

## Práctica 6 – Enrutamiento básico con BGP

### 1- Objetivos

En esta práctica vamos a ver una configuración básica del protocolo de enrutamiento BGP4 en los routers Cisco y en PCs con sistema operativo GNU/Linux empleando el software Quagga.

### 2- Trabajo previo

Se recomienda localizar previamente documentación que haga referencia a cómo hacer la configuración para cada caso (IOS y Quagga). Las versiones de IOS en los routers del laboratorio son 12.3(1.6)T y 12.3(3.5)T. La versión de Quagga es la 0.99.20.1. No es necesario referirse a los comando de esas versiones exactas pues lo que vamos a usar es muy básico y seguramente no haya cambiado entre versiones cercanas.

### 3- Configurando un peering eBGP entre dos routers Quagga

Comenzaremos empleando los PCs como routers. Prepare la topología de la Figura 1. La Subred A va a representar un sistema autónomo con direccionamiento asignado 172.16.128.0/23 y la Subred B a otro con direccionamiento 172.17.0.0/24. PCA será el router frontera del primer AS y PCB del segundo. En el enlace entre los dos ASs empleamos 192.168.130.0/30.

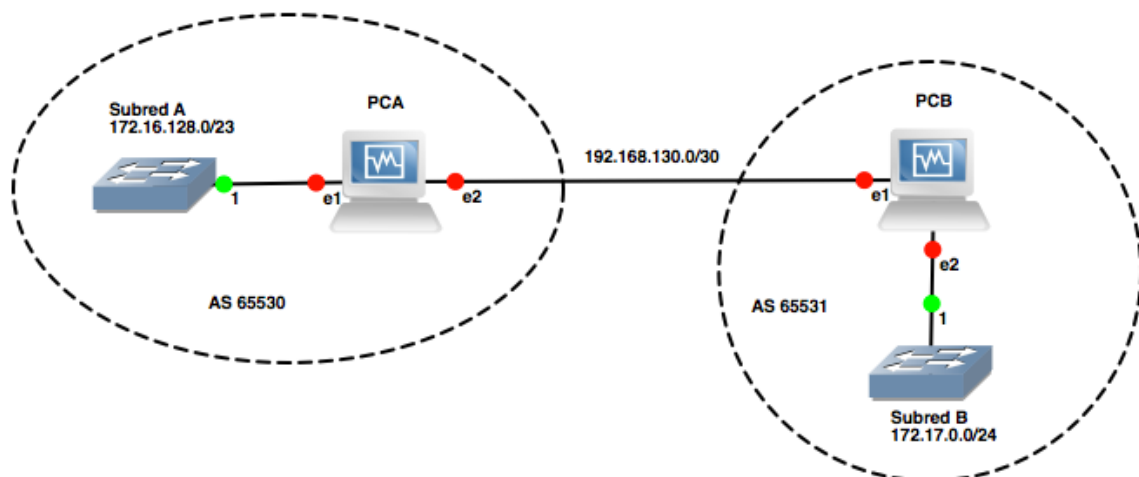


Figura 1 - Topología básica eBGP

Una vez hecha la configuración IP de los interfaces de ambos PCs puede activar el proceso BGP en el demonio bgpd mediante:

```
# router bgp <ASN>
```

Donde tendrá que indicar el valor de ASN para cada uno. El rango 64512-65534 está reservado para uso privado (no deben salir a la Internet). Emplee un ASN de ese rango para cada uno (lo cierto es que podría emplear cualquier valor, total, no vamos a hacer peering con un router en Internet).

Indique la subred interna en cada router para que se incluya en los anuncios.

```
(config-router)# network <red>/<longPrefijo>
```

Puede ver las rutas aprendidas mediante:

```
# show ip bgp
```

Todavía no se ha establecido la vecindad entre ambos routers. Para que establezcan la conexión TCP y negocien BGP por ella hay que indicarle a cada router con quién debe

establecerla (hasta que no esté hecho en ambos fallará). Para ver esto, ponga Wireshark a capturar en uno de los PCs, en el interfaz que le lleva al otro.

Asegúrese de que el identificador que va a usar cada PC es diferente. Lo puede ver en el running-config de bgpd en la directriz “bgp router-id” y en caso de que sean iguales en ambos PCs deberá cambiarlo en uno de ellos. También puede cambiarlo en la configuración de zebra pues bgpd lo obtiene de ahí.

Indique en un PC que quiere que establezca una conexión con el otro:

```
(config-router)# neighbor <direccionDelVecino> remote-as <ASNdelVecino>
```

Espere a ver en Wireshark cómo lo intenta y vea en qué punto falla.

Ahora haga la configuración simétrica en el otro PC.

Si ha funcionado correctamente podrá ver la tabla de rutas BGP de cada uno de ellos, con el camino a la subred del otro Sistema Autónomo, el cual es muy simple pues solo cuenta el ASN del destino. Analice el intercambio de mensajes del protocolo BGP.

#### 4- Configurando un escenario con tres Sistemas Autónomos

Configure el escenario de la Figura 2. En este caso añadimos un tercer sistema autónomo y el intermedio queremos que curse tránsito. Esto nos permitirá ver el AS\_PATH que por ejemplo aprende PC A para llegar a la Subred C.

El nuevo AS tiene asignado el direccionamiento 172.18.0.0/24 y parte de él se emplea en el enlace con el sistema de tránsito.

Analice el intercambio de mensajes BGP así como el estado de cada router y sus tablas de rutas y de BGP.

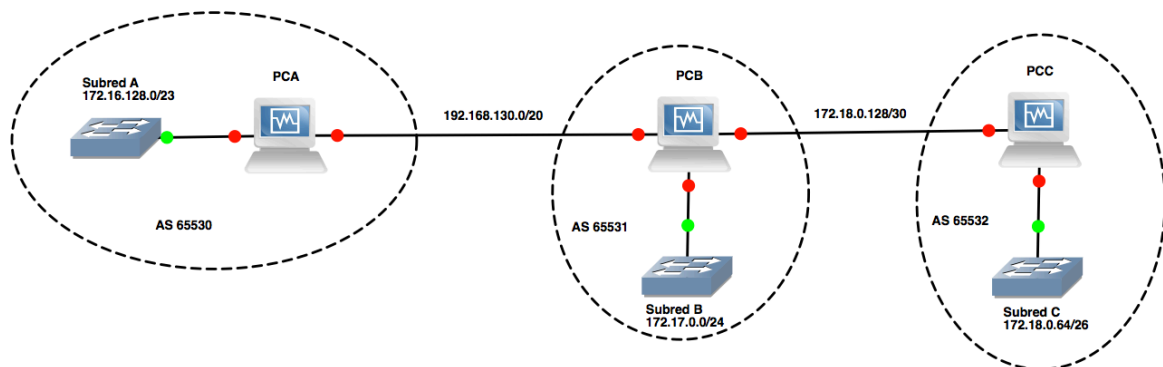


Figura 2 - Topología con 3 ASs

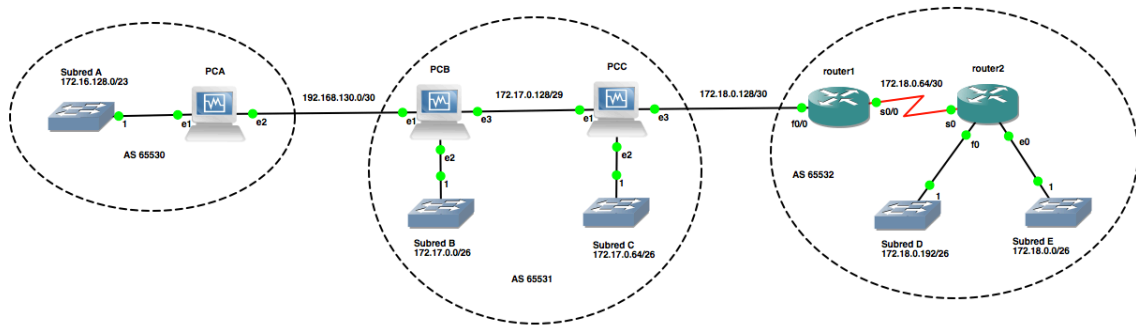
¿Puede hacer traceroute entre los dos ASs extremo? ¿Falta algo?

Pruebe a dejar de anunciar una ruta en BGP (comando “no network”) y vea los mensajes del *withdraw* de la ruta.

Punto de control (2/3): Muestre al profesor de prácticas este último escenario y que ha entendido el funcionamiento básico de BGP.

#### 5- Topología con eBGP, iBGP y OSPF

En esta ocasión vamos a extender el sistema autónomo de tránsito con dos routers frontera que empleen entre ellos iBGP (Figura 3), así como redistribución de rutas entre BGP y un IGP.

Figura 3 - iBGP en sistema de tránsito<sup>1</sup>

En el sistema autónomo de la derecha empleamos el router1 que es el único router Cisco del armario con una imagen de IOS que soporta BGP. Los comandos de configuración son análogos a los de Quagga. Haga que router1 pueda introducir en BGP anuncios de la red 172.18.0.0/24 mediante el comando “network”.

Haga que PC B introduzca en BGP la ruta a la Subred B y que PC C introduzca las rutas a la Subred C y al enlace entre PC B y PC C. Con eso PC C debería aprender la Subred A a través de iBGP. Sin embargo, la dirección que aprende como next-hop es la de PC A. Este es el funcionamiento normal de BGP, que por iBGP anuncia como NEXT\_HOP lo que le había venido por eBGP. El problema es que el *BGP speaker* (PC C) que recibe ese anuncio debe ser capaz de alcanzar a esa red. En este caso, PC C necesita saber cómo llegar a la red 192.168.130.0/30 pues le viene como next-hop una dirección de esa red. Esto es un problema que debería resolver el IGP.

Una solución sencilla para este caso concreto es pedirle a PC B que cuando haga los anuncios a PC C por iBGP lo haga cambiando el NEXT\_HOP por su dirección en la relación de vecindad. Esto lo puede hacer con:

```
(config-router)# neighbor <IP_PCC> next-hop-self
```

Hará falta algo parecido en PC C hacia PC B pues el mismo caso se da con lo que aprenda PC C del AS de la derecha.

Llegado este punto aún no hay comunicación entre la subred A y las subredes D y E (vea cómo están las tablas de rutas de los routers intermedios). Por ejemplo router2 no sabe cómo llegar a ninguna red que no tenga directamente conectada pues no participa en BGP ni router1 sabe llegar a las subredes D y E.

Para resolver esto vamos a correr OSPF dentro de este tercer sistema autónomo. Su configuración es sencilla y solo hay que activarla en router1 y router2. Por otro lado no queremos que router1 haga anuncios OSPF hacia el sistema autónomo de tránsito. Esto lo puede evitar activando OSPF en router1 solo para el interfaz serie.

Una vez se establezca la adyacencia, router1 debería aprender por OSPF las rutas a las subredes D y E, sin embargo no hay nada que aprender para router2. Para que router2 aprenda rutas hacia el exterior necesitaríamos que router1 introdujera en OSPF las rutas que ha aprendido por BGP. Esto es lo que se llama hacer “redistribución” de rutas entre dos protocolos de encaminamiento. Vamos a optar por una alternativa un poco más simple que va a ser que router1 introduzca en el IGP (es decir, en OSPF) el anuncio de una ruta por defecto. Esto se puede hacer introduciendo una ruta por defecto estática en router1 hacia el

<sup>1</sup> La numeración de los interfaces serie de los routers Cisco puede variar

PC C y a continuación empleando este comando en la configuración OSPF de router1 para que la redistribuya:

```
(config-router)# default-information originate
```

Vea el nuevo LSA en la base de datos, así como la ruta por defecto que llega hasta router2 a través de OSPF.

Finalmente, podemos redistribuir las rutas del IGP en BGP con este comando dentro de la configuración de BGP del router1:

```
(config-router)# redistribute ospf <processID>
```

¿Funciona ahora la comunicación entre todas las subredes?

Entregable (1/3): Entregue un documento de un máximo de 2-3 páginas con la explicación de la configuración de red que ha hecho (comandos quagga e IOS) para este apartado y de cómo quedan las bases de datos y tablas de rutas ante esta topología.