Software de Encaminamiento Quagga v0.96.1 - Manual

Quagga es un paquete de software de encaminamiento avanzado que proporciona los protocolos de encaminamiento basados en TCP/IP. Este es el Manual para quagga-0.96.1 Quagga es una bifurcación o fork de GNU Zebra. Esta documentación ha sido traducida al Español por Eduardo Collado [*edu@eduangi.com*] y Mariano Juliá [*mjuliaq@guaca.net*].

Contents

1 Introducción

Quagga es un paquete de software de encaminamiento que proporciona encaminamiento basado en servicios de TCP/IP con protocolos de encaminamiento que soportan RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4 y BGP-4+.Quagga también soporta el comportamiento especial de BGP Route Reflector y Route Server. Además de los protocolos de encaminamiento tradicionales basados en IPv4, Quagga tambien soporta protocolos de encaminamiento basados en IPv6. El demonio SNMP es soportado por el protocolo SMUX, Quagga proporciona también las MIBs correspondientes.

Quagga utiliza una arquitectura de software avanzada para proporcionar una gran calidad, con un motor multi servidor de encaminamiento. Quagga tiene un interfaz de usuario interactivo para cada protocolo de routing y soporta a también soporta protocolos de encaminamiento basados en IPv6. El demonio de SNMP

comandos de cliente en sus interfaces. Debido a su diseo es posible añadir nuevos demonios de protocolos fácilmente a Zebra. Zebra se puede también utilizar como librería para un programa cliente de interfaz de usuario.

Zebra es un software oficial GNU y está distribuido bajo la Licencia General Pública GNU .

1.1 Sobre Quagga

Hoy en día, las redes TCP/IP están convergiendo todas ellas en todo el Mundo. Internet ha sido desarrollado en muchos países, entornos empresariales y en entornos domésticos. Cuando un usuario se conecta a Internet sus paquetes atravesarán muchos routers que utilicen la funcionalidad del routing TCP/IP.

Un sistema con Quagga instalado actúa como router dedicado. Con Quagga, una máquina intercambia información de routing con otros routers utilizando protocolos de routing. Quagga utiliza esa información para actualizar el núcleo de las tablas de routing de forma que la información correcta esté en el lugar correcto. Quagga permite la configuración dinámica y es posible ver la información de la tabla de routing desde el interfaz de terminal de Quagga.

Añadiendo soporte al protocolo de routing, Quagga puede configurar las banderas (flags) de los interfaces, direcciones de los interfaces, rutas estáticas y muchas más cosas. Si se utiliza en una red pequeña o en una conexión xDSL, la configuración del software Quagga es muy sencilla. Lo único que hay que pensar es en levantar los interfaces e introducir unos pocos comandos sobre rutas estáticas y/o rutas por defecto. Si en cambio estamos utilizando una red más grande, o la estructura de la red cambia frecuentemente, entonces utilizaremos la ventaja que nos ofrece Quagga sobre los protocolos de routing dinámicos, soportando protocolos como RIP, OSPF, o BGP.

Tradicionalmente, la configuración de un router basado en UNIX se realizaba mediante los comandos *ifconfig* y los comandos del tipo *route*. El estado de las tablas se podía mostrar mediante la utilidad *netstat*. Estos comandos solamente se podían utilizar trabajando como root. Quagga, sin embargo tiene otro método de administración. En Quagga existen dos modos de usuario. Uno es el modo normal y el otro es el modo de enable (habilitado). El usuario de modo normal únicamente puede ver el estado del sistema, sin embargo el usuario de modo enable puede cambiar la configuración del sistema, Esta cuenta independiente de UNIX puede ser de gran ayuda para el administrador del router.

Actualmente, Zebra soporte los protocolos de unicast más comunes. Los protocolos de routing Multicast como BGMP, PIM-SM, PIM-DM serán soportados en Quagga 2.0. El soporte de MPLS está siendo programado actualmente. En el futuro, control de filtros TCP/IP, control de calidades de servicio QoS, la configuración de diffserv será añadida a Zebra. El objetivo de Zebra es conseguir un software de routing productivo de calidad y libre.

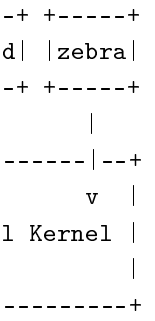
1.2 Arquitectura del Sistema

El software tradicional de routing esta compuesto por un programa o proceso único que proporciona todas las funcionalidades de los protocolos de routing. Quagga sin embargo tiene una visión distinta. Está compuesto por una colección de varios demonios que trabajan juntos para construir una tabla. Hay vario demonios de routing específicos que se ejecutan junto con e zebra, el kernel gestor del routing.

El demonio *ripd* maneja el protocolo RIP, mientras que el demonio *ospfd* controla el protocolo OSPFv2. *bgpd* soporta el protocolo BGP-4. Para cambiar la tabla de routing del kernel y la redistribución de rutas entre distintos protocolos de routing tenemos la table de routing del kernel controlada por el demonio *zebra*.

Es sencillo añadir nuevos demonios de protocolos de routing el sistema global de routing sin afectar a otro software. Para ello hay sólo es necesario ejecutar los demonios asociados a los protocolos de routing a utilizar. Realizando esta operación, el usuario puede ejecutar un determinado demonio y enviar reportes a la consola central de routing.

No es necesario ejecutar esos demonios en la misma máquina. Es posible ejecutar varias instancias del mismo demonio de routing en la misma máquina. Esta arquitectura crea nuevas posibilidades para el sistema de routing.



Arquitectura del sistema Quagga

La arquitectura multiproceso nos permite un sistema más fácilmente extensible y gestionado y por supuesto nos permite un sistema totalmente modular, a la vez que nos permite varios ficheros de configuración e interfaz de terminal.

Ya que cada demonio tiene su propio fichero de configuración e interfaz de terminal, cuando se quiere configurar una ruta estática, esta se configura en el fichero de configuración *zebra*. Cuando se configura una red BGP hay que hacerlo en el fichero de configuración *bgpd*, esto es bastante fastidioso. Para solucionar este problema existe un interfaz shell integrado llamado *vtysh*. *vtysh* conecta cada demonio funcionando como un proxy para la entrada del usuario.

Quagga se planteó para ser utilizado con mecanismos de multi-hilos (multi-threaded) cuando se ejecutase en un kernel que lo soportara. Pero en este momento, la librería de hilos que viene con GNU/Linux o FreeBSD tiene varios fallos para ejecutar servicios fiables como un software de routing, así que de momento las versiones de Quagga no utilizan hilos, en vez de utilizar hilos Zebra utiliza la llamada al sistema *select(2)* para multiplexar los eventos.

Cuando *zebra* se ejecute bajo el kernel GNU/Hurd actuará como tabla de routing del kernel. Bajo GNU/Hurd, todos los servicios TCP/IP se proporcionan por los procesos de usuario llamados *pfinet*. Zebra proporcionará toda la selección de mecanismos de routing para el proceso. Esta característica será implementada cuando GNU/Hurd sea estable.

1.3 Plataformas Soportadas

Actualmente Quagga soporta GNU/Linux, BSD y Solaris. Abajo tenemos un lista de las versiones de SO en los que Quagga funciona. Portar Quagga a otras platadormas no es muy difícil. Las dependencias de plataforma en el código sólo existen en el demonio *zebra*. Los demonios de protocolos son independientes a la plataforma. Por favor, haganos saber cuando encuentre que Quagga funciona en otra plataforma que no esté en la lista.

- GNU/Linux 2.0.37
- GNU/Linux 2.2.x
- GNU/Linux 2.3.x
- FreeBSD 2.2.8
- FreeBSD 3.x
- FreeBSD 4.x
- NetBSD 1.4
- OpenBSD 2.5
- Solaris 2.6
- Solaris 7

Varias pilas de IPv6 están actualmente en desarrollo. Zebra soporta las siguientes pilas de IPv6. Para BSD, se recomiendo la pila KAME IPv6. La pila Solaris IPv6 no está soportada aún.

- Pila Linux IPv6 para GNU/Linux 2.2.x y superiores
- Pila Kame IPv6 para BSD
- Pila INRIA IPv6 para BSD

1.4 RFCs Soportados

A continuación mostramos los RFCs de los protocolos de routing soportados:

- RFC1058
  - *Routing Information Protocol. C.L. Hedrick. 1 de Julio de 1998*
- RFC1771
  - *A Border Gateway Protocol (BGP-4). Y. Rekher & T. Li. Marzo 1995*
- RFC1997
  - *BGP Communities Attribute. R. Chandra, P. Traina & T.Li. Agosto 1996*
- RFC2080
  - *RIPng for IPv6. G.Malkin, R.Minnear. Enero 1997*
- RFC2283
  - *Multiprotocol Extension for BGP-4. T.Bates, R.Chandra, D.Katz, Y. Rekhter. Febrero 1998*
- RFC2328
  - *OSPF Version 2. J. Moy. Abril 1998*

- RFC2453
  - *RIP Version 2. G.Malkin. Noviembre 1998*
- RFC2545
  - *Use of BGP-4 Multiprotocol Extension for IPv6 Inter-Domain Routing. P. Marques, F. Dupont. Marzo 1999*
- RFC2740
  - *OSPF for IPv6. R. Coltun, D. Perguson, J.Moy. Diciembre 1999.*
- RFC2796
  - *BGP Route Reflection An alternative to full mesh IBGP T. Bates R. Chandrasekeran. Junio 1996.*

Cuando está habilitado el SNMP soportado, las siguientes RFCs están soportadas:

- RFC1227
  - *SNMP MUX protocol and MIB. M.T. Rose. Mayo 1991.*
- RFC1657
  - *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2. S. Willis, J.Burruss, J. Chu, Editor. Julio 1994.*
- RFC1850
  - *OSPF Version 2 Management Information Base. F. Baker, R. Coltun. Noviembre 1995.*

1.5 Cómo Conseguir Quagga

Quagga sigue siendo un software en versión beta y no existe todavía una versión oficial. Quagga actualmente está distribuida desde el FTP GNU y está replicado en varios sitios espejo. Tenemos planeado que Quagga-1.0 sea la primera versión::

Web oficial de Zebra

http://www.gnu.org/software/zebra/zebra.html

La web oficial de Zebra está localizada en:

http://www.zebra.org

En fecha de esta escritura, el desarrollo por zebra.org de Zebra se ha ralentizado. Mucho trabajo ha sido realizado por terceras personas para intentar mantener libre de bugs el código actual de Zebra, como resultado del cual es la bifurcación de Zebra llamada Quagga:

http://www.quagga.net/

para más información, así como enlaces a recursos adicionales de zebra.

1.6 Listas de Correo

Existe una lista de correo para discutir sobre Quagga. Si tiene cualquier comentario o sugerencia de Quagga, por favor suscribase a *http://lists.quagga.net/mailman/listinfo/quagga-users* .

Existe una lista adicional, *ZONG* para discusión general de asuntos relacionados con zebra y operación de red. Para suscribirse mande un e-mail a znog-subscribe@dishone.st, el cual contenga en el cuerpo del mensaje únicamente:

subscribe znog

Para borrarse, envíe un e-mail a *znog-unsubscribe@dishone.st* con el único cuerpo del mensaje que incluya:

unsubscribe znog

Alternativamente, se dispone de un interfaz web localizado en *http://www.dishone.st/mailman/listinfo/znog* . Los enlaces a los archivos de la lista de zong están disponibles en esa URL.

1.7 Reporte de Bugs

Si piensa que ha encontrado un bug, por favor, envíe un mensaje a *bug-zebra@zebra.org* . Cuando envíe un e-mail sobre un bug por favor remita la información de los siguientes puntos

* Por favor comente el Sistema Operativo que está utilizando. Si utilizas una pila de IPv6 por favor comente cual estás utilizando

* Envíe los resultados de *netstat -rn* e *ifconfig -a*.. La información extraída de la VTY de Zebra puede ser muy útil también con el comando *show ip route*.

* Por favor envíe sus ficheros de configuración comentados. Si especifica argumentos a los scripts de configuración, explíquelos también

Los mensajes de bugs son muy importantes para el proyecto, ya que mejoran la calidad de Quagga. Quagga actualmente está bajo desarrollo, pero por favor, no dude enviar sus bugs a *bug-zebra@gnu.org* .

2 Instalación de Zebra

Existen tres pasos para instalar el software: configuración, compilación e instalación.

La forma más sencilla de poder tener Zebra ejecutándose es el siguiente conjunto de comandos.

```
% configure
% make
% make install
```

2.1 Configuración del Software

Quagga posee un excelente script de configuración, el script de configuración detecta automáticamente muchas de las configuración del host. Hay muchas opciones de configuración que se pueden utilizar para desactiva el soporte para IPv6, para deshabilitar la compilación de demonios específicos, y para habilitar el soporte para SNMP..

- '–enable-guile'
  - *Comienza la compilación del interprete e zebra-guile (astucia). Usted necesitará la biblioteca guile para hacer esto. La aplicación de Zebra-guile no está terminada aún. Así que esta opción sólo es útil para diseñadores de zebra-guile.*
- '–disable-ipv6'
  - *Desactiva las características relacionadas con demonios de IPv6. Quagga configura la escritura automáticamente descubra la pila de IPv6. Pero a veces es posible querer desactivarla.*
- '–disable-zebra'
  - *Desactiva el demonio zebra.*
- '–disable-ripd'
  - *Desactiva el demonio ripd.*
- '–disable-ripngd'
  - *Desactiva el demonio ripng.*
- '–disable-ospfd'
  - *Desactiva el demonio ospfd.*
- '–disable-ospf6d'
  - *Desactiva el demonio ospf6d.*
- '–disable-bgpd'
  - *Desactiva el demonio bgpd.*
- '–disable-bgp-announce'
  - *Hace que bgpd que no anuncie el bgp en absoluto. Este rasgo es bueno para usar el bgpd como el BGP anuncio oyente.*
- '–enable-netlink'
  - *Obliga a habilitar el interfaz GNU/Linux netlink. El script de configuración de Zebra descubre los interfaces netlink verificando el archivo de cabecera. Cuando el archivo de cabecera no lo encuentra en el script de configuración del kernel en ejecución no se habilita el soporte para netlink.*
- '–enable-snmp'
  - *Habilita el soprte de SNMP. En el valor predeterminado, el soporte de SNMP no está habilitado.*
- '–enable-nssa'
  - *Habilita el soporte para Not So Stubby Area (RFC3101) en ospfd.*
- '–enable-opaque-lsa'
  - *Habilita el soporte para LSAs Opacos (RFC2370) en ospfd.*

- '–disable-ospfapi'
  - *Deshabilita el soporte para OSPF-API, un API al interfaz directamente con ospfd. OSPF-API está habilitado si se ha habilitado con la opción –enable-opaque-lsa.*
- '–disable-ospfclient'
  - *Deshabilita la contrucción del ejemplo del cliente OSPF-API.*
- '–enable-ospf-te'
  - *Habilita el soporte para OSPF Traffic Engineering Extension (intenet-draft), esto requiere el soporte para LSAs Opacos.*
- '–enable-multipath=ARG'
  - *Habilita el soporte para Multipath de Coste Igual. ARG es el máximo número para habilitar caminos ECMP, seleccionado a 0 permite ilimitado número de caminos.*
- '–enable-rtadv'
  - *Habilita el soporte para el anuncio de encaminamiento IPv6 en zebra.*

Se pueden especificar cualquier combinación de las opciones anteriores al script de configuración. También, puede ser útil cambiar el directorio de la instalación; especifique las opciones siguientes al configure la escritura.

Por defecto, los ejecutables se ponen en '/usr/local/sbin' y la configuración archiva en '/usr/local/etc'. El prefijo de la instalación '/usr/local/' puede cambiarse cambiando las opciones en el script de configuración.

- '–prefix=prefix'
  - *Instala los archivos independientes de la arquitectura en prefix [/usr/local].*
- '–sysconfdir=dir'
  - *Guarda la configuración de solo lectura de ejemplo en el directorio dir [prefix/etc]. Nótese que los ficheros de configuración de ejemplo serán instalados aquí.*
- '–localstatedir=dir'
  - *Configura zebra para utilizar dir para los ficheros de estado local, como los picheros pid y sockets de unix.*

Adicionalmente, se puede configurar zebra para descartar sus elevados privilegios justo después de su arranque y cambiarlos a otro usuario, existen tres opciones de configuración para controlar el comportamiento de zebra.

- '–enable-user=user'
  - *Cambia al usuario ARG justo después del arranque, y se ejecuta como usuario ARG en operación normal.*
- '–enable-group=group'
  - *Cambia de forma real y efectiva el grupo a group justo después de arrancar.*

- '–enable-vty-group=group'
  - *Crea sockets Unix Vty (para usar con vtysh) este el propietario de este grupo se selecciona con group. Esto permite crear un grupo separada conn restricción del acceso a sólo los sockets Vty, permitiendo delegar este grupo a usuarios individuales, o ejecutando vtysh setgip a ese grupo.*

El usuario y grupo por defecto si no se configura otro es 'quagga'. Nótese que este usuario o grupo requiere permiso de escritura al directoro local de estado (ver –localstatedir) y requiere al menos acceso de lectura, y escritura si no desea permitir a los demonios escribir su propia configuración, al directorio de configuración (ver –sysconfdir).

En sistemas donde tenga la librería de manipulación de capacidades 'libcap' (actualmente sólo en linux), el sistema quagga retendrá sólo las capacidades mínimas requeridas, y sólo utilizará esas capacidades en los periodos necesarios. En sistemas sin libcap, quagga se ejecutará sólo con el usuario especificado y sólo cambiará su uid a uid 0 por pequeños periodos.

% ./configure –disable-ipv6

Este comando configurará zebra y los demonios de routing deshabilitando las características relacionadas con los demonios relacionados con IPv6.

Hay varias opciones disponibles sólo para sistemas GNU/Linux:

2.2 Construir el Software

Después de configurar el software, hay que compilarlo para el sistema en el que se va a utilizar. Simplemente es necesarioejecutar el comando make en el directorio raíz de la fuente y el software será compilado. Si tiene algún problema en este punto, por favor envíe un reporte de bug. Ver sección 1.7 Reporte de Bugs.

```
% ./configure
.
.
.
./configure output
.
.
.
% make
```

2.3 Instalar el Software

Instalando el software a su sistema consiste en copiar los programas compilados y los archivos de apoyo a su situación normal. Después de que el proceso de la instalación ha completado, estos archivos se han copiado de su directorio de trabajo en *'/usr/local/bin'*, y en *'/usr/local/etc'*

Para instalar la Zebra, emita el orden siguiente a su prompt del shell: *make install*.

```
%
% make install
%
```

Los demonios de Zebra tienen su propio interfaz terminal o VTY. Después de la instalación, usted tiene que instalar el número del puerto de cada bestia para poder conectarse a ellos. Por favor agregue las entradas siguientes a '/etc/services'

```
zebrasrv 2600/tcp    # zebra service
zebra    2601/tcp    # zebra vty
ripd     2602/tcp    # RIPd vty
ripngd   2603/tcp    # RIPngd vty
ospfd    2604/tcp    # OSPFd vty
bgpd     2605/tcp    # BGPd vty
ospf6d   2606/tcp    # OSPF6d vty
```

Si usted usa un FreeBSD más nuevo que 2.2.8, las entradas anteriores ya se agregan a *'/etc/services'* no hay ninguna necesidad de agregarlo así que. Si usted especifica un número del puerto al empezar el demonio, estas entradas no pueden necesitarse.

Una vez llegados a este punto es necesario realizar cambios en los ficheros de configuración en */usr/local/etc/*.conf*

3 Comandos Básicos

Hay cinco demonios en uso, y hay un demonio gerente. Estos demonios pueden localizarse en las máquinas separadas del demonio del gerente. Cada uno de estos demonios escuchará en un puerto particular para las conexiones de VTY entrantes. Los demonios de routing son:

- *ripd*, *ripngd*, *ospfd*, *ospf6d*, *bgpd*
- *zebra*

Las siguientes secciones discuten los comandos comunes a todos los demonios de encaminamiento.

3.1 Comandos de Configuración

En un archivo de configuración, usted puede escribir las opciones de la depuración, la contraseña de un vty, las configuraciones del demonio de oruting, un nombre de archivo de log, y muchas más cosas. Esta información forma la configuración para que el demonio de routing pueda empezar.

Los archivos de configuración generalmente se encuentran en:

- *'/usr/local/etc/*.conf'*

Cada uno de los demonios tiene su propio archivo de configuración. Por ejemplo, el archivo de configuración por defecto de Zebra es:

- *'/usr/local/etc/zebra.conf'*

El nombre del demonio más *'.conf'* es el nombre por defecto del fichero de configuración. Se puede especificar un fichero de configuración utilizando la opción *-f* o *–config-file* cuando se arranca el demonio.

3.1.1 Comandos Básicos de Configuración

- Comando: **hostname** ***hostname*** {}
  - Configura el nombre del host en el router.
- Comando: **password** ***password*** {}
  - Configura el password para el interfaz vty. Si no hay password, el vty no aceptará conexiones
- Comando: **enable** ***password password*** {}
  - Configura el password del modo enable
- Comando: **log** ***stdout*** {}
- Comando: **no log** ***stdout*** {}
  - Configuran la salida a stdout
- Comando: **log file** ***filename*** {}
  - Si quiere que los logs se guarden en un fichero en particular por favor, especifique el *filename* de la siguiente manera:
  - *log file /usr/local/etc/bgpd.log*
- Comando: **log syslog** {}
- Comando: **no log syslog** {}
  - Configuran la salida del syslog
- Comando: **write** ***terminal*** {}
  - Muestra la configuración actual por el interfaz vty
- Comando: **write file** {}
  - Guarda la configuración actual al fichero de configuración
- Comando: **configure terminal**{}
  - Cambia al modo de configuración. Este comando es el primer paso de la configuración
- Comando: **terminal length** ***<0-512>*** {}
  - Configura la longitud de la pantalla de terminal de <0-512>, si la longitud se pone a 0, no se puede ver nada.
- Comando: **who** {}
- Comando: **list**{}
  - Comandos para listar
- Comando: **service password-encryption** {}
  - Encripta el password

- Comando: **service advanced-vty** {}
  - Habilita el modo avanzado de VTY
- Comando: **service terminal-length** <*0-512*> {}
  - Configura la longitud de la linea del interfaz. Esta configuración se aplica a todas las interfaces VTY
- Comando: **show version** {}
- Muestra la versión actual de Zebra y la información sobre la compilación en el host.
- Comando: **line vty**{}
  - Entra en el modo de configuración de los vty
- Comando: **banner motd default** {}
  - Configura la cadena de caracteres del motd
- Comando: **no banner motd** {}
  - No se muestra el banner
- Comando de Línea: **exec-timeout** ***minute*** {}
- Comando de Línea: **exec-timeout** ***minute second*** {}
  - Establece el tiempo de conexión en los VTY. Cuando sólo se especifica un argumento, este se utiliza como el tiempo máximo en minutos. Opcionalmente se puede añadir un segundo argumento que es utilizado como tiempo en segundos. El valor por defecto es de 10 minutos. Si se configura un tiempo 0, este es entendido como que no hay límite de tiempo.
- Comando de Línea: **no exec-timeout** {}
  - Este comando es igual a *exec-timeout 0 0*

3.1.2 Ejemplo de Fichero de Configuración

A continuación se muestra una configuración de ejemplo para el fichero de configuración del demonio Zebra

```
!
! Zebra configuration file
!
hostname Router
password zebra
enable password zebra
!
log stdout
!
!
```

Los caracteres "!" y "#" indican que se trata de un comentario. Si el primer carácter de una línea es uno de estos caracteres entonces se considera que toda la línea es un comentario, si uno de estos caracteres se encuentra en otra posición se entiende como un caracter normal.

password zebra!password

En este caso se entiende que la password es zebra!password y no que es un comentario, así que cuidado con esto.

3.2 Opciones de Invocación Comunes

Las siguientes opciones son las más comunes para todos los demonios de Quagga

- '-d'
- '–daemon'
  - Se ejecuta en modo demonio.
- '-f file'
- '–config_file=file'
  - Selecciona el nombre del fichero de configuración.
- '-h'
- '–help'
  - Muestra esta ayuda y sale.
- '-i file'
- '–pid_file=file'
  - Hasta el arranque el identificador de proceso del demonio se escribe típicamente en el directorio '/var/run'. Este fichero puede ser utilizado en el arranque del sistema para implementar comandos como .../init.d/zebra status, .../init.d/zebra restart o .../init.d/zebra stop.

El nombre del fichero es una opción de ejecución, así como una opción de configuración que permite que varios demonios de encaminamiento puedan funcionar simultáneamente. Este es útil cuando se utiliza Quagga para implemtnar un looking glass. Una máquina puede ser uilizada para recolectar diferentes visiones de diversos puntos de la red..

- '-A address'
- '–vty_addr=address'
  - Selecciona la dirección local del VTY a cegar, el socket VTY será sólamente abierto en esa dirección.
- '-P port'
- '–vty_port=port'

  - Selecciona el número de puerto TCP para el VTY. Si se escoge 0 entonces no se abriran los sockets.

- '-u user'
- '–vty_addr=user'
  - Selecciona el usuarui y el grupo con el que se va a ejecutar.
- '-v'
- '–version'
  - Muestra la versión del programa.

3.3 Interfaces Virtuales de Terminal

VTY - Interfaz Virtual de Terminal (Virtual TeletYpe) es un interfaz de línea de comandos (CLI) para modificar y/o ver la configuración actual.

3.3.1 Introducción a VTY

El VTY tiene la función de interfaz para el Virtual TeletYpe. Esto significa que es posible conectarse al demonio vía protocolo telnet. Para habilitar el interfaz VTY, es necesario configurar un password para el VTY. Si no existe un password para el VTY no es posible conectarse al interfaz VTY.

```
% telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello, this is zebra (version 0.88)
Copyright 1997-2000 Kunihiro Ishiguro
User Access Verification
Password: XXXXX
Router> ?
enable Turn on privileged commands
exit Exit current mode and down to previous mode
help Description of the interactive help system
list Print command list
show Show running system information
who Display who is on a vty
Router> enable
Password: XXXXX
Router# configure terminal
Router(config)# interface eth0
Router(config-if)# ip address 10.0.0.1/8
Router(config-if)# ^Z
Router#
```

El comando ? es muy útil para buscar otros comandos.

3.3.2 Modos de VTY

VTY View Mode

Este modo es sólo para acceso solo lectura al CLI. Al abandonar este modo se abandona el sistema, o también es posible entrar en el modo de 'enable'.

VTY Enable Mode

Este es el modo de acceso lectura/escritura al CLI. Es posible salir del sistema o volver al modo de solo lectura.

VTY Other Modes

Este modo existe para comandos o funciones especiales.

3.3.3 Comandos de CLI para el VTY

Los comandos que se utilizan para en el CLI se describen en las próximas tres subsecciones.

Comandos de movimiento del cursor en el CLI

Estos son los comandos que se utilizan para el movimiento del cursor en el CLI. Nótese que el caracter <C> significa mantener apretada la tecla Ctrl (Control).

- C-f
- <CURSOR DERECHA>
  - Mueve hacia delante un caracter
- C-b
- <CURSOR IZQUIERDA>
  - Mueve hacia atrás un carácter
- M-f
  - Desplaza el cursor hacia delante una palabra
- M-b
  - Desplaza el cursor hacia atrás una palabra
- C-a
  - Desplaza el cursor al inicio de la línea
- C-e
  - Desplaza el cursor al final de la línea

Comandos Avanzados en el CLI

Aquí se encuentran varios comandos adicionales del CLI para completar líneas, ayuda y gestión de la sesión VTY

- C-c
  - Interrumpe la entrada actual y se mueve a la siguiente línea
- C-z
  - Finaliza la configuración actual y se mueve al nodo inicial
- C-n
- <CURSOR ABAJO>
  - Se desplaza a la siguiente línea en el buffer del historial
- C-p
- <CURSOR ARRIBA>
  - Se desplaza a la línea anterior en el buffer del historial
- <TAB>
  - Completa la línea pulsando sobre la tecla de tabulación

Se puede utilizar el comando de ayuda pulsando *help* al inicio de la línea. Pulsando ? en cualquier punto de la línea se mostrará las posibilidades de completar la línea.

4 Demonio Zebra

Zebra es el gestor de encaminamiento IP. Proporciona las actualizaciones de la tabla de encaminamiento del kernel, lookup del interfaz, y redistribución de rutas entre los diferentes protocolos de encaminamiento.

4.1 Invocando Zebra

Junto a las opciones de invocación comunes *zebra* dispone de las siguientes opciones de invocación específicas.

- -b
- –batch
  - Ejecuta zebra en modo de procesamiento por lotes. Zebra analiza los ficheros de configuración y termina inmediatamente.
- -k
- –keep_kernel
  - En el momento de arrancar Zebra, no borra las rutas antiguas.
- -l
- –log-mode

  - Comienza la sesión de forma detallada.

- -r
- –retain
  - Cuando termina el programa se retienen las rutas añadidas por zebra.

4.2 Comandos de Interfaz

- Comando: **interface *NOMBRE_DEL_INTERFAZ*** {}
- Comando del Interfaz: **shutdown** {}
- Comando del Interfaz: **no shutdown** {}
  - Levanta o tira el interfaz en el interfaz actual.
- Comando del Interfaz: **ip address *DIRECCIÓN/PREFIJO*** {}
- Comando del Interfaz: **ip6 address *DIRECCIÓN/PREFIJO*** {}
- Comando del Interfaz: **no ip address *DIRECCIÓN/PREFIJO*** {}
- Comando del Interfaz: **no ip6 address *DIRECCIÓN/PREFIJO*** {}
  - Configura la dirección IP del interfaz ya sea IPv4 o IPv6.
- Comando del Interfaz: **ip address *DIRECCIÓN/PREFIJO secondary***{}
- Comando del Interfaz:**no ip address *DIRECCIÓN/PREFIJO secondary*** {}
  - Configura con la bandera (flag) secundaria esta dirección. Esto causa que ospfd no trate la dirección como de distinta subred.
- Comando del Interfaz: **description *DESCRIPCIÓN ...*** {}
  - Configura una descripción al interfaz.
- Comando del Interfaz: **multicast** {}
- Comando del Interfaz:**no multicast** {}
  - Habilita o deshabilita la bandera de multicast para el interfaz.
- Comando del Interfaz: **bandwidth <1-10000000>** {}
- Comando del Interfaz: **no bandwidth <1-10000000>** {}
  - Configura el ancho de banda del interfaz.
  - Este es para calcular el coste en OSPF. Este comando no modifica la configuración del interfaz físico.
- Comando del Interfaz: **link-detect** {}
- Comando del Interfaz: **no link-detect** {}
  - Habilita/deshabilita los enlaces detectados en plataformas que soportan esta funcionalidad. Actualmente sólo linux con ciertos drivers - aquellos que soportan la propiedad del flag IFF_RUNNING.

4.3 Comandos de Rutas Estáticas

El encaminamiento estático es una característica muy fundamental de las tecnologías de encaminamiento. Describe el prefijo estático y la puerta de enlace o gateway.

- Comando: **ip route *RED PUERTA_DE_ENLACE*** {}
  - *RED* es el la red de destino con el formato A.B.C.D/M
  - *PUERTA_DE_ENLACE* es la puerta de enlace con el siguiente formato A.B.C.D., en vez de puerta de enlace también podemos poner el nombre del interfaz.

```
ip route 10.0.0.0/8 10.0.0.2
ip route 10.0.0.0/8 ppp0
ip route 10.0.0.0/8 null0
```

En el primer ejemplo se define una ruta estática a la red 10.0.0.0/8 a través de la puerta de enlace 10.0.0.2

En el segundo ejemplo se define la misma ruta estática, pero esta vez es a través del interfaz ppp0

- Comando: **ip route *RED MASCARA PUERTA_DE_ENLACE*** {}
  - Esta es una alternativa a la versión de antes. Utilizamos la máscara con el formato A.B.C.D en vez de utilizar el número de bits de la misma.

```
ip route 10.0.0.0 255.255.255.0 10.0.0.2
ip route 10.0.0.0 255.255.255.0 ppp0
ip route 10.0.0.0 255.255.255.0 null0
```

Este sería el mismo ejemplo que antes

- Comando: **ip route *RED PUERTA_DE_ENLACE DISTANCIA*** {}
  - Instala la ruta con una distancia especificada.

Ruta estática con múltiples puertas de enlace:

```
ip route 10.0.0.1/32 10.0.0.2
ip route 10.0.0.1/32 10.0.0.3
ip route 10.0.0.1/32 eth0
```

En este caso no tenemos ruta a 10.0.0.2 ni a 10.0.0.3, y el interfaz ethernet0 está alcanzable, entonces la última ruta es la que se instala en el kernel.

```
zebra> show ip route
S> 10.0.0.1/32 [1/0] via 10.0.0.2 inactive
via 10.0.0.3 inactive
```

```
* is directly connected, eth0

ip route 10.0.0.0/8 10.0.0.2
ip route 10.0.0.0/8 10.0.0.3
ip route 10.0.0.0/8 null0 255
```

Este instalará una ruta con múltiples saltos si es que son alcanzables, así como una ruta de gran métrica, la cual puede ser útil para prevenir tráfico destinado para un prefijo para encontrar rutas menos específicas (p.e. por defecto) debe ser especificados las puertas de enlace no alcanzables, por ejemplo:

```
zebra> show ip route 10.0.0.0/8
Routing entry for 10.0.0.0/8
Known via "static", distance 1, metric 0
10.0.0.2 inactive
10.0.0.3 inactive
Routing entry for 10.0.0.0/8
Known via "static", distance 255, metric 0
directly connected, Null0
```

- Comando: **ipv6 route *RED PUERTA_DE_ENLACE*** {}
- Comando: **ipv6 route *RED PUERTA_DE_ENLACE DISTANCIA*** {}
  - Estos son parecidos a sus contadores ipv4.
- Comando: **table *NOMBRE_DE_LA_TABLA*** {}
  - Selecciona la tabla de routing del kernel principal a utilizar. Esta funcionalidad sólo funciona para kernels que soporten múltiples tablas de routing (i.e. Linux 2.2.x).

4.4 Comandos del Modo Terminal de Zebra

- Comando: **show ip route** {}
  - Muestra las rutas que actualmente tiene zebra en su base de datos

```
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
B - BGP * - FIB route.
K* 0.0.0.0/0 203.181.89.241
S 0.0.0.0/0 203.181.89.1
C* 127.0.0.0/8 lo
C* 203.181.89.240/28 eth0
```

- Comando: **show ipv6 route** {}
- Comando: **show interface**{}

- Comando: **show ipforward** {}
  - Muestra si la función de IP forwarding (reenvío) está habilitada o no. Casi todos los kernel del tipo UNIX pueden ser configurados con el IP forwarding deshabilitado. Si esto ocurre, la caja no podrá funcionar como un router.
- Comando: **show ipv6forward** {}
  - Muestra si el IPv6 forwarding está activado o no.

5 Demonio de RIP

RIP - Protocolo de Información de Encaminamiento (Routing Information Protocol) es un protocolo de pasarela interior ampliamente desplegado. RIP fue desarrollado en la década de 1970 en los Laboratorios Xerox como parte del protocolo de encaminamiento XNS. RIP es un protocolo de vector-distancia y está basado en el algoritmo de Bellman-Ford. Como protocolo de vector-distancia, un router funcionando con RIP envía actualizaciones a sus vecinos periódicamente, permitiendo la convergencia así a una topología conocida. En cada actualización, la distancia a una red dada será comunicada a todos los demás routers vecinos del mismo.

ripd soporta RIP versión 2 tal y como viene descrito en el RFC2453 y RIP versión 1 tal y como viene descrito en el RFC1058.

5.1 Arrancar y Parar ripd

El nombre del fichero de configuración por defecto de *ripd* es 'ripd.conf'. Cuando se invoca ripd busca el directorio /usr/local/etc. Si *ripd.conf* no se encuentra continúa buscando en el directorio actual.

RIP utiliza el puerto UDP 521 para envíar y recibir los paquetes RIP. Así que el usuario debe tener la capacidad para enlazar el puerto, generalmente esto significa que el usuario debe tener los privilegios del superusuario (root). El protocolo RIP requiere información del interfaz mantenida por el demonio *zebra*. Así que ejecutar *zebra* es obligatorio. Así la sucesión mínima para RIP es la siguiente:

```
# zebra -d
# ripd -d
```

Hay que tener en cuenta que se tiene que invocar *zebra* antes de invocar *ripd*.

Para parar *ripd*. Por favor use *kill 'cat /var/run/ripd.pid'*. Ciertas señales tienen un significado especial para ripd.

- 'SIGHUP'
  - Recarga el fichero de configuración 'ripd.conf'. Todas las configuraciones son reseteadas. Todas las rutas aprendidas son borradas y eliminadas de la tabla de encaminamiento.
- 'SIGUSR1'
  - Rota el fichero de log de *ripd*.

- 'SIGINT'
- 'SIGTERM'
  - *ripd* realiza un barrido de todas las rutas de RIP existentes para terminarlas correctamente.

Las opciones de invocación comunes de *ripd* están especificadas.

- '-r'
- '–retain'
  - Cuando el programa termina, se retienen ciertas rutas añadidas por *ripd*.

5.1.1 Máscara de red de RIP

La característica de máscara de red de *ripd* está soportada para ambas versiones, la 1 y la 2. Aunque la versión 1 de RIP originalmente no contenía información de la máscara. En la versión 1 de RIP, las clases eran utilizadas originalmente para determinar el tamaño de la máscara. Las redes de Clase A utilizaban 8 bits para la máscara, las redes de Clase B utilizaban 16 bits para la máscara, mientras que las redes de Clase C utilizaban 24 bits para la máscara. Actualmente, el método más utilizado para el tamaño de la máscara de un paquete consiste en asignar al paquete la máscara en base al interfaz que recibió el paquete. La versión 2 de RIP soporta la máscara de subred de tamaño variable variable length subnet mask (VLSM). Extendiendo la submáscara de red, la máscara puede ser dividida y puede reusarse. Cada subred puede usarse para propósitos diferentes como grandes o medianas LANs y enlaces WAN. El demonio *ripd* de Quagga no soporta las máscaras nosecuenciales, las cuales están incluidas en la especificación de RIP versión 2.

En caso de existir información similar con el mismo prefijo y métrica, la información antigua será eliminada. Ripd actualmente no soporta rutas multipath con el mismo coste.

5.2 Configuración de RIP

- Comando: **router rip** {}
  - El comando *router rip* es necesario para habilitar RIP. Para deshabilitar RIP hay que utilizar el comando *no router rip*. RIP debe habilitarse antes de llevar a cabo cualquiera de los comandos.
- Comando: **no router rip**{}
  - Deshabilita RIP

RIP puede configurarse para procesar paquetes de Versión 1 o Versión 2, el modo predefinido es Versión 2. Si ninguna versión se especifica, entonces los demonios de RIP tendrán como valor predefinido Versión 2. Si RIP se pone a Versión 1, la configuración de "Versión 1" se desplegará, pero la configuración de "Versión 2" no se desplegará mientras no se especifique que se utiliza Versión 2.

- Comando de RIP: **version version** {}
  - Configura la versión del proceso, la versión puede ser "1" o "2"
- Comando de RIP: **network *network***{}

- Comando de RIP: **no network *network***{}
  - Selecciona el interfaz habilitado por red. Los interfaces que posean direcciones que dicha red son habilitados. Este grupo de comandos habilita o deshabilita RIP de los interfaces entre ciertas redes de direcciones. Por ejemplo, si la red 10.0.0.0/24 está habilitada por RIP, esto significa que desde la dirección 10.0.0.0 hasta la 10.0.0.255 está habilitada par RIP. El comando *no network* deshabilitará el RIP de la red específica.
- Comando de RIP: **network *ifname*** {}
- Comando de RIP: **no network *ifname*** {}
  - Configura a habilitado el interfaz descrito como ifname. Se habilita tanto el envío como la recepción de paquetes RIP en el puerto especificado en el comando *network ifname*. El comando *no network ifname* desabilita RIP en el interfaz especificado.
- Comando de RIP: **neighbor *a.b.c.d*** {}
- Comando de RIP: **no neighbor *a.b.c.d*** {}
  - Especifica el vecino de RIP. Cuando un vecino no entiende multicast, este comando se utiliza para especificar los vecinos. En muchos casos, no todos los routers son capaces de entender multicasting, cuando los paquetes se envían a una red o grupo de direcciones. En una situación donde el vecino no procesa los paquetes de multicast, es necesario establecer un enlace directo entre los routers. El comando neighbor permite al administrador de la red especificar un router como vecino de RIP. El comando *no neighbor a.b.c.d* deshabilita el vecino de RIP.

A continuación se muestra una configuración simple de RIP. El interfaz *eth0* tiene habilitada la red *10.0.0.0/8* y el RIP.

```
!
router rip
network 10.0.0.0/8
network eth0
!
```

Interfaz Pasivo

- Comando de RIP: **passive-interface *IFNAME*** {}
- Comando de RIP: **no passive-interface*IFNAME*** {}
  - Este comando configura el interfaz especificado en modo pasivo. Cuando un interfaz se configura en modo pasivo, todos los paquetes recibidos se procesan como normales y ripd no envía paquetes multicast o unicast excepto a los vecinos especificados en el comando *neighbor*.

Control de versión de RIP

- Comando de Interfaz: **ip rip send version *versión*** {}

  - La versión puede ser "1", "2" o "1 2". Este comando de configuración sobrescribe la configuración de rip del router. El comando habilitará el interfaz seleccionado para enviar paquetes con versión RIP versión 1, RIP versión 2 o ambos. En el caso de elegir "1 2", los paquetes serán envíados mediante broadcast y multicast.
  - Por defecto la versión envíada es únicamente "2".

- Configuración de Interfaz: **ip rip receive version *versión*** {}
  - La configuración de la versión para recibir paquetes RIP. Este comando habilitará el interfaz seleccionado en versión RIP versión 1, RIP versión 2 o ambas.

Horizonte Divido con RIP

- Comando de Interfaz: **ip split-horizon** {}
- Comando de Interfaz: **no ip split-horizon** {}
  - Controla el horizonte dividido en el interfaz. Por defecto el *horizonte dividido está habilitado*. Si se quiere no habilitar el horizonte dividido es necesario especificarlo mediante el comando *no ip split-horizon*.

5.3 Cómo anunciar rutas por RIP

- Comando de RIP: **redistribute kernel** {}
- Comando de RIP: **reditribute kernel metric <0-16>**{}
- Comando de RIP: **redistribute kernel route-map *route-map*** {}
- Comando de RIP: **no redistribute kernel**{}
  - redistribute kernel redistribuye la información de encaminamiento desde las entradas de encaminamiento del kernel a las tablas de RIP, no redistribute kernel deshabilita esas rutas.
- Comando de RIP: **redistribute static** {}
- Comando de RIP:**redistribute static metric <0-16>** {}
- Comando de RIP: **redistribute static route-map route-map**{}
- Comando de RIP: **no redistribute static** {}
  - redistribute static redistribuye la información de encaminamiento desde las entradas de rutas estáticas a las tablas de RIP, el comando no redistribute static deshabilita dichas rutas.
- Comando de RIP: **redistribute connected** {}
- Comando de RIP: **redistribute connected metric <0-16>** {}
- Comando de RIP: **redistribute connected route-map *route-map*** {}
- Comando de RIP: **no redistribute connected**{}

  - Redistribuye las rutas de conexiones directas en las tablas de RIP. no redistribute connected deshabilita dichas rutas. Este comando redistribuye las rutas conectadas directamente al interfaz en el cual no está habilitado RIP. Esta funcionalidad está habilitada por defecto.

- Comando de RIP: **redistribute ospf** {}
- Comando de RIP: **redistribute ospf metric <0-16>** {}
- Comando de RIP: **redistribute ospf route-map *route-map*** {}
- Comando de RIP: **no redistribute ospf** {}
  - redistribute ospf redistribuye la información de encaminamiento desde las entradas de rutas de ospf a las tablas de RIP. no redistribute ospf deshabilita las rutas.
- Comando de RIP: **redistribute bgp** {}
- Comando de RIP: **redistribute bgp metric <0-16>**{}
- Comando de RIP: **redistribute bgp route-map *route-map*** {}
- Comando de RIP: **no redistribute bgp** {}
  - redistribute bgp redistribuye la información de encaminamiento desde las entradas de rutas de bgp a las tablas de RIP. no redistribute bgp deshabilita las rutas.

Si sólo se quiere especificar rutas estáticas por RIP

- Comando de RIP: **default-information originate**{}
- Comando de RIP:**route a.b.c.d/m** {}
- Comando de RIP: **no route a.b.c.d/m** {}
  - Este comando es específico de Quagga. El comando route crea una ruta estática sólo dentro de RIP. Este comando debería de ser utilizado sólo por usuarios avanzados, los cuales tengan unos buenos conocimientos del protocolo RIP. En muchos casos, se recomienda crear una ruta estática y redistribuirla en RIP utilizando el comando redistribute static.

5.4 Filtrado de rutas RIP

Las rutas de RIP pueden ser filtradas mediante una distribute-list.

Comando: **distribute-list access_list direct *ifname*** {}

Es posible crear listas de acceso (access lists) a un interfaz mediante el comando *distribute-list*. *access_list* es el nombre de la lista de acceso. direct es "*in*" o "*out*". Si *direct* está en "*in*" la lista de acceso se aplica a los paquetes entrantes.

El comando *distribute-list* puede ser utilizado para filtrar el camino de RIP. *distribute-list* puede puede aplicar listas de acceso a un interfaz específico. Primero, se debe especificar la lista de acceso (access list). Acto seguido, el nombre de la lista de acceso a utilizar en el comando distribute-list. Por ejemplo, en la siguiente configuración "*eth0*" permitirá únicamente caminos que se correspondan con la ruta 10.0.0.0/8

```
!
router rip
distribute-list private in eth0
!
access-list private permit 10 10.0.0.0/8
access-list private deny any
!
```

El comando *distribute-list* puede ser aplicado tanto en entrada como en salida de datos (in/out)

Comando: **distribute-list prefix prefix_list (in|out)** ***ifname*** {}

Es posible aplicar listas de prefijos al interfaz mediante un comando de *distribute-list*. *prefix_list* es el nombre de la lista de prefijos. Posteriormente la dirección de "*in*" o "*out*". Si direct está en "*in*" la lista de acceso se aplicará a los paquetes entrantes.

5.5 Manipulación de la métrica de RIP

La métrica de RIP es la distancia de la red. Normalmente ripd incremente la métrica cuando recibe información de la red. Las rutas redistribuidas tienen una métrica de 1.

- Comando de RIP: **default-metric <1-16>** {}
- Comando de RIP: **no default-metric <1-16>** {}
  - Este comando modifica la métrica por defecto para las rutas redistribuidas. El valor por defecto es 1. Este comando no afecta a la métrica de las rutas conectadas directamente si están redistribuidas mediante el comando *redistribute conected*. Para modificar el valor de la métrica de la ruta, es necesario utilizar el comando *redistribute conected metric* o *route-map*. *offser-list* también afecta a las rutas directamente conectadas
- Comando de RIP: **offset-list access-lisst (in|out)** {}
- Comando de RIP: **offset-list access-list (in|out)** {}

5.6 Distancia de RIP

El valor de la distancia se utiliza en el demonio Zebra. Por defecto el valor de la distancia de RIP es 120.

- Comando de RIP: **distance <1-255>** {}
- Comando de RIP: **no distance <1-255>** {}
  - Configura la distancia por defecto de RIP a un valor específico.
- Comando de RIP: **distance <1-255>** ***A.B.C.D/M*** {}
- Comando de RIP: **no distance <1-255>** ***A.B.C.D/M*** {}
  - Configura la distancia por defecto de RIP cuando la dirección IP del router de origen coincide con el prefijo específico.

- Comado de RIP: **distance <1-255> A.B.C.D/M** ***access-list*** {}
- Comando de RIP: **no distance <1-255> A.B.C.D/M** ***access-list*** {}
  - Configura la distancia por defecto de RIP a un valor específico cuando la dirección IP del router de origen coincide con el prefijo especificado en la lista de acceso especificada.

5.7 RIP route-map

Utilización de ripd como soporte para la utilización de route-maps.

El argumento opcional de route-map MAP_NAME puede ser añadido a cada declaración de redistribución.

```
redistribute static [route-map MAP_NAME]
redistribute connected [route-map MAP_NAME]
.....
```

El fabricante Cisco aplica las route-map antes que las rutas sean exportadas a la tabla de RIP. En la versión de test de Quagga, ripd aplica las route-map después de que las rutas sean listadas en la tabla de rutas y antes de que las rutas sean anunciadas al interfaz (algo parecido a un filtro de salida). Actualmente este asunto no está del todo claro, pero es un borrador y es posible que cambie en un futuro.

La declaración de la route-map es necesaria para utilizar la funcionalidad de route-map.

- Route Map: **match interface** ***word*** {}
  - Este comando realiza una búsqueda en el interfaz de entrada. La notificación de esta búsqueda es diferente a Cisco. Cisco utiliza una lista de interfaces NOMBRE1 NOMBRE2 ... NOMBREN. Ripd permite sólo uno (tal vez cambie en el futuro). Además, Cisco quiere decir incluye incluyendo las rutas al siguiente salto (es algo parecido a la declaración "ip next-hop"). Ripd quiere decir interfaz al lugar donde se enviarán las rutas. Esta diferencia es debida a que el "next-hop" de varias rutas que se envían por diferentes interfaces mueden ser distintas. Tal vez sería mejor crear nuevas búsquedas diciendo "match interface-out NOMBRE" o algo parecido a esto.
- Route Map:**match ip address** ***word*** {}
- Route Map: **match ip address prefix-list** ***word*** {}
  - Busca si la ruta al destino está permitida en la lista de acceso.
- Route Map: **match ip next-hop A.B.C.D** {}
  - Cisco aquí utiliza el comando <access-list>,dirección IPv4 de ripd. Busca si la ruta tiene un siguiente salto (significando siguiente salto a los listados en la tabla de rutas de rip - "show ip rip").
- Route Map: **match metric <0-4294967295>** {}
  - Este comando busca el valor de la métrica de las actualizaciones de RIP. Para la compatibilidad con otros protocolos el rango de la métrica se muestra como <0-4294967295>. Pero para el protocolo RIP el rango de la métrica es sólamente <0-16>. Recordemos con con 16 saltos se considera el siguiente salto inaccesible.

- Route Map: **set ip next-hop A.B.C.D** {}
  - Este comando configura el valor del siguiente salto en el protocolo RIPv2. Este comando no afecta a RIPv1 porque no hay campo de siguiente salto en el paquete.
- Route Map: **set metric <0-4294967295>** {}
  - Configura la métrica para la ruta buscada cuando se envía un anuncio. El valor del rango de la métrica es muy grande para la compatibilidad con otros protocolos. Para RIP, los valores válidos de métrica son del 1 al 16.

5.8 Autenticación de RIP

- Comando de Interfaz: **ip rip authentication mode md5** {}
- Comando de Interfaz: **no ip rip authentication mode md5** {}
  - Configura el interfaz con RIPv2 con autenticación MD5.
- Comando de Interfaz: **ip rip authentication mode text** {}
- Comando de Interfaz: **no ip rip authentication mode text** {}
  - Configura el interfaz con RIPv2 con autenticación con password simple.
- Comando de Interfaz: **ip rip authentication string *string*** {}
- Comando de Interfaz: **no ip rip authentication string *string*** {}
  - La versión 2 de RIP tiene autenticación simple por texto. Este comando configura la cadena de autenticación. La cadena tiene que ser más corta que 16 caracteres.
- Comando de Interfaz: **ip rip authentication key-chain *key-chain*** {}
- Comando de Interfaz: **no ip rip authentication key-chain *key-chain*** {}
  - Especifica la cadena MD5

```
!
key chain test
key 1
key-string test
!
interface eth1
ip rip authentication mode md5
ip rip authentication key-chain test
!
```

5.9 Temporizadores de RIP

- Comando de RIP: **timers basic** ***update timeout garbage*** {}
  - El protocolo RIP tiene varios temporizadores. El usuario configura los valores de los temporizadores con el comando timers *basic*.

Los valores por defecto de los temporizadores son los siguientes:

- Tiempo de actualización (update) cada 30 segundos. Cada vez que el temporizador llega a 30 segundos se produce una actualización, el proceso de RIP despierta y envía un mensaje de respuesta no solicitado conteniendo la tabla completa de encaminamiento de todos los routers vecinos.
- El temporizador de pausa (timeout) tiene un valor por defecto de 180 segundos. Una vez llegado el tiempo sin actualizaciones la ruta se considera no válida, sin embargo la ruta se retiene un cierto periodo de tiempo, lo cual permite avisar a los vecinos que la ruta ha sido eliminada.
- El temporizador del recolector de basura (garbage) es por defecto de 120 segundos. Una vez transcurrido el tiempo de expiración del recolector de basura, la ruta es definitivamente eliminada de la tabla de encaminamiento.

El comando *timers basic* permite modificar los valores por defecto de los temporizadores que utiliza el protocolo.

- Comando de RIP: **no timers basic** {}
  - El comando *no timers basic* resetea los contadores a los valores por defecto (30, 180, 120).

5.10 Mostrar la Información de RIP

Para mostrar las rutas de RIP.

- Comando:**show ip rip** {}
  - Muestra las rutas de RIP

El comando muestra todas las rutas de RIP. Para rutas que han sido recibidas a través de RIP, este comando mostrará el tiempo que el paquete se envió y la información de la etiqueta. Este comando también mostrará esta información para las rutas redistribuidas por RIP.

- Comando: **show ip protocols** {}
  - El comando muestra el estado actual de RIP. Esto incluye los temporizadores de RIP, versión, interfaz habilitado y la información de los vecinos de RIP.

```
ripd> <bf>show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds with +/-50%, next due in 35 seconds
Timeout after 180 seconds, garbage collect after 120 seconds
```

```
Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set
Default redistribution metric is 1
Redistributing: kernel connected
Default version control: send version 2, receive version 2
Interface Send Recv
Routing for Networks:
eth0
eth1
1.1.1.1
203.181.89.241
Routing Information Sources:
Gateway BadPackets BadRoutes Distance Last Update
```

Comandos de depuración de RIP

5.11 Depuración para el protocolo RIP.

- Comando: **debug rip events** {}
  - Depura los eventos de RIP.

debug rip muestra los eventos de RIP. Enviando y recibiendo paquetes, temporizadores, y cambios en interfaces son los eventos que se pueden ver con *ripd*.

- Comando: **debug rip packet** {}
  - Depura el paquete de rip.

debug rip packet muestra información detallada sobre los paquetes de RIP. El origen y número de puerto del paquete, así como el volcado del paquete.

- Comando: **debug rip zebra** {}
  - Depura la comunicación entre zebra y ripd.

Este comando muestra la comunicación entre ripd y zebra. La principal información incluida además de la eliminación de caminos en el kernel y la información de envío y recepción.

- Comando: **show debugging rip** {}
  - Muestra la opción de depuración de ripd.

show debugging rip muestra toda la información configurada actualmente para la depuración de ripd.

6 Demonio de RIPng

ripngd soporta el protocolo RIPng, el cual esta descrito en la RFC2080. Este protocolo es la adaptación del protocolo RIP a IPv6.

6.1 Invocación de ripngd

No existen opciones de invocación especificas para *ripngd*.

6.2 Configuración de ripng

Actualmente ripngd soporta los siguientes comandos:

- Comando: **router ripng** {}
  - Habilita RIPng.
- Comando de RIPng: **flush_timer*time*** {}
  - Configura el tiempo de nivelado.
- Comando de RIPng: **network *network*** {}
  - Configura el RIPng como habilitado en la red.
- Comando de RIPng: **network *ifname*** {}
  - Configura el IPng como habilitado en el interfaz por el nombre del interfaz
- Comando de RIPng: **route *network*** {}
  - Configura el anuncio de routing estático a la red por RIPng.
- Comando: **router zebra** {}
  - Este comando está habilitado por defecto y no aparece en la configuración. Mediante este comando las rutas de RIPng van directamente al demonio de RIPng.

6.3 Comandos del Modo Terminal de ripng

- Comando: **show ip ripng**{}
- Comando: **show debugging ripng** {}
- Comando: **debug ripng events** {}
- Comando: **debug ripng packet** {}
- Comando: **debug ripng zebra** {}

6.4 Comandos de Filtering de ripng

- Comando: distribute-list access_list (in|out) ifname {}
  - Es posible aplicar una lista de acceso al Interfaz utilizando una lista de comando de *distribute-list*. En este comando *access_list* es el nombre de la lista de acceso. La dirección es '*in*' o '*out*'. Si la dirección es '*in*', la lista de acceso se aplica solo a los paquetes entrantes.

```
distribute-list local-only out sit1
```

7 Demonio de OSPF Versión 2

OSPF versión 2 es un protocolo de encaminamiento, el cual está descrito en la RFC2328 - OSPF Versión2. OSPF es un protocolo de encaminamiento IGP (Interior Gateway Protocol - Protocolo de Pasarela Interna). Comparado con RIP, OSPF puede proporcionar soporte a una red escalable y un tiempo convergencia más corto. La utilización de OSPF está muy extendida en grandes redes como podrían ser backbones de ISP y redes de grandes empresas.

7.1 Configurando ospfd

En Zebra no hay opciones específicas para ospfd (ver Opciones de Invocación Comunes). Ospfd necesita información de *zebra*, así que es necesario asegurarse que zebra se está ejecutando antes de invocar a ospfd.

Como otros demonios, la configuración de *ospfd* está en el fichero específico de configuración de OSPF "*ospfd.conf*".

7.2 Router OSPF

Para iniciar el proceso de OSPF es necesario especificar el router OSPF. Ospfd no soporta múltiples procesos OSPF.

- Comando: **router ospf** {}
- Comando: **no router ospf** {}
  - Habilita o deshabilita el proceso OSPF. Ospfd todavía no soporta múltiples procesos OSPF. Así que no es posible especificar el número de proceso de OSPF
- Comando de OSPF: **ospf router-id *a.b.c.d*** {}
- Comando de OSPF: **no ospf router-id** {}
- Comando de OSPF: **ospf abr-type *TIPO*** {}
- Comando de OSPF: **no ospf abr-type *TIPO*** {}
  - El TIPO puede ser cisco|ibm|shortcut|standard
- Comando de OSPF: **ospf rfc1583compatibility** {}
- Comando de OSPF: **no ospf rfc1583compatibility**{}
- Comando de OSPF: **passive interface *INTERFAZ*** {}
- Comando de OSPF: **no passive interface *INTERFAZ*** {}
- Comando de OSPF: **timers spf <0-4294967295> <0-4294967295>** {}
- Comando de OSPF: **no timers spf** {}
- Comando de OSPF: **refresh group-limit <0-10000>** {}
- Comando de OSPF: **refresh per-slice <0-10000>** {}

- Comando de OSPF: **refresh age-diff <0-10000>** {}
- Comando de OSPF: **auto-cost refrence-bandwidth <1-4294967>** {}
- Comando de OSPF: **no auto-cost refrence-bandwidth** {}
- Comando de OSPF: **network a.b.c.d/m area *a.b.c.d*** {}
- Comando de OSPF: **network a.b.c.d/m area <*0-4294967295*>** {}
- Comando de OSPF: **no network a.b.c.d/m area *a.b.c.d*** {}
- Comando de OSPF: **no network a.b.c.d/m area <0-4294967295>** {}
  - Este comando especifica que el interfaz OSPF está habilitado. Si el interfaz tiene una dirección de 10.0.0.1/8 entonces el comando siguiente proporciona información de la red a los routers OSPF.

```
router ospf
network 10.0.0.0/8 area 0
```

La longitud de la máscara del comando network debería de ser la misma que la longitud de la máscara de la dirección del interfaz.

7.3 Área OSPF

- Comando de OSPF: **area *a.b.c.d* range *a.b.c.d/m*** {}
- Comando de OSPF: **area <0-4294967295> range *a.b.c.d/m*** {}
- Comando de OSPF: **no area *a.b.c.d* range *a.b.c.d/m*** {}
- Comando de OSPF: **no area <0-4294967295> range a.b.c.d/m** {}
- Comando de OSPF: **area *a.b.c.d* range IPV4_PREFIX suppress**{}
- Comando de OSPF: **no area *a.b.c.d* range IPV4_PREFIX suppress** {}
- Comando de OSPF: **area *a.b.c.d* range IPV4_PREFIX substitute IPV4_PREFIX**{}
- Comando de OSPF: **no area *a.b.c.d* range IPV4_PREFIX substitute IPV4_PREFIX** {}
- Comando de OSPF: **area*a.b.c.d* virtual-link *a.b.c.d*** {}
- Comando de OSPF: **area <0-4294967295> virtual-link *a.b.c.d*** {}
- Comando de OSPF: **no area *a.b.c.d* virtual-link *a.b.c.d*** {}
- Comando de OSPF: **no area <0-4294967295> virtual-link *a.b.c.d*** {}
- Comando de OSPF: **area *a.b.c.d* shortcut** {}
- Comando de OSPF: **area <0-4294967295> shortcut** {}
- Comando de OSPF: **no area *a.b.c.d* shortcut** {}
- Comando de OSPF: **no area <0-4294967295> shortcut** {}

- Comando de OSPF: **area *a.b.c.d* stub** {}
- Comando de OSPF: **area <0-4294967295> stub** {}
- Comando de OSPF: **no area *a.b.c.d* stub** {}
- Comando de OSPF: **no area <0-4294967295> stub**{}
- Comando de OSPF: **area *a.b.c.d* stub no-summary** {}
- Comando de OSPF: **area <0-4294967295> stub no-summary** {}
- Comando de OSPF: **no area *a.b.c.d* stub no-summary** {}
- Comando de OSPF: **no area <0-4294967295> stub no-summary**{}
- Comando de OSPF: **area *a.b.c.d* default-cost <0-16777215>** {}
- Comando de OSPF: **no area *a.b.c.d* default-cost <0-16777215>** {}
- Comando de OSPF: **area *a.b.c.d* export-list NAME** {}
- Comando de OSPF: **area <0-4294967295> export-list NAME** {}
- Comando de OSPF: **no area *a.b.c.d* export-list NAME** {}
- Comando de OSPF: **no area <0-4294967295> export-list NAME** {}
- Comando de OSPF: **area *a.b.c.d* import-list NAME** {}
- Comando de OSPF: **area <0-4294967295> import-list NAME**{}
- Comando de OSPF: **no area *a.b.c.d* import-list NAME** {}
- Comando de OSPF: **no area <0-4294967295> import-list NAME** {}
- Comando de OSPF: **area *a.b.c.d* authentication** {}
- Comando de OSPF: **area <0-4294967295> authentication** {}
- Comando de OSPF: **no area *a.b.c.d* authentication** {}
- Comando de OSPF: **no area <0-4294967295> authentication** {}
- Comando de OSPF:**area *a.b.c.d* authentication message-digest**{}
- Comando de OSPF: **area <0-4294967295> authentication message-digest** {}

7.4 Interfaz OSPF

- Comando de Interfaz: **ip ospf authentication-key AUTH_KEY**{}
- Comando de Interfaz: **no ip ospf authentication-key** {}
  - Configura la autenticación de OSPF a una simple contraseña. Después de configurar *AUTH_KEY*, todos los paquetes OSPF son autenticados. *AUTH_KEY* tiene un tamaño de de hasta 8 caracteres.

- Comando de Interfaz: **ip ospf message-digest-key KEYID md5 KEY** {}
- Comando de Interfaz: **no ip ospf message-digest-key** {}
  - Configura la autenticación de OSPF con una clave criptográfica. El algoritmo criptográfico es MD5. KEYID identifica la clave secreta utilizada para crear el compendio del mensaje. KEY es el compendio actual del mensaje, la clave puede ser de hasta 16 caracteres.
- Comando de Interfaz: **ip ospf cost <1-65535>** {}
- Comando de Interfaz: **no ip ospf cost** {}
- Configurarción del coste del interfaz para el interfaz específico. El valor del coste está configurado por la métrica del campo de métrica del LSA del router, y se utiliza para el cálculo del algoritmo SPF.
- Comando de Interfaz: **ip ospf dead-interval <1-65535>**{}
- Comando de Interfaz: **no ip ospf dead-interval** {}
  - Configura el número de segundos para el intervalo en el que se considera el router muerto, este tiempo es el valor utilizado para el contador de inactividad. Este valor debe tener el mismo valor para todos los routers añadidos a una red común. El valor por defecto es de 40 segundos.
- Comando de Interfaz: **ip ospf hello-interval <1-65535>** {}
- Comando de Interfaz: **no ip ospf hello-interval** {}
- Configura el número de segundos para en temporizador del intervalo de Hello. Al configurar este valor, los paquetes Hello serán envíados cada vez que el temporizador expire. Este valor debe tener el mismo valor para todos los routers añadidos a una red común. El valor por defecto es de 10 segundos.
- Comando de Interfaz: **ip ospf network (broadcast|non-broadcast|point-to-multipoint|point-to-point)** {}
- Comando de Interfaz: **no ip ospf network** {}
- Configura explícitamente el tipo de red para el interfaz específico.
- Comando de Interfaz: **ip ospf priority <0-255>**{}
- Comando de Interfaz: **no ip ospf priority** {}
  - Configura la prioridad del router con un número entero. Configurando un valor más alto, el router tendrá más posibilidades de convertirse en Router Designado (Designated Router). Si configuramos el valor a 0, el router no será nunca elegido Router Designado (Designated Router). El valor por defecto es 1.
- Comando de Interfaz: **ip ospf retransmit-interval <1-65535>** {}
- Comando de Interfaz: **no ip ospf retransmit interval** {}
  - Configura el número de segundos para el temporizador de retransmisión. Este valor se utiliza para retransmitir la descripción de la base de datos los paquetes de LSR. El valor por defecto es 5 segundos.
- Comando de Interfaz: **ip ospf transmit-delay** {}

- Comando de Interfaz: **no ip ospf transmit-delay** {}
  - Configura el número de segundos para el valor de retraso (delay) del valor de la variable InfTrans-Delay. La edad de los LSAs se puede incrmentar por este valor cuando se transmiten. El valor por efecto es 1 segundo.

7.5 Redistribución de rutas a OSPF

- Comando de OSPF: **redistribute (kernel|connected|static|rip|bgp)** {}
- Comando de OSPF:**redistribute (kernel|connected|static|rip|bgp)** ***route-map*** {}
- Comando de OSPF: **redistribute (kernel|connected|static|rip|bgp) metric-type (1|2)** {}
- Comando de OSPF: **redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) route-map *word*** {}
- Comando de OSPF: **redistribute (kernel|connected|static|rip|bgp) metric <0-16777214>** {}
- Comando de OSPF:**redistribute (kernel|connected|static|rip|bgp) metric <0-16777214> route-map *word*** {}
- Comando de OSPF: **redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) metric <0-16777214>** {}
- Comando de OSPF:**redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) metric <0-16777214> route-map word** {}
- Comando de OSPF: **no redistribute (kernel|connected|static|rip|bgp)** {}
- Comando de OSPF: **default-information originate** {}
- Comando de OSPF: **default-information originate metric <0-16777214>** {}
- Comando de OSPF: **default-information originate metric <0-16777214> metric-type (1|2)** {}
- Comando de OSPF: **default-information originate metric <0-16777214> metric-type (1|2) route-map *word*** {}
- Comando de OSPF: **default-information originate always** {}
- Comando de OSPF: **default-information originate always metric <0-16777214>**{}
- Comando de OSPF: **default-information originate always metric <0-16777214> metric-type (1|2)** {}
- Comando de OSPF: **default-information originate always metric <0-16777214> metric-type (1|2) route-map*word*** {}
- Comando de OSPF: **no default-information originate** {}
- Comando de OSPF: **distribute-list NAME out (kernel|connected|static|rip|ospf)** {}
- Comando de OSPF: **no distribute-list NAME out (kernel|connected|static|rip|ospf** {}
- Comando de OSPF: **default-metric <0-16777214>**{}

- Comando de OSPF: **no default-metric**{}
- Comando de OSPF: **distance <1-255>** {}
- Comando de OSPF: **no distance <1-255>** {}
- Comando de OSPF: **distance ospf (intra-area|inter-area|external) <1-255>** {}
- Comando de OSPF: **no distance ospf**{}
- Comando: **router zebra** {}
- Comando: **no router zebra** {}

7.6 Mostrando la información de OSPF

- Comando: **show ip ospf** {}
- Comando: **show ip ospf interface [INTERFACE]** {}
- Comando: **show ip ospf neighbor**{}
- Comando: **show ip ospf neighbor INTERFACE** {}
- Comando: **show ip ospf neighbor detail**{}
- Comando: **show ip ospf neighbor INTERFACE detail** {}
- Comando: **show ip ospf database**{}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary)** {}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary)** ***link-state-id*** {}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary)** ***link-state-id*** **adv-router** ***adv-router*** {}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary) adv-router** ***adv-router*** {}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary) link-state-id self-originate**{}
- Comando: **show ip ospf database (asbr-summary|external|network|router|summary) self-originate** {}
- Comando: **show ip ospf database max-age**{}
- Comando: **show ip ospf database self-originate** {}
- Comando: **show ip ospf refresher** {}
- Comando: **show ip ospf route**{}

7.7 Depurando OSPF

- Comando: **debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (send|recv) [detail]** {}
- Comando: **no debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (send|recv) [detail]** {}
- Comando:**debug ospf ism** {}
- Comando: **debug ospf ism (status|events|timers)**{}
- Comando: **no debug ospf ism** {}
- Comando: **no debug ospf ism (status|events|timers)** {}
- Comando: **debug ospf nsm** {}
- Comando: **debug ospf nsm (status|events|timers)**{}
- Comando: **no debug ospf nsm** {}
- Comando: **no debug ospf nsm (status|events|timers)** {}
- Comando: **debug ospf lsa**{}
- Comando: **debug ospf lsa (generate|flooding|refresh)** {}
- Comando: **no debug ospf lsa**{}
- Comando: **no debug ospf lsa (generate|flooding|refresh)** {}
- Comando: **debug ospf zebra** {}
- Comando: **debug ospf zebra (interface|redistribute)** {}
- Comando: **no debug ospf zebra** {}
- Comando: **no debug ospf zebra (interface|redistribute)**{}
- Comando: **show debugging ospf** {}

8 Demonio de OSPFv3

ospf6d es un demonio que soporta la versión de OSPF 3 para redes en IPv6. OSPF v3 está descrito en la RFC2740.

8.1 Router OSPF6

- Comando: **router ospf6** {}
- Comando de OSPF6: **router-id *a.b.c.d*** {}
  - Configura el ID del router.
- Comando de OSPF6: **interface ifname area *area*** {}
  - Asigna el interfaz a un área específica, y empieza a mandar paquetes OSPF. Area puede ser específicado con 0.

8.2 Área OSPF6

El soporte de área de OSPFv3 todavía no está implementado.

8.3 Interfaz OSPF6

- Comando de Interfaz: **ipv6 ospf6 cost COSTE**{}
  - Configura el coste de salida del interfaz. El valor por defecto es 1.
- Comando de Interfaz: **ipv6 ospf6 hello-interval INTERVALOHELLO** {}
  - Configura el intervalo de envío de mensajes Hello, el valor por defecto es 40
- Comando de Interfaz: **ipv6 ospf6 dead-interval INTERVALODEAD**{}
  - Configura el intevalo de muerte del interfaz del router, por defecto este valor es de 40.
- Comando de Interfaz: **ipv6 ospf6 retransmit-interval INTERVALO DE RETRASMISION** {}
  - Configura el intervalo de retransmisión. Por defecto es 5.
- Comando de Interfaz: **ipv6 ospf6 priority PRIORIDAD** {}
  - Configura la prioridad del interfaz del router. Por defecto es 1.
- Comando de Interfaz: **ipv6 ospf6 transmit-delay RETARDORETRANSMISION** {}
  - Configura el Inf-Trans-Delay. El valor por defecto es 1.

8.4 Redistribución de rutas a OSPF6

- Comando de OSPF6: **redistribute static** {}
- Comando de OSPF6: **redistribute connected**{}
- Comando de OSPF6: **redistribute ripng** {}

8.5 Mostrar la Información de OSPF6

- Comando: **show ipv6 ospf6 [INSTANCIA_ID]** {}
  - INSTANCIA_ID es un ID opcional de instancia de OSPF. Para ver el router ID y la instancia ID, hay que escribir "show ipv6 ospf6".
- Comando: **show ipv6 ospf6 database**{}
  - Este comando muestra la base de datos de LSA. Es posible especificar el tipo de LSA.
- Comando: **show ipv6 ospf6 interface**{}
  - Para ver la configuración de los interfaces de OSPF como costes.
- Comando: **show ipv6 ospf6 neighbor** {}

  - Muestra el estado y el BDR elegido.

- Comando: **show ipv6 ospf6 request-list A.B.C.D**{}
  - Muestra la lista requerido de vecinos.

- Comando: **show ipv6 route ospf6** {}
  - Este comando muestra la información interna de la tabla de routing

9 Demonio de BGP-4

bgpd es un demonio de BGP-4 (Border Gateway Protocol 4, Protocolo de Pasarela Fronteriza 4). BGP-4 se describe en el RFC1771. bgpd también soporta las Extensiones Multiprotocolo para BGP-4 (tambien conocidas como BGP-4+ o MBGP) que se describen en el RFC2283.

BGP-4 es uno de los EGPs (Exterior Gateway Protocols, Protocolos de Pasarela Exterior) y se para el encaminamiento entre dominios.

9.1 Empezando con bgpd

El fichero de configuración por defecto de bgpd es bgpd.conf. bgpd busca en el directorio actual antes de buscar en /usr/local/etc/bgpd.conf. Todos los commandos de bgpd deben estar configurados en bgpd.conf.

Las opciones especificas de invocación de bgpd, se describen debajo. Las opciones communes deben ser también escificadas.

- -p *PUERTO*
- –bgp_port=*PUERTO*
  - Asigna el número puerto del protocolo BGP
- -r
- –retain
  - Caundo el demomio terminates, mantienen las rutas que bgpd a propagado a zebra.

9.2 Router BGP

Lo primero que se debe hacer es activar el encaminamiento BGP con el comando *router bgp*. Para configurar el encaminamiento BGP, necesitas un número de SA (AS number). El número de SA es una identificación de sistema autónomo. El protocolo BGP usa el número de SA para detectar si la conexión BGP es interna o externa.

El número de SA es un número entero entre 1 y 65535. Como usar un número se describe en el RFC1930. Los números de SA del 64512 al 65535 se definen como números de uso privado. Los SA privados no deben nunca anunciarse a la Internet global.

- Comando: **router bgp NÚMERO-SA** {}

  - Habilita el proceso de protocolo BGP con el número de SA especificado. Después de este comando, puedes introducir cualquier comando BGP. No pueden crearse un proceso BGP bajo un SA diferente a menos que se especifique Multi-instancia

- Comando: **no router bgp NÚMERO-SA** {}
  - Destruye un proceso BGP con el NÚMERO-SA especificado.
- Comando BGP: **bgp router-id ROUTER-ID** {}
  - Este commando especifica el router-ID (Indetificador de router). Si bgpd conecta con zebra, obtiene la información de los interfaces y de dirección. En ese caso el router-id por defecto se obtiene tomando la dirección de mayor numeración de todos los interfaces. Cuando router zebra no está habilitado, bgpd no puede obtener la información de los interfaces y router-id se fija en 0.0.0.0. En ese caso se debe especificar a mano.

9.2.1 Distancia BGP

- Comando BGP: **distance bgp <1-255> <1-255> <1-255>**{}
  - Este comando cambia el valor de la distancia de BGP. Cada argumento es un valor de distancia para rutas externas, rutas internas y rutas locales.
- Comando BGP: **distance <1-255> A.B.C.D/M**{}
- Comando BGP:**distance <1-255> A.B.C.D/M word** {}
  - Este comando configura el valor de la distancia a un desino.

9.2.2 Proceso de decisión de BGP

- Comprobación de pesos.
- Comprobación de preferencia local.
- Comprobación de ruta local.
- Comprobación de longitud del camino del Sistema Autónomo.
- Comprobación del origen.
- Comprobación de MED.

9.3 Red BGP

9.3.1 Ruta BGP

- Comando BGP: **network *A.B.C.D/M*** {}
  - Este comando activa el anuncio de esa red.

```
router bgp 1
network 10.0.0.0/8
```

Esta configuración de ejemplo nos dice que la red 10.0.0.0/8 será anunciada a todos los vecinos. Varios vendedores de routers no anuncian las rutas si estas no están presentes en sus tablas de encaminamiento IGP; *bgp* no tiene en cuenta su las rutas están anunciadas en las rutas IGP·

- Comando BGP: **no network *A.B.C.D/M*** {}
  - Desactiva el anuncio de prefijo previamente anunciado.

9.3.2 Agregación de Rutas

- Comando BGP: **aggregate-address *A.B.C.D/M*** {}
  - Este comando especifica que se agregen un conjunto de rutas recibidas en un PREFIJO menos especifico.
- Comando BGP:**no aggregate-address *A.B.C.D/M*** {}
  - Desactiva la agregación de prefijos.
- Comando BGP: **aggregate-address A.B.C.D/M summary-only** {}
  - Este comando especifica una dirección agregada. Las rutas agregadas no serán anunciadas
- Comando BGP: **no aggregate-address A.B.C.D/M** {}

9.3.3 Redistribución a BGP

- Comando BGP: **redistribute kernel** {}
  - Inyecta las rutas del kernel que no pertenecen a zebra en el proceso de BGP.
- Comando BGP: **redistribute static** {}
  - Inyecta las rutas estáticas de zebra en el proceso de BGP.
- Comando BGP: **redistribute connected** {}
  - Inyecta las rutas conectadas de zebra en el proceso de BGP.
- Comando BGP: **redistribute rip** {}
  - Inyecta las rutas de ripd en el proceso BGP.
- Comando BGP: **redistribute ospf** {}
  - Inyecta las rutas de ospfd en el proceso de BGP.

9.4 Vecino BGP

- Comando BGP: **neighbor *VECINO* remote-as *NÚMERO-SA*** {}
  - Crea un nuevo vecino cuyo SA es NÚMERO-SA. VECINO puede ser una dirección IPv4 ó IPv6.
- Comando BGP: **no neighbor *VECINO* remote-as *NÚMERO-SA*** {}
  - Elimina un vecino y toda la configuración asociada a él.

```
router bgp 1
 <verb>
neighbor 10.0.0.1 remote-as 2
```

En este caso mi encaminador, en AS-1, trata de conectar contra el encaminador de AS-2 con dirección 10.0.0.1.

Este debe ser el primer comando cuando se configura un vecino. Si remote-as no se especifica, bgpd se quejará de esta forma:

```
can't find neighbor 10.0.0.1
```

9.5 Configuración de vecinos

- Comando BGP: neighbor *VECINO* shutdown {}
- Comando BGP: no neighbor *VECINO* shutdown{}
  - Desactiva el vecino manteniendo la configuración asociada a este. Podemos desactivar un vecino con "no neighbor VECINO remote-as NÚMERO-SA pero a la vez borraremos la configuración asociada. Usa esta sintaxis cuando quieras tirar la sesión con un vecino pero preservando toda su configuración. Con la operación "no neighbor VECINO shutdown" activaremos la negociación de nuevo.
- Comando de BGP: **neighbor *VECINO* ebgp-multihop [TTL]** {}
- Comando de BGP: **no neighbor *VECINO* ebgp-multihop** {}
  - Activa el modo ebgp-multihop, usado para establecer sesisiones BGP entre sistemas de distintos SA, con no se encuentran directamente conectados al mismo segmento de red y en ciertas configuraciones de tunnel, gre e ipip. TTL establece el número de saltos al los que se encuentra el vecino, si no se especifica, el valor por defecto es el máximo, 255. Con "no" se desactiva la configuración ebgp-multihop.
- Comando de BGP: **neighbor *VECINO* description** {}
- Comando de BGP: **no neighbor *VECINO* description** {}
  - Establece/Elimina una descripción del vecino en texto libre.

- Comando de BGP: **neighbor *VECINO* version *VERSION*** {}
- Comando de BGP: **no neighbor *VECINO* version *VERSION*** {}
  - Fija la versión BGP del vecino que puede ser 4, 4+ o 4-. BGP version 4, es el valor por defecto para conecxiones BGP. BGP version 4+ significa que el vecino soporta Extensiones Multiprotocolo para BGP-4. BGP version 4- es similar pero el vecino habla Extensiones Multiprotocolo para BGP-4 en la antigua versión "Internet-Draft revision 00". Algún software de enrutamiento todavía lo usa.
- Comando de BGP: **neighbor *VECINO* interface *NOMBRE_IF*** {}
- Comando de BGP: **no neighbor *VECINO* interface *NOMBRE_IF*** {}
  - Cuando conecta con un vecino BGP sobre una dirección IPv6 de enlace-local, debes especificar el nombre del interfaz usado para esa conexión.
- Comando BGP: **neighbor *VECINO* next-hop-self** {}
- Comando BGP: **no neighbor *VECINO* next-hop-self** {}
  - Este comando fuerza al encaminador a anunciarse como el próximo salto, para las rutas que distribuya a sus vecinos.
- Comando BGP: **neighbor *VECINO* update-source *NOMBRE_IF*** {}
- Comando BGP:**no neighbor *VECINO* update-source*NOMBRE_IF*** {}
  - Con este comando, bgpd usará la dirección del interfaz designado para establecer la sesión BGP, es casi imprescindible cuando se usa ebgp-multihop.
- Comando BGP: **neighbor *VECINO* default-originate** {}
- Comando BGP: **no neighbor *VECINO* default-originate** {}
  - El comportamiento por defecto de bgpd es no anunciar la ruta por defecto (0.0.0.0/0), incluso si esta se encuentra en la tabla de rutas. Este comando anunciará la ruta (0.0.0.0/0) a ese vecino concreto, si existe, o le generará una.
- Comando BGP: **neighbor *VECINO* port *PUERTO*** {}
- Comando BGP: **no neighbor *VECINO* port *PUERTO*** {}
  - Especifica que puerto tcp se usará para establecer la sesión BGP con ese vecino si es distinto del puerto estándar (179).
- Comando BGP: **neighbor *VECINO* send-community** {}
- Comando BGP: **no neighbor*VECINO* send-community** {}
  - Instruye al demonio BGP a enviar las comunidades, asociadas a los distintos prefijos, este el comportamiento por defecto de bgpd, por lo que, debemos instruirle "no neighbor VECINO send-community" para deshabilitar el envío de comunidades.
- Comando BGP: **neighbor *VECINO* weight *PESO*** {}

- Comando BGP: **no neighbor *VECINO* weight *PESO*** {}
  - Este comando especifica un peso por defecto a las rutas aprendidas de ese vecino. (Es específico de Cisco).
- Comando BGP: **neighbor *VECINO* maximum-prefix *NÚMERO*** {}
- Comando BGP: **no neighbor *VECINO* maximum-prefix *NÚMERO*** {}
  - Este comando fija un tope máximo de prefijos que un vecino nos puede enviar, y se usa para evitar una inundación de prefijos que ponga en peligro la estabilidad de la red. Al llegar al número máximo de prefijos la sesión se cerrará hasta que el administrador, ejecute el comando "no neighbor VECINO shutdown"

9.6 Filtrado de Vecinos

- Comando BGP: **neighbor *VECINO* distribute-list *NOMBRE* [IN | OUT]** {}
- Comando BGP: **no neighbor *VECINO* distribute-list *NOMBRE* [IN | OUT]** {}
  - Este comando especifica una lista para el filtrado de actualizaciones enviadas a un vecino desde <*IN*> o hacia este <*OUT*>
- Comando BGP: **neighbor *VECINO* prefix-list *NOMBRE* [IN | OUT]** {}
- Comando BGP: **no neighbor *VECINO* prefix-list *NOMBRE* [IN | OUT]** {}
  - Este comando especifica una lista de prefijos, que se aceptan desde un vecino.
- Comando BGP: **neighbor *VECINO* filter-list *NOMBRE* [IN | OUT]**{}
- Comando BGP:**no neighbor *VECINO* filter-list *NOMBRE* [IN | OUT]** {}
  - Este comando especifica un filtrado de las rutas del vecino en base a su AS_PATH.
- Comando BGP:**neighbor *VECINO* route-map *NOMBRE* [IN | OUT]** {}
- Comando BGP: **no neighbor *VECINO* route-map *NOMBRE* [IN | OUT]** {}
  - Aplica un route-map a un vecino para un control avanzado de las rutas desde <*IN*> o hacia este <*OUT*>

9.7 Grupo de Vecinos de BGP

- Comando BGP: **neighbor *PALABRA* peer-group** {}
  - Este comando define un nuevo grupo de vecinos BGP.
- Comando BGP: **neighbor *VECINO* peer-group *PALABRA*** {}
  - Este comando relaciona un vecino específico a una palabra de peer-group.

9.8 Familia de Direccionamiento de BGP

9.9 Sistema Autónomo

SA (Sistema Autónomo) es uno de los elementos esenciales de BGP. BGP es un protocolo de encaminamiento vector distancia. El marco del SA proporciona el vector de distancia métrico y detección del lazo a BGP. La *RFC1930 - Guidelines for creation, selection, and registration of an Autonomous System (AS)* describe como utilizar el SA.

El SA es el valor digital del octeto. Así pues el rango valido es de 1 a 65535. Los SA definidos en el rango 64512 a 65535 están definidos como SA Privados y se deben utilizar para anunciar en la Internet Global.

9.9.1 Expresiones Regulares de Camino del SA

Las expresiones regulares de camino del SA pueden ser utilizadas para mostrar rutas BGP y listas de acceso (access list) del camino al SA. Las expresiones regulares del camino del SA están basadas en las expresiones regulares *POSIX 1003.2*. La siguiente descripción es sólo un subconjunto de las expresiones regulares POSIX. El usuario puede utilizar todas las expresiones regulares *POSIX*. Añadiendo a ese caracter especial un '_' para las expresiones regulares de caminio del SA.

- .
  - Busca un único caracter.
- *
  - Busca 0 o más ocurrencias en el patrón.
- +
  - Busca 1 o más ocurrencias en el patrón.
- ?
  - Busca 0 ó 1 ocurrencia en el patrón.
- ^
  - Busca el inicio de la línea.
- $
  - Busca el final de la línea.
- _ El Caracter _ tiene significados especiales en las expresiones regulares de camino del SA. Busca un espacio y una coma , y el SA pone el delimitador {and} y el delimitador de confederación del SA *(* y *)*. Y también busca el inicio de la línea y el final de la línea. Así _ puede ser utilizado para encontrar el valor de la frontera del SA. *show ip bgp regexp _7675_* busca en todas las rutas BGP las cuales incluyan el número de SA *7675*.

9.9.2 Mostrando Rutas BGP con Camino SA

Para mostrar las rutas BGP que tienen un SA específico utilizaremos el comando *show ip bgp*.

- Comando: **show ip bgp regexp*line*** {}
  - Muestra los prefijos que coinciden en su AS_PATH con la expresin regular introducida.

9.9.3 Listas de Acceso de Camino de SA

Las listas de acceso de camino del SA son definidas por el usuario.

- Comando:**ip as-path access-list *word*** {**permit**|**deny**} ***line*** {}
  - Este comando define una nueva lista de acceso de caminio al SA.
- Comando: **no ip as-path access-list *word*** {}
- Comando: **no ip as-path access-list *word*** {**permit**|**deny**} ***line*** {}

9.9.4 Utilización de Caminio de SA en un Route Map

- Route Map: **match as-path *word*** {}
- Route Map: **set as-path prepend *as-path*** {}

9.9.5 Números Privados de SA

9.10 Atributo de Comunidades BGP

El atributo de comunidades de BGP es ampliamente utilizado para implementar políticas de encaminamiento. Las operadoras de red pueden manipular el atributo de comunidades de BGP basándose en su propia política de red. El atributo de comunidades de BGP está definido en el RFC1997 - BGP Communities Attribute y en el *RFC1998* - An Application of the BGP Community Attribute in Multi-home Routing. Es un atributo opcional, sin embargo las políticas locales pueden navegar a través de diferentes sistemas autónomos.

El atributo de comunidades consiste en un conjunto de valores de las comunidades. Cada valor de comunidad tiene un valor con longitud de 4 octetos. El siguiente formato es utilizado para definir el valor de las comunidades.

- *AS:VAL*
  - Este firnati representa el valor de las comunidades de 4 octetos.*SA* son los dos primeros octetos de más peso en formato digital. *VAL* son los dos octetos con menor peso en formato digital. Este formato es útil para definir un valor orientado a la política del SA. Por ejemplo, *7675:80* puede ser utilizado cuando el SA 7675 quiere pasar el valor de política local con valor 80 a su vecino.
- *internet*
  - *internet* representa a las bien conocidas comunidades con valor 0.

- *no-export*
  - *no-export* represnta a las bien conocidas comunidades con valor *NO_EXPORT*
  - (0xFFFFFF01). Todas las rutas que lleven este valor no serán anunciadas fuera de los límites de la confederación BGP, así que la ruta será anunciada al vecino.
- *no-advertise*
  - *no-advertise* represnta a las bien conocidas comunidades con valor *NO_ADVERTISE*
  - (0xFFFFFF02). Todas las rutas que lleven este valor no serán anunciadas a otros vecinos BGP.
- *local-AS*
  - *local-AS* represnta a las bien conocidas comunidades con valor *NO_EXPORT_SUBCONFED*
  - (0xFFFFFF03). Todas las rutas que lleven este valor no serán anunciadas a vecinos externos BGP. Tanto si el encaminamiento de la vecindad es parte de la confederación, se considera como un vecino BGP externo, así la ruta no será anunciada a ese vecino.

Cuando el atributo de las comunidades BGP es recibido, los valores de comunidades duplicados serán ignorados y cada valor de las comunidades será ordenado en orden numérico.

9.10.1 BGP Community Lists

BGP community list es un atributo de lista de comunidades. BGP community list puede ser útil para encontrar o manipular atributos de comunidades en actualizaciones.

Existen dos tipos de community list. Una es standard community list y el otro es expanded community list. Standard community list define atributos de comunidades. Expanded community list define cadenas de atributos con expresiones regulares. Standard community list está compilado dentro del formato binario cuando el usuario lo define. Standard community list será directamente comparado al atributo de comunidades BGP en las actualizaciones de BGP.

Sin embargo la comparación es más rápida que explicar la community list.

- Comando: **ip community-list standard** ***name*** {**permit**|**deny**} ***community*** {}
  - Este comando define una nueva community list estándard. La community se compila dentro de la estructura de la comunidad. Podemos definir múltiples community list bajo el mismo nombre. En ese caso la correlación ocurrirá en el orden definido por el usuario. Una vez la community list se relaciona com el atributo de comunidades en las actualizaciones de BGP el retorno es permitido o denegado por la definición de la community list. Cuando no se relaciona con ninguna entrada, se devolverá denegado. Cuando la community está vacia no se relaciona con ninguna ruta.
- Comando: **ip community-list expanded** ***name*** {**permit**|**deny**} ***line*** {}
  - Este comando define una nueva community list expandida. *line* es una cadena que relaciona los atributos de comunidades en las actualizaciones de BGP.
- Comando:**no ip community-list** ***name*** {}
- Comando: **no ip community-list standard** ***name*** {}

- Comando: **no ip community-list expanded** *name* {}
  - Estos comandos borran la communoty list especificada por el nombre (*name*). Todas las comparticiones de las community lists serán borradas simplemente especificando el nombre de la community list.
- Comando: **show ip community-list** {}
- Comando: **show ip community-list** *name* {}
  - Este comando muestra información de la community list actual. Cuando el nombre se especifica la community list especificada se muestra.

```
# show ip community-list
Named Community standard list CLIST
permit 7675:80 7675:100 no-export
deny internet
Named Community expanded list EXPAND
permit :
# show ip community-list CLIST
Named Community standard list CLIST
permit 7675:80 7675:100 no-export
deny internet
```

9.10.2 BGP Community Lists Numeradas

Cuando se utiliza un número para el nombre de la community list, este tiene varios significados. Un número de community list entre 1 y 99 es una community list estándard. Un númerdo de community list en el rango de 100 a 199 es una community list expandida. Estas community lists son llamadas community list numeradas. Por otro lado las community lists normales son llamadas como community list nombradas.

- Comando: **ip community-list <1-99> {permit|deny}***community* {}
  - Este comando define una nueva community list. <1-99> es una community list numerada estándard. Si la community list está en este rango se trata de una community list estándard. Cuando la community estña vacía esta no se relaciona con ninguna ruta.
- Comando: **ip community-list <100-199> {permit|deny}** ***community*** {}
  - Este comando define una nueva community list. <100-199> se trata de una community list numerada expandida. Cuando la community esta vaía esta no se relaciona con ninguna ruta.
- Comando:**ip community-list name {permit|deny}** ***community*** {}
  - Cuando no se especifica el tipo de la community list este es automáticamente detectado. Si la community puede ser compilada en el atributo de las comunidades, la community list es una community list estándard. De otro modo está definida como community list expandida. Esta característica se deja para compatibilidad posterior. El uso de esta característica no está recomendado.

9.10.3 Communities Extendias BGP en un Route Map

- Route Map: **match extcommunity *word*** {}
- Route Map: **set extcommunity rt *extcommunity*** {}
  - Este comando configura el valor de la ruta destino.
- Route Map: **set extcommunity soo *extcommunity*** {}
  - Este comando configura el valor del sitio de origen.

9.11 Mostrando Rutas BGP

9.11.1 Show IP BGP

- Comando:**show ip bgp** {}
- Comando: **show ip bgp *A.B.C.D*** {}
- Comando: **show ip bgp *X:X::X:X*** {}
  - Este comando muestra las rutas BGP. Cuando no se especifica ruta se muestran todas las rutas BGP IPv4.

```
BGP table version is 0, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop        Metric LocPrf Weight Path
*> 1.1.1.1/32       0.0.0.0              0        32768  i
Total number of prefixes 1
```

9.11.2 Más Show IP BGP

- Comando:**show ip bgp *[PREFIJO]*** {}
  - Lista la tabla BGP. Con un prefijo, lista la información extendida del primer prefijo más especifico, que coincida con la entrada.
- Comando: **show ip bgp regexp*[AS-REGEXP]*** {}
  - Muestra los prefijos que coinciden en su AS_PATH con la expresión regular introducida.
- Comando:**show ip bgp community *community*** {}
- Comando: **show ip bgp community *community* exact-match** {}
  - This command display BGP routes using community.
- Comando: **show ip bgp community-list *word*** {}
- Comando: **show ip bgp community-list *word* exact-match** {}
  - This command display BGP routes using community list

- Comando: **show ip bgp summary** {}
  - Muestra los atributos generales de la tabla de BGP y el estado de los vecinos configurados.
- Comando: **show ip bgp neighbor*[VECINO]*** {}
  - Muestra un detalle extendido de un vecino y sus parámetros de configuración.
- Comando: **clear ip bgp *VECINO*** {}
  - Reinicia la sesión en frío (desde cero) con VECINO
- Comando: **clear ip bgp *VECINO* soft in** {}
  - Pide al vecino que reenvíe su toda su información de BGP.
- Comando: **show debug**{}
  - Muestra los procesos de depuración (debug) activos
- Comando: **debug event** {}
  - Activa el proceso de depuración de eventos BGP.
- Comando:**debug update** {}
  - Activa el proceso de depuración de actualizaciones BGP.
- Comando: **debug keepalive**{}
  - Activa el proceso de depuración de latidos BGP.
- Comando: **no debug event**{}
  - Deshabilita el proceso de depuración de eventos BGP.
- Comando: **no debug update** {}
  - Deshabilita el proceso de depuración de actualizaciones BGP.
- Comando: **no debug keepalive** {}
  - Deshabilita el proceso de depuración de latidos BGP.

9.12 Capacidad de Negociación

Cuando se añadió la capacidad de intercambio información de enrutamiento IPv6 a BGP, Existían varias propuestas. El grupo de trabajo IDR del IETF finalmente adoptó una propuesta llamada Extensión Multiprotocolo para BGP. La especificación se describe en el RFC2283 que no define un nuevo protocolo, sino nuevos atributos al BGP existente. Cuando se usa para intercambiar información de enrutamiento IPv6 es llamado BGP-4+. Cuando es usado en el intercambio de encaminamiento BGP es llamado MBGP.

bgpd soporta las Extensiones Multiprotocolo para BGP, así que, si un vecino soporta el protocolo, *bgpd* puede intercambiar rutas IPv6 y/o multicast.

El BGP tradicional no incluye la funcionalidad de detectar la capacidades de un vecino en el manejo de otras rutas distintas de IPv4 unicast. Esto es un gran problema a la hora de usar Extensiones Multiprotocolo

para BGP en una red operativa. El borrador 'draft-ietf-idr-bgp4-cap-neg-04.txt' propone una característica llamada Capacidad de Negociación (Capability Negotiation). bgpd usa esta Capacidad de Negociación para detectar las posibilidades de un vecino. Si el vecino está sólo configurado como vecino IPv4 unicast, bgpd no envía el paquete de Capacidad de Negociación.

Por defecto, zebra iniciará la sesión con la mínima capacidad común de ambos lados, por ejemplo, el encaminador local tiene la capacidad de unicast y multicast y el encaminador remoto tiene sólo capacidad unicast. En este caso, el encaminador local establecerá la sesión con capacidad de unicast únicamente, cuando no existan capacidades comunes se enviará un ERROR de Capacidad No Soportada y se reiniciará la conexión. Si quieres coincidir exactamente con las capacidades del vecino, debes usar el comando *neighbor VECINO strict-capability-match*.

- Comando BGP: **neighbor *VECINO* strict-capability-match** {}
- Comando BGP: **no neighbor *VECINO* strict-capability-match**{}
  - Compara de forma estrictamente las capacidades del vecino con las locales, si no coinciden se envía un ERROR de Capacidad no Soportada y se reinicia la sesión.

Si deseas deshabilitar el envió paquete de APERTURA (OPEN) de negociación de capacidades, cuando el vecino no soporta capacidad de negociación, usa el *comando neighbor VECINO dont-capability-negotiate*, para deshabilitar esta funcionalidad.

- Comando BGP: **neighbor *VECINO* dont-capability-negotiate** {}
- Comando BGP: **no neighbor *VECINO* dont-capability-negotiate** {}
  - Suprime el envió del paquete APERTURA de Capacidad de Negociación. Este comando sólo afecta si hay más capacidades configuradas aparte de unicast.

Cuando el vecino no incorpora Capacidades de Negociación, este no enviará ninguna capacidad en absoluto, en este caso bgpd establece el vecino con las capacidades configuradas.

En el caso de que prefieras capacidades prefijadas localmente en lugar de capacidades negociadas, aunque el vecino remoto envíe Negociación de Capacidades, si el vecino se configura con override-capability, bgpd ignorará las capacidades recibidas y establecerá sesión con las configuradas localmente.

- Comando BGP: **neighbor VECINO override-capability** {}
- Comando BGP: **no neighbor VECINO override-capability**{}
  - Ignora el resultado de la Negociación de Capacidades, y establece sesión con las capacidades fijas en la configuración local.

9.13 Reflector de Rutas

- Comando BGP: bgp cluster-id *A.B.C.D* {}
- Comando BGP: neighbor *VECINO* route-reflector-client {}
- Comando BGP: no neighbor *VECINO* route-reflector-client {}

  - Establece al vecino como un cliente de reflector de rutas, en este caso un vecino iBGP (BGP interno), le enviará todas las rutas, tengan o no origen en propio encaminador, ignorando así el comportamiento por defecto de iBGP (no enviar rutas que no tiene origen en el encaminador local)

9.14 Servidor de Rutas

En un nodo de intercambio de Internet, muchos ISPs (Proveedores de Servicio de Internet), se encuentran conectados unos con otros a través de conexiones eBGP (BGP externo). Normalmente estas conexiones eBGP se hacen formando un mallado completo. Del mismo modo que en la formación de un mallado iBGP, este método tiene un problema de escalabilidad.

Este problema de escalabilidad es bien conocido, el Servidor de Rutas es un método para solventar este problema, el encaminador BGP de cada ISP establece sesión sólo con el Servidor de Rutas, este a su vez, sirve la información BGP a todos los demás encaminadotes BGP. Aplicando este método el número de conexiones BGP se reduce de C=(n*(n-1)/2) a C=(n).

A diferencia de los encaminadores BGP normales, el Servidor de Rutas debe tener muchas tablas de rutas distintas para manejar cada una de las diferentes políticas de enrutamiento de cada portavoz BGP. Llamamos a cada una de estas tablas una vista (view) diferente. bgpd puede trabajar como un encaminador BGP normal, como un Servidor de Rutas o como los dos a la vez.

9.14.1 Multi-instancia BGP

Para habilitar la funcionalidad de *bgpd* de múltiples vistas, debes activar la funcionalidad multiple-instance de antemano.

- Comando: **bgp multiple-instance**{}
  - Habilita la funcionalidad BGP de múltiples instancias. Después de ejecutar este comando puedes establecer múltiples instancias BGP o múltiples vistas.
- Comando: **no bgp multiple-instance** {}
  - Deshabilita BGP multi-instancia. No puedes deshabilitar multi-instancia, cuando exista más de una instancia o vista de BGP.

Cuando se quiere configurar al estilo de Cisco,

- Comando: bgp config-type cisco {}
  - Salida de configuración BGP compatible con Cisco.

Cuando se especifica bgp config-type cisco,

Se muestra "no synchronization". Se muestra "no auto-summary".

los argumentos "network" y "aggregate-address" se muestran como "A.B.C.D M.M.M.M"

Quagga: network 10.0.0.0/8 Cisco: network 10.0.0.0

Quagga: aggregate-address 192.168.0.0/24 Cisco: aggregate-address 192.168.0.0 255.255.255.0

El manejo de los atributos de comunidad también es distinto. Si no hay ninguna configuración especificada para el atributo de comunidad y se envía al vecino el atributo de comunidad extendido. Cuando el usuario manualmente deshabilita la característica del atributo de comunidad no se envía al vecino. Para envíar el atributo de comunidad el usuario tiene que especificar el comando "neighbor A.B.C.D send-community".

```
! router bgp 1 neighbor 10.0.0.1 remote-as 1 no neighbor 10.0.0.1 send-community !
! router bgp 1 neighbor 10.0.0.1 remote-as 1 neighbor 10.0.0.1 send-community !
```

- Comando: **bgp config-type zebra** {}
  - Estilo de Quagga de configuración de BGP. Este es por defecto.

9.14.2 Instancia y vista BGP

La instancia BGP es un proceso BGP normal. El resultado de la selección de rutas va la FIB del kernel. Puedes establecer diferentes SA a la vez, cuando BGP multi-instancia se encuentra habilitado.

- Comando: **router bgp *NÚMERO-SA*** {}
  - Crea una nueva instancia de BGP.

```
bgp multiple-instance
!
router bgp 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.2 remote-as 3
!
router bgp 2
neighbor 10.0.0.3 remote-as 4
neighbor 10.0.0.4 remote-as 5
```

La vista BGP es casi lo mismo que el proceso BGP normal con la diferencia de que el resultado de la selección de rutas no se instala en la FIB del kernel. La vista BGP es sólo para el intercambio de información de enrutamiento.

- Comando: **router bgp *NÚMERO-SA* view *NOMBRE*** {}
  - Crea una nueva vista de BGP. La variable NOMBRE puede ser cualquier palabra el resultado de selección de rutas de esta vista, no se instala en la tabla de rutas del kernel.

Con este comando se establece un Servidor de Rutas como se muestra debajo:

```
bgp multiple-instance
!
router bgp 1 view 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.2 remote-as 3
```

```
!
router bgp 2 view 2
neighbor 10.0.0.3 remote-as 4
neighbor 10.0.0.4 remote-as 5
```

9.14.3 Política de Encaminamiento

Se pueden establecer distintas políticas de encaminamiento para un mismo vecino. Por ejemplo, estableciendo filtros diferentes.

```
bgp multiple-instance
!
router bgp 1 view 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.1 distribute-list 1 in
!
router bgp 1 view 2
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.1 distribute-list 2 in
```

Esto significa que las actualizaciones del vecino 10.0.0.1 se dirigen a ambas vistas, la "1" y la "2". Cuando una actualización se inserta en la vista "1" la lista 1 se aplica para el filtrado. Del mismo modo, la actualización se inserta en la vista "2" y se aplica el comando distribute-list 2.

9.15 Viendo la Vista

Para mostrar la tabla BGP de una vista debes añadir al comando view NOMBRE.

- Comando: **show ip bgp view *NOMBRE*** {}
  - Muestra la tabla BGP de la vista NOMBRE

9.16 Como configurar una conexión al 6-bone

```
zebra configuration
===================
!
! Actually there is no need to configure zebra
!
bgpd configuration
==================
!
! Esto significa que rutas seleccionadas van a Zebra y de ahí a la FIB del kernel
!
router zebra
!
```

```
! BGP-4+ configuration
!
router bgp 7675
bgp router-id 10.0.0.1
!
ipv6 bgp network 3ffe:506::/32
ipv6 bgp neighbor 3ffe:1cfa:0:2:2a0:c9ff:fe9e:f56 remote-as AS-NUMBER
ipv6 bgp neighbor 3ffe:1cfa:0:2:2a0:c9ff:fe9e:f56 route-map set-nexthop out
ipv6 bgp neighbor 3ffe:1cfa:0:2:2c0:4fff:fe68:a231 remote-as AS-NUMBER
ipv6 bgp neighbor 3ffe:1cfa:0:2:2c0:4fff:fe68:a231 route-map set-nexthop out
!
ipv6 access-list all permit any
!
! Set output nexthop address.
!
route-map set-nexthop permit 10
match ipv6 address all
set ipv6 nexthop global 3ffe:1cfa:0:2:2c0:4fff:fe68:a225
set ipv6 nexthop local fe80::2c0:4fff:fe68:a225
!
! logfile ARCHIVO es obsoleto. Usa log file ARCHIVO
!
log file bgpd.log
!
```

9.17 Volcado de tablas y paquetes BGP

- Comando: **dump bgp all *ARCHIVO*** {}
- Comando: **dump bgp all *ARCHIVO INTERVALO*** {}
  - Vuelca todos los paquetes y eventos de BGP al archivo especificado.
- Comando: **dump bgp updates *ARCHIVO*** {}
- Comando: **dump bgp updates *ARCHIVO INTERVALO*** {}
  - Vuelca las actualizaciones al FICHERO.
- Comando: **dump bgp routes *ARCHIVO*** {}
- Comando: **dump bgp routes *ARCHIVO*** {}
  - Vuelca toda la tabla BGP al FICHERO. Este es un proceso intensivo.

10 VTY shell

vtysh es el interfaz de commandos integrado de Quagga.

Para usar vtysh, especifica –enable-vtysh para configurar el script. Para usar PAM en la autenticación usa la opción, –with-libpam para configurar el script.

vtysh sólo busca vtysh.conf, el archivo de configuración vtysh, en /usr/local/etc. Vtysh no busca en el directorio actual porque el fichero incluye opciones de autenticación de usuarios.

Actualmente, vtysh.conf tiene solo un comando.

```
!
username foo nopassword
!
```

Con esta configuración, el usuario "foo" no necesita en la autenticación para el uso de vtysh. Con PAM, vtysh usa los mecanismosn de PAM para la autenticación.

Si vtysh se compila si autenticación PAM, cada usuario puede usar vtysh sin autenticación.

11 Filtering

Quagga provee de muchas y flexibles capacidades de filtrado. El filtrado se usa tanto para el envío como la recepción de información de encaminamiento. Una vez el filtrado se define puede ser aplicado en cualquier dirección.

11.0.1 Access List (Lista de Acceso IP)

- Comando: **access-list *NOMBRE permit RED-IPV4*** {}
- Comando: **access-list *NOMBRE deny RED-IPV4*** {}
- El filtrado básico se hace con access-list como se muestra en el siguiente ejemplo.

```
access-list filter deny 10.0.0.0/9
access-list filter permit 10.0.0.0/8
```

11.0.2 Prefix List

ip prefix-list ofrece el modo más potente de filtrado basado en prefijos. Además de las funcionalidades de access-list, ip prefix-list permite especificar longitud de un rango de prefijos así como la especificación de número de secuencia. Estos permiten añadir o borrar prefijos en puntos arbitrarios de la lista especificando el número de secuencia.

Si se introduce no ip prefix-list, este actua como aceptar (permit). Si ip prefix-list y no se encuentran coincidencias en la lista, se aplica denegar (deny) por defecto.

- Comando: **ip prefix-list *NOMBRE* (permit|deny) *PREFIJO [le LON] [ge LON]*** {}
- Comando: **ip prefix-list *NOMBRE* seq NÚMERO (permit|deny) *PREFIJO [le LON] [ge LON]*** {} Puede crear ip prefix-list usando los siguientes comandos:

  seq

seq *NUMERO* se establece de forma manual o automatica. En el caso de establececer los números manualmente, el usuario puede elegir un número menor a 4294967295. En el caso de que sean asignados automáticamente, estos se incrementaran de cinco en cinco en cada lista. Si se crea una entrada en una lista con números de secuencia sin especificar un número de secuencia, automáticamente se asignará el siguiente múltiplo de cinco como número de secuencia. Por ejemplo, si una entrada en una lista tiene el número de secuencia 2 y se crea una entrada sin especificar un número de secuencia, a esta se le asignará el número de secuencia 5, si las entradas 2 y 7 ya existen, a la próxima entrada sin número de secuencia, se le asignará el número 10.

le

La opción le (less than, menor que), especifica la longitud del prefijo. Se aplicará coincidencia en la lista si el prefijo es menor o igual al prefijo especificado.

ge

La opción ge (greater than, mayor que), al igual que le, aplicará la coincidencia si el prefijo es mayor o igual al prefijo especificado en la lista.

Menor que o igual a un prefijo y mayor que o igual a un prefijo, pueden ser usandos conjuntamente. El orden de introducción de le y ge no tiene ningún efecto.

Crear una entrada con un número de secuencia diferente pero con una misma regla introducida anteriormente, dará como resultado un error. Sin embargo, si el número secuencial y la regla son iguales, no se producirá el error.

Si se crea una lista con el mismo número de lista que una lista anteriormente creada, la nueva lista sobrescribirá la anterior.

La coincidencia de prefijos se realiza siguiendo desde el menor al mayor número desecuencia. La búsqueda por la lista terminará con la primera coincidencia.

En el caso de no existir le o ge:

Version 0.85: La coincidencia se aplicará a toda la longitud de prefijos contenidos dentro un prefijo menos especifico.

Version 0.86 o posterior: En el caso de no existir le o ge, la longitud del prefijo debe coincidir exactamente con el especificado por la lista.

- Comando: **no ip prefix-list *name*** {}

ip prefix-list description

- Comando: **ip prefix-list name description *DESCRIPCIÓN*** {}
  - Se permite la adición de descripciones a una lista. Este comando añade una descripción a la lista.
- Comando: **no ip prefix-list name description *[DESCRIPCIÓN]*** {}
  - Borra la descripción de una lista de prefijos. Es posible usar este comando se incluir toda la descripción

ip prefix-list sequential number control

- Comando: **ip prefix-list sequence-number** {}
  - Con este comando se muestran los números secuénciales de una lista. Este es el comportamiento por defecto.
- Comando: **no ip prefix-list sequence-number** {}
  - Con este comando, los números secuénciales no se mostrarán.

Mostrando ip prefix-list

- Comando: **show ip prefix-list** {}
  - Muestra todas las listas.
- Comando: **show ip prefix-list *NOMBRE*** {}
  - Muestra todas entradas dentro de la lista con ese nombre.
- Comando: **show ip prefix-list *NOMBRE* seq *NÚMERO*** {}
  - Muestra la entrada con ese número en la lista con ese nombre.
- Comando: **show ip prefix-list *NOMBRE a.b.c.d/m*** {}
  - Si el atributo longer se usa, todas las entradas con prefijos iguales o mayores que el prefijo insertado se mostrarán. Si se usa first match, se mostrará la primera coincidencia.
- Comando: **show ip prefix-list *NOMBRE a.b.c.d/m longer*** {}
- Comando: **show ip prefix-list *NOMBRE a.b.c.d/m first-match*** {}
- Comando: **show ip prefix-list summary** {}
- Comando: **show ip prefix-list summary *NOMBRE*** {}
- Comando: **show ip prefix-list detail** {}
- Comando: **show ip prefix-list detail *NOMBRE*** {}

Reinicio de contadores en ip prefix-list

- Comando: **clear ip prefix-list** {}
  - Pone a cero todos los contadores para todas las listas. Puede ser tambien usado especificando un nombre y/o prefijo.
- Comando:**clear ip prefix-list *NOMBRE*** {}
- Comando: **clear ip prefix-list *NOMBRE a.b.c.d/m*** {}

12 Route Map

Route map es una función muy útil de Zebra. Existen dos atributos "match" (concordancia) y "set" (fijar) en route-map.

```
route-map test permit 10
match ip address 10
set local-preference 200
```

Esto significa que si una ruta concuerda con la access-list 10, su valor de local-preference se fija en 200.

12.0.3 Comando Route Map

- Comando: **route-map *NOMBRE* permit *PRIORIDAD*** {}

12.0.4 Comando Route Map Match

- Comando Route-map: **match ip address *ACCESS_LIST*** {}
  - Concuerda con la *ACCESS_LIST* especificada.
- Comando Route-map: **match ip next-hop *DIRECCIÓN_IPV4*** {}
  - Concuerda con la *DIRECCIÓN_IPV4* especificada
- Comando Route-map: **match aspath *AS_PATH*** {}
  - Concuerda con el *AS_PATH* especificado.
- Comando Route-map: **match metric *MÉTRICA*** {}
  - Concuerda con la *MÉTRICA* especificada
- Comando Route-map: **match community *COMMUNITY_LIST*** {}
  - Concuerda con la community_list especificada.

12.0.5 Comando Route Map Set

- Comando Route-map: **set ip next-hop *DIRECCIÓN_IPV4*** {}
  - Establece la dirección de nexthop en BGP.
- Comando Route-map: **set local-preference *LOCAL_PREF*** {}
  - Estable la local preference de BGP.
- Comando Route-map: **set weight *PESO*** {}
  - Establece el PESO en BGP.
- Comando Route-map: **set metric *MÉTRICA*** {}

  - Establece la métrica.

- Comando Route-map: **set as-path prepend *AS_PATH*** {}
  - Establece como "prepends" el *AS_PATH* especificado.
- Comando Route-map: **set community *COMMUNITY*** {}
  - Marca con la *COMMUNITY* especificada..
- Comando Route-map: **set ipv6 next-hop global *DIRECCIÓN_IPV6*** {}
  - Establece la dirección "next-hop" IPv6 global en BGP-4+.
- Comando Route-map: **set ipv6 next-hop local *DIRECCIÓN_IPV6*** {}
  - Establece la dirección "next-hop" IPv6 local-link en BGP-4+.

13 Soporte para IPv6

Quagga soporta completamente el enrutamiento IPv6. Como se ha dicho anteriormente, Quagga soporta RIPng, OSPFv3 y BGP-4+. Puedes especificar direcciones IPv6 a un interfaz y configurar información de enrutamiento estático IPv6. Además de esto, Quagga-IPv6 provee de configuración de direccionamiento automático a través de una característica llamada address autoconfiguration. Para llevarlo acabo, el enrutador debe enviar mensajes de anuncio de enrutador "router advertisement" a todos los nodos que existan en la red.

13.1 Router Advertisement

Comando de interfaz:**ipv6 nd send-ra**{}

Comando de interfaz: **ipv6 nd prefix-advertisement *PRÉFIJO_IPV6*** {}

```
interface eth0
ipv6 nd send-ra
ipv6 nd prefix-advertisement 3ffe:506:5009::/64
```

Activa el anuncio del prefijo 3ffe:506:5009::/64 para la autoconfiguración de los nodos conectados a la red de cara al interfaz eth0.

14 Soporte SMNP

SNMP (Simple Network Managing Protocol) está ampliamente extendido debido a su capacidad de recolectar información de la red desde un router y/o un host. Zebra por si mismo no soporta la funcionalidad del agente SNMP. Pero en conjunción con el agente SNMP, Zebra porporciona la MIB para los protocolos de routing.

Zebra utiliza el protocolo SMUX (RFC1227) para establecer la comunicación con el agente SNMP. Existen varios agentes SNMP que soportan SMUX. Nosotros recomendamos utilizar el último software ucd-snmp

14.1 Cómo conseguir el ucd-snmp

ucd-snmp es un software libre distribuido con licencia. Por favor, comprueba la licencia que viene con la distribución del ucd-snmp. Los autores son de la "University of California, the University of California at Davis", y el "Electrical Engineering department at the University of California at Davis"

Se puede conseguir el ucd-snmp desde ftp://ucd-snmp.ucdavis.edu/.

Para habilitar el soporte para el protocolo SMUX hay que compilar el ucd-snmp como se muestra a continuación

```
% configure --with-mib-modules=smux
```

Después de compilar e instalar el ucd-snmp, hay que configurar el smuxpeer. La configuración que utilizamos es la que se muestra a continuación. Esto significa que las conexiones de los clientes SMUX se conectan a la MIB 1.3.6.1.6.3.1 con password test.

```
/usr/local/share/snmp/snmpd.conf
================================
smuxpeer 1.3.6.1.6.3.1 test
```

14.2 Configuración del SMUX

Para habilitar el soporte de SNMP en Zebra, hay que configurar Zebra con la opción –enable-snmp

- Comando: **smux peer *oid*** {}
- Comando: **no smux peer *oid*** {}
- Comando: **smux peer oid *password*** {}
- Comando: **no smux peer oid *password*** {}

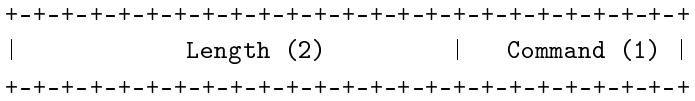
```
!
smux peer .1.3.6.1.6.3.1 test
!
```

15 Protocolo Zebra

El protocolo Zebra es el que se utiliza entre el demonio del protocolo y zebra. Cada demonio de protocolo envía las rutas al demonio Zebra. Entonces Zebra gestiona qué rutas están instaladas en la tabla de forwarding.

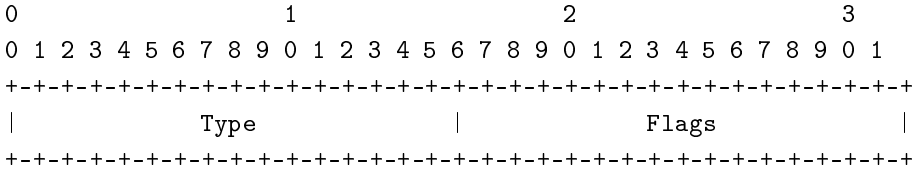
EL protocolo Zebra es un protocolo basado en TCP. Seguidamente se muestra la cabecera del protocolo Zebra.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```



La longitud es el total del paquete, incluyendo la cabecera. Así que la mínima cabecera es tres.

```
ZEBRA_INTERFACE_ADD 1
ZEBRA_INTERFACE_DELETE 2
ZEBRA_INTERFACE_ADDRESS_ADD 3
ZEBRA_INTERFACE_ADDRESS_DELETE 4
ZEBRA_INTERFACE_UP 5
ZEBRA_INTERFACE_DOWN 6
ZEBRA_IPV4_ROUTE_ADD 7
ZEBRA_IPV4_ROUTE_DELETE 8
ZEBRA_IPV6_ROUTE_ADD 9
ZEBRA_IPV6_ROUTE_DELETE 10
ZEBRA_REDISTRIBUTE_ADD 11
ZEBRA_REDISTRIBUTE_DELETE 12
ZEBRA_REDISTRIBUTE_DEFAULT_ADD 13
ZEBRA_REDISTRIBUTE_DEFAULT_DELETE 14
ZEBRA_IPV4_NEXTHOP_LOOKUP 15
ZEBRA_IPV6_NEXTHOP_LOOKUP 16
```

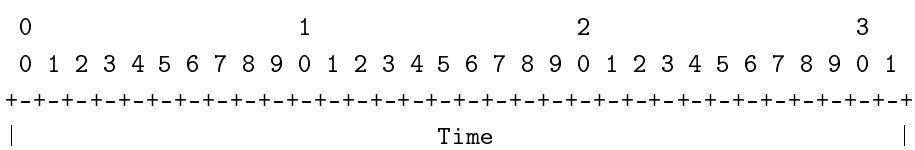


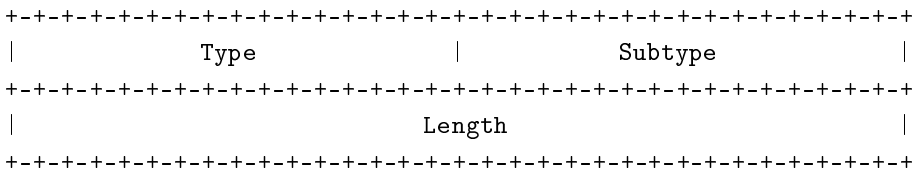
16 Formato de Descarga de Paquete Binario

Quagga puede descargar el paquete del protocolo de la encaminamiento en archivo con un formato binario.

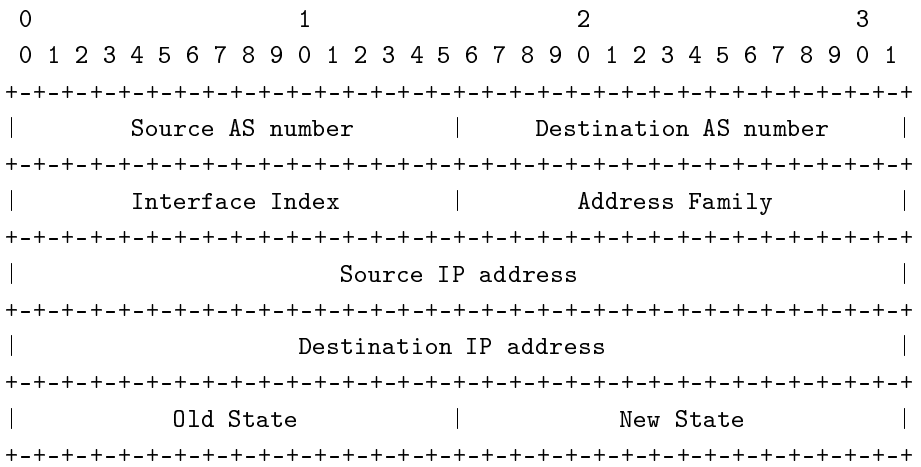
Parece ser mejor que compartimos el formato de la cabecera del MRT para la compatibilidad hacia atrás con los registros de la descarga de MRT. Debemos también definir el formato binario excepto la cabecera, porque debemos soportar las direcciones IP v4 y v6 como direcciones del socket y/o entradas del encaminamiento.

Aquí está el formato común de la cabecera. Éste es igual que el de MRT.



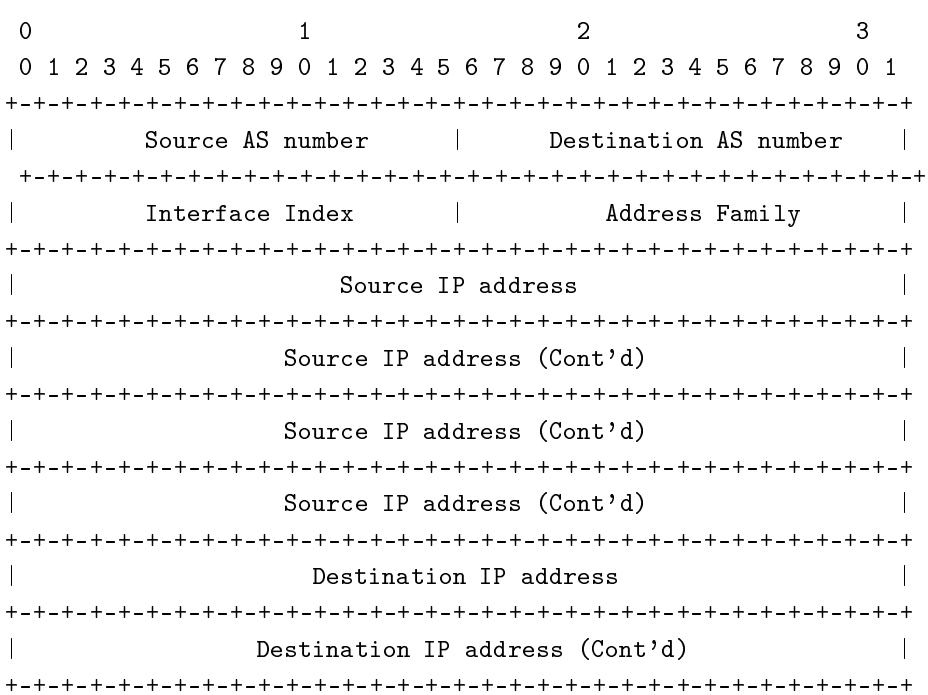


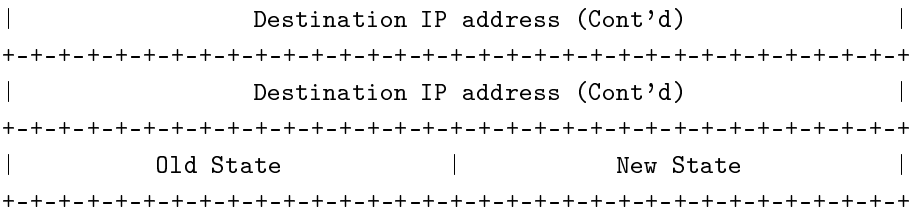
Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_STATE_CHANGE, y Address Family == IP (version 4)



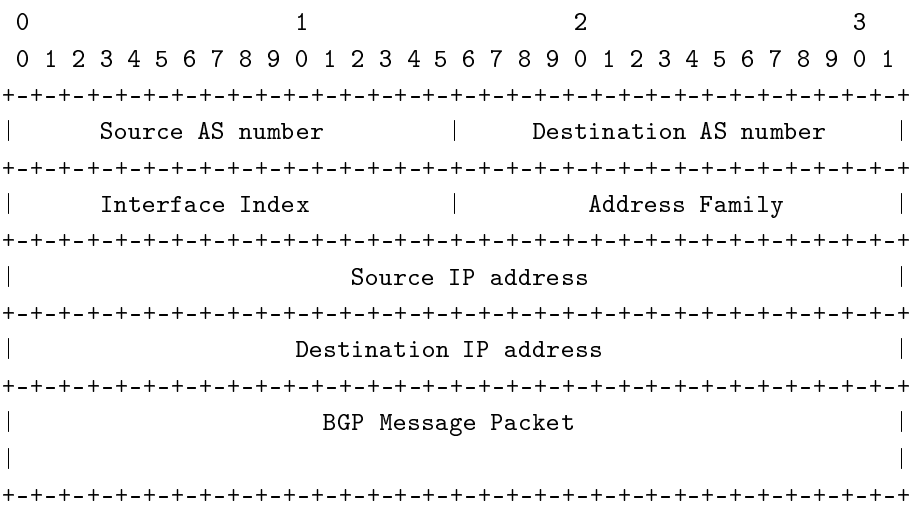
Donde State está definido en el RFC1771.

Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_STATE_CHANGE, y Address Family == IP version 6



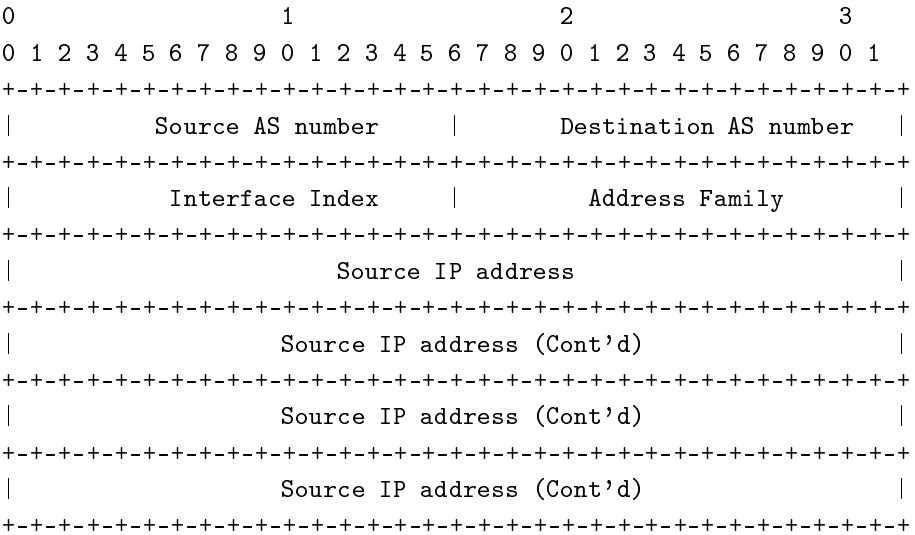


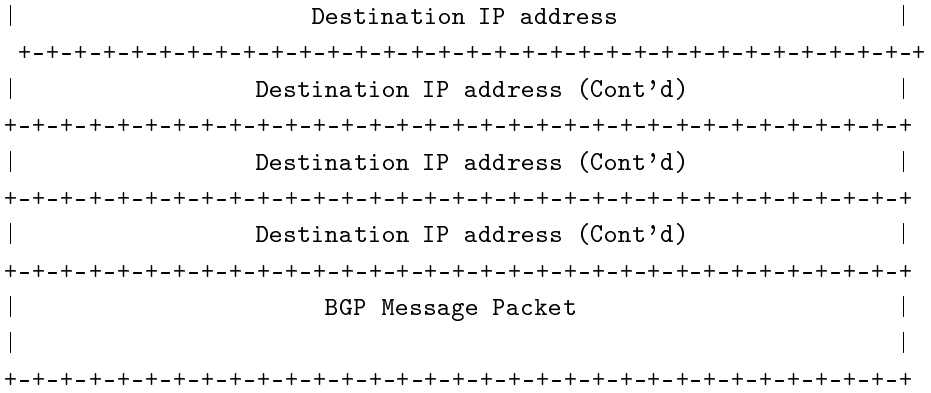
Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_MESSAGE, y Address Family == IP (version 4)



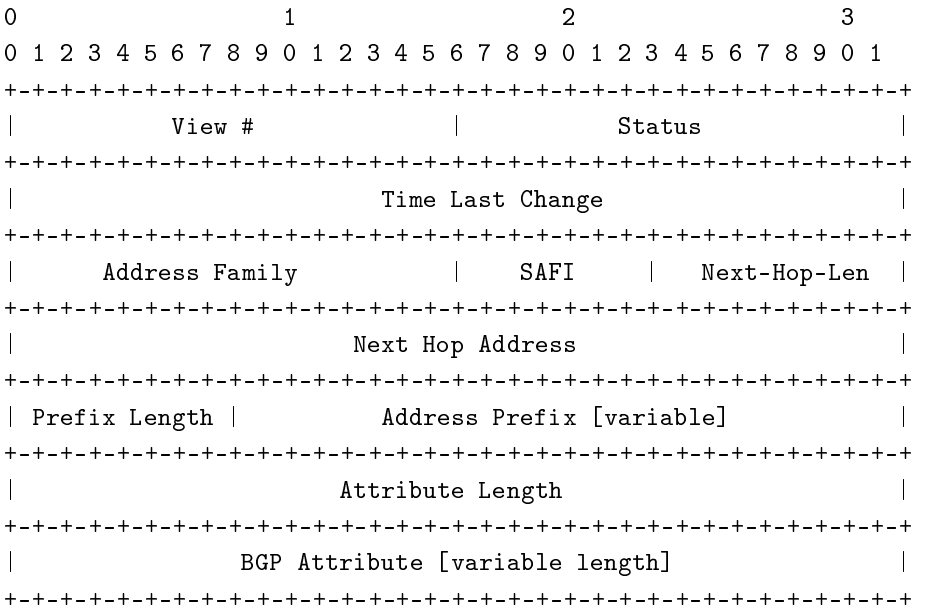
Donde el Paquete de Mensaje BGP es el total de contenido del mensaje BGP4 incluyendo la porción de cabecera.

Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_MESSAGE, y Address Family == IP version 6

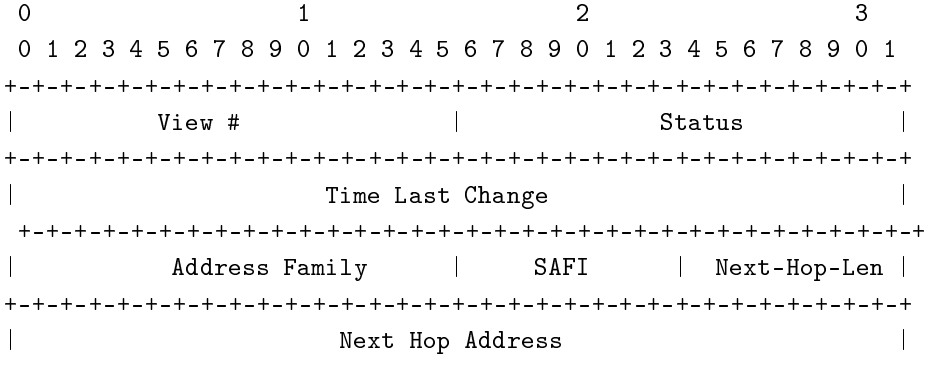


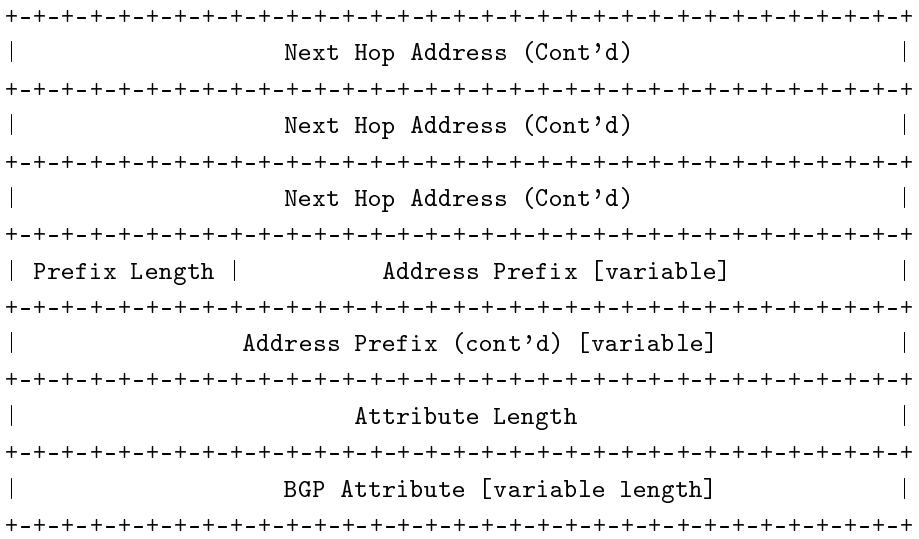


Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_ENTRY, y Address Family == IP (version 4)

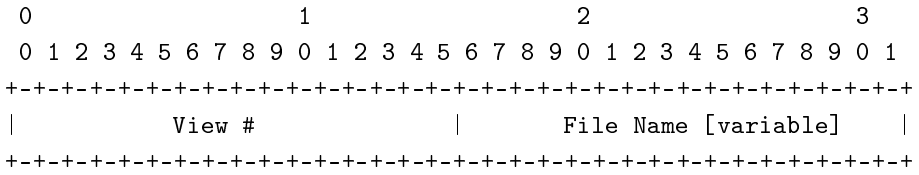


Si 'type' es PROTOCOL_BGP4MP, 'subtype' es BGP4MP_ENTRY, y Address Family == IP version 6





BGP4 Attribute no debe contener MP_UNREACH_NLRI. Si BGP Attribute tiene el campo MP_REACH_NLRI, éste debe tener una longitud de cero NLRI, por ejemplo, MP_REACH_NLRI tiene sólo Address Family, SAFI y valores de siguiente salto.
Si 'type' es PROTOCOL_BGP4MP y 'subtype' es BGP4MP_SNAPSHOT,



El fichero especificado en "File Name" contiene todas las entradas de routing, las cuales están en el formato de "subtype == BGP4MP_ENTRY".
Constantes:

```
/* type value */
#define MSG_PROTOCOL_BGP4MP 16
/* subtype value */
#define BGP4MP_STATE_CHANGE 0
#define BGP4MP_MESSAGE 1
#define BGP4MP_ENTRY 2
#define BGP4MP_SNAPSHOT 3
```