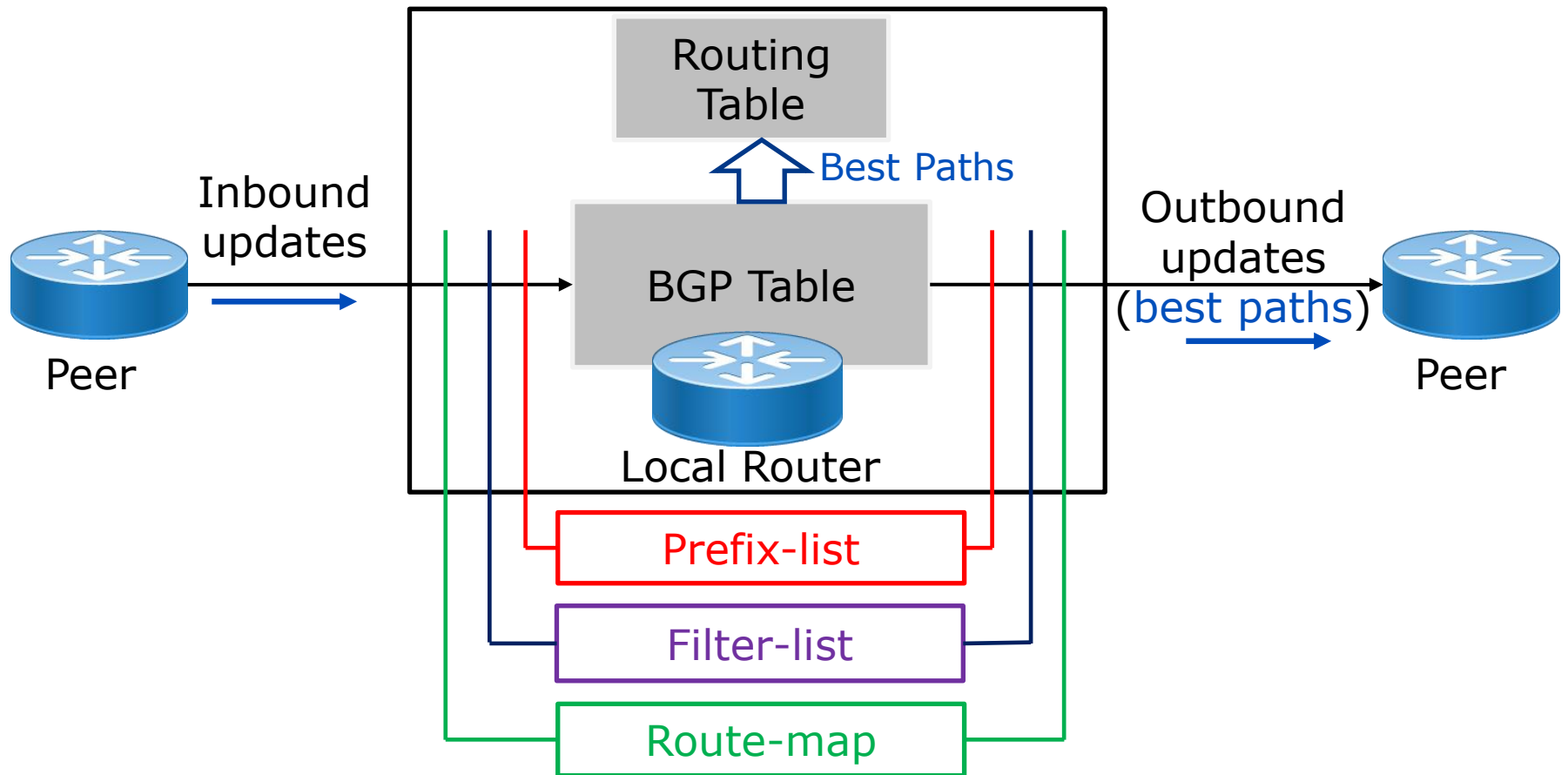


BGP Policy Control

Influence Path Selection – Policy Control



Policy Tools

- Prefix-list
 - To filter routes/prefixes
 - More granularity than as-path filters
- Filter-list
 - To filter based on AS-path
 - To apply AS-path ACLs
- Route-map
 - modify attributes based on condition matches

Path control - Attributes

- Inbound Traffic:
 - AS-Path, MED, Community
- Outbound Traffic:
 - Local Preference

Prefix List

```
ip prefix-list name/num [seq#] permit | deny  
prefix/length [ge value][le value]
```

- Ex 1:

```
ip prefix-list TEST permit 0.0.0.0/0 ge 8 le 24
```

- Allows any prefix with prefix length between 8 and 24
- Implicit **DENY** at the end!

- Ex 2:

```
ipv6 prefix-list TEST-v6 permit 2001:6400::/32 le 48
```

- Permit the prefix 2400:6400::/32 up to /48
- Implicit **DENY** at the end!

Prefix List

- Ex 3:

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

- Deny default route

- Ex 4:

```
ipv6 prefix-list TEST-v6 deny ::/0
```

- Deny IPv6 default routes

Prefix List

```
router bgp 17821
  network 100.100.0.0 mask 255.255.224.0
  neighbor 20.20.20.1 remote-as 20
  neighbor 20.20.20.1 prefix-list MY-PREFIX out
  neighbor 20.20.20.1 prefix-list PEER-PREFIX in
!
ip prefix-list MY-PREFIX permit 100.100.0.0/19
ip prefix-list MY-PREFIX deny 0.0.0.0/0 le 32
!
ip prefix-list PEER-PREFIX permit 200.200.0.0/16
ip prefix-list PEER-PREFIX deny 0.0.0.0/0 le 32
```

AS-path ACL

```
ip as-path access-list num [permit|deny] regex
```

- AS-path access list use regular expressions
 - . Matches any one character
 - * Matches any sequence of pattern before *
 - + match at least one preceding expression
 - ^ beginning with
 - \$ ending with
 - _ matches start, end, space, comma, braces

AS-path ACL

- Example regular expressions:

<code>^\$</code>	locally originated routes
<code>_100\$</code>	originated by AS 100
<code>_100_200_</code>	passing through 100 and 200
<code>^(_100)+\$</code>	originated by 100, multiple occurrence

- Example 1:

```
ip as-path access-list 10 permit ^100$
```

- Allow any prefix originated and received from AS100
- Implicit **DENY** at the end

- Use **filter-list** to apply AS-PATH access-lists

AS-path ACL

```
router bgp 17821
 network 100.100.0.0 mask 255.255.224.0
 neighbor 30.30.30.1 remote-as 30
 neighbor 30.30.30.1 filter-list 30 out
 neighbor 30.30.30.1 filter-list 40 in
!
ip as-path access-list 30 permit ^$
ip as-path access-list 40 permit ^30$
```

Route Map

```
route-map name [permit | deny] [sequence]
```

- Default is permit
 - Implicit **DENY** at the end!

```
route-map TEST permit 10  
  match A B C  
  match D  
  set X  
  set Y
```

```
If {(A or B or C)  
and D} match  
Then {set X and Y}
```

```
route-map TEST permit 20  
  match E  
  set Z
```

```
Else  
If E matches  
Then set Z
```

```
route-map TEST permit 30
```

```
Else (for everything else)  
Do/set nothing
```

Match (conditions) & Set (actions)

Command	Description
<code>match community</code>	BGP community tag
<code>match as-path</code>	AS-path access list
<code>match ip address</code>	Access list or prefix-list

Command	Description
<code>set as-path <prepend></code>	Modify AS-path
<code>set community</code>	Apply BGP community tag
<code>set metric</code>	Modify MED
<code>set local-preference</code>	Modify local preference

Route Map

```
router bgp 17821
  neighbor 30.30.30.1 remote-as 30
  neighbor 30.30.30.1 route-map AS-OUT out
  neighbor 30.30.30.1 route-map LP-IN in
!
route-map AS-OUT permit 10
  set as-path prepend 17821 17821 17821
!
route-map LP-IN permit 10
  match as-path 1
  set local-preference 150
!
route-map LP-IN permit 20
!
ip as-path access-list 1 permit _30$
```

Route Map

- Setting and Matching communities:

```
router bgp 17821
  network 100.100.0.0 mask 255.255.224.0 route-map SET-AGG
  neighbor 20.20.20.1 remote-as 20
  neighbor 20.20.20.1 send-community
  neighbor 20.20.20.1 route-map TR-IN in
!
route-map SET-AGG permit 10
  set community 100:1000
!
route-map TR-IN permit 10
  match community 5
  set local-preference 150
!
route-map TR-IN permit 20
!
ip community-list 5 permit 20:3000
ip community-list 5 permit 20:4000
```

Applying Policy Filters

- Incoming/Outgoing updates are filtered through policies
 - BGP table does not contain routes rejected by policies
- Whenever there is a BGP policy change, we need to
 - Trigger an update to force in/outbound routes through the new filters (else only the ones already in BGP table)
 - either through a **Hard Reset** or a **Soft Reset** (Route Refresh)
- If the filter is applied to:
 - Outbound routes: need to resend its BGP table through the filter
 - Inbound routes: need its neighbors to resend their BGP tables

Hard Reset

- Hard reset of a BGP session
 - Tears down the TCP connection
 - Re-establish the TCP session
 - Resend the BGP table to neighbors affected by the reset
 - Relearn all routes from neighbors

```
clear ip bgp *  
clear ip bgp <peer-address>  
clear bgp ipv6 unicast *  
clear bgp ipv6 unicast <peer-address>
```

- Disrupts network connectivity
 - Same as a **router reboot!**

Route Refresh

- Route refresh (soft reset) does not tear down BGP session
 - Not disruptive!
- No manual configuration required
 - Negotiated automatically
 - Peering routers need to support route refresh

```
sh ip bgp neighbor <peer-address>  
sh ip bgp all neighbors  
sh bgp ipv6 unicast neihgbers
```

Neighbor capabilities:

Route refresh: advertised and received

Route Refresh

```
clear ip bgp <peer-address> [soft] in  
clear bgp ipv6 unicast <peer-address> [soft] in
```

- Tells neighbor to resend its BGP table

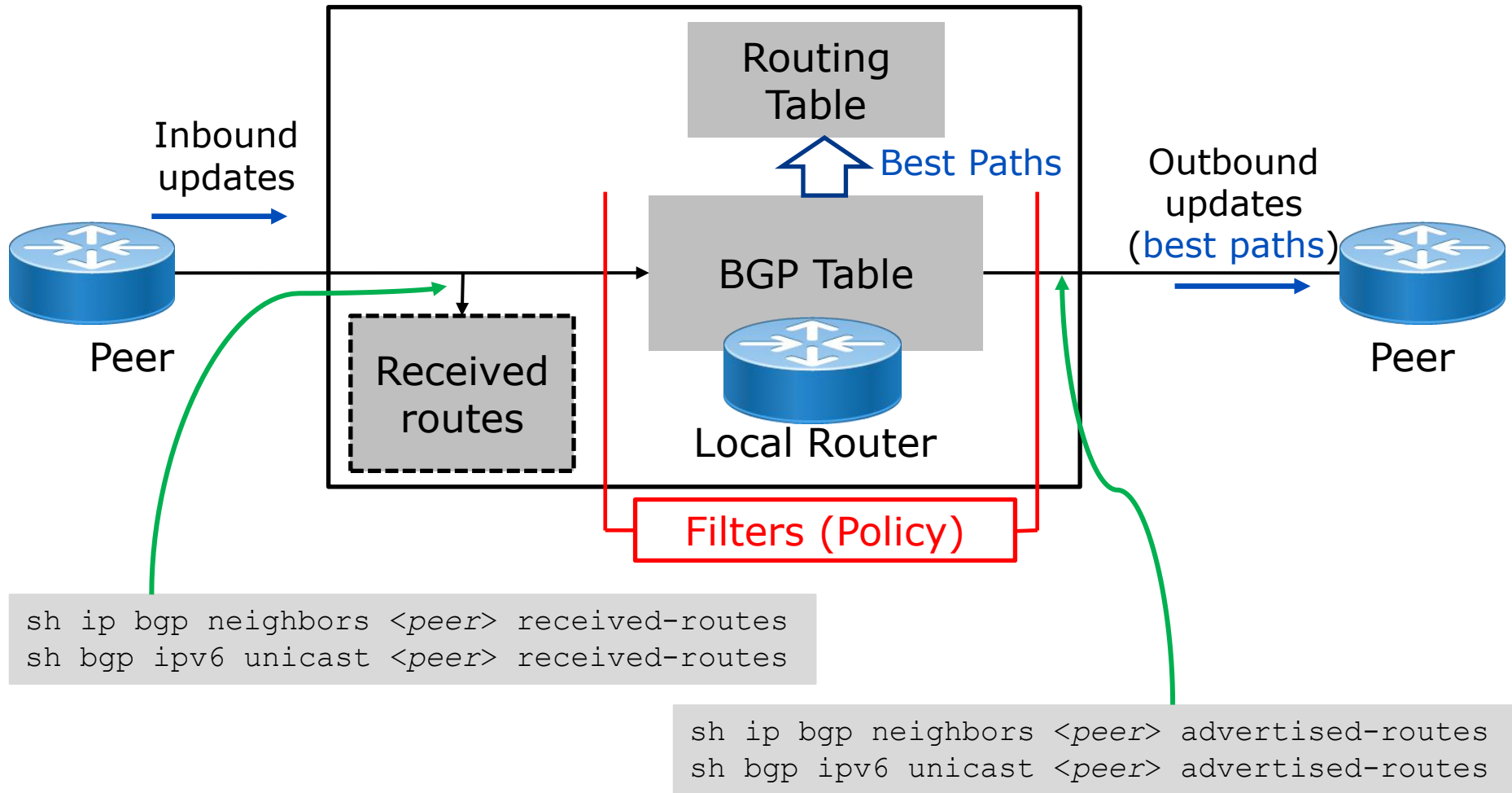
```
clear ip bgp <peer-address> [soft] out  
clear bgp ipv6 unicast <peer-address> [soft] out
```

- Resends full BGP table to its neighbor

Soft Reconfiguration

- All current routers should support route refresh
 - If the router (local or peer) does not have route refresh capability, use **soft-reconfiguration**
- With soft-reconfiguration, the router stores a copy of the received routes in addition to the BGP table (allowed by policy filters)
 - Thus, requires additional memory!

Soft Reconfiguration



Soft Reconfiguration

```
router bgp 17821
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map SET-LPREF in
neighbor 1.1.1.1 soft-reconfiguration inbound
```

– Whenever there is change in policy

```
clear ip bgp 1.1.1.1 soft [in|out]
```

- If "in", runs the stored received routes through the new filter
 - If "out", sends the BGP table through the filters
 - Does not tear down the BGP session!
- Route refresh capability cannot be used if soft-reconfig is used!

Acknowledgement:

- Philip Smith
- Cisco Systems



Questions

