Consider only routes with no AS loops and a valid next hop.

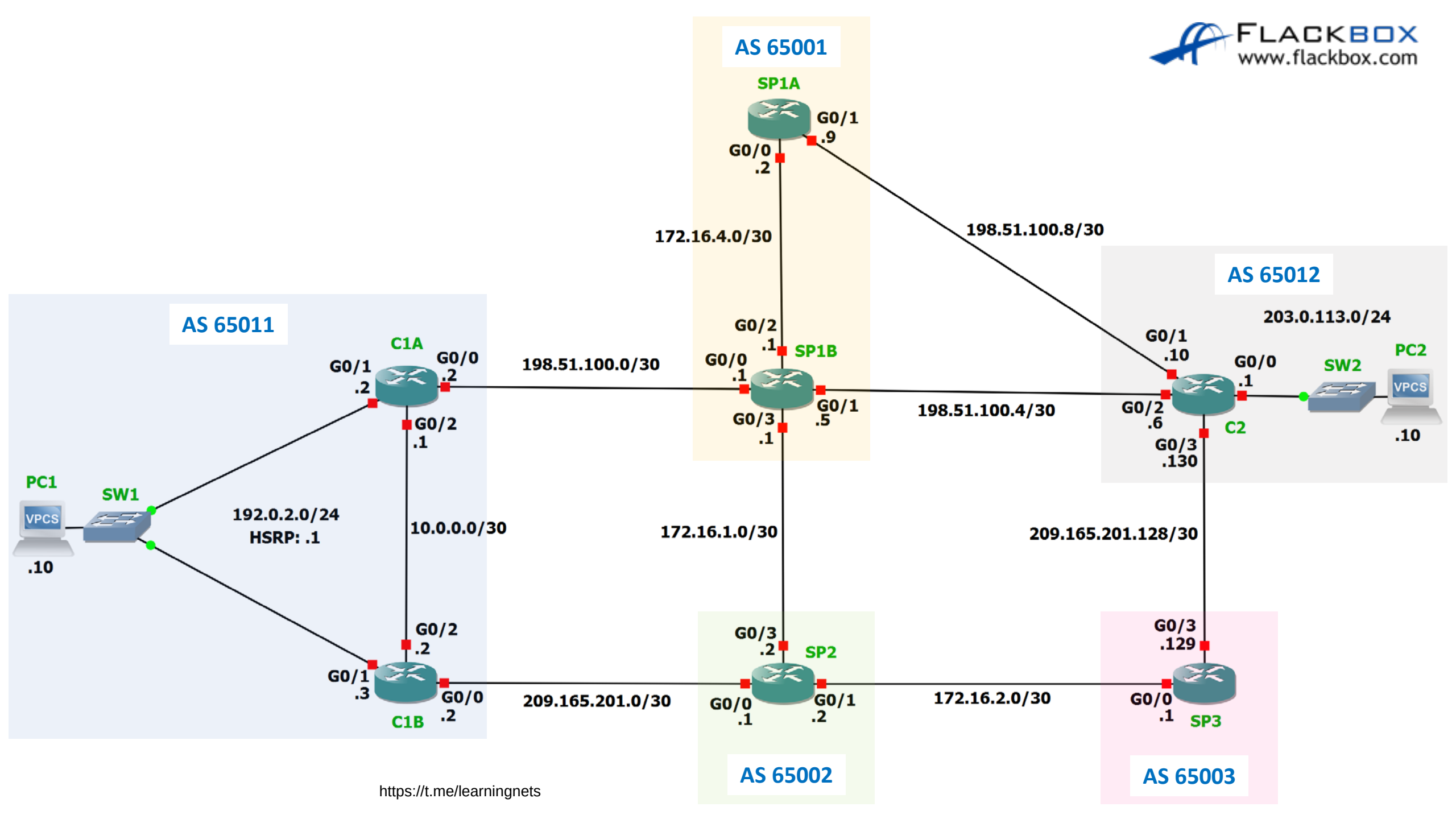
Use Longest Prefix Match.

Where multiple routes are available to identical network and prefix:

- Prefer highest weight (local to router).
- Prefer highest local preference (global within AS).
- Prefer route originated by the local router ('network' command or redistribution).
- Prefer shortest AS path.
- Prefer lowest origin code: IGP ('network') < EGP (legacy) < incomplete (redistributed).
- Prefer lowest MED (exchanged between autonomous systems).
- Prefer EBGP path over IBGP path.
- Prefer the path through the closest IGP neighbor.
- Prefer oldest route for EBGP paths.
- Prefer the path with the lowest neighbor BGP router ID.
- Prefer the path with the lowest neighbor IP address.

BGP Path Manipulation

- An administrator can influence the path **outbound** traffic from an AS takes by altering the path attributes of BGP packets received from neighbors:
  - Weight (local to an individual router)
  - Local Preference (advertised to iBGP neighbors)
- An administrator can influence the path **inbound** traffic to an AS takes by altering the path attributes of BGP packets sent to neighbors:
  - AS Path Prepending
  - MED Multi Exit Discriminator

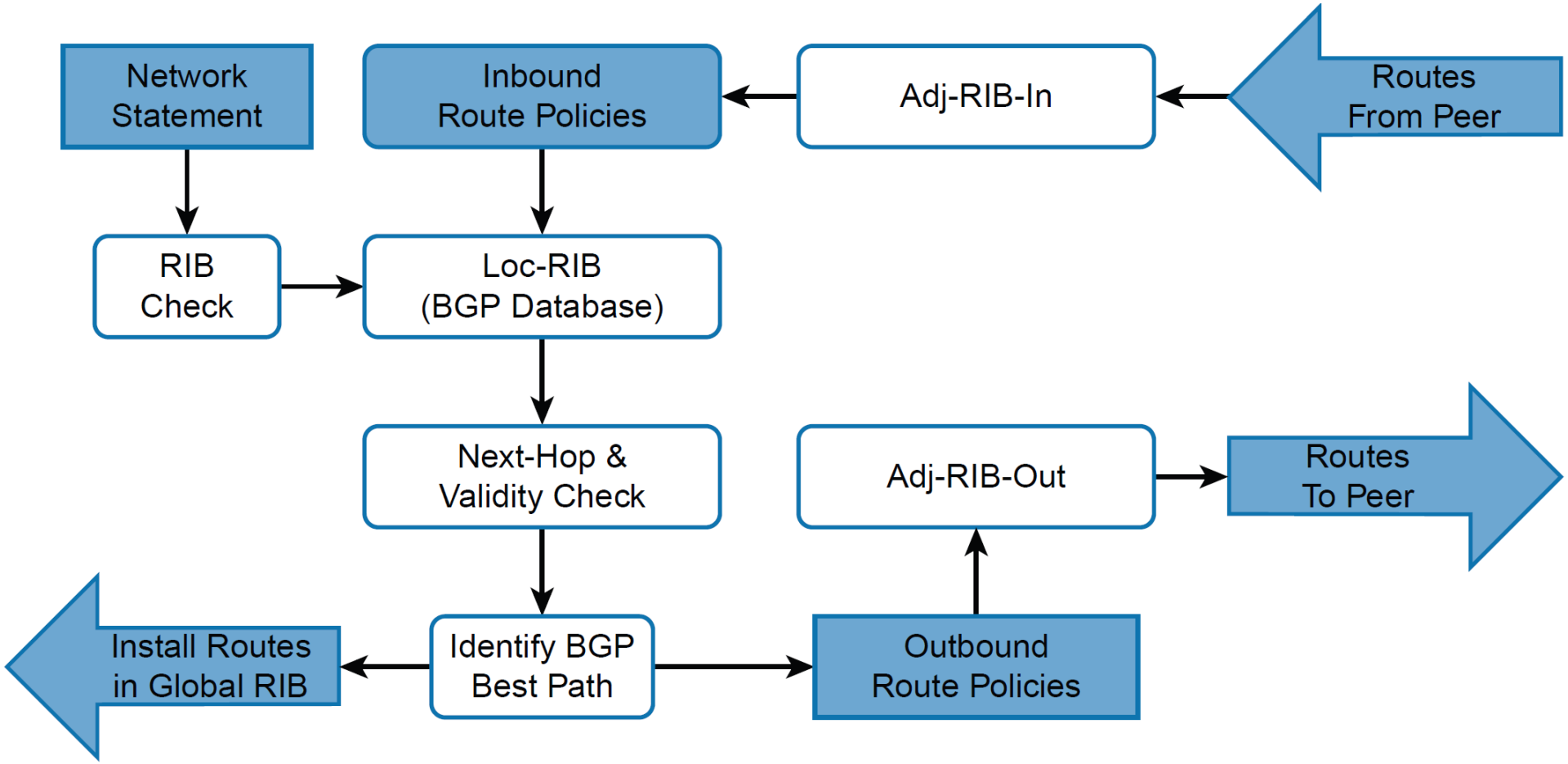


BGP Policy – Route Maps

- BGP Policy is typically configured with Route Maps

```
R1(config-route-map)#set ?
  as-path            Prepend string for a BGP AS-path attribute
  community          BGP community attribute
  extcommunity       BGP extended community attribute
  local-preference   BGP local preference path attribute
  metric             Metric value for destination routing protocol (MED)
  origin             BGP origin code
  tag                Tag value for destination routing protocol
  weight             BGP weight for routing table
```

BGP Operation



BGP RIB Routing Information Base

- The BGP standard mandates that the RIB in a BGP router consists of 3 distinct parts:
- **Adj-RIB-In**: Stores original routing information that has been learned from inbound Update messages, before policy is applied.
- **Loc-RIB**: The BGP table. Contains the local routing information after applying local policies and the path selection process to locally originated and received routes.
- **Adj-RIB-Out**: Stores the information that will be sent to peers in outbound Update messages.
- The BGP RIB is a conceptual model which does not require 3 separate copies of the routing information. The choice of implementation (for example, 3 copies of the information vs 1 copy with pointers) is not fixed.

BGP Policy Changes

- The BGP path selection decision is made when the route enters the BGP table (Loc-RIB)
- Incoming policies are applied to received routes before they enter the BGP table
- Incoming policy can change attributes on the received routes eg Weight
- Inbound policy must be applied to received routes in their original Adj-RIB-In form from the neighbor (not a potentially changed form in Loc-RIB)
- To reduce the amount of required memory, routes are flushed from Adj-RIB-In after policy has been applied

BGP Policy Changes (Cont.)

- BGP is designed for stability. BGP routers do not send update packets unless something has changed from their perspective
- By default, incoming policies have no effect until neighbors send update packets. Existing entries in the BGP table remain unchanged until then
- If you edit BGP policy to affect received routes, you need to take action to trigger updates

- New outgoing policies typically trigger update packets to be sent to neighbors automatically

clear ip bgp

```
R1#clear ip bgp 192.0.2.1
```

- ‘clear ip bgp’ tears down then rebuilds the BGP session
- Updates are sent in both directions – applying policy in both directions
- `clear ip bgp *` resets all BGP sessions
- This can be highly disruptive in a production environment

Soft Reconfiguration – clear ip bgp soft

- Soft Reconfiguration keeps the neighbor's originally received routing information in Adj-RIB-In. This negates the need to reset the session with the neighbor
- It can require a significant amount of memory
- Once Soft Reconfiguration is enabled, it starts to keep routes received from the neighbor in Adj-RIB-In. Routes received earlier have already been flushed.
- Reset the BGP session with the neighbor if you need to force it to resend all routes and have them stored in Adj-RIB-In

Soft Reconfiguration

```
R1(config)#router bgp 65001
R1(config-router)#neighbor 192.0.2.1 soft-reconfiguration
inbound
R1#clear ip bgp 192.0.2.1 ! Hard resets neighbor so it resends all routes

R1#clear ip bgp 192.0.2.1 soft in ! Applies inbound policy to routes
received from neighbor
```

Soft Reconfiguration

- To send updates to a neighbor without requiring a session reset:
- `R1#clear ip bgp 192.0.2.1 soft out`
- (This is not usually required because outbound policy changes automatically trigger Updates to be sent)
- (There is no '`neighbor 192.0.2.1 soft-reconfiguration` *`outbound`*' command because the outgoing routes and policy are local to the router and do not require a separate database table to be built, unlike '*inbound*' which stores a copy of the original external information from the neighbor before local policy was applied)

Soft Reconfiguration

- `show ip bgp neighbor received-routes` shows the contents of Adj-RIB-In and requires Soft Reconfiguration to be enabled
- (`show ip bgp neighbor advertised-routes` does not)
- It shows path attribute values received from a neighbor, before any incoming BGP policy is applied

```
R1(config-router)#neighbor 192.0.2.1 soft-reconfiguration inbound
R1#sh ip bgp neighbors 192.0.2.1 received-routes
BGP table version is 3, local router ID is 192.168.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  203.0.113.0     192.0.2.1                              0 65001 65012 i
```

Route Refresh

- Hard resets are disruptive, and soft reconfiguration can require a significant amount of memory
- Route Refresh avoids those issues by providing a mechanism where the router asks its neighbor to send route Updates again
- It uses the Route Refresh BGP packet
- Route Refresh must be supported on both routers, and will be negotiated when the session is initiated

Route Refresh

- Route Refresh uses the same command as Soft Reconfiguration:
- `R1#clear ip bgp 192.0.2.1 soft in`
- The difference is that the '`R1(config-router)#neighbor 192.0.2.1 soft-reconfiguration inbound`' command is not used
- Route Refresh and Soft Reconfiguration are mutually exclusive. If Soft Reconfiguration is enabled for the neighbor it will be used

Route Refresh Support Verification

```
R1#show ip bgp neighbors 192.0.2.1
BGP neighbor is 192.0.2.1,  remote AS 65001, external link
  BGP version 4, remote router ID 192.168.0.5
  BGP state = Established, up for 01:12:54
  Last read 00:00:12, last write 00:00:47, hold time is 180, keepalive interval
is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
! truncated
```

Troubleshooting

- Outbound Policy:

```
R1#sh ip bgp neighbors 192.0.2.1 advertised-routes
```

- Inbound Policy:

```
R1#sh ip bgp neighbors 192.0.2.1 received-routes
```

  - Requires Soft Reconfiguration to be enabled

```
R1#debug ip bgp <ip-address> updates
R1(config-router)#neighbor 192.0.2.1 soft-reconfiguration inbound
*Jul  3 21:45:50.152: BGP(0): 198.51.100.5 rcvd 192.0.2.0/24 -- DENIED due to: distribute/prefix-list;
*Jul  3 21:47:11.307: BGP(0): 198.51.100.5 rcv UPDATE about 203.0.113.0/24 -- DENIED due to: AS-PATH
contains our own AS;
```