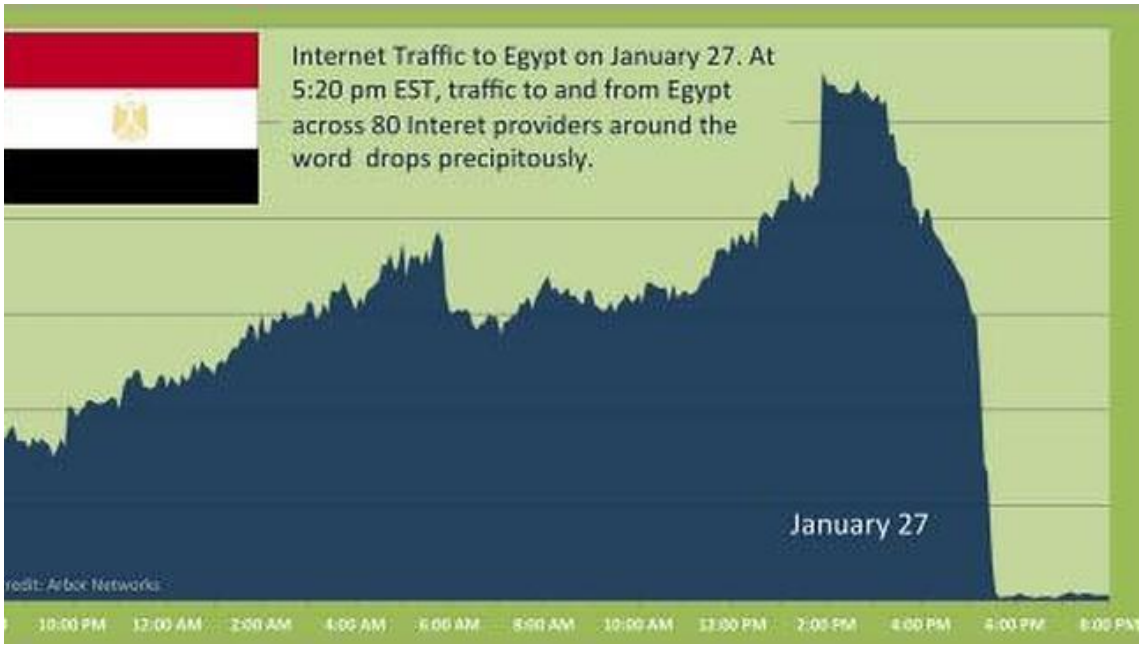


Figura 25. Estado rutas BGP en Egipto durante la primavera arabe

Capítulo 8. Diseño de una red utilizando BGP

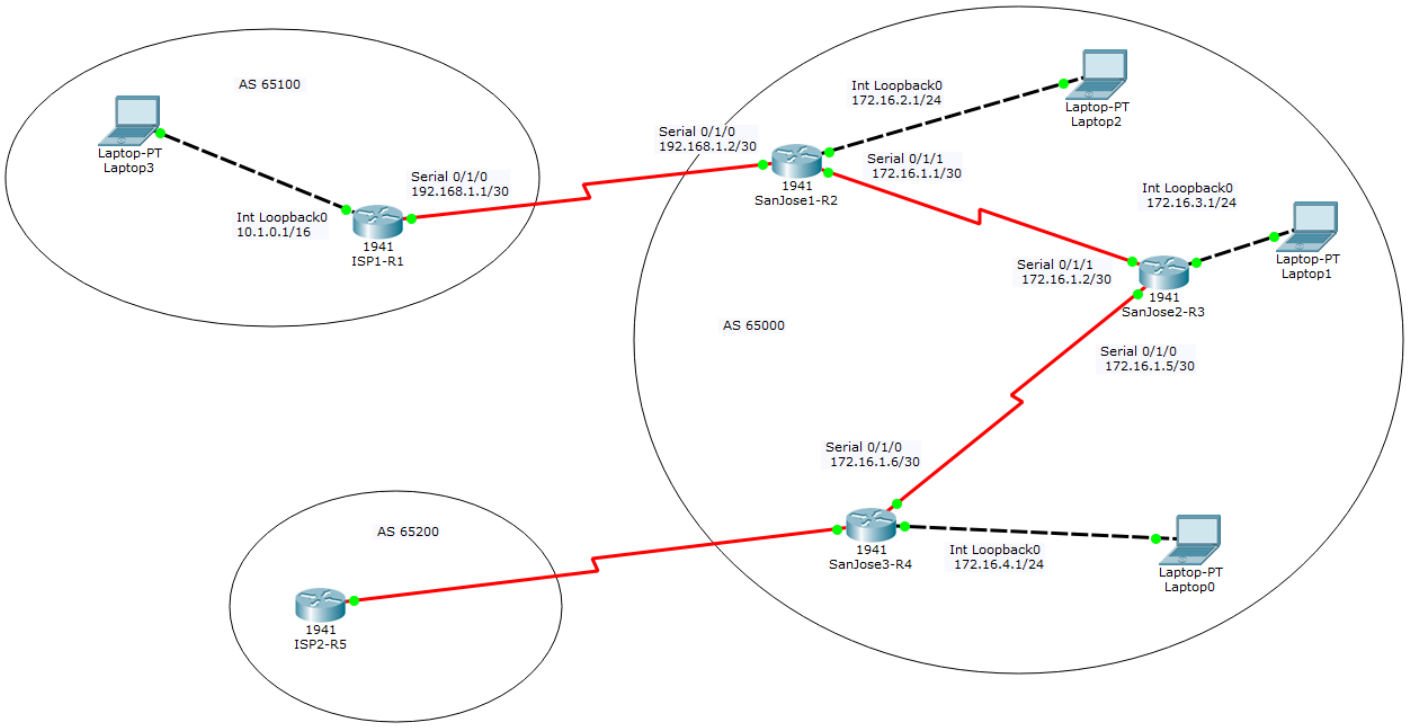


Figura 26. Red Ejemplo

Después de analizar en profundidad el protocolo BGP, para un mejor entendimiento del mismo utilizaremos uno de los esquemas ejemplo que propone Cisco para poner en práctica todos aquellos conocimientos adquiridos sobre BGP.

Básicamente, esta red esta compuesta por 3 AS distintos:

- AS 65100 – ISP1
- AS 65200 – ISP2
- AS 65000 – Red Cliente

Con este ejemplo, podremos visualizar la configuración del esquema de red más común entre los clientes BGP; para ello vamos a configurar la red del cliente para que pueda enrutar todo su tráfico hacia el ISP1. Para ello, como veremos durante el desarrollo del ejemplo tendremos que utilizar conocimientos de eBGP e iBGP; la implementación la llevaremos a cabo con el simulador Packet Tracer, aunque como he comentado anteriormente, al tener que utilizar el protocolo iBGP, una parte del ejemplo será desarrollada con equipamiento Cisco real por la imposibilidad del Packet Tracer de configurar iBGP.

En primer lugar, se lleva a cabo la configuración de las diferentes interfaces de los routers para que queden configurados tal y como vemos en la imagen; los comandos necesarios son los que se muestran en el siguiente recuadro (posteriormente, se adjuntara el fichero .pkt donde se establecen todos estos comandos y se pueden verificar el resto de configuraciones).

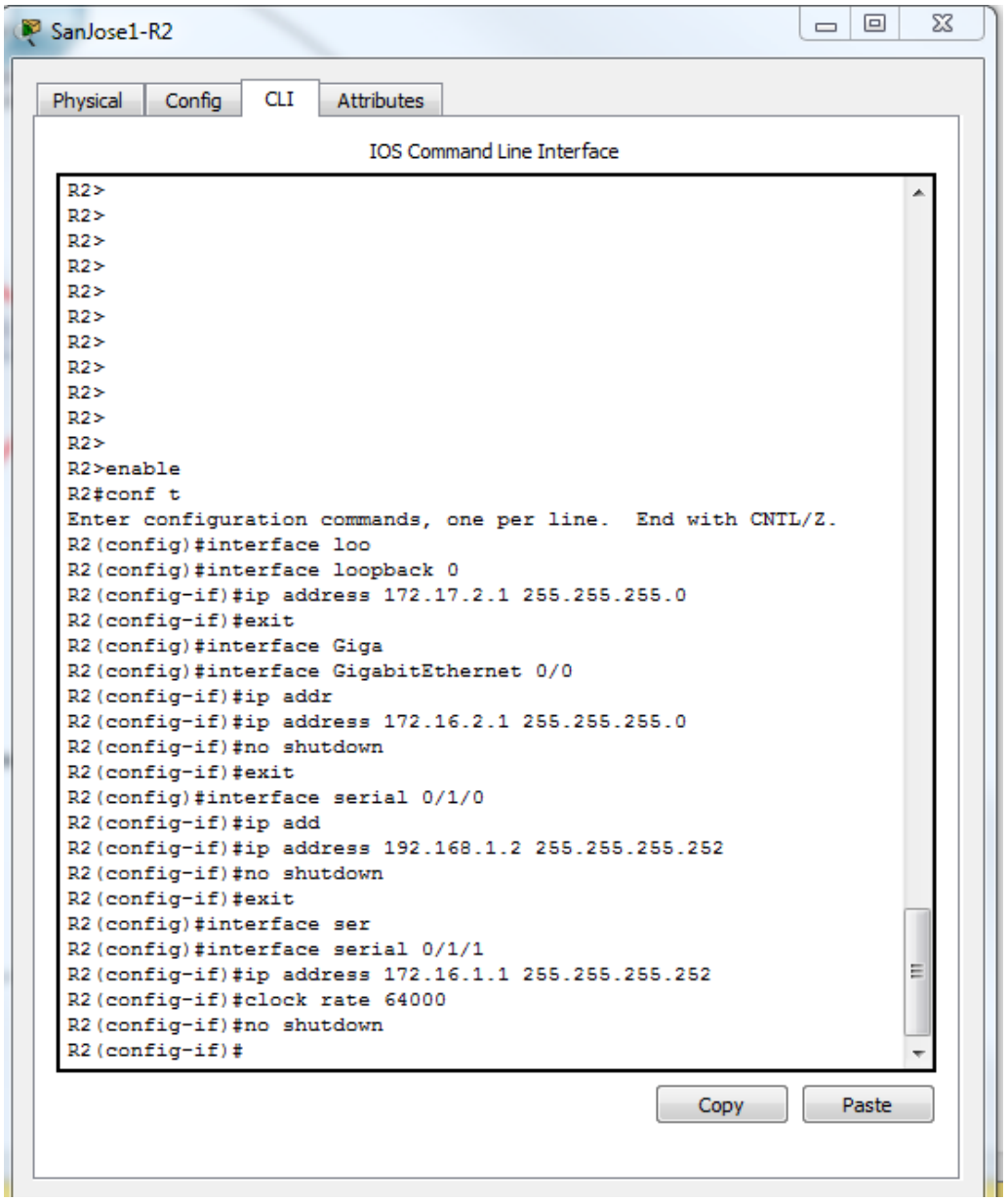


Figura 27. Configuración interfaces "*SanJose1*"

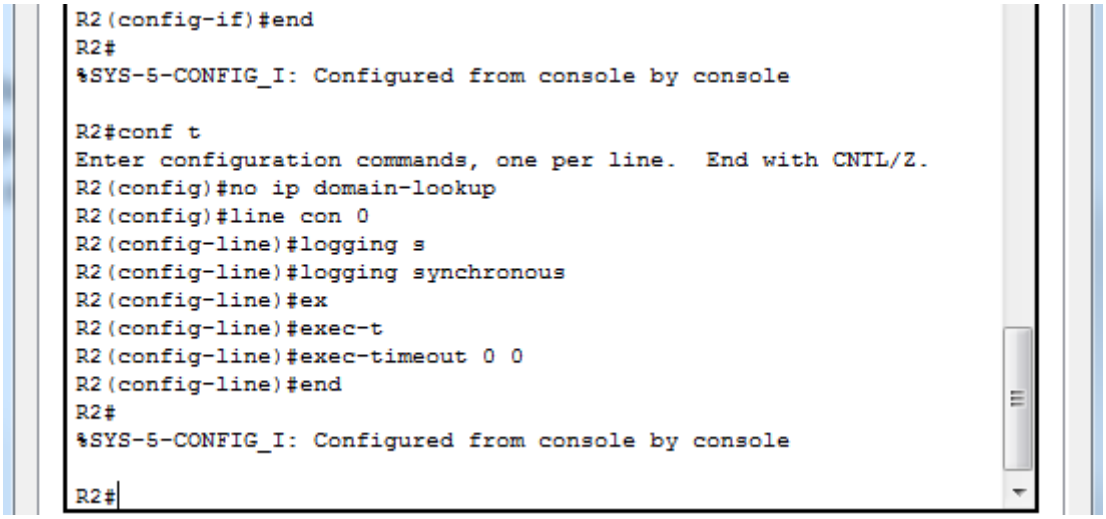


Figura 28. Configuración visualización comandos

Adicionalmente y para permitir un mejor desarrollo de las configuraciones habilitamos los comandos que podemos ver en la imagen superior, para evitar que nos desloggeemos del router por timeout y que además las líneas salgan independientes y visualizar mejor las configuraciones que vayamos realizando.

Al tratarse de una red de cliente, necesitamos que haya interconexión entre los distintos routers que la conforman. Para ello, escogemos el protocolo de routing interno EIGRP para establecer conexión entre todos los nodos de la red. Al igual que en el ejemplo anterior, mostraremos el ejemplo de configuración para un router y se aplicará igualmente en el resto. Lo único que tenemos que tener en cuenta son las redes que queremos publicar, que en nuestro caso serán la 172.16.0.0 y la 172.17.0.0.

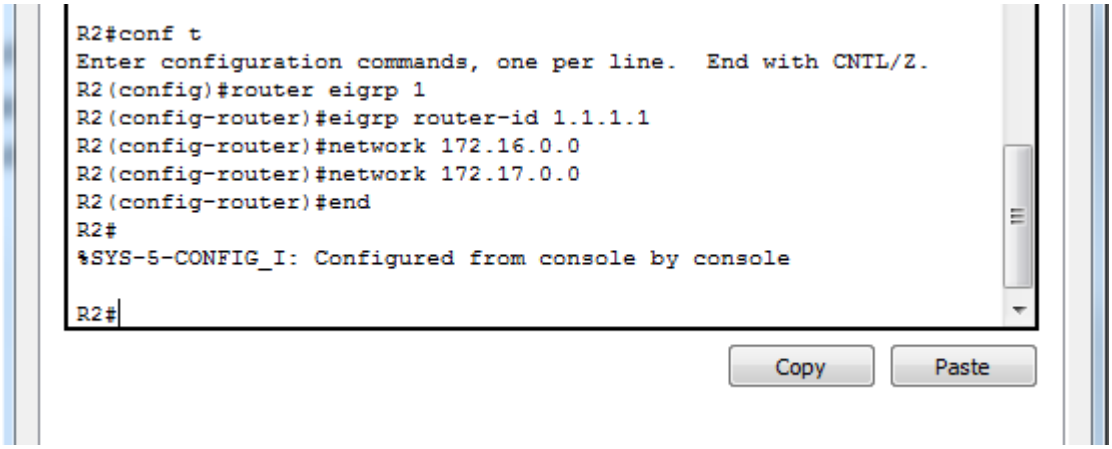


Figura 29. Configuracion EIGRP *"SanJose1"*

Para comprobar que efectivamente tenemos configurado correctamente el routing interno podemos hacer ping desde cualquiera de los routers a los interfaces loopback del resto.

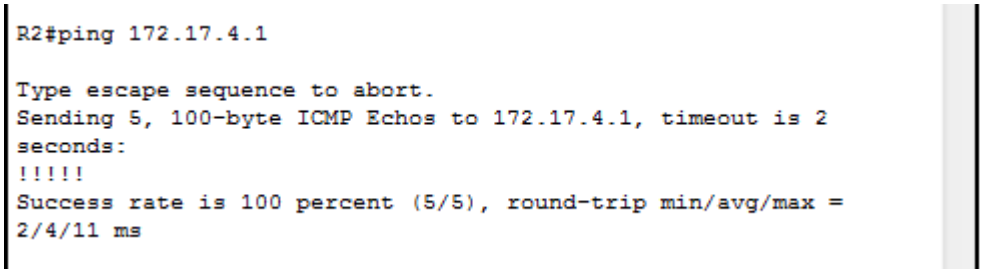


Figura 30. Prueba conectividad red *"SanJose3"*

A partir de este momento tendremos que configurar el protocolo BGP en nuestros equipos, puesto que ya tenemos enrutamiento interno; por lo que usaremos equipamiento real para el resto de la práctica.

En primer lugar, tendremos que configurar las sesiones eBGP, tanto desde el router SanJose1 como el de SanJose3 hacia los diferentes ISPs. Para nuestra práctica utilizaremos como salida real solo el ISP1; por lo que configuraremos este último para que establezca la sesión BGP con SanJose1. Estas configuraciones podemos observarlas en las siguientes imágenes.

```
*Jul 14 16:31:07.811: %LINK-3-UPDOWN: Interface Serial0/0/0, changed state to down
Router(config)#router bgp 65100
Router(config-router)#bgp router-id 1.0.0.1
Router(config-router)#neighbor 192.168.1.2 remote-as 65000
Router(config-router)#network 10.1.0.0 mask 255.255.0.0
Router(config-router)#exit
Router(config)#save runn
Router(config)#end
Router#save r
*Jul 14 16:32:21.663: %SYS-5-CONFIG_I: Configured from console by console
Router#save ru
Router#save run
Router#save runn
Router#copy run
Router#copy running-config st
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

Figura 31. Configuración BGP *"ISP1"*

```
SanJose1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SanJose1(config)#router bgp 65000
SanJose1(config-router)#bgp router-id 1.1.1.1
SanJose1(config-router)#neighbor 192.168.1.1 remote-as 65100
SanJose1(config-router)#neighbor
*Jul 14 17:05:43.651: %BGP-5-ADJCHANGE: neighbor 192.168.1.1 Up
SanJose1(config-router)#neighbor 172.17.4.1 remote-as 65000
SanJose1(config-router)#neighbor 172.17.4.1 update-source Loopback0
SanJose1(config-router)#network 172.16.2.0 mask 255.255.255.0
SanJose1(config-router)#exit
SanJose1(config)#
```

Figura 32. Configuracion BGP *"SanJose1"*

```
SanJose3(config-router)#exit
SanJose3(config)#router bgp 65000
SanJose3(config-router)#bgp router-id 3.3.3.3
SanJose3(config-router)#neighbor 172.17.2.1 remote-as 65000
SanJose3(config-router)#neighbor 172.17.2.1
*Jul 14 17:15:20.315: %BGP-5-ADJCHANGE: neighbor 172.17.2.1 Up
SanJose3(config-router)#neighbor 172.17.2.1 update-source Loopback0
SanJose3(config-router)#network 172.16.4.0 mask 255.255.255.0
SanJose3(config-router)#
```

Figura 33. Configuracion BGP *"SanJose3"*

Una vez configurado el BGP para aquellos routers que tenemos destinados a la conexión con los ISPs, podemos revisar mediante el comando *"show ip bgp"* la tabla BGP de los routers.

```
SanJose1#show ip bgp
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/16      192.168.1.1              0             0 65100 i
*> 172.16.2.0/24    0.0.0.0                  0         32768 i
r>i172.16.4.0/24    172.17.4.1               0    100      0 i
SanJose1#
```

Figura 34. Tabla BGP *"SanJose1"*

Como se observa en la tabla BGP, la última entrada de la tabla indica que existe un fallo de enrutamiento hacia la red 172.16.4.0 que es la perteneciente al router SanJose3. Este error puede "debuggearse" mediante el comando *"show ip bgp rib-failure"*, el cual, como se puede observar indica que hay una distancia administrativa mayor con respecto a otra fuente de enrutamiento.

Revisando la configuración, debemos de recordar que hemos implementado el protocolo EIGRP para el enrutamiento interno, y este tiene una distancia administrativa (90) menor que iBGP (200), por lo que todas aquellas rutas aprendidas por EIGRP van a ser escogidas por delante de aquellas publicadas mediante iBGP; podemos observarlo en la tabla de enrutamiento de *SanJose1*.

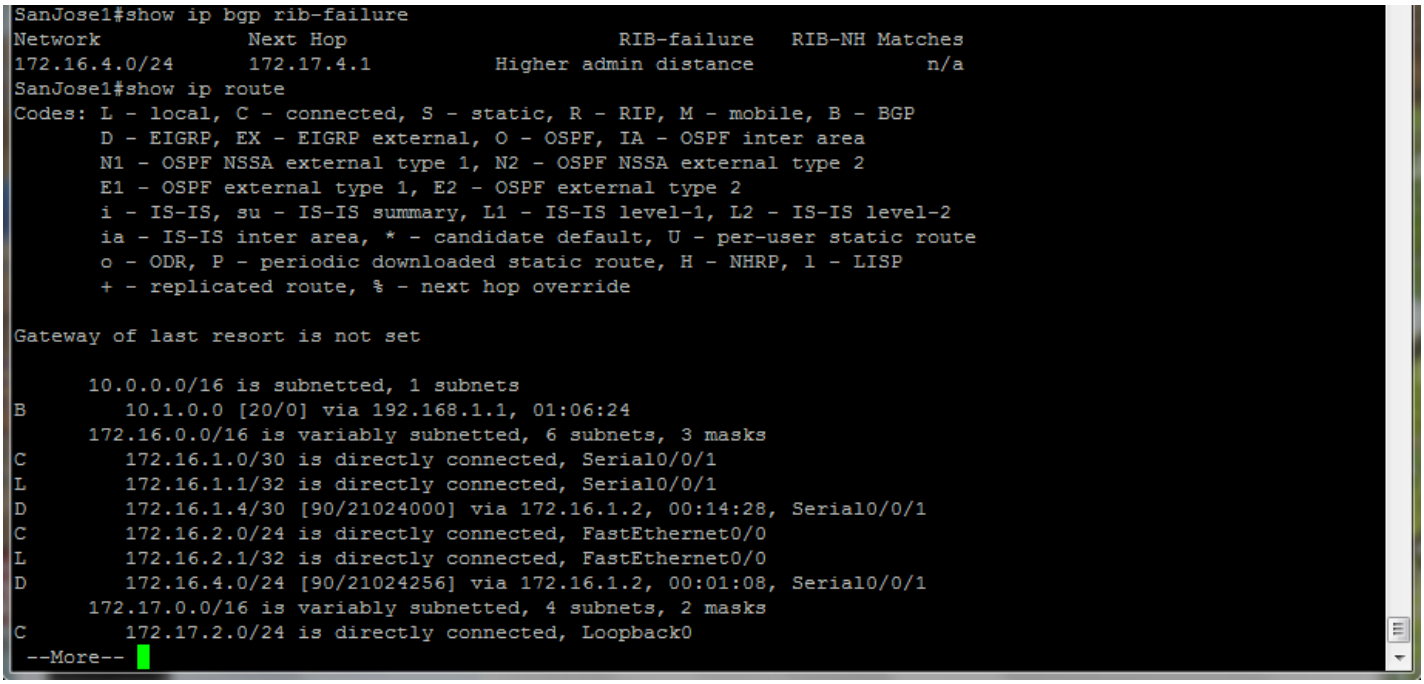


Figura 35. Tabla IP *"SanJose1"*

Nosotros, desde nuestra postura de administradores de la empresa, queremos que el enrutamiento se haga a través del protocolo iBGP, para ello necesitamos tener conectividad completa con los interfaces loopback de todos los routers. Debemos de tener todas estas rutas configuradas, porque tal y como podemos observar en la siguiente imagen, desde *SanJose3* no tenemos conectividad con el *ISP1*, esto lo deducimos porque tal y como se observa:

- No hay ninguna ruta en la tabla de enrutamiento IP hacia la 10.1.0.0/16

- En la tabla BGP la ruta que existe hacia esa red, no tiene el símbolo ">"; lo que indica que no se está ofreciendo esa ruta a la tabla IP. Ya que el siguiente salto es la 192.168.1.1 y como se observa en la tabla de enrutamiento no hay ninguna ruta hacia ese destino por lo cual no se pasará esa entrada de la tabla BGP a la IP

```
SanJose3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
D        172.16.1.0/30 [90/21024000] via 172.16.1.5, 00:10:22, Serial0/0/0
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.6/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/21026560] via 172.16.1.5, 00:06:26, Serial0/0/0
C        172.16.4.0/24 is directly connected, GigabitEthernet0/0
L        172.16.4.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/21152000] via 172.16.1.5, 00:10:23, Serial0/0/0
D        172.17.3.0/24 [90/2297856] via 172.16.1.5, 00:10:23, Serial0/0/0
C        172.17.4.0/24 is directly connected, Loopback0
L        172.17.4.1/32 is directly connected, Loopback0
SanJose3#show ip bgp
BGP table version is 5, local router ID is 3.3.3.3
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
* i10.1.0.0/16      192.168.1.1              0    100      0 65100 i
r>i172.16.2.0/24    172.17.2.1               0    100      0 i
*> 172.16.4.0/24    0.0.0.0                  0         32768 i
SanJose3#
```

Figura 36. Tabla IP *"SanJose3"*

Para solventar el problema, lo que haremos será configurar el router *SanJose1* para que use su interfaz Loopback como dirección de "next-hop" en sus actualizaciones iBGP. Esto lo hacemos con los comandos que se ven en la siguiente imagen.

```
SanJose1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SanJose1(config)#router bgp 65000
SanJose1(config-router)#neighbor 172.17.4.1 next-hop-self
```

Figura 37. Modificación actualizaciones iBGP *"SanJose1"*

Si volvemos a observar la tabla BGP veremos que efectivamente ahora si que es alcanzable.

```
*Jul 14 18:15:50.515: %SYS-5-CONFIG_I: Configured from console by console
SanJose1#show ip bgp
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/16      192.168.1.1              0             0 65100 i
*> 172.16.2.0/24    0.0.0.0                  0         32768 i
r>i172.16.4.0/24    172.17.4.1               0    100      0 i
SanJose1#
```

Figura 38. Tabla IP *"SanJose1"*

Por lo que obviamente, también tendremos una entrada en la tabla IP; como puede observarse en la imagen. Esa ruta tendrá como dirección de "next-hop" la interfaz de loopback del router SanJose1.

```
SanJose3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:00:59
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
D        172.16.1.0/30 [90/21024000] via 172.16.1.5, 00:01:28, Serial0/0/0
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.6/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/21026560] via 172.16.1.5, 00:01:28, Serial0/0/0
C        172.16.4.0/24 is directly connected, GigabitEthernet0/0
L        172.16.4.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/21152000] via 172.16.1.5, 00:01:29, Serial0/0/0
D        172.17.3.0/24 [90/2297856] via 172.16.1.5, 00:01:29, Serial0/0/0
C        172.17.4.0/24 is directly connected, Loopback0
L        172.17.4.1/32 is directly connected, Loopback0
SanJose3#
```

Sin embargo, si realizamos un ping desde la red perteneciente al router SanJose3 al ISP1 veremos que seguimos sin tener conexión. Por lo que podemos afirmar que el problema lo tenemos en el router SanJose2, en el cual no hemos tenido en cuenta que es el que debe publicar la red del ISP1 10.1.0.0/16 al router SanJose3; y no dispone de ninguna ruta para alcanzar al ISP1.

Esto es fácilmente comprobable, visualizando la tabla IP del router SanJose2 tal y como se ve en la siguiente imagen.

```
SanJose2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 8 subnets, 3 masks
C        172.16.1.0/30 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.5/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/20514560] via 172.16.1.1, 00:25:41, Serial0/0/1
C        172.16.3.0/24 is directly connected, FastEthernet0/0
L        172.16.3.1/32 is directly connected, FastEthernet0/0
D        172.16.4.0/24 [90/20512256] via 172.16.1.6, 00:05:32, Serial0/0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/20640000] via 172.16.1.1, 01:33:58, Serial0/0/1
C        172.17.3.0/24 is directly connected, Loopback0
L        172.17.3.1/32 is directly connected, Loopback0
D        172.17.4.0/24 [90/20640000] via 172.16.1.6, 00:05:32, Serial0/0/0
SanJose2#
```

Este caso, es muy común en las implementaciones de iBGP en redes del cliente; ya que el esquema de red debe estar completamente mallado de manera que existan conexiones entre todos y cada uno de los routers de la red. Para solucionar este problema, debemos establecer las configuraciones BGP en todos los routers del AS 65000 de manera que todos los routers establezcan las relaciones iBGP entre ellos.

Faltará pues:

- Que en SanJose2 establezcamos relación con SanJose1 y SanJose3.
- En SanJose3 establecer la relación con SanJose2.
- En SanJose1 establecer la relación con SanJose2.

Esto lo hacemos con los comandos que se muestran en las imágenes siguientes.

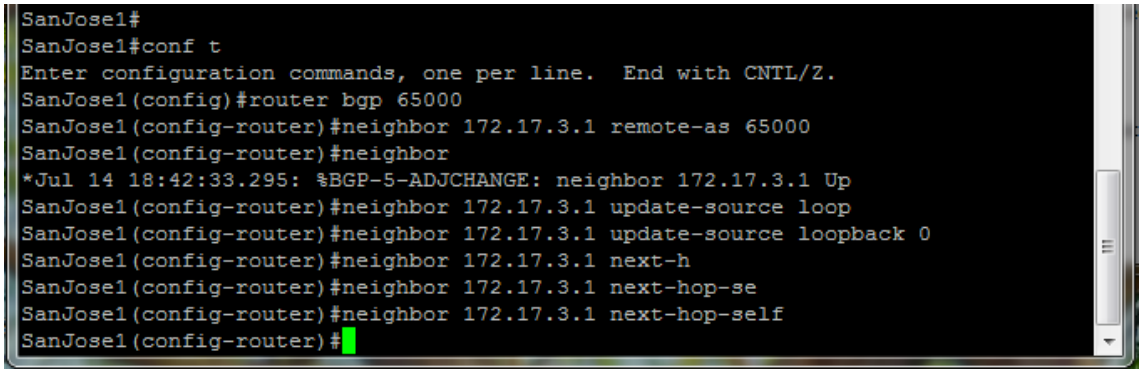


Figura 39. Configuracion BGP *"SanJose1"*

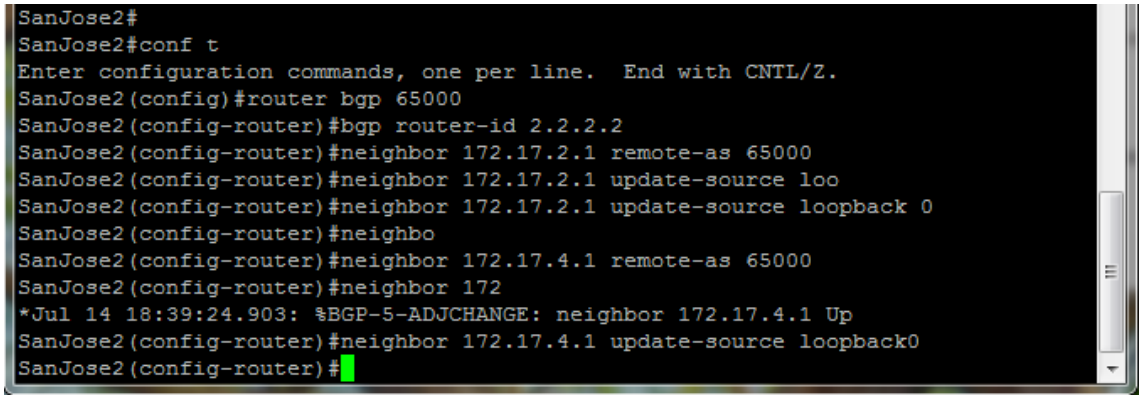


Figura 40. Configuracion BGP *"SanJose2"*

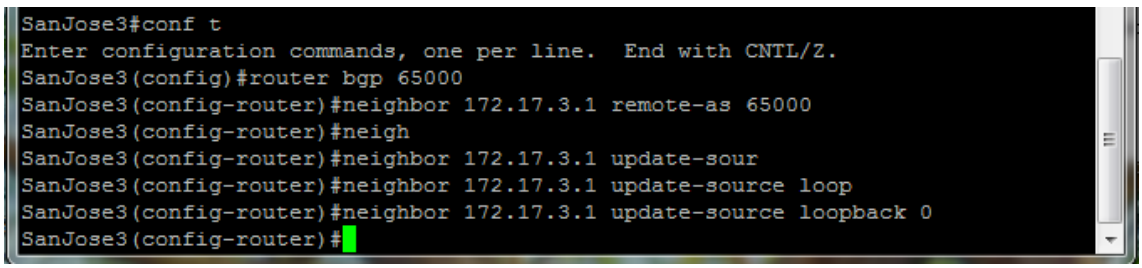


Figura 41. Configuracion BGP *"SanJose3"*

Si ahora examinamos de nuevo la tabla BGP e IP del router SanJose2 veremos que ya tenemos la ruta definida hacia el ISP1.

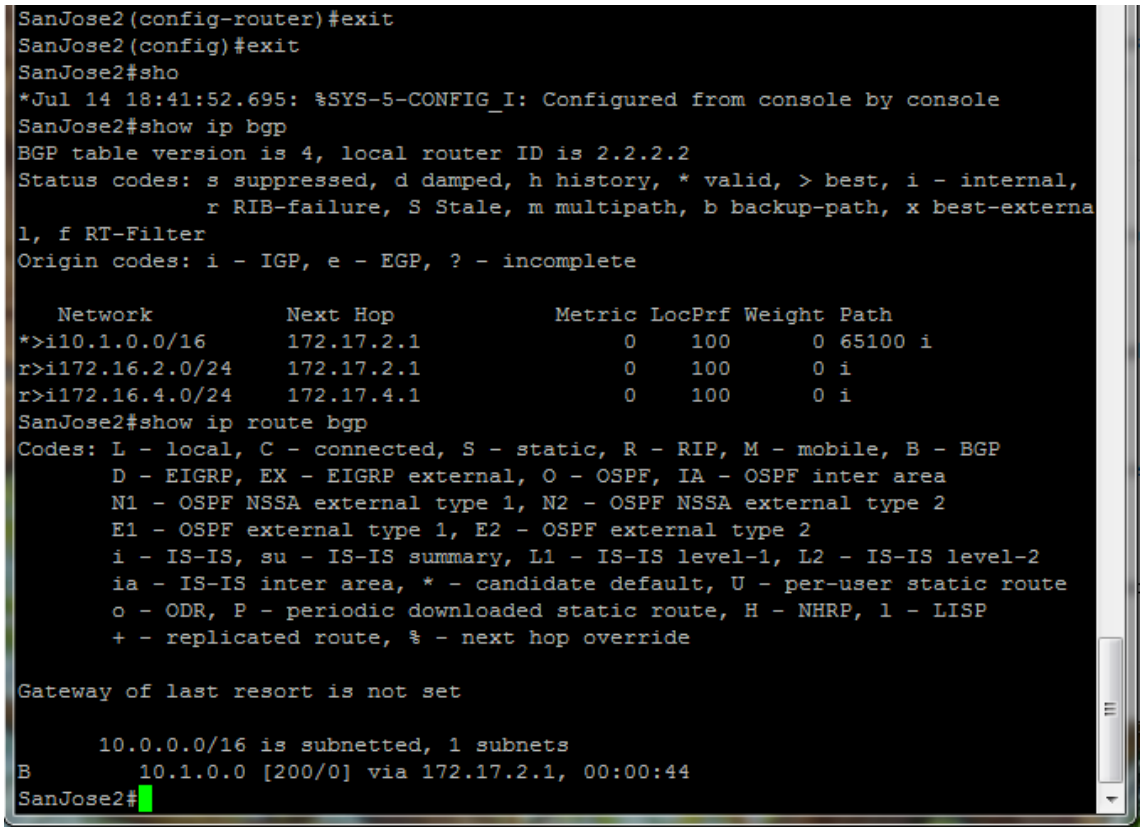


Figura 42. Configuracion BGP *"SanJose2"*

De esta forma, si realizaramos ahora un "Ping" desde la red de SanJose3 hacia el ISP1 veremos como ahora si que se realiza de forma correcta.

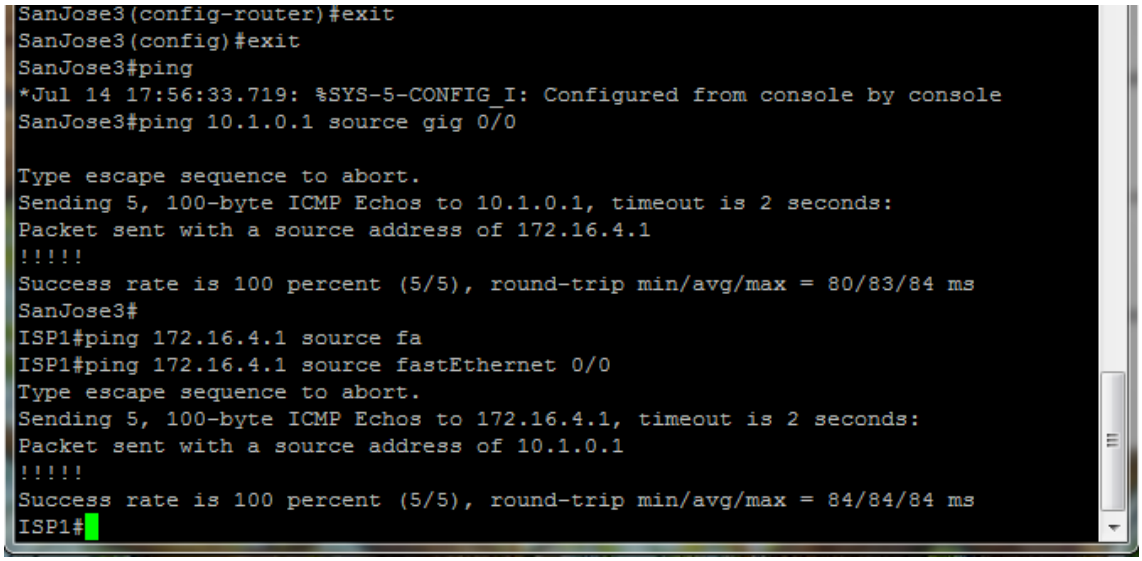


Figura 43. Ping exitoso al "*ISP1*"

Capítulo 9. Conclusiones

Como conclusión del proyecto podemos asumir que es completamente realizable el diseño, implementación y puesta en marcha de una red utilizando el protocolo BGP.

Durante la investigación realizada para recoger información, la parte que más me sorprendió fue la falta de seguridad del protocolo; resalto esta parte, ya que BGP es el encargado de las comunicaciones entre los grandes nodos del BGP actual, y que un protocolo de estas características tenga estos fallos me parecía una cosa impensable. Uno de los principios sobre los que se basa el protocolo, dice que *"se supone que todos los nodos conectados a una red BGP son fiables"*; en la época actual dudo mucho que si se creara un nuevo protocolo de enrutamiento tuviera estas bases debido a la más que creciente inseguridad de las comunicaciones en Internet.

Dejando esto a un lado, el profundizar en todos los procesos que lleva a cabo para la toma de decisiones de enrutamiento, proporciona una visión muy útil a la hora de realizar tareas de troubleshooting sobre una red de estas características.

En general, me ha parecido un trabajo muy interesante por conocer el protocolo que controla Internet en la actualidad, y como trabajo futuro me planteo la búsqueda de las alternativas, que, seguro que las hay, al protocolo BGP y que planteen menos problemas de seguridad.

Capítulo 10. Bibliografía

[1] BGP, “BGP Routing Table Analysis Reports”

http://bgp.potaroo.net/ [Online]

[2] Blog, “Análisis de BGP”

http://analisisbgp.blogspot.com.es/ [Online]

[3] Blog, “Ques BGP”

http://bgpmribera.blogspot.com.es/2012/02/bgp.html [Online]

[4] Cisco Blog, “Implementing a Border Gateway Protocol Solution for ISP Connectivity (Part01)”

http://ciscodocuments.blogspot.com.es/2011/05/chapter-06-implementing-border-gateway.html [Online]

[5] Un informático en el lado del mar, “IGP, EGP & BGP: Desconectar un país de Internet”

http://www.elladodelmal.com/2012/11/igp-egp-bgp-desconectar-una-red-de.html [Online]

[6] Genbeta, “Así se usó un fallo en el tejido de Internet para robar Bitcoins”

https://www.genbeta.com/seguridad/asi-se-uso-un-fallo-en-el-tejido-de-internet-para-robar-bitcoins [Online]

[7] Qore, “Descubren operación masiva de redirección dirigida de tráfico de Internet. Los atacantes explotaron una vulnerabilidad en el protocolo BGP”

http://www.qore.com/noticias/12351/Descubren-operacion-masiva-de-redireccion-dirigida-de-trafico-de-Internet [Online]

[8] La nación, “Internet pegó el estirón y ahora está en apuros”

http://www.lanacion.com.ar/1722869-internet-pego-el-estiron-y-ahora-esta-en-apuros [Online]