

## Módulo 3 – Filtrado de rutas en BGP y funcionalidades avanzadas

**Objetivo:** Utilizando la red configurada en el Módulo 2, utilizar varios métodos de configuración sobre peers BGP para demostrar filtrado con vecinos y funcionalidades más avanzadas de IOS.

**Prerequisitos:** Módulo 2

Topología:

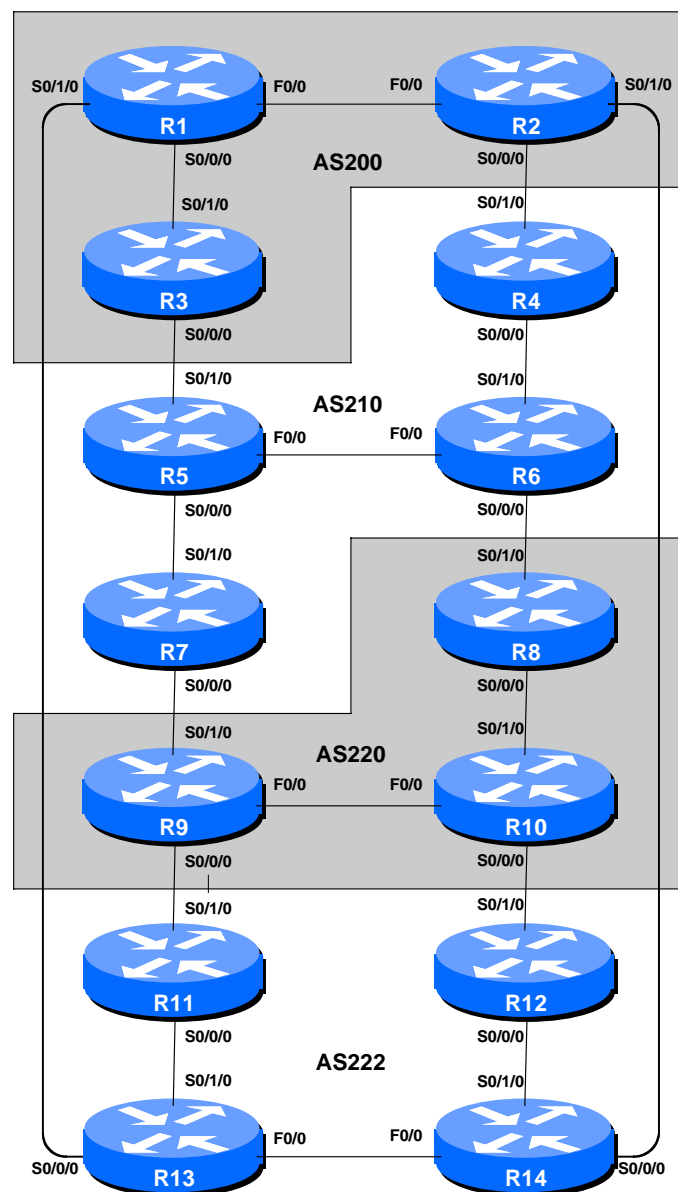


Figura 1 – Número de AS BGP

## Notas para el Laboratorio

El módulo anterior introdujo la configuración de BGP externo, pero no provee un método de controlar qué redes son anunciadas a algún AS. El propósito de este módulo es introducir al estudiante a los tipos de políticas de enrutamiento que están disponibles usando BGP.

En los pasos 1) al 6) queremos configurar a cada AS para que sean **AS sin-tránsito**, i.e. el AS no permitirá que tráfico de un AS conectado atravesase a este AS para llegar a otro AS. Esto implica pérdida de la conectividad general en la clase – por ejemplo, el SA220 no podrá ver las redes del SA200, etc. Esta es una política intencional para demostrar la efectividad del empleo de filtros en BGP.

En los pasos 1) y 2) vamos a configurar un filtro para rutas de salida que permita enviar solamente el(los) prefijo(s) local(es) a un peer eBGP. Al mismo tiempo, vamos a asegurar que los peers únicamente nos envían sus propios prefijos. Esto lo garantizamos al configurar un filtro de entrada. En general, es una buena práctica configurar filtros en ambas direcciones para proteger contra errores de configuración en cada extremo de los peers. En los pasos 3) al 6) vamos a usar comunidades en BGP para lograr el mismo efecto.

**Importante:** cada paso deberá ser elaborado y completado por el taller completo **antes** de comenzar el siguiente paso. No empiece de inmediato con el siguiente paso sin recibir el visto bueno de los instructores del workshop. Si lo hace, es probable que se rompa el enrutamiento, y es posible que los otros equipos no entiendan los resultados de la configuración que están tratando implantar.

**Importante:** mantenga la configuración usada en el Módulo 2, pero quite la línea de *aggregate-address* de la configuración de BGP.

## Ejercicio de Laboratorio

1. **Configure un filtro para prefijos basado en una dirección de red:** Este paso configura un filtro para prefijos basado en una dirección de red. Esto se hace usando prefix-lists, y es un método para controlar qué redes son intercambiadas en peers BGP. El propósito aquí es configurar los peers eBGP para que solamente se intercambien las redes de AS **vecino**.

### Ejemplo: Ruteador R13 (peer con R1)

```
!  
ip prefix-list out-peer permit 222.222.0.0/16 le 32  
ip prefix-list out-peer deny 0.0.0.0/0 le 32  
!  
ip prefix-list in-peer permit 200.200.0.0/16 le 32  
ip prefix-list in-peer deny 0.0.0.0/0 le 32  
!  
router bgp 222  
no synchronization
```

```

network 222.222.16.0 mask 255.255.252.0
neighbor 200.200.6.1 remote-as 200
neighbor 200.200.6.1 description eBGP peering with Router1
neighbor 200.200.6.1 prefix-list out-peer out
neighbor 200.200.6.1 prefix-list in-peer in
no auto-summary

```

### Ejemplo: Ruteador R1 (peer con R13)

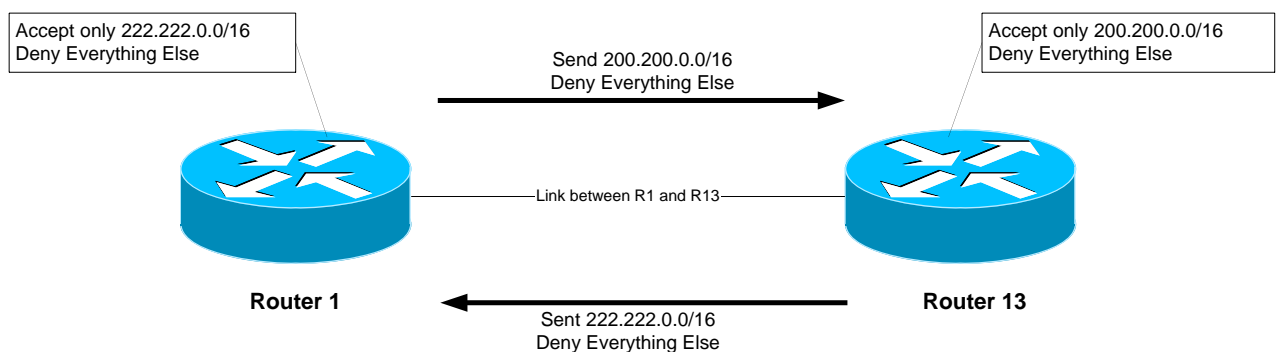
```

!
ip prefix-list out-peer permit 200.200.0.0/16 le 32
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 222.222.0.0/16 le 32
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
router bgp 200
no synchronization
network 200.200.4.0 mask 255.255.252.0
neighbor 200.200.6.2 remote-as 222
neighbor 200.200.6.2 description Peering with Router13
neighbor 200.200.6.2 prefix-list out-peer out
neighbor 200.200.6.2 prefix-list in-peer in
no auto-summary

```

**Nota:** en IOS un prefix-list siempre contiene una línea implícita *deny any* al final aunque no esté indicado en la configuración. Algunos ISPs agregan la línea implícita *deny any* porque lo consideran una buena práctica y una precaución para seguridad.

**Nota:** sólo se aplican estas listas prefix-list en las sesiones con otros ASs. Esto se llama **peering externo** (usando eBGP). Generalmente no se necesita aplicar filtros para peering iBGP.



Para implantar el filtro anterior en un peering, haz un *clear ip bgp <dirección del vecino>*.

**P:** ¿Por qué necesitamos el comando *clear ip bgp <dirección del vecino>*?

Monday, December 03, 2007

**R:** El proceso de BGP solamente anuncia cambios de la tabla de enrutamiento al vecino. Si una nueva red es agregada, será anunciada por BGP. Sin embargo, agregar un access-list al peer de BGP requiere que la tabla de enrutamiento anunciada actualmente pase por el access-list. Esto no es un proceso dinámico ya que la configuración con el peer ha cambiado, entonces hay que hacer un `clear ip bgp <dirección del vecino>` para reinicializar la sesión con el peer del vecino.

***Verificación #1:*** llame al asistente del laboratorio para verificar la conectividad. Cada equipo de ruteo debe verificar el peering para ver el efecto de este paso. Utilice el comando “`show ip bgp neigh x.x.x.x advertise/route`”. Una vez completada y el instructor del laboratorio indique que continúe, quite el prefix-list de la configuración y los access-list, y siga con el paso 2.

### Ahora para IPv6:

#### Ejemplo: Ruteador R13 (peer con R1)

```
!  
ipv6 prefix-list out-peer permit FEC0:222::/32 le 128  
ipv6 prefix-list out-peer deny ::/0 le 128  
!  
ipv6 prefix-list in-peer permit FEC0:200::/32 le 128  
ipv6 prefix-list in-peer deny ::/0 le 128  
!  
router bgp 222  
no synchronization  
address-family ipv6  
network FEC0:222:16::/48  
neighbor FEC0:200:4:6::1 remote-as 200  
neighbor FEC0:200:4:6::1 description eBGP peering with Router1  
neighbor FEC0:200:4:6::1 prefix-list out-peer out  
neighbor FEC0:200:4:6::1 prefix-list in-peer in  
no auto-summary
```

#### Ejemplo: Ruteador R1 (peer con R13)

```
!  
ipv6 prefix-list out-peer permit FEC0:200::/32 le 128  
ipv6 prefix-list out-peer deny ::/0 le 128  
!  
ipv6 prefix-list in-peer permit FEC0:222::/32 le 128  
ipv6 prefix-list in-peer deny ::/0 le 128  
!  
router bgp 200  
no synchronization  
address-family ipv6  
network FEC0:200:4::/48  
neighbor FEC0:200:4:6::2 remote-as 222  
neighbor FEC0:200:4:6::2 description Peering with Router13  
neighbor FEC0:200:4:6::2 prefix-list out-peer out  
neighbor FEC0:200:4:6::2 prefix-list in-peer in
```

```
no auto-summary
```

En este caso, para implantar el filtro, haga *clear bgp ipv6 <dirección del vecino>*.

Verifique las sesiones y las rutas anunciadas y recibidas haciendo “*show bgp ipv6 neigh x:x:x:x::x advertise/route*”

- 2. Quite la configuración del ejemplo anterior.** Este paso va a demostrar cómo recuperar la configuración que se introdujo en el ejemplo anterior. Es esencial hacer esto antes de seguir con el siguiente paso.

### Ejemplo: Router R1

```
router bgp 200
!
! Quite el prefix list del peering BGP con R13
!
no neighbor 200.200.6.2 prefix-list out-peer out
no neighbor 200.200.6.2 prefix-list in-peer in
address-family ipv6
  no neighbor FEC0:200:4:6::2 prefix-list out-peer out
  no neighbor FEC0:200:4:6::2 prefix-list in-peer in
exit
!
! Ahora quite los mismos prefix-list
!
no ip prefix-list out-peer
no ip prefix-list in-peer
no ipv6 prefix-list out-peer
no ipv6 prefix-list in-peer
!
! Ahora la configuración esta limpia, como debe de estar.
!
end
!
! Ahora hay que reinicializar el peer de bgp para quitar la política vieja
!
clear ip bgp 200.200.6.2
clear bgp ipv6 FEC0:200:4:6::2
```

- 3. Configure un filtro de prefijo basado en el atributo de AS path:** Este paso configura un filtro de prefijo basado en AS path. Esto se hace usando listas de acceso para AS, y es otro método para controlar qué redes son intercambiadas con peering BGP.

### Ejemplo en dirección de Salida – Ruteador R13

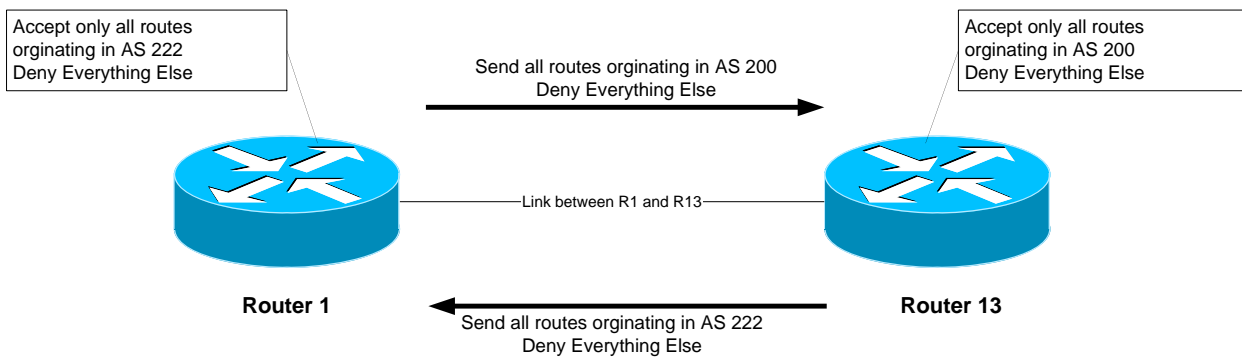
```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^200$
!
```

Monday, December 03, 2007

```
router bgp 222
 neighbor 200.200.6.1 remote-as 200
 neighbor 200.200.6.1 filter-list 2 out
 neighbor 200.200.6.1 filter-list 3 in
```

### Ejemplo en dirección de entrada – Ruteador R1

```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^222$
!
router bgp 200
 neighbor 200.200.6.2 remote-as 222
 neighbor 200.200.6.2 filter-list 2 out
 neighbor 200.200.6.2 filter-list 3 in
```



Para verificar que la expresión regular funciona como se espera, utilice el comando EXEC “*show ip bgp regexp <expresión-regular>*” para mostrar todos los path que resultan de la expresión regular especificada. No olvide que se requiere usar *clear ip bgp <dirección del vecino>* para implantar este filtro.

**Verificación #2 :** con el asistente de laboratorio verifique la conectividad. Cada equipo de ruteo deberá otra vez verificar su peer para ver el efecto en este momento. Una vez completada, mantenga la configuración de filter-list y siga con el siguiente paso.

Ahora haga lo mismo para Ipv6:

### Ejemplo en dirección de entrada – Ruteador R1

```
i
router bgp 200
 address-family ipv6
  neighbor FEC0:200:4:6::2 remote-as 222
  neighbor FEC0:200:4:6::2 filter-list 2 out
  neighbor FEC0:200:4:6::2 filter-list 3 in
end
```

Recuerde hacer también `clear bgp ipv6 <dirección>` para que el filtro haga efecto y `show bgp ipv6 regexp <expresión-regular>` y para las verificaciones.

#### 4. Configure el peer eBGP peer como peer con soft-reconfiguration

##### Ejemplo en Ruteador R1:

```
router bgp 200
 neighbor 200.200.6.2 soft-reconfiguration inbound
```

**P.** ¿Qué sucede al peer eBGP?

- Ahora, verifique los prefijos recibidos con “**show ip bgp**”. Ahora en los siguientes pasos en particular, recuerde verificar con “**show ip bgp <dirección>**” donde `<dirección>` es el prefijo se usa en un route-maps para cambiar atributos. Negocie con el AS vecino para que quiten los filtros en peering eBGP temporalmente. Observe la salida cuando se hace “**show ip bgp <red>**”. Tome nota de los comentarios *received-only* en la salida.

##### NOTA sobre soft-reconfiguration

Para permitir que una reconfiguración no quebranta el peer BGP, se necesita configurar soft-reconfiguration en el ruteador para mantener todos los prefijos que se recibe del peer. Los prefijos que son filtrados por cualquier método de entrada (**inbound**) en el ruteador de recepción son simplemente descartados y solamente los prefijos aceptados son almacenados en la tabla de BGP (vea esto con **show ip bgp**). Si hay algún cambio en el filtro, i.e. cambio a la política de AS, se requiere reinicializar el peer (i.e. **clear ip bgp <x.x.x.x>**) para que los prefijos se intercambien de nuevo cuando los dos enrutadores establecen su peering (y se usará el nuevo filtro sobre la actualización). Esto ayuda reducir el uso de memoria cuando se requiere filtros pesados pero hay un costo de reinicializar el peer BGP cada vez que se cambia la política de filtrado de entrada.

Dado que una reinicialización a BGP es una pérdida de conectividad, algunos clientes están dispuestos a gastar en más memoria para evitar esto. Por tanto, Cisco agregó una nueva funcionalidad llamado **soft-reconfiguration**. Con este comando, el ruteador BGP mantendrá los prefijos recibidos sin importar si son aceptadas o denegadas por los filtros de entrada. Cuando el administrador cambia la política de filtrado, todo lo que necesita hacer es ejecutar **clear ip bgp <x.x.x.x> soft in** en el ruteador y se vuelve a analizar su tabla y filtrar todos los prefijos en base a los nuevos filtros de entrada. Esto no causa una reinicialización del peer.

Note que no existe un comando **neighbor <x.x.x.x> soft-reconfiguration outbound**. Esto es porque si se cambia el filtro de **salida**, todo lo que requiere es que el ruteador envíe la actualización. Esto implica que el ruteador revise su tabla de BGP y cree la actualización para enviar hacia fuera. El ruteador no requiere mantener información adicional. Un comando **clear ip bgp <x.x.x.x> soft out** hará esto.

Monday, December 03, 2007

Con la funcionalidad de soft-reconfiguration activada, un comando ***show ip bgp <n.n.n.n>*** mostrará todos los paths que han sido recibidos e indica si han sido aceptadas o no. Un camino (path) que ha sido aceptado en cambio de atributos (de un filtro de entrada) será marcado como **received & used**. Un path que es denegado o tiene un atributo cambiado es marcado como **received-only**.

Para IPv6, los comandos equivalentes son:

```
router bgp 200
  address-family ipv6
  neighbor FEC0:200:4:6::2 soft-reconfiguration inbound
```

y

```
show bgp ipv6
show bgp ipv6 <prefijo>
clear bgp ipv6 <vecino> soft in
```

**Verificación #3:** llame al asistente del laboratorio para verificar la conectividad. Una vez que el instructor del laboratorio indique que continúe, quite la configuración de atributo y filter list, y continúe con el siguiente paso.

**6. Quite la configuración del ejemplo anterior.** Este paso demostrará cómo quitar la configuración introducida en el ejemplo anterior. Es esencial antes de continuar con el siguiente paso.

### **Ejemplo: Ruteador 1**

```
router bgp 200
!
! Primero quite el filter list del peering BGP con R13
!
no neighbor 200.200.6.2 filter-list 2 out
no neighbor 200.200.6.2 filter-list 3 in
no neighbor FEC0:200:4:6::2 filter-list 2 out
no neighbor FEC0:200:4:6::2 filter-list 3 in
!
! Ahora quite las listas de filtros
!
no ip as-path access-list 2
no ip as-path access-list 3
!
! Ahora la configuración está limpia, como de ser
!
end
!
! Ahora haga un clear del peering bgp para quitar la política anterior
!
clear ip bgp 200.200.6.2 soft
```

```
clear bgp ipv6 FEC0:200:4:6::2 soft
```

- 7. Introducción a route-maps:** En todos los ruteadores, configure BGP para que envíe comunidades para *todos* los prefijos que pertenecen al AS local anunciados a peers externos BGP. La comunidad debe ser en la forma *[número AS]:[número Ruteador]*. Por ejemplo, en el ruteador R9 deberá usarse la comunidad 220:9.

### Ejemplo en Ruteador R9:

```
ip bgp-community new-format
! se necesita para tratar a las comunidades en el formato 16-bit:16-bit
! en vez de un entero de 32 bits.
!
ip prefix-list out-match permit 220.220.0.0/16 le 32
!
route-map outfilter permit 10
  match ip address prefix-list out-match
  set community 220:9
!
router bgp 220
  neighbor 220.220.9.2 remote-as 210
  neighbor 220.220.9.2 route-map outfilter out
  neighbor 220.220.9.2 send-community
```

### Nota:

- 1) Un atributo de comunidad se puede ver en la tabla de BGP vía el comando *show ip bgp <network>*.
- 2) Un atributo de comunidad es un campo de 32 bits. Por convención del IETF éste se divide en dos campos de 16-bits para una interpretación mas fácil. Los primeros 16 bits contienen el número del AS, los últimos 16 bits representan un entero que tiene un significado específico entre los dos ASs con los que se hace el peering. La excepción a esto es el uso de las atributos de comunidades bien conocidas como *no-export* o *local-as*.

**P:** ¿Por qué se necesita *send-community* para los peering de eBGP?

**R:** Como los valores de comunidades no son pasados entre dos peers BGP por defecto, se necesita indicar al ruteador hacer esto explícitamente.

Haga algo similar para IPv6:

```
ipv6 prefix-list out-match permit FEC0:220::/32 le 128
!
route-map outfilter permit 10
  match ip address prefix-list out-match
  set community 220:9
!
router bgp 220
  address-family ipv6
    neighbor FEC0:220:32:9::2 remote-as 210
```

Monday, December 03, 2007

```
neighbor FEC0:220:32:9::2 activate
neighbor FEC0:220:32:9::2 route-map outfilter out
neighbor FEC0:220:32:9::2 send-community
```

***Verificación #4:*** llame al asistente del laboratorio para verificar la conectividad. Cada de equipo de ruteo deberá verificar los peerings para ver que efecto tiene esta vez.

**8. Quite la configuración del ejemplo anterior.** En este paso se demostrará como quitar la configuración introducida en el ejemplo anterior. Es esencial hacer esto antes de seguir con el siguiente paso.

### **Ejemplo: Ruteador 9**

```
Router bgp 220
!
! Primero quite el route-map del peering eBGP
!
no neighbor 220.220.9.2 route-map outfilter out
address-family ipv6
  no neighbor FEC0:220:32:9::2 route-map outfilter out
!
! Ahora quite el route-map
!
no route-map outfilter
!
! Ahora quite el prefix-list usado por el route-map
!
no ip prefix-list out-match
no ipv6 prefix-list out-match
!
! Ahora la configuración esta limpia, como debe ser
!
end
!
! Ahora haz un clear del peering bgp para quitar la política anterior
!
clear ip bgp 220.220.9.2 soft
clear bgp ipv6 FEC0:220:32:9::2 soft
```

**9. Configurar Comunidades en BGP.** En el paso 7 la comunidad a la que pertenece una red fue generada en el punto en que un router habla BGP con otro router hablando BGP. Mientras esta situación es útil para demostrar como fijar comunidades, el escenario mas común es donde un ISP fija una comunidad a la red cuando la red es inyectada a la tabla de ruteo BGP.

Cada equipo de ruteo debe asignar una comunidad al block de redes /22 que han sido asignados en el Módulo 1. Revise la documentación de BGP para encontrar cómo hacer esto. Cada ruteador

debe fijar la comunidad en el formato *[número AS]:[número Ruteador]* exactamente como en el paso anterior.

### Ejemplo para el Ruteador R1:

```
ip bgp-community new-format
!
route-map community-tag permit 10
  set community 200:1
!
router bgp 200
  no synchronization
  network 200.200.4.0 mask 255.255.252.0 route-map community-tag
  neighbor 200.200.6.2 remote-as 222
  neighbor 200.200.6.2 send-community
  no auto-summary
!
ip route 200.200.4.0 255.255.252.0 null0
```

Revise que la red aparece con su comunidad en la tabla de ruteo BGP.

**P:** ¿Por qué los peers externos, pero no internos, ven la comunidad que ha sido asignada?

**R:** Vea anteriormente. Cada peering requiere usar el subcomando “send-community” para que el atributo de comunidad sea enviado entre peers BGP.

Ahora para IPv6 nos falta:

```
router bgp 200
  neighbor FEC0:200:4:6::2 remote-as 222
  address-family ipv6
    neighbor FEC0:200:4:6::2 activate
    network FEC0:200:4::/48 route-map community-tag
    neighbor FEC0:200:4:6::2 send-community
  !
  ipv6 route FEC0:200:4::/48 null0
```

**10. Comunidades en peering interno BGP.** Siguiendo con el paso anterior, ahora configure peering interno de tal manera que el atributo de comunidad para su red sea enviado a peers locales.

**Sugerencia:** para hacer esto, simplemente agregue en la línea de configuración “neighbor x.x.x.x send-community” para todos los peers iBGP. No olvide hacer clear de los peers BGP de tal manera que el cambio de configuración sea implantada.

**Verificación #5:** llame al asistente de laboratorio y demuestre como la comunidad ha sido fijada en su red usando los comandos “show ip bgp”. También, demuestre que puede ver las comunidades fijadas por los peers internos y externos BGP.

- 11. Configure un prefix-filter de entrada basado en el atributo de comunidad.** El objetivo aquí es sólo aceptar redes que son recibidas del peer vecino de BGP externo. (Esto es similar a lo que se intentó en los pasos 1 y 3 con filtrado usando prefix y AS path.) Por ejemplo, R13 sólo debe aceptar la red originada por R1, y debe usar la información de la comunidad que R1 ha agregado al prefijo de red para lograr esto.

### Ejemplo en Ruteador R13:

```
route-map infilter permit 10
  match community <numero-lista-comunidad>
  !
ip community-list <numero-lista-comunidad> permit 200:1
!
router bgp 222
  neighbor 200.200.6.1 route-map infilter in
```

Similarmente, para IPv6:

```
!
router bgp 222
  address-family ipv6
    neighbor FEC0:200:4:6::1 route-map infilter in
```

El <numero-lista-comunidad> es una opción para cada equipo de ruteo – no es anunciado en algún peering BGP o utilizado en alguna forma aparte de identificar la lista de comunidad (compare con el número de lista de acceso).

- 12. Fije el atributo local-preference en las rutas recibidas por eBGP.** En este ejemplo, se fijará “local-preference” en las rutas seleccionadas por el filtro para comunidades en el paso 10. Mantenga el route-map utilizado en el paso 10 – una línea adicional será agregada. Deberá permitir que las otras redes sean escuchadas a través de los filtros con local-preference en el valor por defecto.

**P.** ¿Por qué?

**R.** Sin el segundo directivo de permit el route-map implementa un deny por omisión y ningún prefijo será permitido.

### Ejemplo:

```
route-map infilter permit 10
  set local-preference 120
!
route-map infilter permit 20
```

Note que después de configurar una nueva política, la sesión de BGP requiere ser reinicializada para que el peer reenvie el prefijo y la nueva política puede ser aplicada. El ruteador no mantiene automáticamente todas las actualizaciones que son recibidas del peer y por tanto es necesario. Esto se hace usando el comando “*clear ip bgp <dirección del peer>*” ( y *clear bgp ipv6 <dirección del peer>* )

- 13. Revise la operación de soft-reconfiguration.** Compare la salida entre “*show ip bgp neigh x.x.x.x received-routes*” y “*show ip bgp neigh x.x.x.x route*” (y “*show bgp ipv6 neigh x:x:x:x::x received-routes*” y “*show bgp ipv6 neigh x:x:x:x::x routes*”)

**Revisión #6:** llame al asistente de laboratorio y muestre cómo las rutas originadas por los peers eBGP ahora tienen asignado un local preference de 120. También muestre cómo las otras rutas tienen un valor de omisión de 100.

- 14. Configure la funcionalidad peer-group para peers iBGP.** Peer-groups en BGP ayudan a reducir la carga al procesador del ruteador en enviar las actualizaciones a los peers con la misma política. Este paso configura peer-groups en BGP para los peers iBGP para cada AS. Reemplace la configuración individual para cada peer iBGP con la configuración de peer-group, como se muestra en el ejemplo a continuación.

#### Ejemplo para el ruteador R9:

```
router bgp 220
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers description Peer-Group para todos los peers iBGP
 neighbor ibgp-peers remote-as 220
 neighbor ibgp-peers update-source loopback 0
 neighbor ibgp-peers send-community
 neighbor 220.220.7.224 peer-group ibgp-peers
 neighbor 220.220.19.224 peer-group ibgp-peers
```

De forma similar, para IPv6 (los miembros de un peer-group deben ser de la misma familia de direcciones, por lo que hay que crear un grupo aparte):

```
router bgp 220
 address-family ipv6
 neighbor ibgp-v6-peers peer-group
 neighbor ibgp-v6-peers description Peer-Group para todos los peers v6 iBGP
 neighbor ibgp-v6-peers remote-as 220
 neighbor ibgp-v6-peers update-source loopback 0
 neighbor ibgp-v6-peers send-community
 neighbor FEC0:220:4:7::1 peer-group ibgp-v6-peers
 neighbor FEC0:220:16:19::1 peer-group ibgp-v6-peers
```

**P:** ¿Cuáles son las ventajas de usar *peer-groups*?

Monday, December 03, 2007

**R:** Los peer groups en BGP permiten que una configuración común sea utilizada para varios peers de BGP. La aplicación mas común es para iBGP. Todos los peer internos de BGP en un ISP tienden a tener la misma relación entre ellos. En vez de tener una configuración en cada peer, y tener que cambiar cada peer cuando los detalles requieren un cambio, la configuración puede utilizar un peer-group, y solo el peer-group requiere ser cambiada para alterar la configuración para todos los peers iBGP. Esto reduce sustancialmente el trabajo requerido en hacer cambios, el procesamiento de CPU del ruteador, y limpia significativamente la configuración para su visualización.

Es recomendable que el comando peer-group sea el método por “omisión” para la configuración de todos los peers BGP. Como mencionamos anteriormente, los peers iBGP tienen la misma configuración, y es un beneficio para el ruteador, el personal de operación y el personal de ingeniería de red para simplificar la configuración donde sea posible. Una configuración que hace uso extensivo de peer-groups es mucho más fácil de leer que una que usa configuraciones distintas por cada peer, especialmente en redes con un número grande de peers.

## 15. Configure la funcionalidad de “route-dampening” (humedecimiento de rutas) de BGP.

### Ejemplo:

```
router bgp 200
  bgp dampening
```

Hable con otros ASs y trate de provocar un “flap” de red quitando una red de un bloque /22 de la configuración BGP (network x.x.x.x). Los vecinos eBGP deben ver que el prefijo es quitado y recibe una penalización de 1000 debido al RETIRO. Ahora pida al equipo que reinstale la ruta en BGP. Cuando reaparece en su tabla de BGP pida que lo quiten otra vez. Repita lo mismo y hasta obtener 3 “flaps” verá que el prefijo es “dampened”. Ahora verifique con *show ip bgp flap* para ver la penalización asignada a los prefijos.

Después de un “flap” debe ver algo similar a en la salida de *sh ip bgp <prefijo>* - este es tomado de un lab de un ISP:

```
alpha#
BGP: charge penalty for 158.43.0.0/16 path 2830 with halflife-time 30
reuse/suppress 750/3000
      flapped 1 times since 00:00:00. New penalty is 1000
alpha#sh ip bgp 158.43.0.0
BGP routing table entry for 158.43.0.0/16, version 79
Paths: (1 available, no best path)
  Not advertised to any peer
    2830 (history entry)
      192.168.4.130 from 192.168.4.130 (192.168.9.13)
        Origin IGP, metric 0, localpref 100, external
        Dampinfo: penalty 992, flapped 1 times in 00:00:22

alpha#sh ip bgp 158.43.0.0
```

```
BGP routing table entry for 158.43.0.0/16, version 79
Paths: (1 available, no best path)
  Not advertised to any peer
  2830 (history entry)
    192.168.4.130 from 192.168.4.130 (192.168.9.13)
      Origin IGP, metric 0, localpref 100, external
      Dampinfo: penalty 984, flapped 1 times in 00:00:43
```

**P.** ¿Qué significa la entrada de “history”?

**R.** La entrada de “history” significa que el vecino eBGP ha quitado el prefijo y hay una penalización por “flap” asignado a él. El prefijo no está disponible para ser enviado a otros peers BGP, y por tanto se despliega *no best path*. Cuando el prefijo es reanunciado por el vecino eBGP, la entrada cambia de “history” a ser disponible otra vez, como se muestra:

```
alpha#sh ip bgp 158.43.0.0
BGP routing table entry for 158.43.0.0/16, version 80
Paths: (1 available, best #1)
  Advertised to non peer-group peers:
    192.168.12.1 192.168.18.2 192.168.20.1
  2830
    192.168.4.130 from 192.168.4.130 (192.168.9.13)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Dampinfo: penalty 978, flapped 1 times in 00:01:02
alpha#
```

Los valores por omisión de Cisco son **bgp dampening 15 750 2000 60**. Cada “flap” acumula una penalización fija de 1000.

Las opciones para el comando *bgp dampening* son:

- 15 – media-vida (en minutos)
- 750 – límite de reutilización (valor de penalización en la que la ruta es reutilizada)
- 2000 – suprima en (valor de penalización en la que la ruta es suprimida)
- 60 – límite de supresión (tiempo máximo en minutos donde la ruta será suprimida)

La penalización degrada con una granularidad de cada 5 segundos. Entonces cada 5 segundos la penalización se reduce de acuerdo al tiempo de media-vida (degradación exponencial). Un “flap” acumula una penalización de 1000 unidades. Entonces en este ejemplo, la ruta es suprimida la primera vez que hace un “flap”. Una vez que 800 unidades han sido removidas del “flap”, la ruta es reanunciada. Una vez que la penalización cae por debajo de la mitad del límite de reutilización, el “flap” es quitado de la tabla de “flap” – el siguiente “flap” empieza a contar desde cero otra vez.

## 16. BGP Dampening controlado:

Hacer “dampening” en BGP en forma controlada es una práctica cada vez más común hoy en el Internet. Por ejemplo, una recomendación del grupo de trabajo de ruteo de RIPE, es que los

Monday, December 03, 2007

prefijos /24 acumulen diferente tasa de dampening que los /22 y /23 y el resto de prefijos de otro tamaño.

Trate de utilizar un route-map para el control más preciso de dampening. Este ejemplo “humedece” las rutas listadas en el prefix-list *damp-prefix*.

```
ip prefix-list damp-prefix permit x.x.x.x/m
!
route-map damp-some permit 10
  match ip address prefix-list damp-prefix
  set dampening 15 750 2000 60
!
router bgp 200
  bgp dampening route-map damp-some
```

y para IPv6:

```
ipv6 prefix-list damp-v6-prefix permit x:x:x:x::x/x
!
route-map damp-some permit 20
  match ipv6 address prefix-list damp-v6-prefix
  set dampening 15 750 2000 60
!
```

***Verificación #7:*** llame al asistente de laboratorio para verificación la operación de dampening y su configuración.

**17. Resumen:** Este modulo ha introducido algunas funcionalidades básicas disponibles para configurar peerings BGP con IOS de Cisco. Se recomienda al lector probar otras permutaciones de los ejemplos dados aquí. El uso de comunidades está ganando popularidad como una funcionalidad que da ventajas considerables para controlar la política de ruteo entre diferentes ASs. Dampening, soft reconfiguration, y peer-groups en BGP son usados en redes de ISPs y considerablemente facilitan la administración y configuración de la red en operación. Para el uso recomendado de parámetros de operación de flap dampening en BGP para sitios conectados al Internet, se sugiere consultar a <http://www.ripe.net/docs/ripe-229.html> y el documento de Cisco *IOS Essentials* en <ftp://ftp-eng.cisco.com/cons/isp>. Para mas información sobre los algoritmos detrás de la implantación de route-dampening, el lector puede consultar el RFC2439.

***Preguntas para revisión:***

## **Notas de Configuración**

¡La documentación es crítica! Deberá hacer un registro de la configuración en cada **Revisión**, así como la configuración al final del módulo.