BGP Attributes and Policy Control

ISP/IXP Workshops

Agenda

- BGP Attributes
- BGP Path Selection
- Applying Policy

BGP Attributes

BGP Attributes

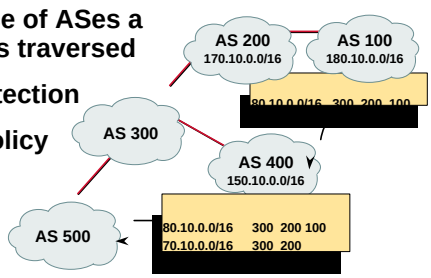
- What is an attribute?
- AS path
- Next hop
- Local preference
- Multi-Exit Discriminator (MED)
- Community
- Others include Origin and Aggregator

What Is an Attribute?

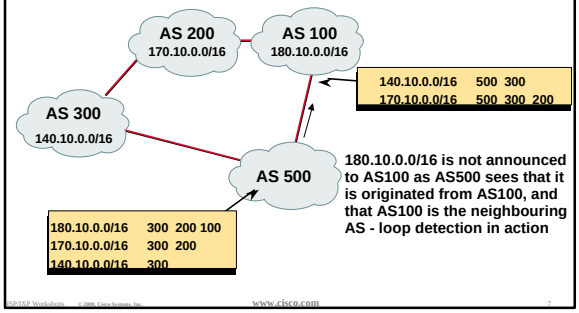
- Describes the characteristics of prefix
- Transitive or non-transitive
- Some are mandatory

AS-Path

- Sequence of ASes a route has traversed
- Loop detection
- Apply policy

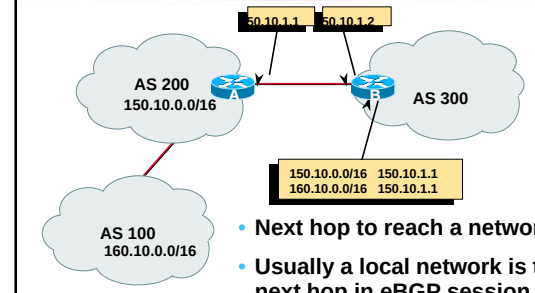


AS-Path loop detection



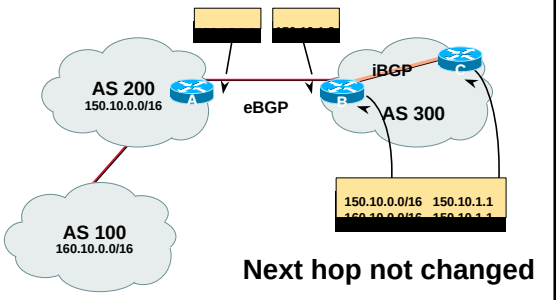
180.10.0.0/16 is not announced to AS100 as AS500 sees that it is originated from AS100, and that AS100 is the neighbouring AS - loop detection in action

Next Hop



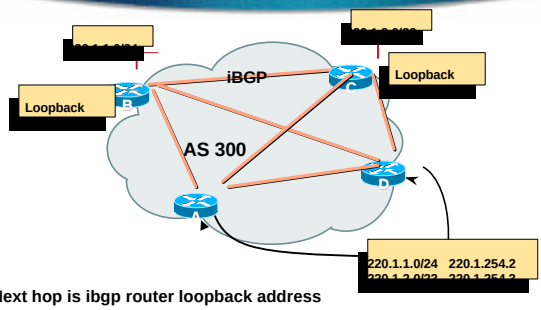
- Next hop to reach a network
- Usually a local network is the next hop in eBGP session

Next Hop



Next hop not changed

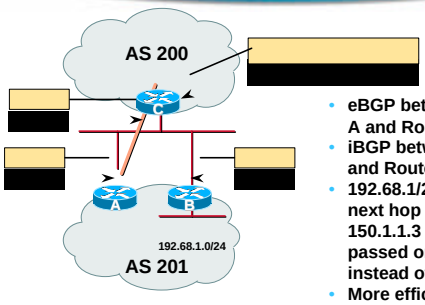
iBGP Next Hop



Next hop is ibgp router loopback address

Recursive route look-up

Third Party Next Hop



- eBGP between Router A and Router C
- iBGP between RouterA and RouterB
- 192.68.1.0/24 prefix has next hop address of 150.1.1.3 - this is passed on to RouterC instead of 150.1.1.2
- More efficient
- No extra config needed

Next Hop (summary)

- IGP should carry route to next hops
- Recursive route look-up
- Unlinks BGP from actual physical topology
- Allows IGP to make intelligent forwarding decision

Origin

- Conveys the origin of the prefix
- Influence best path selection
- Three values - IGP, EGP, incomplete

  IGP - generated from BGP network statement

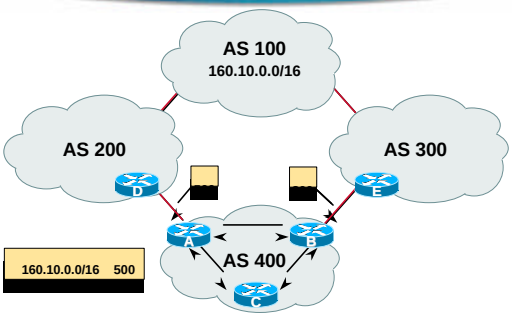
  EGP - generated from EGP

  incomplete - generated by "redistribute" action

Aggregator

- Useful for debugging purposes
- Conveys the IP address of the router/BGP speaker generating the aggregate route
- Doesn't influence path selection

Local Preference



Local Preference

- Local to an AS - non-transitive

  local preference set to 100 when heard from neighbouring AS

- Used to influence BGP path selection

  determines best path for outbound traffic

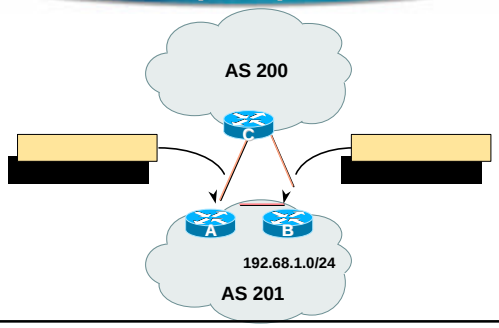
- Path with highest local preference wins

Local Preference

- Configuration of Router B:

```
router bgp 400
 neighbor 220.5.1.1 remote-as 300
 neighbor 220.5.1.1 route-map local-pref in
!
route-map local-pref permit 10
 match ip address prefix-list MATCH
 set local-preference 800
!
ip prefix-list MATCH permit 160.10.0.0/16
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```


Multi-Exit Discriminator (MED)



Multi-Exit Discriminator

- Inter-AS - non-transitive
  - metric reset to 0 on announcement to next AS
- Used to convey the relative preference of entry points
  - determines best path for inbound traffic
- Comparable if paths are from same AS
- IGP metric can be conveyed as MED

MED & IGP Metric

- set metric-type internal
  - enable BGP to advertise a MED which corresponds to the IGP metric values
  - changes are monitored (and re-advertised if needed) every 600s
  - bgp dynamic-med-interval <secs>

Multi-Exit Discriminator

- Configuration of Router B:

```
router bgp 400
 neighbor 220.5.1.1 remote-as 200
 neighbor 220.5.1.1 route-map set-med out
!
route-map set-med permit 10
 match ip address prefix-list MATCH
 set metric 1000
!
ip prefix-list MATCH permit 192.68.1.0/24
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```


Weight

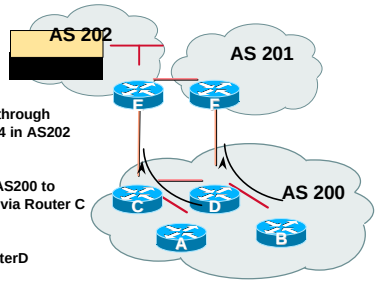
- Not really an attribute - local to router
- Highest weight wins
- Applied to all routes from a neighbour

```
neighbor 220.5.7.1 weight 100
```

- Weight assigned to routes based on filter

```
neighbor 220.5.7.3 filter-list 3 weight 50
```


Weight



Traffic from or passing through Router D to 192.68.1.0/24 in AS202 should go via AS201

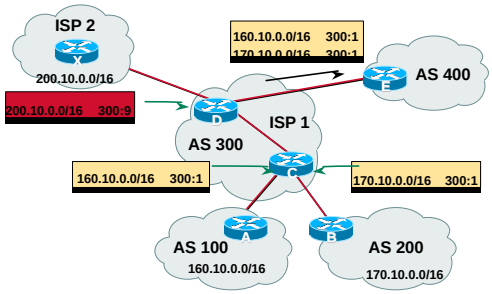
Traffic from the rest of AS200 to 192.68.1.0/24 should go via Router C (normal selection rules)

Use high weight on RouterD

Community

- BGP attribute
- Used to group destinations
- Represented as two 16bit integers
- Each destination could be member of multiple communities
- Community attribute carried across AS's
- Useful in applying policies

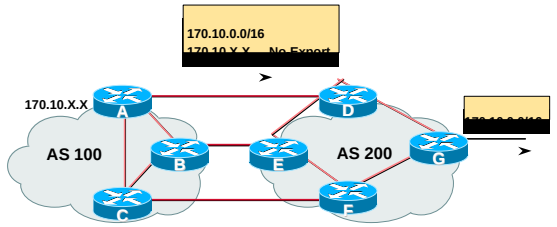
Community



Well-Known Communities

- internet = all routes are members of this community
- no-export = do not advertise to eBGP peers
- no-advertise = do not advertise to any peer
- local-AS = do not advertise outside local AS (used with confederations)

No-Export Community



No-Export Community

- AS100 announces aggregate and subprefixes
  - aim is to improve loadsharing between AS100 and AS200 by leaking subprefixes
- Subprefixes marked with no-export community
- Router G in AS200 strips out all prefixes with no-export community set

BGP Path Selection Algorithm

Why is this the best route?

BGP Path Selection Algorithm

- Do not consider path if no route to next hop
- Do not consider iBGP path if not synchronised
- Highest weight (local to router)
- Highest local preference (global within AS)
- Prefer locally originated route
- Shortest AS path

BGP Path Selection Algorithm (continued)

- Lowest origin code
  - IGP < EGP < incomplete
- Lowest Multi-Exit Discriminator (MED)
  - If bgp deterministic-med, order the paths before comparing
  - If bgp always-compare-med, then compare for all paths
  - otherwise MED only considered if paths are from the same AS (default)

BGP Path Selection Algorithm (continued)

- Prefer eBGP path over iBGP path
- Path with lowest IGP metric to next-hop
- For eBGP paths
  - if multipath enabled, install N parallel paths in routing table
  - if router-ID is the same, go to next step
  - if router-ID not the same, select “oldest”

BGP Path Selection Algorithm (continued)

- Lowest router-id (originator-id for reflected routes)
- Shortest Cluster-List
  - Client must be aware of Route Reflector attributes!
- Lowest neighbor IP address

Applying Policy with BGP

- Policy-based on AS path, community or the prefix
- Rejecting/accepting selected routes
- Set attributes to influence path selection
- Tools:
  - Distribute-list (access-list) or prefix-list
  - Filter-list (as-path access-list)
  - Route-maps and communities

Policy Control - Distribute List

- Per neighbour access-list
- Inbound or Outbound
- Based upon network numbers (e.g. through the use of access-lists)

Policy Control - Distribute List

- Example Configuration

```
router bgp 200
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 210
 neighbor 220.200.1.1 distribute-list 5 in
 neighbor 220.200.1.1 distribute-list 6 out
!
access-list 5 deny 218.10.0.0 0.0.255.255
access-list 5 permit any
access-list 6 permit 215.7.0.0 0.0.255.255
access-list 6 deny any
```


Policy Control - Prefix List

- Per neighbour prefix filter

  incremental configuration
- High performance access-list
- Inbound or Outbound
- Based upon network numbers (using familiar IPv4 address/mask format)

Prefix-list Command

[no] ip prefix-list <*list-name*> [seq <*seq-value*>] deny | permit <*network*>/<*len*> [ge <*ge-value*>] [le <*le-value*>]

<*network*>/<*len*>: The prefix and its length

ge <*ge-value*>: "greater than or equal to"

le <*le-value*>: "less than or equal to"

Both "ge" and "le" are optional. Used to specify the range of the prefix length to be matched for prefixes that are more specific than <*network*>/<*len*>

Prefix Lists - Examples

- Deny default route

```
ip prefix-list EG deny 0.0.0.0/0
```

- Permit the prefix 35.0.0.0/8

```
ip prefix-list EG permit 35.0.0.0/8
```

- Deny the prefix 172.16.0.0/12

```
ip prefix-list EG deny 172.16.0.0/12
```

- In 192/8 allow up to /24

```
ip prefix-list EG permit 192.0.0.0/8 le 24
```

This allows all prefix sizes in the 192.0.0.0/8 address block, apart from /25, /26, /27, /28, /29, /30, /31 and /32.

Prefix Lists - Examples

- In 192/8 deny /25 and above

```
ip prefix-list EG deny 192.0.0.0/8 ge 25
```

This denies all prefix sizes /25, /26, /27, /28, /29, /30, /31 and /32 in the address block 192.0.0.0/8.

It has the same effect as the previous example

- In 193/8 permit prefixes between /12 and /20

```
ip prefix-list EG permit 193.0.0.0/8 ge 12 le 20
```

This denies all prefix sizes /8, /9, /10, /11, /21, /22, … and higher in the address block 193.0.0.0/8.

- Permit all prefixes

```
ip prefix-list EG permit 0.0.0.0/0 le 32
```


Policy Control - Prefix List

- Example Configuration

```
router bgp 200
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 210
 neighbor 220.200.1.1 prefix-list PEER-IN in
 neighbor 220.200.1.1 prefix-list PEER-OUT out
!
ip prefix-list PEER-IN deny 218.10.0.0/16
ip prefix-list PEER-IN permit 0.0.0.0/0 le 32
ip prefix-list PEER-OUT permit 215.7.0.0/16
ip prefix-list PEER-OUT deny 0.0.0.0/0 le 32
```

Policy Control - Filter List

- Filter routes based on AS path
- Inbound or Outbound
- Example Configuration:

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 filter-list 5 out
 neighbor 220.200.1.1 filter-list 6 in
!
ip as-path access-list 5 permit ^200$
ip as-path access-list 6 permit ^150$
```


Policy Control - Regular Expressions

- Like Unix regular expressions

.	Match one character
*	Match any number of preceding expression
+	Match at least one of preceding expression
^	Beginning of line
$	End of line
_	Beginning, end, white-space, brace
|	Or
()	brackets to contain expression

Policy Control - Regular Expressions

- Simple Examples

.*	Match anything
.+	Match at least one character
^$	Match routes local to this AS
_1800$	Originated by 1800
^1800_	Received from 1800
_1800_	Via 1800
_790_1800_	Passing through 1800 then 790
_(1800_)+	Match at least one of 1800 in sequence
\(65350\)	Via 65350 (confederation AS)

Policy Control - Regular Expressions

- Not so simple Examples

^[0-9]+$	Match AS_PATH length of one
^[0-9]+_[0-9]+$	Match AS_PATH length of two
^[0-9]*_[0-9]+$	Match AS_PATH length of one or two
^[0-9]*_[0-9]*$	Match AS_PATH length of one or two
^[0-9]+_[0-9]+_[0-9]+$	Match AS_PATH length of three
_(701|1800)_	Match anything which has gone through AS701 or AS1800
_1849(_.+_)12163$	Match anything of origin AS12163 and passed through AS1849

Policy Control - Route Maps

- Example Configuration - route map and prefix-lists

```
ip prefix-list HIGH-PREF permit 10.0.0.0/8
ip prefix-list HIGH-PREF deny 0.0.0.0/0 le 32
ip prefix-list LOW-PREF permit 20.0.0.0/8
ip prefix-list LOW-PREF deny 0.0.0.0/0 le 32
!
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
!
router bgp 100
 neighbor 1.1.1.1 route-map infilter in
```


Policy Control - Route Maps

- Example Configuration - route map and filter lists

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-as-path in
!
route-map filter-on-as-path permit 10
 match as-path 1
 set local-preference 80
!
route-map filter-on-as-path permit 20
 match as-path 2
 set local-preference 200
!
ip as-path access-list 1 permit _150$
ip as-path access-list 2 permit _210_
```

Policy Control - Route Maps

- Example configuration of AS-PATH prepend

```
router bgp 300
 network 215.7.0.0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
 set as-path prepend 300 300
```

- Standard practice implements two occurrences of the ASN when prepending

Policy Control - Route Maps

- Route Map MATCH Articles

as-path	ip next-hop
clns address	ip route-source
clns next-hop	length
clns route-source	metric
community	nlri
interface	route-type
ip address	tag

Policy Control - Route Maps

- Route map SET Articles

as-path	dampening
automatic-tag	default interface
clns	interface
comm-list	ip default next-hop
community	ip next-hop

Policy Control - Route Maps

- Route map SET Articles

ip precedence	next-hop
ip qos-group	nlri multicast
ip tos	nlri unicast
level	origin
local preference	tag
metric	traffic-index
metric-type	weight

Policy Control - Matching Communities

- Example Configuration

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-community in
!
route-map filter-on-community permit 10
 match community 1
 set local-preference 50
!
route-map filter-on-community permit 20
 match community 2 exact-match
 set local-preference 200
!
ip community-list 1 permit 150:3 200:5
ip community-list 2 permit 88:6
```


Policy Control - Setting Communities

- Example Configuration

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 200
 neighbor 220.200.1.1 send-community
 neighbor 220.200.1.1 route-map set-community out
!
route-map set-community permit 10
 match ip address prefix-list NO-ANNOUNCE
 set community no-export
!
route-map set-community permit 20
 match ip address prefix-list EVERYTHING
!
ip prefix-list NO-ANNOUNCE permit 172.168.0.0/16 ge 17
ip prefix-list EVERYTHING permit 0.0.0.0/0 le 32
```

Aggregation Policies

- **Suppress Map**

  Used to suppress selected more-specific prefixes (e.g. defined through a route-map) in the absence of the summary-only keyword.

- **Unsuppress Map**

  Used to unsuppress selected more-specific prefixes per BGP peering when the summary-only keyword is in use.

Aggregation Policies Suppress Map

- **Example**

```
router bgp 100
 network 220.10.10.0
 network 220.10.11.0
 network 220.10.12.0
 network 220.10.33.0
 network 220.10.34.0
 aggregate-address 220.10.0.0 255.255.0.0 suppress-map block-net
 neighbor 222.5.7.2 remote-as 200
!
route-map block-net permit 10
 match ip address prefix-list SUPPRESS
!
ip prefix-list SUPPRESS permit 220.10.8.0/21 le 32
ip prefix-list SUPPRESS deny 0.0.0.0/0 le 32
!
```


Aggregation Policies Suppress Map

- **show ip bgp on the local router**

```
router1#sh ip bgp
BGP table version is 11, local router ID is 222.5.7.1
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
*> 220.10.0.0/16  0.0.0.0                  32768 i
s> 220.10.10.0    0.0.0.0               0          32768 i
s> 220.10.11.0    0.0.0.0               0          32768 i
s> 220.10.12.0    0.0.0.0               0          32768 i
*> 220.10.33.0    0.0.0.0               0          32768 i
*> 220.10.34.0    0.0.0.0               0          32768 i
```


Aggregation Policies Suppress Map

- **show ip bgp on the remote router**

```
router2#sh ip bgp
BGP table version is 90, local router ID is 222.5.7.2
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
*> 220.10.0.0/16    222.5.7.1                    0  100 i
*> 220.10.33.0      222.5.7.1         0          0  100 i
*> 220.10.34.0      222.5.7.1         0          0  100 i
```


Aggregation Policies Unsuppress Map

- **Example**

```
router bgp 100
 network 220.10.10.0
 network 220.10.11.0
 network 220.10.12.0
 network 220.10.33.0
 network 220.10.34.0
 aggregate-address 220.10.0.0 255.255.0.0 summary-only
 neighbor 222.5.7.2 remote-as 200
 neighbor 222.5.7.2 unsuppress-map leak-net
!
route-map leak-net permit 10
 match ip address prefix-list LEAK
!
ip prefix-list LEAK permit 220.10.8.0/21 le 32
ip prefix-list LEAK deny 0.0.0.0/0 le 32
!
```


Aggregation Policies Unsuppress Map

- **show ip bgp on the local router**

```
router1#sh ip bgp
BGP table version is 11, local router ID is 222.5.7.1
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
*> 220.10.0.0/16  0.0.0.0                  32768 i
s> 220.10.10.0    0.0.0.0               0          32768 i
s> 220.10.11.0    0.0.0.0               0          32768 i
s> 220.10.12.0    0.0.0.0               0          32768 i
s> 220.10.33.0    0.0.0.0               0          32768 i
s> 220.10.34.0    0.0.0.0               0          32768 i
```

Aggregation Policies Unsuppress Map

- **show ip bgp** on the remote router

```
router2#sh ip bgp
BGP table version is 90, local router ID is 222.5.7.2
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
*> 220.10.0.0/16    222.5.7.1                        0  100 i
*> 220.10.10.0      222.5.7.1         0              0  100 i
*> 220.10.11.0      222.5.7.1         0              0  100 i
*> 220.10.12.0      222.5.7.1         0              0  100 i
```


Aggregation Policies Aggregate Address

- **Summary-only used**
  - **all subprefixes suppressed**
  - **unsuppress-map to selectively leak subprefixes**
  - **bgp per neighbour configuration**
- **Absence of summary-only**
  - **no subprefixes suppressed**
  - **suppress-map to selectively suppress subprefixes**
  - **bgp global configuration**