

BGP

Border Gateway Protocol

Mariela Rocha

mariela.c.rocha@gmail.com



Conceptos basicos

- Protocolos de ruteo
- Sistemas Autónomos
- Tablas de protocolos y tabla de ruteo

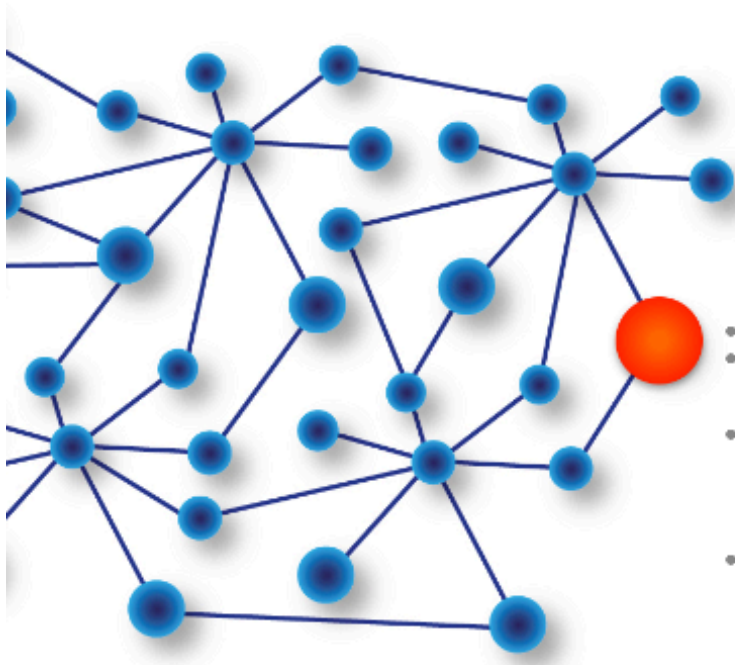
Protocolos de ruteo



Router: maneja múltiples protocolos de ruteo.

- **IGP** (Interior Gateway Protocol) - Protocolo de ruteo usado para intercambiar información de ruteo dentro de un sistema autónomo.
- **EGP** (Exterior Gateway Protocol) - Protocolo de ruteo usado para intercambiar información de ruteo entre sistemas autónomos.

Sistema Autónomo



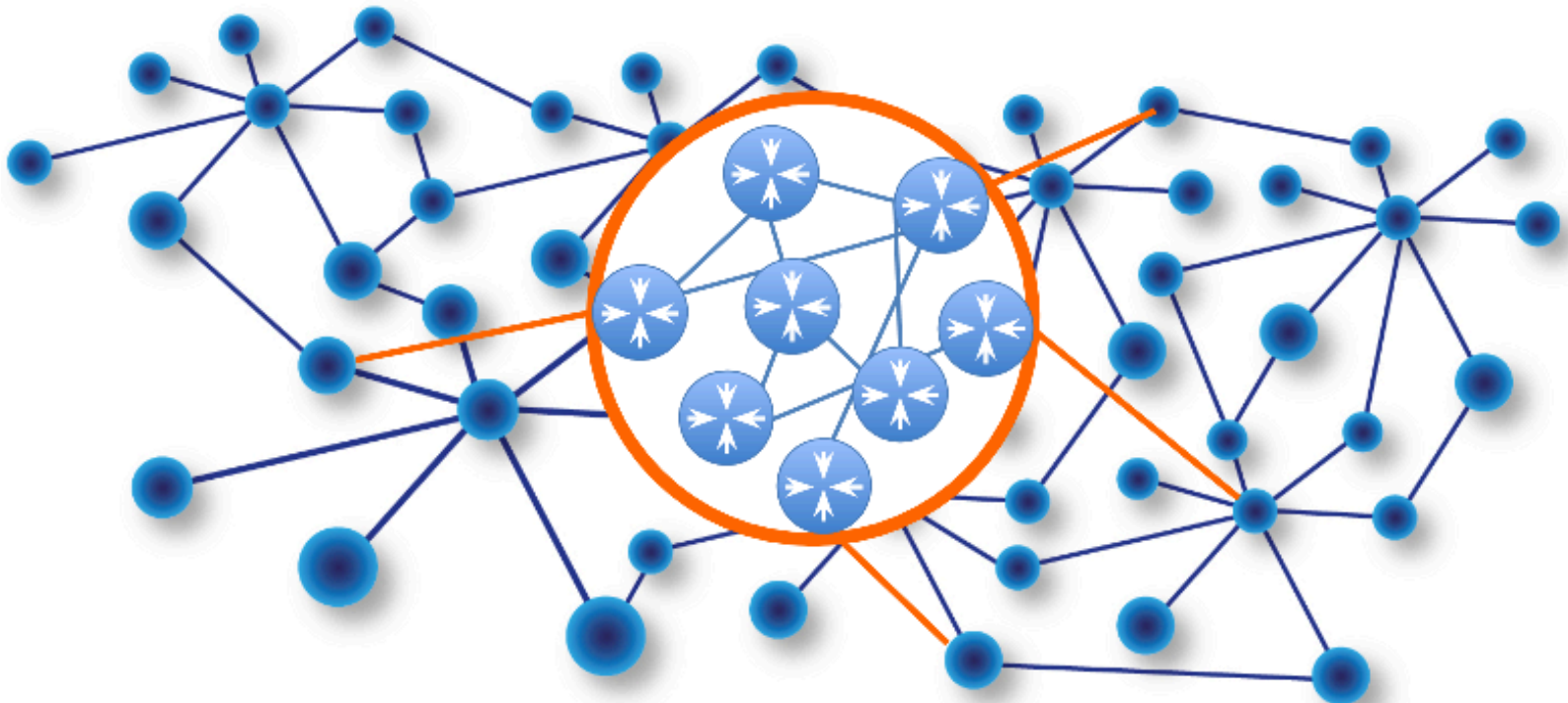
- Áreas a conectar
- Dispositivos
- “Visibles”
- Protocolos de Comunicación
- Direcciones IPs

**Política
de
Ruteo
Propia**

Sistema Autónomo: Grupo de redes IP que comparten una política de ruteo propia e independiente.

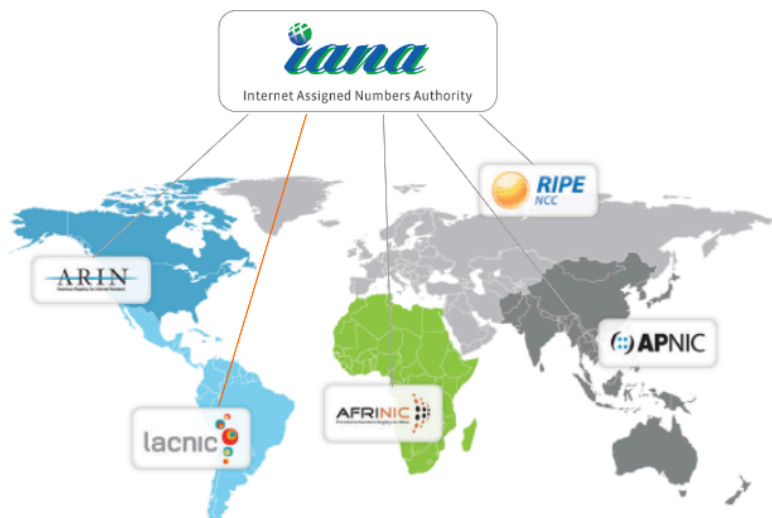
Sistema Autónomo

- Desde afuera el AS es visto como una entidad única.
- Tiene su propia política de ruteo interna (IGPs) y su propia política de ruteo externa (EGP)



Sistema Autónomo

- Cada AS tiene un identificador: **ASN** (Autonomous System Number).
- **ASN**: 16 o 32 bits
- Asignación:



Número de AS/bloque	Asignación
0 y 65535	Reservados
entre 1 y 64495	Internet Pública
entre 64496 y 64511	Documentación para ASN 16 bits – RFC 5737
entre 64512 y 65534	Uso sólo privado
entre 65536 y 65551	Documentación para ASN 32 bits – RFC 5398
entre 65552 y 4294967295	Internet Pública

Sistema Autónomo

Interconexión de Sistemas Autónomos



Tabla de ruteo y Tabla de BGP

- Existe una tabla con rutas por cada protocolo que el router maneja: tabla de IS-IS, tabla de RIP, tabla de BGP.

Tabla de	
Protocolo 1	
Destino	Vía

Tabla de	
IS - IS	
Destino	Vía

Tabla de	
Protocolo 2	
Destino	Vía

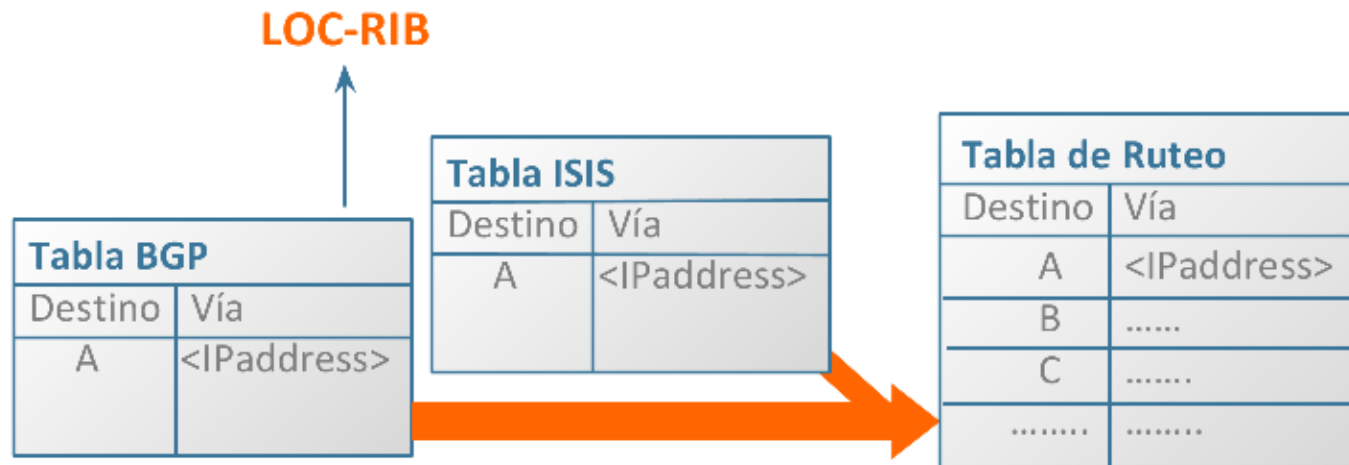
Tabla de	
OSPF	
Destino	Vía

Tabla de	
Protocolo 3	
Destino	Vía

Tabla de	
BGP	
Destino	Vía

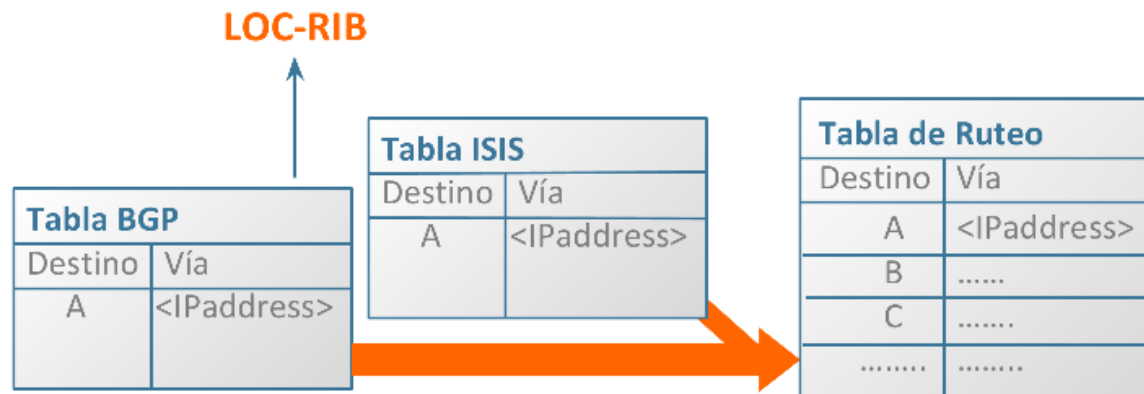
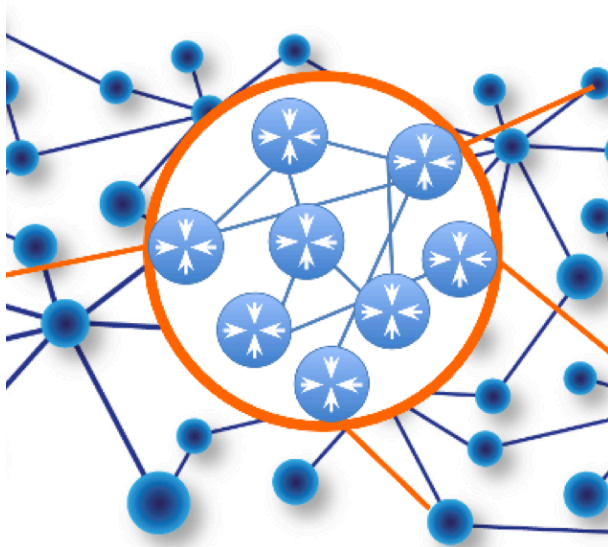
Tabla de ruteo y Tabla de BGP

- Los protocolos “compiten” para que sus rutas sean las que finalmente se utilicen (en función de una propiedad que en Cisco es denominada Distancia Administrativa).
 - Determina cuan confiable es un protocolo. Si hay dos rutas similares, se elige la de menor distancia administrativa. Cuanto menor es la distancia administrativa, más confiable es el protocolo.
- La ruta “elegida” es la que pasa a formar parte de la tabla de ruteo..



Importante: existen varias tablas de protocolos (una por cada uno de ellos) pero sólo una Tabla de Ruteo

¿Qué vimos?



BGP

- Qué es BGP?
- Cómo trabaja BGP?
- Configuración Básica

Qué es “Border Gateway Protocol” (BGP)?

EGP – Exterior Gateway Protocol

Opera intercambiando información de rutas y garantiza un camino libre de loops

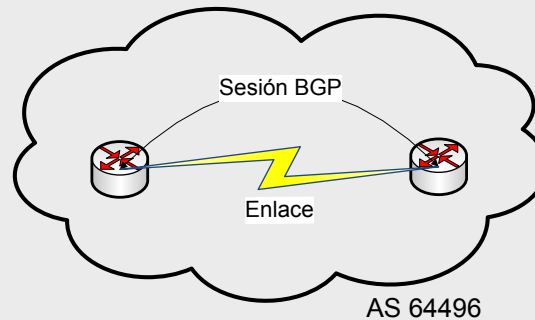
TIPOS de BGP

iBGP: implementado
dentro del AS

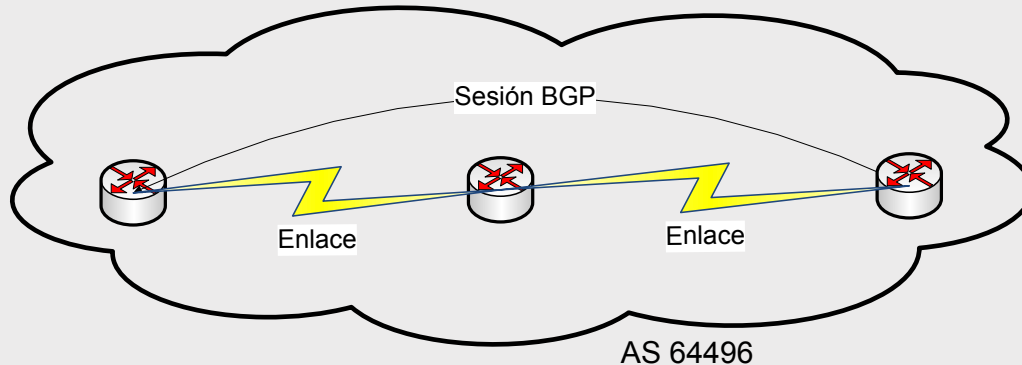
eBGP: implementado
fuera del AS

Cómo trabaja BGP?

- Usa TCP como protocolo de transporte (port 179)
- Se establece entre un par de routers (neighbors o peers) una sesión TCP abierta, mediante la cual intercambian información de ruteo BGP.

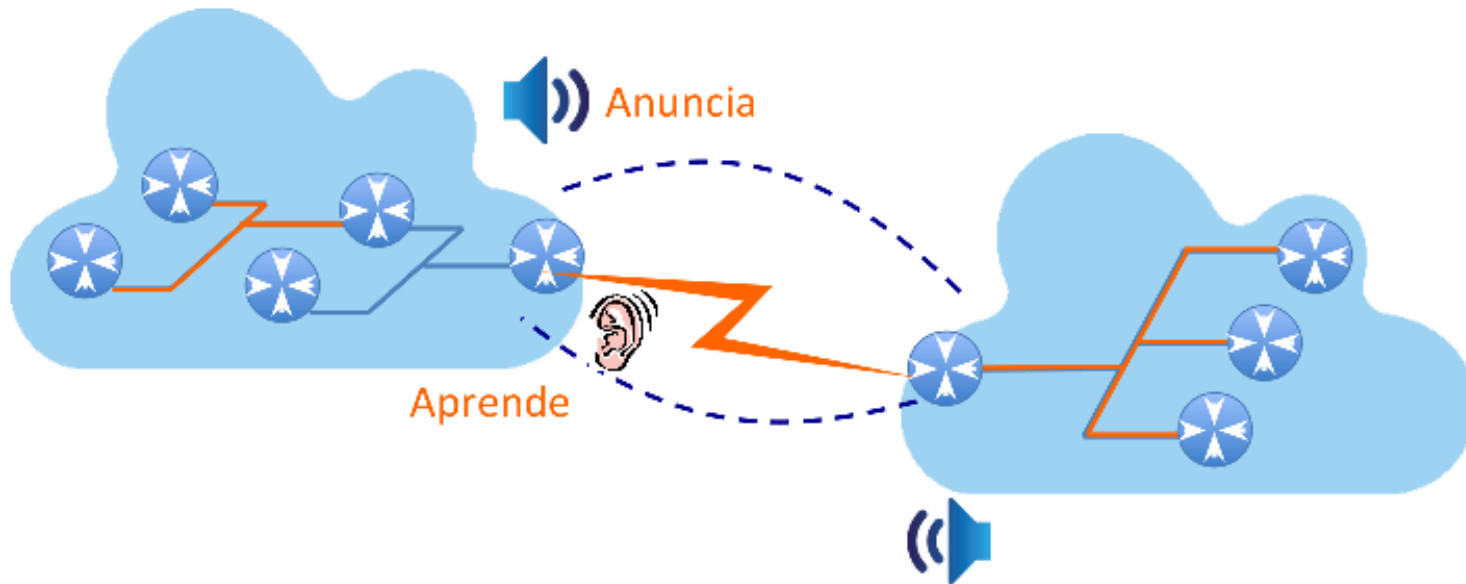


- Los peers BGP no necesitan estar directamente conectados.



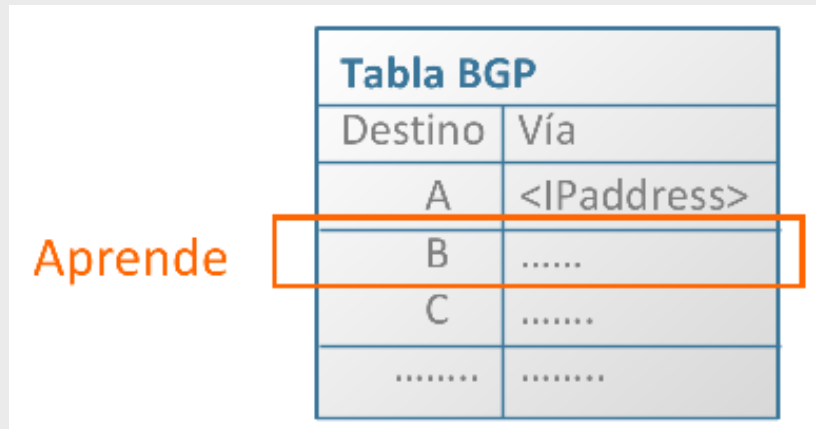
Cómo trabaja BGP?

APRENDER Y ANUNCIAR



Aprender y anunciar rutas

Aprender una ruta: significa que voy a incorporar en mi **tabla de BGP** alguna ruta que me están enseñando.



Destino	Vía
A	<IPaddress>
B
C
.....

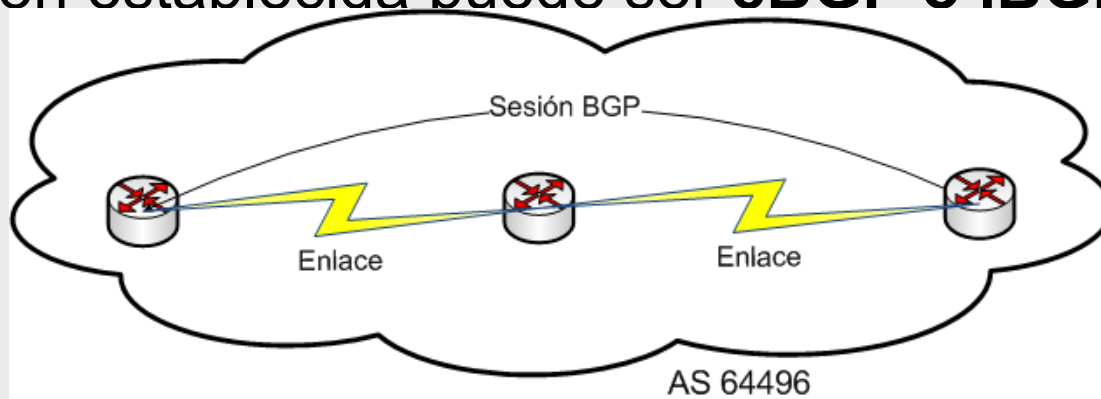
Anunciar una ruta: significa que le voy a decir a alguien que tengo una ruta para llegar a determinado destino, y que esa ruta está en la **tabla de ruteo**.



Destino	Vía
A	<IPaddress>
B
C
.....

Aprender y anunciar rutas (Resumiendo)

- **Neighbor:** punto remoto a quien voy a querer enseñar y/o de quien aprender rutas.
- Para hacer este pasaje de rutas debe establecerse una “**sesión BGP**”.
- Para que la sesión BGP se establezca, el potencial neighbor debe ser perfectamente **alcanzable por IP** (cuidado con filtros!).
- La sesión establecida puede ser **eBGP** o **iBGP**.



Aprender y anunciar rutas

- Qué tráfico afecta las rutas que aprendo?

saliente

- Qué tráfico afecta las rutas que enseño?

entrante

- Qué pasa si no aprendo nada?

Depende de lo que tenga configurado localmente en mi tabla de ruteo.

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*

Configuración Básica

- Ejemplo:

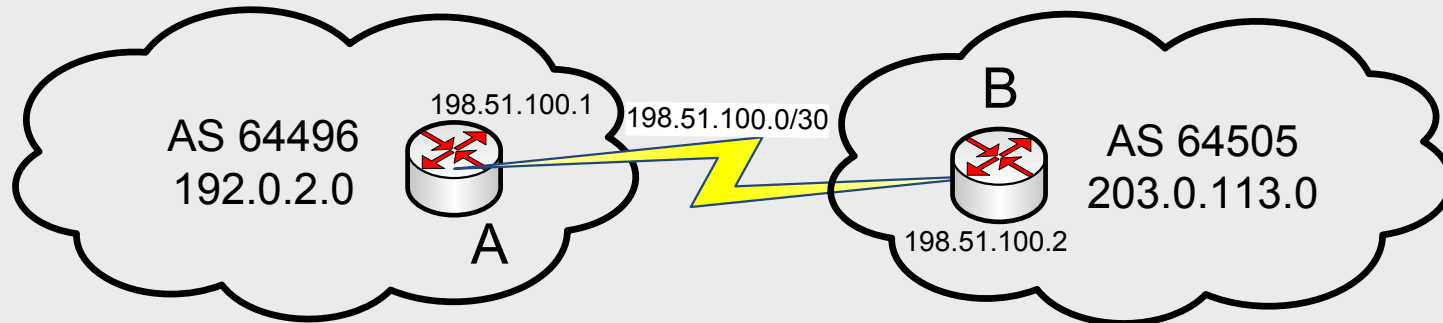
```
router bgp 64496
```

```
network 203.0.113.0 mask 255.255.255.0
```

```
neighbor 192.0.2.2 remote-as 64500
```

```
neighbor 192.0.2.22 remote-as 64505
```

Configuración Básica



Configuración Router A

```
router bgp 64496
network 192.0.2.0
neighbor 198.51.100.2 remote-as 64505
```

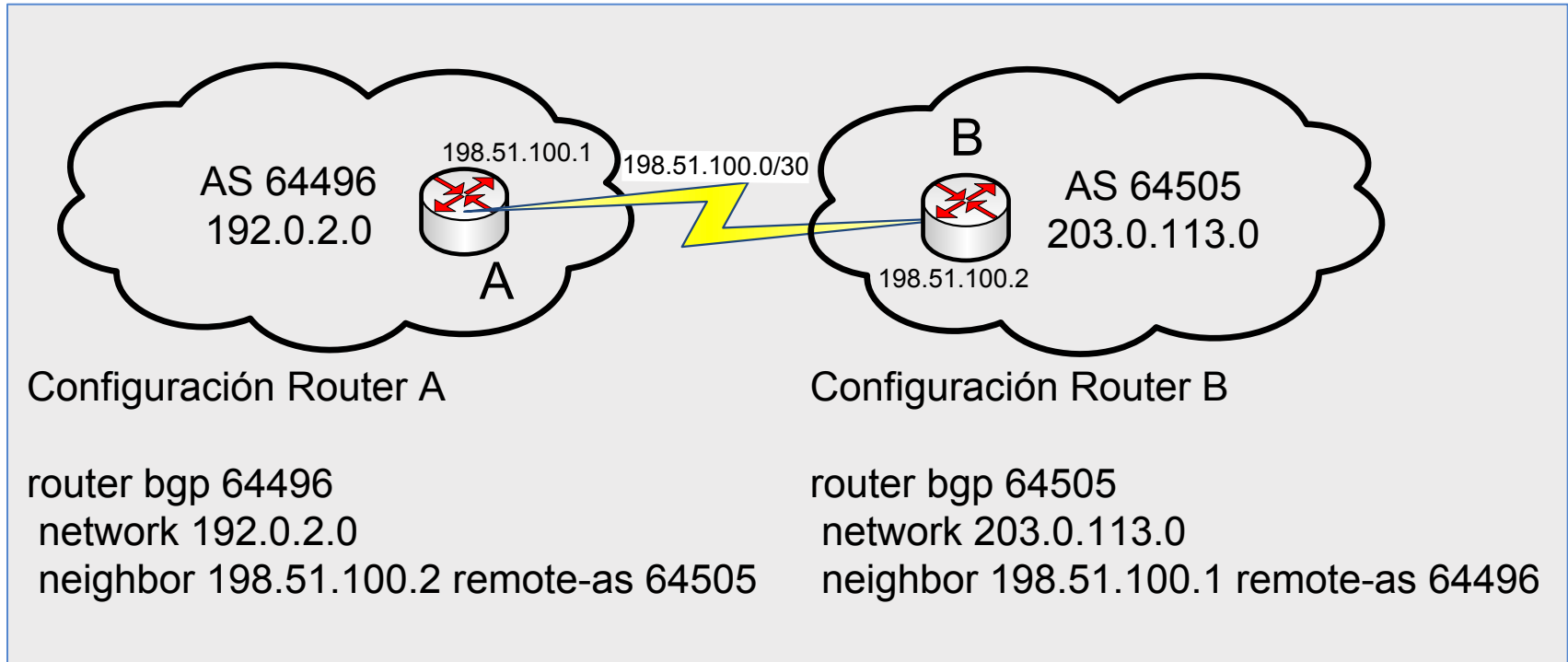
Configuración Router B

```
router bgp 64505
network 203.0.113.0
neighbor 198.51.100.1 remote-as 64496
```

Tener en cuenta:

- Las rutas a anunciar deben existir en la **tabla de ruteo** del router local o no serán enviadas en las actualizaciones.
- Las rutas aprendidas serán propagadas por defecto

Configuración Básica



Tener en cuenta:

- Para establecer más sesiones BGP simplemente se agregan más comandos “neighbor..”
- Ese comando sirve para identificar un vecino con el cual el router local establece la sesión.
- El argumento ASN determina si el router vecino es EBGP o IBGP.

Comandos: show ip bgp

- Las primeras tres columnas listan el status de cada ruta.
- Un * en la primer columna, indica que la ruta tiene un next-hop válido. Otras opciones:
 - "s" suppressed: BGP conoce la ruta, pero no está siendo anunciada, usualmente porque es parte de una ruta sumarizada.
 - "d" dampened: BGP detiene el anuncio de una ruta que produce un efecto denominado flapping (se levanta y se baja) demasiado rápido hasta que se estabilice por un periodo de tiempo.
 - "h" history: BGP conoce la red, pero no hay una ruta válida hacia ella.
 - "r" RIB* failure: la ruta es anunciada a BGP pero no es instalada en la tabla de ruteo. Esto puede suceder porque hay otro protocolo que tiene la ruta con una mejor distancia administrativa.
* RIB: Routing Information Base
 - "S" stale: indica que la ruta está "detenida" y requiere ser refrescada cuando se reestablezca la conexión con su vecino.

BGP table version is 134358, local router ID is 198.51.100.1

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

Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1		0		64496 65511 i
*>	192.0.2.128/26	198.51.100.1		0		64496 64506 i
*>i		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1		0		64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1				64496 ?

Comandos: show ip bgp

- **La segunda columna** tiene un signo mayor al lado de la ruta que fue seleccionada como el mejor camino hacia una red determinada.
- **La tercera columna** está en blanco, indicando que el router aprendió todas las rutas de un vecino externo. Una ruta aprendida de un vecino iBGP debería tener una i.

BGP table version is 134358, local router ID is 198.51.100.1

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

Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1			0	64496 65511 i
*>	192.0.2.128/26	198.51.100.1			0	64496 64506 i
*>i		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1			0	64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1			0	64496 ?

Comandos show ip bgp

- La cuarta columna lista las redes. La que no tiene una máscara de subred, usa la máscara classfull (A /8, B /16, C /24). Cuando el router aprende la misma red desde múltiples fuentes, sólo lista la red una vez.

BGP table version is 134358, local router ID is 198.51.100.1

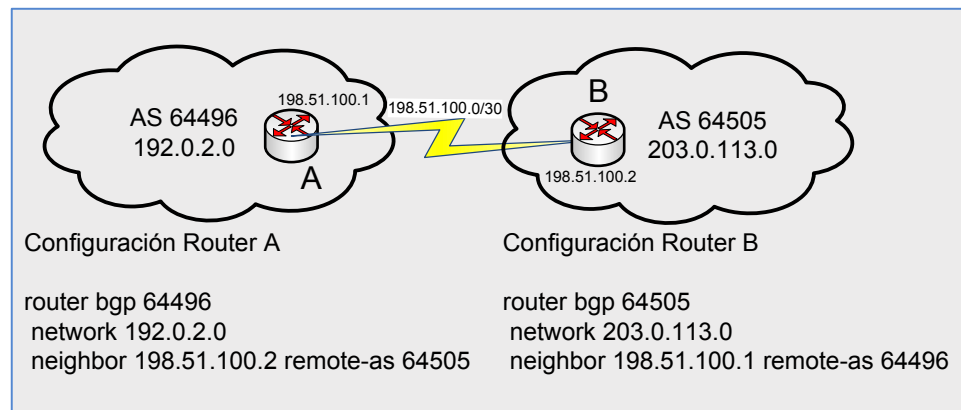
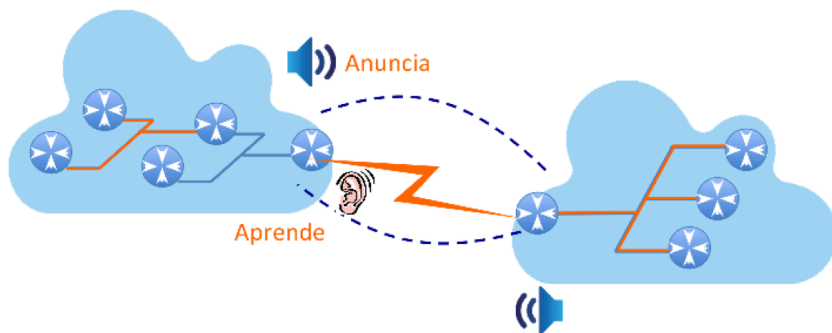
Status codes: s suppressed, d damped, h history, * valid, > best, i – internal, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1		0	64496	65511 i
*>	192.0.2.128/26	198.51.100.1		0	64496	64506 i
*>i		198.51.100.22	0	0	64506	e
*	203.0.113.128/25	198.51.100.1		0	64496	64501 i
*>		198.51.100.114	0	0	64501	i
*>	203.0.113.0/25	198.51.100.1		0	64496	?

¿Qué vimos?

APRENDER Y ANUNCIAR



BGP table version is 134358, local router ID is 198.51.100.1

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

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.0/26	198.51.100.1			0	64496 65511 i
*> 192.0.2.128/26	198.51.100.1			0	64496 64506 i
*>i	198.51.100.22	0	0		64506 e
* 203.0.113.128/25	198.51.100.1			0	64496 64501 i
*>	198.51.100.114	0	0		64501 i
*> 203.0.113.0/25	198.51.100.1			0	64496 ?

Atributos

- Definición
- Categorías de atributos
- Cómo se comportan

Atributos de rutas

Son parámetros preestablecidos que viajan junto a la información de la rutas.

Permiten poder ser manipulados por los administradores de redes para influir en las decisiones de BGP

Categorías de Atributos

(RFC 4271)

- 1. Mandatorios bien conocidos** (Well-known mandatory): son atributos que tienen que estar si o si, en forma mandatoria, en el paquete UPDATE de BGP. Debe poder ser reconocido por todas las implementaciones de BGP.
- 2. Discrecional bien conocido** (Well-known discretionary): son atributos que deben ser reconocidos por todas las implementaciones de BGP, pero pueden o no enviarse en el mensaje UPDATE de BGP.
- 3. Opcional transitivo** (Optional transitive): Si un atributo es transitivo, significa que BGP deberá aceptar y publicar el atributo, aun si este no es reconocido dentro de la implementación.
- 4. Opcional no transitivo** (Optional non-transitive): Si un atributo es opcional no transitivo, significa que el atributo, al ser recibido en el mensaje UPDATE, deberá ignorarse y no pasarse a otros peers BGP.

Atributos que veremos

Attribute Code	Type
ORIGIN	Well-known mandatory
NEXT--HOP	Well-known mandatory
AS_PATH	Well-known mandatory
MED ó Multi_EXIT_DISCriminator	Optional nontransitive
LOCAL_PREF ó Local Preference	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

Atributos de rutas: origin

- **Informa a todos los sistemas autónomos como fue introducido el prefijo de red**
- 3 valores: IGP, EGP, incomplete
 - i originada en un IGP, anunciada con “network”
 - e originada en un EGP (BGP a BGP)
 - ? origen desconocido, normalmente producto de una redistribución incompleta desde otro protocolo de ruteo

BGP table version is 134358, local router ID is 198.51.100.1

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

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1			0	64496 65511 i
*	192.0.2.128/26	198.51.100.1			0	64496 64506 i
*>		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1			0	64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1			0	64496 ?

Atributos de rutas: next-hop

- **Indica cuál es el próximo salto para alcanzar un destino**
 - No necesariamente es la IP del router directamente conectado.
- En un IGP, el next-hop (NH) es la IP del router que anunció la ruta. **En BGP no:**
 - **En las sesiones eBGP**, el NH es la IP del neighbor que anunció la ruta.
 - **En las sesiones iBGP:**
 - Rutas originadas dentro del AS: NH es la IP del router que la origina.
 - Rutas incorporadas por eBGP: se transporta inalterado el NH aprendido por eBGP.
- **Un next-hop 0.0.0.0 indica que el router local originó la ruta.**

BGP table version is 134358, local router ID is 198.51.100.1

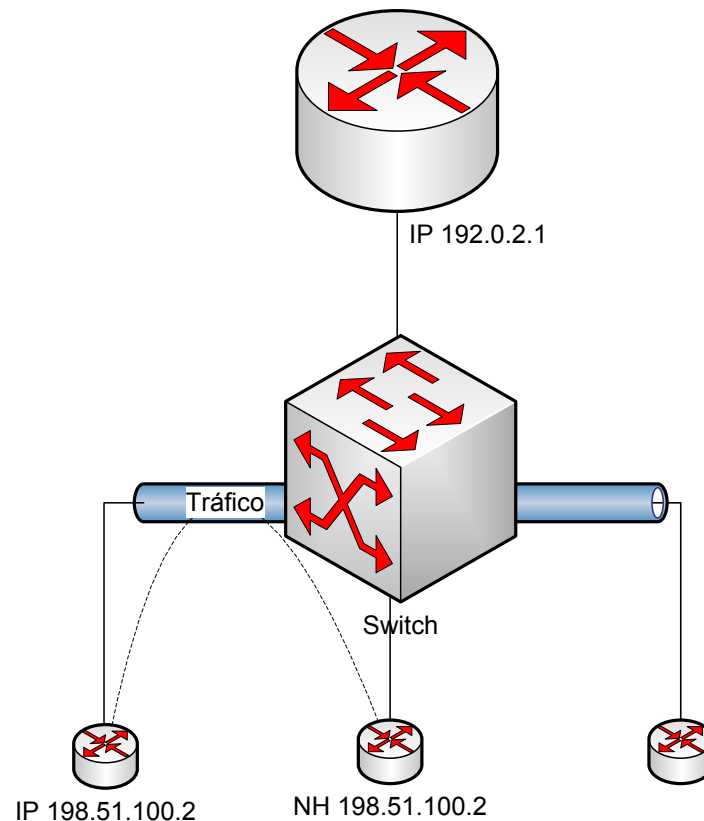
Status codes: s suppressed, d damped, h history, * valid, > best, i – internal,
S Stale

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

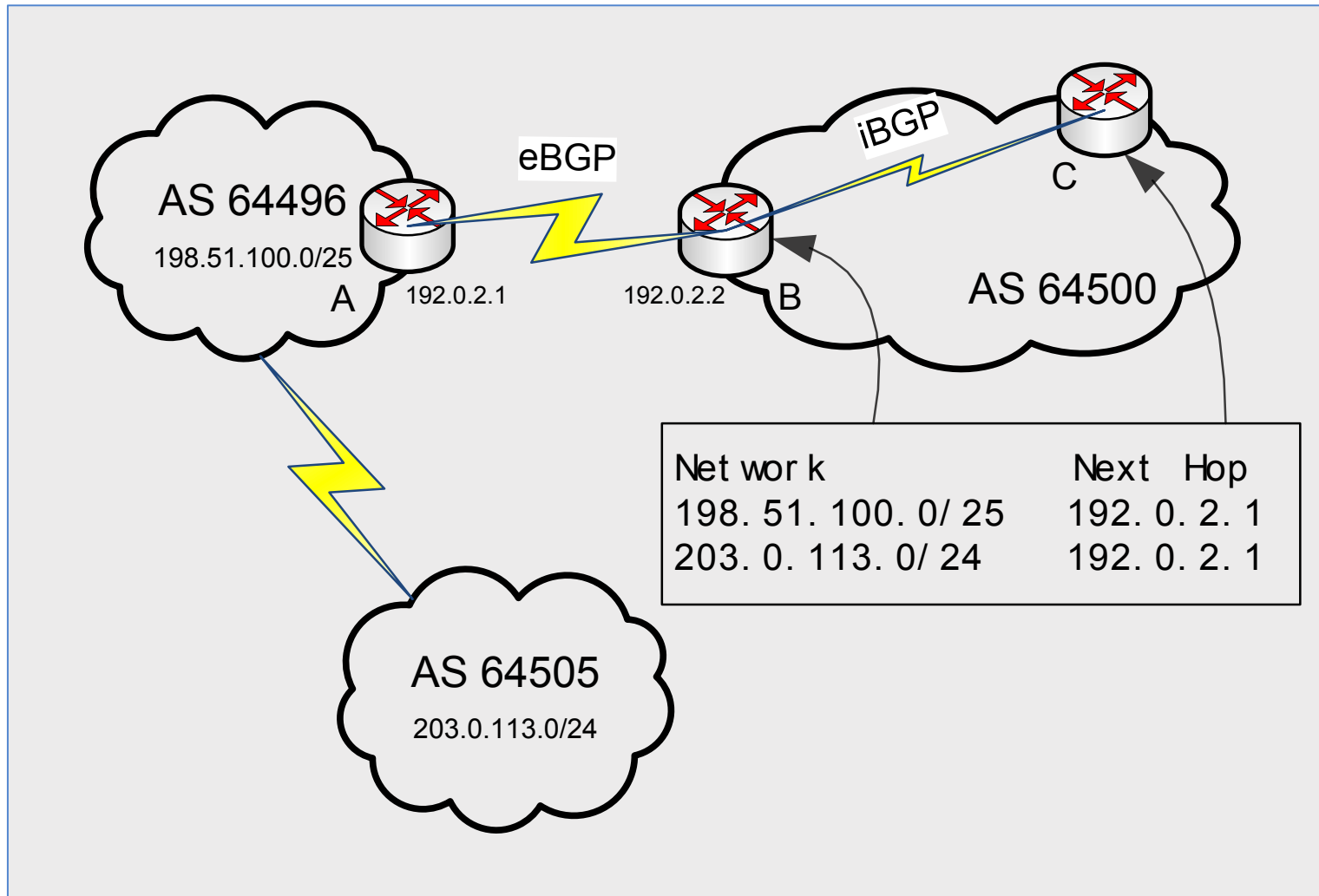
	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1			0	64496 65511 i
*	192.0.2.128/26	198.51.100.1			0	64496 64506 i
*>		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1			0	64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1			0	64496 ?

Atributos de rutas: next-hop

- En medios multiacceso (Ethernet), el NH es la IP de la interfaz del router que originó la ruta
 - Ejemplo: IXP



Atributos de rutas: next-hop



Atributos de rutas: AS-Path

- Secuencia de ASN que se deben atravesar para llegar al AS destino.
- Clave en el algoritmo de selección de rutas.
- Un AS Path en blanco, significa que la ruta fue originada en el AS local.

BGP table version is 134358, local router ID is 198.51.100.1

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

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1			0	64496 65511 i
*	192.0.2.128/26	198.51.100.1			0	64496 64506 i
*>		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1			0	64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1			0	64496 ?

Atributos de rutas: MED

- MED es utilizado para anunciar a los vecinos eBGP preferencia sobre el tráfico entrante a nuestro AS
 - Indica a otro AS cuál debería ser la puerta de entrada a nuestro AS.
- Los path con el valor MED más bajo son los más preferidos.
- Baja precedencia en el algoritmo de selección de rutas.

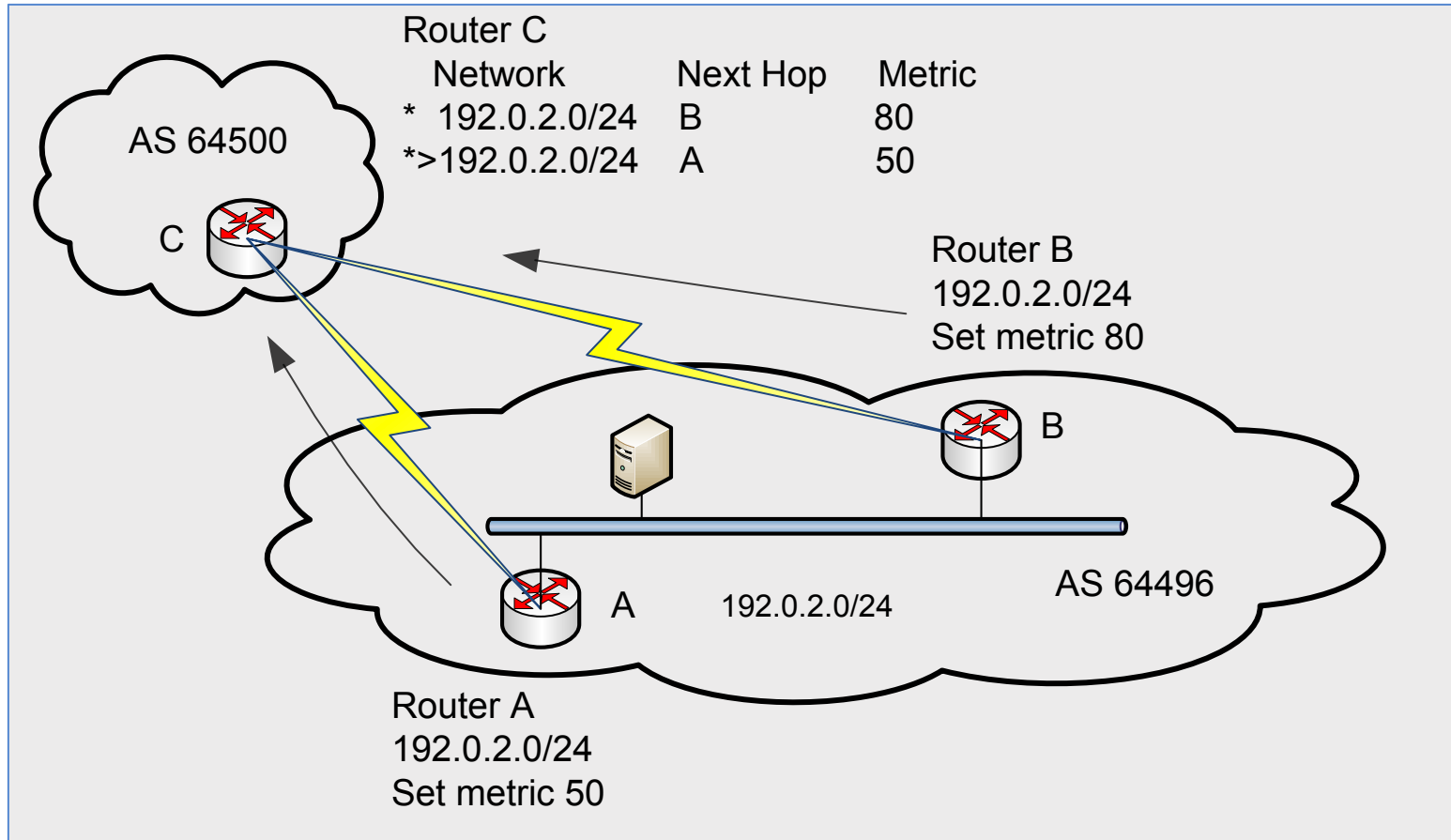
BGP table version is 134358, local router ID is 198.51.100.1

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

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1	95		0	64496 65511 i
*	192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>		198.51.100.22	95		0	64506 e
*	203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>		198.51.100.114		100	0	64501 i
*>	203.0.113.0/25	198.51.100.1	0	100	0	64496 64497 64498 i

Atributos de rutas: MED



- Los routers A y B anuncian el prefijo 192.0.2.0/24 con MED 50 y 80 a un eBGP vecino (AS 64500).
- El tráfico entrante al AS 64496 desde el AS 64500 elegirá el camino por el router A, pues tiene menor MED.

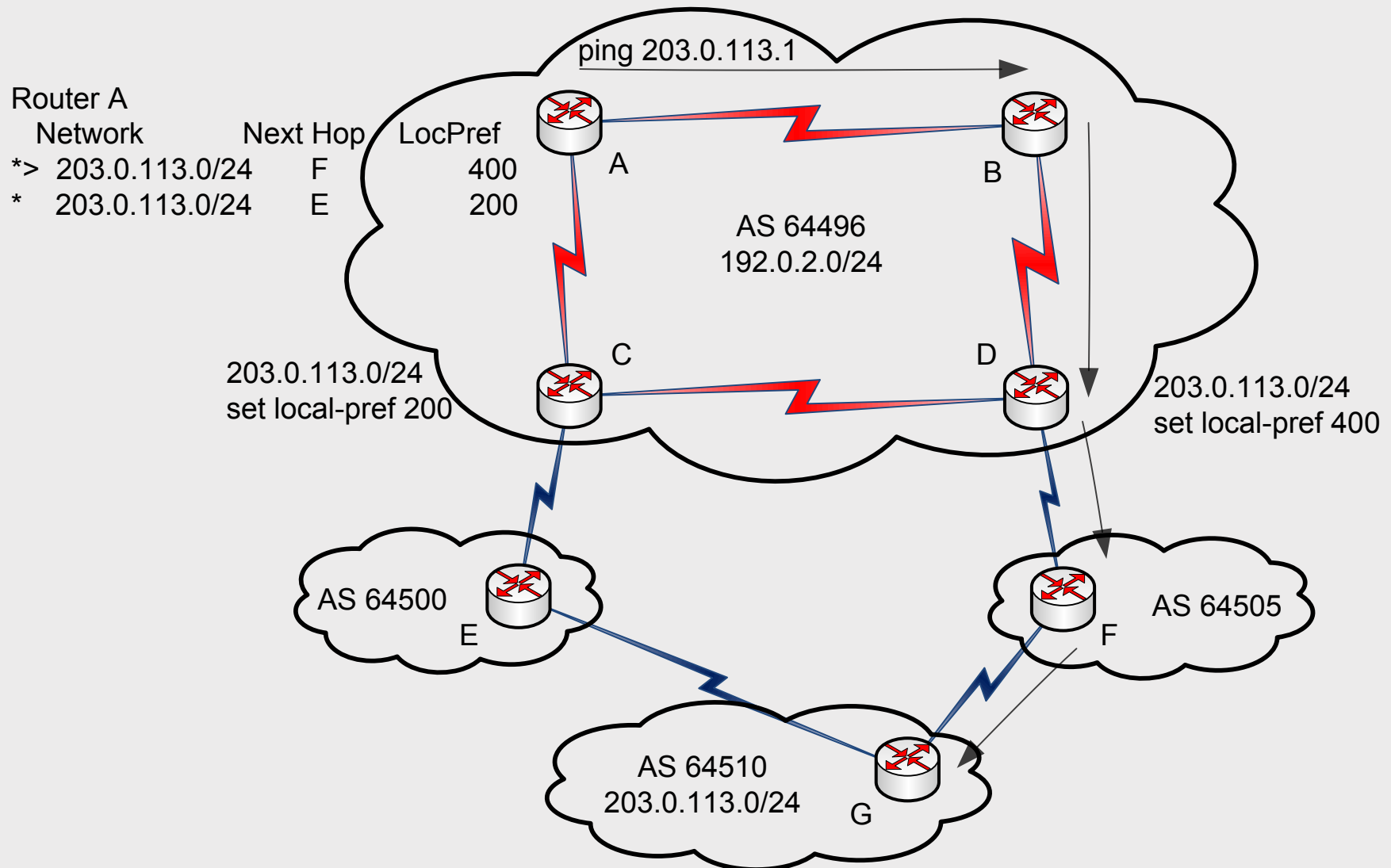
Atributos de rutas:

LOCAL_PREF

- Sirve para indicar un grado de preferencia respecto a otras rutas al mismo destino.
- Es un atributo **local al sistema autónomo** (se propaga por iBGP pero no por eBGP).
- Mayor local preference indica mejor preferencia:
 - local pref 600 es mejor que 100 (este es el valor por defecto)
- Se setea mediante **route-maps**.

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1	95		0	64496 65511 i
*	192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>		198.51.100.22	95		0	64506 e
*	203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>		198.51.100.114		200	0	64501 i
*>	203.0.113.0/25	198.51.100.1	0	200	0	64496 64497 64498 i

Atributos de rutas: LOCAL_PREF

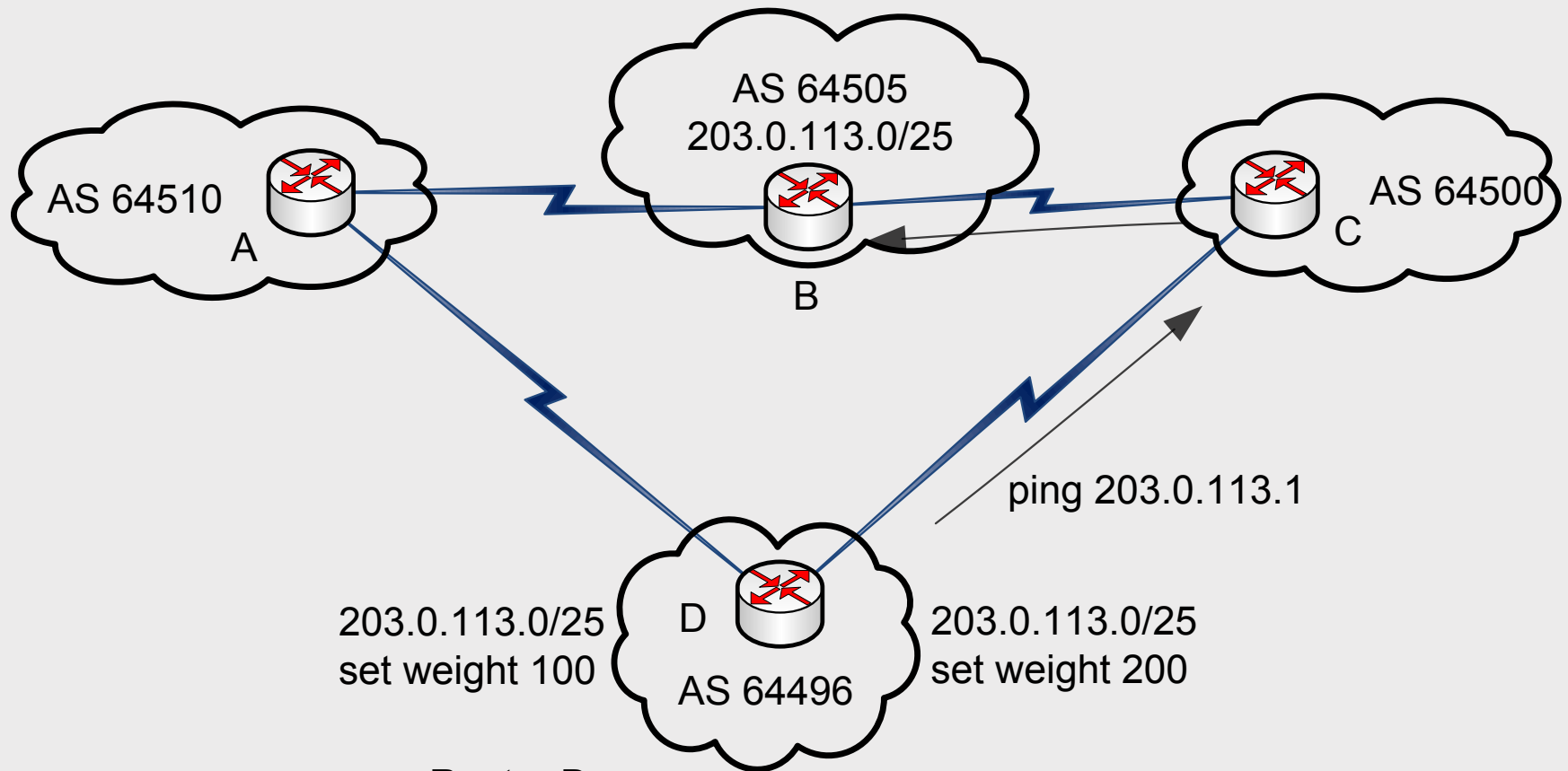


Atributos de rutas: Weight

- Es similar a local_pref, sólo que tiene **efecto local en el router** que se define.
- Tiene precedencia más alta que cualquier otro atributo.
- Sirve para discriminar rutas entre proveedores conectados al mismo router.
- Se setea mediante **route-maps**.
- Inicialmente propietario de CISCO, aunque ya otros fabricantes lo han incorporado.

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1	95		300	64496 65511 i
*	192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>		198.51.100.22	95		100	64506 e
*	203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>		198.51.100.114		100	0	64501 i
*>	203.0.113.0/25	198.51.100.1	0	100	0	64496 64497 64498 i

Atributos de rutas: Weight



Router D

	Network	Next Hop	Weight
*	203.0.113.0/25	A	100
*>	203.0.113.0/25	C	200

¿Qué vimos?

Atributos de rutas

Son parámetros preestablecidos que viajan junto a la información de la rutas.

Permiten poder ser manipulados por los administradores de redes para influir en las decisiones de BGP

Attribute Code	Type
ORIGIN	Well-known mandatory
NEXT-HOP	Well-known mandatory
AS_PATH	Well-known mandatory
MED ó Multi_EXIT_DISCriminator	Optional nontransitive
LOCAL_PREF ó Local Preference	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

BGP table version is 134358, local router ID is 198.51.100.1

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

Origin codes: i — IGP, e — EGP, ? — incomplete

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1	95		300	64496 65511 i
*	192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>		198.51.100.22	95		100	64506 e
*	203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>		198.51.100.114		100	0	64501 i
*>	203.0.113.0/25	198.51.100.1	0	100	0	64496 64497 64498 i

Decisiones administrativas

- Redistribución de la ruta default
- Insertando prefijos en BGP
- Filtrado de rutas

Redistribución del default

- **network 0.0.0.0**
 - Debe existir la ruta en la tabla de ruteo
 - Se enseña a todos los neighbors
 - Importante: poner filtros!!
- **default-information originate**
 - Anunciada a todos los neighbor
 - No importa si el router tiene una ruta por default
- **neighbor x.x.x.x default-originate**
 - Aplicado a un sólo neighbor
 - No importa si el router tiene una ruta por default

Insertando prefijos en BGP

Comando network

- Ejemplo:

```
router bgp 64496
  network 203.0.113.128 mask 255.255.255.128
ip route 203.0.113.128 255.255.255.128 serial0
```

- Una ruta coincidente debe existir en la tabla de ruteo antes de que la red sea anunciada.
- Y si esta ruta no está?

Insertando redes en BGP

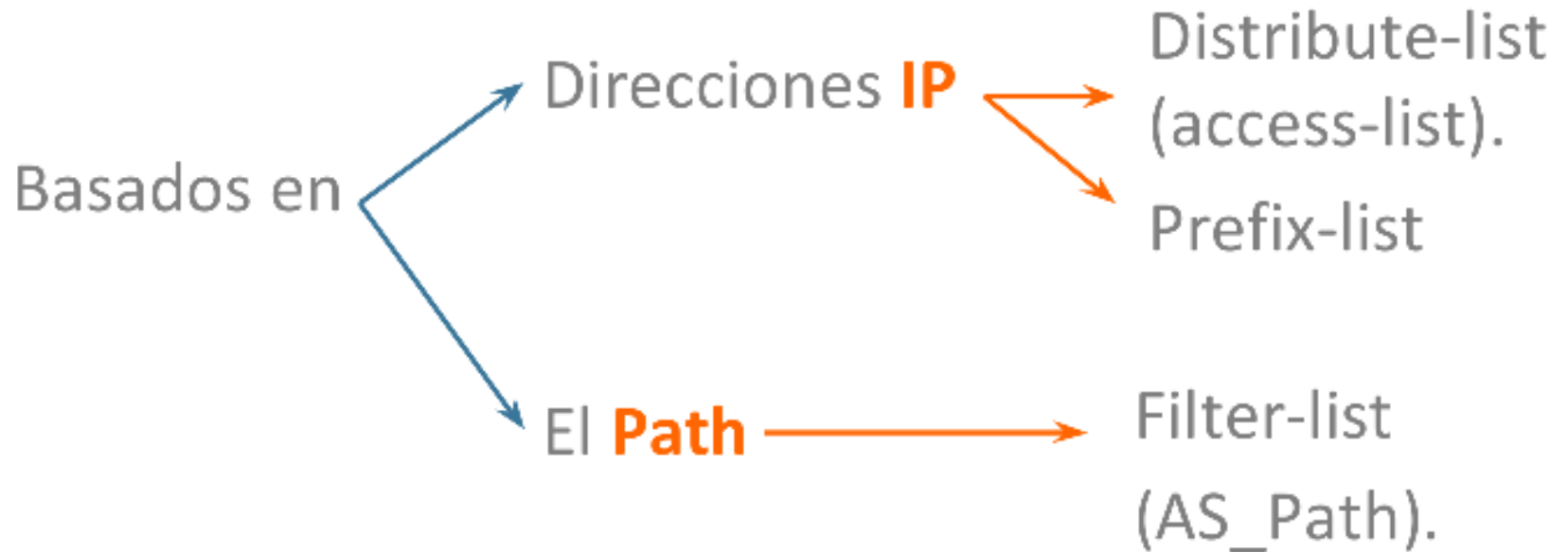
- Se inserta una ruta “pull up” en la tabla de ruteo, idéntica al prefijo que se quiere anunciar
- Se trata de una ruta estática a “null0”
 - sólo se usará si no hay una ruta más específica en la tabla de ruteo.
- Ejemplo

```
router bgp 64496
  network 203.0.113.128 mask 255.255.255.128
ip route 203.0.113.128 255.255.255.128 null0 250
```

Filtrado de rutas

- Proceso muy importante a fin de garantizar la estabilidad de nuestro AS y los AS vecinos.
- **Filtrado de entrada:** es aplicado a rutas aprendidas
 - Entonces la rutas no se incluyen en nuestra tabla de ruteo.
- **Filtrado de salida:** se aplica a rutas previamente a ser anunciadas a un vecino.
 - Entonces las rutas no se incluirán en las tablas de ruteo remotas.
- **Razones?**
 - Económicas – Ej: Transit ISP vs peering
 - Seguridad – Ej: sólo rutas asignadas a nuestros clientes
 - Técnicas – Ej: problemas de memoria

Filtrado de rutas



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 <regex>
```

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_
```

...

Expresiones Regulares

Caracter	Función
^	empieza con
\$	termina con
.	cualquier caracter
_	cualquier delimitador (espacio, comienzo, fin, coma)
[0-9]	rango del 0 al 9
[123]	1, 2 ó 3
()	asocia
	ó
*	ceros o más veces
?	ceros o una vez
+	una o más veces
\#	llama a la expresión ubicada en la posición # del regexp

Filtrado de rutas – Route-maps

- Los route-map son similares a las sentencias de un lenguaje de programación,
 - “if then”
- Cada instancia del route-map tiene un número de secuencia.
- Son ejecutados en orden desde la sentencia con menor número de secuencia hasta el más alto. Es posible editarlos o modificarlos utilizando este número de secuencia.
- Si en un route-map, una sentencia con un determinado criterio de coincidencia resulta verdadera, la ejecución del route-map se detiene.
- Se puede utilizar route-map para permitir o denegar según el criterio encontrado por la sentencia *match*.

Filtrado de rutas – Route-maps

- Si no existiera una sentencia *match* dentro de una instancia de un route-map, todas las rutas resultan con criterio verdadero. Las sentencias *set* son aplicadas a todas las rutas *.
- Si no existiera una lista de acceso para la sentencia *match* dentro de la instancia del route-map, todas las rutas resultan con criterio verdadero. Las sentencias *set* se aplican a todas las rutas.
- Tal como con las listas de acceso, una denegación implícita es incluida al final del route-map.
- Si múltiples sentencias *match* son utilizadas dentro de una instancia de un mapa de ruteo, todas las sentencias *match* deben resultar verdaderas para que de la instancia surja un resultado verdadero
* o paquetes

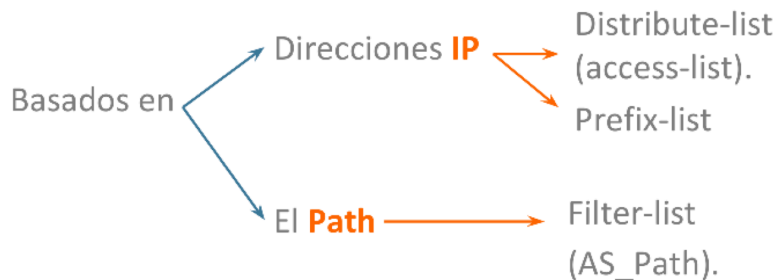
Route-map – Ejemplo con prefix-list

```
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_
```

¿Qué vimos?



Filtrado de rutas – Route-maps

- Los route-map son similares a las sentencias de un lenguaje de programación,
 - "if then"
- Cada instancia del route-map tiene un número de secuencia.
- Son ejecutados en orden desde la sentencia con menor número de secuencia hasta el más alto. Es posible editarlos o modificarlos utilizando este número de secuencia.
- Si en un route-map, una sentencia con un determinado criterio de coincidencia resulta verdadera, la ejecución del route-map se detiene.
- Se puede utilizar route-map para permitir o denegar según el criterio encontrado por la sentencia *match*.

```
...
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_
```

```
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
```

```
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_
```

Puesta a Punto

- Algoritmo de selección del mejor camino
- Mejores prácticas recomendadas

Selección del mejor camino

1. Si el next-hop es inaccesible, descartar la ruta.
2. Si el camino es interno (iBGP), synchronization está habilitado y la ruta no está en el IGP, descartarla.
3. Se prefiere el camino con mayor peso “weight” (propietario Cisco).
4. Luego, se prefiere la ruta de mayor “local preference”.

Selección del mejor camino

5. En caso del mismo local-pref, se prefiere una ruta que es originada por el router (comando network o redistribución).
6. Si la ruta no fue originada por el router y local-pref es igual, se prefiere la ruta con el path de sistemas autónomos más corto (shortest as-path).
7. Si todo es igual, se prefiere el menor código de “origen” (IGP<EGP<Incomplete).

Selección del mejor camino

8. A igualdad de “origen”, se prefiere el menor valor de MED. Esta comparación se realiza sólo si los neighbors de los que se aprendieron la ruta pertenecen todos a un mismo AS (a menos que se especifique “bgp always-compare-med”).
9. Se prefieren rutas aprendidas por eBGP que por iBGP.
10. Se prefiere la ruta cuyo next-hop tiene menor métrica en el IGP.
11. Si hasta aquí no hay decisión, se prefiere la ruta correspondiente al neighbor de menor router-id.

Sobre mejores prácticas recomendadas

BGP vs. IGPs

Protocolos de ruteo interno (IGP):

- Ejemplos OSPF, RIP, etc..
- Usados para transportar las direcciones de la **infraestructura**.
- No usados para transportar los prefijos de Internet o clientes.
- Diseñados de tal forma que se minimice el número de prefijos.

Sobre mejores prácticas recomendadas

BGP vs IGP's

- BGP: usar internamente (iBGP) y externamente (eBGP)
- iBGP: usado para transportar:
 - Los prefijos de Internet a través del Backbone
 - Los prefijos de los clientes
 - Utilizar /32 para las interfaces loopbacks de enrutamiento y levantar las sesiones de iBGP con ellas
 - Usar Peer Groups
 - Usar Passwords en la sesiones de iBGP (robos de prefijos)

Sobre mejores prácticas recomendadas

BGP vs IGP's

- IGP
 - El IGP debe contener a las redes de la DMZ o usar
neighbor x.x.x.x next-hop-self
- eBGP usados para
 - Intercambiar prefijos con otros AS
 - Clave de la implementación de políticas de ruteo

Sobre mejores prácticas recomendadas

BGP

- Nunca hacer
 - Redistribuir prefijos BGP en un IGP
 - Redistribuir rutas IGP dentro de BGP
 - NO Usar IGP para transportar los prefijos de los clientes u otras redes externas

Sobre mejores prácticas recomendadas

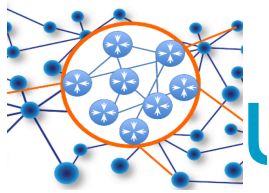
BGP

- Que anuncios no debería recibir
 - No recibir los prefijos definidos en el RFC1918
 - No aceptar mis propios prefijos
 - No aceptar el default (a menos que se requiera)
 - No aceptar prefijos mayores de /24

Sobre mejores prácticas recomendadas

- Que prefijos no debería recibir:

```
router bgp 64496
  network 192.0.2.0 mask 255.255.255.0
  neighbor 203.0.113.100 remote-as 64505
  neighbor 203.0.113.100 prefix-list in-filter in
  !
  ip prefix-list in-filter deny 0.0.0.0/0          ! Block default
  ip prefix-list in-filter deny 0.0.0.0/8 le 32
  ip prefix-list in-filter deny 10.0.0.0/8 le 32
  ip prefix-list in-filter deny 101.10.0.0/19 le 32 ! Block local prefix
  ip prefix-list in-filter deny 127.0.0.0/8 le 32
  ip prefix-list in-filter deny 169.254.0.0/16 le 32
  ip prefix-list in-filter deny 172.16.0.0/12 le 32
  ip prefix-list in-filter deny 192.0.2.0/24 le 32
  ip prefix-list in-filter deny 192.168.0.0/16 le 32
  ip prefix-list in-filter deny 224.0.0.0/3 le 32   ! Block multicast
  ip prefix-list in-filter deny 0.0.0.0/0 ge 25     ! Block prefixes >/24
  ip prefix-list in-filter permit 0.0.0.0/0 le 32
```

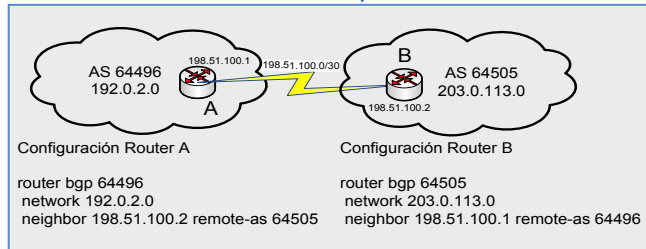
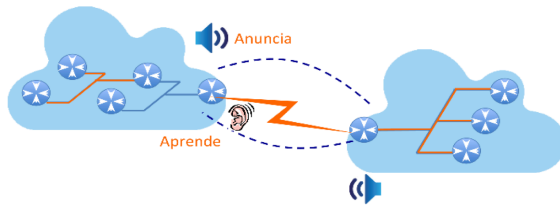


Finalmente... ¿qué vimos?

LOC-RIB

Tabla BGP	Tabla ISIS	Tabla de Ruteo
Destino Via	Destino Via	Destino Via
A <IPaddress>	A <IPaddress>	A <IPaddress>
	B 	B
	C 	C

APRENDER Y ANUNCIAR



Son parámetros preestablecidos que viajan junto a la información de la rutas.

Permiten poder ser manipulados por los administradores de redes para influir en las decisiones de BGP

Attribute Code	Type
ORIGIN	Well-known mandatory
NEXT-HOP	Well-known mandatory
AS_PATH	Well-known mandatory
MED o Multi_EXIT_Disriminator	Optional nontransitive
LOCAL_PREF o Local Preference	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference (MCI)	
12 — Advertiser (Baynet)	
13 — rcid_path (Baynet)	
255 — Reserved	—

BGP table version is 134358, local router ID is 198.51.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i — internal, S Stale
Origin codes: i — IGP, e — EGP, ? — incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.0/26	198.51.100.1	95		300	64496 65511 i
* 192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>	198.51.100.22	95		100	64506 e
* 203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>	198.51.100.114		100	0	64501 i
*> 203.0.113.0/25	198.51.100.1	0	100	0	64496 64497 64498 i

Selección del mejor camino

1. Si el next-hop es inaccesible, descartar la ruta.
2. Si el camino es interno (iBGP), synchronization está habilitado y la ruta no está en el IGP, descartarla.
3. Se prefiere el camino con mayor peso "weight" (propietario Cisco).
4. Luego, se prefiere la ruta de mayor "local preference".

Sobre mejores prácticas recomendadas

BGP vs. IGP

Protocolos de ruteo interno (IGP):

- Ejemplos OSPF, RIP, etc..
- Usados para transportar las direcciones de la infraestructura.
- No usados para transportar los prefijos de Internet o clientes.
- Diseñados de tal forma que se minimice el número de prefijos.



Filtrado de rutas – Route-maps

- Los route-map son similares a las sentencias de un lenguaje de programación.
- "if ... then ..."
- Cada instancia del route-map tiene un número de secuencia.
- Son ejecutados en orden desde la sentencia con menor número de secuencia hasta el más alto. Es posible editarlos o modificarlos utilizando este número de secuencia.
- Si en un route-map, una sentencia con un determinado criterio de coincidencia resulta verdadera, la ejecución del route-map se detiene.
- Se puede utilizar route-map para permitir o denegar según el criterio encontrado por la sentencia match.

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
```

```
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
!
```

```
...
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_
```

”Fundamentos de BGP e Introducción a RPKI”

- Modalidad:
 - Online con asistencia de tutor
 - Teórico / Práctico
- Duración: 6 semanas
- Primera edición 2018: 25 de mayo
- Informes e inscripción:

<http://campus.lacnic.net>

Muchas gracias!

mariela.c.rocha@gmail.com

