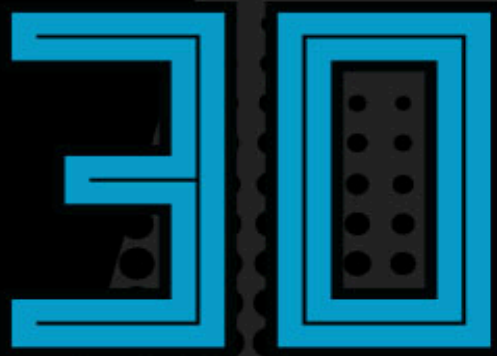


lacnic
lacnog 2018



24/28 SETIEMBRE 2018
ROSARIO • ARGENTINA

Práctica de BGP

Mariela Rocha

Erika Vega

Guillermo Cicileo

Ejercicio 1: Configurar las sesiones BGP

Levantar sesiones BGP entre los siguientes peers:

- ISP A y Cliente1
- ISP A y Cliente2
- A1 y A2
- A2 y B1

Ejercicio 2: anuncio de rutas

Configurar los routers del Cliente1 y Cliente2 para que anuncien las siguientes redes:

- Cliente1: 203.0.113.0/24
- Cliente2: 192.0.2.0/24

Ejercicio 3: analizar aprendizaje de prefijos

- ¿Las rutas anunciadas son aprendidas por A2?
- ¿Por qué cree que sucede esto?
- ¿Qué haría para revertir la situación?
- Configurar

Ejercicio 4: configuración de local pref

- Configurar al router A1 para que al recibir el prefijo 203.0.113.0/24 le configure un local_pref = 300.
- Verifique con qué local_pref se recibe el prefijo tanto en el router A1 como en A2.

Ejercicio 5: configuración de weight

- Realice la misma acción que con local_pref, pero para el comando Weight, configurándolo con valor 200.
- Verifique con qué Weight se recibe el prefijo en el router A2.

Ejercicio 6: analizar publicación de atributos

- Verifique en el router B1 cuáles prefijos son los que se anuncian a Internet y con qué atributos les recibe B1.

Ejercicio 7: configuración de filtros (prefix-list)

- Configure los routers del ISP para que los clientes 1 y 2 no se vean entre sí, es decir, para que un cliente no reciba las rutas del otro cliente. Hágalo utilizando filtros basados en direcciones IP (prefix-list).

Ejercicio 8: configuración de filtros (as-path)

- Aplique un filtro basado en el path, para que el Cliente2 no utilice al ISP A como AS de tránsito a Internet. Es decir, configure A2 para que no permita la salida a Internet del Cliente2.

Ejercicio 9: análisis de publicación a Internet

- Supongamos que:
 - El ISP acepta dar tránsito al Cliente2
 - Configura en el router A2 un filtro por AS-PATH
 - Permitiendo las rutas originadas en el AS64498
- ¿Qué problemas podrían surgir?

Ejemplos de configuración

Configuración Básica

1)_ Crea el proceso BGP dentro del router:

```
router bgp <ASN>
```

2)_ Definir contra qué peers estableceremos la sesión:

```
neighbor <ip> remote-as <AS-remoto>
```

- vecinos internos (mismo AS)
- vecinos externos (distinto AS)

Generalmente los externos comparten una subred, son adyacentes. Los internos (iBGP) pueden estar en cualquier parte del Sistema Autónomo.

Notar: En un IGP los vecinos son descubiertos automáticamente; en BGP se configuran en forma explícita

Configuración Básica

3)_ Definir las redes que serán anunciadas a través de BGP:

```
network <red> [mask <máscara>]
```

- *Se da de alta una red en la tabla de BGP y la marca como local al AS*
- *La máscara permite especificar supernetting o subnetting*

Filtros con prefix-list

Access-list (ACL)	Prefix-list
<pre>access-list <nro_access-list> permit deny ip <prefijo> <máscara de wildcard></pre>	<pre>ip prefix-list <nombre_prefix-list> <nro_seq> permit deny <red/prefijo> [ge length le length]</pre>
<pre>access-list 101 permit ip 10.0.0.0 0.255.255.255 access-list 101 permit ip 203.0.113.0 0.0.0.255 access-list 101 permit ip 192.0.2.0 0.0.0.255</pre>	<pre>ip prefix-list Entrada seq 5 deny 10.0.0.0/8 le 32 ip prefix-list Entrada seq 10 deny 203.0.113.0/24 le 32 ip prefix-list Entrada seq 15 permit 0.0.0.0/0 le 32</pre>

Cómo se aplica el prefix-list a la sesión BGP?

```
neighbor <ip-address | peer-group> prefix-list <nombre_prefix-list> in|out
```

Filtros con prefix-list - Ejemplo

Prefix-list

```
router bgp 64496
  neighbor 203.0.113.100 remote-as 65551
  neighbor 203.0.113.100 prefix-list PEER-IN in
  neighbor 203.0.113.100 prefix-list PEER-OUT out
!
ip prefix-list PEER-IN deny 198.51.100.0/24
ip prefix-list PEER-IN permit 0.0.0.0/0 le 32
ip prefix-list PEER-OUT permit 192.0.2.0/24
```

Filtros por AS-PATH

El filtro actúa según el **camino** hecho por los prefijos

- **Dos pasos:**

1. Crear sentencia con expresión regular.

```
ip as-path access-list <nro_filtro> permit|deny <regexp>
```

2. Aplicar el filtro

```
neighbor <IP_neighbor> filter-list <nro_filtro> in|out
```


Filtros con AS-PATH- Ejemplo

```
...  
neighbor 198.51.100.22 filter-list 10 in  
neighbor 198.51.100.22 filter-list 11 out  
...  
ip as-path access-list 10 permit ^$  
ip as-path access-list 11 deny 64496$  
ip as-path access-list 11 deny ^645  
ip as-path access-list 11 permit _64497_64498_  
...
```

Route-map – Ejemplo con prefix-list

```
router bgp 64496
  neighbor 203.0.113.10 route-map infilter in
  !
route-map infilter permit 10
  match ip address prefix-list HIGH-PREF
  set local-preference 120
  !
route-map infilter permit 20
  match ip address prefix-list LOW-PREF
  set local-preference 80
  !
route-map infilter permit 30
  !
ip prefix-list HIGH-PREF permit 192.0.2.0/25
ip prefix-list LOW-PREF permit 192.0.2.128/25
```

Route-map – Ejemplo con as-path

```
router bgp 64496
  neighbor 203.0.113.10 route-map filter-on-as-path in
  !
route-map filter-on-as-path permit 10
  match as-path 1
  set local-preference 80
  set weight 200
  set metric 127
  set next-hop 192.0.2.10
  !
route-map filter-on-as-path permit 20
  match as-path 2
  set local-preference 200
  set weight 500
  set metric 327
  set next-hop 192.0.2.100
  !
route-map filter-on-as-path permit 30
  !
ip as-path access-list 1 permit _64505$
ip as-path access-list 2 permit _64510_
```