

Stuff to Memorize

Friday, March 23, 2018 9:37 AM

- FF01::/16 – node-local
- FF02::/16 – link-local
- FF05::/16 – site-local
- FF08::/16 – organization-local
- FF0E::/16 – global

NAT64:

Use Network-specific prefix

Modify session during translation

NPTv6:

Modify IP header in transit

Map one IPv6 address prefix to another IPv6 prefix

OpenSent: wait for an OPEN message

OpenConfirm: wait for a KEEPALIVE or NOTIFICATION message

Established: UPDATE, NOTIFICATION and KEEPALIVE messages are exchanged with peers

Idle: refuse connections

Active: listen for and accept connection

Connect: wait for the connection to be completed

SVC: A circuit that provides temporary on-demand connections between DTEs

LMI: A signaling mechanism for Frame Relay devices

DLCI: A locally significant ID

FECN: An indicator of congestion on the network

PVC: A logical connection comprising two endpoints and a CIR

From <<https://www.digitaltut.com/new-route-questions-part-5>>

Type 9 link-state advertisements (LSAs) are Intra-Area Prefix LSAs and are unique to Open Shortest Path First version 3 (OSPFv3). OSPFv3 Type 9 LSAs carry IP version 6 (IPv6) prefix information, much like OSPF version 2 (OSPFv2) Type 1 and Type 2 LSAs carry IP version 4 (IPv4) prefix information. In OSPFv3, Type 1 and Type 2 LSAs no longer carry route prefixes. LSAs carry only routing information; they do not contain a full network topology. Both Type 9 LSAs and Type 8 LSAs are new in OSPFv3.

Type 8 LSAs are Link LSAs and are unique to OSPFv3. Type 8 LSAs are used to advertise the router's link-local IPv6 address, prefix, and option information. These LSAs are never flooded outside the local link.

Type 3 LSAs are Inter-Area Prefix LSAs for area border routers (ABRs). Type 3 LSAs are used to advertise internal networks to other areas. Like Type 9 LSAs, Type 3 LSAs also carry IPv6 prefix information.

Type 4 LSAs are Inter-Area Router LSAs for autonomous system boundary routers (ASBRs). Type 4 LSAs are used to advertise the location of an ASBR so that routers can determine the best next-hop path to an external network.

Type 5 LSAs are autonomous system (AS)-External LSAs. Type 5 LSAs are used to advertise external routes that are redistributed into OSPF.

ANSWER:

well-known NAT64 prefix	64:ff9b::/96
global unicast	2000::/3
6to4 unicast	2002::/16
site-local unicast	FC00::/7
link-local unicast	FE80::/10

Study/Workout Plans

Monday, March 19, 2018 11:17 AM

May 27 to June 6 (Exam June 11 @ Cisco Live)

Workout Schedule

May 26 - June 1

- CBT Nuggets = 9 Videos left = 3 a day.
- CCNP Dump - 105 questions = 15 questions each day

June 2 - 6

- CCNP dump - 58 questions = 11 questions each day.
- Take Boson exam 1 time each day.

June 7 - 10

- Refresh with CCNP Official Guide
- Take Boson exam 1 time each day

Goal: June 11 PASS CCNP ROUTE 300-101

Monday, Wednesday:

Abs and Oblique's

1 minute:

- Butt Kickers
- Lateral Jumps
- High knees
- Bend and Reach
- 1 minute x 3 reps
- Flutter Kicks
- Bicycle Crunches
- Mtn Climbers
- Toe Touch Crunches
- Russian Twists
- 10 min heavy bag

Tuesday, Thursday:

Chest and Triceps

1 minute:

- front plank
- left side plank
- right side plank
- Single leg over
- 1 minute x 3 reps
- Dumbbell incline press
- pushups
- dumbbell pullover
- decline pushups
- tricep pushdowns (cable)
- 5 min rowing machine

Friday :

Legs and Biceps

1 minute:

- Bend and Reach
- High knees
- Squats
- Single leg over
- 1 minute x 3 reps
- Dumbbell curls
- Bench Press
- Hanging Dips
- Leg Press
- Free weigh squats

Goal: Keep calories under (2100)/daily.

Breakfast - Bagel (71), Yogurt (150), Extreme Sausage Sandwich (650), Hash browns (150), tacos (172 ea), Trix/Special K (120).

Lunch - Turkey Sandwich (160), Chipotle Burrito (1050), Chipotle Bowl (890), Jimmy Johns Italian Club (560).

Dinner - **Keep your calories below 1100 throughout the day.** Steak (200), Chicken Breast (171), Ground Turkey (160), Pasta Noodles (160), Pasta Sauce (80), Veggies (below 100), white rice (1 cup 200).

Snacks - Pistachio (691 per cup), Beef Jerky (80 per oz), Hot Cheetos (160 1 oz bag), Blue Bell (180 1/2 cup).

BEST DAILY DIET - Yogurt (150) + Cereal (120), Turkey Sandwich (160), Steak (200) + white rice (200) + veggies (100), Blue Bell (180) + Pistachio (691) = 1801

May 27 to June 6 (Exam June 11 @ Cisco Live)

Workout Schedule

May 26 - June 1

- CBT Nuggets = 9 Videos left = 3 a day.
- CCNP Dump - 105 questions = 15 questions each day

June 2 - 6

- CCNP dump - 58 questions = 11 questions each day.
- Take Boson exam 1 time each day.

June 7 - 10

- Refresh with CCNP Official Guide
- Take Boson exam 1 time each day

Goal: June 11 PASS CCNP ROUTE 300-101

Monday, Wednesday:

Abs and Oblique's

1 minute:

- Butt Kickers
- Lateral Jumps
- High knees
- Bend and Reach
- 1 minute x 3 reps
- Flutter Kicks
- Bicycle Crunches
- Mtn Climbers
- Toe Touch Crunches
- Russian Twists
- 10 min heavy bag

Tuesday, Thursday:

Chest and Triceps

1 minute:

- front plank
- left side plank
- right side plank
- Single leg over
- 1 minute x 3 reps
- Dumbbell incline press
- pushups
- dumbbell pullover
- decline pushups
- tricep pushdowns (cable)
- 5 min rowing machine

Friday :

Legs and Biceps

1 minute:

- Bend and Reach
- High knees
- Squats
- Single leg over
- 1 minute x 3 reps
- Dumbbell curls
- Bench Press
- Hanging Dips
- Leg Press
- Free weigh squats

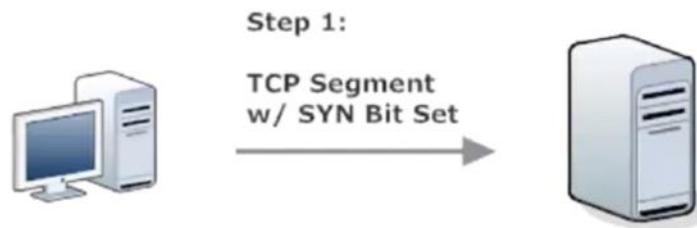
3/19/2018

5/24/2018 2:48 PM - Screen Clipping

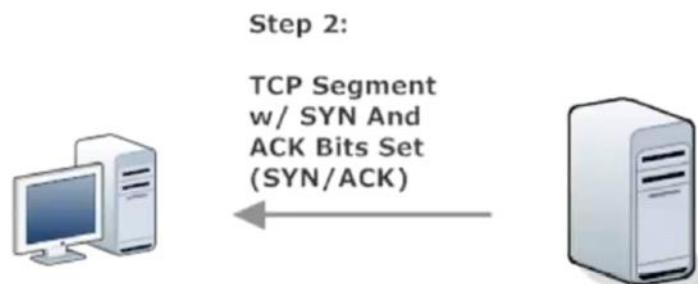
TCP vs UDP Part 1

10:48 AM

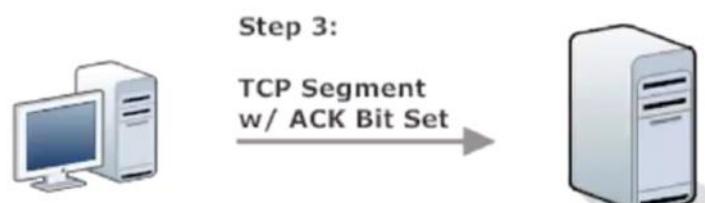
USB (Initial sequence number) is agreed upon during the handshake, via the syn bit set. The ack bit set agrees on the ISN.



The recipient responds with a TCP segment of its own, this one with the ACKnowledgement (see what I did there?) bit set in addition to the SYN bit.

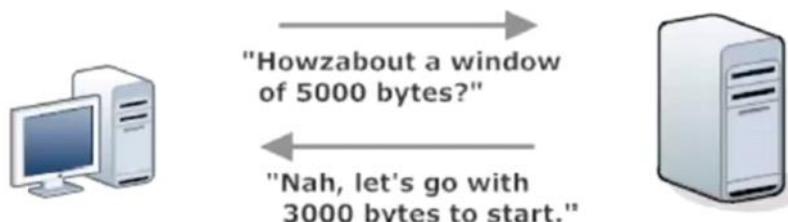


Part of that SYN/ACK is the server acknowledging receipt of the original SYN, so it makes sense for the server receiving the SYN/ACK to ACK it in turn. That's the final shake of our three-way handshake!



Flow Control and windowing:

How fast are we going? The TCP sliding window allows the recipient to adjust the amount of bytes being sent.



To enable TCP window scale on a Cisco router, use *ip tcp window-size*. The numeric value involved is bytes, which this particular IOS version assumes you know!

```
R1(config)#ip tcp ?
  async-mobility      Configure async-mobility
  chunk-size          TCP chunk size
  ecn                 Enable Explicit Congestion Notification
  intercept           Enable TCP intercepting
  mss                 TCP initial maximum segment size
  path-mtu-discovery Enable path-MTU discovery on new TCP connections
  queuemax            Maximum queue of outgoing TCP packets
  selective-ack       Enable TCP selective-ACK
```

Because TCP will use flow control and UDP will not, UDP will take bandwidth being surrendered by TCP sessions. In order to avoid this we use traffic classes in our QoS to indicate which traffic should be dropped when congestion occurs and which should not. Mixing UDP and TCP traffic without some form of QoS is a bad idea. You end up with lower throughput for your TCP traffic while turning into latency issues since UDP is taking for itself any bandwidth TCP gives up. Flow control can also lead to global synchronization. Basically, if a recipient of TCP traffic is overloaded, flow control kicks in and the host will throttle back on the transmission. Problem is, they all slow down at the same time creating peaks and valleys. Those peaks and valleys mean bandwidth is either underutilized or saturated. Weighted Random Early Detection is a tool to fight against global synch.

TCP

Source Port		Destination Port			
Sequence Number					
Acknowledgement Number					
Data Offset	Res.	Flags	Window		
Checksum		Urgent			
Options and Padding					
Data					

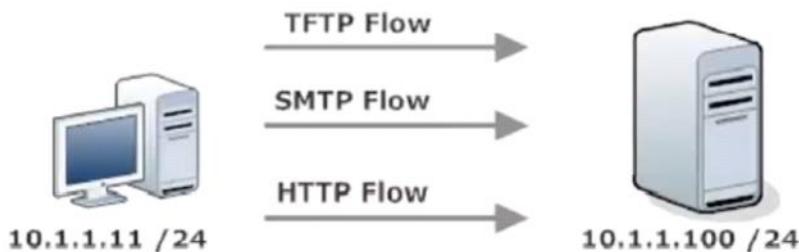
UDP

Source Port	Destination Port
Length	Checksum
Data	

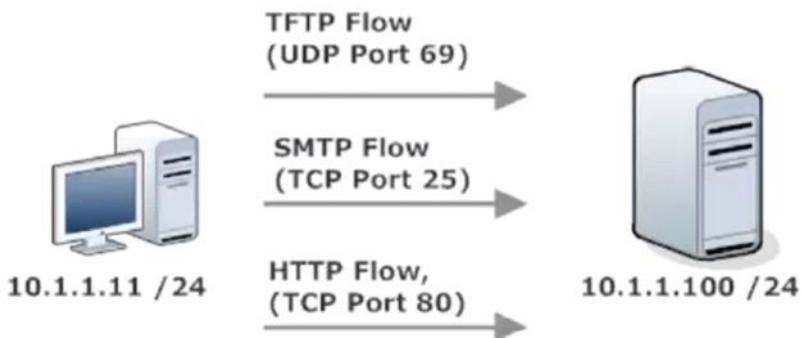
Everything you do on a router has a cost. Routing and network features included.

TCP vs UDP Part 2

Monday, February 12, 2018 11:03 AM



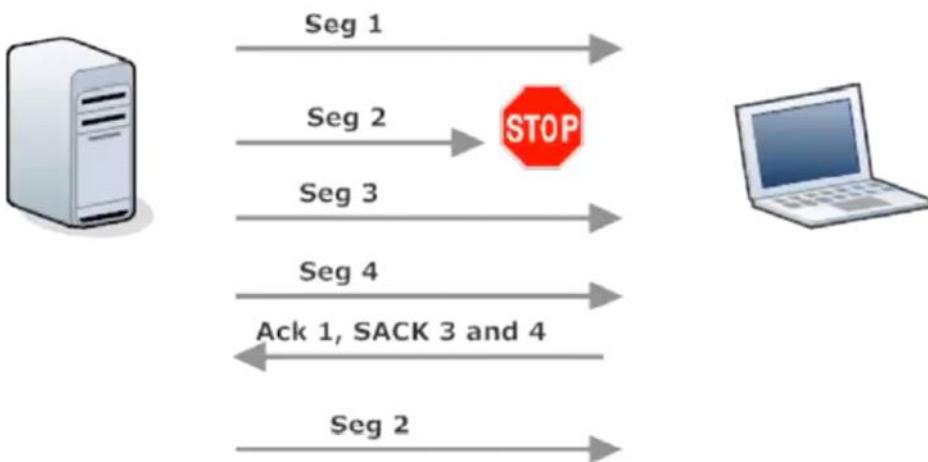
The server needs a way to keep incoming flows separate so it can send each type of data to the appropriate application. That's where well-known port numbers come in, including these three you bumped into during your CCNA studies!



This is why we use well known port numbers and multiplexing for the data streams.

Tcp sack:

TCP Selective Acknowledgements acknowledges segments that were received in addition to the segment being acknowledged.



Timestamps:

A quick word about those TCP timestamps while we're here! From ForensicsWiki.org:

"TCP timestamps are used to provide protection against wrapped sequence numbers. It is possible to calculate system uptime (and boot time) by analyzing TCP timestamps. These calculated uptimes (and boot times) can help in detecting hidden network-enabled operation systems, linking spoofed IP and MAC addresses together...etc."

TCP Keepalives:

These services are disabled by default as they are security risks. Don't touch them unless you need them for something specific.

tcp-keepalives-in Generate keepalives on idle incoming network connections

tcp-keepalives-out Generate keepalives on idle outgoing network connections

PPP - oE

Monday, February 12, 2018 11:13 AM

HDLC is the default. You must configure PPP encapsulation.



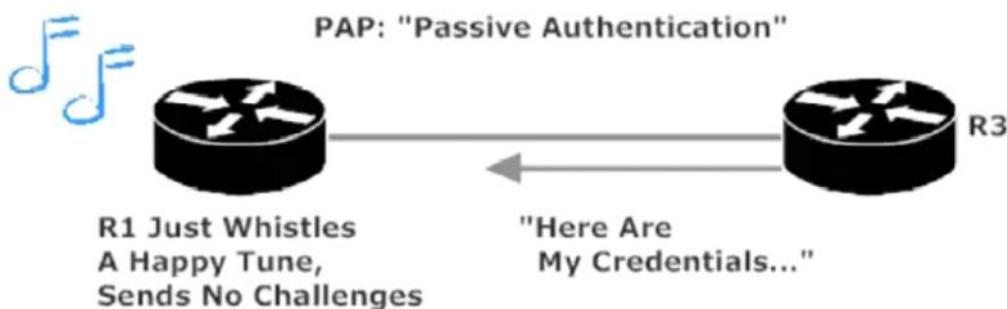
```
R1#show int serial 1/0
Serial1/0 is up, line protocol is up
  Hardware is CD2430 in sync mode
  Internet address is 172.12.123.1/24
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
```

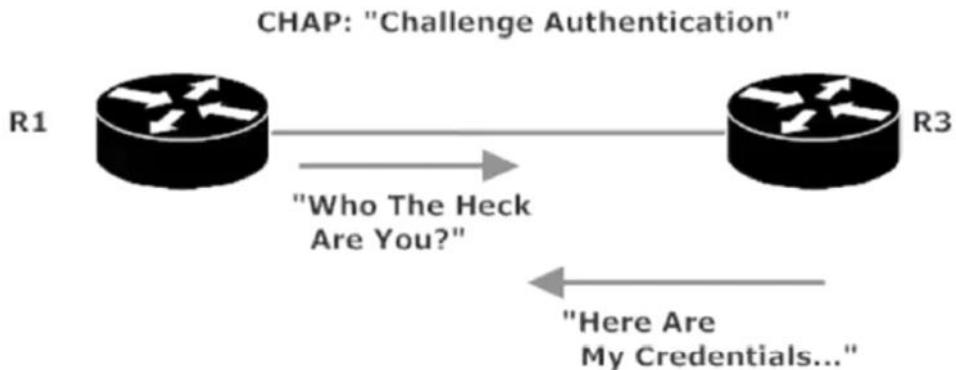
```
 R1(config)#int serial 1/0
R1(config-if)#encap ppp

R3(config)#int serial 1
R3(config-if)#encap ppp

R1#show int serial 1/0
Serial1/0 is up, line protocol is up
  Hardware is CD2430 in sync mode
  Internet address is 172.12.123.1/24
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open
```

HDLC has no authentication. PPP has PAP and CHAP. PAP sends its password in plain text. CHAP is not.





CHAP Configurations

```

R1(config)#int serial 1/0
R1(config-if)#ppp authen chap

R3(config)#int serial 1
R3(config-if)#ppp authen chap

R1(config)#username R3 password CCNP
R3(config)#username R1 password CCNP

```

DEBUG

```

2d01h: Sel CHAP: O CHALLENGE id 33 len 23 from "R3"
2d01h: Sel CHAP: I CHALLENGE id 50 len 23 from "R1"
2d01h: Sel CHAP: O RESPONSE id 50 len 23 from "R3"
2d01h: Sel CHAP: I RESPONSE id 33 len 23 from "R1"
2d01h: Sel CHAP: O SUCCESS id 33 len 4
2d01h: Sel CHAP: I SUCCESS id 50 len 4

```

That's just what we wanted to see! A challenge from each router, a response from each, and success for each authentication! Note the "O" and "I" letting us know whether the message is Outbound or Inbound.

```

2d01h: Sel PAP: I AUTH-REQ id 1 len 12 from "R1"
2d01h: Sel PAP: O AUTH-REQ id 1 len 12 from "R3"
2d01h: Sel PAP: Authenticating peer R1
2d01h: Sel PAP: O AUTH-ACK id 1 len 5
2d01h: Sel PAP: I AUTH-ACK id 1 len 5

```

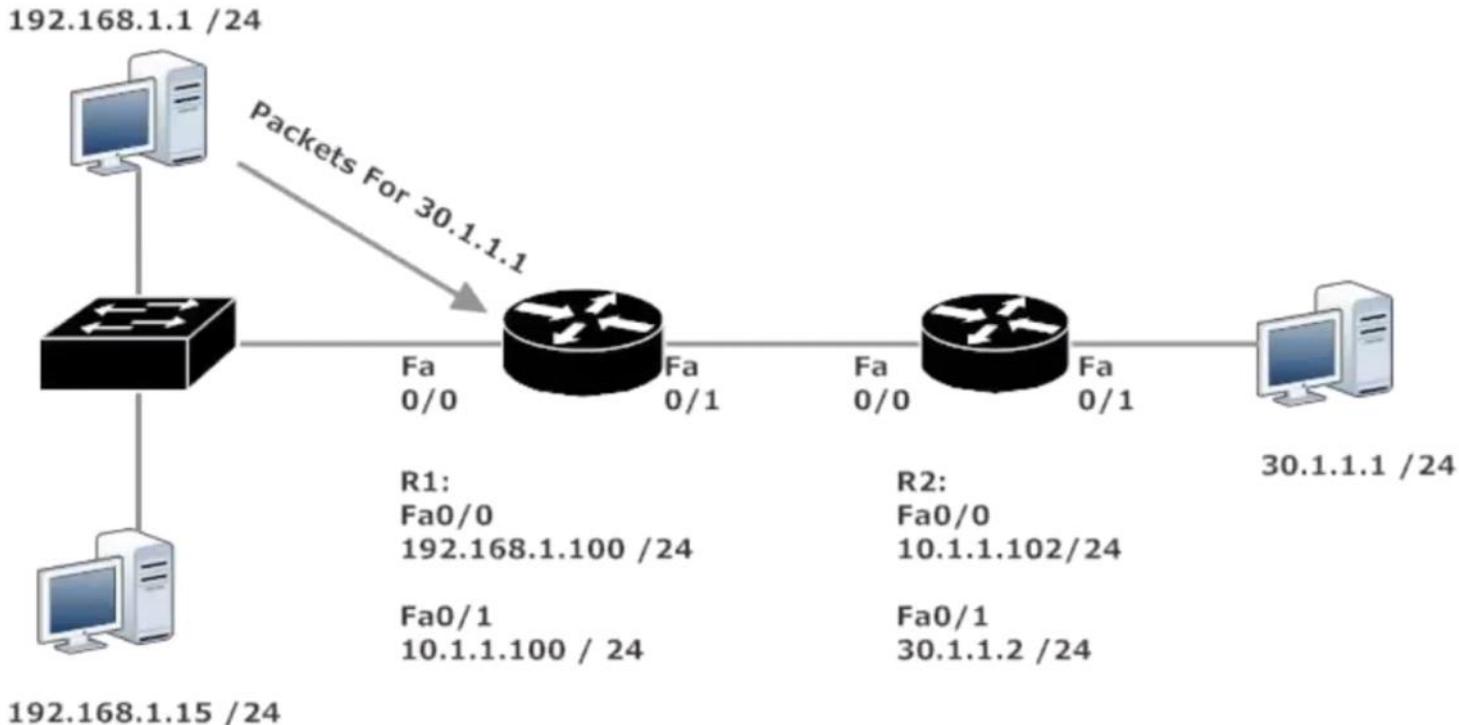
PPPoE is a combo of PPP and Ethernet that's used by ISPs. PPPoE clients are typically personal computers connected to an ISP over a remote broadband connections, such as DSL or cable. Supports high speed broadband access using the remote access infrastructure. The Active discovery phase of the PPPoE session is the first part where the PPPoE client and server negotiate.

EIGRP

Monday, February 12, 2018 10:38 AM

A router has 3 options for an entry:

- Either it has directly connected route
- A non directly connected route, but with an entry for the packet in its IP routing table
- A non directly connected route with no entry in its routing table.



A routing table lookup for 30.1.1.1 will fail, unless we put a static route.

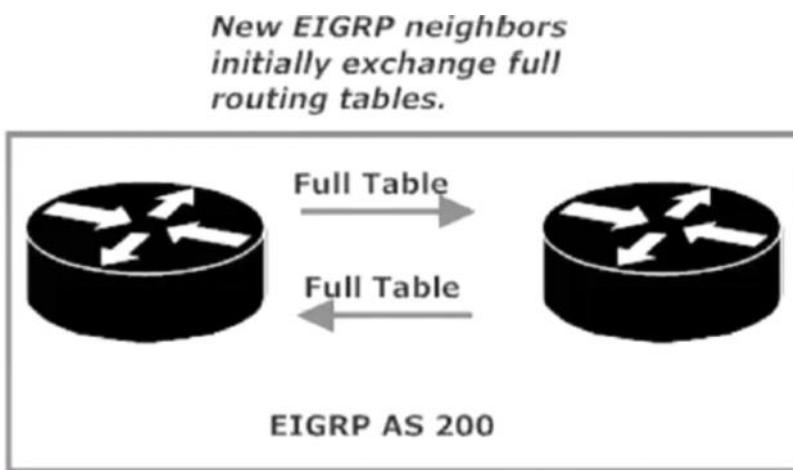
```
R1(config)#ip route 30.1.1.0 255.255.255.0 10.1.1.102
```

2/12/2018 11:30 AM - Screen Clipping

Enhanced Interior Gateway Routing Protocol. IGRP no longer used. Hybrid protocol, since it has link state and distance vector protocols. Full exchanges only happen when they first become neighbors. Otherwise updates will show only the change.

This, Rapid Convergence, is calculated using successors and feasible successors. Multicast for Hello packets is 224.0.0.10.

Uses AS to identify routers that belong to the same logical group. Separate AS cannot neighbor.
Adjacency will drop if Hello packets don't come through. Feasible successor is a valid loop free route who's metric isn't as good as the successor.

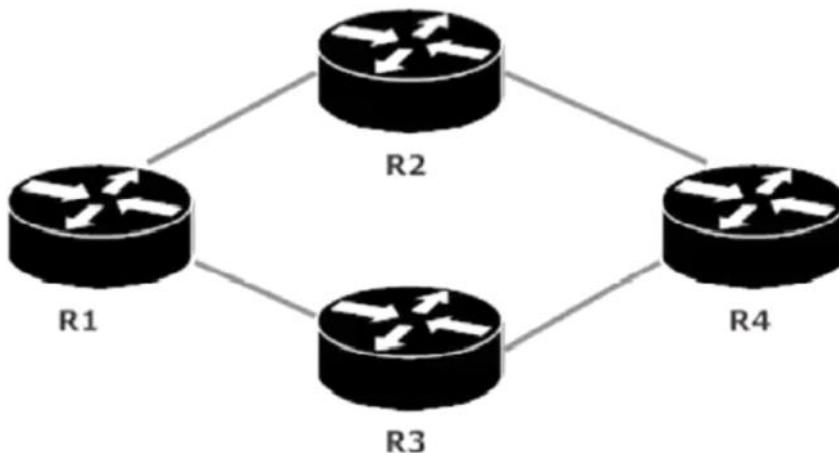


There are 3 different table from EIGRP:

The route table - stores best route to any known destination

Topology table - Keeps all known valid loop free routes to same network. (S and FS)

Neighbor table - Info on adjacency.



R1 to R4:

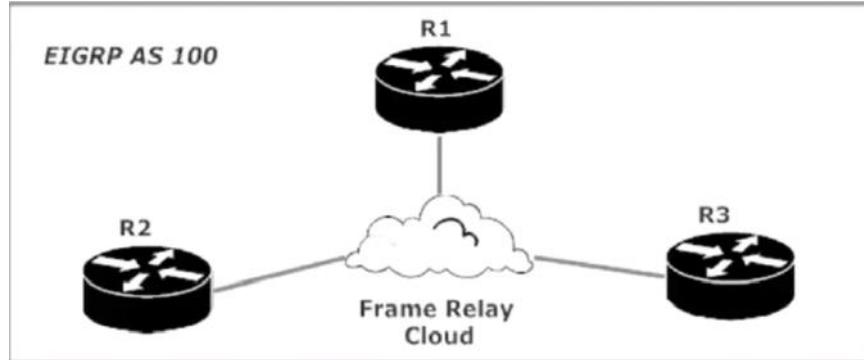
Successor: R2 is next hop

*Feasible Successor:
R3 is next hop*

EIGRP doesn't need to calculate new routes if one fails, because it has 2 backups.

NBMA (Non Broadcast Multiple Access) Network Lab

Monday, February 12, 2018 10:47 AM



Each router's Serial 0/1/0 interface is connected to the frame relay network, and we're using the 172.12.123.0 /24 subnet for IP addressing. The router numbers are used for the 4th octet of the address – R1's Serial 0/1/0 interface is 172.12.123.1, and so forth.

No Auto - No Auto summarization. It is not illegal to use only the network command with the IP and no wild card.

```
R1(config)#router eigrp 100
R1(config-router)#no auto
R1(config-router)#network 172.12.123.0 ?
  A.B.C.D  EIGRP wild card bits
  <cr>
R1(config-router)#network 172.12.123.0 0.0.0.255
```

```
R2(config)#router eigrp 100
R2(config-router)#no auto
R2(config-router)#network 172.12.123.0 0.0.0.255
R2(config-router)#
*May 18 14:18:37.971: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.12
Serial0/1/0) is up: new adjacency
```

```
R3(config)#router eigrp 100
R3(config-router)#no auto
R3(config-router)#network 172.12.123.0 0.0.0.255
R3(config-router)#
*May 18 14:06:03.227: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.12
Serial0/1/0) is up: new adjacency
```

Remember that EIGRP routes have to be discovered in order to populate the routing table. If they are directly connected, you won't see them either. Thus, the reason for loopbacks.

```
R2(config)#int loopback2
R2(config-if)#ip address 2.2.2.2 255.255.255.0
R2(config-if)#exit
R2(config)#^Z
```

```
R2(config)#router eigrp 100
R2(config-router)#network 2.2.2.2 0.0.0.0
R2(config-router)#^Z
R2#wr
```

```
R3(config)#int Loopback 3
R3(config-if)#ip address 3.3.3.3 255.255.255.0
R3(config-if)#
```

```
R3(config)#int Loopback 3
R3(config-if)#ip address 3.3.3.3 255.255.255.0
R3(config-if)#router eigrp 100
R3(config-router)#network 3.3.3.3 0.0.0.0
R3(config-router)#^Z
```

```
R1#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS le
      ia - IS-IS inter area, * - candidate default, U - per-user stati
      o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LIS
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
D 2.0.0.0/24 is subnetted, 1 subnets
D    2.2.2.0 [90/2297856] via 172.12.123.2, 00:00:21, Serial0/1/0
D 3.0.0.0/24 is subnetted, 1 subnets
D    3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:31, Serial0/1/0
```

Troubleshoot using debug:

Notice the "unroutable" message.

```
R3#debug ip packet
IP packet debugging is on
R3#ping 2.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

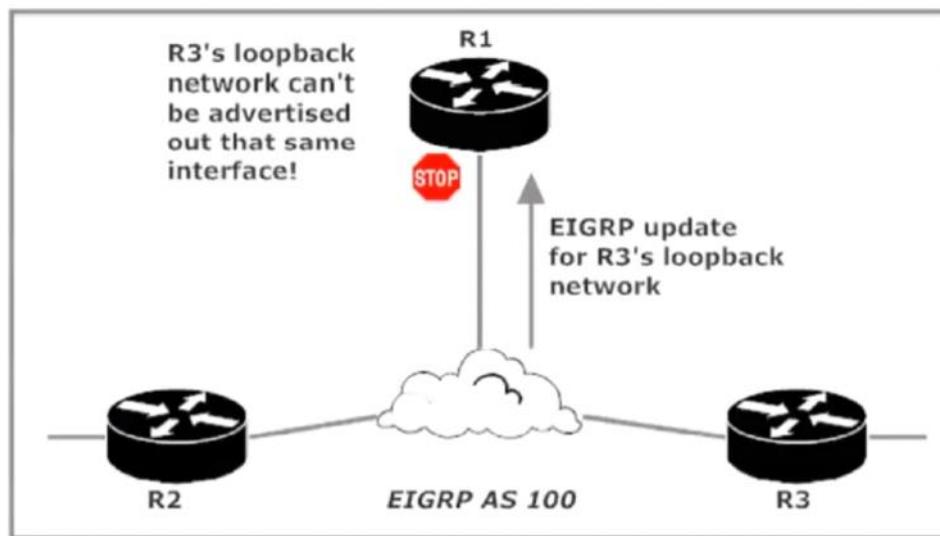
*May 18 14:14:17.331: IP: s=3.3.3.3 (local), d=224.0.0.10 (Loopback3), len
ending broad/multicast
*May 18 14:14:17.331: IP: s=3.3.3.3 (local), d=224.0.0.10 (Loopback3), len
ending full packet
*May 18 14:14:17.331: IP: s=3.3.3.3 (Loopback3), d=224.0.0.10, len 60, inpu
ture, MCI Check(67), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FA
*May 18 14:14:17.331: IP: s=3.3.3.3 (Loopback3), d=224.0.0.10, len 60, rcvd
*May 18 14:14:17.331: IP: s=3.3.3.3 (Loopback3), d=224.0.0.10, len 60, stop
ess pak for forus packet
*May 18 14:14:19.075: IP: s=3.3.3.3 (local), d=2.2.2.2, len 100, unrouteable
```

2/12/2018 11:54 AM - Screen Clipping

Split Horizon

Monday, February 12, 2018 12:23 PM

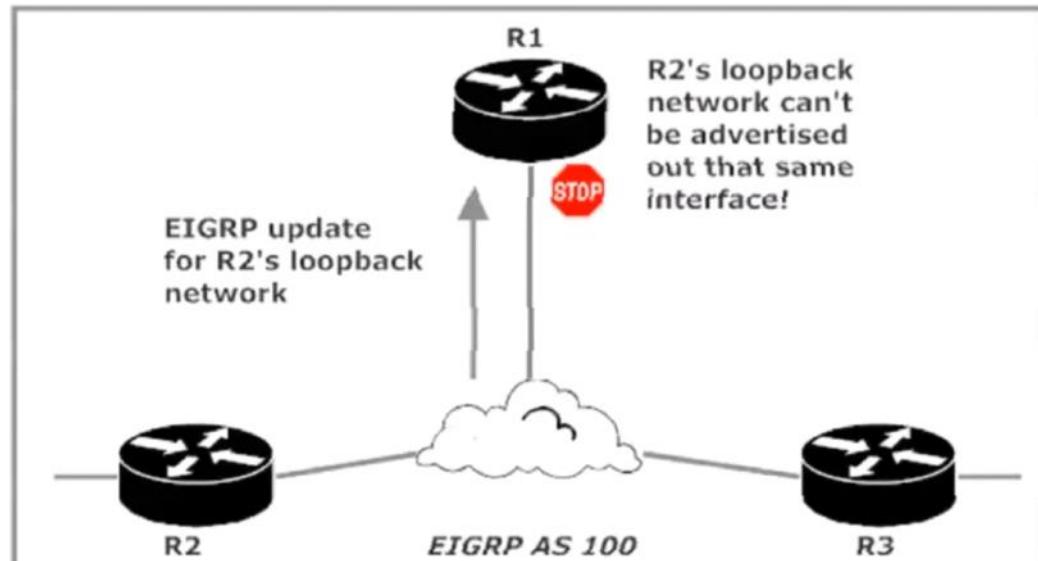
There is no direct connection between R2 and R3. Any traffic sent by one of those routers to the other must pass through the hub router (R1), and that's where split horizon comes in. The rule of split horizon dictates that a router cannot advertise a route back out the same interface upon which it was originally learned. R1 is learning about R3's loopback network via an EIGRP update received on Serial 0/1/0, so R1 can't advertise that same network out that same interface. Therefore, R2 can't learn about that network.



2/12/2018 12:28 PM - Screen Clipping

Split Horizon dictates a router can't advertise a route out of the same interface on which it was learned.

R1 is learning about R2's loopback network via an update received on Serial 0/1/0 as well, so R1 can't advertise that network via that same interface. R3 is left out in the cold!



2/12/2018 12:35 PM - Screen Clipping

```
R1(config)#int serial 0/1/0
R1(config-if)#no ip split-horizon ?
  eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)
<cr>

R1(config-if)#no ip split-horizon eigrp ?
<1-65535> AS number

R1(config-if)#no ip split-horizon eigrp 100
```

A few seconds later, I received these console messages:

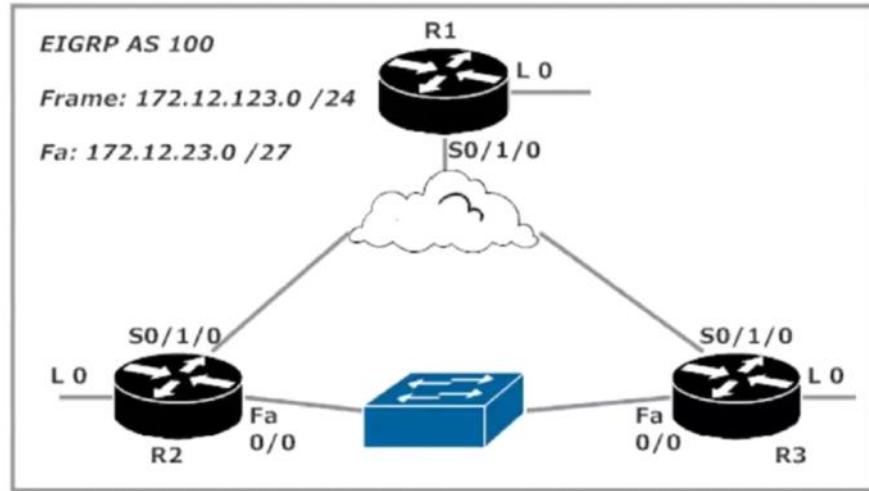
```
May  6 12:20:13: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.123.3
Serial 10/1/0) is resync: split horizon changed
May  6 12:20:13: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.123.2
Serial 10/1/0) is resync: split horizon changed
```

2/12/2018 12:36 PM - Screen Clipping

4/17/2018 3:04 PM - Screen Clipping

Successor and Feasible Successor

Tuesday, February 13, 2018 8:57 AM



We're now going to add the 172.12.23.0 /27 network to our lab. Addresses from that subnet will be used on the new Ethernet segment connecting R2 and R3.

```
R2(config)#router eigrp 100  
R2(config-router)#network 172.12.23.0 0.0.0.31
```

```
R3(config)#router eigrp 100  
R3(config-router)#network 172.12.23.0 0.0.0.31
```

```
R1#show ip route eigrp  
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-  
ia - IS-IS inter area, * - candidate default, U - per-user static ro  
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP  
+ - replicated route, % - next hop override
```

Gateway of last resort is not set

```
      2.0.0.0/24 is subnetted, 1 subnets  
D        2.2.2.0 [90/2297856] via 172.12.123.2, 00:02:31, Serial0/1/0  
      3.0.0.0/24 is subnetted, 1 subnets  
D        3.3.3.0 [90/2297856] via 172.12.123.3, 00:02:31, Serial0/1/0  
      172.12.0.0/16 is variably subnetted, 3 subnets, 3 masks  
D          172.12.23.0/27 [90/2172416] via 172.12.123.3, 00:02:31, Serial0/1/  
D          [90/2172416] via 172.12.123.2, 00:02:31, Serial0/1/0
```

Equal cost load balancing creates a seven digit metric that are exactly the same. EIGRP has decided to perform ecld over the two paths.

```
      172.12.0.0/16 is variably subnetted, 3 subnets, 3 masks  
D          172.12.23.0/27 [90/2172416] via 172.12.123.3, 00:02:31, Serial0/1/  
D          [90/2172416] via 172.12.123.2, 00:02:31, Serial0/1/0
```

In the topology table you will see the successors and feasible successors.

FD = Successor route.

```
R1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R -
      r - reply Status, s - sia Status

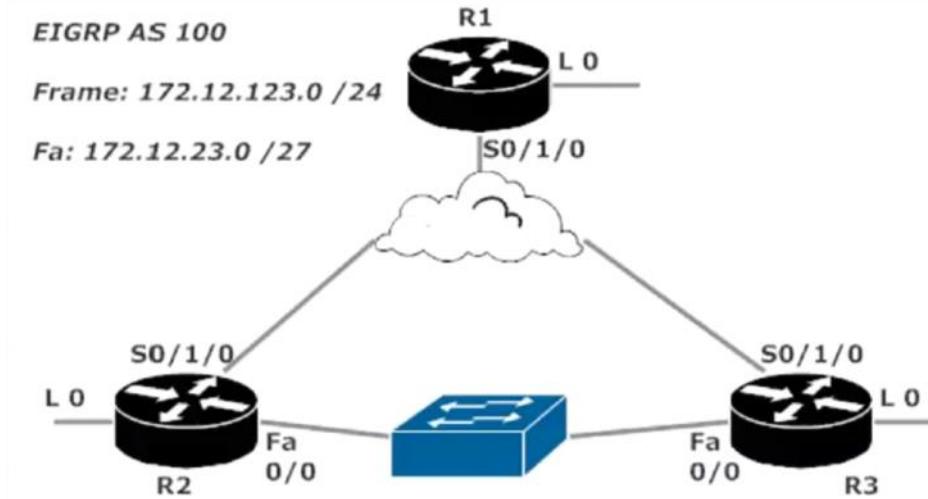
P 172.12.123.0/24, 1 successors, FD is 2169856
    via Connected, Serial0/1/0
P 172.12.23.0/27, 2 successors, FD is 2172416
    via 172.12.123.2 (2172416/28160), Serial0/1/0
    via 172.12.123.3 (2172416/28160), Serial0/1/0
P 2.2.2.0/24, 1 successors, FD is 2297856
    via 172.12.123.2 (2297856/128256), serial0/1/0
    via 172.12.123.3 (2300416/156160), Serial0/1/0
P 3.3.3.0/24, 1 successors, FD is 2297856
    via 172.12.123.3 (2297856/128256), serial0/1/0
    via 172.12.123.2 (2300416/156160), Serial0/1/0
P 1.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
```

FD = Successor route.

Variance

Tuesday, February 13, 2018 9:22 AM

How do you bring FS into the routing table to use them for unequal cost load balancing?



We have equal cost load balancing happening on router 1, but unequal on 2 and 3.

```
R1#show ip eigrp top
EIGRP-IPv4 Topology Table for AS(100)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, F
      r - reply Status, s - sia Status

P 172.12.123.0/24, 1 successors, FD is 2169856
    via Connected, Serial0/1/0
P 172.12.23.0/27, 2 successors, FD is 2172416
    via 172.12.123.2 (2172416/28160), Serial0/1/0
    via 172.12.123.3 (2172416/28160), Serial0/1/0
P 2.2.2.0/24, 1 successors, FD is 2297856
    via 172.12.123.2 (2297856/128256), Serial0/1/0
    via 172.12.123.3 (2300416/156160), Serial0/1/0
P 3.3.3.0/24, 1 successors, FD is 2297856
    via 172.12.123.3 (2297856/128256), Serial0/1/0
    via 172.12.123.2 (2300416/156160), Serial0/1/0
P 1.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
```

The variance command enables unequal cost load balancing in EIGRP. The variance command is simply a multiplier, which the route multiplies the feasible distance by the variance value. Any FD with a metric less than that new value will be entered into the routing table.

```
P 2.2.2.0/24, 1 successors, FD is 2297856
    via 172.12.123.2 (2297856/128256), Serial0/1/0
    via 172.12.123.3 (2300416/156160), Serial0/1/0
P 3.3.3.0/24, 1 successors, FD is 2297856
```

```
R1(config-router)#variance 2  
R1(config-router)#[
```

With the variance enabled.

```
R1#show ip route eigrp  
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS  
ia - IS-IS inter area, * - candidate default, U - per-user sta-  
o - ODR, P - periodic downloaded static route, H - NHRP, l - L  
+ - replicated route, % - next hop override
```

Gateway of last resort is not set

```
      2.0.0.0/24 is subnetted, 1 subnets  
D        2.2.2.0 [90/2300416] via 172.12.123.3, 00:00:18, Serial0/1/0  
                  [90/2297856] via 172.12.123.2, 00:00:18, Serial0/1/0  
      3.0.0.0/24 is subnetted, 1 subnets  
D        3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:18, Serial0/1/0  
                  [90/2300416] via 172.12.123.2, 00:00:18, Serial0/1/0  
  172.12.0.0/16 is variably subnetted, 3 subnets, 3 masks  
D        172.12.23.0/27 [90/2172416] via 172.12.123.3, 00:00:18, Seri  
[90/2172416] via 172.12.123.2, 00:00:18, Seri  
[90/2172416] via 172.12.123.1, 00:00:18, Seri
```

Note the metric of the FS (2300416) didn't change. Variance will only make a higher metric more acceptable to the routing table. **Show IP protocol** is the command to check your variance.

```
R1#show ip protocol  
*** IP Routing is NSF aware ***  
  
Routing Protocol is "eigrp 100"  
  Outgoing update filter list for all  
  Incoming update filter list for all  
  Default networks flagged in outgoing updates  
  Default networks accepted from incoming updates  
  Redistributing: eigrp 100  
  EIGRP-IPv4 Protocol for AS(100)  
    Metric weight K1=1, K2=0, K3=1, K4=0  
    NSF-aware route hold timer is 240 seconds  
    Router-ID: 1.1.1.1  
    Topology : 0 (base)  
      Active Timer: 3 min  
      Distance: internal 90 external 150  
      Maximum path: 4  
      Maximum hopcount 100  
      Maximum metric variance 2
```

Paths:

Default is 4, but can go up to 32 paths. Imagine Path 1 has a metric of 5000, path 7000, and path 3 55k. That gives us two links close in speed and then a one that's trash. Perhaps you don't want to load balance that last path, but still want to utilize the backup path. Select the number of paths available instead of increasing variance?

```
R1(config)#router eigrp 100
R1(config-router)#maximum-path ?
    <1-32> Number of paths
```

DUAL, Passive, and Active routes

Tuesday, February 13, 2018 1:22 PM

What if there are no feasible successors for a cut over?

DUAL calculates route metrics and query's neighbor routers. The first thing it does, is marks the route in question as active. That doesn't mean its actively used or routing data. Instead it means that EIGRP is actively calculating the route. You want your routes to be passive, not active because that means the routes are stable. If they're using DUAL they're active.

```
R1#show ip eigrp top
EIGRP-IPv4 Topology Table for AS(100)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query,
       r - reply Status, s - sia Status

P 172.12.123.0/24, 1 successors, FD is 2169856
    via Connected, Serial0/1/0
P 172.12.23.0/27, 2 successors, FD is 2172416
    via 172.12.123.2 (2172416/28160), Serial0/1/0
    via 172.12.123.3 (2172416/28160), Serial0/1/0
P 2.2.2.0/24, 2 successors, FD is 2297856
    via 172.12.123.2 (2297856/128256), Serial0/1/0
    via 172.12.123.3 (2300416/156160), Serial0/1/0
P 3.3.3.0/24, 2 successors, FD is 2297856
    via 172.12.123.3 (2297856/128256), Serial0/1/0
    via 172.12.123.2 (2300416/156160), Serial0/1/0
P 1.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
```

If a route is lost, DUAL will query other routers. The process continues until a queried router replies with the desired route or they run out of other routers to ask. As soon as DUAL is no longer needed, P will show up.



*DUAL
Query*



"Hey, neighbor! Do you have a valid path to (Route X), the destination I just lost?"

ADV EIGRP: Packet Types and Timers

Tuesday, February 13, 2018 1:40 PM

Hello Packets = keep alive for adjacency and discover neighbors.

Ack packets = hello packets with no data

Neither use RTP and are considered unreliable.

Update packets = sent to new neighbors, when there's a change in the network

Query Packets = sent when a router loses a successor and has no FS.

Replay packets - sent in response to query packets. Indicates a new router to the destination has been found.

All three use RTP

Multicast = 224.0.0.10

To see how many of these packets have passed through, run *show ip eigrp traffic*.

```
R1#show ip eigrp traffic
EIGRP-IPv4 Traffic Statistics for AS(100)
    Hellos sent/received: 4412/5447
    Updates sent/received: 44/38
    Queries sent/received: 2/5
    Replies sent/received: 5/2
    Acks sent/received: 39/45
```

When update, query, and reply packets remain stable, you've got a stable network.

Initial Route Exchange:

On a serial interface. In this case a unicast, instead of multicast, update packet is sent. R1 then sends an EIGRP ack back to R2, letting R2 know of the routes received.



R1 will then send an ACK and Update containing all the EIGRP routes R1 knows about. R2 will then send an ack and the route exchange is complete. This only happens during initial discovery, from their updates will be all that occur.

EIGRP Adjacency Issues:

EIGRP hold time has the same function as an OSPF dead timer. They both define the amount of time in which a hello packet must be received in order to retain the neighbor relationship.

Hold time = default time is 3 x hello packet interval. You can see the times in bold below.

```
R3#show ip eigrp interfaces detail fast0/0
EIGRP-IPv4 Interfaces for AS(100)
```

Interface	Xmit Peers	Queue	Mean	Pacing Time	Multicast	Pending Routes
		Un/Reliable	SRTT	Un/Reliable	Flow Timer	Routes
Fa0/0	1	0/0	1218	0/1	6092	0

Hello-interval is 5, Hold-time is 15

Split-horizon is enabled

Next xmit serial <none>

Un/reliable mcasts: 0/17 Un/reliable ucasts: 34/12

Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0

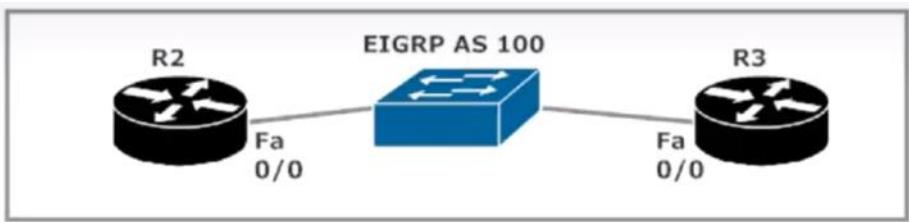
Retransmissions sent: 0 Out-of-sequence rcvd: 2

Topology-ids on interface - 0

Authentication mode is not set

ADV EIGRP Hello timers lab

Tuesday, February 13, 2018 2:06 PM



Let's verify adjacencies and then screw around with the config a bit!

```
R2#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address           Interface      Hold Uptime    SRTT      RTO      Q      Sec
   Address           Interface      (sec) (ms)      (ms)      Cnt  Num
1   172.12.23.3       Fa0/0          11  06:58:47    4     200    0    36
```

```
R3#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address           Interface      Hold Uptime    SRTT      RTO      Q      Sec
   Address           Interface      (sec) (ms)      (ms)      Cnt  Num
1   172.12.23.2       Fa0/0          10  07:00:32    5     200    0    47
```

Let's change the hello timers for EIGRP. It's on a per interface level.

```
R3(config)#int fast 0/0
R3(config-if)#ip hello ?
  eigrp  Enhanced Interior Gateway Routing Protocol
R3(config-if)#ip hello eigrp ?
  <1-65535>  AS number
R3(config-if)#ip hello eigrp 100 ?
  <1-65535>  Seconds between hello transmission
R3(config-if)#ip hello eigrp 100 7 ?
  <cr>
R3(config-if)#ip hello eigrp 100 7
```

Notice the hold times don't change even though the hello timer does.

```

R3#show ip eigrp int detail
EIGRP-IPv4 Interfaces for AS(100)
          Xmit Queue  Mean   Pacing T
Interface    Peers Un/Reliable SRTT  Un/Relia
Lo3           0      0/0       0      0/1
Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Un/reliable mcasts: 0/0  Un/reliable ucasts: 0/0
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 0  Out-of-sequence rcvd: 0
Topology-ids on interface - 0
Authentication mode is not set
Fa0/0          1      0/0       128     0/1
Hello-interval is 7, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Un/reliable mcasts: 0/3  Un/reliable ucasts: 5/3
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 1  Out-of-sequence rcvd: 2
Topology-ids on interface - 0

```

Now, say we decide to change the hello timer to 30 seconds. We will then see the interface going up and down. This is a **flapping adjacency**, this is what happens when the hello timers are changed but not synced.

```

R3(config)#int fast 0/0
R3(config-if)#ip hello eigrp 100 30
R3(config-if)#^Z
R3#
*May 20 17:27:25: %SYS-5-CONFIG_I: Configured from console by console
R3#
*May 20 17:27:41: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23.2
ethernet0/0) is down: Interface PEER-TERMINATION received
R3#
*May 20 17:27:45: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23.2
ethernet0/0) is up: new adjacency
R3#
*May 20 17:28:00: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23.2
ethernet0/0) is down: Interface PEER-TERMINATION received
R3#
*May 20 17:28:05: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23.2
ethernet0/0) is up: new adjacency
R3#
*May 20 17:28:20: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23.2
ethernet0/0) is down: Interface PEER-TERMINATION received

```

When the hellos do come through, the routers reset the timer causing the link to go up and down. Even though **the timer don't have to be the same on each router** its best practice not to set them higher than the hold time. Hold times are not dynamically changed.

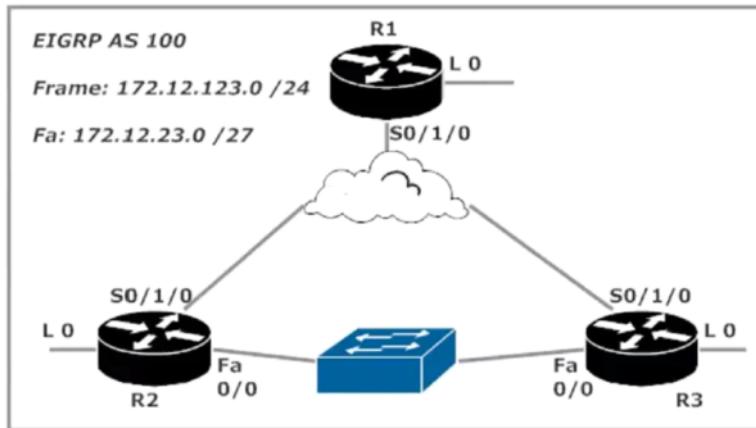


"It's been 15 seconds
since I got a hello from
R3, so I'm dropping
the adjacency."

*Sending Hellos
Every 30 Seconds*

ADV EIGRP: Feasible, Reported, and Advertised Distances

Tuesday, February 13, 2018 2:27 PM



Here are the EIGRP topology table entries for 172.12.23.0 /27 on R1:

```
P 172.12.23.0/27, 2 successors, FD is 2172416
via 172.12.123.2 (2172416/28160), Serial0/1/0
via 172.12.123.3 (2172416/28160), Serial0/1/0
```

The first number 2172416 is the full metric of the route or its feasible distance from route to destination network. The full trip.

28160 is the advertised distance or reported distance, which is the metric from the next hop router to the destination. Only visible in EIGRP topology. Distance advertised to local router.

Why do we have this?

There are loop free routes, which are defined by the AD which has to be less than the feasible distance.

These distances are also used by EIGRP to determine feasible successors. Let's walk through this important process using R1's topology entries for 3.3.3.0 /24.

```
P 3.3.3.0/24, 1 successors, FD is 2297856
via 172.12.123.3 (2297856/128256), Serial0/1/0
via 172.12.123.2 (2300416/156160), Serial0/1/0
```

The table is kind enough to tell us the successor's FD is 2297856, which matches that of the route using 172.12.123.3 as the next-hop address. What of the other route? It can't be a successor, since its FD is higher than the FD of the successor. Can the other route possibly be a feasible successor? It can, because its *advertised*

THE ROUTES AD IS LESS THAN THE FD OF THE SUCCESSOR - THAT'S THE FEASIBILITY CONDITION.

Successor: FD 5, AD 4

Successor

Possible FS 1: FD 9, AD 7

Can't be FS

Possible FS 2: FD 8, AD 6

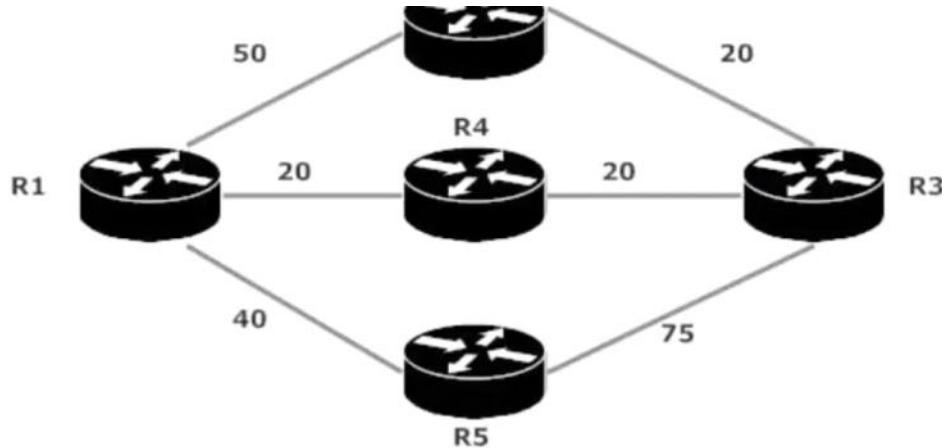
Can't be FS

Possible FS 2: FD 8, AD 6

Can't be FS

Possible FS 3: FD 6, AD 4

Can be FS



Before we even begin with the feasibility condition, we have to know the AD and FD for each path! The advertised distance is the next-hop router's metric to the destination, and the feasible distance is the local router's metric to that same destination. From R1's point of view, the FD and AD of each path:

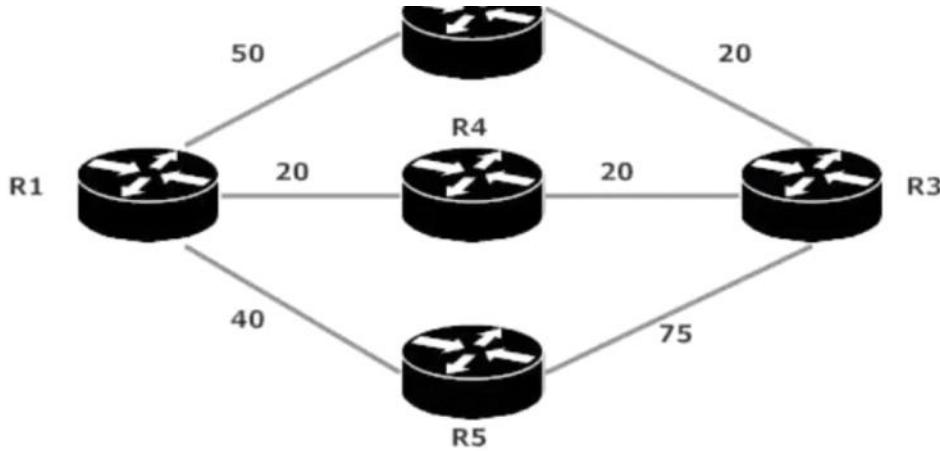
R1 – R4 – R3: FD 40, AD 20

R1 – R2 – R3: FD 70, AD 20

R1 – R5 – R3: FD 115, AD 75

ADV EIGRP: FD, AD, and Variance

Wednesday, February 14, 2018 11:44 AM



Before we even begin with the feasibility condition, we have to know the AD and FD for each path! The advertised distance is the next-hop router's metric to the destination, and the feasible distance is the local router's metric to that same destination. From R1's point of view, the FD and AD of each path:

R1 – R4 – R3: FD 40, AD 20

R1 – R2 – R3: FD 70, AD 20

R1 – R5 – R3: FD 115, AD 75

The path with the best metric or shortest path (40) is going to be the successor. You FS (70) can be a FS, but (115) cannot.

The Feasibility Condition and Variance

A route that doesn't meet the feasibility condition cannot be used in unequal-cost load balancing. In the previous example, the R1 – R5 – R3 path couldn't be used for load balancing.

It's a great idea to write out the FD and AD of all routes in your network before deciding on the variance value and to determine whether all of the desired paths can actually be used in unequal-cost load balancing... as in the following illustration!

ADV EIGRP: Default and Non-Default Administrative Distances

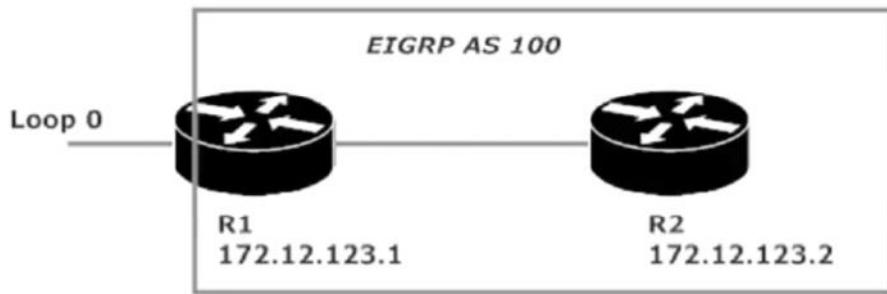
Wednesday, February 14, 2018 12:17 PM

EIGRP AD

Internal routes - 90 "D" code.

External routers - 170 - Route redistributions - Code D EX

Summary routes - 5 -



Loopback 0 is not a part of the adjacency.

```
R2(config-router)#distance eigrp ?
<1-255> Distance for internal routes
R2(config-router)#distance eigrp 90 ?
<1-255> Distance for external routes
R2(config-router)#distance eigrp 90 200 ?
<cr>
R2(config-router)#distance eigrp 90 200
R2(config-router)#^Z
R2#
```

```
Gateway of last resort is not set

    1.0.0.0/24 is subnetted, 1 subnets
D EX      1.1.1.0 [170/2297856] via 172.12.123.1, 00:02:25, Serial0/
R2#
*Jun 3 13:06:54.606: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1
Serial0/1/0) is down: route configuration changed
*Jun 3 13:06:54.606: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1
FastEthernet0/0) is down: route configuration changed
R2#
*Jun 3 13:06:56.470: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1
FastEthernet0/0) is up: new adjacency
R2#
*Jun 3 13:07:22.898: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1
Serial0/1/0) is up: new adjacency
R2#
R2#
R2#
R2#
```

Gateway of last resort is not set

```
    1.0.0.0/24 is subnetted, 1 subnets
D EX      1.1.1.0 [200/2297856] via 172.12.123.1, 00:00:43, Serial0/1/0
```

This AD has changed. So for this same route on R3 the AD is going to be 170, because when you use the distance EIGRP command only effects the local router. It will not effect another router.

```
R2(config-router)#distance eigrp 90 100
```

```
    %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.123.1 (Serial0/1/0) is
down: route configuration changed
    %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.123.1 (Serial0/1/0) is up:
new adjacency
```

```
R2#show ip route eigrp
    1.0.0.0/24 is subnetted, 1 subnets
D EX      1.1.1.0 [100/2297856] via 172.12.123.1, 00:00:07, Serial0/1/0
            3.0.0.0/24 is subnetted, 1 subnets
D          3.3.3.0 [90/156160] via 172.12.23.3, 00:00:34, FastEthernet0/0
```

We'll see the third EIGRP AD in action in the next section, which just happens to start right now!

EIGRP Route Summarization: Automatic and Manual

EIGRP has a pesky little default behavior. *Autosummarization* occurs when routes are advertised across classful network boundaries. That doesn't sound all that bad,

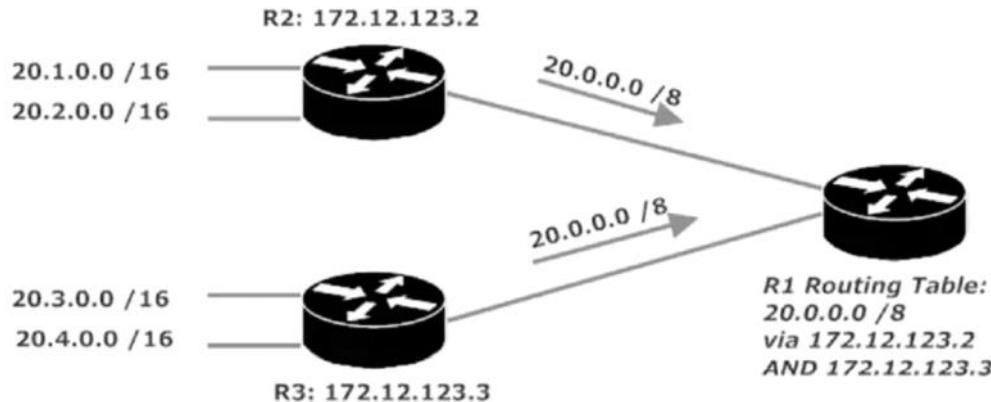
ADV EIGRP: Route Summarization

Thursday, February 15, 2018 11:57 AM

ROUTE SUMMARIZATION

Auto summarizations occur when routes are advertised across classful network boundaries. Discontiguous networks, networks separated by another network number. With AutoSum R1 will receive two advertisements for the 20.0 network.

AutoSum, or automatic major network number, 20.0.0.0 /8



Autosummarization

```
Maximum hopcount 100
Maximum metric variance 1

Automatic Summarization: enabled
  172.12.0.0/16 for Lo0, Lo1
    Summarizing 1 component with metric 2169856
  20.0.0.0/8 for Se0/1/0
    Summarizing 2 components with metric 128256
Maximum path: 4
Routing for Networks:
  20.3.0.0/16
  20.4.0.0/16
```

```
R1#
R1#show ip route eigrp
D  20.0.0.0/8 [90/20640000] via 172.12.123.3, 00:06:18, Serial1/0
               [90/20640000] via 172.12.123.2, 00:06:18, Serial1/0
R1#
```

You're getting two of the same route. You'll get some packets through, but still won't get 100%.

What if you had to disable autosummary only where it's necessary to resolve a problem?

You only need to disable it where the summarization is actually taking place on R2 and R3.

```

R2(config)#router eigrp 100
R2(config-router)#no auto
R2(config-router)#^Z
R2#
*Jun 3 16:41:18.203: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172
Serial0/1/0) is resync: summary configured
*Jun 3 16:41:19.071: %SYS-5-CONFIG_I: Configured from console by cor
R2#

```

```

R3#
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router eigrp 100
R3(config-router)#no auto
R3(config-router)#^Z
R3#
*Jun 3 16:28:36: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.123.1

```

Summary routes are gone and now we see the subnets.

```

R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
H   Address           Interface      Hold Uptime    SRTT     RTO
  (sec)   (ms)
1   172.12.123.2      Sel/0          169  00:09:13   38    1140
0   172.12.123.3      Sel/0          173  03:05:20   55    1140
R1#show ip route eigrp
  20.0.0.0/16 is subnetted, 4 subnets
D     20.4.0.0 [90/20640000] via 172.12.123.3, 00:00:30, Serial1/0
D     20.1.0.0 [90/20640000] via 172.12.123.2, 00:01:04, Serial1/0
D     20.2.0.0 [90/20640000] via 172.12.123.2, 00:01:04, Serial1/0
D     20.3.0.0 [90/20640000] via 172.12.123.3, 00:00:30, Serial1/0
R1#

```

ADV EIGRP: Manual Summarization

Tuesday, February 27, 2018 1:47 PM

MANUAL SUMMARIZATION

When used at the proper points, summarization is useful. Why?

The routing tables are smaller, making the entire routing process faster.

Since the tables are smaller, the load on the CPU from the routing process is lessened

Routing updates themselves are smaller

The more-specific network numbers are hidden, a small boost to our overall network security plan

The impact of flapping links on the rest of the network is lessened

Overall number of EIGRP queries can be lessened.

In this lab, R1 has seven subnets it's advertising to EIGRP neighbor R2.



We can have 1 single line summarized into one.

```
Gateway of last resort is not set
```

```
100.0.0.0/16 is subnetted, 7 subnets
D 100.1.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.2.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.3.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.4.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.5.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.6.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
D 100.7.0.0 [90/2297856] via 172.12.123.1, 00:14:48, Serial0/1/0
R2#ping 100.5.1.1
```

```
Type escape sequence to abort
```

So how do you summarize? Write it out in binary.

```

100.1.0.0      01100100  00000001  00000000  00000000
100.2.0.0      01100100  00000010  (every route ends in 16 zeroes)
100.3.0.0      01100100  00000011
100.4.0.0      01100100  00000100
100.5.0.0      01100100  00000101
100.6.0.0      01100100  00000110
100.7.0.0      01100100  00000111

```

That's actually the hardest part of summarizing routes!

You now need a summary route and mask.
To get the summary work from left to right and identify the common bits.

Our common route is **100.0.0.0** and the mask is **/13** or **255.248.0.0**

We then configure AutoSumm from the INTERFACE. Not EIGRP.

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int serial 1/0
R1(config-if)#ip summary-address ?
  eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)
  rip   Routing Information Protocol (RIP)

R1(config-if)#ip summary-address eigrp ?
  <1-65535> Autonomous system number

R1(config-if)#ip summary-address eigrp 100

```

```

R1(config-if)#ip summary-address eigrp 100 100.0.0.0 ?
  A.B.C.D IP network mask

R1(config-if)#ip summary-address eigrp 100 100.0.0.0 255.248.0.0
R1(config-if)^Z
R1#
*May 26 00:40:02.814: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 1
(Serial1/0) is resync: summary configured
*May 26 00:40:02.814: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 1
(Serial1/0) is resync: summary configured
*May 26 00:40:03.708: %SYS-5-CONFIG_I: Configured from console by co
R1#

```

That's all for the summary configuration. Now do **ip route eigrp** on R2. Notice it has an AD of 90, but not on R1.

```

          100.0.0.0/13 is subnetted, 1 subnets
D          100.0.0.0 [90/2297856] via 172.12.123.1, 00:00:44, serial0/1/0

R1#show ip route eigrp
          100.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
D          100.0.0.0/13 is a summary, 00:02:11, Null0

```

Null0 is a routing loop prevention mechanism. It's a trashcan.

```
Gateway of last resort is not set

      100.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C        100.4.0.0/16 is directly connected, Loopback104
C        100.5.0.0/16 is directly connected, Loopback105
C        100.6.0.0/16 is directly connected, Loopback106
C        100.7.0.0/16 is directly connected, Loopback107
D        100.0.0.0/13 is a summary, 00:04:20, Null0
C        100.1.0.0/16 is directly connected, Loopback101
C        100.2.0.0/16 is directly connected, Loopback102
C        100.3.0.0/16 is directly connected, Loopback103
      172.12.0.0/24 is subnetted, 1 subnets
C          172.12.123.0 is directly connected, Serial1/0
R1#
```

2/27/2018 2:17 PM - Screen Clipping

ADV EIGRP: The AD 5 and Stub Routing Theory

Tuesday, February 27, 2018 2:17 PM

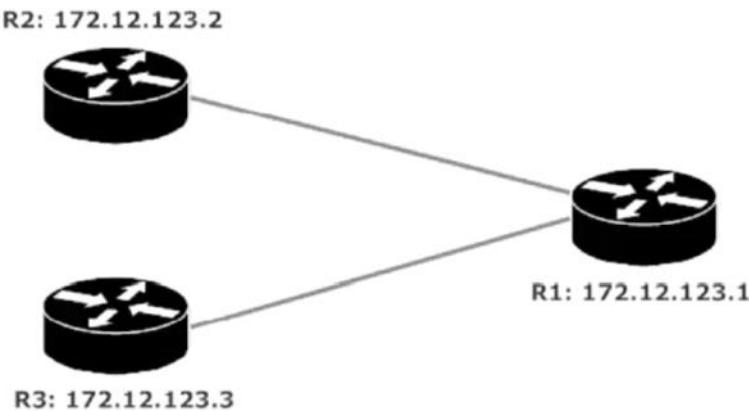
Show ip route 100.0.0.0 255.248.0.0

```
interface Serial1/0
  ip address 172.12.123.1 255.255.255.0
  encapsulation frame-relay
  ip summary-address eigrp 100 100.0.0.0 255.248.0.0 5
  frame-relay map ip 172.12.123.2 122 broadcast
  frame-relay map ip 172.12.123.3 123 broadcast
  no frame-relay inverse-arp
```

```
R1#show ip route 100.0.0.0 255.248.0.0
Routing entry for 100.0.0.0/13
  Known via "eigrp 100", distance 5, metric 128256, type internal
  Redistributing via eigrp 100
  Routing Descriptor Blocks:
    * directly connected, via Null0
      Route metric is 128256, traffic share count is 1
      Total delay is 5000 microseconds, minimum bandwidth is 10000000 Kbit
      Reliability 255/255, minimum MTU 1514 bytes
      Loading 1/255, Hops 0
```

EIGRP Stub Routing

Stub routing is a fantastic method of limiting the size of some routing tables in your EIGRP network while at the same time limiting the scope of DUAL queries. EIGRP doesn't have the stub area options that OSPF has, but EIGRP does allow a router to be configured as a stub. This is often done with a hub-and-spoke configuration like the one we've used in this course.



2/27/2018 2:20 PM - Screen Clipping

ADV EIGRP: Passive Interfaces

Monday, March 19, 2018 5:56 PM

We want rtr1 to have the /27 network in the routing table, but you only want to advertise, not send.

```
R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
H   Address           Interface      Hold Uptime    SRTT     RTO  Q  Se
1   172.12.123.2       Se1/0          (sec)  (ms)      Cnt Nu
R1#show ip route eigrp

R1#
BRYANT_ADV_1#2
[Resuming connection 2 to r2 ... ]

R2#
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router eigrp 100
R2(config-router)#network 172.12.23.0 0.0.0.31
```

DEBUG the EIGRP Debug eigrp packets

```
*Jun  9 22:15:57.220: EIGRP: Received HELLO on Serial0/1/0 nbr 172.12.123.1
*Jun  9 22:15:57.226:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0 peerQ un/rely 0/0
R2#
*Jun  9 22:15:58.266: EIGRP: Sending HELLO on FastEthernet0/0
*Jun  9 22:15:58.266:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0
R2#
*Jun  9 22:16:02.630: EIGRP: Sending HELLO on FastEthernet0/0
*Jun  9 22:16:02.630:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0
R2#
*Jun  9 22:16:07.238: EIGRP: Sending HELLO on FastEthernet0/0
*Jun  9 22:16:07.238:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
```

In order to not send hellos we have to make the interfaces passive, but you do it from the protocol and configure it.

```
R2(config-router)#passive-interface fast 0/0
R2(config-router)#^Z
R2#
*Jun  9 22:17:13.130: %SYS-5-CONFIG_I: Configured from console by console
R2#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address           Interface      Hold Uptime    SRTT     RTO  Q  Se
0   172.12.123.1       Se0/1/0        (sec)  (ms)      Cnt Nu
```

Now run the **Debug eigrp packets** command again and youll see the hello packets arent going out at all.

```
(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQU
IAREPLY)
EIGRP Packet debugging is on
R2#
*Jun 9 22:17:49.978: EIGRP: Sending HELLO on Serial0/1/0
*Jun 9 22:17:49.978:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0
R2#
*Jun 9 22:17:53.258: EIGRP: Received HELLO on Serial0/1/0 nbr 172.12.123.
*Jun 9 22:17:53.258:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0 peerQ un/rely 0/0
R2#u all
All possible debugging has been turned off
R2#
BRYANT_ADV_1#1
```

If you put **passive interface default** it means every interface on the router will be passive. You can set one as non-passive with the following.

```
Serial0/1/0 is down. Interface passive
R2(config-router)#no passive-interface serial 0/1/0
R2(config-router)#^Z
R2#
*Jun 9 22:20:35.726: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

ADV EIGRP: The Metric Weights

Monday, March 19, 2018 6:08 PM

The metric weights tell EIGRP how much importance to give a particular value when calculating route metrics

EIGRP uses the metric weights to determine how much importance to give a particular value when calculating route metrics. The k-values, in order:

- K1: bandwidth
- K2: load
- K3: delay
- K4: reliability
- K5: MTU

We are telling EIGRP to take bandwidth and delay into consideration. That's why the default is 10100 for EIGRP.

The **metric weights** command can change these values.

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router eigrp 100
R1(config-router)#metric ?
    holddown      Enable EIGRP holddown
    maximum-hops  Advertise EIGRP routes greater than <hops> as unreachable
    weights       Modify EIGRP metric coefficients

R1(config-router)#metric weights ?
    <0-8>  Type Of Service (Only TOS 0 supported)

R1(config-router)#metric weights 0 ?
    <0-255> K1
```

```
R1(config-router)#metric weights ?
    <0-8>  Type Of Service (Only TOS 0 supported)

R1(config-router)#metric weights 0 ?
    <0-255> K1

R1(config-router)#metric weights 0 2 ?
    <0-255> K2

R1(config-router)#metric weights 0 2 0 ?
    <0-255> K3

R1(config-router)#metric weights 0 2 0 1 ?
    <0-255> K4

R1(config-router)#metric weights 0 2 0 1 0 ?
    <0-255> K5

R1(config-router)#metric weights 0 2 0 1 0
```

You will have 6 values because TOS is 0. K value mismatches will begin to populate if the value of K1 is

changed.

```
| 9 18:07:37.570: %SYS-3-CONFIG_I: Configured from console by
| 9 18:07:41.809: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
| 9 18:07:41.817: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
|
| 9 18:07:46.609: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
| 9 18:07:46.617: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
|
| 9 18:07:51.609: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
| 9 18:07:51.617: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
|
| 9 18:07:56.208: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
| 9 18:07:56.216: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
|    all1/0) is down: K-value mismatch
|
| 9 18:08:00.908: %DU_
```

```
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router eigrp 100
R1(config-router)#metric weights 0 2 0 1 0 0
*Jun 9 18:08:33.409: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 172.12.
(Serial1/0) is down: K-value mismatch
*Jun 9 18:08:33.421: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 172.12.
(Serial1/0) is down: K-value mismatch
R1(config-router)#no metric weights 0 2 0 1 0 0
R1(config-router)#
```

To change them back simply apply "no".

Know this command for your exam.

ADV EIGRP: Know Thy Interface

Monday, March 19, 2018 6:14 PM

Tweaking The Bandwidth Value To Match Your Network Design

Here are some (very) general guidelines when it comes to designing networks that will be running EIGRP:

Invest the time necessary to set up a solid IP address allocation scheme before deploying your addresses. Measure twice, configure once!

Maximize your opportunities for manual route summarization.

Avoid discontiguous networks when possible, and keep EIGRP's autosummarization behavior in mind when you can't avoid them.

Your hub routers in any hub-and-spoke network will have the largest workload of all, so be sure those routers have the CPU and memory resources they need.

There are extra details to attend to when allocating bandwidth over an NBMA network. The typical Serial interface will have multiple Virtual Circuits (VCs), and the calculation for the *bandwidth* command varies according to the type of interface used on the hub and the CIR assigned to each circuit.

Some good questions to answer before you start configuring:

3/19/2018 6:24 PM - Screen Clipping

Your hub routers in any hub-and-spoke network will have the largest workload of all, so be sure those routers have the CPU and memory resources they need.

There are extra details to attend to when allocating bandwidth over an NBMA network. The typical Serial interface will have multiple Virtual Circuits (VCs), and the calculation for the *bandwidth* command varies according to the type of interface used on the hub and the CIR assigned to each circuit.

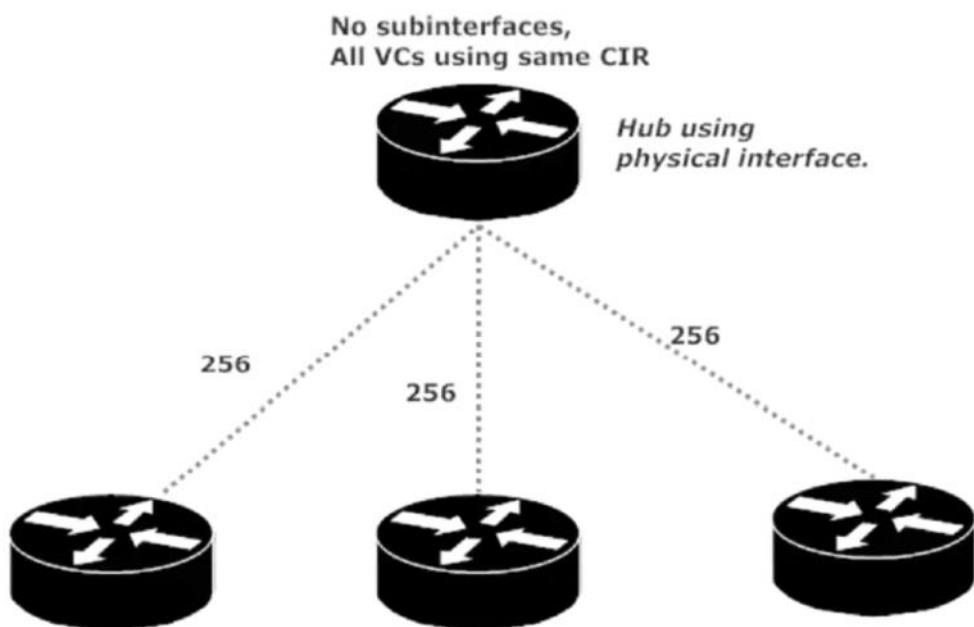
Some good questions to answer before you start configuring:

What IS the current CIR?

What are the current bandwidth values of our interfaces?

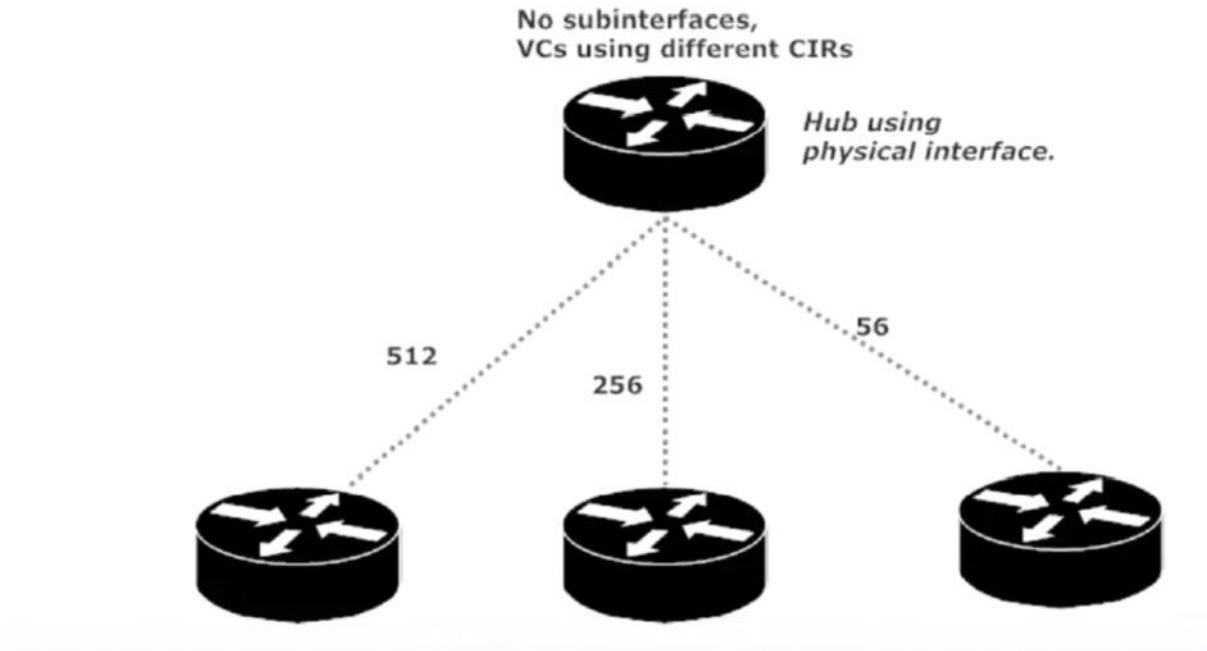
Does the client have any special requests or policies that need to be taken into consideration? Bandwidth maximum / minimums, stub routing, etc.?

We'll now take a look at several different scenarios and how they affect the bandwidth setting.



With no subinterfaces in use and the CIR the same for every VC, just add the CIRs and you have your bandwidth value. This ensures none of the circuits are overloaded, which could happen if we left bandwidth at the default of 1544.

```
R1 (config-if) #bandwidth 768
```



3/19/2018 6:18 PM - Screen Clipping

You can take the lowest CIR value and multiple it by the number of VCs to obtain optimal bandwidth. Or you can create point to point or multipoint sub interfaces and assign individual bandwidth values to each CIR.

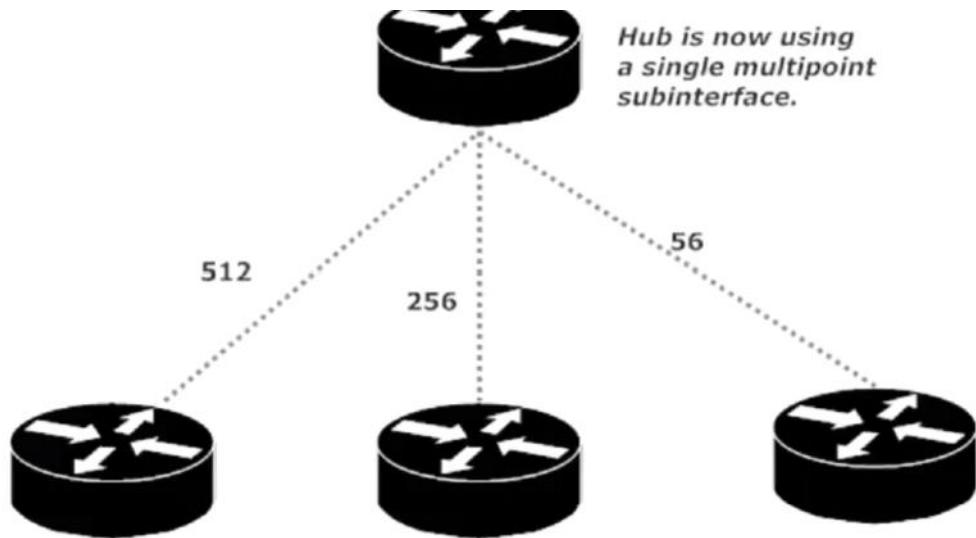
```
R1 (config)#int serial 0/1/0.1 point-to-point
R1 (config-subif)#ip address 20.1.1.1 255.255.255.0
R1 (config-subif)#bandwidth 512

R1 (config)#int serial 0/1/0.2 point-to-point
R1 (config-subif)#ip address 20.2.1.1 255.255.255.0
R1 (config-subif)#bandwidth 256

R1 (config)#int serial 0/1/0.3 point
R1 (config-subif)#ip address 20.3.1.1 255.255.255.0
R1 (config-subif)#bandwidth 56
```

3/19/2018 6:20 PM - Screen Clipping

What if you're using multipoint with a single subinterface on the hub?



Just add the CIR values to get the correct bandwidth value for the subinterface.

```
R1(config)#int serial 0/1/0.123 multipoint
R1(config-subif)#bandwidth 824
```

Eigrps default bandwidth usage is 50% of the interfaces bandwidth as set by the bandwidth command.
 You can change this.

```
R1(config)#int serial 0/1/0.123 multipoint
R1(config-subif)#bandwidth 824
R1(config-subif)#ip bandwidth-percent eigrp ?
<1-65535> AS number

R1(config-subif)#ip bandwidth-percent eigrp 100 ?
<1-999999> Maximum bandwidth percentage that EIGRP may use

R1(config-subif)#ip bandwidth-percent eigrp 100 300
```

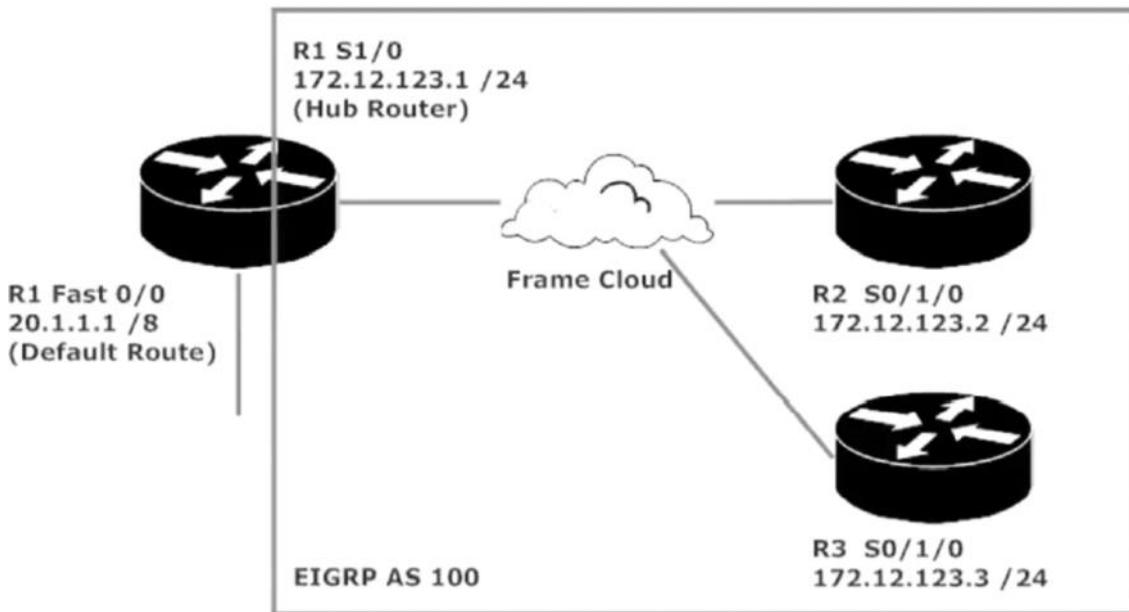
Note that the actual speed of the interface can exceed the logical setting. Just watch your syntax.

ADV EIGRP: Propagating Static Default Routes

Monday, March 19, 2018 6:24 PM

We can either inject a static route into EIGRP with route redistribution or we can indicate a default network with ip default-network.

For the first well create a default route on R1 and use its fast- interface as the exit interface. Well then redistribute that route into the EIGRP AS.



First create the default static route.

```
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#ip route 0.0.0.0 0.0.0.0 fast 0/0
```

Now redistribute it to EIGRP As 100

```
R1#  
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#ip route 0.0.0.0 0.0.0.0 fast 0/0  
R1(config)#router eigrp 100  
R1(config-router)#redistribute ?  
bgp Border Gateway Protocol (BGP)  
connected Connected  
eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)  
isis ISO IS-IS  
iso-igrp IGRP for OSI networks  
metric Metric for redistributed routes  
mobile Mobile routes  
odr On Demand stub Routes  
ospf Open Shortest Path First (OSPF)  
rip Routing Information Protocol (RIP)  
route-map Route map reference  
static Static routes
```

```

R1(config-router)#redistribute static metric 1544 10 ?
<0-255> EIGRP reliability metric where 255 is 100% reliable

R1(config-router)#redistribute static metric 1544 10 255 ?
<1-255> EIGRP Effective bandwidth metric (Loading) where 255 is

R1(config-router)#redistribute static metric 1544 10 255 1 ?
<1-65535> EIGRP MTU of the path

R1(config-router)#redistribute static metric 1544 10 255 1 1500
R1(config-router)#^Z
R1#

```

Use the question mark to determine values. Now the static route will be redistributed into EIGRP routing table.

```

R2#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS le
      ia - IS-IS inter area, * - candidate default, U - per-user stati
      o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LIS
      + - replicated route, % - next hop override

Gateway of last resort is 172.12.123.1 to network 0.0.0.0

D*EX 0.0.0.0/0 [170/2172416] via 172.12.123.1, 00:00:20, Serial0/1/0
R2#ping 20.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68, ms

```

And of course you can also remove said route by just using "no" before the command. Duh.

ADV EIGRP: The IP Default network Command

Monday, March 19, 2018 6:36 PM

First, I'll show you how the *ip default-network* command *should* work, and that's what you should go with on your exam. It works a little oddly in IOS 15, and while the CCNP Route exam is not IOS-specific, you should be aware of the quirkiness of this command with that particular IOS.

You can advertise a non-zero network number as the default route with *ip default-network*, but you have to watch out for a little something. Actually, it's a BIG something.

The router that originates this advertisement MUST have that network number in its IP routing table.

Why am I making such a big deal out of this? With OSPF, we have the option of advertising a default route even when the router didn't have a default route in its routing table ("*default-information originate always*"). We have no such option with EIGRP.

We start with two routes.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile,
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - O
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA e
        E1 - OSPF external type 1, E2 - OSPF external ty
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-
        ia - IS-IS inter area, * - candidate default, U
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    20.0.0.0/8 is directly connected, FastEthernet0/0
      172.12.0.0/24 is subnetted, 1 subnets
C          172.12.123.0 is directly connected, Serial1/0
R1#
```

```

R1#conf t
Enter configuration commands, one per line. End
R1(config)#router eigrp 100
R1(config-router)#network 20.0.0.0 0.255.255.255
R1(config-router)#exit
R1(config)#ip default-network ?
  A.B.C.D  IP address of default network

R1(config)#ip default-network 20.0.0.0 ?
<cr>

R1(config)#ip default-network 20.0.0.0
R1(config)#^Z
R1#wr
Building configuration...

BRYANT_ADV_1#2
[Resuming connection ? to r2] 1

```

Now look at router 2's routing table.

```

R2#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
      ia - IS-IS inter area, * - candidate default, U - per-user static ro
      o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LISP
      + - replicated route, % - next hop override

Gateway of last resort is 172.12.123.1 to network 20.0.0.0

D*    20.0.0.0/8 [90/2172416] via 172.12.123.1, 00:00:29, Serial0/1/0
R2#
```

Notice the *.

Gateway of last resort not set

ADV EIGRP: Neighbor Adjacency

Wednesday, March 21, 2018 5:59 PM

3/21/2018 6:05 PM - Screen Clipping

```
% Unrecognized command
R2(config)#key chain ?
  WORD [Key-chain] name

R2(config)#key chain CCNP
R2(config-keychain)#key ?
  <0-2147483647> Key identifier

R2(config-keychain)#key 1
R2(config-keychain-key)#?
Key-chain key configuration commands:
  accept-lifetime  Set accept lifetime of key
  default          Set a command to its defaults
  exit             Exit from key-chain key configuration mode
  key-string       Set key string
  no               Negate a command or set its defaults
  send-lifetime   Set send lifetime of key
```

This is where you're putting the password when applying it to an interface.

```
no           Negate a command or set its defaults
send-lifetime Set send lifetime of key
```

This is the EIGRP Authentication key and you can set the lifetime using this command.

```
R2(config-keychain-key)#accept-lifetime ?
  hh:mm:ss Time to start
  local      Specify time in local timezone
```

```
R2(config-keychain-key)#key-string CISCO
R2(config-keychain-key)#^Z
R2#
R2#
R2#
```

3/21/2018 6:03 PM - Screen Clipping

Now apply it to the interface.

```
R2(config)#int fast 0/0
R2(config-if)#ip authentication ?
  key-chain key-chain
  mode      mode

R2(config-if)#ip authentication mode ?
  eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)

R2(config-if)#ip authentication mode eigrp ?
  <1-65535> AS number

R2(config-if)#ip authentication mode eigrp 100 ?
  md5  Keyed message digest
```

```

R2(config)#int fast 0/0
R2(config-if)#ip authentication ?
  key-chain  key-chain
    mode      mode

R2(config-if)#ip authentication mode ?
  eigrp  Enhanced Interior Gateway Routing Protocol (EIGRP)

R2(config-if)#ip authentication mode eigrp ?
  <1-65535> AS number

R2(config-if)#ip authentication mode eigrp 100 ?
  md5  Keyed message digest

R2(config-if)#ip authentication mode eigrp 100 md5 ?
  <cr>

R2(config-if)#ip authentication mode eigrp 100 md5

```

3/21/2018 6:03 PM - Screen Clipping

Your adjacency will go down because the other router doesn't have an auth keychain.

```

*Nov 17 23:29:26.135: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.12.23
lastEthernet0/0) is down: authentication mode changed

```

3/21/2018 6:04 PM - Screen Clipping

```

lastEthernet0/0) is down: authentication mode changed
R2(config-if)#ip authentication ?
  key-chain  key-chain
    mode      mode

R2(config-if)#ip authentication key-chain ?
  eigrp  Enhanced Interior Gateway Routing Protocol (EIGRP)

R2(config-if)#ip authentication key-chain eigrp ?
  <1-65535> AS number

R2(config-if)#ip authentication key-chain eigrp 100 ?
  WORD  name of key-chain

R2(config-if)#ip authentication key-chain eigrp 100 CCNP
R2(config-if)#^Z
R2#
*Nov 17 23:30:45.063: %SYS-5-CONFIG_I: Configured from console by console
R2#debug eigrp packets
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUE
  [AREPLY])
EIGRP Packet debugging is on

```

```
*Nov 17 23:31:07.691: EIGRP: FastEthernet0/0: ignored packet from 172.12.23
pcode = 5 (missing authentication)
R2#
*Nov 17 23:31:08.823: EIGRP: Sending HELLO on FastEthernet0/0
*Nov 17 23:31:08.823:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 i
n/rely 0/0
R2#u all
All possible debugging has been turned off
R2#
*Nov 17 23:31:12.039: EIGRP: FastEthernet0/0: ignored packet from 172.12.23
pcode = 5 (missing authentication)
```

3/21/2018 6:06 PM - Screen Clipping

You then have to go to R3 and apply the same key chain, string, and authentication.

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#key chain CCNP
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CISCO
R3(config-keychain-key)#int fast 0/0
R3(config-if)#ip authentication mode eigrp 100 md5
R3(config-if)#ip authentication key-chain eigrp 100 CCNP
R3(config-if)#^Z
R3#
*Nov 17 23:17:55.143: %SYS-5-CONFIG_I: Configured from console by
R3#
*Nov 17 23:17:57.431: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor (FastEthernet0/0) is up: new adjacency
R3#
```

3/21/2018 6:06 PM - Screen Clipping

OSPF Fund 1: Link State Protocol Operation

Wednesday, March 21, 2018 6:04 PM

RIP sucks because it wastes bandwidth and CPU. Link stat routers form adjacencies and exchange LSU link state updates. **Link state protocols were the answer to replacing RIPS costly updates.**

```
R1#show ip ospf database

        OSPF Router with ID (1.1.1.1) (Process ID 1)

        Router Link States (Area 0)

Link ID      ADV Router      Age      Seq#      Checksum Link count
1.1.1.1      1.1.1.1        1790      0x80000004 0x00D5A8 1
2.2.2.2      2.2.2.2        1794      0x80000004 0x0097DD 1
3.3.3.3      3.3.3.3        1646      0x80000003 0x005B12 1

        Net Link States (Area 0)

Link ID      ADV Router      Age      Seq#      Checksum
172.12.123.1 1.1.1.1        1645      0x80000002 0x0027BC
```

The Dijkstra algorithm (also known as the Shortest Path First algorithm, or simply SPF) is run against the contents of the database to create the OSPF routing table, shown here with *show ip route ospf*.

3/21/2018 6:11 PM - Screen Clipping

```
R1#show ip route ospf
    2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA      2.2.2.2/32 [110/65] via 172.12.123.2, 00:00:13, Serial0/1/0
            3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA      3.3.3.3/32 [110/65] via 172.12.123.3, 00:00:02, Serial0/1/0
```

3/21/2018 6:11 PM - Screen Clipping

LSAs are assigned sequence number and when the router receives one, it checks its database first to see if its already there. If there's no entry the receiving router will flood that LSA out to every interface besides the one it came on. Sequence # = LSA is ignored.

Sequence # is lower=router ignores updates because of old info

Sequence # is higher = Adds to DB and sends back an LSA ack. It will flood and then update the routing table/LSDB

There will now, not be another update without a change and then one scheduled LSA summary every 30 minutes.

Before the LSA exchange begins, OSPF-speaking routers must become neighbors by forming an adjacency. Routers must agree on the area number, the hello and dead timer settings, and whether the area is a stub area. If link authentication is configured, it must be configured on both sides of the adjacency. The OSPF process number itself is locally significant only and does not have to be agreed upon for an adjacency to form. (We'll see all of these values along with the following show commands in action during our labs.)

3/21/2018 6:15 PM - Screen Clipping

You need the Area to match but not the Process ID. The Process ID is only locally significant.

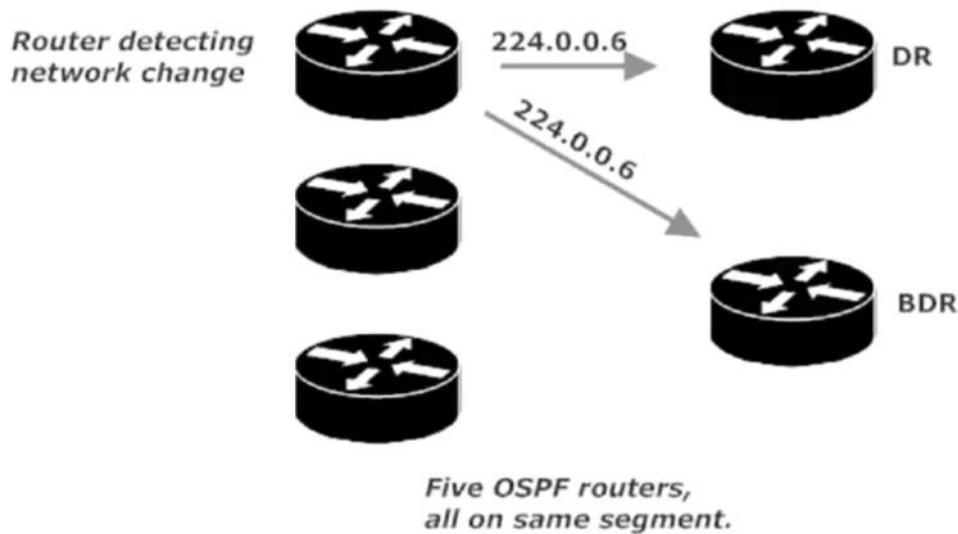
```
R1#show ip ospf int serial 0/1/0
Serial0/1/0 is up, line protocol is up
  Internet Address 172.12.123.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0            64          no          no           Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 172.12.123.1
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
    Hello due in 00:00:23
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 2, Adjacent neighbor count is 2
  Adjacent with neighbor 2.2.2.2
  Adjacent with neighbor 3.3.3.3
  Suppress hello for 0 neighbor(s)
```

3/21/2018 6:18 PM - Screen Clipping

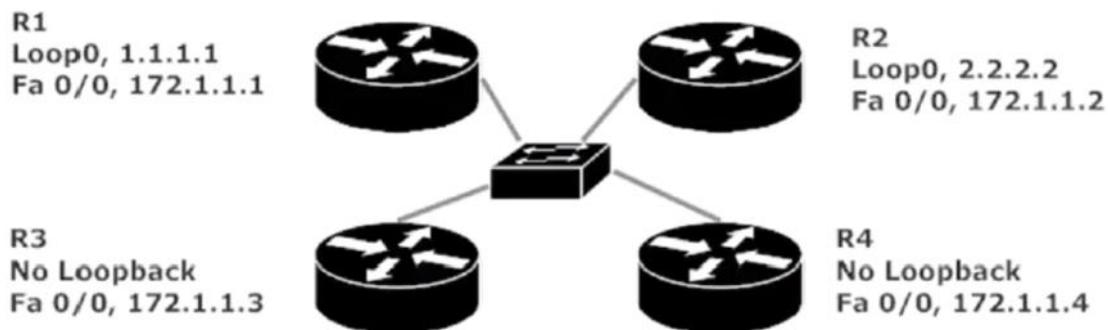
OSPF Fund 2: The DR and BDR

Wednesday, March 21, 2018 6:18 PM

Distance vector protocols have slow convergence. EIGRP uses successors/feasible successors. In order to speed that up OSPF uses designated and backup designated routers. **Routers use multicast address 224.0.0.6 for DR's address.** 224.0.0.5 is the All OSPF Routers Address. Using this address all routers can send requests out to this address if they're in the same multicast group. The BDR will also receive this update and become the failover device should an outage occur.



3/21/2018 6:23 PM - Screen Clipping



3/21/2018 6:25 PM - Screen Clipping

All routers with an OSPF interface priority of 1 or greater are eligible to participate for election. The default priority is 1. The router with the **highest priority** wins, a tie is broken by the **OSPF Router ID (RID)** which is the highest IP address of the interfaces or it will be the **highest loopback** on the router. The router **does not have to be ospf enabled**.

The RIDs:

Router 1: 1.1.1.1

Router 2: 2.2.2.2

Router 3: 172.1.1.3

Router 4: 172.1.1.4

R4 is the DR, R3 the BDR, and the other two routers are DROthers.

Summing up this section, there are three ways to manipulate the DR and BDR selection:

Changing the OSPF interface priority with *ip ospf priority*

Setting the OSPF RID manually with *router-id*

Setting the OSPF RID to an appropriate value with a loopback interface's IP address

3/21/2018 6:30 PM - Screen Clipping

OSPF Fund 3: Challenging a DR with new Router

Wednesday, March 21, 2018 6:32 PM

```
R1#
R1#show ip ospf neigh

Neighbor ID      Pri  State            Dead Time   Address
172.12.123.2     50   FULL/BDR        00:00:35    172.12.123.2  I
0
172.12.123.3     1    FULL/DROTHER   00:00:33    172.12.123.3  F
0
R1#
```

```
R1#show ip ospf int fast 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.12.123.1/24, Area 0
  Process ID 1, Router ID 172.12.123.1, Network Type BROADCAST, Cost
  Transmit Delay is 1 sec, State DR, Priority 100
  Designated Router (ID) 172.12.123.1, Interface address 172.12.123.1
  Backup Designated router (ID) 172.12.123.2, Interface address 172.12.123.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 172.12.123.2 (Backup Designated Router)
    Adjacent with neighbor 172.12.123.3
  Suppress hello for 0 neighbor(s)
```

3/21/2018 6:36 PM - Screen Clipping

Priority, RID, Highest IP, Highest Loopback.

OSPF Fund 4: Broadcast Segment

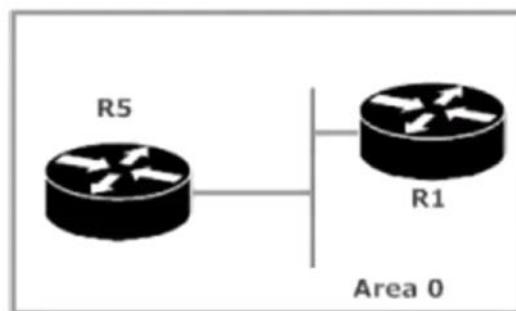
Wednesday, March 21, 2018 6:40 PM

We're now going to build an OSPF network from scratch, one segment at a time, starting with a broadcast segment between R1 and R5. We'll put this 10.1.1.0 /24 subnet into Area 0.

Each router has a single loopback that uses the router number for each octet.

Loopback interfaces will be advertised all at once later in this lab.

All previous OSPF-related commands have been removed from every router.



3/21/2018 6:41 PM - Screen Clipping

Start with router 5

```
R5(config)#router ospf 1
R5(config-router)#network 10.1.1.0 0.0.0.255
% Incomplete command.

R5(config-router)#network 10.1.1.0 0.0.0.255 ?
  area  Set the OSPF area ID

R5(config-router)#network 10.1.1.0 0.0.0.255 area ?
  <0-4294967295>  OSPF area ID as a decimal value
  A.B.C.D          OSPF area ID in IP address format

R5(config-router)#network 10.1.1.0 0.0.0.255 area 0
R5(config-router)#

```

Then 1.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 10.1.1.0 0.0.0.255 area 0
R1(config-router)#^Z
```

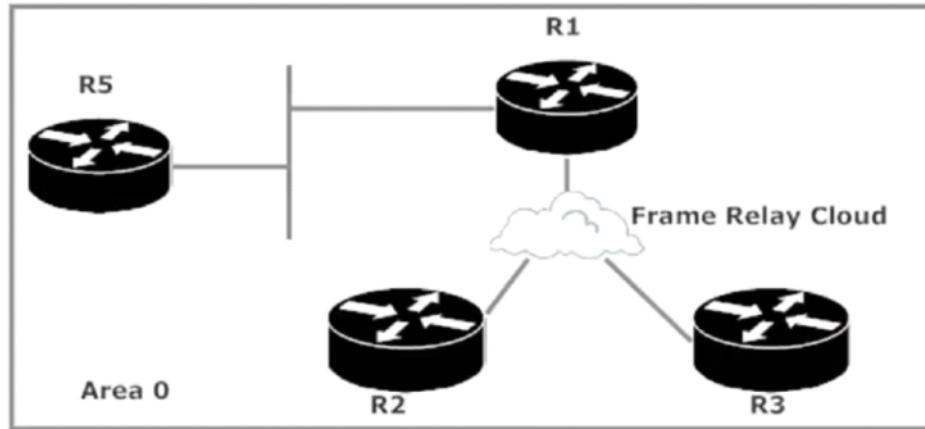
3/21/2018 6:45 PM - Screen Clipping

Verify

```
*Jun  9 17:01:24.629: %SYS-5-CONF
R1#show ip ospf neighbor
Neighbor ID      Pri  State
5.5.5.5          1    2WAY/DROTHE
0
R1#show ip ospf neighbor
Neighbor ID      Pri  State
5.5.5.5          1    2WAY/DROTHE
0
R1#
```

OSPF Fund 5: Building an NBMA Network

Wednesday, March 21, 2018 6:50 PM



R1 must be made the DR on this segment, and there should be no BDR. It's vital for both the DR and BDR to be able to get a multicast to all other routers on the segment. With a hub-and-spoke topology, a spoke router cannot get a multicast to the other spoke. All spoke-to-spoke traffic goes through the hub router, and routers do not forward broadcasts or multicasts.

Helpful lab hint: Be sure you have the *broadcast* option enabled on your frame map statements, or your multicasts ain't goin' anywhere!

```
R1#show frame map
```

3/21/2018 6:51 PM - Screen Clipping

```
R1#show frame map
Serial1/0 (up): ip 172.12.123.2 dlci 122(0x7A,0x1CA0), static,
                 broadcast,
                 CISCO, status defined, active
Serial1/0 (up): ip 172.12.123.3 dlci 123(0x7B,0x1CB0), static,
                 broadcast,
                 CISCO, status defined, active
```

3/21/2018 6:52 PM - Screen Clipping

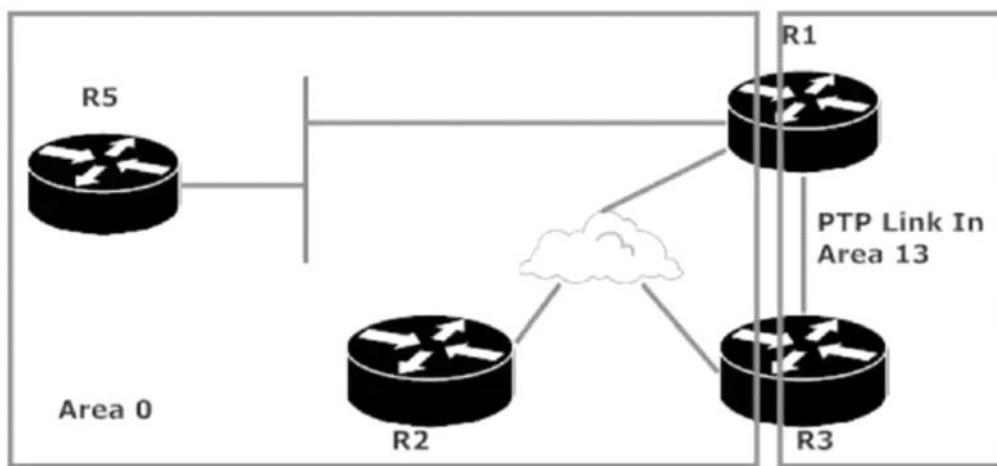
If you leave your broadcast off the frame map, it won't be on the multicast statement either, therefore no adjacencies.

OSPF Fund 6: Building a Point to Point Network

Thursday, March 22, 2018 4:35 PM

The OSPF Point-to-Point And Point-to-Multipoint Networks

We'll add a direct connection between R1 and R3, and put this one into Area 13. The network number is 172.12.13.0 /27. R1 is using its Serial 1/1 interface, R3 its Serial 1 interface.



None Zero areas must contain a router that has physical or logical interface in Area 0.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 172.12.13.0 0.0.0.31
% Incomplete command.

R3(config-router)#network 172.12.13.0 0.0.0.31 area 13
R3(config-router)#^Z
R3#
```

Add router 3

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 172.12.13.0 0.0.0.31 area 13
R1(config-router)#^Z
R1#
*Jun 15 05:42:04.611: %SYS-5-CONFIG_I: Configured from console
R1#
*Jun 15 05:42:12.985: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 is
LOADING to FULL, Loading Done
```

Add route 1

We're missing something though. DR/BDR isn't there.

```
R1#show ip ospf int serial 1/1
Serial1/1 is up, line protocol is up
  Internet Address 172.12.13.1/27, Area 13
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Index 1/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 3.3.3.3
  Suppress hello for 0 neighbor(s)
```

However this doesn't mean it's wrong since you don't need the overhead. Why? Because there's no need on a point to point.



No matter who announces
the change, there's only
one other router to tell!

```
R5#conf t
Enter configuration commands, one per line. End
R5(config)#router ospf 1
R5(config-router)#network 5.5.5.5 ?
  A.B.C.D  OSPF wild card bits

R5(config-router)#network 5.1.1.1 ?
  A.B.C.D  OSPF wild card bits

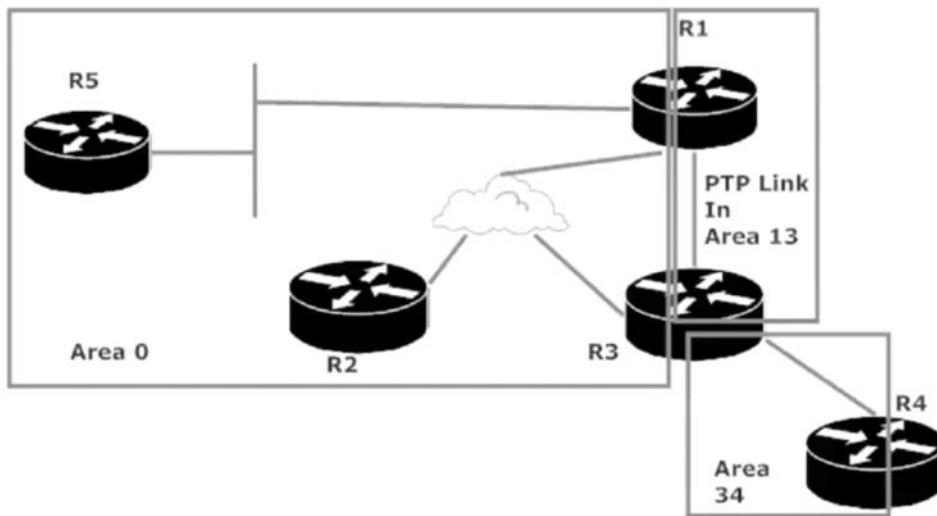
R5(config-router)#network 5.1.1.1 0.0.0.0 ?
  area  Set the OSPF area ID

R5(config-router)#network 5.1.1.1 0.0.0.0 area 5
R5(config-router)#+
```


OSPF Fund 7: The Missing Subnets and Virtual Links

4:46 PM

Let's add another broadcast segment to our network!



Every router should see each other's loopbacks. We don't see router 4 here though!

```
R5#show ip route ospf
Codes: L - local, C - connected, S - static, R
      D - EIGRP, EX - EIGRP external, O - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF
      E1 - OSPF external type 1, E2 - OSPF ex
      i - IS-IS, su - IS-IS summary, L1 - IS-I
      ia - IS-IS inter area, * - candidate de
      o - ODR, P - periodic downloaded static
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/2] via 10.1.1.1, 01:16:5
      2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/783] via 10.1.1.1, 01:17
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/783] via 10.1.1.1, 01:17
      172.12.0.0/16 is variably subnetted, 3 s
O IA    172.12.13.0/27 [110/782] via 10.1.1.1
O IA    172.12.34.0/24 [110/792] via 10.1.1.1
O       172.12.123.0/24 [110/782] via 10.1.1.1
R5#
```

Let's check router 1 to see if we're missing 4.4.4.4 on that as well. The biggest question in troubleshooting is **where is the problem?** So localize and identify what and where the issue is.

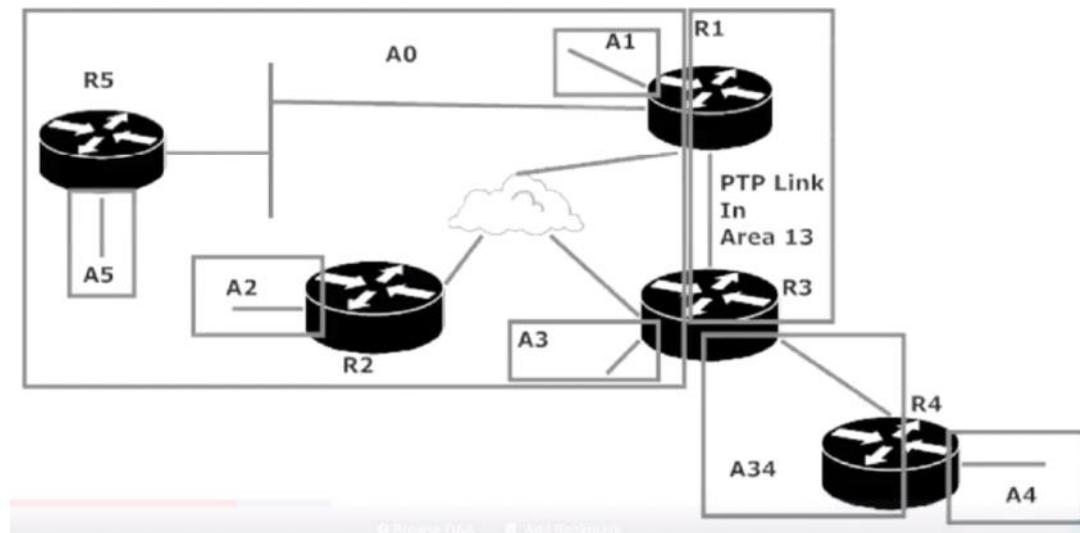
```
R1#show ip route ospf
  2.0.0.0/32 is subnetted
0 IA    2.2.2.2 [110/782] via
      3.0.0.0/32 is subnetted
0 IA    3.3.3.3 [110/782] via
      5.0.0.0/32 is subnetted
0 IA    5.5.5.5 [110/2] via
      172.12.0.0/16 is variab
0 IA    172.12.34.0/24 [110/
R1#
```

Router 4 loopback isn't on 1 either.

```
R3#show ip route ospf
  1.0.0.0/32 is subnette
0 IA    1.1.1.1 [110/65] vi
      2.0.0.0/32 is subnette
0 IA    2.2.2.2 [110/65] vi
      5.0.0.0/32 is subnette
0 IA    5.5.5.5 [110/66] vi
      10.0.0.0/24 is subnett
0       10.1.1.0 [110/65] v
R3#
```

Router 4 loopback isn't on Router 3.

The problem is in the design.



The #1 rule of design is that every area must contain an interface on a router that has a physical or logical interface in area 0. Area 4 is only connected to router 4, but router 4 doesn't have a physical interface in area 0.

VIRTUAL LINKS are used to connect a virtual link into Area 0. Transit area 34 is the space connecting Router 4 to area 0 interface. THIS IS NOT A STUB AREA.

So now we need to start on router 3 or 4.

```
R3#conf t
Enter configuration command
R3(config)#router ospf 1
R3(config-router)#area 34 virtual-link ?
  A.B.C.D  ID (IP addr) associated with virtual link neighbor
```

```
R3(config-router)#area 34 virtual-link 4.4.4.4
R3(config-router)#^Z
R3#wr
```

Now router 4, don't mind the logs. Finish the config.

```
R4(config-router)#area 34 virtual-link 3.3.3.3
R4(config-router)#^Z
```

```
R4(config-router)#
*Jan 1 02:24:06.243: %OSPF-4-ERRRCV: Received invalid packet: mismatch area
  from backbone area must be virtual-link but not found from 172.12.34.3, FastEthernet0/0
```

Now check the routing table. You should also start getting adjacency logs.

```
02:24:59.943: %SYS-5-CONFIG_I: Configured from console by console
02:25:06.307: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on OSPF_VL0 from L
  FULL, Loading Done
```

```
R4#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time
3.3.3.3	0	FULL/ -	00:00:06
3.3.3.3	1	FULL/BDR	00:00:39

```
R4#show ip ospf vir
Virtual Link OSPF_VL0 to router 3.3.3.3 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 34, via interface FastEthernet0/0
Topology-MTID      Cost      Disabled      Shutdown      Topology Name
          0          1          no          no          Base
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retrans
Hello due in 00:00:01
  Adjacency State FULL (Hello suppressed)
  Index 1/2, retransmission queue length 0, number of retransmi
First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
  Last retransmission scan length is 1, maximum is 1
  Last retransmission scan time is 0 msec, maximum is 0 msec
R4#
```

```
R3#show ip route ospf
  1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/65] via 172.12.123.1
  2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/65] via 172.12.123.2
  4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/11] via 172.12.34.4
  5.0.0.0/32 is subnetted, 1 subnets
O IA    5.5.5.5 [110/66] via 172.12.123.1
  10.0.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/65] via 172.12.123.1
R3#
```

Router 4's loopback is now in the routing table of router 3 and so on.

OSPF Fund 8: One Big Area 0

Thursday, March 22, 2018 5:00 PM

Why don't we do just one giant area 0? Areas allows us to create hierarchy or build layered networks to reduce router resource use like CPU and memory. Areas also help limit LSU and LSA traffic since notifications of changes in a multi area network can be limited to the area in which the change took place.

```
-----  
Area BACKBONE (0)  
Number of interfaces in this area is 1  
Area has no authentication  
SPF algorithm last executed 00:23:46.568 ago  
SPF algorithm executed 31 times  
Area ranges are  
Number of LSA 16. Checksum Sum 0x07326D  
Number of opaque link LSA 0. Checksum Sum 0x000000  
Number of DCbitless LSA 0  
Number of indication LSA 0  
Number of DoNotAge LSA 3  
Flood list length 0  
Area 5  
Number of interfaces in this area is 1 (1 loopback)  
Area has no authentication  
SPF algorithm last executed 00:51:14.332 ago  
SPF algorithm executed 2 times  
Area ranges are  
Number of LSA 9. Checksum Sum 0x0415A5  
Number of opaque link LSA 0. Checksum Sum 0x000000  
Number of DCbitless LSA 0  
Number of indication LSA 0  
Number of DoNotAge LSA 0  
Flood list length 0
```

3/22/2018 5:04 PM - Screen Clipping

Cisco offers these OSPF design guidelines to help you decide when it's time to add areas to your deployment:

No router should be in more than three areas.

No area should contain more than 50 routers.

No router should have more than 60 neighbors.

A router can be a DR or a BDR for more than one network segment, but monitor the workload on that switch carefully. Watch for an overtaxed CPU.

Do not run more than one OSPF process on an Area Border Router (ABR).

More on multi-area OSPF in the next section. (Yeah, there's more!) Right now, let's revisit an old friend and important OSPF value...

OSPF Fund 9: Interface Cost and Reference Bandwidth

Thursday, March 22, 2018 5:06 PM

```
R3#show ip ospf int e0
Ethernet0 is up, line protocol is up
  Internet Address 172.12.34.3/24, Area 34
  Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 172.12.34.4
  Backup Designated router (ID) 3.3.3.3, Interface address 172.12.34.3
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
    Hello due in 00:00:07
  Index 1/4, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 8
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 4.4.4.4 (Designated Router)
  Suppress hello for 0 neighbor(s)
```

```
R5#show ip ospf int gig0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.1.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                  1          no            no           Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 5.5.5.5, Interface address 10.1.1.5
  Backup Designated router (ID) 1.1.1.1, Interface address 10.1.1.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 2
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1 (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

To adjust the bandwidth.

```
R3(config-router)#auto-cost ref 1000
% OSPF: Reference bandwidth is changed.
  Please ensure reference bandwidth is consistent across all routers.
R3(config-router)#^Z
```

Please insure reference bandwidth is consistent across all routers. You don't want to have interfaces with different bandwidth limitations. You don't want gig interfaces to have a cost of 1 on router 1 then 10 on router 2. The value has to be consistent.

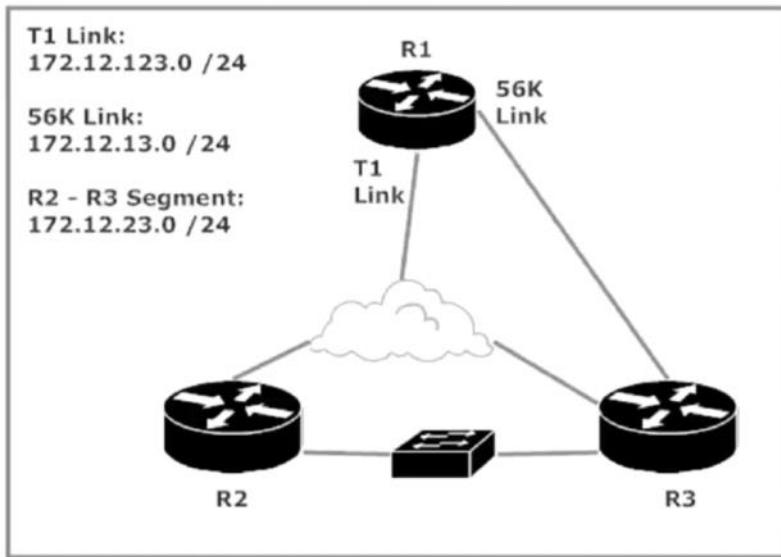
```
R3#show ip ospf int e0
Ethernet0 is up, line protocol is up
  Internet Address 172.12.34.3/24, Area 34
  Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 172.12.34.4
  Backup Designated router (ID) 3.3.3.3, Interface address 172.12.34.3
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Index 1/4, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 8
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 4.4.4.4 (Designated Router)
  Suppress hello for 0 neighbor(s)
```

This can be applied at the interface or global level.

OSPF Fund 10: Bandwidth Command and Interface Cost

Thursday, March 22, 2018 5:15 PM

We'll tweak bandwidth in the following lab to give OSPF a more accurate view of our network. Every segment is in Area 0.



According to R1's OSPF route table, there's some serious load balancing going on. OSPF will load balance over four paths by default, and we're already up to three.

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.12.123.3	0	FULL/ -	00:00:32	172.12.13.3	Serial1/1
172.12.123.2	0	FULL/DROTHER	00:01:46	172.12.123.2	Serial1/0
172.12.123.3	0	FULL/DROTHER	00:01:57	172.12.123.3	Serial1/0

How do we let the routers aware of the different bandwidths for each interface?

```
R1#conf t
Enter configuration commands, one per line. End with
R1(config)#int serial 1/1
R1(config-if)#bandwidth ?
  <1-10000000> Bandwidth in kilobits
  inherit      Specify that bandwidth is inherited
  receive       Specify receive-side bandwidth

R1(config-if)#bandwidth 56
R1(config-if)#^Z
```

As long as your adjacency remains the command worked.

```

show ip ospf int serial 1/1
  1/1 is up, line protocol is up
    Internet Address 172.12.13.1/24, Area 0
      Process ID 1, Router ID 172.12.123.1, Network Type POINT_TO_POINT, Cost: 1785
      Transmit Delay is 1 sec, State POINT_TO_POINT,
      Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Stub-resync timeout 40
      Hello due in 00:00:08
      Supports Link-local Signaling (LLS)
      Max 2/2, flood queue length 0
      MTU 0x0(0)/0x0(0)
      Flood scan length is 1, maximum is 1
      Flood scan time is 0 msec, maximum is 0 msec
      Neighbor Count is 1, Adjacent neighbor count is 1
      Adjacent with neighbor 172.12.123.3
      Address hello for 0 neighbor(s)

```

1785 is the value for a 56k link

This is how you can use the bandwidth command to let OSPF adjusted costs on interfaces.

```

R1#show ip route ospf
  172.12.0.0/24 is subnetted, 2 subnets
  O      172.12.23.0 [110/1786] via 172.12.13.3, 00:00:00, Serial1/1
R1#

```

The Bandwidth command is used by OSPF, EIGRP, QoS, and many other features.

The OSPF Adjacency States

It's hard to see every adjacency state during the actual forming of the adjacency! Here's a review of those states along with a description of what's going on during each state.

Down: No hellos received from that neighbor. Nothing much going on yet!

Attempt: Unicast hello packets are being sent to the neighbor. You'll only see this stage on the hub router in an NBMA network:

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
N/A	0	ATTEMPT/DROTHER	00:01:55	172.12.123.2	Serial1/0
N/A	0	ATTEMPT/DROTHER	00:01:55	172.12.123.3	Serial1/0

3/22/2018 5:23 PM - Screen Clipping

Init: Hey, we're getting somewhere! The first hello packet has been received from the neighbor.

2-Way: When you're here, you're almost gold. At this point, each router has received a Hello packet containing its own RID, indicating bidirectional communication. When a router receives a hello packet containing its own RID, it's not just talking to itself; that's the remote router's way of saying "I received that hello packet you sent me earlier."

Exstart: Following the DR/BDR election, the exchange of link state database info can begin! The router with the highest RID will begin the exchange and increment the initial sequence number, which is determined during this stage.

Exchange: Database Descriptor (DBD) packets are exchanged. As you'd expect, these packets contain a description of the link state database.

Loading: Routers now send Link State Request (LSR) packets to the almost-neighbor.

Full: Router databases are synced and the adjacency has been formed.

```
R1#show ip ospf neighbor

Neighbor ID      Pri   State          Dead Time     Address       Interface
172.12.123.2      0    FULL/DROTHER  00:01:44      172.12.123.2  Serial1/0
172.12.123.3      0    FULL/DROTHER  00:01:33      172.12.123.3  Serial1/0

R1#show ip ospf int serial 1/0
Serial1/0 is up, line protocol is up
  Internet Address 172.12.123.1/24, Area 0
  Process ID 1, Router ID 172.12.123.1, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.12.123.1, Interface address 172.12.123.1
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 172.12.123.2
    Adjacent with neighbor 172.12.123.3
  Suppress hello for 0 neighbor(s)
```

OSPF Fund 11: Lab Confirm and Changing the RID

Thursday, March 22, 2018 5:26 PM

You can change your RID. You can hardcode and override the rules.

```
R3(config)#router ospf 1
```

```
R3(config-router)#router-id 3.3.3.3 ?  
<cr>
```

```
R3(config-router)#router-id 3.3.3.3
```

```
Reload or use "clear ip ospf process" command, for this to take effect
```

To clear the process.

```
R3#clear ip ospf pro  
Reset ALL OSPF processes? [no]: y  
R3#  
01:31:59: %OSPF-5-ADJCHG: Process 1, Nbr 1  
WN, Neighbor Down: Interface down or detach  
01:31:59: %OSPF-5-ADJCHG: Process 1, Nbr 1  
WN, Neighbor Down: Interface down or detach  
01:31:59: %OSPF-5-ADJCHG: Process 1, Nbr 1  
FULL, Loading Done  
R3#  
BRYANT_ADV_1#1  
[Resuming connection 1 to r1 ... ]
```

```
R1#show ip ospf neighbor
```

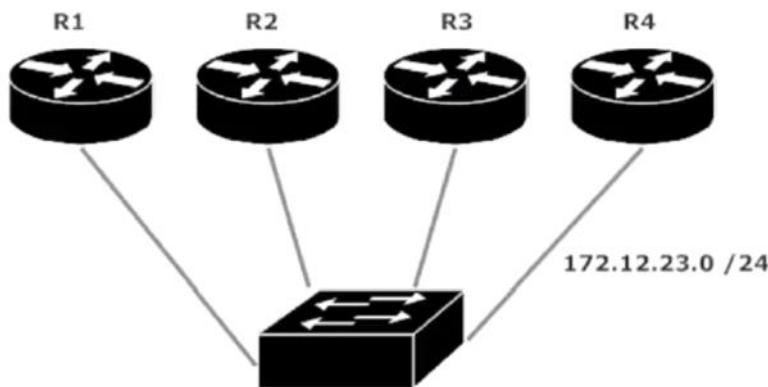
Neighbor ID	Pri	State	Dead Time	Address
3.3.3.3	0	FULL/-	00:00:37	172.1
172.12.123.2	0	2WAY/DROTHER	00:01:52	172.1
N/A	0	ATTEMPT/DROTHER	00:01:58	172.1

3/22/2018 5:30 PM - Screen Clipping

OSPF Fund 12: Troubleshooting Lab

Thursday, March 22, 2018 5:30 PM

It's important to know when to troubleshoot, and just as important to know when *not* to troubleshoot. Here's such a situation...



Let's have a look at our neighbor tables, shall we?

```
R4#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address
172.12.23.10	1	FULL/DROTHER	00:00:34	172.12.23.1
172.12.23.20	1	FULL/DROTHER	00:00:32	172.12.23.2
172.12.23.30	1	FULL/BDR	00:00:36	172.12.23.3

Router 4 is the DR. All routers are priority 1. Now check router 3.

```
R3#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address
172.12.23.10	1	FULL/DROTHER	00:00:33	172.12.23.1
172.12.23.20	1	FULL/DROTHER	00:00:35	172.12.23.2
172.12.23.40	1	FULL/DR	00:00:39	172.12.23.4

Now 2.

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time
172.12.23.10	1	2WAY/DROTHER	00:00:39
172.12.23.30	1	FULL/BDR	00:00:36
172.12.23.40	1	FULL/DR	00:00:34

Why is it at 2way with a DROTHER? Lets check router 1.

```
R1#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time
172.12.23.20	1	2WAY/DROTHER	00:00:33
172.12.23.30	1	FULL/BDR	00:00:38
172.12.23.40	1	FULL/DR	00:00:34

Router 1 is also in a 2way. The dead timers are working correctly. So what's the issue with the adjacency between router 1 and 2?

Lets go back to router 4 and add a loopback. Make sure to add it to OSPF as well.

```
R4#conf t
Enter configuration commands, one per line. End with Ctrl-Z
R4(config)#int loopback4
R4(config-if)#ip address 4.4.4.4 21
*Jan 1 01:52:44.367: %LINEPROTO-5-UPDOWN: Line protocol
  changed state to up
R4(config-if)#ip address 4.4.4.4 255.255.255.255
R4(config-if)#router ospf 1
R4(config-router)#network 4.4.4.4 0.0.0.0 area 0
R4(config-router)#^Z
R4#
```

```
R3#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos
!!!!!
```

```
R2#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos
!!!!!
Success rate is 100 percent (5/5)
```

```
R1#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos
!!!!!
Success rate is 100 percent (5/5)
```

The situation here is actually normal. We're not "Stuck in 2way". This is actually the correct behavior when you have 4 or more routers on an OSPF segment. DROTHERS will never have full adjacency with each other. No sooner the DR/BDR disappear then these adjacency's will form. Lets shut down router 4's interface to see this.

```
R4#conf t
Enter configuration commands
R4(config)#int fast 0/0
R4(config-if)#shut
R4(config-if)#^Z
```

Give it some time to die out.

```
R2#
*Jun 25 20:27:35.687: %OSPF-5-ADJCHG: Process 1, Nbr 172.12
0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
R2#show ip ospf neigh
```

```
R2#show ip ospf neigh
```

Neighbor ID	Pri	State
172.12.23.1	1	2WAY/DROTHER
0		
172.12.23.3	1	FULL/DR
0		

```
R2#
*Jun 25 20:27:41.127: %OSPF-5-ADJCHG: Process 1, Nbr 172.12.23.1 on FastEt
0/0 from LOADING to FULL, Loading Done
R2#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.12.23.1	1	FULL/DROTHER	00:00:35	172.12.23.1	FastEth0
0					
172.12.23.3	1	FULL/DR	00:00:35	172.12.23.3	FastEth0
0					

Now with router 4 gone, router 3 is the DR and Router 2 is the BDR.

OSPF Adv 1: Router and LSA Type Review

Friday, March 23, 2018 6:12 PM

We will use the same topology throughout these videos. Names may change.



An Area Border Router is a router with at least one interface in Area 0 and another in a non-backbone area. All ABRs are backbone routers, but not all backbone routers are ABRs.

An ASBR is a router injecting routes into the OSPF domain via route redistribution. To verify the ABR and ASBR status of the local router, run show ip ospf.

```
R1#show ip ospf
Routing Process "ospf 1" with ID 172.12.123.1
Start time: 02:01:03.214, Time elapsed: 00:55:10.023
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
It is an area border and autonomous system boundary router
Redistributing External Routes from,
connected, includes subnets in redistribution
```

Type 1s ("Router Link States") are generated by each router for every area the router has a link in. These are flooded to a single area only. The name is the recipe, as LSA Type 1s contain the "router link states" for this particular router.

Type 2:

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
172.12.123.1	1.1.1.1	34	0x80000001	0x0029BB

Type 2 LSAs are sent out only by DRs. The only Type 2 LSA in R3's OSPF database for Area 0 is from Advertising Router 1.1.1.1, the OSPF RID of R1.

LSA Types 1 and 2 are confined to a single area, which helps multi-area OSPF reduce the load on router resources. If you had only one large OSPF area, every router in the area would receive every single Type 1 and Type 2 LSA!

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
1.1.1.1	1.1.1.1	149	0x80000001	0x0047EC
2.2.2.2	2.2.2.2	132	0x80000001	0x00FA31
3.3.3.3	3.3.3.3	113	0x80000001	0x00AE75
33.3.3.3	3.3.3.3	113	0x80000001	0x0027DE

172.12.34.0 3.3.3.3 113 0x80000001 0x006CE8

These summary link advertisements are generated by ABRs and describe inter-area routes. They summarize the networks from one area to another, and are not flooded into a total stub area.

area routes. They summarize the networks from one area to another, and are not flooded into a total stub area.

Type 4:

Summary ASB Link States (Area 3)

Link ID	ADV Router	Age	Seq#	Checksum
1.1.1.1	3.3.3.3	28	0x80000001	0x007576

Type 4s are generated only by ABRs and describe the path to the ASBR. Type 4 LSAs are not flooded into a total stub area.

Type 5:

Type-5 AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum	Tag
11.1.1.0	1.1.1.1	153	0x80000001	0x009F70	0

Type 5 LSAs describe links external to the OSPF domain. This link describes the network injected into the OSPF domain via route redistribution on R1, verified by the address listed as the advertising router. Type 5 LSAs are generated only by ASBRs, and they're flooded to all areas except stub and total stub areas.

```

R4#show ip ospf database

    OSPF Router with ID (172.12.34.4) (Process ID 1)

        Router Link States (Area 34)

Link ID          ADV Router      Age       Seq#      Checksum Link count
3.3.3.3          3.3.3.3        24        0x8000000A 0x00FF1B 2
172.12.34.4     172.12.34.4   23        0x8000000A 0x00EDC2 1

        Net Link States (Area 34)

Link ID          ADV Router      Age       Seq#      Checksum
172.12.34.4     172.12.34.4   23        0x80000009 0x004A41

        Summary Net Link States (Area 34)

Link ID          ADV Router      Age       Seq#      Checksum
0.0.0.0          3.3.3.3        39        0x80000001 0x0057DA

```

This is a relatively small OSPF database, but it still allows R4 to reach every destination in our network. Configuring Area 34 as a total stub makes the OSPF database smaller as well as shrinking the OSPF route table. That's another way multi-area OSPF makes life easier on the router's memory and CPU!

3/23/2018 6:20 PM - Screen Clipping

A summary of which router types send which LSA types:

LSA Type 1: Sent by all routers.

LSA Type 2: Sent by DRs only.

LSA Type 3, 4: Sent by ABRs only.

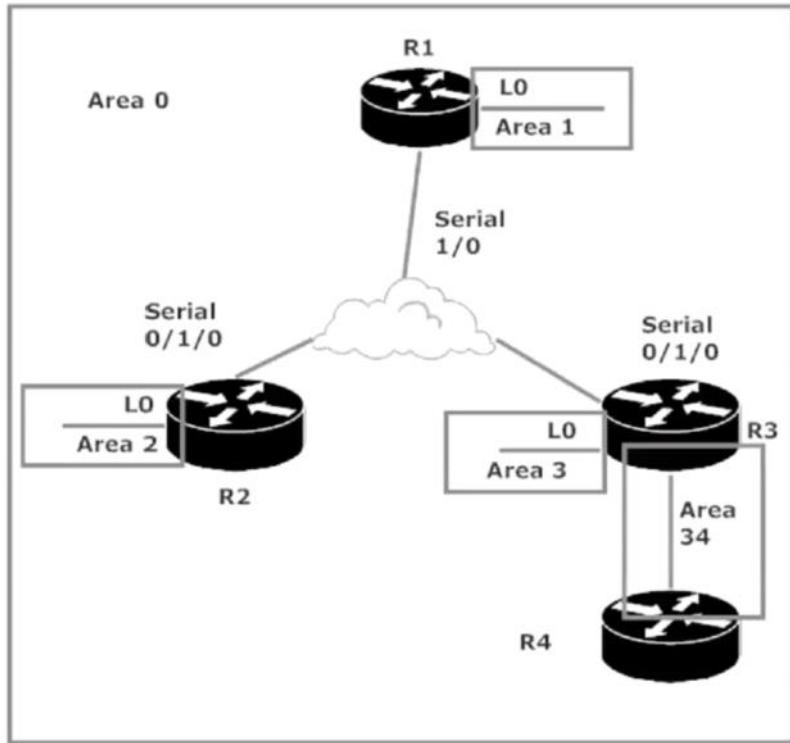
LSA Type 5, 7: Sent by ASBRs only.

LSA Type 6: Reserved for MOSPF.

3/23/2018 6:22 PM - Screen Clipping

OSPF Adv 2: Intro to Redistribution and Stub Areas

Friday, March 23, 2018 6:22 PM



3/23/2018 6:23 PM - Screen Clipping

```
R4#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
      ia - IS-IS inter area, * - candidate default, U - per-user static ro
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override
```

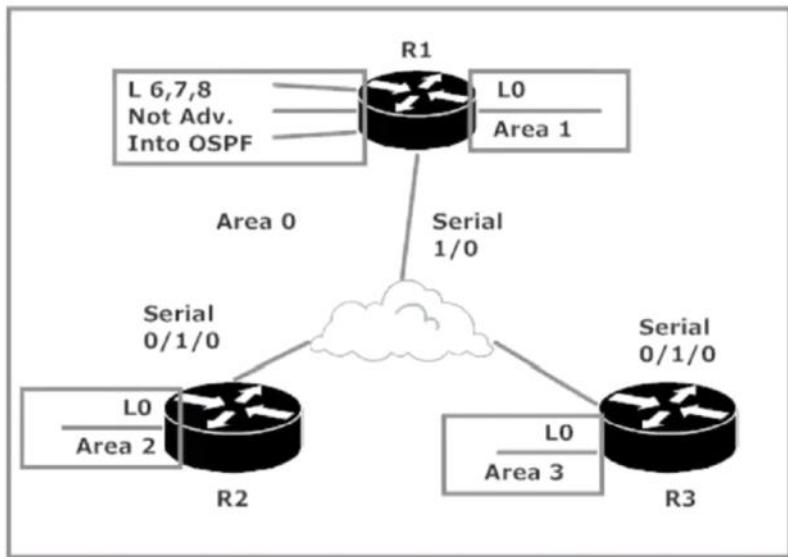
Gateway of last resort is not set

```
      1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3, 02:57:54, FastEthernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3, 02:57:54, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 02:57:54, FastEthernet0/0
      172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA    172.12.123.0/24 [110/65] via 172.12.34.3, 02:57:54, FastEthernet0/0
R4#
```

3/23/2018 6:25 PM - Screen Clipping

The next hop address is the only thing these routes have in common. 172.12.34.3

So now lets do route redistribution.



3/23/2018 6:26 PM - Screen Clipping

Default seed metric is 20

```

R4# 1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3, 02:57:54, FastEthernet0/0
O IA    2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3, 02:57:54, FastEthernet0/0
O IA    3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 02:57:54, FastEthernet0/0
          172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA    172.12.123.0/24 [110/65] via 172.12.34.3, 02:57:54, FastEthernet0/
R4#
BRYANT ADV 1#1

```

3/23/2018 6:26 PM - Screen Clipping

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#redistribute ?
bgp                  Border Gateway Protocol (BGP)
connected            Connected
eigrp                Enhanced Interior Gateway Routing Protocol (EIGRP)
isis                 ISO IS-IS
iso-igrp              IGRP for OSI networks
maximum-prefix       Maximum number of prefixes redistributed to protocol
metric               Metric for redistributed routes
metric-type          OSPF/IS-IS exterior metric type for redistributed routes
mobile               Mobile routes
odr                  On Demand stub Routes
ospf                 Open Shortest Path First (OSPF)
rip                  Routing Information Protocol (RIP)
route-map            Route map reference
static               Static routes
subnets              Consider subnets for redistribution into OSPF
tag                  Set tag for routes redistributed into OSPF
<cr>

```

```
R1(config-router)#redistribute connected ?
metric      Metric for redistributed routes
metric-type OSPF/IS-IS exterior metric type for redistributed routes
route-map   Route map reference
subnets     Consider subnets for redistribution into OSPF
tag         Set tag for routes redistributed into OSPF
<cr>

R1(config-router)#redistribute connected
% Only classful networks will be redistributed
R1(config-router)#
udem
```

```
R1(config-router)#redistribute connected ?
metric      Metric for redistributed routes
metric-type OSPF/IS-IS exterior metric type for redistributed routes
route-map   Route map reference
subnets     Consider subnets for redistribution into OSPF
tag         Set tag for routes redistributed into OSPF
<cr>

R1(config-router)#redistribute connected
% Only classful networks will be redistributed
R1(config-router)#no redistribute connected
R1(config-router)#redistribute connected subnets
R1(config-router)#^Z
R1#wr
```

```
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3, 03:03:01, FastEthernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3, 03:03:01, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 03:03:01, FastEthernet0/0
      6.0.0.0/32 is subnetted, 1 subnets
O E2    6.6.6.6 [110/20] via 172.12.34.3, 00:02:06, FastEthernet0/0
      7.0.0.0/32 is subnetted, 1 subnets
O E2    7.7.7.7 [110/20] via 172.12.34.3, 00:02:06, FastEthernet0/0
      8.0.0.0/32 is subnetted, 1 subnets
O E2    8.8.8.8 [110/20] via 172.12.34.3, 00:02:06, FastEthernet0/0
      172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA    172.12.123.0/24 [110/65] via 172.12.34.3, 03:03:01, FastEth
```

E2 is the default routing code. This will be covered later. This measures the metric from the ASBR to the remote Destination. Doesn't include local distance. If the seed metric is 20, then we know the path from router 4 and 1 is not included.

OSPF Adv 3: Configuring Stub and Total Stub Areas

Friday, March 23, 2018 6:34 PM

The Stub Flag or bit is when two routers agree on if the area is a stub.

```
R3(config)#router ospf 1
R3(config-router)#area 34 ?
  authentication  Enable authentication
  capability      Enable area specific capability
  default-cost    Set the summary default-cost of a NSSA/stub area
  filter-list     Filter networks between OSPF areas
  nssa            Specify a NSSA area
  range           Summarize routes matching address/mask (border routers o
  sham-link       Define a sham link and its parameters
  stub            Specify a stub area
  virtual-link   Define a virtual link and its parameters

R3(config-router)#area 34 stub ?
  no-ext-capability  Do not send domain specific capabilities into stub ar
  no-summary        Do not send summary LSA into stub area
<cr>

R3(config-router)#area 34 stub
R3(config-router)#^Z
R3#
*Jun 26 17:25:49.394: %OSPF-5-ADJCHG: Process 1, Nbr 172.12.34.4 on FastEt
0/0 from FULL to DOWN, Neighbor Down: Adjacency forced to reset
R3#
*Jun 26 17:25:50.538: %SYS-5-CONFIG_I: Configured from console by console
^Z
```

3/23/2018 6:36 PM - Screen Clipping

We have to set the adjacency for router 4.

```
R4#
R4#conf t
Enter configuration commands, one per line.
R4(config)#router ospf 1
R4(config-router)#area 34 stub
R4(config-router)#^Z
R4#
*Jan  1 05:09:44.298: %OSPF-5-ADJCHG: Proces
from FULL to DOWN, Neighbor Down: Adjacency
*Jan  1 05:09:44.978: %SYS-5-CONFIG_I: Confi
R4#
```

3/23/2018 6:36 PM - Screen Clipping

Now look at the routing table. We don't have the E2s anymore. Instead we have an *IA

```
O*IA 0.0.0.0/0 [110/2] via 172.12.34.3, 00:00:08, FastEthernet0/0
  1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3, 00:00:08, FastEthernet0/0
O IA    2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3, 00:00:08, FastEthernet0/0
O IA    3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 00:00:08, FastEthernet0/0
O IA  172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA    172.12.123.0/24 [110/65] via 172.12.34.3, 00:00:08, FastEthernet0/0
R4#
```

3/23/2018 6:37 PM - Screen Clipping

```
R4#ping 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#ping 7.7.7.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.7, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#ping 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#
```

3/23/2018 6:37 PM - Screen Clipping

```
R4#ping 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#ping 7.7.7.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.7, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#ping 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, t
!!!!!
Success rate is 100 percent (5/5), round-trip time = 0ms
R4#
```

3/23/2018 6:38 PM - Screen Clipping

OSPF Adv 4: Configuring NSSAs

Monday, March 26, 2018 8:32 PM

A quick review of stub vs. total stub areas:

With an OSPF stub area, your OSPF routing table can contain routes to networks in the same area (O), inter-area routes (O IA), and a default inter-area route to reach external destinations (O *IA).

A total stub area's OSPF routing table can contain only routes to other networks in the total stub area (O) and a single default route for all other routes (O *IA). If we add a network to Area 34 on R3 and advertise it via the network command, R4 will see it as an inter-area route, and as such it will have a specific entry in the OSPF table.

```
R3(config-if)#router ospf 1
R3(config-router)#network 33.3.3.3 0.0.0.0 area 34

R4#show ip route ospf

Gateway of last resort is 172.12.34.3 to network 0.0.0.0

O*IA  0.0.0.0/0 [110/21] via 172.12.34.3, 00:07:38, FastEthernet0/0
      33.0.0.0/32 is subnetted, 1 subnets
O        33.3.3.3 [110/2] via 172.12.34.3, 00:00:06, FastEthernet0/0
```

Backbone cant become stub because of area 0

```
R2(config-router)#
R2(config-router)#
R2(config-router)#exit
R2(config)#router ospf 1
R2(config-router)#area 0 stub
% OSPF: Backbone can not be configured as stub area
R2(config-router)#

```

Not so Stubby Stub Area is a stub area that contains a limited number of external routes. Only type that uses type 7 LSAs.

Lets add a loopback to r3

```

R3#
R3#conf t
Enter configuration commands, one per line. End
R3(config)#int loopback14
R3(config-if)#
*Jun 26 18:38:39.618: %LINK-3-UPDOWN: Interface
*Jun 26 18:38:40.618: %LINEPROTO-5-UPDOWN: Line
, changed state to up
R3(config-if)#ip address 14.1.1.1 255.255.255.0
R3(config-if)#router ospf 1

```

3/26/2018 9:16 PM - Screen Clipping

Now do a redistribute command.

```

R3(config-if)#router ospf 1
R3(config-router)#redis conn subnets
R3(config-router)#^Z
R3#wr
Building configuration...

*Jun 26 18:39:08.962: %SYS-5-CONFIG_I: Configured from console by c
BRYANT_ADV_1#1
[Resuming connection 1 to r1 ... ]

R1#
R1#show ip route ospf
  2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/782] via 172.12.123.2, 00:00:00, Serial1/0
      33.0.0.0/32 is subnetted, 1 subnets
O IA    33.3.3.3 [110/782] via 172.12.123.3, 00:00:00, Serial1/0
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/782] via 172.12.123.3, 00:00:00, Serial1/0
      172.12.0.0/24 is subnetted, 2 subnets
O IA    172.12.34.0 [110/782] via 172.12.123.3, 00:00:00, Serial1/0
      14.0.0.0/24 is subnetted, 1 subnets
O E2    14.1.1.0 [110/20] via 172.12.123.3, 00:00:00, Serial1/0
R1#

```

3/26/2018 9:17 PM - Screen Clipping

You still see the one individual route, but you don't see the router for 14.1.1.0. In order to make this an NSSA we have to take the total stub off.

```

R4(config)#router ospf 1
R4(config-router)#no area 34 stub
R4(config-router)#^Z
R4#wr
Building configuration...

*Jan
BRYANT_ADV_1#3
[Resuming connection 3 to r3 ... ]
[OK
R3#
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#no area 34 stub no-summary
R3(config-router)#no area 34 stub
R3(config-router)#^Z
R3#
*Jun 26 18:40:34.478: %OSPF-5-ADJCHG: Process 1, Nbr 172.12.34
0/0 from FULL to DOWN, Neighbor Down: Adjacency forced to rese
*Jun 26 18:40:34.842: %OSPF-5-ADJCHG: Process 1, Nbr 172.12.34
0/0 from LOADING to FULL, Loading Done
*Jun 26 18:40:35.314: %SYS-5-CONFIG_I: Configured from console
R3#_

```

3/26/2018 9:18 PM - Screen Clipping

The neighbor adjacency between router 3 and 4 is now down.

Now lets look at the routes, which are no win a regular OSPF area.

Gateway of last resort is not set	1.0.0.0/32 is subnetted, 1 subnets 0 IA 1.1.1.1 [110/66] via 172.12.34.3, 00:00:44, FastEt 2.0.0.0/32 is subnetted, 1 subnets 0 IA 2.2.2.2 [110/66] via 172.12.34.3, 00:00:44, FastEt 3.0.0.0/32 is subnetted, 1 subnets 0 IA 3.3.3.3 [110/2] via 172.12.34.3, 00:00:44, FastEth 6.0.0.0/32 is subnetted, 1 subnets 0 E2 6.6.6.6 [110/20] via 172.12.34.3, 00:00:44, FastEt 7.0.0.0/32 is subnetted, 1 subnets 0 E2 7.7.7.7 [110/20] via 172.12.34.3, 00:00:44, FastEt 8.0.0.0/32 is subnetted, 1 subnets 0 E2 8.8.8.8 [110/20] via 172.12.34.3, 00:00:44, FastEt 14.0.0.0/24 is subnetted, 1 subnets 0 E2 14.1.1.0 [110/20] via 172.12.34.3, 00:00:44, FastE 33.0.0.0/32 is subnetted, 1 subnets 0 33.3.3.3 [110/2] via 172.12.34.3, 00:00:44, FastEt 172.12.0.0/16 is variably subnetted, 3 subnets, 2 mas 0 IA 172.12.123.0/24 [110/65] via 172.12.34.3, 00:00:44
-----------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3/26/2018 9:19 PM - Screen Clipping
So lets specify the NSSA.

```
R4(config)#router ospf 1
R4(config-router)#area 34 ?
  authentication  Enable authentication
  capability      Enable area specific capability
  default-cost    Set the summary default-cost of a
  filter-list     Filter networks between OSPF areas
  nssa            Specify a NSSA area
  range           Summarize routes matching address/
  sham-link       Define a sham link and its parameter
  stub            Specify a stub area
  virtual-link   Define a virtual link and its parameters

R4(config-router)#area 34 nssa ?
  default-information originate  Originate Type 7 default route
  no-ext-capability          Do not send domain border routers
  no-redistribution           No redistribution from external routes
  no-summary                  Do not send summary LSA
  translate                   Translate LSA
<cr>

R4(config-router)#area 34 nssa
```

And the same on router 3

```
3#
3#conf t
[config]#router ospf 1
3(config-router)#area 34 nssa
3(config-router)#^Z
3#
Jun 26 18:42:15.326: %OSPF-5-ADJCHG: R3/0 from FULL to DOWN, Neighbor Down: Area 0
Jun 26 18:42:16.310: %SYS-5-CONFIG_I:
3#
Jun 26 18:42:19.482: %OSPF-5-ADJCHG: R3/0 from LOADING to FULL, Loading Done
```

3/26/2018 9:20 PM - Screen Clipping

Now lets look at the routing table on 4

```

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O IA      1.1.1.1 [110/66] via 172.12.34.3, 00:00:07, FastEt
      2.0.0.0/32 is subnetted, 1 subnets
O IA      2.2.2.2 [110/66] via 172.12.34.3, 00:00:07, FastEt
      3.0.0.0/32 is subnetted, 1 subnets
O IA      3.3.3.3 [110/2] via 172.12.34.3, 00:00:07, FastEth
      14.0.0.0/24 is subnetted, 1 subnets
O N2      14.1.1.0 [110/20] via 172.12.34.3, 00:00:07, FastE
      33.0.0.0/32 is subnetted, 1 subnets
O          33.3.3.3 [110/2] via 172.12.34.3, 00:00:07, FastEt
      172.12.0.0/16 is variably subnetted, 3 subnets, 2 mas
O IA      172.12.123.0/24 [110/65] via 172.12.34.3, 00:00:07
R4#

```

3/26/2018 9:20 PM - Screen Clipping

The N2 is the NSSA area. The 6 7 and 8 networks are no longer there and neither is the default route.
You don't have a default route by default. You have to make it a NSSTA. Not so stub tototal stub area

```

R3#conf t
Enter configuration commands, one per line. End with a
R3(config)#router ospf 1
R3(config-router)#area 34 nssa ?
  default-information-originate   Originate Type 7 de
  no-ext-capability              Do not send domain
                                    NSSA
  no-redistribution               No redistribution i
  no-summary                      Do not send summary
  translate                        Translate LSA
<cr>

R3(config-router)#area 34 nssa no-summary
R3(config-router)#

```

3/26/2018 9:21 PM - Screen Clipping

Now youll see the default route.

```

      * - ODR, + - per route downloaded static route, # - NHRP ,
      + - replicated route, % - next hop override

Gateway of last resort is 172.12.34.3 to network 0.0.0.0

O*IA  0.0.0.0/0 [110/2] via 172.12.34.3, 00:00:08, FastEthernet0/0
      14.0.0.0/24 is subnetted, 1 subnets
O N2   14.1.1.0 [110/20] via 172.12.34.3, 00:01:34, FastEthernet0/0
      33.0.0.0/32 is subnetted, 1 subnets
O      33.3.3.3 [110/2] via 172.12.34.3, 00:01:34, FastEthernet0/0
R4#

```

3/26/2018 9:22 PM - Screen Clipping

You can now ping since your default route is back.

```
Type escape sequence to a
Sending 5, 100-byte ICMP
!!!!!
Success rate is 100 perce
R4#ping 7.7.7.7

Type escape sequence to a
Sending 5, 100-byte ICMP
!!!!!
Success rate is 100 perce
R4#ping 8.8.8.8

Type escape sequence to a
Sending 5, 100-byte ICMP
!!!!!
Success rate is 100 perce
R4#
```

3/26/2018 9:22 PM - Screen Clipping

OSPF Adv 5: E1, E3, N1, N2

Monday, March 26, 2018 9:22 PM

We're using the same topology. We have taken off the stub areas and have a regular OSPF areas.

```
R4#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level
      ia - IS-IS inter area, * - candidate default, U - per-user static r
      o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LIS
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
      1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3, 00:00:02, FastEthernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3, 00:00:02, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 00:00:02, FastEthernet0/0
      6.0.0.0/32 is subnetted, 1 subnets
O E2    6.6.6.6 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
      7.0.0.0/32 is subnetted, 1 subnets
O E2    7.7.7.7 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
      8.0.0.0/32 is subnetted, 1 subnets
O E2    8.8.8.8 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
```

3/26/2018 9:23 PM - Screen Clipping

The difference in e1 and e2 is in the metric.

E1 route reflects the cost of the entire path from the local router to the destination network.

E2 includes only the metric from path to ASBR destination.

Gateway of last resort is no

```
      1.0.0.0/32 is subnetted
O IA    1.1.1.1 [110/66] via 172.12.34.3, 00:00:02, FastEthernet0/0
      2.0.0.0/32 is subnetted
O IA    2.2.2.2 [110/66] via 172.12.34.3, 00:00:02, FastEthernet0/0
      3.0.0.0/32 is subnetted
O IA    3.3.3.3 [110/2] via 172.12.34.3, 00:00:02, FastEthernet0/0
      6.0.0.0/32 is subnetted
O E2    6.6.6.6 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
      7.0.0.0/32 is subnetted
O E2    7.7.7.7 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
      8.0.0.0/32 is subnetted
O E2    8.8.8.8 [110/20] via 172.12.34.3, 00:00:02, FastEthernet0/0
      14.0.0.0/24 is subnetted
```

3/26/2018 9:25 PM - Screen Clipping

Look at the metric for 6, 7, 8. The metric is 20. E2 will default to a seed metric of 20. Since they're directly connected to router.

```
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O IA      1.1.1.1 [110/66] via 172.12.34.3
      2.0.0.0/32 is subnetted, 1 subnets
O IA      2.2.2.2 [110/66] via 172.12.34.3
      3.0.0.0/32 is subnetted, 1 subnets
O IA      3.3.3.3 [110/2] via 172.12.34.3,
      6.0.0.0/32 is subnetted, 1 subnets
O E2      6.6.6.6 [110/20] via 172.12.34.3
      7.0.0.0/32 is subnetted, 1 subnets
O E2      7.7.7.7 [110/20] via 172.12.34.3
      8.0.0.0/32 is subnetted, 1 subnets
O E2      8.8.8.8 [110/20] via 172.12.34.3
      14.0.0.0/24 is subnetted, 1 subnets
O E2     14.1.1.0 [110/20] via 172.12.34.3
      33.0.0.0/32 is subnetted, 1 subnets
O       33.3.3.3 [110/2] via 172.12.34.3
      172.12.0.0/16 is variably subnetted
O IA     172.12.123.0/24 [110/65] via 172
```

3/26/2018 9:28 PM - Screen Clipping

How do we choose E1 over E2? Adjust the metric.

```
R1#conf t
Enter configuration commands, one per line. End with Ctrl-Z.
R1(config)#router ospf 1
R1(config-router)#no redis conn subnets
R1(config-router)#no redis conn
R1(config-router)#redis conn subnets ?
  metric          Metric for redistributed routes
  metric-type    OSPF/IS-IS exterior metric type for
  route-map      Route map reference
  tag            Set tag for routes redistributed into OSPF
<cr>
R1(config-router)#redis conn subnets metric-type ?
  1  Set OSPF External Type 1 metrics
  2  Set OSPF External Type 2 metrics
```

3/26/2018 9:29 PM - Screen Clipping

```
R1(config-router)#redis conn subnets metric-type 1
R1(config-router)#^Z
R1#
```

3/26/2018 9:29 PM - Screen Clipping

```
      1.0.0.0/32 is subnetted, 1 subnets
O IA    1.1.1.1 [110/66] via 172.12.34.3,
      2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/66] via 172.12.34.3,
      3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/2] via 172.12.34.3, 0
      6.0.0.0/32 is subnetted, 1 subnets
O E1    6.6.6.6 [110/85] via 172.12.34.3,
      7.0.0.0/32 is subnetted, 1 subnets
O E1    7.7.7.7 [110/85] via 172.12.34.3,
      8.0.0.0/32 is subnetted, 1 subnets
O E1    8.8.8.8 [110/85] via 172.12.34.3,
      14.0.0.0/24 is subnetted, 1 subnets
O E2    14.1.1.0 [110/20] via 172.12.34.3,
      33.0.0.0/32 is subnetted, 1 subnets
O       33.3.3.3 [110/2] via 172.12.34.3,
      172.12.0.0/16 is variably subnetted,
O IA    172.12.123.0/24 [110/65] via 172.1
R4#
```

3/26/2018 9:30 PM - Screen Clipping

You can now see the routes are E1.

N1 and N2 routes are only found in NSSAs. N1 and N2 differences are the same as e1 and e2.

OSPF Adv 6: More Route Redistribution

Monday, March 26, 2018 9:31 PM

OSPF Route Summarization Techniques

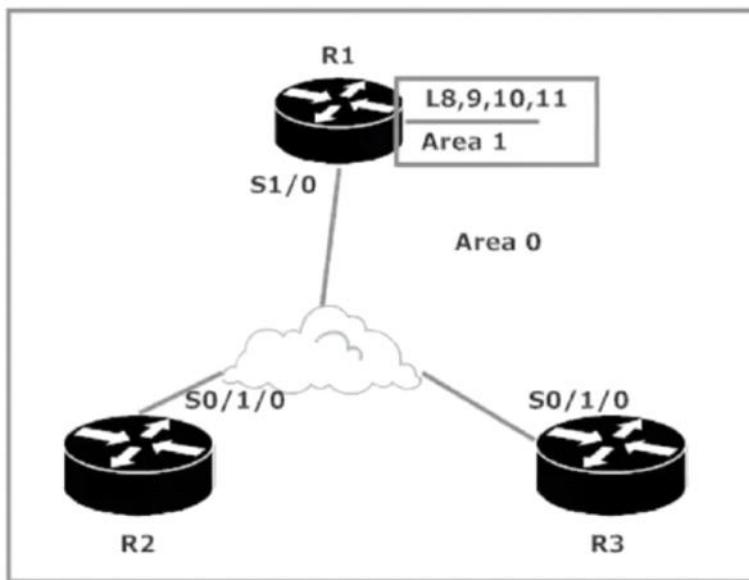
We always want our routing tables complete and concise. OSPF stub and total stub areas help us accomplish that goal by replacing external and inter-area routes with default routes, but we know it's not always possible to configure stub and total stub areas. Area 0 can't be a stub or total stub, and neither can an area serving as a transit area for a virtual link.

We can further shrink our routing table via route summarization. There are two ways to perform this summarization in OSPF, and the method you choose is dependent on the OSPF router types in use. (OSPF performs no autosummarization.)

Our first route summarization lab will use our familiar hub-and-spoke network. Our hub router, R1, has four loopback networks in Area 1. The IP addresses for those four loopbacks are 8.1.1.1, 9.1.1.1, 10.1.1.1, and 11.1.1.1, all with /8 masks.

3/26/2018 9:34 PM - Screen Clipping

The trick to OSPF is knowing when to use the area vs summary command.



Lets go to router 2 and make sure we have those routes.

```
|      8.0.0.0/32 is subnetted, 1 subnets  
| 0 IA    8.1.1.1 [110/65] via 172.12.123.1,  
|      9.0.0.0/32 is subnetted, 1 subnets  
| 0 IA    9.1.1.1 [110/65] via 172.12.123.1,  
|      10.0.0.0/32 is subnetted, 1 subnets  
| 0 IA    10.1.1.1 [110/65] via 172.12.123.1,  
|      11.0.0.0/32 is subnetted, 1 subnets  
| 0 IA    11.1.1.1 [110/65] via 172.12.123.1,  
R2# ping 81.
```

3/26/2018 9:36 PM - Screen Clipping

```
Type escape sequence  
Sending 5, 100-by  
!!!!  
Success rate is 1  
R2#ping 9.1.1.1  
  
Type escape sequence  
Sending 5, 100-by  
!!!!  
Success rate is 1  
R2#ping 10.1.1.1  
  
Type escape sequence  
Sending 5, 100-by  
!!!!  
Success rate is 1  
R2#ping 11.1.1.1
```

3/26/2018 9:36 PM - Screen Clipping

If you want to summarize these routes to make the routes better and faster using a summary route.
(You already know how to do this by identifying the common bits. When they're no longer common you are given the summary route and mask. Then is ospf going to use summary range or area?)

8 **00001000** (last three octets are all zeroes for all four routes)
9 **00001001**
10 **00001010**
11 **00001011**

3/26/2018 9:37 PM - Screen Clipping

Add the numbers in **bold**.

The common bits end after the 6th bit, since the first two addresses have a "0" for the 7th bit and the last two addresses have a "1" in that slot. Just add the numbers in bold and you have your summary route, which in this case is 00001000 for the first octet and all zeroes for the last three. That gives us 8.0.0.0 for the summary route, but we need a mask to go with that route! For the mask, just put a "1" for each common bit and "0" for all others. Since the first six bits are the common bits for these four routes, the mask is 11111100 00000000 00000000 00000000, or 252.0.0.0.

3/26/2018 9:38 PM - Screen Clipping

Now that we have our summary route and mask, we need to apply it! When you're configuring OSPF route summarization on an ABR, use the *area range* command. IOS Help lets us know this command can only be used on ABRs, but the CCNP Route exam will not likely be as kind!

3/26/2018 9:39 PM - Screen Clipping

Heres how to configure that.

```
1(config-router)#area 1 ?
 authentication    Enable authentication
 default-cost      Set the summary default-cost of a
 filter-list       Filter networks between OSPF areas
 nssa              Specify a NSSA area
 range             Summarize routes matching address/
 sham-link         Define a sham link and its parameters
 stub              Specify a stub area
 virtual-link     Define a virtual link and its parameters

1(config-router)#area 1 range ?
 A.B.C.D          IP address to match

1(config-router)#area 1 range 8.0.0.0 ?
 A.B.C.D          IP mask for address

1(config-router)#area 1 range 8.0.0.0 252.0.0.0 ?
 advertise        Advertise this range (default)
 cost             User specified metric for this range
 not-advertise    DoNotAdvertise this range
 <cr>

1(config-router)#area 1 range 8.0.0.0 252.0.0.0 _
```

3/26/2018 9:40 PM - Screen Clipping

```
advertise      Advertise this range (default)
cost          User specified metric for this range
not-advertise DoNotAdvertise this range
<cr>
```

This is more CCIE level to advertise but not create a route.

3/26/2018 9:40 PM - Screen Clipping

And then you can check the summarization an test.

```
Gateway of last resort is not set
  0 IA  8.0.0.0/6 [110/65] via 172.12.123.1, 00:00:06, Serial0/1/0
R2#ping 8.1.1.1
```

3/26/2018 9:41 PM - Screen Clipping

```
!!!!!
Success rate is 10
R2#ping 9.1.1.1

Type escape sequen
Sending 5, 100-byt
!!!!!
Success rate is 10
R2#ping 10.1.1.1

Type escape sequen
Sending 5, 100-byt
!!!!!
Success rate is 10
R2#ping 11.1.1.1
```

3/26/2018 9:42 PM - Screen Clipping

There are other loopbacks that need to be summarized. In this case we wont to redistribute connected subnets. When we do that it makes this router an **ASBR**.

```
!interface Loopback4
 ip address 4.1.1.1 255
!
!interface Loopback5
 ip address 5.1.1.1 255
!
!interface Loopback6
 ip address 6.1.1.1 255
!
!interface Loopback7
 ip address 7.1.1.1 255
!
!interface Loopback8
 ip address 8.1.1.1 255
!
!interface Loopback9
 ip address 9.1.1.1 255
!
!interface Loopback10
 ip address 10.1.1.1 255
```

3/26/2018 9:43 PM - Screen Clipping

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#redis conn subnets
R1(config-router)#^Z
R1#wr
Building configuration
```

3/26/2018 9:44 PM - Screen Clipping

Now check the routing table.

```
Gateway of last resort is not set
```

```
O IA 8.0.0.0/6 [110/65] via 172.12.123.1, 00:01:48, Serial0/1/0
R2#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
       ia - IS-IS inter area, * - candidate default, U - per-user
       o - ODR, P - periodic downloaded static route, H - NHRP, 1
       + - replicated route, % - next hop override
```

```
Gateway of last resort is not set
```

```
O E2 4.0.0.0/8 [110/20] via 172.12.123.1, 00:00:03, Serial0/1/0
O E2 5.0.0.0/8 [110/20] via 172.12.123.1, 00:00:03, Serial0/1/0
O E2 6.0.0.0/8 [110/20] via 172.12.123.1, 00:00:03, Serial0/1/0
O E2 7.0.0.0/8 [110/20] via 172.12.123.1, 00:00:03, Serial0/1/0
O IA 8.0.0.0/6 [110/65] via 172.12.123.1, 00:01:51, Serial0/1/0
R2#
R2#
```

3/26/2018 9:44 PM - Screen Clipping

```
R2#ping 4.1.1.1
Type escape sequence (press Ctrl-C to abort): Sending 5, 100-bytes of data:
!!!!!
Success rate is 100% (5/5)
R2#ping 5.1.1.1
Type escape sequence (press Ctrl-C to abort): Sending 5, 100-bytes of data:
!!!!!
Success rate is 100% (5/5)
R2#ping 6.1.1.1
Type escape sequence (press Ctrl-C to abort): Sending 5, 100-bytes of data:
!!!!!
```

3/26/2018 9:45 PM - Screen Clipping

You can reach them from 2.

Now lets do the **summary address** command. Were not using an area range.

A.V.B.C.V.B Summary mask

```
R1(config-router)#summary-address 4.0.0.0 252.0.0.0 ?
  not-advertise  Do not advertise when translating OSPF type-7 LSA
  tag           Set tag
<cr>

R1(config-router)#summary-address 4.0.0.0 252.0.0.0
R1(config-router)#^Z
R1#
```

3/26/2018 9:46 PM - Screen Clipping

```
R2#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route, H - NHRP, l -
      + - replicated route, % - next hop override

Gateway of last resort is not set

O E2  4.0.0.0/6 [110/20] via 172.12.123.1, 00:00:01, Serial0/1/0
O IA  8.0.0.0/6 [110/65] via 172.12.123.1, 00:03:36, Serial0/1/0
R2#
```

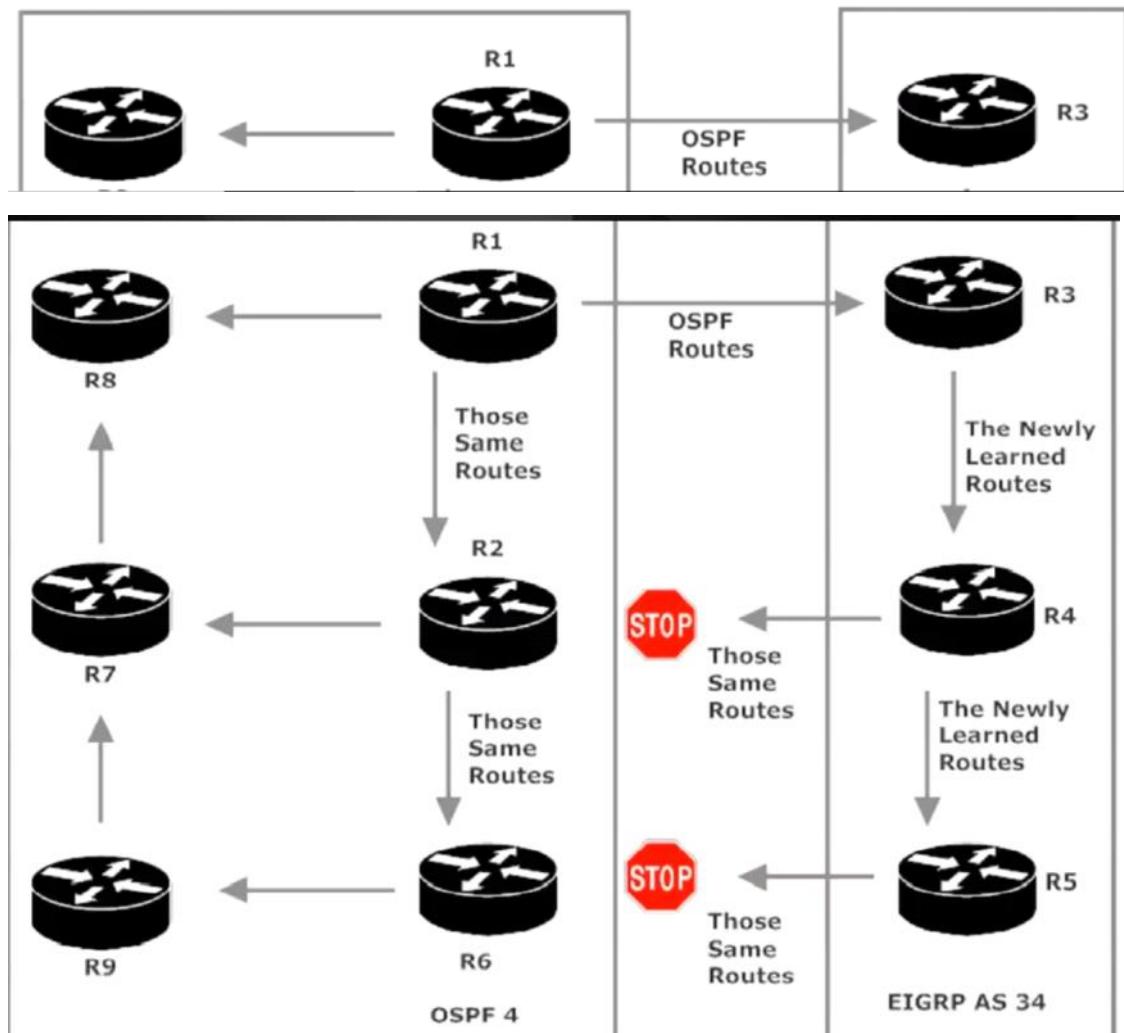
8 separate areas became 2.

Route Redis 1: Route Redistribution

Monday, March 26, 2018 9:48 PM

Simply takes routes from their source and place them into a separate routing process.

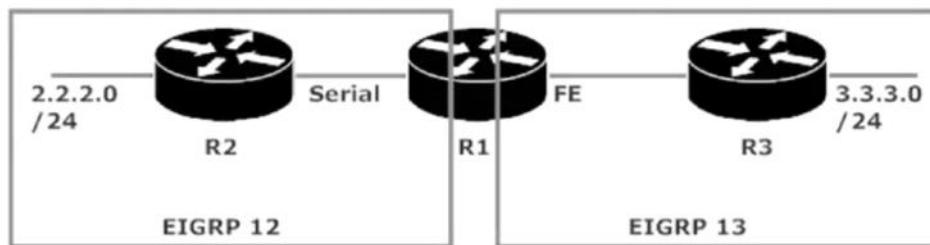
The more route redistribution however, the greater chance of routing loops.



Watch out for route redistribution details regarding seed metrics. Some protocols require the admins to set the seed metric. Another protocol has the metric built in.

One of the rules involving multiple instances.

One of those rules involves running multiple instances of EIGRP on a single router. Routes in one EIGRP AS on that router will *not* be automatically redistributed to the other AS.



R1 has an adjacency with R2 in AS 12 and with R3 in AS 13. You can see the interfaces and subnets in use in the output of *show ip eigrp neighbor*.

```
R1#show ip eigrp neigh
IP-EIGRP neighbors for process 13
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   Address           Interface      (sec) (ms)       Cnt Num
0   10.1.1.3          Fa0/0        11 00:15:13   1     300  0   3
IP-EIGRP neighbors for process 12
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   Address           Interface      (sec) (ms)       Cnt Num
```

So we have router 1 routes.

```
R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 12
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Se
   Address           Interface      (sec) (ms)       Cnt Nu
0   172.12.123.2      Se1/0        130 00:29:38   31    1140  0   2
IP-EIGRP neighbors for process 13
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Se
   Address           Interface      (sec) (ms)       Cnt Nu
0   172.12.13.3       Fa0/0        10 00:16:27    4     200  0   2
R1#
R1#
R1#
R1#show ip route eigrp
 2.0.0.0/24 is subnetted, 1 subnets
D      2.2.2.0 [90/20640000] via 172.12.123.2, 00:29:40, Serial1/0
 3.0.0.0/24 is subnetted, 1 subnets
D      3.3.3.0 [90/156160] via 172.12.13.3, 00:16:29, FastEthernet0/0
R1#
```

But not router 2, because the routes are in separate AS

```
R2#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(12)
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Se
  (sec)   (ms)          Cnt  Nu
0   172.12.123.1      Se0/1/0        179  00:34:52  1521  5000  0   3
R2#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
      ia - IS-IS inter area, * - candidate default, U - per-user static ro
      o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LIS
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

Route Redis 2: OSPF-RIP Redistribution

Wednesday, March 28, 2018 8:47 PM

NOTE: RIP for IPv4 was removed from the CCNP Route exam in 2015. Since its Admin Distance of 120 makes it perfect to illustrate route redistribution details regarding OSPF and EIGRP, and since RIP is still in use in real-world networking, and since RIP for IPv6 is still on the Route exam, I've included RIPv2 in this section's labs.

Using RIPv2 also allows me to introduce you to the seed metric.

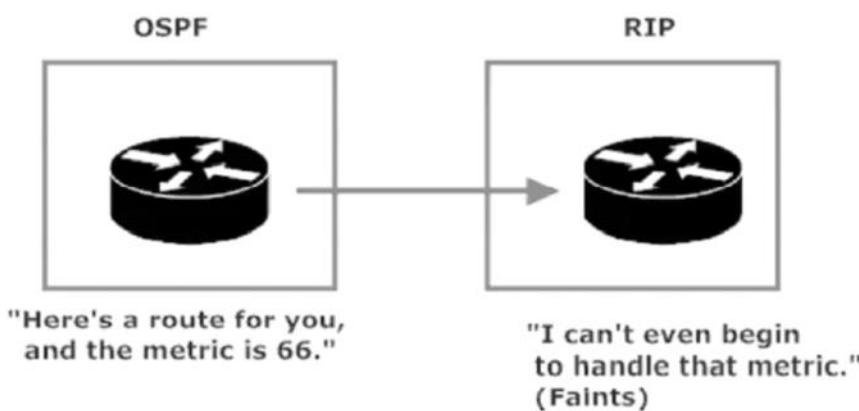
RIP And The Seed Metric

See?

Seriously, folks, configuring RIPv2 is as simple as it gets, and redistributing routes into RIP is just as simple – as long as you remember the seed metric!

```
R1(config)#router rip
R1(config-router)#redistribute static ?
      metric      Metric for redistributed routes
      route-map   Route map reference
<cr>
```

Technically, *redistribute static* is a legal command. Practically, you're not going to get much done, since no seed metric for RIP was defined. RIP's sole metric is hop count. If we redistribute an OSPF route with a cost of 74 into RIP, RIP will not accept the route, since RIP considers a metric of 16 to be unreachable. Anything higher than that is *really* unreachable, and not many OSPF routes have a metric of less than 16.

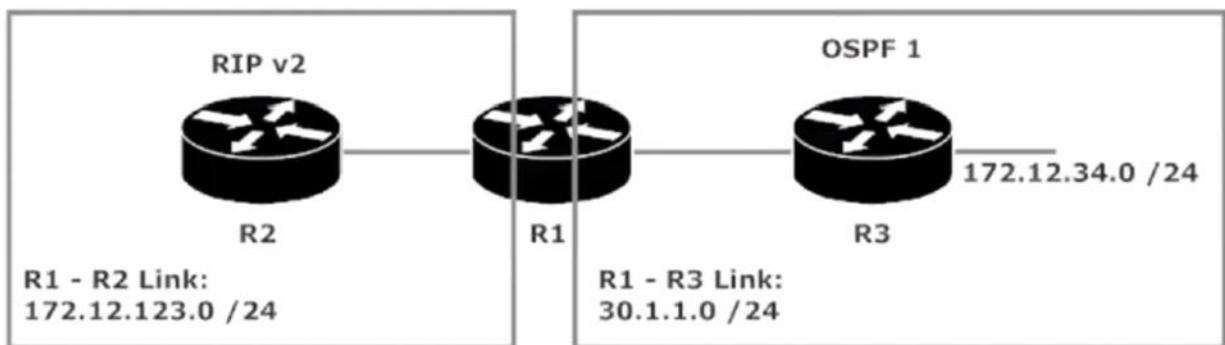


We have to give RIP a value it understands, and that's where the seed metric comes in. That seed metric will increment as the route travels through its new domain, just as it would for a route not learned via route redistribution.

3/28/2018 8:52 PM - Screen Clipping

seed metric is used during redistribution, some protocols have a default **seed metric** of infinite (RIP/EIGRP) and so when you redistribute another **routing protocol** into these you'll have to configure a **seed metric** (not mandatory for redistributing connected **route** and static **route** though). Sep 11, 2013

3/28/2018 8:54 PM - Screen Clipping



The config on R1:

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto
R1(config-router)#network 172.12.123.0

R1(config)#router ospf 1
R1(config-router)#network 30.1.1.0 0.0.0.255 area 0
```

R1 has the 172.12.34.0 /24 network in its OSPF routing table.

3/28/2018 8:54 PM - Screen Clipping

Lets do some route redistribution
Route maps allow you to send a chosen number of routes to be redistributed
You're wanting to redistribute from RIP to OSPF.

```

R1#conf t
Enter configuration commands, one per line
R1(config)#router rip
R1(config-router)#redis ?
  bgp      Border Gateway Protocol (BGP)
  connected  Connected
  eigrp    Enhanced Interior Gateway Ro
  isis     ISO IS-IS
  iso-igrp  IGRP for OSI networks
  metric   Metric for redistributed rou
  mobile   Mobile routes
  odr      On Demand stub Routes
  ospf     Open Shortest Path First (os
  rip      Routing Information Protocol
  route-map Route map reference
  static   Static routes
<cr>

R1(config-router)#redis ospf ?
<1-65535> Process ID

R1(config-router)#redis ospf 1 ?

```

3/28/2018 8:56 PM - Screen Clipping

RIP has slow convergences and a maximum hop count of 16 making it useless. In order to avoid this clear the ip table.

```

R2#clear ip route *
R2#show ip route rip
Codes: L - local, C - connected, S
      D - EIGRP, EX - EIGRP exteri
      N1 - OSPF NSSA external type 1
      E1 - OSPF external type 1,
      i - IS-IS, su - IS-IS summar
      ia - IS-IS inter area, * - c
      o - ODR, P - periodic download
      + - replicated route, % - ne
      ? - candidate default, # - n
      > - selected route

Gateway of last resort is not set

R2#

```

Now on router 1 you need to redis ospf to rip.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#no redistribute ospf 1
R1(config-router)#redistribute ospf 1 ?
  match      Redistribution of OSPF routes
  metric     Metric for redistributed routes
  route-map  Route map reference
  vrf        VPN Routing/Forwarding Table
<cr>
R1(config-router)#redistribute ospf 1
```

3/28/2018 8:58 PM - Screen Clipping

```
R1(config-router)#redistribute ospf 1 metric 2
R1(config-router)#redistribute connected metric 2
R1(config-router)#

```

3/28/2018 8:58 PM - Screen Clipping

And the routes almost immediately come up.

```
R2#show ip route rip
Codes: L - local, C - connected, S - static
      D - EIGRP, EX - EIGRP external, O
      N1 - OSPF NSSA external type 1, N2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1
      ia - IS-IS inter area, * - candidate default
      o - ODR, P - periodic downloaded static
      + - replicated route, % - next hop via interface
      ? - resolve state
Gateway of last resort is not set

      30.0.0.0/24 is subnetted, 1 subnets
R          30.1.1.0 [120/2] via 172.12.123.1
      172.12.0.0/16 is variably subnetted
R          172.12.34.0/24 [120/2] via 172.12.123.1
R2#
```

3/28/2018 8:58 PM - Screen Clipping

You always want to send pings to make sure.

```
R2#ping 30.1.1.1
Type escape sequence
Sending 5, 100-byte I
!!!!!
Success rate is 100 p
R2#
```

3/28/2018 9:00 PM - Screen Clipping

```
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 30.1.1.1
!!!!!
Success rate is 100 percent (5/5)
R2#ping 30.1.1.3

Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 30.1.1.3
.....
Success rate is 0 percent (0/5)
R2#ping 172.12.34.3

Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 172.12.34.3
```

3/28/2018 9:00 PM - Screen Clipping

Why cant we ping 30.1.1.3 or the interface?

[Go to Next Lesson](#)

Route Redis 3: Ping both ways after Redistribution

Wednesday, March 28, 2018 9:01 PM

So why can't we see the pings?

Lets run a debug then ping

```
R2#ping
Protocol [ip]:
Target IP address: 172.12.34.3
Repeat count [5]: 10000
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to
.
BRYANT_ADV_1#3
[Resuming connection 3 to r3 ... ]
```

3/28/2018 9:04 PM - Screen Clipping

```
R3#
R3#debug ip packet
IP packet debugging is on
R3#
*Jun 29 13:12:27.463: IP: s=172.12.123.2 (FastEthernet0/0), d=172.12.34.3,
00, input feature, MCI Check(67), rtype 0, forus FALSE, sendself FALSE, mtu
wdchk FALSE
*Jun 29 13:12:27.463: IP: tableid=0, s=172.12.123.2 (FastEthernet0/0), d=17
34.3 (FastEthernet0/1), routed via RIB
*Jun 29 13:12:27.463: IP: s=172.12.123.2 (FastEthernet0/0), d=172.12.34.3,
00, rcvd 4
*Jun 29 13:12:27.463: IP: s=172.12.123.2 (FastEthernet0/0), d=172.12.34.3,
```

3/28/2018 9:05 PM - Screen Clipping

We're looking for the packets sent from 172.12.34.3 as its telling us that these packets are unrouteable.
The packets are getting to this interface from router 2, but our reply is unrouteable because in the routing table there's no match.

```
R3# show ip route
Codes: L - local, C - connected, S - stati
      D - EIGRP, EX - EIGRP external, O -
      N1 - OSPF NSSA external type 1, N2
      E1 - OSPF external type 1, E2 - OSP
      i - IS-IS, su - IS-IS summary, L1 -
      ia - IS-IS inter area, * - candidat
      o - ODR, P - periodic downloaded st
      + - replicated route, % - next hop

Gateway of last resort is not set

      3.0.0.0/8 is variably subnetted, 2 s
C          3.3.3.0/24 is directly connected,
L          3.3.3.3/32 is directly connected,
      30.0.0.0/8 is variably subnetted, 2
C          30.1.1.0/24 is directly connected
L          30.1.1.3/32 is directly connected
      172.12.0.0/16 is variably subnetted,
C          172.12.34.0/24 is directly connec
L          172.12.34.3/32 is directly connec
```

3/28/2018 9:06 PM - Screen Clipping

```
R1#conf t
Enter configuration commands, one per line. End
R1(config)#router ospf 1^Z
R1#show ip route
*Jun 15 14:50:53.157: %SYS-5-CONFIG_I: Configured
R1#show ip route rip

R1#conf t
Enter configuration commands, one per line. End
R1(config)#router ospf 1
R1(config-router)#redis connected
% Only classful networks will be redistributed
R1(config-router)#
R1#wr
Building configuration...
```

3/28/2018 9:07 PM - Screen Clipping

```
R1(config)#router ospf 1
R1(config-router)#redis connected
% Only classful networks will be redistributed
R1(config-router)#redis connected subnets
R1(config-router)#+Z
R1#wr
Building configuration...
```

3/28/2018 9:08 PM - Screen Clipping

Now we will see the route in the routing table.

```
R3#show ip route ospf
Codes: L - local, C - connected, S - static,
       D - EIGRP, EX - EIGRP external, O - O
       N1 - OSPF NSSA external type 1, N2 -
       E1 - OSPF external type 1, E2 - OSPF
       i - IS-IS, su - IS-IS summary, L1 - I
       ia - IS-IS inter area, * - candidate
       o - ODR, P - periodic downloaded stat
       + - replicated route, % - next hop ov

Gateway of last resort is not set

          172.12.0.0/16 is variably subnetted, 3
o E2      172.12.123.0/24 [110/20] via 30.1.1
```

3/28/2018 9:08 PM - Screen Clipping

Remember an E2 metric only gives you distance from ASBR to destination.

```
    172.12.0.0/16 is
o E2      172.12.123.0/
R3#ping 172.12.123.2

Type escape sequence t
Sending 5, 100-byte IP
!!!!!
Success rate is 100 per
R3#
```

3/28/2018 9:09 PM - Screen Clipping

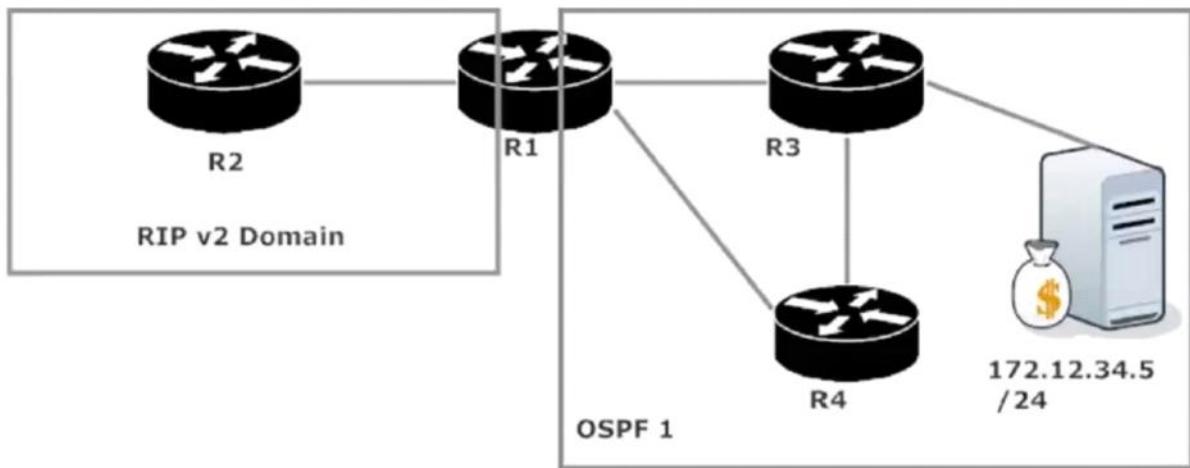
3/28/2018 9:09 PM - Screen Clipping

Route Redis 4: Beware the Routing Suboptimal

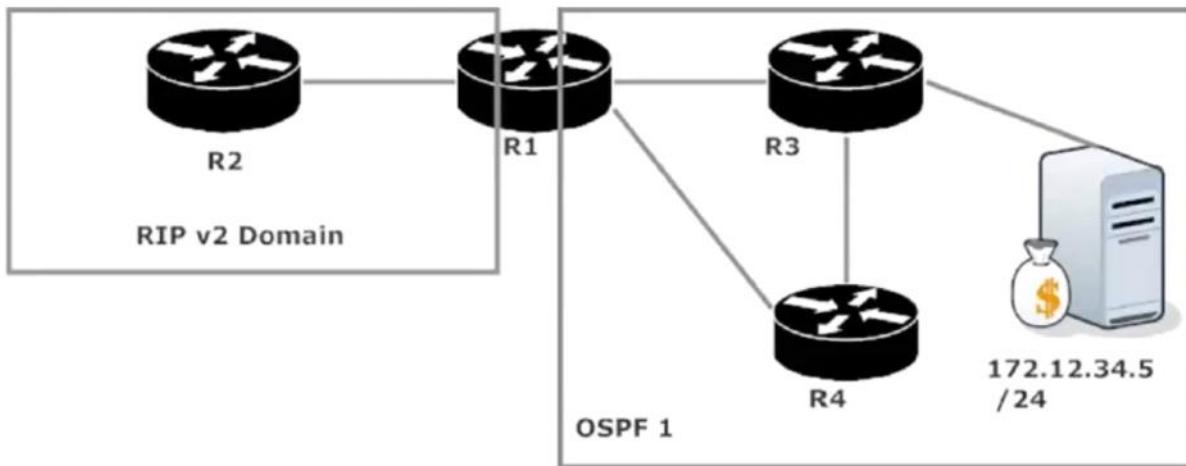
Wednesday, March 28, 2018 9:11 PM

Know The Enemy: Suboptimal Routing and Routing Loops

When route distribution is working, it's a beautiful thing. When it's not working correctly, it can be a major pain in obvious and less-than-obvious ways. The two most common such pains are suboptimal routing (bad) and routing loops (veydy bad). With suboptimal routing, packets eventually get where they're supposed to go, but they're just not getting there as efficiently as they should. As in...



3/28/2018 9:13 PM - Screen Clipping



Only the *important* servers have giant money bags next to them.

There are two valid paths that R2 can use to get packets to that server, R1-R4-R3 and R1-R3. The physically shortest path isn't necessarily the logically shortest path, but for this discussion we'll assume it is. In the case of suboptimal routing, R2 would end up using the longer path. The packets would still get to 172.12.34.5, but not as quickly as they should. Using the longer path would put an unnecessary strain on R4 as well, since it would end up handling packets it shouldn't be handling.

3/28/2018 9:14 PM - Screen Clipping

Packets that enter a routing loop have a much rougher time of it, as do we. Packets in a routing loop will be sent back and forth the same group of routers over and over (and over) without ever reaching their destination. Traceroute is an excellent tool for spotting a routing loop. If you see the same IP addresses over and over with a traceroute, you, my friend, have a routing loop on your hands.

3/28/2018 9:15 PM - Screen Clipping

If you see this then you have a routing loop

```
R2#traceroute 4.4.4.4
```

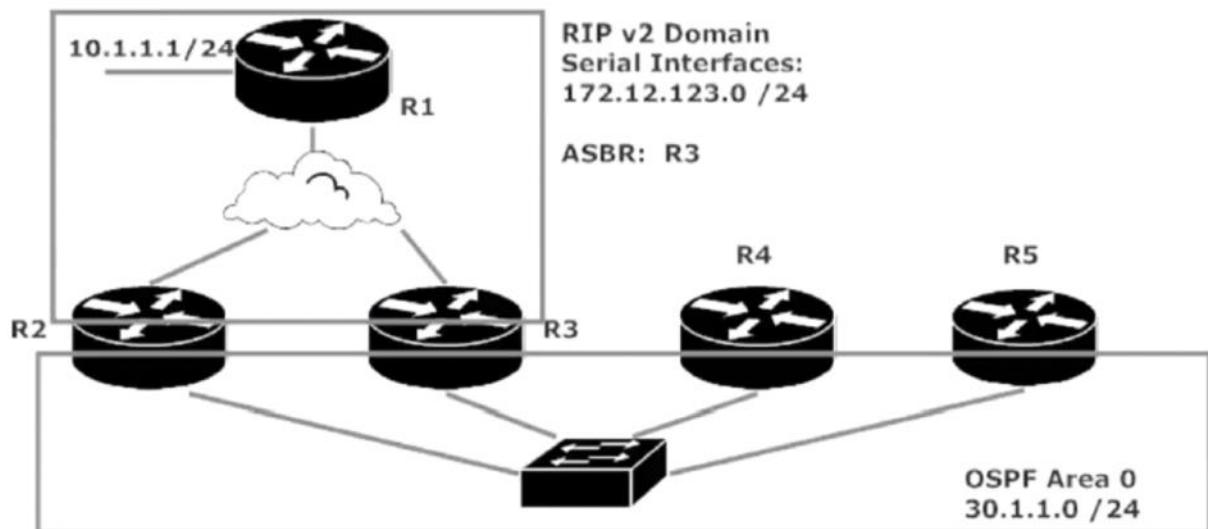
Type escape sequence to abort. (Same as for pings - CB)
Tracing the route to 4.4.4.4

```
1 172.12.23.3 4 msec  
  172.12.123.1 36 msec 32 msec  
2 172.12.123.1 36 msec  
  172.12.123.3 24 msec 28 msec  
3 172.12.123.3 24 msec  
  172.12.123.1 56 msec 56 msec  
4 172.12.123.1 56 msec  
  172.12.123.3 48 msec 48 msec
```

3/28/2018 9:16 PM - Screen Clipping

Redistribution And Adjusting Admin Distances

You know what the AD is and you know the common and not-so-common ADs. That's a good thing, since we're going to need that knowledge in our next lab!



Our objective is for all routers to have the 10.1.1.0 /24 network in their routing tables. The ASBR in this lab is R3. Our first step is *always* to make sure the ASBR has the route to be redistributed!

3/28/2018 9:16 PM - Screen Clipping

```

R3#
R3#show ip route rip
Codes: L - local, C - connected, S - static
      D - EIGRP, EX - EIGRP external, O -
      N1 - OSPF NSSA external type 1, N2 -
      E1 - OSPF external type 1, E2 - OSPF
      i - IS-IS, su - IS-IS summary, L1 -
      ia - IS-IS inter area, * - candidate
      o - ODR, P - periodic downloaded sta-
      + - replicated route, % - next hop o
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
R            10.1.1.0 [120/1] via 172.12.123.1,
R3#

```

3/28/2018 9:17 PM - Screen Clipping

These are the exact same routes but now were going to do two way route redis.

First redis into OSPF

```

R          10.1.1.0 [120/
R3#conf t
Enter configuration mode
R3(config)#router ospf
R3(config-router)#redis
R3(config-router)#redis
R3(config-router)#^Z
R3#
*7Jun 29 14:04:03.803% 

```

3/28/2018 9:18 PM - Screen Clipping

Lets see Router 4 before we config rip

```
R4#show ip route ospf
Codes: L - local, C - connected, S
      D - EIGRP, EX - EIGRP exten-
      N1 - OSPF NSSA external type 1
      E1 - OSPF external type 1,
      i - IS-IS, su - IS-IS summar-
      ia - IS-IS inter area, * -
      o - ODR, P - periodic download-
      + - replicated route, % - ren-
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1
O E2      10.1.1.0 [110/20] via 30.
      172.12.0.0/24 is subnetted,
O E2      172.12.123.0 [110/20] via
R4#
```

3/28/2018 9:18 PM - Screen Clipping

Now redis rip

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router rip
R3(config-router)#redis connected metric 2
R3(config-router)#
R3#
```

3/28/2018 9:18 PM - Screen Clipping

```
R4#show ip route ospf
Codes: L - local, C - connected, S
      D - EIGRP, EX - EIGRP exten-
      N1 - OSPF NSSA external type 1
      E1 - OSPF external type 1,
      i - IS-IS, su - IS-IS summar-
      ia - IS-IS inter area, * -
      o - ODR, P - periodic download-
      + - replicated route, % - ren-
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1
O E2      10.1.1.0 [110/20] via 30.
      172.12.0.0/24 is subnetted,
O E2      172.12.123.0 [110/20] via
R4#
```

3/28/2018 9:19 PM - Screen Clipping

So lets ping to test our redis.

```
O E2      172.17.125.  
R4#ping 10.1.1.1  
  
Type escape sequence  
Sending 5, 100-byte  
!!!!  
Success rate is 100  
R4#  
BRYANT_ADV_1#5  
[Resuming connection]
```

3/28/2018 9:19 PM - Screen Clipping

Whenever I do something at 1 ASBR its good to check out any additional ones.

However we now notice that router 2 is missing a route.

Hmm. Maybe we better stick around a while. What about the OSPF table?

```
R2#show ip route ospf  
 10.0.0.0/24 is subnetted, 1 subnets  
O E2      10.1.1.0 [110/20] via 30.1.1.3, 00:04:24, FastEthernet0/0
```

R2 is now showing an OSPF route where it once had a RIP route, and that "once had" time was before route redistribution. Why the change? Admin distance! R2 is now hearing about the 10.1.1.0 route from two sources. R1 is advertising that route over the RIP domain, and R3 is advertising it via OSPF.



3/28/2018 9:21 PM - Screen Clipping

Router 2 is hearing about 10 route from 2 different sources and since the prefix length is the same, then AD comes into the picture. Therefore, OSPF having a lower metric 110 will be chosen over RIP 120. Therefore it will go to router 3 for the route.

Route Redis 5: Changing the AD for an entire Protocol

Wednesday, March 28, 2018 9:23 PM

We want to change the AD.

```
R2#show ip route ospf
Codes: L - local, C - connected, S - static, R - RI
      D - EIGRP, EX - EIGRP external, O - OSPF, IA
      N1 - OSPF NSSA external type 1, N2 - OSPF NS
      E1 - OSPF external type 1, E2 - OSPF external
      i - IS-IS, su - IS-IS summary, L1 - IS-IS le
      ia - IS-IS inter area, * - candidate default
      o - ODR, P - periodic downloaded static rout
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
          10.0.0.0/24 is subnetted, 1 subnets
O E2      10.1.1.0 [110/20] via 30.1.1.3, 00:02:37,
R2#
```

3/28/2018 9:26 PM - Screen Clipping

Change AD on OSPF routes past and present.

```
          10.0.0.0/24 is subnetted, 1 s
O E2      10.1.1.0 [110/20] via 30.1
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#distance ?
<1-255>  Administrative distance
ospf      OSPF distance
```

```
R2(config-router)#distance 121 ?
A.B.C.D  IP Source address
<cr>
```

```
R2(config-router)#distance 121
R2(config-router)#
```

So now the RIP route should be chosen because it has a small ad

```
R2#show ip route rip
Codes: L - local, C - connected, S - stati
      D - EIGRP, EX - EIGRP external, O -
      N1 - OSPF NSSA external type 1, N2
      E1 - OSPF external type 1, E2 - OSP
      i - IS-IS, su - IS-IS summary, L1 -
      ia - IS-IS inter area, * - candidat
      o - ODR, P - periodic downloaded st
      + - replicated route, % - next hop

Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
R          10.1.1.0 [120/1] via 172.12.123.1
R2#
```

Now lets make sure the OSPF route comes back into the table if the RIP leaves. Lets close the serial interface on router 2

```
R2#
R2#conf t
Enter configuration commands, one
R2(config)#int serial 0/1/0
R2(config-if)#shut
R2(config-if)#^Z
R2#
*Jun 29 18:06:45.090: %SYS-5-CONFI
```

```
R2#show ip route ospf
Codes: L - local, C - connected, S - static
      D - EIGRP, EX - EIGRP external, O - ODR
      N1 - OSPF NSSA external type 1, N2 -
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - Level 1
      ia - IS-IS inter area, * - candidate default
      o - ODR, P - periodic downloaded static
      + - replicated route, % - next hop over interface
      ? - route to interface
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
O E2      10.1.1.0 [121/20] via 30.1.1.3, 00:00:00
      172.12.0.0/24 is subnetted, 1 subnets
O E2      172.12.123.0 [121/20] via 30.1.1.3, 00:00:00
R2#
```

3/28/2018 9:28 PM - Screen Clipping

```
R2#ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echo Request(s) to 10.1.1.1
!!!!!
Success rate is 100 percent
R2#
```

3/28/2018 9:29 PM - Screen Clipping

And we do. We have 2 separate routes and until the serial interface was closed, the connected network went down.

Route Redis 6: Changing the AD Based on OSPF route

Wednesday, March 28, 2018 9:31 PM

RIP has been removed.

```
R2#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA
      E1 - OSPF external type 1, E2 - OSPF external
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level 1
      ia - IS-IS inter area, * - candidate default,
      o - ODR, P - periodic downloaded static route
      + - replicated route, % - next hop override

Gateway of last resort is not set

      3.0.0.0/32 is subnetted, 1 subnets
O IA      3.3.3.3 [110/2] via 30.1.1.3, 00:05:13, FastEthernet0/0
      5.0.0.0/24 is subnetted, 1 subnets
O E2      5.5.5.0 [110/20] via 30.1.1.3, 00:05:13, FastEthernet0/1
      6.0.0.0/24 is subnetted, 1 subnets
O E2      6.6.6.0 [110/20] via 30.1.1.3, 00:05:13, FastEthernet0/2
      10.0.0.0/24 is subnetted, 1 subnets
O E2      10.1.1.0 [110/20] via 30.1.1.3, 00:05:13, FastEthernet0/3
      33.0.0.0/32 is subnetted, 1 subnets
O          33.3.3.3 [110/2] via 30.1.1.3, 00:05:13, FastEthernet0/4
      172.12.0.0/24 is subnetted, 1 subnets
O E2      172.12.123.0 [110/20] via 30.1.1.3, 00:05:13, FastEthernet0/5
R2#
```

What if we want to change the distance on a particular route?

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z
R2(config)#router ospf 1
R2(config-router)#distance ?
<1-255>  Administrative distance
ospf      OSPF distance

R2(config-router)#distance ospf ?
external   External type 5 and type 7 routes
inter-area Inter-area routes
intra-area Intra-area routes
```

3/29/2018 9:03 AM - Screen Clipping

```
R2(config-router)#distance ospf external 175 inter-area 165 intra-area 155
R2(config-router)#^Z
R2#
*Jun 29 18:39:01.746: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

3/29/2018 9:04 AM - Screen Clipping

```
R2#show ip route ospf
Codes: L - local, C - connecte
      D - EIGRP, EX - EIGRP e
      N1 - OSPF NSSA external
      E1 - OSPF external type
      i - IS-IS, su - IS-IS s
      ia - IS-IS inter area,
      o - ODR, P - periodic d
      + - replicated route, %

Gateway of last resort is not

      3.0.0.0/32 is subnetted,
0 IA      3.3.3.3 [165/2] via 3
      5.0.0.0/24 is subnetted,
0 E2      5.5.5.0 [175/20] via
      6.0.0.0/24 is subnetted,
0 E2      6.6.6.0 [175/20] via
      10.0.0.0/24 is subnetted
0 E2      10.1.1.0 [175/20] via
      33.0.0.0/32 is subnetted
0         33.3.3.3 [155/2] via
      172.12.0.0/24 is subnetted
0 E2      172.12.123.0 [175/20]
```

3/29/2018 9:05 AM - Screen Clipping

Not used very much. You'll typically use the distance more with the protocol than via the route.

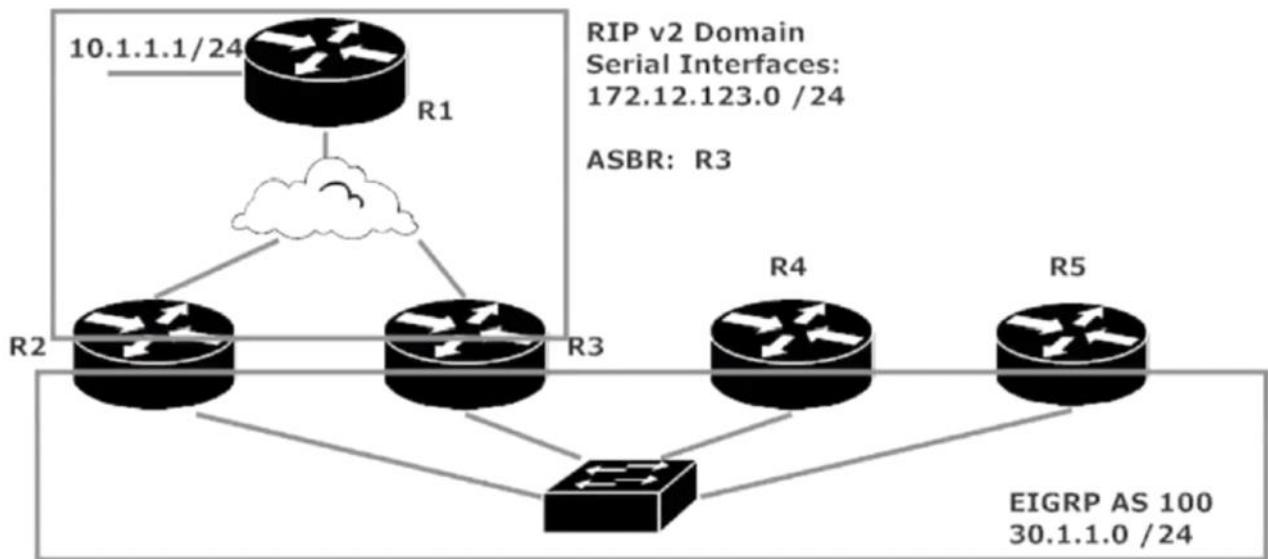
Route Redis 7: Redistributing EIGRP and tweaking the AD

Thursday, March 29, 2018 9:05 AM

RIP > EIGRP

3/29/2018 9:09 AM - Screen Clipping

EIGRP And Route Redistribution



The addressing is the same as in the previous lab. Only the dynamic routing protocol has been changed (to protect the innocent). I've also removed the 5.0.0.0 and 6.0.0.0 networks from R1 that were added at the end of the previous lab.

3/29/2018 9:06 AM - Screen Clipping

```
R3(config-router)#default-metric ?
<1-4294967295> Bandwidth in Kbits per second

R3(config-router)#default-metric 1544 ?
<0-4294967295> delay metric in 10 microsecond units

R3(config-router)#default-metric 1544 10 ?
<0-255> Reliability metric where 255 is 100% reliable

R3(config-router)#default-metric 1544 10 255 ?
<1-255> Effective bandwidth metric (Loading) where 255

R3(config-router)#default-metric 1544 10 255 1 ?
<1-65535> Maximum Transmission Unit metric of the path

R3(config-router)#default-metric 1544 10 255 1 1500 ?
<cr>
```

```
R3(config-router)#redistribute rip metric ?
<1-4294967295> Bandwidth metric in Kbits per second

R3(config-router)#redistribute rip metric 1544 ?
<0-4294967295> EIGRP delay metric, in 10 microsecond units

R3(config-router)#redistribute rip metric 1544 10 ?
<0-255> EIGRP reliability metric where 255 is 100% reliable

R3(config-router)#redistribute rip metric 1544 10 25 ?
<1-255> EIGRP Effective bandwidth metric (Loading) where 25

R3(config-router)#redistribute rip metric 1544 10 255 ?
<1-255> EIGRP Effective bandwidth metric (Loading) where 255

R3(config-router)#redistribute rip metric 1544 10 255 1 ?
<1-65535> EIGRP MTU of the path

R3(config-router)#redistribute rip metric 1544 10 255 1 1500 ?
route-map Route map reference
<cr>

R3(config-router)#redistribute rip metric 1544 10 255 1 1500
```

3/29/2018 9:09 AM - Screen Clipping

Both tables will now show the redis routes.

```
R4#show ip route eigrp
Codes: L - local, C - connected, S - static
      D - EIGRP, EX - EIGRP external
      N1 - OSPF NSSA external type 1
      E1 - OSPF external type 1, E2 -
      i - IS-IS, su - IS-IS summary
      ia - IS-IS inter area, * - candidate
      o - ODR, P - periodic downloaded
      + - replicated route, % - next
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
D EX      10.1.1.0 [170/1662976] via 30
      172.12.0.0/24 is subnetted, 1 subnets
D EX      172.12.123.0 [170/1662976]
R4#
```

3/29/2018 9:10 AM - Screen Clipping

```
R5#show ip route eigrp
Codes: L - local, C - connected, S - static
      D - EIGRP, EX - EIGRP external,
      N1 - OSPF NSSA external type 1,
      E1 - OSPF external type 1, E2 -
      i - IS-IS, su - IS-IS summary,
      ia - IS-IS inter area, * - candidate
      o - ODR, P - periodic downloaded
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
D EX      10.1.1.0 [170/1662976] via 30
      172.12.0.0/24 is subnetted, 1 subnets
D EX      172.12.123.0 [170/1662976] via 30
```

3/29/2018 9:10 AM - Screen Clipping

D EX = route learned via route redis EIGRP. Notice redis EIGRP external routes have an ad of 170 and internal have 90.

In the RIP-OSPF redistribution lab, R2 has this route as a RIP route before redistribution and an OSPF route afterwards. In this lab, R2 has this route as a RIP route before *and* after redistribution. Why? Because the EIGRP external route AD of 170 is higher than that of RIP's 120, so the RIP route is preferred over the EIGRP external route.



Should R2's serial interface mysteriously go down again (heh heh heh), the EIGRP external path will be put into the routing table.

```
R2(config)#int serial 0/1/0  
R2(config-if)#shut
```

3/29/2018 9:12 AM - Screen Clipping

You can either change the AD or shut down the interface.

If you want to change the AD you have to put an internal and external AD

```
R2(config)#router eigrp 100  
R2(config-router)#distance ?  
  <1-255> Set route administrative distance  
eigrp      Set distance for internal and external routes  
  
R2(config-router)#distance eigrp ?  
  <1-255> Distance for internal routes  
  
R2(config-router)#distance eigrp 90 ?  
  <1-255> Distance for external routes  
  
R2(config-router)#distance eigrp 90 119 ?  
  <cr>  
  
R2(config-router)#distance eigrp 90 119  
R2(config-router)#^Z
```

3/29/2018 9:14 AM - Screen Clipping

```
R2#  
*Jun 29 20:12:02.891: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is down: route configuration changed  
*Jun 29 20:12:02.891: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is down: route configuration changed  
*Jun 29 20:12:02.891: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is down: route configuration changed  
R2#  
*Jun 29 20:12:04.719: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is up: new adjacency  
*Jun 29 20:12:04.723: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is up: new adjacency  
*Jun 29 20:12:04.727: %DUAL-5-NBRCHANGE: EIGRP-IPv4  
Ethernet0/0) is up: new adjacency  
R2#
```

3/29/2018 9:15 AM - Screen Clipping

Now the adjacency's come back up.

```
R2#show ip route eigrp  
Codes: L - local, C - connected,  
       D - EIGRP, EX - EIGRP ext  
       N1 - OSPF NSSA external t  
       E1 - OSPF external type 1  
       i - IS-IS, su - IS-IS sum  
       ia - IS-IS inter area, *  
       o - ODR, P - periodic dow  
       + - replicated route, % -  
  
Gateway of last resort is not set  
  
          10.0.0.0/24 is subnetted,  
D EX      10.1.1.0 [119/1662976]  
R2#
```

3/29/2018 9:15 AM - Screen Clipping

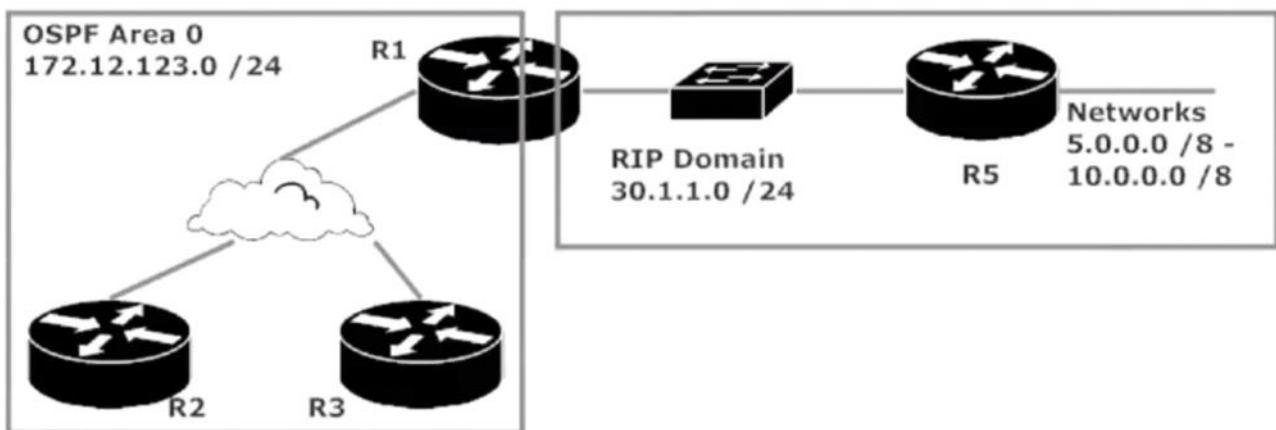
Route Redis 8: Using Distribute Lists with OSPF

Thursday, March 29, 2018 9:16 AM

Fine-Tuning Route Redistribution

You'll often run into route redistribution situations where you want some routes to be fully redistributed, and others to be only partially so. A great way to fine-tune redistribution is with distribute-lists. Distribute lists use ACLs (there they are again!) to define the routes to be redistributed. They also define the routes to *not* be redistributed, whether that denial be explicit or implicit.

You'll see what I mean as we work through this lab!



3/29/2018 9:17 AM - Screen Clipping

Distribute lists use ACLs to define the route to be redis and defining routes not doing redis whether the deny be explicit or implicit.

```
R1#show ip route rip
R  5.0.0.0/8 [120/1] via 30.1.1.5, 00
R  6.0.0.0/8 [120/1] via 30.1.1.5, 00
R  7.0.0.0/8 [120/1] via 30.1.1.5, 00
R  8.0.0.0/8 [120/1] via 30.1.1.5, 00
R  9.0.0.0/8 [120/1] via 30.1.1.5, 00
R  10.0.0.0/8 [120/1] via 30.1.1.5, 0
R1#
R1#show config | section router ospf 1
router ospf 1
  log-adjacency-changes
  redistribute connected subnets
  redistribute rip subnets
  network 172.12.123.0 0.0.0.255 area 0
  neighbor 172.12.123.2
  neighbor 172.12.123.3
```

3/29/2018 9:18 AM - Screen Clipping

What were doing is writing an ACL to identify these routes. The distribute list is what goes into the ospf config.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z
R1(config)#access-list 17 deny 8.0.0.0 0.255.255.255
R1(config)#access-list 17 deny 9.0.0.0 0.255.255.255
R1(config)#access-list 17 perm any
```

3/29/2018 9:20 AM - Screen Clipping

Now lets take this distributed list and add it to the ospf protocol

```
R1(config-router)#distribute-list 17 ?
  in   Filter incoming routing updates
  out  Filter outgoing routing updates

R1(config-router)#distribute-list 17 out ?
  Async           Async interface
  BVI             Bridge-Group Virtual Interface
  CDMA-IX         CDMA Ix interface
  CTunnel          CTunnel interface
  Dialer          Dialer interface
  FastEthernet    FastEthernet IEEE 802.3
  Lex              Lex interface
  Loopback         Loopback interface
  MFR              Multilink Frame Relay bundle interface
  Multilink        Multilink-group interface
  Null             Null interface
  Port-channel    Ethernet Channel of interfaces
  Serial           Serial
  Tunnel           Tunnel i
```

3/29/2018 9:21 AM - Screen Clipping

```
R1(config-router)#distribute-list 17 out serial1/0
% Interface not allowed with OUT for OSPF
R1(config-router)#

```

3/29/2018 9:22 AM - Screen Clipping

```
R2#show ip route ospf
Codes: L - local, C - connected, S - si-
      D - EIGRP, EX - EIGRP external,
      N1 - OSPF NSSA external type 1,
      E1 - OSPF external type 1, E2 -
      i - IS-IS, su - IS-IS summary,
      ia - IS-IS inter area, * - candi-
      o - ODR, P - periodic downloaded
      + - replicated route, % - next

Gateway of last resort is not set

O E2  5.0.0.0/8 [110/20] via 172.12.12.1
O E2  6.0.0.0/8 [110/20] via 172.12.12.1
O E2  7.0.0.0/8 [110/20] via 172.12.12.1
O E2  10.0.0.0/8 [110/20] via 172.12.12.1
      30.0.0.0/24 is subnetted, 1 subne
O E2      30.1.1.0 [110/20] via 172.12.12.1
R2#
```

3/29/2018 9:22 AM - Screen Clipping

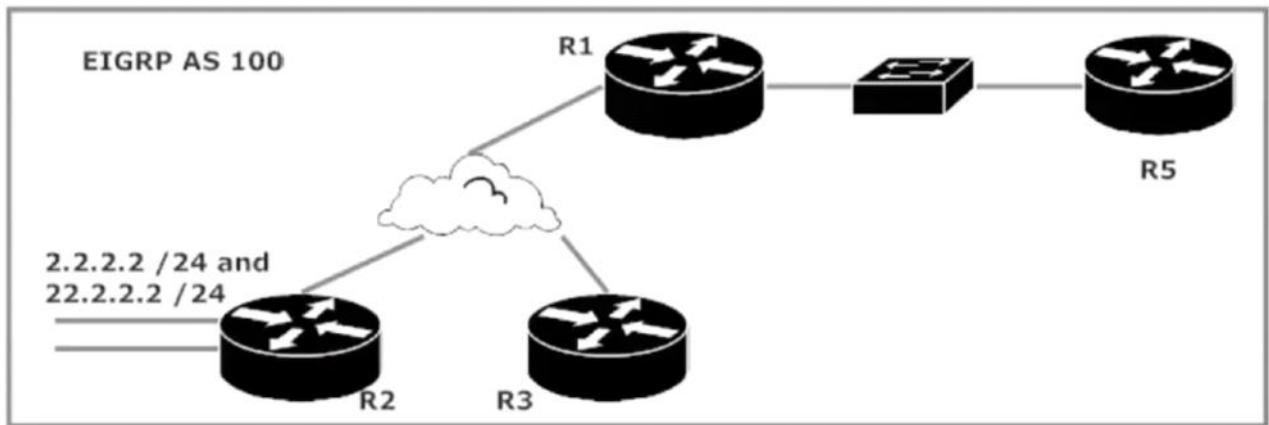
And the routes are now gone.

Route Redis 9: Distribute Lists and EIGRP updates

Thursday, March 29, 2018 9:24 AM

You can do the same distribute lists with EIGRP without making the interface passive.

Distribute lists can filter all routes from being advertised via a given interface without making that interface passive and losing the adjacency, as you'll see in our next lab! The addressing over the R1-R2-R3 and R1-R5 networks is the same as the previous lab.



R2 is advertising two routes into EIGRP. R1 sees them, as does R5.

3/29/2018 9:27 AM - Screen Clipping

```
R5#show ip route eigrp
Codes: L - local, C - connected, S - static,
        D - EIGRP, EX - EIGRP external, O - OSPF
        N1 - OSPF NSSA external type 1, N2 - OSPF
        E1 - OSPF external type 1, E2 - OSPF ext
        i - IS-IS, su - IS-IS summary, L1 - IS
        ia - IS-IS inter area, * - candidate default
        o - ODR, P - periodic downloaded stat
Gateway of last resort is not set
```

```
      2.0.0.0/24 is subnetted, 1 subnets
D          2.2.2.0 [90/20642560] via 30.1.1.1,
      22.0.0.0/24 is subnetted, 1 subnets
D          22.2.2.0 [90/20642560] via 30.1.1.1
      172.12.0.0/24 is subnetted, 1 subnets
D          172.12.123.0 [90/20514560] via 30.1
```

3/29/2018 9:27 AM - Screen Clipping

Router 5 sees all of the EIGRP routes. So now lets filter them without bringing down the adjacency.
Passive interfaces will not accept hello advertisements.

```
R1(config-router)#passive-int fast0/0
R1(config-router)#^Z
R1#
*Jun 15 23:50:24.384: %DUAL-5-NBRCHANGE: 
  interface Ethernet0/0) is down: interface passive
R1#
*Jun 15 23:50:25.425: %SYS-5-CONFIG_I: 
  R1#conf t
Enter configuration commands, one per line
R1(config)#router eigrp 100
R1(config-router)#no passive-int fast0/0
R1(config-router)#
*Jun 15 23:50:52.746: %DUAL-5-NBRCHANGE: 
  interface Ethernet0/0) is up: new adjacency
R1(config-router)#^Z
R1#
*Jun 15 23:50:54.838: %SYS-5-CONFIG_I: 
  R1#x
BRYANT_ADV_1#5
[Resuming connection 5 to r5 ... ]
```

3/29/2018 9:29 AM - Screen Clipping

Create the access list and then distribute via EIGRP protocol

```
R1(config)#access-list 5 deny any
R1(config)#router eigrp 100
R1(config-router)#distribute-list 5 ?
  in  Filter incoming routing updates
  out Filter outgoing routing updates

R1(config-router)#distribute-list 5 out ?
  Async                  Async interface
  BVI                   Bridge-Group Virtual Interface
  CDMA-IX               CDMA IX interface
```

3/29/2018 9:30 AM - Screen Clipping

```
R1(config-router)#distribute-list 5 out fast0/0 ?
<cr>
R1(config-router)#distribute-list 5 out fast0/0
R1(config-router)#+Z
```

3/29/2018 9:30 AM - Screen Clipping

The adjacency stays but the routes are gone.

```
R5#show ip route eigrp
*Jul 4 19:47:39.867: %DUAL-5-NBRCHANGE:
bitEthernet0/0) is resync: peer graceful
R5#show ip route eigrp
Codes: L - local, C - connected, S - sta
      D - EIGRP, EX - EIGRP external, O
      N1 - OSPF NSSA external type 1, N
      E1 - OSPF external type 1, E2 - O
      i - IS-IS, su - IS-IS summary, L1
      ia - IS-IS inter area, * - candid
      o - ODR, P - periodic downloaded
```

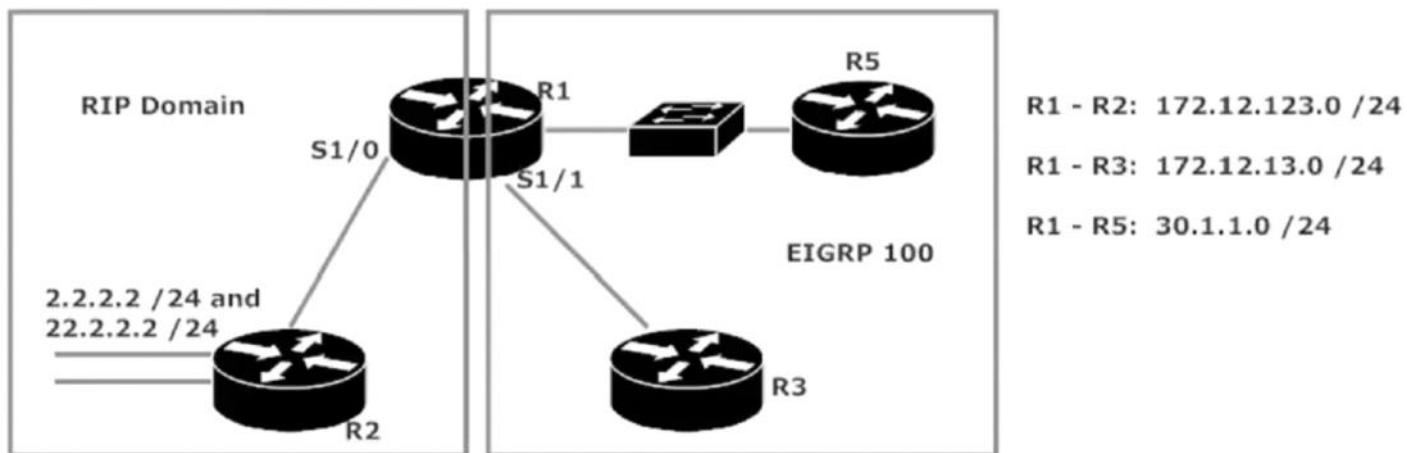
Gateway of last resort is not set

```
R5#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address                  Interface
0   30.1.1.1                 Gi0/0
R5#
```

Route Redis 10: Route redis and Distribute lists

Thursday, March 29, 2018 9:40 AM

We can also use *distribute-list* to filter EIGRP routes when redistribution is involved. This lab has a slightly different topology from the last one; R1 and R3 are connected over a separate subnet than the one used for R1 and R2. R2 is advertising the same two routes as in the previous lab.



R1 sees both routes advertised by R2.

3/29/2018 9:40 AM - Screen Clipping

```
R1#show ip route rip
      2.0.0.0/24 is subnetted, 1 subnets
R          2.2.2.0 [120/1] via 172.12.123.2, 00:00:10, Serial1/0
      22.0.0.0/24 is subnetted, 1 subnets
R          22.2.2.0 [120/1] via 172.12.123.2, 00:00:10, Serial1/0
R1#
```

3/29/2018 9:40 AM - Screen Clipping

```
R1#conf t
Enter configuration commands, one per line.  En
R1(config)#router eigrp 100
R1(config-router)#redistribute rip ?
    metric      Metric for redistributed routes
    route-map   Route map reference
<cr>
```

3/29/2018 9:41 AM - Screen Clipping

If I was just going to redistribute rip I would do this command.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router eigrp 100
R1(config-router)#redistribute rip ?
  metric      Metric for redistributed routes
  route-map   Route map reference
<cr>

R1(config-router)#redistribute rip metric ?
  <1-4294967295>  Bandwidth metric in Kbits per second

R1(config-router)#redistribute rip metric 1544 ?
  <0-4294967295>  EIGRP delay metric, in 10 microsecond units

R1(config-router)#redistribute rip metric 1544 10 ?
  <0-255>  EIGRP reliability metric where 255 is 100% reliable

R1(config-router)#redistribute rip metric 1544 10 1 ?
  <1-255>  EIGRP Effective bandwidth metric (Loading) where 255 i

R1(config-router)#redistribute rip metric 1544 10 255 1 ?
  <1-65535>  EIGRP MTU of the path

R1(config-router)#redistribute rip metric 1544 10 255 1 1500
```

3/29/2018 9:42 AM - Screen Clipping

Route Redis 11: Multiple Redis List

Saturday, March 31, 2018 10:40 AM

```
R3#
R3#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OS
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ex
      E1 - OSPF external type 1, E2 - OSPF external typ
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route, H
      + - replicated route, % - next hop override

Gateway of last resort is not set

      2.0.0.0/24 is subnetted, 1 subnets
D EX    2.2.2.0 [170/2172416] via 172.12.13.1, 00:03:55
      22.0.0.0/24 is subnetted, 1 subnets
D EX    22.2.2.0 [170/2172416] via 172.12.13.1, 00:03:5
      30.0.0.0/24 is subnetted, 1 subnets
D       30.1.1.0 [90/2172416] via 172.12.13.1, 00:55:18
      172.12.0.0/16 is variably subnetted, 3 subnets, 2
D EX    172.12.123.0/24 [170/21024000] via 172.12.13.1,
R3#
```

3/31/2018 10:40 AM - Screen Clipping

Lets deny the one filtered route.

```
^Jun
R1(config-router)#
R1(config-router)#exit
R1(config)#access-list 3 deny 30.1.1.0 0.0.0.255
R1(config)#access-list 3 perm any
R1(config)#
```

3/31/2018 10:41 AM - Screen Clipping

No we lets filter routing updates OUT (as always). Distribute list 3 out as a general list.

```
R1(config-router)#distribute-list 3 out
R1(config-router)#

```

3/31/2018 10:42 AM - Screen Clipping

```
R1(config-router)#distribute-list 3 out
R1(config-router)#
*Jun 16 14:37:08.670: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100:
tEthernet0/0) is resync: route configuration changed
*Jun 16 14:37:08.670: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100:
Serial1/1) is resync: route configuration changed
R1(config-router)#

```

3/31/2018 10:42 AM - Screen Clipping

Adjacencys resync

Now lets got to routers 3 and 5

```
R3#show ip route eigrp
Codes: L - local, C - co
      D - EIGRP, EX - E
      N1 - OSPF NSSA ex
      E1 - OSPF external
      i - IS-IS, su - I
      ia - IS-IS inter
      o - ODR, P - peri
      + - replicated ro
Gateway of last resort is
      2.0.0.0/24 is sub
D EX      2.2.2.0 [170/21]
      22.0.0.0/24 is sub
D EX      22.2.2.0 [170/21]
      172.12.0.0/16 is v
D EX      172.12.123.0/24
R3#

```

3/31/2018 10:42 AM - Screen Clipping

The internal routes are gone. We wanted make sure router 3 didn't have to 30.1.1.3 route anymore, but we also see no internal routes. On 3 to 5

```

R5#show ip route eigrp
Codes: L - local, C - con
      D - EIGRP, EX - EI
      N1 - OSPF NSSA ext
      E1 - OSPF external
      i - IS-IS, su - IS
      ia - IS-IS inter a
      o - ODR, P - perio
Gateway of last resort is

R5#show ip eigrp neighbor
EIGRP-IPv4 Neighbors for
H   Address
0   30.1.1.1

```

3/31/2018 10:43 AM - Screen Clipping

BUT (and you knew that was coming), what if we now want to filter a single route from R3's table? We'd obviously need a different ACL and a different distribute-list. That brings up three questions...

Can we use more than one distribute-list on the same router?

If so, can we use more than one distribute-list in the same protocol config?

If we can, what's the net effect to all of our routing tables?

Let's find out! We just got the word that R3 shouldn't have the 30.1.1.0 /24 network in its table, but it should continue to have the external routes. We'll write an ACL identifying the route to be filtered while allowing all others...

```

R1(config)#access-list 33 deny 30.1.1.0 0.0.0.255
R1(config)#access-list 33 permit any

```

.. and we'll apply another distribute-list in the EIGRP config without specifying an interface or protocol.

```

R1(config)#router eigrp 100
R1(config-router)#distribute-list 33 out

```

```

*Jun 14 07:48:20.511: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor

```

3/31/2018 10:43 AM - Screen Clipping

Now we know that a general distribute list, one that doesn't specify interface or protocol, doesn't overwrite and specific distribute list on interface or protocol.

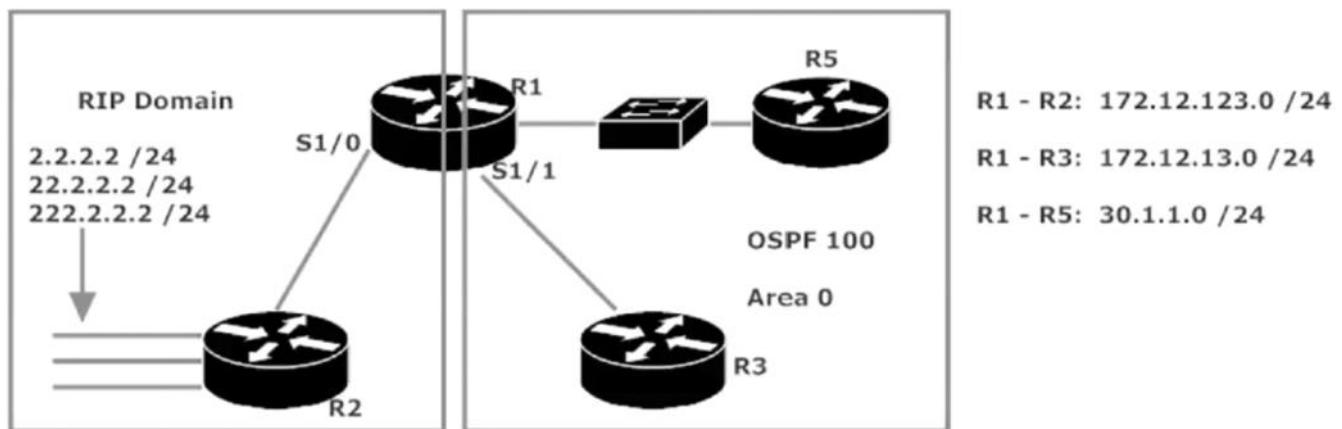
Sometime however we may need to just change metrics for an external route.

Route Redis 12: Writing and Verifying Route Maps

Saturday, March 31, 2018 10:45 AM

Route maps operate in a similar fashion to access-lists. Both route maps and ACLs arrive at a decision of "permit" or "deny". Route maps give us power over the packets beyond that simple "send" or "don't send"; we can actually change BGP route attributes with these babies, among other things.

Let's get to our next lab, which looks just a bit like our last lab! Note the change in the routing protocol on the right side of the network. R2 is now advertising an additional route.



3/31/2018 10:45 AM - Screen Clipping

R1 sees all three routes advertised by R2.

```
R1#show ip route rip
R  222.2.2.0/24 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
R  2.0.0.0/8 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
R  22.0.0.0/8 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
```

We have three routes, so why not three requirements for the lab?

2.0.0.0 /8: Double the default seed metric and set the route type to E1.

22.0.0.0 /8: Keep the default seed metric and set the route type to E1.

222.2.2.0 /24: Don't redistribute this route at all.

Just one more thing (*Columbo™*) All routes redistributed into OSPF in the future should keep the default seed metric and the default route type.

That ought to keep us busy! The first step, as always, is to identify each route or group of routes with an ACL.

3/31/2018 10:46 AM - Screen Clipping

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 1 permit 2.0.0.0 0.255.255.255
R1(config)#
R1(config)#access-list 22 permit 22.0.0.0 0.255.255.255
R1(config)#
R1(config)#access-list 44 permit 222.2.2.0 0.0.0.255
R1(config)#

```

3/31/2018 10:47 AM - Screen Clipping

Just use ACLs to identify and let the route map take the actions.

Now create the route map.

```
R1(config)#route-map RIP2OSPF ?
<0-65535> Sequence to insert to/delete from existing route-map entry
deny      Route map denies set operations
permit    Route map permits set operations
<cr>

R1(config)#route-map RIP2OSPF permit ?
<0-65535> Sequence to insert to/delete from existing route-map entry
<cr>

R1(config)#route-map RIP2OSPF permit 10
R1(config-route-map)#
Route Map configuration commands:
  continue      Continue on a different entry within the route-map
  default       Set a command to its defaults
  description   Route-map comment
  exit          Exit from route-map configuration mode
  help          Description of the interactive help system
  match         Match values from routing table
  no            Negate a command or set its defaults
  set           Set values in destination routing protocol

```

3/31/2018 10:48 AM - Screen Clipping

Select the match and set values

```
R1(config-route-map)#match ?
  as-path          Match BGP AS path list
  clns            CLNS information
  community       Match BGP community list
  extcommunity    Match BGP/VPN extended community list
  interface        Match first hop interface of route
  ip               IP specific information
  ipv6             IPv6 specific information
  length           Packet length
  local-preference Local preference for route
  metric           Match metric of route
  mpls-label       Match routes which have MPLS labels
  nlri             BGP NLRI type
  policy-list      Match IP policy list
  route-type       Match route-type of route
  source-protocol Match source-protocol of route
  tag              Match tag of route
```

3/31/2018 10:48 AM - Screen Clipping

```
R1(config-route-map)#match ip ?
  address          Match address of route or match packet
  next-hop         Match next-hop address of route
  route-source     Match advertising source address of route

R1(config-route-map)#match ip address ?
  <1-199>          IP access-list number
  <1300-2699>      IP access-list number (expanded range)
  WORD              IP access-list name
  prefix-list       Match entries of prefix-lists

R1(config-route-map)#match ip address _
```

3/31/2018 10:49 AM - Screen Clipping

You're matching against an ACL but first have to choose the IP, address, and then the UP access list number.

```

R1(config-route-map)#match ip address 1
R1(config-route-map)#set ?
  as-path          Prepend string for a BGP AS-path attribute
  automatic-tag   Automatically compute TAG value
  c1ns            OSI summary address
  comm-list       set BGP community list (for deletion)
  community       BGP community attribute
  dampening      Set BGP route flap dampening parameters
  default         Set default information
  extcommunity   BGP extended community attribute
  interface       Output interface
  ip              IP specific information
  ipv6           IPv6 specific information
  level          Where to import route
  local-preference BGP local preference path attribute
  metric          Metric value for destination routing protocol
  metric-type     Type of metric for destination routing protocol
  mpls-label      Set MPLS label for prefix
  nlri            BGP NLRI type
  origin          BGP origin code
  tag             Tag value for destination routing protocol
  traffic-index  BGP traffic classification number for accounting
  vrf             Define VRF name
  weight          BGP weight for routing table

```

3/31/2018 10:50 AM - Screen Clipping

IP 1 is the same as ACL 1

Lets double the seed metric per requirements.

```

R1(config-route-map)#set metric ?
  +/-<metric>    Add or subtract metric
  <0-4294967295> Metric value or Bandwidth in Kbits per second

R1(config-route-map)#set metric 40 ?
  +/-<delay>      Add or subtract delay
  <1-4294967295>  IGRP delay metric, in 10 microsecond units
  <cr>

R1(config-route-map)#set metric 40

```

3/31/2018 10:51 AM - Screen Clipping

OSPF is 20 by default so set it hire to 40. Now also choose the type 1/

```

R1(config-route-map)#set metric 40
R1(config-route-map)#set metric-type ?
  external  IS-IS external metric
  internal  IS-IS internal metric or Use IGP me
  type-1    OSPF external type 1 metric
  type-2    OSPF external type 2 metric

R1(config-route-map)#set metric-type type-1
R1(config-route-map)#

```

3/31/2018 10:51 AM - Screen Clipping

We've matched Acl for network 2 and now set the metric to 40 and the value to type 1 as requested.

Now do it for the next acl. Remember were just changing the type on this one, not the seed metric.

3/31/2018 10:52 AM - Screen Clipping

Now do it for the last one via the ACL

```
R1(config-route-map)#route-map RIP2OSPF ?
% Unrecognized command
R1(config-route-map)#route-map RIP2OSPF deny 30
R1(config-route-map)#match ip address 44
R1(config-route-map)#

```

3/31/2018 10:53 AM - Screen Clipping

One more thing we need to do is the catch all. It applies to all routes that don't specifically match the networks we used.

```
R1(config-route-map)#route-map RIP2OSPF permit 40  
R1(config-route-map)#{
```

3/31/2018 10:54 AM - Screen Clipping

Check your work

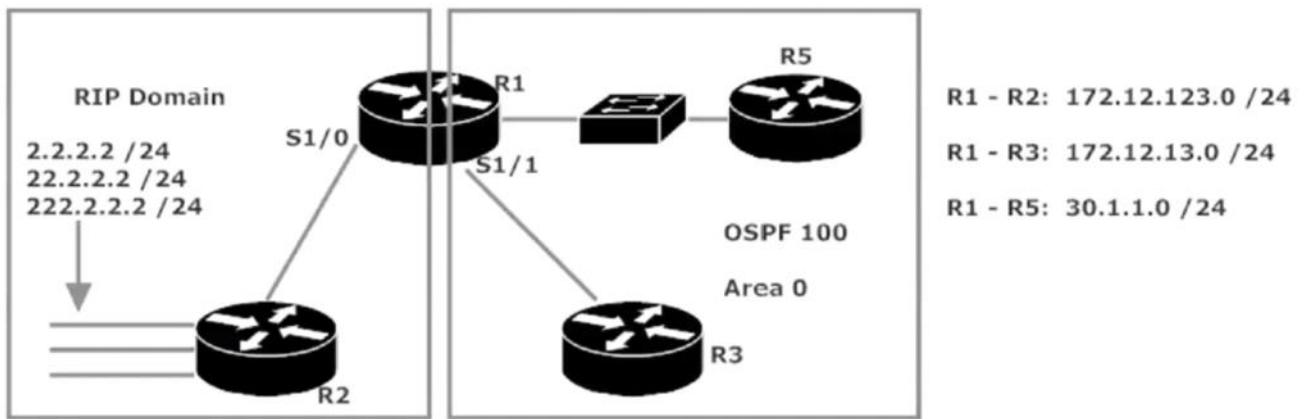
```
R1#SHOW
*Jun 16 15:31:09.306: %SYS-5-CONFIG_I: Configuration change
R1#show route-map
route-map RIP2OSPF, permit, sequence 10
Match clauses:
  ip address (access-lists): 1
Set clauses:
  metric 40
  metric-type type-1
Policy routing matches: 0 packets, 0 bytes
route-map RIP2OSPF, permit, sequence 20
Match clauses:
  ip address (access-lists): 22
Set clauses:
  metric-type type-1
Policy routing matches: 0 packets, 0 bytes
route-map RIP2OSPF, deny, sequence 30
Match clauses:
  ip address (access-lists): 44
Set clauses:
Policy routing matches: 0 packets, 0 bytes
route-map RIP2OSPF, permit, sequence 40
Match clauses:
Set clauses:
Policy routing matches: 0 packets, 0 bytes
R1#
```

3/31/2018 10:54 AM - Screen Clipping

Route Redis 13: Applying Route Maps During Redis

Saturday, March 31, 2018 10:55 AM

Let's get to our next lab, which looks just a bit like our last lab! Note the change in the routing protocol on the right side of the network. R2 is now advertising an additional route.



R1 sees all three routes advertised by R2.

```
R1#show ip route rip
R    222.2.2.0/24 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
R    2.0.0.0/8 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
R    22.0.0.0/8 [120/1] via 172.12.123.2, 00:00:07, Serial1/0
```

3/31/2018 10:55 AM - Screen Clipping

Lets do the redis without the route map to see the results.

```
R1#conf t
Enter configuration commands, one per
R1(config)#router ospf 1
R1(config-router)#redis rip subnets
R1(config-router)#redis conn subnets
R1(config-router)#^Z
R1#wr
Building configuration...
```

3/31/2018 10:56 AM - Screen Clipping

```

R3#show ip route ospf
Codes: L - local, C - connected, S - static,
       D - EIGRP, EX - EIGRP external, O - OSPF
       N1 - OSPF NSSA external type 1, N2 -
       E1 - OSPF external type 1, E2 - OSPF
       i - IS-IS, su - IS-IS summary, L1 - I
       ia - IS-IS inter area, * - candidate
       o - ODR, P - periodic downloaded stat
       + - replicated route, % - next hop ov

Gateway of last resort is not set

      2.0.0.0/24 is subnetted, 1 subnets
O E2      2.2.2.0 [110/20] via 172.12.13.1, 0
      22.0.0.0/24 is subnetted, 1 subnets
O E2      22.2.2.0 [110/20] via 172.12.13.1,
      30.0.0.0/24 is subnetted, 1 subnets
O          30.1.1.0 [110/65] via 172.12.13.1,
      172.12.0.0/16 is variably subnetted, 3
O E2      172.12.123.0/24 [110/20] via 172.12.
O E2      222.2.2.0/24 [110/20] via 172.12.13.1
R3#

```

3/31/2018 10:56 AM - Screen Clipping

We now see all the types and seed metric as normal. Our values changed to default.

```

R1#
R1#conf t
Enter configuration commands, one per line. End with
R1(config)#router ospf 1
R1(config-router)#no redis rip subnets
R1(config-router)#no redis rip
R1(config-router)#no redis conn subnets
R1(config-router)#no redis conn
R1(config-router)#^Z
R1#wr
Building configuration

```

3/31/2018 10:57 AM - Screen Clipping

Now lets see what happens when we reapply the route map.

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#redis conn subnets
R1(config-router)#redis rip subnets ?
  metric      Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redistributed routes
  route-map   Route map reference
  tag         Set tag for routes redistributed into OSPF
<cr>

R1(config-router)#redis rip subnets route-map ?
  WORD  Pointer to route-map entries

R1(config-router)#redis rip subnets route-map RIP2OSPF ?
  metric      Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redistributed routes
  tag         Set tag for routes redistributed into OSPF
<cr>

```

3/31/2018 10:58 AM - Screen Clipping

```

R1(config-router)#redis rip subnets route-map RIP2OSPF
R1(config-router)#^Z

```

3/31/2018 10:58 AM - Screen Clipping

And now we see the types have changes as well as the seed metric. They've been redis via route map.

```

R3#show ip route ospf
Codes: L - Local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
      ia - IS-IS inter area, * - candidate default, U - per-user static ro
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

```

Gateway of last resort is not set

```

      2.0.0.0/24 is subnetted, 1 subnets
0 E1    2.2.2.0 [110/104] via 172.12.13.1, 00:00:10, Serial0/1/0
      22.0.0.0/24 is subnetted, 1 subnets
0 E1    22.2.2.0 [110/84] via 172.12.13.1, 00:00:10, Serial0/1/0
      30.0.0.0/24 is subnetted, 1 subnets
0       30.1.1.0 [110/65] via 172.12.13.1, 00:52:59, Serial0/1/0
      172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
0 E2    172.12.123.0/24 [110/20] via 172.12.13.1, 00:00:30, Serial0/1/0
R3#

```

3/31/2018 10:58 AM - Screen Clipping

We know this for sure, because network 22 only had its type changed. Network 2 has doubled its seed and type is changed. Network 222 is gone because we wanted it filtered. Lets check out router 5.

```

R5#
R5#show ip route ospf
Codes: L - local, C - connected, S - static,
      D - EIGRP, EX - EIGRP external, O - OS
      N1 - OSPF NSSA external type 1, N2 - O
      E1 - OSPF external type 1, E2 - OSPF e
      i - IS-IS, su - IS-IS summary, L1 - IS
      ia - IS-IS inter area, * - candidate d
      o - ODR, P - periodic downloaded stati
Gateway of last resort is not set

      2.0.0.0/24 is subnetted, 1 subnets
O E1      2.2.2.0 [110/41] via 30.1.1.1, 00:01
      22.0.0.0/24 is subnetted, 1 subnets
O E1      22.2.2.0 [110/21] via 30.1.1.1, 00:0
      172.12.0.0/24 is subnetted, 2 subnets
O          172.12.13.0 [110/782] via 30.1.1.1,
O E2      172.12.123.0 [110/20] via 30.1.1.1,
R5#

```

3/31/2018 11:00 AM - Screen Clipping

Same results. O E1 code. Metrics different because connections are different. 222 is gone as well.

Great stuff! Route-maps are a powerful tool for controlling redistribution and changing route attributes, BGP or otherwise.

Before we move on, we need to test the 4th clause of our route-map. Let's add a route not specifically named by the first three lines and see what happens.

3/31/2018 11:01 AM - Screen Clipping

```

R2#conf t
Enter configuration commands, one per line. End with a carriage return alone on a line.
R2(config)#int loopback55
R2(config-if)#ip address
*Jul 9 23:35:39.623: %LINEPROTO-5-UPDOWN: Line protocol
, changed state to up
R2(config-if)#ip address 55.5.5.5 255.255.255.0
R2(config-if)#router rip
R2(config-router)#network 55.0.0.0
R2(config-router)#^Z
R2#clar
*Jul 9 23:35:52.243: %SYS-5-CONFIG_I: Configured f
R2#clear ip route *
R2#^Z
R2#wr
Building configuration...

```

3/31/2018 11:01 AM - Screen Clipping

```
R1#show ip route rip
R    222.2.2.0/24 [120/1] via 172.12.123.2
      2.0.0.0/24 is subnetted, 1 subnets
R        2.2.2.0 [120/1] via 172.12.123.2,
          55.0.0.0/24 is subnetted, 1 subnets
R            55.5.5.0 [120/1] via 172.12.123.2,
              22.0.0.0/24 is subnetted, 1 subnets
R                22.2.2.0 [120/1] via 172.12.123.2,
R1#
```

3/31/2018 11:02 AM - Screen Clipping

This should have been left alone. Check 3 and 5.

```
Gateway of last resort is not set

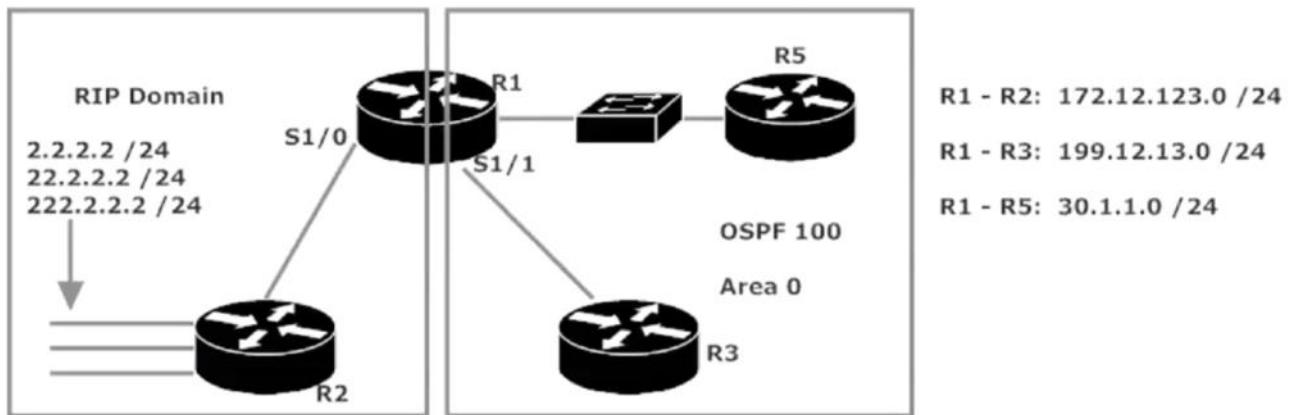
      2.0.0.0/24 is subnetted, 1 subnets
O E1    2.2.2.0 [110/104] via 172.12.13
      22.0.0.0/24 is subnetted, 1 subnet
O E1    22.2.2.0 [110/84] via 172.12.13
      30.0.0.0/24 is subnetted, 1 subnet
O       30.1.1.0 [110/65] via 172.12.13
      55.0.0.0/24 is subnetted, 1 subnet
O E2    55.5.5.0 [110/20] via 172.12.13
      172.12.0.0/16 is variably subnette
O E2    172.12.123.0/24 [110/20] via 17
R3#
```

Perfect. In its default code, seed.

Route Redis 14: Route Maps and 2 way Route Redis

Saturday, March 31, 2018 11:03 AM

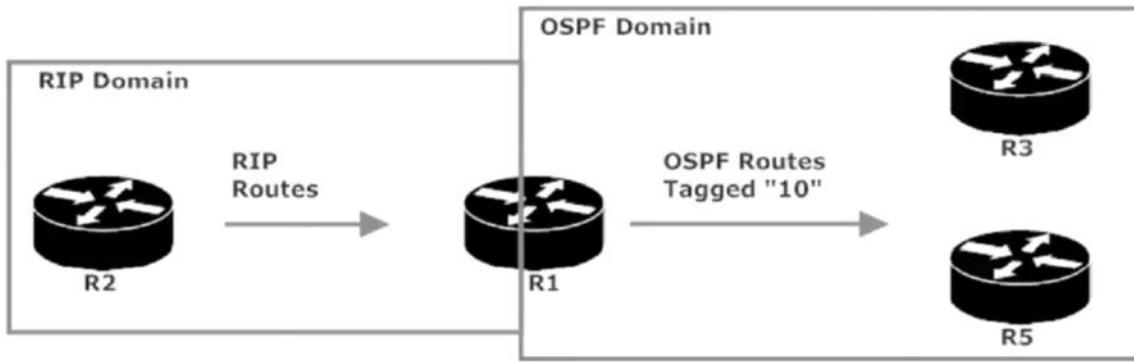
Let's use the previous lab topology for a demo. Everything except the basic RIP and OSPF configurations from the previous lab has been removed, and there's a different address on the R1 – R3 link.



In the previous lab, we configured one-way route redistribution (RIP into OSPF, namely). Let's say we're getting ready to configure two-way redistribution this time, and we want to make absolutely sure routes redistributed into OSPF cannot be brought back into the RIP domains, and vice versa. That includes the possibility of additional border routers being added later.

3/31/2018 11:03 AM - Screen Clipping

What we want to do is to make sure that we don't take routes from one protocol to another. We don't want routes on 2 to be redis into OSPF and then somehow send them back over to a RIP domain. As networks grow, routing loops are more common. Multiple points of 2 redis among 3 routers, all doing route redis can be aided by tagging. But that's CCIE level stuff.



The resulting OSPF table on R3:

```
R3#show ip route ospf

O E2  2.0.0.0/8 [110/20] via 172.12.13.1, 00:00:31, Serial0/1/0
O E2  22.0.0.0/8 [110/20] via 172.12.13.1, 00:00:31, Serial0/1/0
      30.0.0.0/24 is subnetted, 1 subnets
O       30.1.1.0 [110/65] via 172.12.13.1, 00:00:26, Serial0/1/0
      172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
O E2    172.12.123.0/24 [110/20] via 172.12.13.1, 00:00:31, Serial0/1/0
O E2    222.2.2.0/24 [110/20] via 172.12.13.1, 00:00:31, Serial0/1/0
```

3/31/2018 11:10 AM - Screen Clipping

Write the route map first.

```
R1(config)#route-map RIP2OSPF permit 10
R1(config-route-map)#set tag 10
R1(config-route-map)#
R1(config-route-map)#router ospf 1
R1(config-router)#redis rip ?
  metric      Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redis
  route-map   Route map reference
  subnets     Consider subnets for redistribution into
  tag         Set tag for routes redistributed into OSP
<cr>

R1(config-router)#redis rip route-map ?
  WORD  Pointer to route-map entries

R1(config-router)#redis rip route-map RIP2OSPF ?
  metric      Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redis
  subnets     Consider subnets for redistribution into
  tag         Set tag for routes redistributed into OSP
<cr>

R1(config-router)#redis rip route-map RIP2OSPF subnets
```

3/31/2018 11:12 AM - Screen Clipping

```
R1(config-router)#redis rip route-map RIP2OSPF subnets
R1(config-router)#redis conn route-map RIP2OSPF subnets
R1(config-router)#^Z
R1#
```

3/31/2018 11:12 AM - Screen Clipping

You're now redis ospf and connected routes.

```
R3#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA -
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA
       E1 - OSPF external type 1, E2 - OSPF external
       i - IS-IS, su - IS-IS summary, L1 - IS-IS leve
       ia - IS-IS inter area, * - candidate default,
       o - ODR, P - periodic downloaded static route,
       + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
      2.0.0.0/24 is subnetted, 1 subnets
O E2    2.2.2.0 [110/20] via 199.12.13.1, 00:00:17,
      22.0.0.0/24 is subnetted, 1 subnets
O E2    22.2.2.0 [110/20] via 199.12.13.1, 00:00:17,
      30.0.0.0/24 is subnetted, 1 subnets
O        30.1.1.0 [110/65] via 199.12.13.1, 00:07:48,
      172.12.0.0/24 is subnetted, 1 subnets
O E2    172.12.123.0 [110/20] via 199.12.13.1, 00:00:00
O E2    222.2.2.0/24 [110/20] via 199.12.13.1, 00:00:17
R3#
```

3/31/2018 11:13 AM - Screen Clipping

```
R3#show ip route 2.0.0.0
Routing entry for 2.0.0.0/24, 1 known subnets
O E2    2.2.2.0 [110/20] via 199.12.13.1, 00:00:55, Serial0/1/0
R3#show ip route 2.2.2.2
Routing entry for 2.2.2.0/24
  Known via "ospf 1", distance 110, metric 20
  Tag 10, type extern 2, forward metric 64
  Last update from 199.12.13.1 on Serial0/1/0, 00:00:58 ago
  Routing Descriptor Blocks:
    * 199.12.13.1, from 1.1.1.1, 00:00:58 ago, via Serial0/1/0
      Route metric is 20, traffic share count is 1
      Route tag 10
```

3/31/2018 11:13 AM - Screen Clipping

You can now see the tag of 10.

Check the others.

```

R3#show ip route 22.2.2.2
Routing entry for 22.2.2.0/24
  Known via "ospf 1", distance 110, metric 6
  Tag 10, type extern 2, forward metric 6
  Last update from 199.12.13.1 on Serial0
  Routing Descriptor Blocks:
    * 199.12.13.1, from 1.1.1.1, 00:01:18 ago
      Route metric is 20, traffic share count 1
      Route tag 10
R3#show ip route 222.2.2.2
Routing entry for 222.2.2.0/24
  Known via "ospf 1", distance 110, metric 6
  Tag 10, type extern 2, forward metric 6
  Last update from 199.12.13.1 on Serial0
  Routing Descriptor Blocks:
    * 199.12.13.1, from 1.1.1.1, 00:01:23 ago
      Route metric is 20, traffic share count 1
      Route tag 10

```

3/31/2018 11:14 AM - Screen Clipping

Tagging has been successful so far.

Now we want to make sure the routes with a 10 tag can't be redis back from OSPF. So lets create another tag OSPF2RIP

```

R1(config)#route-map OSPF2RIP deny 10
R1(config-route-map)#match ?
  as-path          Match BGP AS path list
  clns            CLNS information
  community       Match BGP community list
  extcommunity   Match BGP/VPN extended community
  interface       Match first hop interface of route
  ip              IP specific information
  ipv6           IPv6 specific information
  length          Packet length
  local-preference Local preference for route
  metric          Match metric of route
  mpls-label      Match routes which have MPLS label
  nlri            BGP NLRI type
  policy-list     Match IP policy list
  route-type      Match route-type of route
  source-protocol Match source-protocol of route
  tag             Match tag of route

```

3/31/2018 11:15 AM - Screen Clipping

First Deny

```

R1(config-route-map)#match tag 10
R1(config-route-map)#route-map OSPF2RIP permit 20
R1(config-route-map)#set tag 20
R1(config-route-map)#

```

3/31/2018 11:15 AM - Screen Clipping

Then Permit. Stop the routes tagged 10 from going back to RIP.

```

R1(config-router)#redis ospf ?
<1-65535> Process ID

R1(config-router)#redis ospf 1 ?
  match    Redistribution of OSPF routes
  metric   Metric for redistributed routes
  route-map Route map reference
  vrf      VPN Routing/Forwarding Instance
<cr>

R1(config-router)#redis ospf 1 route-map ?
  WORD  Pointer to route-map entries

R1(config-router)#redis ospf 1 route-map OSPF2RIP ?
  match    Redistribution of OSPF routes
  metric   Metric for redistributed routes
<cr>

R1(config-router)#redis ospf 1 route-map OSPF2RIP metric 2

```

3/31/2018 11:16 AM - Screen Clipping

```

(config-router)#redis ospf 1 route-map OSPF2RIP metric 2
(config-router)#redis conn route-map OSPF2RIP metric 2
"OSPF2RIP" used as redistribute connected into rip route-map, tag match not
sorted
(config-router)#

```

3/31/2018 11:16 AM - Screen Clipping

This message is when you're going from connect into RIP..

```

R1#
*Jul 14 15:00:41.849: %SYS-5-CONFIG_I: Configu
R1#show route-map
route-map RIP2OSPF, permit, sequence 10
  Match clauses:
  Set clauses:
    tag 10
  Policy routing matches: 0 packets, 0 bytes
route-map OSPF2RIP, deny, sequence 10
  Match clauses:
    tag 10
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map OSPF2RIP, permit, sequence 20
  Match clauses:
  Set clauses:
    tag 20
  Policy routing matches: 0 packets, 0 bytes

```

3/31/2018 11:17 AM - Screen Clipping

Lets take a look at RIP2OSPF and OSPF2RIP.

```
R1#show route-map RIP2OSPF
route-map RIP2OSPF, permit, sequence 10
  Match clauses:
    Set clauses:
      tag 10
  Policy routing matches: 0 packets, 0 bytes
R1#
```

3/31/2018 11:18 AM - Screen Clipping

How do we deny OSPF route with a tag of 20?

You need to make sure you give it a sequence number less than 10. Clauses work like Acl as they're numerical so we need a seqn num of less than 10.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#route-map RIP2OSPF ?
<0-65535> Sequence to insert to/delete from existing route-map entry
deny      Route map denies set operations
permit    Route map permits set operations
<cr>

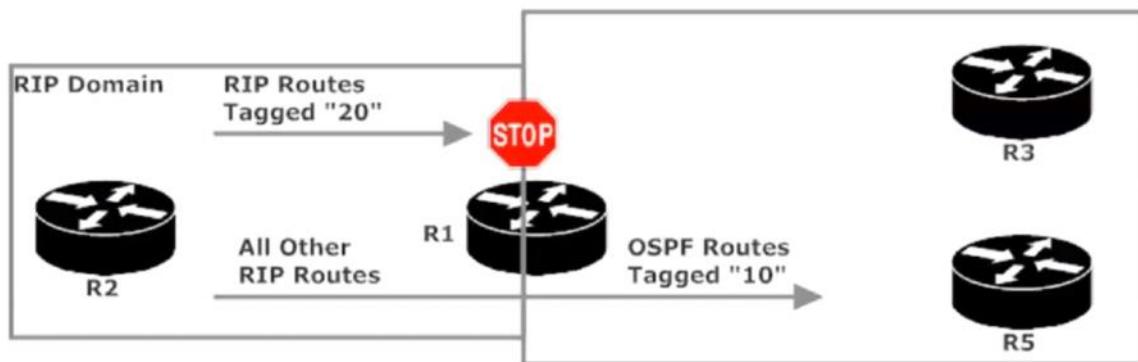
R1(config)#route-map RIP2OSPF deny ?
<0-65535> Sequence to insert to/delete from existing route-map entry
<cr>

R1(config)#route-map RIP2OSPF deny 5 ?
<cr>

R1(config)#route-map RIP2OSPF deny 5
R1(config-route-map)#match tag 20
% "RIP2OSPF" used as redistribute connected into ospf route-map, tag match
supported
R1(config-route-map)#
udemy
```

3/31/2018 11:20 AM - Screen Clipping

Now RIP2OSPF has a deny clause for packets that have a tag of 5. We're stopping routes from being redistributed into one domain and back. This prevents routing loops.



Just as with ACLs, the order of the lines in a route-map is vital. Sequence numbers make it easy to keep your lines in line!

Route Redis 17: Policy Routing Theory and Application

We'll wrap up route manipulation with Policy routing or policy based routing. Well use route maps to apply certain values to certain traffic.

Policy routing doesn't affect the destination of the packet, but it can affect the path the traffic takes to get there, including the next-hop IP address. (Spoiler Alert!)

Policy routing can forward traffic based on the source or destination IP address with the use of an extended ACL. (Spoiler #2 Alert!)

Applying policy routing on an interface affects only packets arriving on that interface.

Applying policy routing locally applies the policy to packets generated on that router.

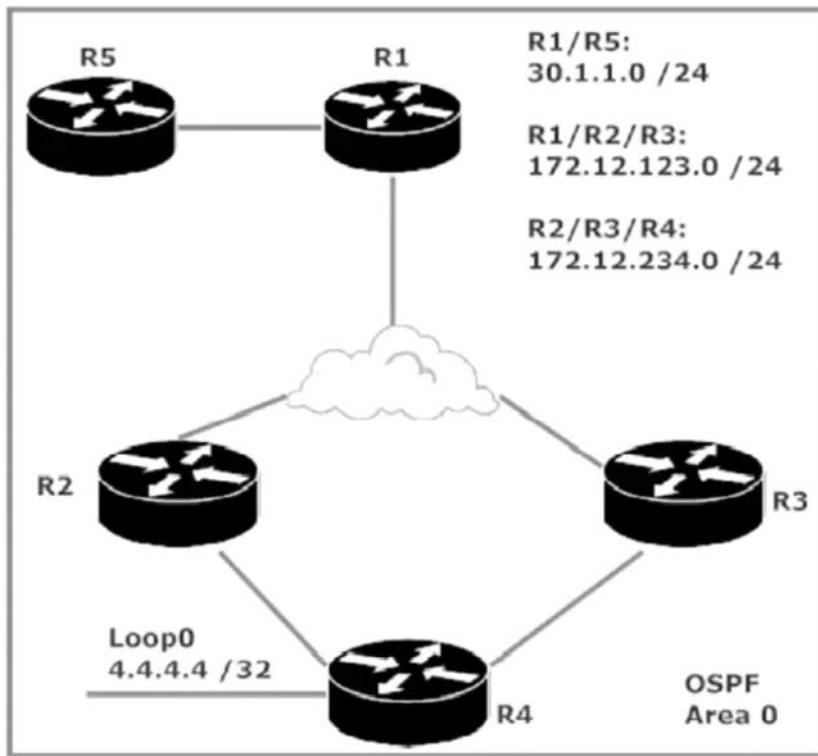
If a packet doesn't match any of the specific criteria in a route map, or does match a line that has an explicit *deny* statement, the data is sent to the routing process and is processed normally.

3/31/2018 11:32 AM - Screen Clipping

If you don't want to route packets that have no match for a route-map clause in policy routing, the *set* command must be used to send those packets to the null0 interface. Naturally, this *set* command should be the final set command in the route map.

Let's see these rules in action with this lab! We'll pay special attention to R4's loopback throughout the next two labs.

3/31/2018 11:35 AM - Screen Clipping



3/31/2018 11:36 AM - Screen Clipping

Traceroute reveals the path packets take from R5 to R4's loopback.

3/31/2018 11:37 AM - Screen Clipping

```
R5#traceroute 4.4.4.4
Type escape sequence to abort.
Tracing the route to 4.4.4.4

 1 30.1.1.1 0 msec 0 msec 0 msec
 2 172.12.123.2 32 msec 32 msec 32 msec
 3 172.12.234.4 36 msec * 32 msec
R5#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4,
!!!!!
Success rate is 100 percent (5/5), round-t
R5#
```

3/31/2018 11:37 AM - Screen Clipping

Lets make the enxt hop be router 3 instead of 2. Where to configure? Policy routing is always configured on inbound traffic. So lets start with router 1.

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#access-list 5 permit host 30.1.1.5  
R1(config)#  
R1(config)#
```

3/31/2018 11:39 AM - Screen Clipping

Identify the source of the traffic.

```
R1(config)#route-map NEXTHOP ?
<0-65535> Sequence to insert to/delete from existing
deny      Route map denies set operations
permit    Route map permits set operations
<cr>

R1(config)#route-map NEXTHOP permit ?
<0-65535> Sequence to insert to/delete from existing
<cr>

R1(config)#route-map NEXTHOP permit 10
R1(config-route-map)#
```

3/31/2018 11:39 AM - Screen Clipping

Create the routemap

Then **set** the enxt hop traffic to match the sequence number and IP address

```

R1(config-route-map)#match ip address 5
R1(config-route-map)#set ?
  as-path          Prepend string for a BGP AS-path
  automatic-tag   Automatically compute TAG value
  c1ns            OSI summary address
  comm-list       set BGP community list (for communities)
  community       BGP community attribute
  dampening      Set BGP route flap dampening
  default         Set default information
  extcommunity   BGP extended community attribute
  interface       Output interface
  ip              IP specific information
  ipv6           IPv6 specific information
  level          Where to import route
  local-preference BGP local preference path attribute
  metric          Metric value for destination
  metric-type    Type of metric for destination
  mpls-label     Set MPLS label for prefix
  nlri           BGP NLRI type
  origin          BGP origin code
  tag             Tag value for destination route
  traffic-index  BGP traffic classification number
  vrf             Define VRF name
  weight          BGP weight for routing table
--More--

```

3/31/2018 11:40 AM - Screen Clipping

```

R1(config-route-map)#set ip next-hop 172.12.123.3
R1(config-route-map)#

```

3/31/2018 11:41 AM - Screen Clipping

Apply policy to inbound interface.

```

R1(config-route-map)#int fast 0/0
R1(config-if)#ip policy ?
  route-map Policy route map

R1(config-if)#ip policy route-map ?
  WORD  Route map name

R1(config-if)#ip policy route-map NEXTHOP ?
  <cr>

R1(config-if)#ip policy route-map NEXTHOP
R1(config-if)#

```

3/31/2018 11:43 AM - Screen Clipping

Now you have your [ACL](#), [ROUTEMAP](#), [IP Address](#) and [Interface](#).

Test router 5.

```
R5#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4
!!!!!
Success rate is 100 percent (5/5), round-
R5#traceroute 4.4.4.4

Type escape sequence to abort.
Tracing the route to 4.4.4.4

 1 30.1.1.1 0 msec 4 msec 0 msec
 2 172.12.123.3 32 msec 36 msec 32 msec
 3 172.12.234.4 32 msec * 32 msec
R5#
```

3/31/2018 11:43 AM - Screen Clipping

You can now see its going to router 3 instead of 2. However, now all of our traffic is going to policy router on router 5 because we applied it locally.

```
3 172.12.234.4 52 msec * 52 msec
R5#traceroute 172.12.123.2

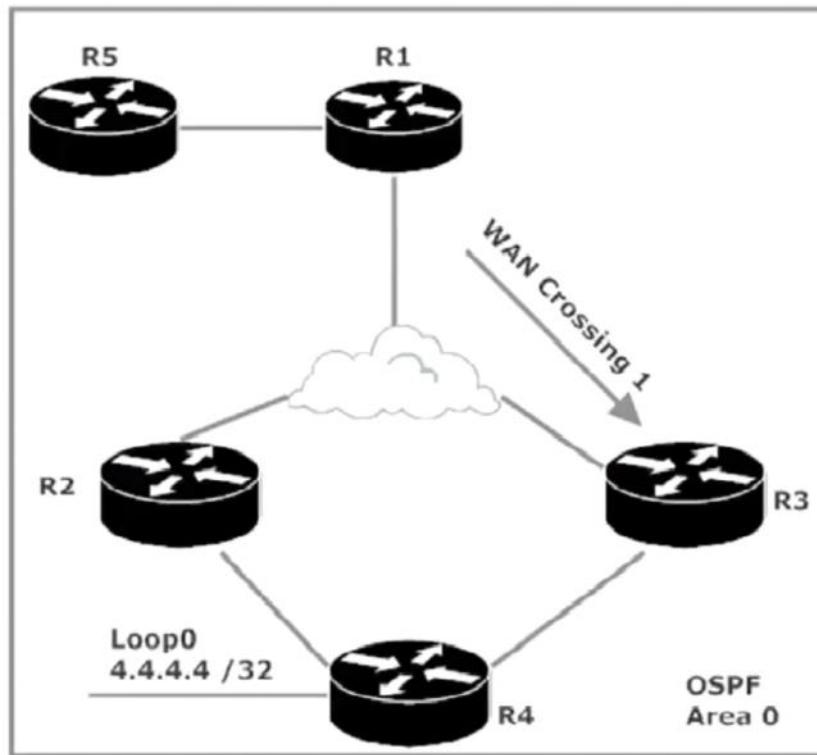
Type escape sequence to abort.
Tracing the route to 172.12.123.2

 1 30.1.1.1 0 msec 4 msec 0 msec
 2 172.12.123.3 32 msec 36 msec 32 msec
 3 172.12.123.1 28 msec 24 msec 24 msec
 4 172.12.123.2 56 msec * 52 msec
R5#
```

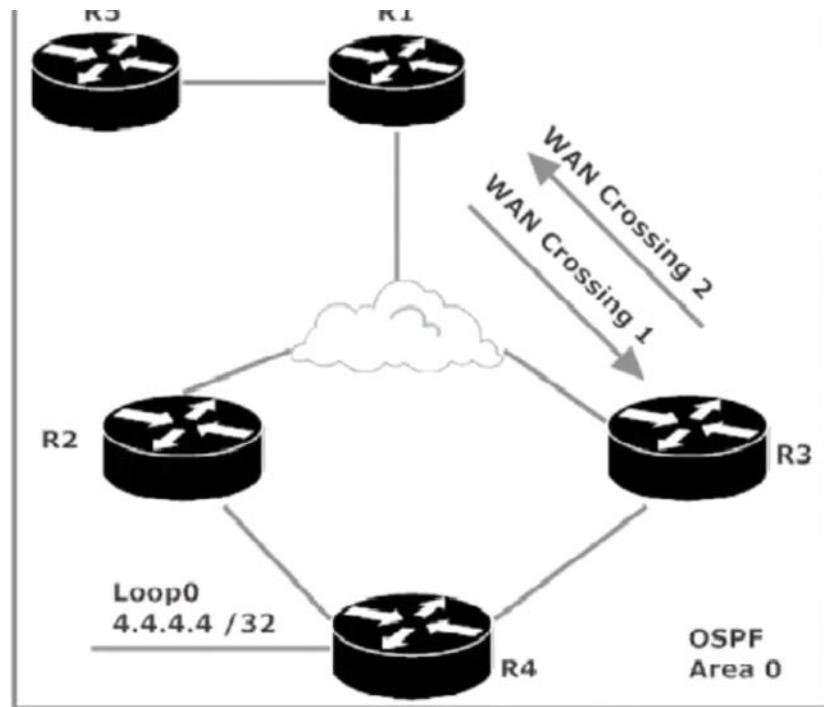
3/31/2018 11:44 AM - Screen Clipping

Now any traffic from 5 will first go to 3. Even if it could go straight to router 2. This isn't efficient. This is **Suboptimal routing**. Not a loop.

Pings from R5 to 172.12.123.2 are actually crossing the WAN three times! First, those pings arrive on R1 and policy routed to 172.12.123.3.

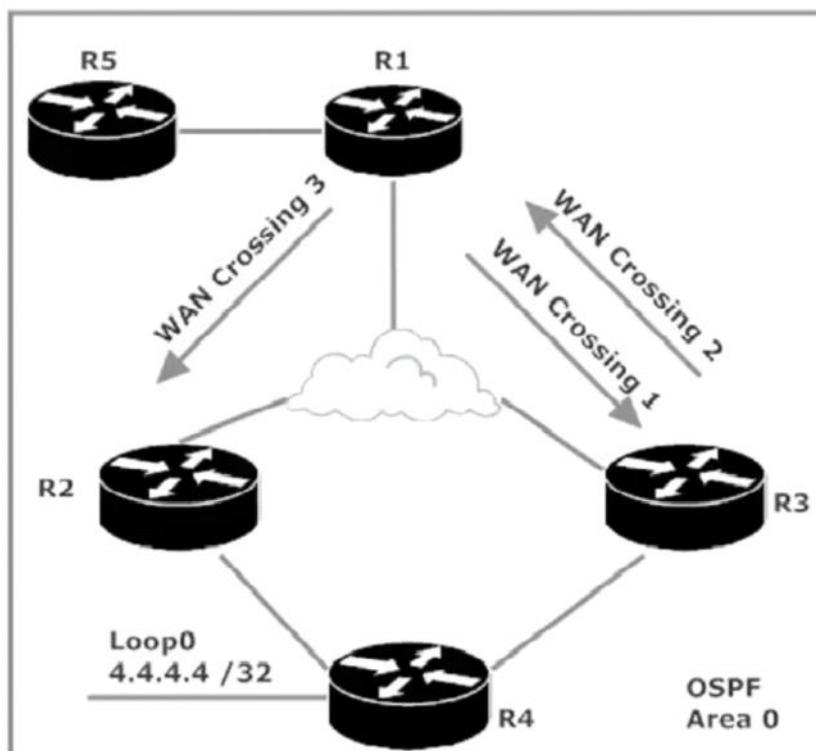


3/31/2018 11:45 AM - Screen Clipping



When R1 gets those packets, they no longer have the source IP address that matches the ACL, so the packets are routed normally to R2. This is suboptimal routing, and it's a situation that calls for policy routing that uses an extended ACL rather than a standard one.

3/31/2018 11:46 AM - Screen Clipping



This is why we typically use extended ACLs instead of standard so we can be more granular with our values.

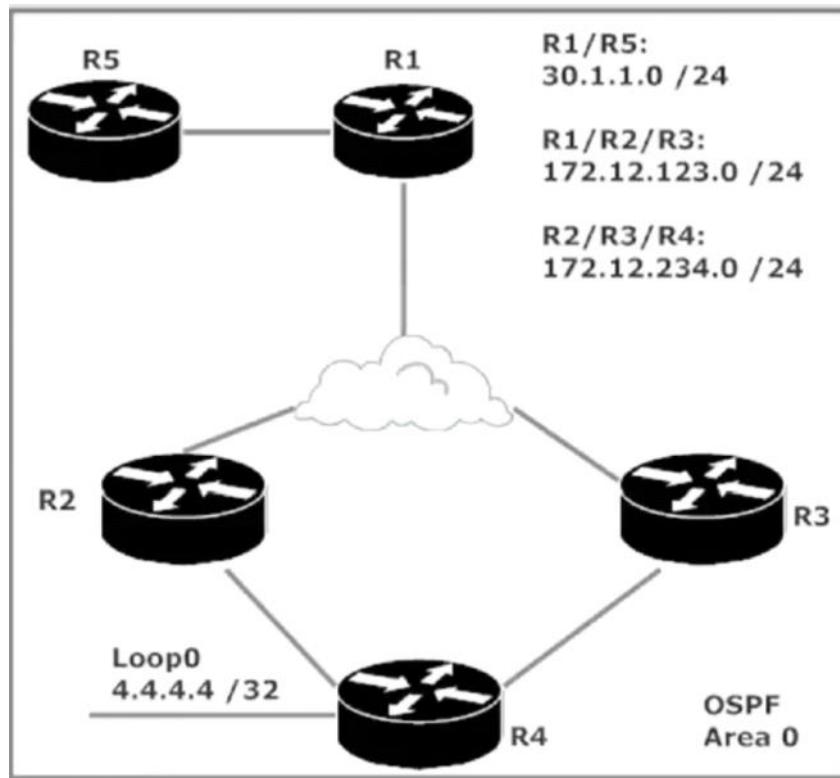
Route Redis 18: More Policy Routing

Saturday, March 31, 2018 11:47 AM

```
R1(config-if)#  
R1(config-if)#  
R1(config-if)#exit  
R1(config)#no route-map NEXTHOP  
R1(config)#no access-list 5  
R1(config)#int fast 0/0  
R1(config-if)#no ip policy route-map NEXTHOP  
R1(config-if)#^Z  
R1#wr  
Building configuration...
```

3/31/2018 11:47 AM - Screen Clipping

We've taken off configs from previous lesson.



3/31/2018 11:47 AM - Screen Clipping

We want traffic to be policy routed on router 1 and go to 3 if traffic is sourced from 30.1.1.5 and is destined for loopback4.

If router 5 goes to router 2 we want it to go 5 1 2. Unless it's coming from 4.

Create EXT ACL.

```
Enter configuration commands, one per line. End with Ctrl/Z.
R1(config)#access-list 105 permit ip host ?
    Hostname or A.B.C.D  Source address

R1(config)#access-list 105 permit ip host 30.1.1.5 ?
    A.B.C.D  Destination address
    any      Any destination host
    host     A single destination host

R1(config)#access-list 105 permit ip host 30.1.1.5 host ?
    Hostname or A.B.C.D  Destination address

R1(config)#access-list 105 permit ip host 30.1.1.5 host 4.4.4.4
```

3/31/2018 11:49 AM - Screen Clipping

30.1.1.5--->>4.4.4.4

ROUTEMAP

```
R1(config)#access-list 105 permit ip host 30.1.1.5 host 4.4.4.4
R1(config)#
R1(config)#
R1(config)#route-map NEXTHOP permit 10
R1(config-route-map)#match ip address 105
R1(config-route-map)#set ip next-hop 172.12.123.3
R1(config-route-map)#
R1#
```

3/31/2018 11:49 AM - Screen Clipping

Traffic that doesn't match 105 will be normally routed.

POLICY

```
R1(config-route-map)#
R1(config-route-map)#int fast 0/0
R1(config-if)#ip policy route-map NEXTHOP
R1(config-if)#
R1(config-if)#^Z
R1#
```

3/31/2018 11:50 AM - Screen Clipping

Go to router 5 to check.

```
R5#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, Success rate is 100%
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R5#ping 172.12.123.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.123.2, Success rate is 100%
!!!!!
```

3/31/2018 11:50 AM - Screen Clipping

```
R5#traceroute 4.4.4.4
Type escape sequence to abort.
Tracing the route to 4.4.4.4
 1 30.1.1.1 0 msec 4 msec 0 msec
 2 172.12.123.3 32 msec 36 msec 32 msec
 3 172.12.234.4 32 msec * 32 msec
R5#
```

3/31/2018 11:51 AM - Screen Clipping

Good on this one.

```
R5#traceroute 172.12.123.2
Type escape sequence to abort.
Tracing the route to 172.12.123.2
 1 30.1.1.1 4 msec 0 msec 0 msec
 2 172.12.123.2 32 msec * 32 msec
R5#
```

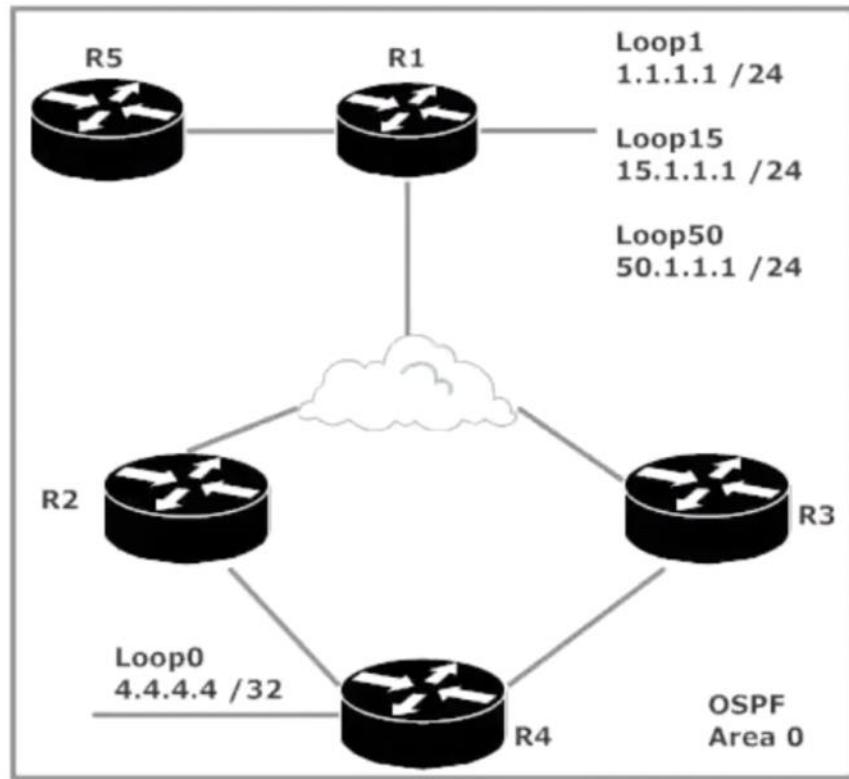
3/31/2018 11:51 AM - Screen Clipping

Good on this one too.

Route Redis 19: Local Policy Routing

Saturday, March 31, 2018 11:51 AM

Once in a while, you might just want to policy route packets that originate on the local router. That's what we'll do in our next lab! I've added three loopbacks to R1 and added them to OSPF Area 0.



3/31/2018 11:52 AM - Screen Clipping

All previous configs still in place.

```
R1#traceroute
Protocol [ip]:
Target IP address: 4.4.4.4
Source address: 1.1.1.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 4.4.4.4

 1 172.12.123.2 32 msec 32 msec 32 msec
 2 172.12.234.4 32 msec * 32 msec
R1#
```

3/31/2018 11:53 AM - Screen Clipping

```
R1#traceroute 4.4.4.4 source ?
A.B.C.D          Source address
Async            Async interface
BVI              Bridge-Group Virtual Interface
CDMA-Ix          CDMA Ix interface
CTunnel           CTunnel interface
Dialer            Dialer interface
FastEthernet      FastEthernet IEEE 802.3
Lex               Lex interface
Loopback          Loopback interface
MFR               Multilink Frame Relay bundle interface
Multilink         Multilink-group interface
Null              Null interface
Port-channel     Ethernet Channel of interfaces
Serial            Serial
Tunnel            Tunnel interface
Vif               PGM Multicast Host interface
Virtual-PPP       Virtual PPP interface
Virtual-Template Virtual Template interface
Virtual-TokenRing Virtual TokenRing
```

3/31/2018 11:54 AM - Screen Clipping

```
R1#traceroute 4.4.4.4 source 15.1.1.1
Type escape sequence to abort.
Tracing the route to 4.4.4.4
 1 172.12.123.2 32 msec 32 msec 32 msec
 2 172.12.234.4 36 msec * 32 msec
```

3/31/2018 11:54 AM - Screen Clipping

Ere just checking the paths being sued.

Lets have the first 2 loopbacks use 123.3 as the next hop if the traffic is going to 4.4.4.4. Loopback 50 should continue to use 123.2 for nh on all traffic now and in the future. So we need a two line acl.

ACL

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 101 permit ip host 1.1.1.1 host 4.4.4.4
R1(config)#access-list 101 permit ip host 15.1.1.1 host 4.4.4.4
R1(config)#
R1(config)
```

3/31/2018 11:56 AM - Screen Clipping

ROUTEMAP

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 101 permit ip host 1.1.1.1 host 4.4.4.4
R1(config)#access-list 101 permit ip host 15.1.1.1 host 4.4.4.4
R1(config)#
R1(config)#
R1(config)#route-map NEXTHOP_R3 permit 10
R1(config-route-map)#match i padd 101
          ^
% Invalid input detected at '^' marker.

R1(config-route-map)#match ip address 101
R1(config-route-map)#set ip next-hop 172.12.123.3
R1(config-route-map)#
R1(config)
```

3/31/2018 11:56 AM - Screen Clipping

Always route traffic as it comes in. What interface do we apply this to? We should be setting it on router 1. However, we can't use regular policy routing. We have to use local.

```
R1(config)#ip local ?
  policy  Enable policy routing
  pool    IP Local address pool lists

R1(config)#ip local policy ?
  route-map Policy route map

R1(config)#ip local policy
```

3/31/2018 11:58 AM - Screen Clipping

```
R1(config)#ip local policy route-map ?
  WORD  Route map name

R1(config)#ip local policy route-map NEXTHOP_R3
R1(config)#
R1(config)#^Z
R1#
```

3/31/2018 11:58 AM - Screen Clipping

```
R1#show route-map
route-map NEXTHOP, permit, sequence 10
  Match clauses:
    ip address (access-lists): 105
  Set clauses:
    ip next-hop 172.12.123.3
  Policy routing matches: 11 packets, 930 bytes
route-map NEXTHOP_R3, permit, sequence 10
  Match clauses:
    ip address (access-lists): 101
  Set clauses:
    ip next-hop 172.12.123.3
  Policy routing matches: 12 packets, 704 bytes
R1#
```

3/31/2018 12:00 PM - Screen Clipping

VPN 1: The Dreaded but Necessary Theory

Monday, April 2, 2018 8:17 AM

VPN and IPsec

It's not enough to have remote communications. We need VPN's to connect from home and we want it to be private. We do this by securing the shared channel and treating it as a private network. This is a tunnel, because the VPN is tunneling to the interface. We can make security changes to the Tunnel without messing with the interface or connections.

VPNs bring data origin authentication, encryption, and data integrity to the table. Data origin authentication allows the receiver to guarantee the source of the packet...



4/2/2018 8:20 AM - Screen Clipping

Guarantees source of packet and encryption.



... encryption makes the contents of packets unreadable during transmission, making intercepted packets useless to the interceptor ...



... and integrity is the receiver's ability to ensure the data was not tampered with during transmission.

One protocol is **GRE Generic Routing Encapsulation**. GRE doesn't have an encryption stream. This security hole is corrected by Ipsec. It offers encryption and authentications.

Authentication Header (AH), which defines a method for authentication and securing data

Encapsulating Security Payload (ESP), which defines a method for authenticating, security, and encrypting data

Internet Key Exchange (IKE), which negotiates the security parameters and authentication keys

The IPSec Packet Format



The IPSec Packet Format



Defined in RFC 2402, Authentication Header (AH) offers solid security, with data origin authentication and optional anti-replay protection. The drawback with AH is that the authentication it provides for the IP header is not complete. Some of the IP fields cannot be correctly predicted by the receiver, since some may change during transmission. AH will protect the IP packet's payload.

In short, AH offers data origin authentication, data integrity, and optional anti-replay protection. AH does not offer data confidentiality.

The Encapsulating Security Payload (ESP) does just that, with an ESP Header and Trailer encapsulating the data. ESP offers data origin authentication, anti-replay protection, and data confidentiality.

When comparing AH and ESP, you may have flashbacks to comparing TCP and UDP. TCP offers a ton of features that UDP doesn't, so why does any protocol or service use UDP instead of TCP? Overhead, of course! It's the same with ESP and AH. ESP is much more processor-intensive than AH, and ESP requires strong cryptography, which isn't available everywhere and isn't allowed everywhere. AH has no such requirement. Still, the full encapsulation of data makes ESP a wise choice whenever it's feasible.

Both ESP and AH can be run in either tunnel mode or transport mode. In tunnel mode, the entire IPSec process is transparent to the end hosts, and specialized IPSec gateway devices handle that part of the load. Tunnel mode encrypts the entire IP packet, and then that encrypted packet is placed into another IP packet. That encapsulating packet will have the IP addresses configured on the tunnel endpoints, and it's those tunnel IP addresses that will be used to route the packet.

Transport mode encrypts the IP packet's payload, but the IPSec header is inserted directly after the IP header. As a result, there is no protection for the original IP address, it's the original IP address that's used for routing, and only data from the Transport layer up is protected by IPSec.

There are five main steps in creating an IPSec VPN. We start with *process initialization* via the receipt of interesting traffic, which is a really fancy way of saying "The VPN creation process starts because traffic we said could do so did so". That interesting traffic is defined by a crypto access-list. Next up is *IKE Phase 1*, where the IKE SA (Internet Key Exchange Security Association) is negotiated.

Next in line is *IKE Phase 2*, where the IPSec SA is negotiated. When this is completed, the IPSec tunnel (also called the IKE Phase 2 tunnel) build is complete.

The self-explanatory *data transfer* now takes place, and when that transfer is complete, the tunnel is torn down. This *tunnel termination* can be configured to happen after a certain number of bytes have passed through the tunnel or after the data transfer has been idle for a given period of time. If traffic is flowing through the tunnel at the same time the tunnel's supposed to come down, a new Security Association can be agreed upon while the existing one is still in place.

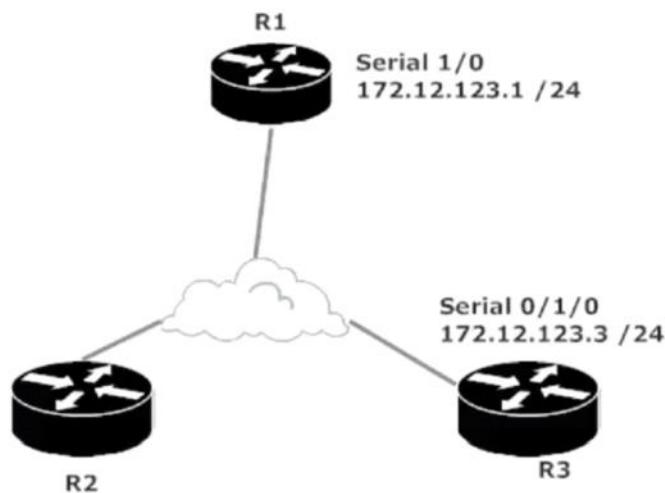
STEPS:

Process initialization - Starts the creation of the vpn
IKE Phase 1 - IKE Internet key exchange security associations
IKE Phase 2 - SA is negotiated. Build the bridge.
Data Transfer - Movement of bits
Tear down - tunnel is destroyed when connection is disconnected.

VPN 2: IKE Phase 1 in Action

Monday, April 2, 2018 8:29 AM

We'll build a VPN between R1 and R3. R2 is still in the picture, but not in the lab.



4/2/2018 8:30 AM - Screen Clipping

Creating The IKE Policy

First things first - be sure the Internet Security Association and Key Management Protocol (ISAKMP) is on via the *crypto isakmp enable* command. It should be on by default. *Should* be.

```
R1(config)#crypto isakmp enable
```

4/2/2018 8:31 AM - Screen Clipping

Crypto isakamp enable

```
R1#conf t  
Enter configuration commands, on  
R1(config)#crypto isakmp enable  
R1(config)#  
R1(config)#
```

4/2/2018 8:31 AM - Screen Clipping

```
R1#show crypto isakmp policy  
  
Global IKE policy  
Default protection suite  
    encryption algorithm: DES - Data Encryption Standard (56 bit keys)  
    hash algorithm: Secure Hash Standard  
    authentication method: Rivest-Shamir-Adleman Signature  
    Diffie-Hellman group: #1 (768 bit)  
    lifetime: 86400 seconds, no volume limit
```

4/2/2018 8:32 AM - Screen Clipping

```
R1#conf t  
Enter configuration commands, one per line  
R1(config)#crypto isakmp policy ?  
    <1-10000> Priority of protection suite  
R1(config)#crypto isakmp policy _
```

4/2/2018 8:33 AM - Screen Clipping

Set priority. The lower you go the higher the priority.

```
R1(config)#crypto isakmp policy 100  
R1(config-isakmp)#
```

4/2/2018 8:33 AM - Screen Clipping

Now do authentication..

```
R1(config-isakmp)#authentication ?  
    pre-share Pre-Shared Key  
    rsa-encr Rivest-Shamir-Adleman Encryption  
    rsa-sig Rivest-Shamir-Adleman Signature  
R1(config-isakmp)#authentication pre-share
```

4/2/2018 8:34 AM - Screen Clipping

Now do the encryption and hash.

```
R1(config-isakmp)#encryption ?  
    3des Three key triple DES  
    aes AES - Advanced Encryption Standard.  
    des DES - Data Encryption Standard (56 bit keys).  
R1(config-isakmp)#encryption 3des ?  
    <cr>  
R1(config-isakmp)#encryption 3des
```

4/2/2018 8:34 AM - Screen Clipping

```
R1(config-isakmp)#encryption aes  
R1(config-isakmp)#hash ?  
    md5 Message Digest 5  
    sha Secure Hash Standard  
R1(config-isakmp)#hash md5  
R1(config-isakmp)#lifetime ?  
    <60-86400> Lifetime in seconds
```

4/2/2018 8:35 AM - Screen Clipping

Now the time.

```
R1(config-isakmp)#hash md5
R1(config-isakmp)#lifetime ?
<60-86400> lifetime in seconds
R1(config-isakmp)#lifetime 86400
R1(config-isakmp)#^Z
```

4/2/2018 8:35 AM - Screen Clipping

Now check your work.

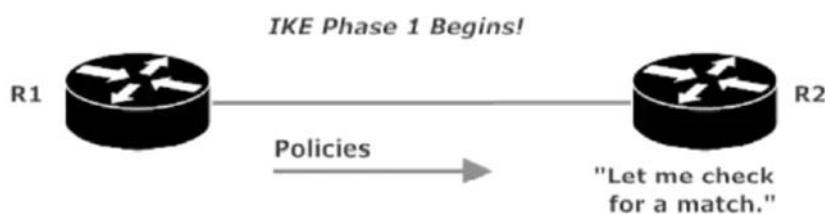
```
R1#show crypto isakmp policy

Global IKE policy
Protection suite of priority 100
    encryption algorithm: Three key triple DES
    hash algorithm: Message Digest 5
    authentication method: Pre-Shared Key
    Diffie-Hellman group: #1 (768 bit)
    lifetime: 86400 seconds, no volume limit
Default protection suite
    encryption algorithm: DES - Data Encryption Standard (56 bit keys)
    hash algorithm: Secure Hash Standard
    authentication method: Rivest-Shamir-Adleman Signature
    Diffie-Hellman group: #1 (768 bit)
    lifetime: 86400 seconds, no volume limit
R1#
```

4/2/2018 8:35 AM - Screen Clipping

These policies are all a part of phase 1.

These policies are all part of IKE Phase 1. If Phase 1 goes badly, you will *not* have a Phase 2! Phase 1 begins with the connection initiator sending its policies to the desired VPN partner.



4/2/2018 8:36 AM - Screen Clipping

The router checks lowest number policy first. Encryption, auth, dh, and hash all have to match, but not the lifetime.

VPN 3: Ipsec SA Config

Monday, April 2, 2018 8:37 AM

```
R1(config)#crypto isakmp key ?
 0 Specifies an UNENCRYPTED password will follow
 6 Specifies an ENCRYPTED password will follow

R1(config)#crypto isakmp key 6 ?
  WORD The HIDDEN user password string

R1(config)#crypto isakmp key 6 CCNP ?
  address define shared key with IP address
  hostname define shared key with hostname

R1(config)#crypto isakmp key 6 CCNP address ?
  A.B.C.D Peer IP address

R1(config)#crypto isakmp key 6 CCNP address 172.12.123.3 ?
  A.B.C.D Peer IP subnet mask
  no-xauth Bypasses XAuth for this peer
<cr>
```

4/2/2018 8:38 AM - Screen Clipping

Transform sets are groups of individual parameters that enforce a security policy. We write them the same as ISAKMP policy.

```
R1(config)#crypto isakmp key 6 CCNP address 172.12.123.3
R1(config)#
R1(config)#
R1(config)#crypto ipsec ?
  client          Configure a client
  df-bit          Handling of encapsulated DF bit.
  fragmentation   Handling of fragmentation of near-MTU sized packets
  nat-transparency IPsec NAT transparency model
  optional         Enable optional encryption for IPsec
  profile          Configure an ipsec policy profile
  security-association Security association parameters
  transform-set    Define transform and settings
```

4/2/2018 8:39 AM - Screen Clipping

```
R1(config)#crypto ipsec transform-set CCNP_LAB ah-md5-hmac
R1(cfg-crypto-trans)#[
```

Create your transform set.

```
R1(cfg-crypto-trans)#mode ?
  transport transport (payload encapsulation) mode
  tunnel   tunnel (datagram encapsulation) mode

R1(cfg-crypto-trans)#mode tunnel
R1(cfg-crypto-trans)#exit
```

4/2/2018 8:40 AM - Screen Clipping

Set it for tunnel mode.

Now set the lifetime.

```
R1(config)#crypto ipsec security-association ?
  idle-time  Automatically delete IPSec SAs after a given idle period
  lifetime   security association lifetime
  replay     Set replay checking.

R1(config)#crypto ipsec security-association lifetime ?
  kilobytes  Volume-based key duration
  seconds    Time-based key duration

R1(config)#crypto ipsec security-association lifetime 300
```

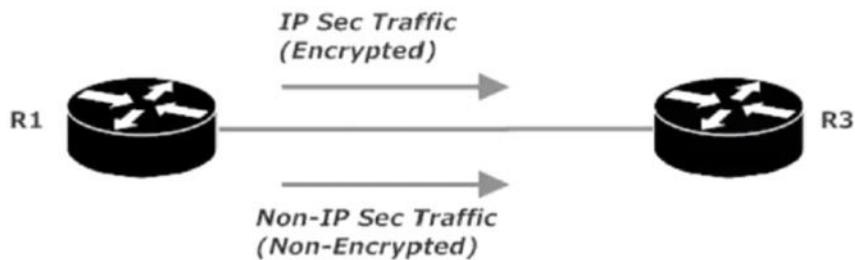
4/2/2018 8:41 AM - Screen Clipping

VPN 4: Building, Verifying, and Debugging your VPN build

Monday, April 2, 2018 8:42 AM

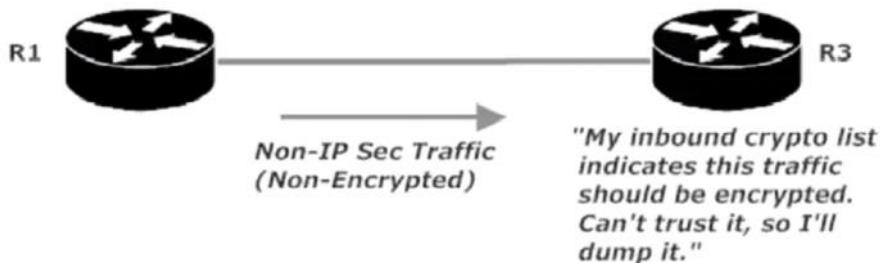
Crypto Access Lists

The crypto ACL is the exact same ACL that we've been writing our entire Cisco career, just put to a different use and a slightly different operation. A crypto ACL evaluating *outbound* traffic decides which traffic flows will be protected by IPSec and which ones will not. Traffic permitted by the ACL is protected while flows denied by the ACL are transmitted without IPSec's protection.



A crypto ACL "permit" on *inbound* traffic indicates traffic that should be IPSec-protected when it arrives. If said traffic arrives and is not so protected, the router will drop that traffic.

4/2/2018 8:44 AM - Screen Clipping



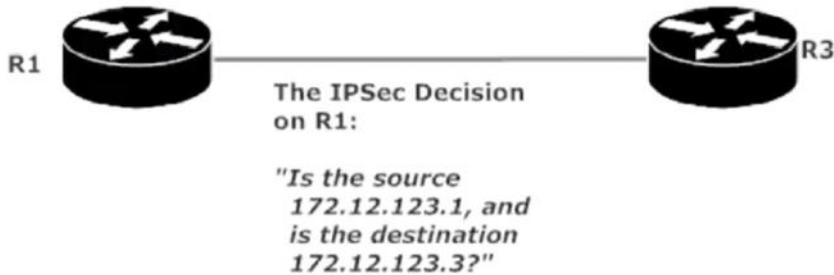
4/2/2018 8:45 AM - Screen Clipping

Crypto ACLs are used to evaluate both inbound and outbound traffic; there's no "in" or "out" when applying a crypto ACL to an interface. Basically, the same ACL is read forward for outbound traffic and backwards for inbound traffic. Assume this ACL is on R1:

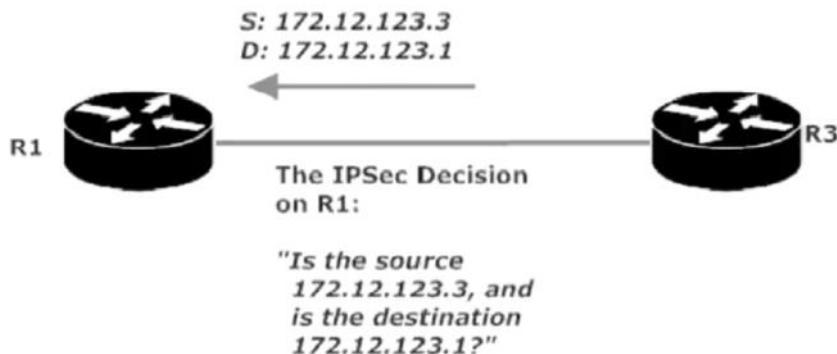
```
R1(config)#access-list 103 permit ip host 172.12.123.1 host 172.12.123.3
```

When it's part of a crypto map, ACL 103 is read forward for outgoing traffic...

4/2/2018 8:46 AM - Screen Clipping



... and backwards for incoming traffic.



4/2/2018 8:47 AM - Screen Clipping

This behavior makes it vital to put a symmetrical ACL on the remote VPN endpoint, not an identical one. We can't just cut and paste this ACL from R1 to R3:

```
R1(config)#access-list 103 permit ip host 172.12.123.1 host 172.12.123.3
```

Instead, we need to put a mirror image of that ACL on R3.

```
R3(config)#access-list 103 permit ip host 172.12.123.3 host 172.12.123.1
```

4/2/2018 8:47 AM - Screen Clipping

There's no problem using *host* in a crypto ACL, but Cisco strongly recommends against using *any*, especially *permit any any*. You can end up with way too much encrypted traffic leaving the interface and / or dropping important but unencrypted incoming control traffic.

That's enough talk about ACL 103. Let's put it into action via a crypto map!

```
R1(config)#crypto map CCNP ?
<1-65535>      Sequence to insert into crypto map entry
client           Specify client configuration settings
gdoi             Configure crypto map gdoi features
isakmp           Specify isakmp configuration settings
isakmp-profile   Specify isakmp profile to use
```

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#access-list 103 permit ip host 172.12.123.1 host 172.12.123.3
```

Create ACL

```
R1(config)#crypto map CCNP 100 ?  
  ipsec-isakmp  IPSEC w/ISAKMP  
  ipsec-manual  IPSEC w/manual keying  
<cr>  
  
R1(config)#crypto map CCNP 100 ipsec-isakmp  
% NOTE: This new crypto map will remain disabled until a peer  
      and a valid access list have been configured.
```

Create crypto map

```
R1(config-crypto-map)#match address ?  
  <100-199>    IP access-list number  
  <2000-2699>  IP access-list number (expanded range)  
  WORD          Access-list name  
  
R1(config-crypto-map)#match address 103  
R1(config-crypto-map)#[
```

Match to the address of ACL.

```
R1(config-crypto-map)#set peer ?  
  A.B.C.D  IP address of peer  
  WORD      Host name of the peer  
  
R1(config-crypto-map)#set peer 172.12.123.3  
R1(config-crypto-map)#set transform-set CCNP_LAB  
R1(config-crypto-map)#exit  
R1(config)#int serial 1/0  
R1(config-if)#crypto map ?  
  WORD  Crypto Map tag  
<cr>
```

Set it to the interface.

```
R1(config-if)#crypto map CCNP ?  
  redundancy  enable redundancy  
<cr>  
  
R1(config-if)#crypto map CCNP  
R1(config-if)#exi  
*Jul 16 00:01:57.744: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON  
R1(config-if)#exit  
R1(config)#[
```

```
R1#debug cr
*Jul 16 00:02:15.092: %SYS-5-COMM-1: R1% debug crypto ipsec
R1#debug crypto ipsec
Crypto IPSEC debugging is on
R1#
```

4/2/2018 8:53 AM - Screen Clipping

```
src port      : 0
dst port      : 0
*Jul 16 00:02:25.157: IPSEC(crypto_ipsec_sa_find_iden
the same proxies and 172.12.123.3
*Jul 16 00:02:25.157: IPSEc: Flow_switching Allocate
*Jul 16 00:02:25.157: IPSEC(policy_db_add_ident): sr
123.3, dest_port 0

*Jul 16 00:02:25.157: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.12.123.1, sa_proto= 51,
    sa_spi= 0x3AECD0F4(988598516),
    sa_trans= ah-md5-hmac , sa_conn_id= 2001
*Jul 16 00:02:25.161: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.12.123.3, sa_proto= 51,
    sa_spi= 0x34B42560(884221280),
    sa_trans= ah-md5-hmac , sa_conn_id= 2002
R1#
```

4/2/2018 8:53 AM - Screen Clipping

```
R1#ping 172.12.123.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.123.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/93/97 ms
R1#
R1#show crypto isakmp sa
dst          src          state           conn-id slot status
172.12.123.3  172.12.123.1  QM_IDLE        1      0 ACTIVE
```

4/2/2018 8:55 AM - Screen Clipping

Double check the SA.

```
R1#show crypto ipsec sa  
interface: Serial1/0  
Crypto map tag: CCNP, local addr 172.12.123.1  
protected vrf: (none)  
local ident (addr/mask/prot/port): (172.12.123.1/255.255.255.0/0/0)  
remote ident (addr/mask/prot/port): (172.12.123.3/255.255.255.0/0/0)  
current_peer 172.12.123.3 port 500  
    PERMIT, flags={origin_is_acl,}  
    #pkts encaps: 9, #pkts encrypt: 9, #pkts digest: 9  
    #pkts decaps: 9, #pkts decrypt: 9, #pkts verify: 9  
    #pkts compressed: 0, #pkts decompressed: 0  
    #pkts not compressed: 0, #pkts compr. failed: 0  
    #pkts not decompressed: 0, #pkts decompress failed: 0  
    #send errors 1, #recv errors 0  
  
    local crypto endpt.: 172.12.123.1, remote crypto endpt.: 172.12.123.3  
    path mtu 1500, ip mtu 1500, ip mtu idb Serial1/0  
    current outbound spi: 0x34B42560(884221280)  
  
    inbound esp sas:  
  
    inbound ah sas:
```

4/2/2018 8:55 AM - Screen Clipping

Find your crypto map, address, port number, (500),

VPN 5: DMVPN, NHRP, and mGRE

Monday, April 2, 2018 8:58 AM

An Introduction To DMVPN

For the sake of clarity and sanity, I've left the cloud out of the following illustrations. We have six remote office spokes and one central office hub.



4/2/2018 8:58 AM - Screen Clipping

We could implement a full mesh of VPNs, but as with static routing and BGP, a full mesh is a technically correct manner of getting things done that comes with a ton of overhead and some administrative nightmares on the side at no extra charge. Each spoke router would require us to configure five VPNs. Worse than the eventual troubleshooting that comes with that much static configuration is the fact that the VPNs will stay up even when there's no traffic going across. That's a real waste of router resources and bandwidth.

A much more efficient solution, *DMVPN (Dynamic Multipoint VPNs)* allow a spoke router to dynamically create a VPN to another spoke when the VPN is actually needed, and then to tear that same VPN down when it's no longer needed.

DMVPN does not work in a vacuum. Far from it! For DMVPNs to work, we need the cooperation of...

- ... a stable dynamic routing protocol (and / or a static route)
- ... GRE (mGRE, to be specific)
- ... NHRP, the Next-Hop Routing Protocol
- ... and IPSec!

We're familiar with IPSec, and we'll discuss mGRE and NHRP in just a moment. Right now, a quick word regarding the routing protocol.

4/2/2018 9:00 AM - Screen Clipping

I know this is *really* obvious, but I'm mentioning it since it's easy to overlook the obvious. For a router to build a dynamic tunnel to another router, that router

4/2/2018 9:04 AM - Screen Clipping

has to know how to reach the remote router. Remember, the hub's not getting directly involved in the tunnel build; the DMVPN goes directly from one spoke to another.

On a related point, if you see the tunnel bouncing or flapping – that is, going up, then down, then up, then down – that indicates the local router is able to reach the remote router only intermittently. That can be anything from a simple adjacency issue to an unstable OSPF area (and just about anything in between). In short, if your local router has issues reaching the remote router for any reason, that tunnel's going down. When it comes to troubleshooting DMVPN, I'd start with the connectivity (or lack of same) to the remote router, then mGRE, then IPSec.

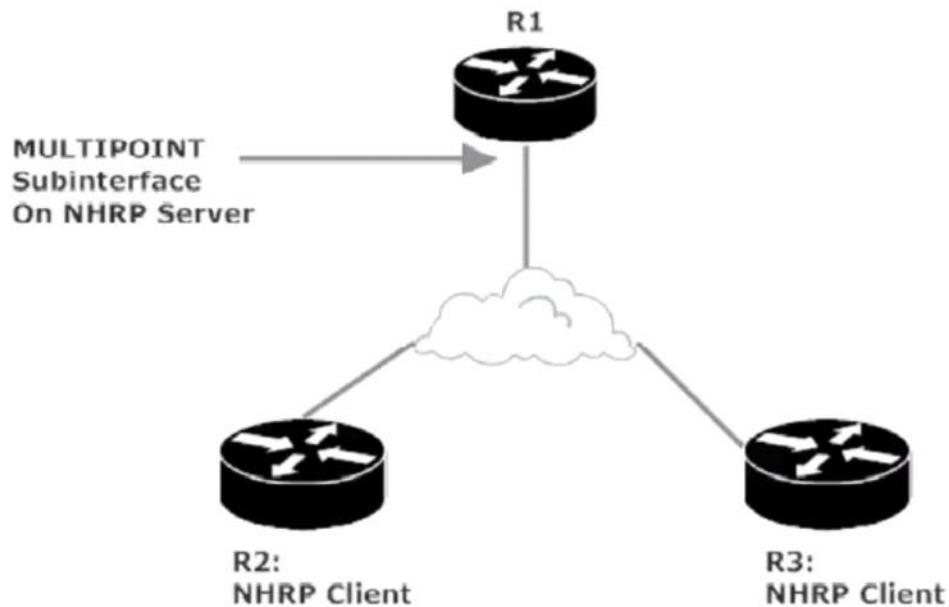
And speaking of mGRE...

"She's got gaps, I got gaps, together we fill gaps." -- Rocky Balboa

IPSec has one glaring weakness – the inability to protect multicast packets. However, GRE can encapsulate multicast packets, bringing us the unusual combination of GRE Over IPSec. Basically, we're encapsulating the multicast packet via GRE and then sending the encapsulated packet over an IPSec tunnel. GRE fills the gap for IPSec not being able to protect multicasts, and IPSec fills the gap for GRE lacking security.

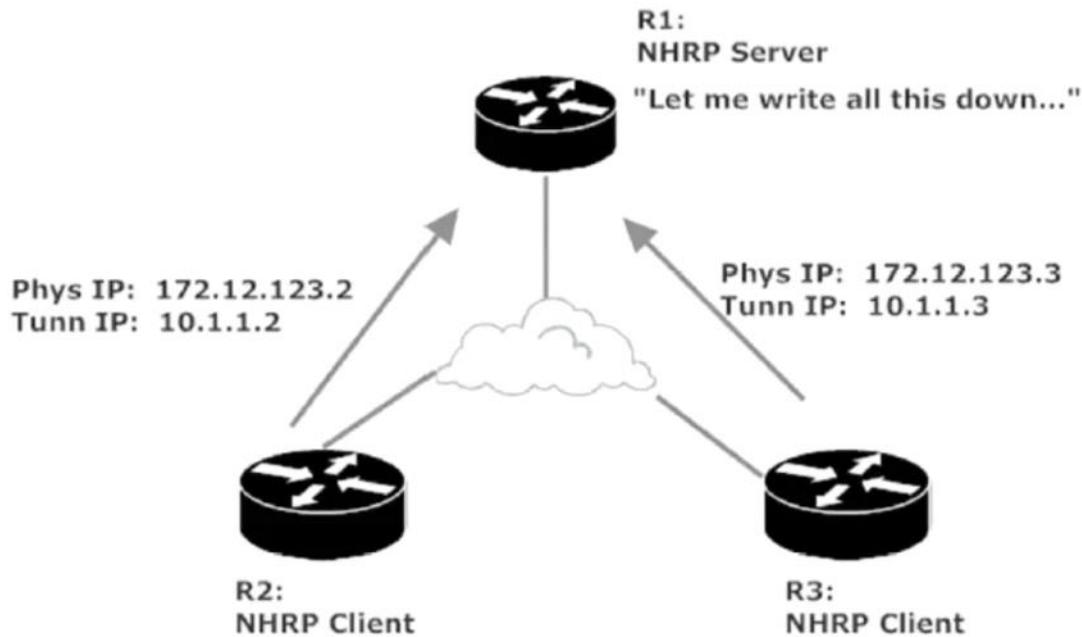
4/2/2018 9:04 AM - Screen Clipping

mGRE really comes in handy in our dreaded hub-and-spoke network! Multipoint GRE makes it possible for the hub to use only one interface for as many tunnels as you need. Thing is, if we have a single multipoint interface on the hub router and multiple endpoints for our tunnels, we gotta have some kind of mapping in there. That's where the *Next Hop Resolution Protocol* (NHRP) comes in.



4/2/2018 9:07 AM - Screen Clipping

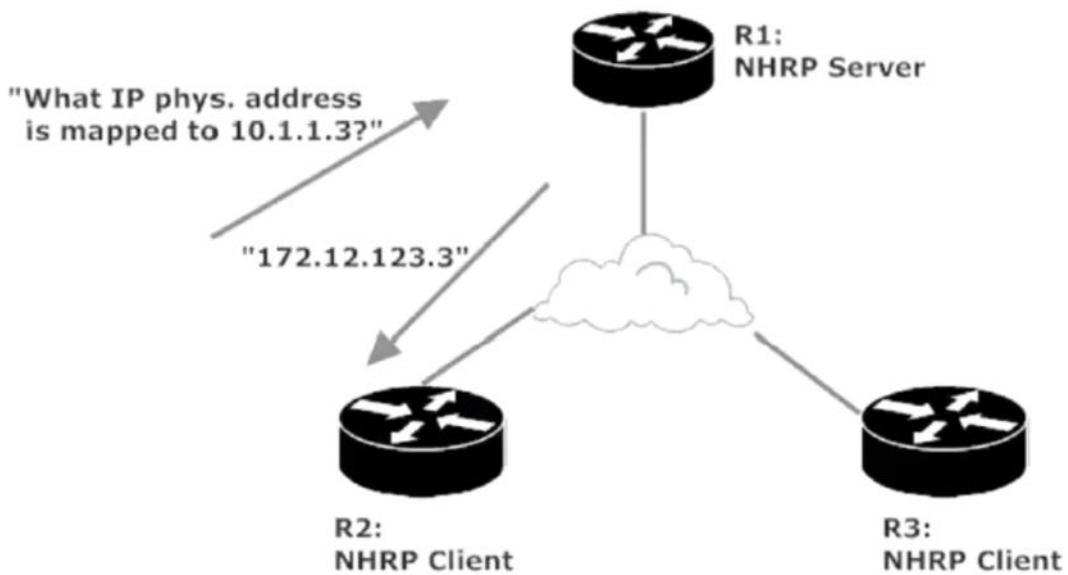
Our NHRP deployment uses the client-server model, with our hub as the server and the spokes as the client. The clients send two IP addresses to the server – one identifying the physical address used for the tunnel and the other the IP address assigned to the tunnel itself. The server puts this info in its NHRP database.



4/2/2018 9:07 AM - Screen Clipping

NHRP works in a client server model.

The database comes into play when one of our spokes needs to build a tunnel to another. When R2 wants to build a tunnel to R3, R2 will ask R1 what physical interface IP address is mapped to the tunnel IP address 10.1.1.3. R1 checks its NHRP database for the answer, R1 answers R2's query, and R2 can then successfully tunnel to R3. When NHRP info is received directly from the NHRP Server, you'll see the *authoritative* flag set in the output of *show ip nhrp*.



4/2/2018 9:08 AM - Screen Clipping

Sh ip NRP - Is kind of like ARP. This is CCIE level.

the tunnel source as usual...

```
R1(config)#int tunnel 0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#tunnel source serial 0/2/0.123
R1(config-if)#tunnel destination ?
    Hostname or A.B.C.D  ip address or host name
    X:X:X::X           IPv6 address
```

... but you will not put a single destination address, since an mGRE tunnel doesn't have a single destination. Instead, be sure to configure the tunnel mode as *gre multipoint*.

```
R1(config)#int tunnel 0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#tunnel source serial 0/2/0.123
R1(config-if)#tunnel destination ?
    Hostname or A.B.C.D  ip address or host name
    X:X:X::X           IPv6 address

R1(config-if)#tunnel mode ?
aurp    AURP TunnelTalk AppleTalk encapsulation
cayman  Cayman TunnelTalk AppleTalk encapsulation
dvmrp   DVMRP multicast tunnel
eon     EON compatible CLNS tunnel
gre     generic route encapsulation protocol
```

4/2/2018 9:09 AM - Screen Clipping

Put the tunnel source, not the destination address, but instead configure the tunnel as *gre multipoint*.

```
R1(config-if)#tunnel mode ?  
aurp    AURP TunnelTalk AppleTalk encapsulation  
cayman  Cayman TunnelTalk AppleTalk encapsulation  
dvmrp   DVMRP multicast tunnel  
eon     EON compatible CLNS tunnel  
gre     generic route encapsulation protocol  
ipip    IP over IP encapsulation  
ipsec   IPSec tunnel encapsulation  
iptalk   Apple IPTalk encapsulation  
ipv6    Generic packet tunneling in IPv6  
ipv6ip  IPv6 over IP encapsulation  
mpls    MPLS encapsulations  
nos     IP over IP encapsulation (KA9Q/NOS compatible)  
rbscp   RBSCP in IP tunnel
```

```
R1(config-if)#tunnel mode gre ?  
ip      over IP  
ipv6   over IPv6  
multipoint over IP (multipoint)
```

```
R1(config-if)#tunnel mode gre multipoint ?  
<cr>
```

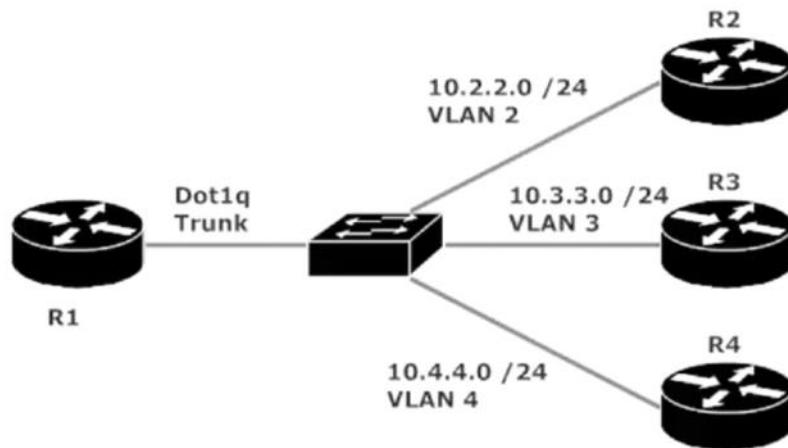
```
R1(config-if)#tunnel mode gre multipoint
```

VPN 6: VRF Lab 1

Monday, April 2, 2018 9:10 AM

Sometimes it's fine to let traffic types and flows mix, and sometimes it's not. Maybe you have some ultra-time-sensitive voice and video traffic and you want to keep it separate from the usual riff-raff traffic. Maybe you just want to keep a certain subnet's traffic separate from others in order to keep that subnet's existence as secret as possible. Whatever the reason, Virtual Routing and Forwarding can help. We're actually going to use a "lite" version here, cleverly known as VRF-Lite.

Here's the physical setup for this lab, along with the IP addresses.



4/2/2018 9:10 AM - Screen Clipping

Virtual routing and forwarding allows us to keep a certain subnet's traffic separate from others in order to keep its existence as a secret.

The Configs for this lab:

The Switch:

```
interface FastEthernet0/1
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface FastEthernet0/2
switchport access vlan 2
switchport mode access
!
interface FastEthernet0/3
switchport access vlan 3
switchport mode access
!
interface FastEthernet0/4
switchport access vlan 4
switchport mode access
```

4/2/2018 9:12 AM - Screen Clipping

R1:

```
interface FastEthernet0/0
  no ip address
```

4/2/2018 9:13 AM - Screen Clipping

```
duplex auto
speed auto
!
interface FastEthernet0/0.2
  encapsulation dot1Q 2
  ip address 10.2.2.1 255.255.255.0
!
interface FastEthernet0/0.3
  encapsulation dot1Q 3
  ip address 10.3.3.1 255.255.255.0
!
interface FastEthernet0/0.4
  encapsulation dot1Q 4
  ip address 10.4.4.1 255.255.255.0
!
```

R2:

```
interface FastEthernet0/0
  ip address 10.2.2.2 255.255.255.0
```

4/2/2018 9:13 AM - Screen Clipping

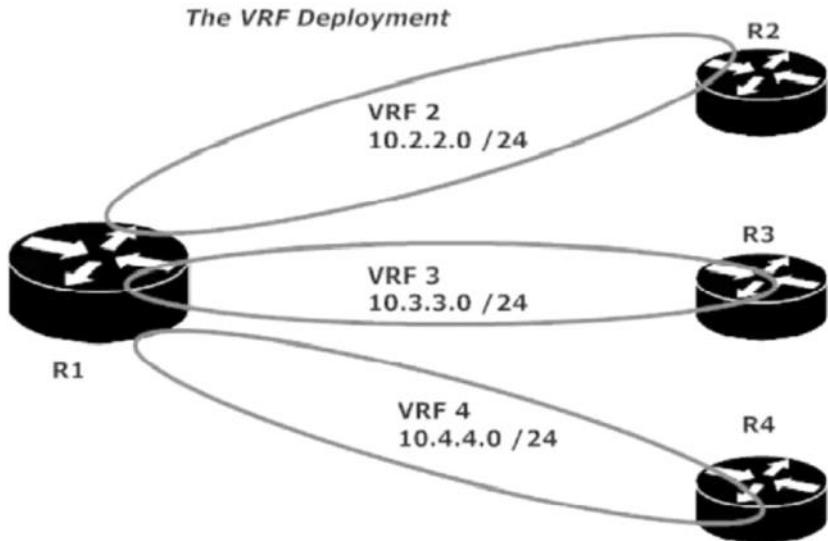
R3:

```
interface FastEthernet0/0
  ip address 10.3.3.3 255.255.255.0
```

R4:

```
interface FastEthernet0/0
  ip address 10.4.4.4 255.255.255.0
```

4/2/2018 9:14 AM - Screen Clipping



4/2/2018 9:14 AM - Screen Clipping

```
R1#conf t
Enter configuration mode
R1(config)#ip vtf ?
% Unrecognized command
R1(config)#ip vrf ?
  WORD  VPN Routing/For

R1(config)#ip vrf VRF2
R1(config-vrf)#exit
R1(config)#ip vrf VRF3
R1(config-vrf)#exit
R1(config)#ip vrf VRF4
```

Create the VRFS

```
R1(config-vrf)#int fast 0/0.2
R1(config-subif)#ip vrf ?
  forwarding  Configure forwarding table
  receive     Add Interface Address into VRF Table
  sitemap     Configure route-map for routes received from this site

R1(config-subif)#ip vrf forwarding ?
  WORD  Table name

R1(config-subif)#ip vrf forwarding VRF2
% Interface FastEthernet0/0.2 IP address 10.2.2.1 removed due to enabling V
F2
```

Now Assign them

```
R1(config-subif)#int fast 0/0.3
R1(config-subif)#ip vrf forwarding VRF3
% Interface FastEthernet0/0.3 IP address 10.3.3.1 removed due to enabling
E3
```

4/2/2018 9:17 AM - Screen Clipping

```
R1(config-subif)#int fast 0/0.4
R1(config-subif)#ip vrf forwarding VRF4
% Interface FastEthernet0/0.4 IP address 10
F4
R1(config-subif)#^Z
```

4/2/2018 9:17 AM - Screen Clipping

Name	Default RD	Interfaces
VRF2	<not set>	Fa0/0.2
VRF3	<not set>	Fa0/0.3
VRF4	<not set>	Fa0/0.4

```
R1#
```

4/2/2018 9:17 AM - Screen Clipping

Verify. RD is route descriptor. CCIE level.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP,
        D - EIGRP, EX - EIGRP external, O -
        N1 - OSPF NSSA external type 1, N2
        E1 - OSPF external type 1, E2 - OSP
        i - IS-IS, su - IS-IS summary, L1 -
        ia - IS-IS inter area, * - candidat
        o - ODR, P - periodic downloaded st
Gateway of last resort is not set
R1#
```

4/2/2018 9:18 AM - Screen Clipping

We don't have any default or even connected routes. Why? The message said it was removing the IP's.
When you're working with VRF after IPs have been configured on the router, you have to re-configure
the IP's. **VRF erases existing IP routes.**

VPN 7: VRF Lab 2

Monday, April 2, 2018 9:20 AM

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fast 0/0.2
R1(config-subif)#ip address 10.2.2.1 255.255.255.0
R1(config-subif)#int fast 0/0.3
R1(config-subif)#ip address 10.3.3.1 255.255.255.0
R1(config-subif)#int fast 0/0.4
R1(config-subif)#ip address 10.4.4.1 255.255.255.0
R1(config-subif)#^Z
R1#
```

4/2/2018 9:21 AM - Screen Clipping

We've reconfigured the IP address on their interfaces.

```
R1#show ip route
Codes: C - connected, S - static, R - RI
      D - EIGRP, EX - EIGRP external, O
      N1 - OSPF NSSA external type 1, N
      E1 - OSPF external type 1, E2 - O
      i - IS-IS, su - IS-IS summary, L1
      ia - IS-IS inter area, * - candid
      o - ODR, P - periodic downloaded

Gateway of last resort is not set

R1#
```

4/2/2018 9:21 AM - Screen Clipping

We still don't see them because VRF has its own **sh ip route vrf** command.

```
R1#show ip route vrf VRF2
```

Routing Table: VRF2

Codes: C - connected, S - static, R - RIP, M - mobile, B - B
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF I
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA exterr
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2
ia - IS-IS inter area, * - candidate default, U - per
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.2.2.0 is directly connected, FastEthernet0/0.2

R1#

R1#

4/2/2018 9:21 AM - Screen Clipping

Now we can check 2

```
Gateway of last resort is not set
```

10.0.0.0/24 is subnetted, 1 s

C 10.3.3.0 is directly connec

R1#show ip route vrf VRF4

Routing Table: VRF4

Codes: C - connected, S - static,
D - EIGRP, EX - EIGRP exten
N1 - OSPF NSSA external typ
E1 - OSPF external type 1,
i - IS-IS, su - IS-IS summar
ia - IS-IS inter area, * -
o - ODR, P - periodic downl

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 s

C 10.4.4.0 is directly connec

4/2/2018 9:22 AM - Screen Clipping

3 and 4.

We still cant ping though?

```
R1#  
R1#ping 10.2.2.2  
  
Type escape sequence  
Sending 5, 100-byte I  
....  
Success rate is 0 per  
R1#ping 10.3.3.3  
  
Type escape sequence  
Sending 5, 100-byte I  
....  
Success rate is 0 per  
R1#ping 10.4.4.4  
  
Type escape sequence  
Sending 5, 100-byte I
```

4/2/2018 9:22 AM - Screen Clipping

VRF has its own ping as well.

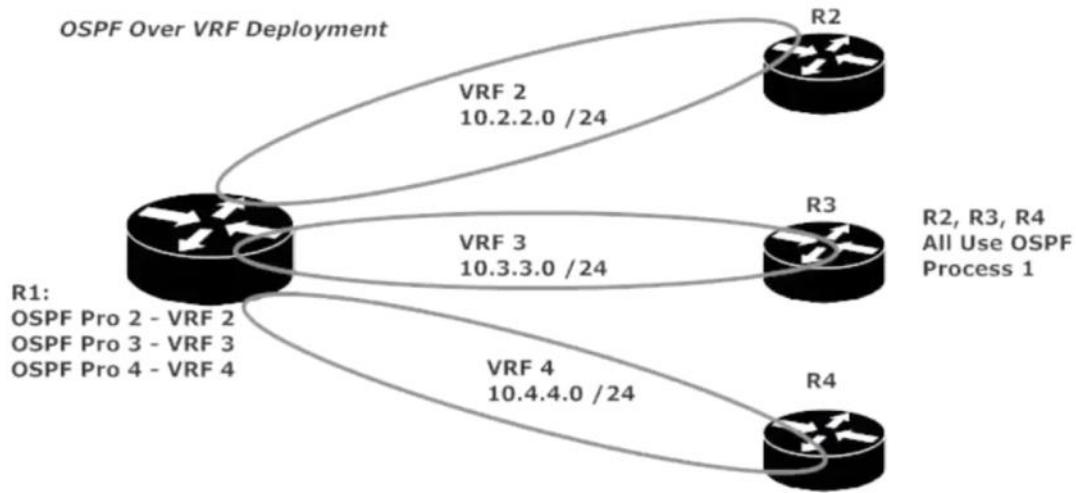
```
R1#ping vrf VRF2 ?  
WORD          Ping destination address  
appletalk    Appletalk echo  
clns         CLNS echo  
decnet       DECnet echo  
ip           IP echo  
ipv6         IPv6 echo  
ipx          Novell/IPX echo  
srb          srb echo  
tag          Tag encapsulated IP echo  
<cr>  
  
R1#ping vrf VRF2 10.2.2.2  
  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.2.2.2  
.!!!!  
Success rate is 80 percent (4/5),
```

4/2/2018 9:24 AM - Screen Clipping

VPN 8: VRF Lab 3

Monday, April 2, 2018 9:25 AM

With our connected routes taken care of, let's introduce OSPF to our VRF configuration.



Just as we have three instances of VRF on R1, we need to create three separate instances of OSPF, and that means adding a little info to the usual *router ospf* command.

4/2/2018 9:25 AM - Screen Clipping

```
R1(config)#router ospf 2 vrf ?
WORD   VPN Routing/Forwarding Instance (VRF) name

R1(config)#router ospf 2 vrf VRF2
R1(config-router)#network 10.2.2.0 0.0.0.255 area 0
R1(config-router)#
R1(config-router)#router ospf 3 vrf VRF3
R1(config-router)#network 10.3.3.0 0.0.0.255 area 0
R1(config-router)#
R1(config-router)#router ospf 4 vrf VRF4
R1(config-router)#network 10.4.4.0 0.0.0.255 area 0
R1(config-router)#

```

4/2/2018 9:27 AM - Screen Clipping

First we set up router 1.

Routers 2 3 and 4 don't really know about the VRFs. We just need to set up the OSPF configs. Make sure the process ID needs to be the same.

```

R2#
R2#conf t
Enter configuration commands, one per line. End with
R2(config)#router ospf 1
R2(config-router)#network 10.2.2.0 0.0.0.255 area 0
R2(config-router)#^Z
R2#wr
Building configuration...

```

4/2/2018 9:28 AM - Screen Clipping

```

R3#
R3#conf t
Enter configuration commands, one per line. End with
R3(config)#router ospf 1
R3(config-router)#network 10.3.3.0 0.0.0.255 area 0
R3(config-router)#^Z
R3#wr
Building configuration...

*Nov 1
BRYANT_ADV_1#4
[Resuming connection 4 to r4 ... ]

R4#
R4#conf t
Enter configuration commands, one per line. End with
R4(config)#router ospf 1
R4(config-router)#network 10.4.4.0 0.0.0.255 area 0
R4(config-router)#^Z

```

4/2/2018 9:29 AM - Screen Clipping

```

R1#show ip osp
*Aug 21 08:24:59.163: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip ospf neigh

Neighbor ID      Pri  State            Dead Time    Address          Interface
4.4.4.4           1    INIT/DROTHER   00:00:33     10.4.4.4        FastEther
0.4
3.3.3.3           1    FULL/BDR       00:00:33     10.3.3.3        FastEther
0.3
2.2.2.2           1    FULL/BDR       00:00:31     10.2.2.2        FastEther
0.2
R1#
*Aug 21 08:25:05.413: %OSPF-5-ADJCHG: Process 4, Nbr 4.4.4.4 on FastEtherne
4 from LOADING to FULL, Loading Done

```

4/2/2018 9:29 AM - Screen Clipping

Now check your routes on 1.

Set your lookbacks in area 2 on routers 2, 3 and 4.

```

R1#show ip route vrf VRF2

Routing Table: VRF2
Codes: C - connected, S - static, R - RIP, M - mobile, B -
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA exten-
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L
      ia - IS-IS inter area, * - candidate default, U - pe-
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O IA   2.2.2.2 [110/2] via 10.2.2.2, 00:00:18, FastEthernet0/0.2
      10.0.0.0/24 is subnetted, 1 subnets
C       10.2.2.0 is directly connected, FastEthernet0/0.2
R1#

```

4/2/2018 9:30 AM - Screen Clipping

There's always room for improvement, though, and *Easy Virtual Networking* (EVN) is an improvement over VRF Lite. The config is a lot simpler, too! Instead of creating the subinterfaces required in our VRF Lite lab, we'll have a Virtual Network Trunk (VNET) carry traffic tagged with – you guessed it! -- a VNET tag.

Another EVN benefit, *route replication*, grants each virtual network access to the Routing Information Base (RIB) for every VRF. That really cuts down on the number of overall duplicate routing table entries, which in turn makes the overall process faster and lessens the hit to our router resources. This keeps each virtual network's traffic separate while making overall network operations much more efficient.

EVN configuration is beyond the scope of the exam, but I would recommend having a look at the following 2-page PDF on Cisco's website:

http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/easy-virtual-network-evn/aag_c45-675118.pdf

To go with that, here's a quick list of EVN requirements, limitations, and benefits!

- Supports up to 32 virtual networks

- EVN trunks can be built on any interface that can run dot1q encapsulation

- However, EVN trunks do not support ACLs, NAT, NetFlow, or WCCP (Web Cache Communication Protocol)

4/2/2018 9:31 AM - Screen Clipping

IPv6 1: Fundamental and Zero Compression Techniques

Tuesday, April 3, 2018 8:20 AM

IP Version 6

The good news: The 128-bit addresses used in IPv6 give us a tremendous number of addresses and is designed specifically with route summarization in mind.

The temporarily bad news: Those 128-bit addresses. The ones that are 128 bits long. That's a long address to type, especially for those of us who hate entering 32-bit addresses.

I say "temporarily bad" because there is a bit of shock factor in just hearing about 128-bit addresses when you're used to IPv4's 32-bit addresses. Once you get used to the IPv6 address format (and you will), working with IPv6 will become second nature.

IPv6 brings us quite a few improvements over IPv4:

Those broadcasts we're always trying to limit are things of the past. IPv6 doesn't use broadcasts.

NAT isn't gone yet, but it will be as IPv6 continues to replace IPv4. (NAT is not a thing of the past when it comes to the CCNP Route exam, and we'll discuss NAT in another part of the course.)

4/3/2018 8:21 AM - Screen Clipping

It bears repeating that IPv6 was designed with route aggregation in mind, which makes aggregation easier and more effective, and in turn keeps our routing tables – say it with me! – *complete and concise*.

DHCP is still around, but IPv6 nodes can assign themselves an address without the help of a DHCP server through a little process called *autoconfiguration*. More on that soon.

Quality of Service (QoS) capabilities are greater with the IPv6 header values.

4/3/2018 8:23 AM - Screen Clipping

IPv6 Header Fields

There are quite a few changes in the headers as we move from IPv4 to IPv6. There are eight header fields overall in IPv6:

Version: Set to "6". And yes, I know you know that.

Traffic Class: In IPv4, this was the Type Of Service field. The "traffic class" name comes from this field's ability to allow us to assign levels of importance to a packet via QoS.

4/3/2018 8:24 AM - Screen Clipping

Flow Label: This field allows a packet to be labeled as part of a particular flow. This also helps with QoS, allowing us to prioritize traffic flows rather than individual packets. This header has no equivalent in IPv4.

Payload Length: Same thing as the Total Length field in IPv4.

Hop Limit: Roughly equivalent to IPv4's Time To Live field. Every hop decrements this counter by one. When this counter hits zero, the Time To Live becomes the Time To Be Discarded.

Next Header: Equivalent to IPv4's Protocol field.

Source Address, Destination Address: Same function, just larger. 128 bits each, to be exact!

A few IPv4 fields didn't make the cut to IPv6: *Header Length, Identification, Flags, Fragment Offset, and Header Checksum.*

Now, about those 128 bits...

The IPv6 Address Format, Zero Compression, and Leading Zero Compression

4/3/2018 8:25 AM - Screen Clipping

Sample IPv4 address: 129.14.12.200

Sample IPv6 address: 1029:9183:81AE:0000:0000:0AC1:2143:019B

A non-compressed IPv6 address has eight sections of four hex values, separated by a total of seven colons. Luckily for you and I (the you-know-whos), there are ways to compress these addresses so not so many numbers and letters are involved. This helps in the field, and it'll really help the day you pass your CCNP Route exam.

From your CCNA studies, you remember there's no difference between an upper-case letter and a lower-case letter in hex. Simple rule, right? Right! The other simple rules deal with all the zeroes you'll deal with in IPv6 addressing.

4/3/2018 8:31 AM - Screen Clipping

From your CCNA studies, you remember there's no difference between an upper-case letter and a lower-case letter in hex. Simple rule, right? Right! The other simple rules deal with all the zeroes you'll deal with in IPv6 addressing.

If you have consecutive fields of zeroes, they can be expressed with two colons. It doesn't matter if you have two fields of zeroes or eight (really!), you can simply type two colons and you're done. The key here with this *zero compression* is that you can only do it once per address. Here's an example:

Original IPv6 address: 1234:1234:0000:0000:0000:3456:3434

Same address with zero compression: 1234:1234::3456:3434

Thought you'd like that! I also know you'll like *leading* zero compression, which allows us to drop the leading zeroes in any field. The key rules with leading zero

4/3/2018 8:32 AM - Screen Clipping

You have to leave at least one number in each field, even if the field is all zeroes.

You can perform leading zero compression as often as needed in a single address.

An example of leading zero compression:

Original address: 1234:0000:1234:0000:1234:0000:1234:0123

Same address using only leading zero compression:
1234:0:1234:0:1234:0:1234:123

One more leading zero / zero compression rule: You're allowed to use both in a single address. Just remember the frequency rules and you're all set! Let's see what we can do using both methods...

Original address: 1111:0000:0000:1234:0011:0022:0033:0044

Newly compressed address: 1111::1234:11:22:33:44

4/3/2018 8:32 AM - Screen Clipping

Original address: 1111:0000:0000:1234:0011:0022:0033:0044

Newly compressed address: 1111::1234:11:22:33:44

We used zero compression to use a double-colon to replace the second and third fields, which were both all zeroes. Leading zero compression replaced the two zeroes at the beginning of each of the last four fields.

Watch out for incorrectly expressed IPv6 addresses, both on your exam and in the field. If you're looking at an IPv6 address with more than two consecutive colons, or you see more than one set of consecutive colons, you're looking at an illegally expressed IPv6 address. The following two addresses illegal just on the base of the colons, so you don't have to look at anything else – they're *immediately illegal* expressions.

Immediately Illegal: 1111::::2222:3333:4444:5555

Immediately Illegal: 1111::2222:3333::

4/3/2018 8:32 AM - Screen Clipping

IPV6 2: EUI64 and the Interface Identifiers

Tuesday, April 3, 2018 8:32 AM

Interface Identifiers And EUI-64

Every interface on any given IPv6 link needs a unique identifier, cleverly called the *interface identifier*. The interface identifier is a 64-bit value that you and I don't have to manually enter, but we do need to know how it's created and how it's assigned.

RFC 2373 defines the 64-bit Extended Unique Identifier (EUI-64). This allows the interface to assign an interface identifier to itself, using the interface's MAC address. Sounds like we need a few extra bits from somewhere, since the MAC address is only 48 bits long! We get those extra bits by dropping "FFFE" right in the middle of the address, between the OUI and the vendor code. If you see "FFFE" in the middle of an interface identifier, you know it's an EUI-64 assigned identifier.

A quick MAC address review: In the MAC address 00-01-02-aa-bb-cc, the Organizationally Unique Identifier (OUI) is 00-01-02 and the vendor code is aa-bb-cc. That makes it easy enough to come up with the interface identifier for this address:

00-01-02-FF-FE-aa-bb-cc

4/3/2018 8:33 AM - Screen Clipping

And that's right...*almost*. There's just one little detail we need to take care of, and that detail is the 7th bit of the first octet (00000000). The 7th bit is the Universal / Local bit, which kind of describes what this bit is all about. This bit tells us whether this address is universally unique or just locally unique (unique only to this link, that is). It's assumed a MAC address is universally unique, so we'll set that U/L bit is set to 1...

00000010

... giving us a final interface identifier of 02-01-02-FF-FE-aa-bb-cc.

4/3/2018 8:36 AM - Screen Clipping

Let's walk through this interface identifier creation process on a live Cisco router. Here's the MAC address of Fast0/0 on R1. What's the interface identifier?

```
R1#show int fast 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 000f.f7c4.09c0 (bia 000f.f7c4.09c0)
```

The MAC is 000f.f7c4.09c0. Here's how we get to the interface identifier:

First, put "FFFE" smack in the middle of the MAC, giving us 000f.f7ff.fec4.09c0.

Now take the first digit in the result, in this case "0", write it in binary, and change the 7th bit to a 1. The result is 00000010, which converts to the decimal "2".

That gives us 200f.f7ff.fec4.09c0. Does it match the router's result? I've already enabled ipv6 on that interface, and *show ipv6 interface fast 0/0* displays the link local

4/3/2018 8:36 AM - Screen Clipping

```
R1#show ipv6 int fast 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:F7FF:FEC4:9C0
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFC4:9C0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

IPV6 Address Types

You know the drill with IPv4 address types! *Unicasts* represent a single host, *multicasts* represent a group of hosts, and *broadcasts* represent all hosts. We still have unicasts and multicasts with IPv6, but broadcasts are gone, and we have a *new* address type. *Anycasts* are addresses that represent multiple interfaces, as does a multicast. The difference is that when an anycast is sent to a group of interfaces, it's sent to the interface that's considered *closest to the sender*.

How is "closest" defined for anycasts? It depends...

If there are directly connected neighbors, the closest one is the first one learned.

If there are no directly connected neighbors, the closest neighbor is determined by the routing protocol metric.

When an IPv6 *multicast* is sent to a group of interfaces, it's received by every member of the group, just as an IPv4 multicast would be.

IPv6 brings us different types of unicast IP addresses as well, including the *global unicast address*. This address is equivalent to the public IPv4 address class. These addresses are fully routable and can be used for Internet access. This range is 2000::/3, meaning any address that begins with 001 is a global unicast address.

The link-local address is an address that's kept on the local link. These addresses have a prefix of Fe80 (1111 1110 10)::/10, followed by the interface identifier.

Two more address types you can spot by their initial bits are multicasts (1111 1111) and *IPv4-compatible addresses*. Any IPv6 address with the first 96 bits set to zero is an IPv4-compatible address. Expressed with zero compression, that's ::x.x.x.x. Using only leading zero compression, that's 0:0:0:0:0:x.x.x.x.

The Reserved IPv6 Addresses

Just as IPv4 has 127.0.0.1 reserved for testing, IPv6 reserves 0000:0000:0000:0000:0000:0000:0001. Thankfully, we can express that address as 0:0:0:0:0:0:1 (with leading zero compression only), or ::1 using a combination of leading zero and zero compression. Compression's looking pretty

4/3/2018 8:41 AM - Screen Clipping

good right now!

Unique to IPv6 is the *unspecified address*, used to express an unknown address. The full address is 0000:0000:0000:0000:0000:0000, which thankfully we can express as 0:0:0:0:0:0 or ::/128. ("::/0" is the IPv6 default route.)

4/3/2018 8:42 AM - Screen Clipping

Multicasts And Anycasts

IPv4 multicast addresses are Class D addresses with a first octet of 224 – 239. The IPv6 multicast range is much larger but easier to remember. Any address that begins with 1111 1111, or “FF” in hex, is a multicast address. The full prefix is FF00::/8.

There are some local-link-only addresses in that range worth noting:

FF02::1 -- All nodes on the local link

FF02::2 – All routers on the local link

FF02::5 – All OSPF routers

FF02::6 – All OSPF DRs

FF02::9 – All RIP routers

FF02::A – All EIGRP routers

FF02::1:FFzz:zzzz/104 – *Solicited-node addresses*. These are used in Neighbor Solicitation messages, which we’ll examine in detail shortly. The “z”s represent the rightmost 24 bits of the unicast/address of the node.

To see the multicast and anycast groups joined by an interface, run `show ipv6 interface`.

```
R1#show ipv6 int fast 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:F7FF:FEC4:9C0
  Global unicast address(es):
    2014::1, subnet is 2014::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::1:FF00:1
    FF02::1:FFC4:9C0
```

The IPv6 Autoconfiguration Process

IPv6 offers two types of autoconfiguration, *stateless* and *stateful*. Stateful autoconfiguration has a host obtain an IPv6 address (along with other info) from a server. That likely sounds a lot like DHCP to you, and for good reason. Stateful autoconfiguration *is* DHCPv6! The key phrase here is "from a server". If the DHCPv6 server goes down, we're out of luck.

4/3/2018 8:46 AM - Screen Clipping



4/3/2018 8:47 AM - Screen Clipping

STATEFUL = HOST GETS IP FROM SERVER

StateLess Address AutoConfiguration (SLAAC) has no such dependence, and that entire process starts with an IPv6 host configuring its *own* link-local address.



4/3/2018 8:47 AM - Screen Clipping

STATELESS = HOST CONFIGURES ITS OWN LINK-LOCAL ADDRESS.

The first 64 bits of this self-generated address are 1111 1110 10 (FE80) followed by 54 zeroes. The last 64 bits are the interface identifier. This self-created address needs a little testing before we allow its use, though.

The *Duplicate Address Detection* (DAD) test consists of the host sending a

4/3/2018 8:48 AM - Screen Clipping

DAD is a test to make sure that we don't duplicate another address.

Neighbor Solicitation (NS) message to see if any other host on the same link is using that same link-local address. You can see the NS leaving the interface and the DAD status by running *debug ipv6 nd* before opening an interface (or closing it, running the debug, and re-opening it, as I did here).

```
ICMPv6-ND: Sending NS for FE80::20F:F7FF:FEC4:9C0 on FastEthernet0/0
ICMPv6-ND: DAD: FE80::20F:F7FF:FEC4:9C0 is unique.
Sending NA for FE80::20F:F7FF:FEC4:9C0 on FastEthernet0/0
Address FE80::20F:F7FF:FEC4:9C0/10 is up on FastEthernet0/0

ICMPv6-ND: Sending NS for 2001:1::1 on FastEthernet0/0
ICMPv6-ND: DAD: 2001:1::1 is unique.
ICMPv6-ND: Sending NA for 2001:1::1 on FastEthernet0/0
ICMPv6-ND: Address 2001:1::1/64 is up on FastEthernet0/0
```

I removed the timestamps to make this easier to read, so take my word that all this happened in about 12 milliseconds. First, the router sent an NS for its own link-local address.



4/3/2018 8:49 AM - Screen Clipping

Had R1 received a *Neighbor Advertisement* (NA) in response to that NS, alerting R1 the link-local address in question was already in use, R1 would have disabled that address. Since no such message was received, DAD determined the link-local address is unique, and R1 sent an NA of its own claiming that address.

R1 then went through the same process with its global address. R1 sent an NS with that address...

4/3/2018 8:50 AM - Screen Clipping

The host now sends a Router Solicitation (RS) to FF02::2, the “all-routers” multicast address. The host is soliciting additional configuration info from a router in the form of a Router Advertisement (RA), shown coming in here:

4/3/2018 8:50 AM - Screen Clipping

Host is sending an RS and router is sending an RA.

ICMPv6-ND: Received RA from FE80::21B:D4FF:FEC2:990 on FastEthernet0/0

Routers send these RAs periodically without being prodded by a client request, but even though the host would only have to wait 10 seconds or so, polling the router immediately upon need with an RS does speed the process up! The info in the Router Advertisement includes the following:

Flags indicating whether the host should use DHCP for addressing information.

If DHCP is in use, the RA tells the host where the DHCP Server is.

If DHCP is not in use, the RA contains the prefix and prefix lifetime information. The router will attach the network prefix to the host’s link-local address, resulting in the host’s full IPv6 address, complete with network prefix.

4/3/2018 8:51 AM - Screen Clipping

4/3/2018 8:48 AM - Screen Clipping

OSPFv3 On Cisco Routers

OSPF for IPv6 is also called OSPFv3, and since OSPFv3 is the term used throughout Cisco documentation, it's likely the term you'll see on the exam. Be ready for either and you're gold. Before we start configuring this protocol, let's see how it compares to OSPF for IPv4 (OSPFv2).

During a production network migration from v2 to v3, you may run both versions of OSPF on the same router. The two OSPF instances are kept as separate as they would be if you ran two instances of v2 on the same router.

With OSPFv3, you won't necessarily start a config with *ipv6 router ospf*. One major difference between v2 and v3 is that v3 is enabled on a per-interface basis, rather than the router config mode of v2. The following command actually starts an OSPF process in v3.

```
R1(config)#int fast 0/0
R1(config-if)#ipv6 ospf 1 area 0
```

There are similarities between the versions, starting with the RID. V3 will use the exact same set of rules as V2 does in RID determination, going as far as to use an IPv4 address! If there is no IPv4 address on the router, you'll need to use *router-id* to create the RID. The RID must be entered in IPv4 format, even if you're only

4/3/2018 8:55 AM - Screen Clipping
OSPF SET PER INTERFACE BASIS FOR OSPFV3.
YOU DON'T USE NETWORK COMMANDS.
OSPFV3 DOESN'T USE IPV6 FOR RID, ONLY IPV4.
RID HAS TO BE ENTERED IN IPV4 FORMAT.

The basic theory of v3 is quite similar to that of v2. Hellos, LSAs, and good ol' Area 0 are still around, as are stub, total stub, and not-so-stub stub areas. The general rules for neighbor discovery and adjacencies are the same, including the rule that the hub(s) in an NBMA network requires a neighbor statement.

Neither v3 nor v2 point-to-point and point-to-multipoint networks elect DRs nor BDRs.

A major version difference is v3 allowing a single link to be part of multiple OSPF instances, where v2 would allow a link to be a part of only one.

The v2 reserved address 224.0.0.5 is represented in v3 by FF02::5.

The v2 reserved address 224.0.0.6 is represented in v3 by FF02::6.

While we certainly have new addresses and slightly different commands to get used to, the theory remains much the same. Let's start working with those addresses and commands right now!

```
interface Loopback1
no ip address
ipv6 address 2010::1/64
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
ipv6 address 2015::1/64
ipv6 enable
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial1/0
no ip address
encapsulation frame-relay IETF
shutdown
frame-relay lmi-type cisco
```

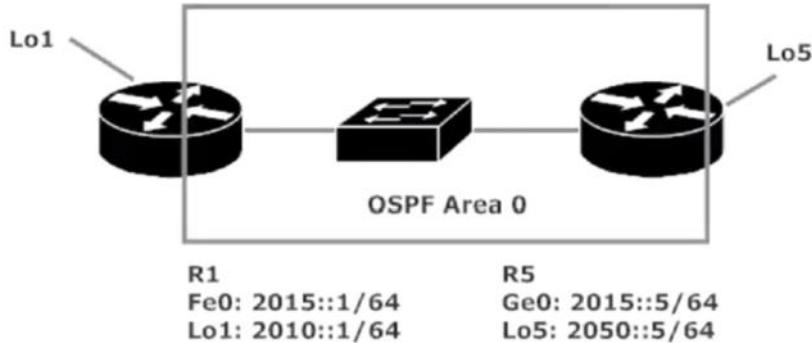
4/3/2018 8:59 AM - Screen Clipping

Router 1

```
interface Loopback5
no ip address
ipv6 address 2050::5/64
!
interface GigabitEthernet0/0
no ip address
duplex full
speed auto
media-type rj45
ipv6 address 2015::5/64
ipv6 enable
!
interface GigabitEthernet0/1
R5#
```

4/3/2018 9:00 AM - Screen Clipping

Router 5



4/3/2018 9:00 AM - Screen Clipping

These two routers have zero IPv4 addresses at present, so we'll get this little reminder when we enable OSPFv3:

```
R1(config)#ipv6 router ospf 1
*Jun 16 02:18:46.822: %OSPFV3-4-NORTRID: OSPFv3 process 1 could not pick a
route
r-id, please configure manually

R5(config)#ipv6 router ospf 1
*Jul  5 15:42:01: %OSPFV3-4-NORTRID: OSPFv3 process 1 could not pick a router-
id
```

4/3/2018 9:00 AM - Screen Clipping

Start with Router 5. Enable IPV6

```
R5(config)#ipv6 unicast-routing
R5(config)#
```

4/3/2018 9:02 AM - Screen Clipping

```
R5#conf t
Enter configuration commands, or
R5(config)#int loopback5
R5(config-if)#ipv6 enable
R5(config-if)#int gig 0/0
```

4/3/2018 9:01 AM - Screen Clipping

Notice you add the area after the process number immediately after.

```
R5(config-if)#ipv6 ospf 1 area ?
<0-4294967295>  OSPF area ID as a decimal value
A.B.C.D          OSPF area ID in IP address format
R5(config-if)#ipv6 ospf 1 area 0
```

4/3/2018 9:01 AM - Screen Clipping

Create OSPFv3

```
R5(config-if)#ipv6 ospf 1 area 0
R5(config-if)#^Z
R5#
*Aug 18 16:44:47.962: %OSPFv3-4-NORTRID: OSPFv3 process 1 could no
r-id,
please configure manually
R5#
*Aug 18 16:44:49.274: %SYS-5-CONFIG_I: Configured from console by
R5#
```

4/3/2018 9:02 AM - Screen Clipping

Console message is stating that we cant form adjacency because theres no RID. You need to manually configure an IPV4 address for the RID.

```
R5(config)#ipv6 router ospf 1
R5(config-rtr)#
      area          OSPF
      auto-cost    Calculat
      default       Set a
      default-information Distr
      default-metric   Set m
      discard-route   Enabl
      distance        Admin
```

4/3/2018 9:03 AM - Screen Clipping

```
router-id          router-id for this OSPF process
```

```
R5(config-rtr)#router-id ?
  A.B.C.D  OSPF router-id in IP address format
R5(config-rtr)#router-id 5.5.5.5 ?
<cr>
R5(config-rtr)#router-id 5.5.5.5
R5(config-rtr)#^Z
```

4/3/2018 9:04 AM - Screen Clipping

No do it on router 1

```
R1(config)#ipv6 unicast-routing
R1(config)#int fast 0/0
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#

```

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#
*Jul 16 09:04:00.773: %OSPFv3-5-ADJCHG: Process 1, Nbr 5.5.5.5 on FastEthe
0 from LOADING to FULL, Loading Done
R1(config-rtr)#

```

4/3/2018 9:04 AM - Screen Clipping
We now have our OSPFv3 adjacency.

Now verify. **Sho ipv6 ospf nei**

R1#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID
5.5.5.5	1	FULL/DR	00:00:33	2
0				

4/3/2018 9:06 AM - Screen Clipping

R1#show ipv6 ospf neighbor detail

Neighbor 5.5.5.5
In the area 0 via interface FastEthernet0/0
Neighbor: interface-id 2, link-local address FE80::216:9DFF:FEF5
Neighbor priority is 1, State is FULL, 6 state changes
DR is 5.5.5.5 BDR is 1.1.1.1
Options is 0x000013 in Hello (V6-Bit E-Bit R-bit)
Options is 0x000013 in DBD (V6-Bit E-Bit R-bit)
Dead timer due in 00:00:37
Neighbor is up for 00:01:44
Index 1/1/1, retransmission queue length 0, number of retransmissions 0
First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
Last retransmission scan length is 0, maximum is 0
Last retransmission scan time is 0 msec, maximum is 0 msec

4/3/2018 9:06 AM - Screen Clipping

R1#show ipv6 ospf

Routing Process "ospfv3 1" with ID 1.1.1.1
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msec
Retransmission pacing timer 66 msec
Number of external LSA 0. Checksum Sum 0x0000000
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Reference bandwidth unit is 100 mbps
Area BACKBONE(0)
Number of interfaces in this area is 1
SPF algorithm executed 1 times
Number of LSA 6. Checksum Sum 0x0300CA
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

4/3/2018 9:07 AM - Screen Clipping

```
R1#show ipv6 interface
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:F7FF:FE4:9C0
    Global unicast address(es):
      2015::1, subnet is 2015::/64
    Joined group address(es):
      FF02::1
      FF02::2
      FF02::5
      FF02::6
      FF02::1:FF00:1
      FF02::1:FFC4:9C0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
```

4/3/2018 9:08 AM - Screen Clipping

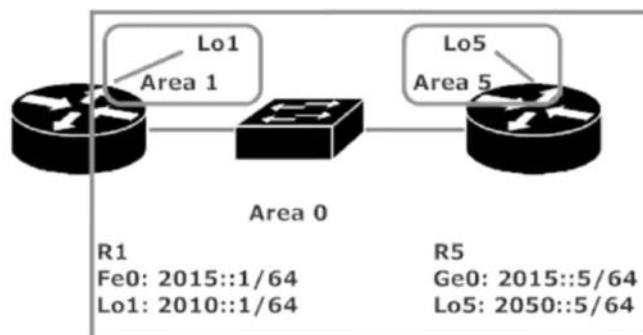
```
Loopback1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:F7FF:FE4:9C0
    Global unicast address(es):
      2010::1, subnet is 2010::/64
    Joined group address(es):
      FF02::1
      FF02::2
      FF02::1:FF00:1
      FF02::1:FFC4:9C0
  MTU is 1514 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

4/3/2018 9:08 AM - Screen Clipping

IPV6 5: Adding Loopbacks to OSPFv3 Network

Tuesday, April 3, 2018 9:06 AM

Let's get those interfaces enabled and place them into non-backbone areas.



```
R1(config)#int loopback1  
R1(config-if)#ipv6 ospf ?  
    1 65535... Press
```

```
|R1(config-if)#ipv6 ospf 1 area 1
```

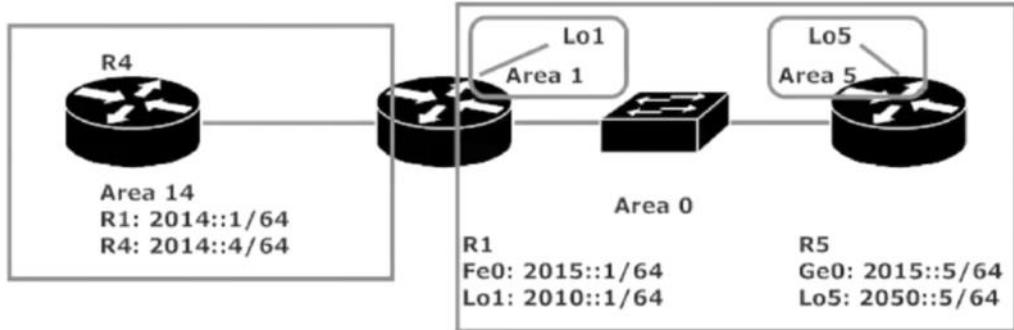
```
R5#conf t  
Enter configuration commands, one per line  
R5(config)#int loopback5  
R5(config-if)#ipv6 ospf 1 area 5  
R5(config-if)#^Z  
R5#
```

```
R5#show ipv6 route ospf  
IPv6 Routing Table - default - 6 entries  
Codes: C - Connected, L - Local, S - Static  
      B - BGP, HA - Home Agent, M - Manual  
      I1 - ISIS L1, I2 - ISIS L2,  
      D - EIGRP, EX - EIGRP external  
      O - OSPF Intra, OI - OSPF Inter  
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2  
O 2010::1/128 [110/1]  
  via FE80::20F:F7FF:FEC4:9C0,  
R5#ping 2010::1
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to  
!!!!!  
Success rate is 100 percent (5/5),  
R5#
```

4/3/2018 9:13 AM - Screen Clipping

Now lets add Area 14



R1#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
5.5.5.5	1	FULL/DR	00:00:39	2	FastEthernet0/0
4.4.4.4	1	FULL/ -	00:00:30	5	Serial1/1

R4#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
-------------	-----	-------	-----------	--------------	-----------

4/3/2018 9:14 AM - Screen Clipping

```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#ipv6 router ospf 1
R4(config-rtr)#router-id
*Jan 1 01:48:47.111: %OSPFv3-4-NOR
r-id, please configure manually
R4(config-rtr)#router-id 4.4.4.4
R4(config-rtr)#

```

4/3/2018 9:15 AM - Screen Clipping

```
*Jan 1 01:48:47.111: %OSPFv3-4-NOR
r-id, please configure manually
R4(config-rtr)#router-id 4.4.4.4
R4(config-rtr)#int serial 0/1/0
R4(config-if)#ipv6 enable
R4(config-if)#ipv6 ospf 1 area 14
R4(config-if)#^Z
R4#wr

```

4/3/2018 9:15 AM - Screen Clipping

```

R1#conf t
Enter configuration commands, one per line. End with
R1(config)#int serial 1/1
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 ospf 1 area 14
R1(config-if)#^Z
R1#
*Jul 16 06:20:40.247: %OSPFV3-5-ADJCHG: Process 1, Nb
  LOADING to FULL, Loading Done
*Jul 16 06:20:40.780: %SYS-5-CONFIG_I: Configured fro
R1#

```

4/3/2018 9:15 AM - Screen Clipping

Neighbor ID	Pri	State	Dead Time	Interface ID
5.5.5.5	1	FULL/DR	00:00:36	2
0				
4.4.4.4	1	FULL/ -	00:00:36	5

4/3/2018 9:15 AM - Screen Clipping

We never have a DR/BDr in a point to point network. The two routers are literally connected so there's no need to have an election between two directly connected routers.

```

R1#show ipv6 ospf int serial 1/1
Serial1/1 is up, line protocol is up
  Link Local Address FE80::20F:F7FF:FEC4:9C0, Interface ID 7
  Area 14, Process ID 1, Instance ID 0, Router ID 1.1.1.1
  Network Type POINT_TO_POINT, Cost: 781
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Ret
    Hello due in 00:00:04
  Index 1/1/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 3, maximum is 3
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 4.4.4.4
  Suppress hello for 0 neighbor(s)

```

4/3/2018 9:16 AM - Screen Clipping

```
R4#show ipv6 ospf int serial 0/1/0
Serial0/1/0 is up, line protocol is up
  Link Local Address FE80::217:59FF:FE2:474A, Interface ID 5
  Area 14, Process ID 1, Instance ID 0, Router ID 4.4.4.4
  Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retr
    Hello due in 00:00:09
  Graceful restart helper support enabled
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1
  Suppress hello for 0 neighbor(s)
```

```
R4#show ipv6 route ospf
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
      B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
      D - EIGRP, EX - EIGRP external, NM - NEMO, ND - Neighbor Discovery
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI  2010::1/128 [110/64]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2015::/64 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2050::5/128 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
R4#udemv
```

4/3/2018 9:17 AM - Screen Clipping

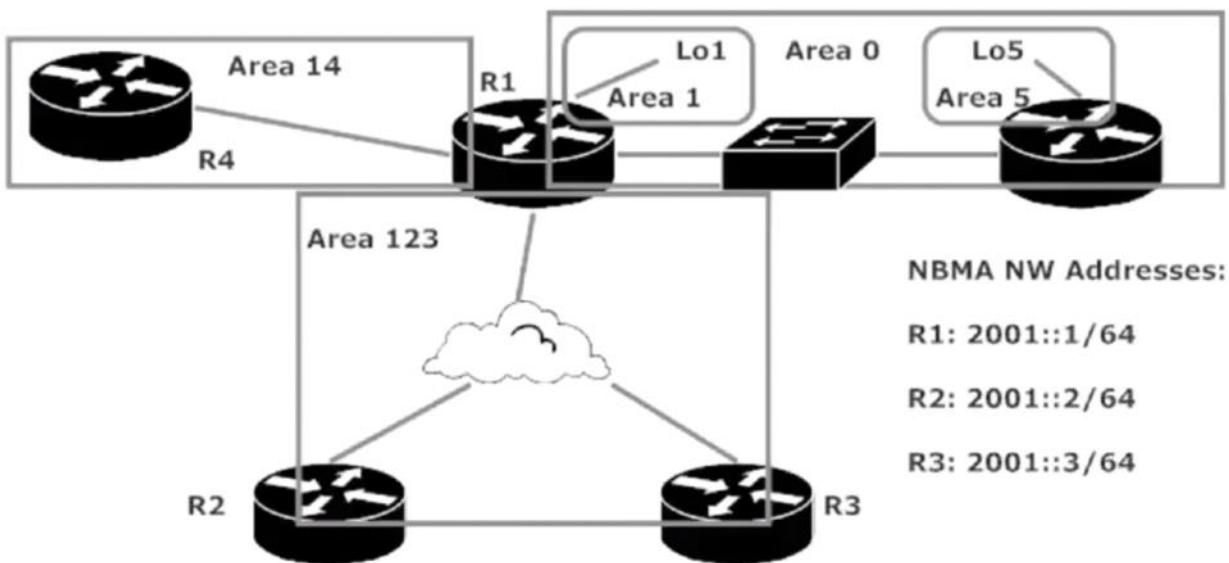
Verify

```
R4#ping 2010::1
Type escape sequence (press Ctrl-C now): Sending 5, 100-byte datagram(s) to 2010::1
!!!!!
Success rate is 100.00% (5/5)
R4#ping 2050::5
Type escape sequence (press Ctrl-C now): Sending 5, 100-byte datagram(s) to 2050::5
!!!!!
Success rate is 100.00% (5/5)
R4#ping 2015::1
Type escape sequence (press Ctrl-C now): Sending 5, 100-byte datagram(s) to 2015::1
!!!!!
```

IPV6 6: Configuring an OSPF NBMA Network

Tuesday, April 3, 2018 9:10 AM

R4 can ping all destinations listed in that table, so we're gold. Let's raise the stakes and configure an NBMA network by adding R2 and R3 back to the network. To squeeze in our NBMA network, I've removed the previously listed IPv6 addresses from the diagram. No addresses have been changed in the lab.



```
R2#conf t
Enter configuration commands, one at a time
R2(config)#ipv6 unicast-routing
R2(config)#ipv6 router ospf 1
R2(config-rtr)#router-id 2.2.2.2
*Aug 19 19:52:14.566: %OSPFv3-4-NODATA-RID:1: Router ID not specified, please configure manually
R2(config-rtr)#router-id 2.2.2.2
R2(config-rtr)#int serial 0/1/0
R2(config-if)#ipv6 ospf pri 0
R2(config-if)#^Z
R2#wr
Building configuration...
```

```
R3#conf t
Enter configuration commands, one per line
R3(config)#ipv6 unicast-routing
R3(config)#ipv6 router ospf 1
R3(config-rtr)#router-id
*Aug 19 19:39:13.074: %OSPFV3-4-N
r-id, please configure manually
R3(config-rtr)#router-id 3.3.3.3
R3(config-rtr)#
R3(config-rtr)#int serial 0/1/0
R3(config-if)#ipv6 ospf pri 0
R3(config-if)#^Z
R3#wr
```

Configure Router 2 and 3.

- unicast routing
- router ospf
- Router-id
- interface

```
R1#
R1#conf t
Enter configuration commands, one per line
R1(config)#int serial 1/0
R1(config-if)#ipv6 ospf neighbor ?
    X:X:X::X Neighbor IPv6 address

R1(config-if)#ipv6 ospf neighbor 2001::2
OSPFv3: Neighbor address needs to be a link-local address
R1(config-if)#

```

Create link local address.

```
R1(config)#int serial 1/0
R1(config-if)#ipv6 ospf neighbor ?
    X:X:X::X Neighbor IPv6 address

R1(config-if)#ipv6 ospf neighbor 2001::2
OSPFv3: Neighbor address needs to be a link-local address
R1(config-if)#ipv6 ospf neighbor FE80::21B:D4FF:FEC2:990
R1(config-if)#ipv6 ospf neighbor FE80::21F:CAFF:FE96:2754
R1(config-if)#ipv6 ospf 1 area 123
R1(config-if)#^Z
R1#wr
Building configuration...
```

```

R2#
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int serial 0/1/0
R2(config-if)#ipv6 ospf 1 area 123
R2(config-if)#^Z
R2#wr
Building configuration...

```

```
R1#show ipv6 ospf neighbor
```

Neighbor	ID	Pri	State	Dead Time	Interf
5.5.5.5		1	FULL/DR	00:00:36	2
0					
4.4.4.4		1	FULL/ -	00:00:39	5
2.2.2.2		0	FULL/DROTHER	00:01:46	5
3.3.3.3		0	FULL/DROTHER	00:01:46	5
R1#					

Complete.

Now look at Router 2 OSPF table

```

R2#show ipv6 route ospf
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U
      B - BGP, HA - Home Agent, MR - Mobile Re
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS i
      D - EIGRP, EX - EIGRP external, NM - NE
      O - OSPF Intra, OI - OSPF Inter, OE1 - O
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA e
OI  2010::1/128 [110/64]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2014::/64 [110/845]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2015::/64 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2050::5/128 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
R2#ping 2010::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2010::1, time 1000ms
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R2#

```


IPV6 7: Route Redistribution and Stub Routing with OSPF

Wednesday, April 4, 2018 8:41 AM

Looks good! We've now built NBMA, point-to-point, and broadcast networks in OSPFv3. Let's do just a *little* route redistribution by creating three new loopbacks on R1 and then redistributing them into OSPF. Note the v3 *redistribute connected* command doesn't require the *subnets* option. (It's not even available!)

```
interface Loopback13
  no ip address
  ipv6 address 2013::1/64
!
interface Loopback12
  no ip address
  ipv6 address 2012::1/64
!
interface Loopback16
  no ip address
  ipv6 address 2016::1/64

R1(config)#ipv6 router ospf 1
R1(config-rtr)#redistribute connected ?
  metric      Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redistributed routes
  route-map   Route map reference
  tag         Set tag for routes redistributed into OSPF
<cr>

R1(config-rtr)#redistribute connected
```

4/4/2018 8:41 AM - Screen Clipping

```
interface Loopback1
  no ip address
  ipv6 address 2010::1/64
  ipv6 ospf 1 area 1
!
interface Loopback12
  no ip address
  ipv6 address 2012::1/64
!
interface Loopback13
  no ip address
  ipv6 address 2013::1/64
!
interface Loopback16
  no ip address
  ipv6 address 2016::1/64
!
interface FastEthernet0/0
  no ip address
```

4/4/2018 8:42 AM - Screen Clipping

Redistribute these loopbacks into IPV6 OSPF.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ipv6 router ospf 1
R1(config-rtr)#redistribute ?
  bgp      Border Gateway Protocol (BGP)
  connected  Connected Routes
  isis     ISO IS-IS
  ospf     Open Shortest Path First (OSPF)
  rip      IPv6 Routing Information Protocol (RIPv6)
  static    Static Routes

R1(config-rtr)#redistribute connected ?
  metric    Metric for redistributed routes
  metric-type OSPF/IS-IS exterior metric type for redistribu
  route-map Route map reference
  tag       Set tag for routes redistributed into OSPF
<cr>

R1(config-rtr)#redistribute connected
```

4/4/2018 8:42 AM - Screen Clipping

Subnets aren't necessary in OSPFv3. You don't need to set a seed metric. All you need to do is connect.

```
R4#show ipv6 route ospf
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U
      B - BGP, HA - Home Agent, MR - Mobile R
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS i
      D - EIGRP, EX - EIGRP external, NM - NE
      O - OSPF Intra, OI - OSPF Inter, OE1 -
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA
OI  2001::/64 [110/845]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2010::1/128 [110/64]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OE2 2012::/64 [110/20]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OE2 2013::/64 [110/20]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2015::/64 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OE2 2016::/64 [110/20]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2050::5/128 [110/65]
    via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
R4#
```

4/4/2018 8:43 AM - Screen Clipping

Notice the OE2 routes. All the routes have the same next hop link local address. This is a good case for creating a stub area.

Lets make area 14 a stub area on router 4.

```

R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#ipv6 router ospf 1
R4(config-rtr)#area 14 ?
  authentication      Enable authentication
  default-cost        Set the summary default-cost of a NSSA/stub
  encryption          Enable encryption
  nssa                Specify a NSSA area
  range               Summarize routes matching address/mask (border)
  stub                Specify a stub area
  virtual-link        Define a virtual link and its parameters

R4(config-rtr)#area 14 stub
R4(config-rtr)#^Z
R4#
Aug 19 15:58:28.190: %OSPFV3-5-ADJCHG: Process 1, Nbr 1.1.1.1
  from FULL to DOWN, Neighbor Down: Adjacency forced to reset
Aug 19 15:58:28.698: %SYS-5-CONFIG_I: Configured from console
R4#wr
Building configuration...

```

4/4/2018 8:44 AM - Screen Clipping

Router 4 is now seeing area 14 as a stub area but router 1 doesn't.

```

R1(config)#ipv6 router ospf 1
R1(config-rtr)#area 14 stub
R1(config-rtr)#
Aug 19 15:58:52.121: %OSPFV3-5-ADJCHG: Process 1, Nbr 4.4.4.4
  from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-rtr)#
Aug 19 15:58:57.586: %OSPFV3-5-ADJCHG: Process 1, Nbr 4.4.4.4
  from LOADING to FULL, Loading Done
R1(config-rtr)#^Z
R1#
Aug 19 15:58:59.461: %SYS-5-

```

4/4/2018 8:45 AM - Screen Clipping

Now the adjacency is back up.

Go back to router 4.

```

R4#show ipv6 route ospf
IPv6 Routing Table - default - 8 entries
Codes: C - Connected, L - Local, S - Static, U
      B - BGP, HA - Home Agent, MR - Mobile R
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS i
      D - EIGRP, EX - EIGRP external, NM - NE
      O - OSPF Intra, OI - OSPF Inter, OE1 -
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA
OI  ::/0 [110/65]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2001::/64 [110/845]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2010::1/128 [110/64]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2015::/64 [110/65]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
OI  2050::5/128 [110/65]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
R4#

```

4/4/2018 8:45 AM - Screen Clipping

OE2 routes are now gone and replaced by OI. We can ping. We've made this into a stub area, but still have four routes with the same link local address. We can actually make this a **total stub**. All you have to do is go to the border router and issue the no summary command.

```

R1#conf t
Enter configuration commands, one per line. End
R1(config)#ipv6 router ospf 1
R1(config-rtr)#area 14 stub ?
  no-summary  Do not send summary LSA into stub a
<cr>

R1(config-rtr)#area 14 stub no-summary
R1(config-rtr)#^Z
R1#
Aug 19 16:00:52.029: %SYS-5-CONFIG_I: Configured
R1#

```

4/4/2018 8:47 AM - Screen Clipping

Stub flag is still set so we won't lose the adjacency, but check out router 4's routing table.

```
R4#show ipv6 route ospf
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-
      B - BGP, HA - Home Agent, MR - Mobile Router,
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interare
      D - EIGRP, EX - EIGRP external, NM - NEMO, ND
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ex
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI  ::/0 [110/65]
      via FE80::20F:F7FF:FEC4:9C0, Serial0/1/0
R4#
```

4/4/2018 8:47 AM - Screen Clipping

Now we only have one route listed. We now use less resources. The total stub configuration is a success.
We've used redis and OSPF for ipv6 including stub and total stubs.

```
R4#show ipv6 ospf
Routing Process "ospfv3 1" with ID 4.4.4.4
Event-Log enabled, Maximum number of events: 100
Initial SPF schedule delay 5000 msec
Minimum hold time between two consecutive SPFs
Maximum wait time between two consecutive SPFs
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msec
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msec
Retransmission pacing timer 66 msec
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 1. 0 normal 1
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
  Area 14
    Number of interfaces in this area is 1
    It is a stub area
```

4/4/2018 8:48 AM - Screen Clipping

IPV6 8: Configuring EIGRP for IPv6/ Route Redis

Wednesday, April 4, 2018 8:49 AM

Configuring EIGRP For IPV6 On Cisco Routers

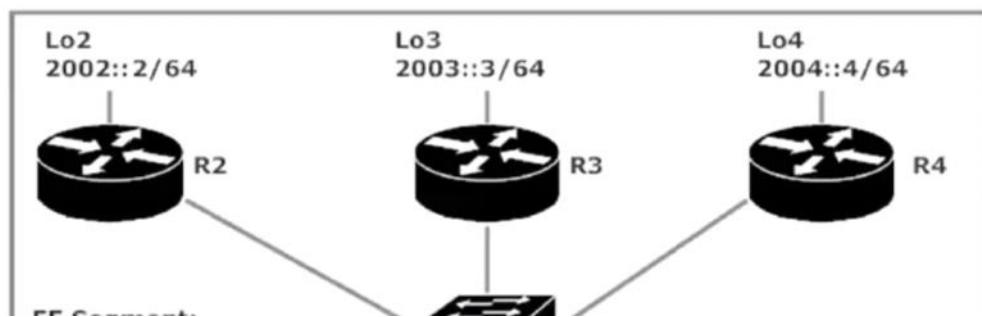
All OSPF-related commands have been removed from all routers. The fundamental IPv6 commands are still running. We'll start by setting an EIGRP RID for R2, R3, and R4.

```
R2(config)#ipv6 router eigrp 100  
R2(config-rtr)#router-id 2.2.2.2
```

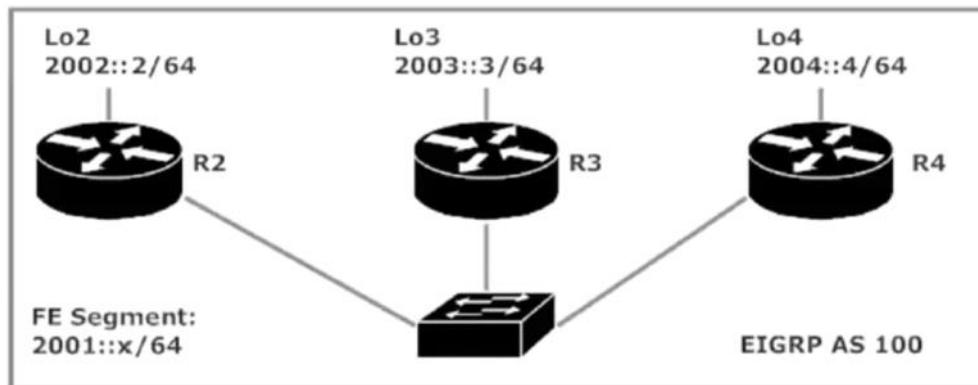
```
R3(config)#ipv6 router eigrp 100  
R3(config-rtr)#router-id 3.3.3.3
```

```
R4(config)#ipv6 router eigrp 100  
R4(config-rtr)#router-id 4.4.4.4
```

We'll build the following network in this lab:



4/4/2018 8:50 AM - Screen Clipping



4/4/2018 8:50 AM - Screen Clipping

Each router can ping the other two router addresses on the 2001::x/64 network, but nothing doing with pinging the loopbacks. Let's fix that right now! First, we need adjacencies over the FE interfaces.

```
R2(config)#int fast 0/0
R2(config-if)#ipv6 eigrp 100
```

```
R3(config)#int fast 0/0
R3(config-if)#ipv6 eigrp 100
```

```
R4(config)#int fast 0/0
R4(config-if)#ipv6 eigrp 100
```

4/4/2018 8:52 AM - Screen Clipping

Remember like OSPF, EIGRP is applied to the interface not the local router.

```
R3#show ipv6 eigrp neighbor
EIGRP-IPv6 Neighbors for AS(100)
H   Address           Interface      Hold Uptime    SRTT     RTO
    (sec)             (ms)          1       200
1   Link-local address: Fa0/0        13 00:00:10  1       200
    FE80::21B:D4FF:FEC2:990
0   Link-local address: Fa0/0        13 00:00:10  9       200
    FE80::217:59FF:FE2:474A
R3#
```

4/4/2018 8:53 AM - Screen Clipping

```
R2#show ipv6 route eigrp
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static
       B - BGP, HA - Home Agent, MR - Multicast
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS Inter-Area
       D - EIGRP, EX - EIGRP external, N - Null
       O - OSPF Intra, OI - OSPF Inter,
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R2#show ipv6 eigrp neighbor
EIGRP-IPv6 Neighbors for AS(100)
H   Address           Interface
```

H	Address	Interface
1	Link-local address: Fa0/0 FE80::21F:CAFF:FE96:2754	
0	Link-local address: Fa0/0	

4/4/2018 8:53 AM - Screen Clipping

We don't have any routes because were directly connected. In order to have routes we need to create loopbacks in AS 100.

```
R4#conf t
Enter configuration commands, one
R4(config)#int loopback4
R4(config-if)#ipv6 eigrp 100
R4(config-if)#^Z
R4#wr
Building configuration...
```

4/4/2018 8:54 AM - Screen Clipping

```
R2#show ipv6 route eigrp
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Pe
      B - BGP, HA - Home Agent, MR - Mobile Router
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS intera
      D - EIGRP, EX - EIGRP external, NM - NEMO, N
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D 2003::/64 [90/156160]
  via FE80::21F:CAFF:FE96:2754, FastEthernet0/0
D 2004::/64 [90/156160]
  via FE80::217:59FF:FE2:474A, FastEthernet0/0
```

4/4/2018 8:54 AM - Screen Clipping

Now we can see the loopbacks for routers 3 and 4. Were also running EIGRP on those routes.

Now lets redistribute!

What do we have todo with EIGRP? Set a default metric. (bandwidth, delay, reliability, MTU)

```
R2(config-rtr)#default-metric 1544 10 100 1 1500
R2(config-rtr)#redistribute ?
  bgp      Border Gateway Protocol (BGP)
  connected  Connected Routes
  eigrp    Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis     ISO IS-IS
  nemo     Network Mobility (NEMO)
  ospf     Open Shortest Path First (OSPF)
  rip      IPv6 Routing Information Protocol (RIPV6)
  static   Static Routes

R2(config-rtr)#redistribute connected ?
  metric   Metric for redistributed routes
  route-map Route map reference
<cr>

R2(config-rtr)#redistribute connected
```

4/4/2018 8:56 AM - Screen Clipping

Now check the route table for eigrp

```
R3#show ipv6 route eigrp
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Pe
      B - BGP, HA - Home Agent, MR - Mobile Router
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS intera
      D - EIGRP, EX - EIGRP external, NM - NEMO, ND
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF e
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
EX  2002::/64 [170/1662976]
      via FE80::21B:D4FF:FEC2:990, FastEthernet0/0
D   2004::/64 [90/156160]
      via FE80::217:59FF:FE02:474A, FastEthernet0/0
R3#
```

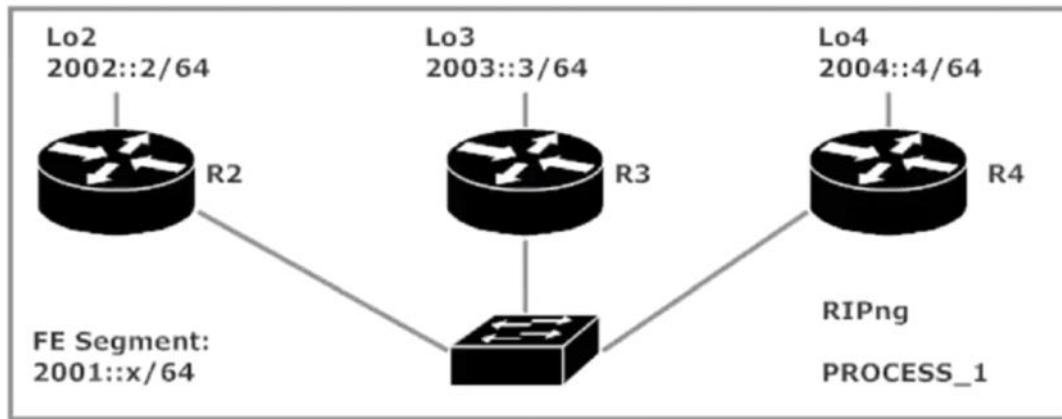
4/4/2018 8:57 AM - Screen Clipping

Notice theres no DEX.

IPV6 9: RIP for IPv6

Wednesday, April 4, 2018 8:59 AM

We'll use the same topology as we had in the EIGRPv6 lab.



With our global IPV6 commands in place, let's enable RIPng – *RIP For The Next Generation!*

```
R2(config)#int fast 0/0
R2(config-if)#ipv6 rip ?
WORD User selected string identifying this RIP process
```

4/4/2018 8:59 AM - Screen Clipping

Select the NAME of the Process.

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/D
R2(config)#int fast 0/0
R2(config-if)#ipv6 rip ?
WORD User selected string identifying this RIP process

R2(config-if)#ipv6 rip PROCESS1 ?
default-information Configure handling of default route
enable           Enable/disable RIP routing
metric-offset    Adjust default metric increment
summary-address Configure address summarization

R2(config-if)#ipv6 rip PROCESS1 enable
R2(config-if)#

```

4/4/2018 9:00 AM - Screen Clipping

Do the same on the other two.

```
R4#conf t
Enter configuration commands, one per line.
R4(config)#int fast 0/0
R4(config-if)#ipv6 rip PROCESS1 enable
R4(config-if)#^Z
R4#wr
Building configuration...
```

Now do the same on the loopbacks.

```
R3#conf t
Enter configuration commands, one per line.
R3(config)#int loopback3
R3(config-if)#ipv6 rip PROCESS1 enable
R3(config-if)#^Z
R3#wr
```

```
R2#conf t
Enter configuration commands, one per line.
R2(config)#int loopback2
R2(config-if)#ipv6 rip PROCESS1 enable
R2(config-if)#^Z
R2#
```

4/4/2018 9:02 AM - Screen Clipping

```
R2#show ipv6 route rip
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-
      B - BGP, HA - Home Agent, MR - Mobile Router,
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interare
      D - EIGRP, EX - EIGRP external, NM - NEMO, ND
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ex
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R    2003::/64 [120/2]
      via FE80::21F:CAFF:FE96:2754, FastEthernet0/0
R    2004::/64 [120/2]
      via FE80::217:59FF:FEE2:474A, FastEthernet0/0
R2#
```

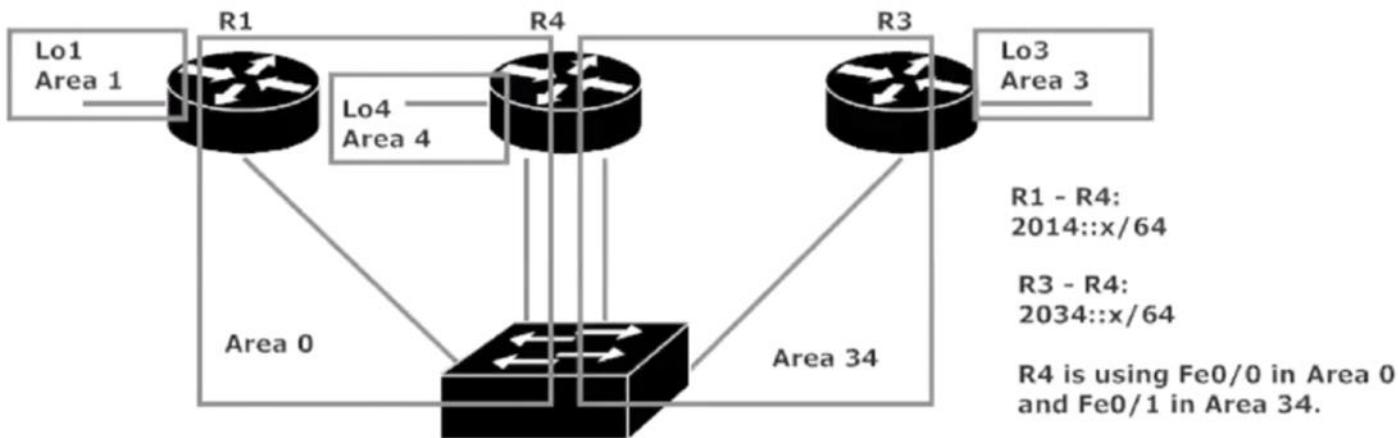
```
R2#show ipv6 rip
RIP process "PROCESS1", port 521, multicast-group FF02::9, pid 82
    Administrative distance is 120. Maximum paths is 16
    Updates every 30 seconds, expire after 180
    Holddown lasts 0 seconds, garbage collect after 120
    Split horizon is on; poison reverse is off
    Default routes are not generated
    Periodic updates 5, trigger updates 3, Full Advertisement 0
Interfaces:
    Loopback2
    FastEthernet0/0
Redistribution:
```

4/4/2018 9:02 AM - Screen Clipping

IPV6 9: OSPFv3 Troubleshooting

Wednesday, April 4, 2018 9:03 AM

Let's add an IPv6 address to the loopbacks on each router and then put them in their own individual OSPF area.



```
R1(config)#int loopback1
R1(config-if)#ipv6 address 2001::1/64
R1(config-if)#ipv6 ospf 1 area 1
```

```
R3#show ipv6 route ospf
IPv6 Routing Table - default - 8 e
Codes: C - Connected, L - Local, S
      B - BGP, HA - Home Agent, M
      I1 - ISIS L1, I2 - ISIS L2,
      D - EIGRP, EX - EIGRP exte
      O - OSPF Intra, OI - OSPF I
      ON1 - OSPF NSSA ext 1, ON2
OI  2001::1/128 [110/2]
    via FE80::217:59FF:FE02:474B,
OI  2004::1/128 [110/1]
    via FE80::217:59FF:FE02:474B,
OI  2014::/64 [110/2]
    via FE80::217:59FF:FE02:474B,
```

```
R4#show ipv6 route ospf
IPv6 Routing Table - default - 8
Codes: C - Connected, L - Local,
       B - BGP, HA - Home Agent,
       I1 - ISIS L1, I2 - ISIS L2
       D - EIGRP, EX - EIGRP exte
       O - OSPF Intra, OI - OSPF
       ON1 - OSPF NSSA ext 1, ON2
OI  2001::1/128 [110/1]
    via FE80::20F:F7FF:FEC4:9C0,
R4#
```

Route 4 only has one route because router 3 is connected. However we should see router 3's loopback.

```
R1#show ipv6 route ospf
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP,
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interare
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ex
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI  2004::1/128 [110/1]
    via FE80::217:59FF:FE02:474A, FastEthernet0/0
OI  2034::/64 [110/2]
    via FE80::217:59FF:FE02:474A, FastEthernet0/0
R1#
```

Router 1 can see 34 which is the shared segment between the routers. However, were still not seeing router 3's loopback.

Why? Area 3 doesn't have a physical connection to anything in Area0. Therefore, we need to create a virtual link.

The key is an OSPF ipv4 virtual link.

```
R4#conf t
Enter configuration commands, one per line.
R4(config)#ipv6 router ospf 1
R4(config-rtr)#area 34 ?
    authentication  Enable authentication
    default-cost   Set the summary default-co
    encryption     Enable encryption
    nssa           Specify a NSSA area
    range          Summarize routes matching
    stub            Specify a stub area
    virtual-link   Define a virtual link and

R4(config-rtr)#area 34 virtual-link ?
    A.B.C.D RouterID associated with virtual

R4(config-rtr)#area 34 virtual-link 3.3.3.3
R4(config-rtr)#^Z
R4#wr
Building configuration...
```

```
R3(config)#ipv6 router ospf 1
R3(config-rtr)#area ?
<0-4294967295> OSPF area ID as a decimal
A.B.C.D OSPF area ID in IP address

R3(config-rtr)#area 34 ?
authentication Enable authentication
default-cost Set the summary default-co
encryption Enable encryption
nssa Specify a NSSA area
range Summarize routes matching
stub Specify a stub area
virtual-link Define a virtual link and

R3(config-rtr)#area 34 virtual-link ?
A.B.C.D RouterID associated with virtual

R3(config-rtr)#area 34 virtual-link 4.4.4.4
R3(config-rtr)#^Z
R3#
*Aug 21 15:02:55.223: %SYS-5-CONFIG_I: Configured
R3#show ipv6
*Aug 21 15:03:00.023: %OSPFV3-5-ADJCHG: Procedure
m LOADING to FULL, Loading Done
R3#
```

```
R3#show ipv6 ospf virtual-link
Virtual Link OSPFv3_VL1 to router 4.4.4.4 is up
  Interface ID 11, IPv6 address 2034::4
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 34, via interface FastEthernet0/0, Cost of using
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retra
    Adjacency State FULL (Hello suppressed)
    Index 1/1/2, retransmission queue length 0, number of retr
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
R3#
```

IPV6 10: Migration

Wednesday, April 4, 2018 9:12 AM

4/4/2018 9:13 AM - Screen Clipping

STACKING

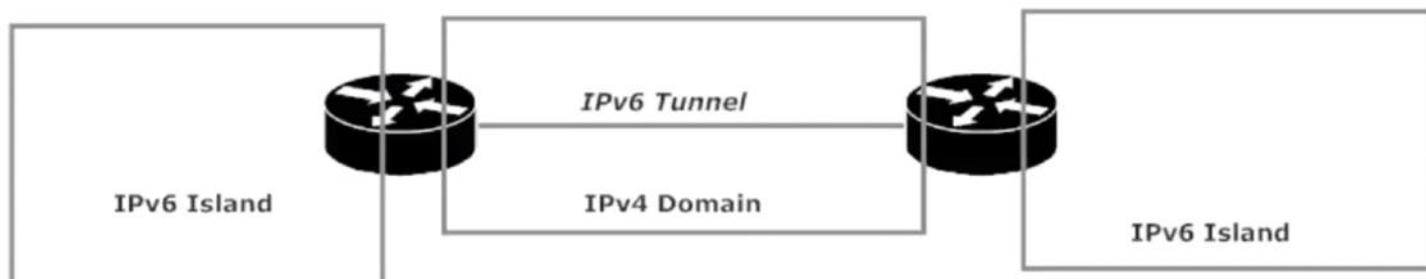
Theory holds that an IPv6 rollout starts at the network edge and works toward the core. That means you and I (the you-know-whos) have to make IPv6 and IPv4 play together nicely as we move toward an all-IPv6 network. This migration can involve *stacking, tunneling, or translating*.

Stacking – *dual stacking*, that is – occurs when you’re running IPv4 and v6 simultaneously across your entire network. You got your v6-to-v6 connections and your v4-to-v4 connections, and all is well! IPv4 hosts don’t have to run IPv6 to get their job done, and IPv6 hosts don’t have to run IPv4 to do their work.

The only real issue with dual stacking is likely the reason you’re not doing it on your network right now, and that’s the fact that your network likely needs some upgrades to hardware and / or software. The move from IPv4 to IPv6 is usually going to require a touch of network redesign for the new addressing scheme. In short, it’s a rare IPv4 network that can go straight to IPv6.

6-TO-4 TUNNEL

The *6-to-4 tunnel* is built when needed, torn down when not needed, and is highly scalable. Sounds great so far! 6-to-4 tunneling is accomplished by taking an IPv6 packet and encapsulating it with an IPv4 packet. This allows transport of the IPv6 packet across an IPv4 section of the network. The packet is then de-encapsulated when the time comes for it to be IPv6-routed. The IPv6 networks separated by an IPv4 core are sometimes called *IPv6 islands*.



You can quickly see one major issue with tunneling – IPv6 hosts can't communicate with IPv4 hosts or access IPv4-based services unless you get dual stacking involved.

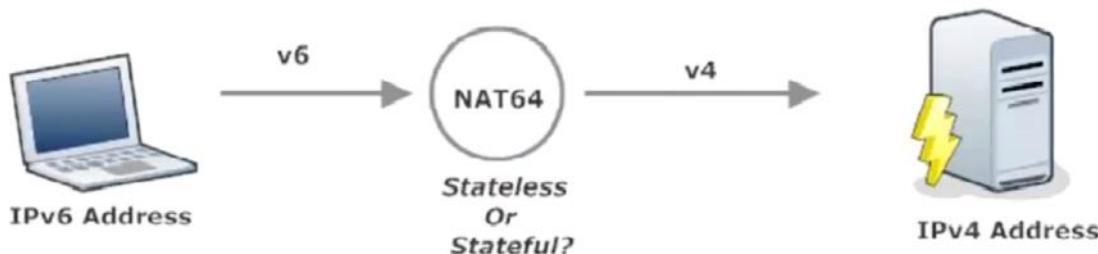
6-to-4 tunnels have a reserved IPV6 address prefix for edge routers, such as those shown in the prior illustration. The prefix begins with 2002 and is followed by the router's IPv4 address express in hex. These addresses carry a /48 mask.

NAT64

The NAT you're going to use for IPv6-IPv4 translation isn't traditional NAT; rather, it's *Network Address Translation IPv6 To IPv4*, thankfully shortened to NAT64.

We used to use *NAT-PT* (NAT-Protocol Translation) for this translation, but NAT-PT is now a thing of the past, largely due to its integrated use of DNS. A major benefit of NAT64 is keeping the NAT64 and DNS64 functions totally separate.

NAT64 can run in stateless or stateful mode, and we know when there are two ways to run something, we better know the major differences between the two!



Stateless NAT64 embeds an IPv4 address directly into an IPv6 address, resulting in a one-to-one mapping of IPv6 addresses to IPv4 addresses. One reason for the move to IPv6 is that we're running out of IPv4 addresses, so the lack of IPv4 address conservation with stateless NAT64 is a concern.

Stateful mode doesn't suck up our IPv4 addresses as quickly, since stateful NAT64 allows multiple IPv6 addresses to use a single IPv4 address. NAT veterans may remember PAT performing a similar form of overloading. (If not, don't sweat it, I'll fill you in during another part of the course!)

BGP 1 : Introduction

Wednesday, April 4, 2018 9:17 AM

BGP

"The study of law is something new and unfamiliar to most of you, unlike any other schooling you have ever known before." -- Charles Kingsfield, Law Professor, *The Paper Chase*

"Same goes for the study of BGP." -- Chris Bryant, Bulldog

Here's the deal with BGP – it's unlike anything you've studied to this point. The rules are different, the concepts are different, everything's different. BGP throws a lot of people for a loop the first time they see it, and that included me. The key to initial BGP success is to take one concept at a time, then one attribute at a time, and before you know it you've mastered the fundamentals of BGP. (For those of you with one eye on the CCIE, this is some of the most important study you'll do.)

The first natural question: "What the heck *is* BGP, anyway?" Here's one definition:

An internet protocol that allows groups of routers (called autonomous systems) to share routing information so that efficient, loop-free routes can be established. BGP is commonly used within and between Internet Service Providers (ISPs).

That definition sounded like EIGRP until we got to the second sentence! Both EIGRP and BGP use autonomous systems, and both protocols use that term to refer to a logical group of routers. The difference comes in where BGP fits in the big scheme of things. BGP is an Exterior Gateway Protocol that runs between ISPs, where EIGRP is an Interior Gateway Protocol that runs in the networks connected by ISPs.

BGP shares other characteristics with EIGRP:

BGP supports VLSM and summarization.

BGP will send full updates when routers initially become neighbors, and after that will send only partial updates reflecting the latest network changes.

4/4/2018 9:20 AM - Screen Clipping

BGP neighbors will exchange *much* more extensive information about networks than our IGPs do. The additional BGP path information comes in the form of *attributes*, and these path attributes are contained in the updates sent by BGP routers. Mastering these attributes are the key to success with BGP, and that starts with knowing which attributes are well-known and which are optional.

LEARN AND MASTER ATTRIBUTES.

When BGP *should* be used:

Your company is connecting to more than one AS or ISP. Decisions on the best links to use can be made by utilizing BGP path attributes.

The routing policy of your organization and your ISP differ.

Your company is an ISP. When traffic from other autonomous systems use your AS as a transit domain, BGP will definitely be needed.

In short, if your AS has more than one connection to other ASes, or other ASes are using your AS as a transit domain, you will definitely be running BGP.

There are also circumstances that dictate when BGP should *not* be used:

When there's a single connection to the Internet or to another AS, and no redundant link exists.

When you don't care which path is used to reach a route in another AS.

When router resources are limited (memory and CPU, that is).

BGP 2 : External BGP Peering

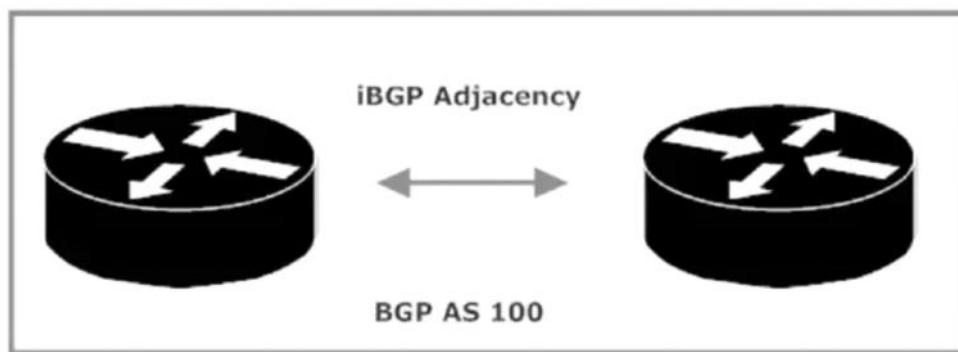
Wednesday, April 4, 2018 9:27 AM

Like TCP, BGP is connection-oriented ("reliable"). An underlying connection between two BGP speakers is established before routing information is actually exchanged. This connection takes place on TCP port 179, an excellent port to leave unblocked by access lists (or anything else!). Once the connection is established, the BGP speakers exchange routes and synch their tables. After this initial exchange, a BGP speaker will send further updates only upon a change in the network.

BGP speakers do not have to be in the same AS in order to become neighbors or exchange routes, as we'll see!

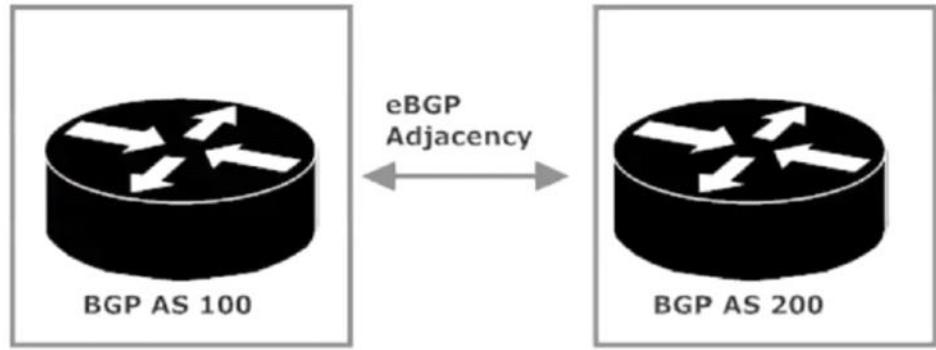
BGP adjacencies are also called "peerings". A BGP peer in the same AS as the local router is an Internal BGP (iBGP) peer...

4/5/2018 8:49 AM - Screen Clipping



...and a BGP peer in another AS is an External BGP (eBGP) peer.

4/5/2018 8:50 AM - Screen Clipping



The “i” and “e” make huge differences in BGP behavior, so watch your internal vs. external adjacencies! Cisco recommends that eBGP peers be directly connected; iBGP peers are not required to be so connected, and generally aren’t.

We’ll start our first lab with the *router bgp* command, followed by a *neighbor* command to identify this BGP speaker’s potential neighbors. Using IOS Help after *neighbor* gives you a hint that BGP is a bit more complex than EIGRP or OSPF:

```
R1(config)#router bgp ?
<1-65535> Autonomous system number
```

4/5/2018 8:50 AM - Screen Clipping



```
R1(config-router)#neighbor 172.12.123.2
% Incomplete command.
```

```
R1(config-router)#neighbor 172.12.123.3 remote-as ?
<1-65535> AS of remote neighbor
```

```
R1(config-router)#neighbor 172.12.123.3 remote-as 200
```

4/5/2018 8:52 AM - Screen Clipping

Let's build this adjacency.

```
R1#conf t
Enter configuration commands, one per line
R1(config)#router bgp ?
  <1-65535> Autonomous system number

R1(config)#router bgp 100
R1(config-router)#neighbor ?
  A.B.C.D    Neighbor address
  WORD        Neighbor tag
  X:X:X:X::X  Neighbor IPv6 address

R1(config-router)#neighbor 172.12.123.3 ?
  activate          Enable the Address
  advertise-map     specify route-map
  advertisement-interval  Minimum interval
  allowas-in        Accept as-path
  capability        Advertise capabilities
  default-originate Originate default routes
  description       Neighbor specification
  disable-connected-check One-hop away EBGP
  distribute-list   Filter updates
  dmzlink-bw        Propagate the D
  eb
```

4/5/2018 8:53 AM - Screen Clipping

You have to put something as **remote-as**. Theres no dynamic discovery and you don't have to be in the same AS.

```
R1(config-router)#neighbor 172.12.123.3 remote-as ?
  <1-65535> AS of remote neighbor

R1(config-router)#neighbor 172.12.123.3 remote-as 300
R1(config-router)#^Z
```

4/5/2018 8:54 AM - Screen Clipping

```

R1#show ip bgp neighbor
BGP neighbor is 172.12.123.3, remote AS 300, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active
  Last read 00:00:14, last write 00:00:14, hold time is 180, keepalive is 60 seconds
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
      Sent          Rcvd
  Opens:            0          0
  Notifications:  0          0
  Updates:         0          0
  Keepalives:      0          0
  Route Refresh:   0          0
  Total:           0          0
Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0/0
  Output queue size : 0
  Index 1, offset 0, Mask 0x2
  1 update-group member

```

The output here can be a little misleading the first time around. The first line says a BGP neighbor is at 172.12.123.3, it's in AS 200, and it's an external link. So far, so good – but what about that BGP state of "active"? We know from our EIGRP studies that "active" isn't always good. Is it good here?

Actually, it's not. The BGP state *Active* indicates a BGP peer connection that does not yet fully exist. Let's have a look at all the BGP states before continuing with this lab.

Idle is the initial state of a BGP peering. Should you note a connection that went to *Idle* and stayed there, check these values:

Make sure the IP address in the neighbor statement is the correct address to use.

Be sure the local router knows how to get to that particular IP address.

Connect follows *Idle*. In *Connect* state, a TCP connection request has been sent but a response has not yet been received. If the TCP connection completes, BGP will move to *OpenSent*. If the connection does not complete, the state goes to *Active*.

Active indicates the BGP speaker is continuing to create a peering with the intended neighbor. Basically, this is the halfway point of the connection. The local router has

successfully sent a BGP Open packet to the potential neighbor, but hasn't heard anything in return. As with Idle, there's nothing wrong with this state unless your connection stays there. In that case, check the remote router's neighbor statement and make sure the AS numbers are correct. (You'd be surprised how often wrong AS numbers are the reason for a peering not forming.)

OpenSent indicates the BGP speaker has received an Open message from the peer. In this state, BGP determines whether the peer is in the same AS (iBGP) or a different AS (eBGP).

OpenConfirm state has the BGP speaker waiting for a keepalive. If one is received, the state moves to Established, and the peering is complete. It's in the Established state that update packets are finally exchanged.

We know the reason the peering between our two routers hasn't completed, so let's complete the config and see if that does the trick!

```
R3(config)#router bgp 200
R3(config-router)#neighbor 172.12.123.1 remote-as 100
*Jun 12 13:52:01.835: %BGP-5-ADJCHANGE: neighbor 172.12.123.1 Up
```

1 update-group member		Sent	Rcvd
Prefix activity:		-----	-----
Prefixes Current:		0	0
Prefixes Total:		0	0
Implicit Withdraw:		0	0
Explicit Withdraw:		0	0
Used as bestpath:	n/a		0
Used as multipath:	n/a		0
Local Policy Denied Prefixes:		Outbound	Inbound
Total:		0	0
Number of NLRI's in the update sent:		max 0, min 0	
Connections established 0; dropped 0			
Last reset never			
No active TCP connection			

1#

```

R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 300
R3(config-router)#neighbor 172.12.123.1 remote-as 100
R3(config-router)#^Z
R3#
*Oct 18 15:05:07.407: %SYS-5-CONFIG_I: Configured from console
R3#
*Oct 18 15:05:08.831: %BGP-5-ADJCHANGE: neighbor 172.12.123.1 u
R3#

```

4/5/2018 9:06 AM - Screen Clipping

We can see that we're now up.

```

R3#show ip bgp neighbor 172.12.123.1
BGP neighbor is 172.12.123.1, remote AS 100
  BGP version 4, remote router ID 172.12.123.1
  BGP state = Established, up for 00:00:21
    Last read 00:00:21, last write 00:00:21,
    is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received
    Four-octets ASN Capability: advertised
    Address family IPv4 Unicast: advertised
    Multisession Capability: advertised
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                    Sent          Rcvd
  Opens:            1              1
  Notifications:   0              0
  Updates:         1              0
  Keepalives:      2              3
  Route Refresh:   0              0
--More--

```

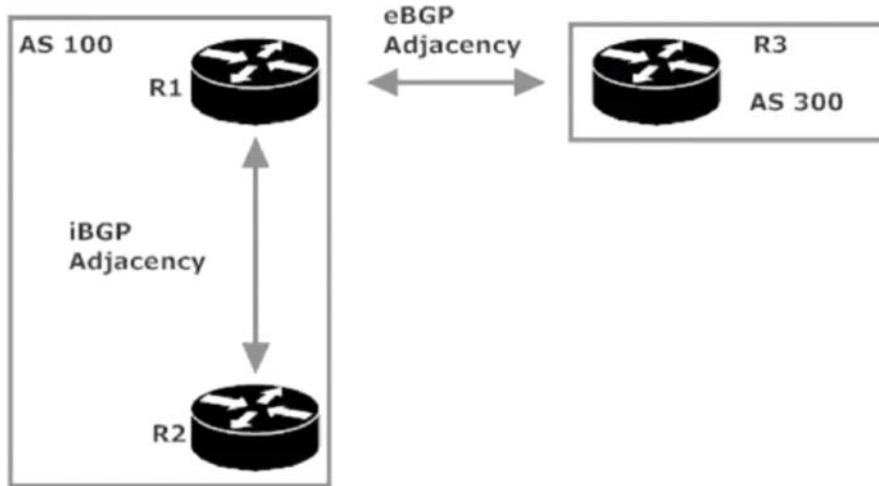
4/5/2018 9:07 AM - Screen Clipping

4/5/2018 9:07 AM - Screen Clipping

BGP 3 : iBGP Peering and Loopback Interface Discussion

Thursday, April 5, 2018 9:08 AM

Looks even better! Our neighbor relationship is *Established* and has been up for about a minute and a half. Let's add an iBGP peering to the mix while bringing R2 into the picture. We'll also see what happens if we try to create a peering between the local router and, well, the local router!



Config t
Router bgp 100
Neighbor x.x.x.x remote-as 100

```
R1#conf t
Enter configuration commands
R1(config)#router bgp 100
R1(config-router)#neighbor 172.12.123.2 remote-as 100
R1(config-router)#^Z
R1#wr
```

```
R2#conf t
Enter configuration commands, one per line. End with Ctrl-Z
R2(config)#router bgp 100
R2(config-router)#neighbor 172.12.123.1 remote-as 100
R2(config-router)#^Z
R2#
```

Verify

```

R2#show ip bgp neighbor
BGP neighbor is 172.12.123.1, remote AS 100, internal link
  BGP version 4, remote router ID 172.12.123.1
  BGP state = Established, up for 00:00:16
  Last read 00:00:16, last write 00:00:15, hold time is 180,
is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised
    Address family IPv4 Unicast: advertised and received
    Multisession Capability: advertised
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

          Sent          Rcvd
  Opens:           1           1
  Notifications:  0           0
  Updates:        1           0
  Keepalives:     2           3
  Route Refresh:  0           0
  Total:          4           4
--More--

```

```

iss: 3338156044  snduna: 3338156164  sndnxt: 3338156164
irs: 3083122790  rcvnxt: 3083122893

sndwnd: 16265  scale:      0  maxrcvwnd: 16384
rcvwnd: 16282  scale:      0  delrcvwnd:   102

SRTT: 269 ms, RTTO: 2307 ms, RTV: 2038 ms, KRTT: 0 ms
minRTT: 36 ms, maxRTT: 650 ms, ACK hold: 200 ms
Status Flags: active open
Option Flags: nagle, path mtu capable
IP Precedence value : 6

Datagrams (max data segment is 1460 bytes):
Rcvd: 4 (out of order: 0), with data: 2, total data bytes: 10
Sent: 7 (retransmit: 0, fastretransmit: 0, partialack: 0, Sec
      with data: 4, total data bytes: 119

```

Most useful command.

```
R2#show ip bgp summary
BGP router identifier 172.12.123.2, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor          V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
/PfxRcd
172.12.123.1     4      100    4        5        1       1       0
0
R2#
```

```
R1#show ip bgp summ
BGP router identifier 172.12.123.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor          V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
172.12.123.2     4      100    5        5        1       0       0 00:01:45
172.12.123.3     4      200    9        8        1       0       0 00:04:38
R1#
```

We used the IP addresses on R1 and R3's physical interfaces to create our first eBGP adjacency. Nothing wrong with that, but we're more likely to use IP addresses from loopback interfaces for such an adjacency. Those physical interfaces can go down for a number of reasons, but the only way a logical interface goes down is if someone intentionally deletes it or the entire router is unavailable – and in that case you have much bigger problems than lost adjacencies!

Loopback interfaces are not considered directly connected even if they share a common subnet. You'll need the *ebgp-multihop* command when configuring eBGP adjacencies with addresses that aren't on the same subnet. When those addresses are on loopback interfaces, you'll also need the *update-source loopback* command.

If you use loopback addresses for eBGP adjacencies, you may need to configure a static route on each router that points to the remote router's loopback. If your local router doesn't know how to get to the address specified by *neighbor*, we're pretty much stuck before we begin!



Loopback interfaces aren't considered directly connected in BGP, therefore you need the *ebgp-multihop* command for ebgp adjacency. You'll probably need to create a static route to the loopback.



neighbor 3.3.3.3 remote-as 200

*"That's fine, but how the heck
do I get to 3.3.3.3?"*

BGP 4 : eBGP Peering and Loopback Interface Discussion

Thursday, April 5, 2018 9:17 AM

```
R3(config)#int loopback0
R3(config-if)#ip address 3.3.3.3 255.255.255.255
R3(config-if)#^Z
R3#wr
```

```
R1#conf t
Enter configuration commands, one per line. End with ^Z.
R1(config)#int loopback0
R1(config-if)#ip address 1.1.1.1 255.255.255.255
Oct 20 20:06:14.674: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0 changed state to up
R1(config-if)#ip address 1.1.1.1 255.255.255.255
R1(config-if)#^Z
R1#
```

Create loopbacks and then create adjacencys.

```
R1#
Oct 20 20:06:18.632: %SYS-5-CONFIG_I: Configured from console by console
R1#conf t
Enter configuration commands, one per line. End with ^Z.
R1(config)#router bgp 100
R1(config-router)#neighbor 3.3.3.3 remote-as 200
R1(config-router)#^Z
```

State will still be idle. But since were using loopbacks we have to create a ebgp multihop command.

```
R1(config-router)#neighbor 3.3.3.3 ebgp-multihop 2
R1(config-router)#^Z
R1#wr
```

```
R3#conf t
Enter configuration commands, one per line. End with ^Z.
R3(config)#router bgp 200
R3(config-router)#neighbor 1.1.1.1 ebgp-multihop 2
R3(config-router)#^Z
R3#
```

```
R1#show ip bgp summ
BGP router identifier 172.12.123.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor          V     AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down State/P
3.3.3.3           4     200      0        0          0      0    0   never  Active
172.12.123.2      4     100      41       38         1      0    0  00:34:30  udemv
R1#
```

We're now in active and need to get to established.

```
R1(config-router)#neighbor 3.3.3.3 update-source loopback0
R1(config-router)#^Z
R1#wr
Building configuration...
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 200
R3(config-router)#neighbor 1.1.1.1 update-source loopback0
R3(config-router)#^Z
```

```
R1#show ip bgp summ
BGP router identifier 172.12.123.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor          V   AS MsgRcvd MsgSent    Tblver  InQ OutQ Up/Down State/P
3.3.3.3           4   200      0       0          0     0     0 never   Active
172.12.123.2     4   100     43      39          1     0     0 00:35:53  udemv
R1#
```

We're STILL stuck in active. We need to create a static route to help them establish.

```
R1(config)#ip route 3.3.3.3 255.255.255.255 172.12.123.3
R1(config)#^Z
R1#wr
Building configuration...
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip route 1.1.1.1 255.255.255.255 172.12.123.1
R3(config)#^Z
R3#
```

```
R3#
*Oct 21 00:27:11.227: %SYS-5-CONFIG_I: Configured from console by console
*Oct 21 00:27:11.547: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
R3#
```

We're now established.

Advertising Routes With BGP

We'll use the *network* command in BGP, but not in quite the same fashion as we do with OSPF and EIGRP. The command will look the same, but where IGPs identify the interfaces to be enabled with the protocol in question, BGP uses this command to identify the networks to be advertised by BGP.

The network specified by the BGP *network* command must be an exact match for a network contained in the IP routing table, and that includes the mask, should you include it. Using the mask in the BGP *network* command is not required, but it's highly recommended. If you're called upon to troubleshoot a BGP configuration and that config is missing the masks on the *network* statements, that's likely the issue. Use the masks or you'll end up with only the classful networks.

Let's get this lab started! We're using the adjacencies we just built.



```
R3#conf t
Enter configuration commands, one per line. End with CNTL-Z
R3(config)#router bgp 200
R3(config-router)#network ?
  A.B.C.D Network number

R3(config-router)#network 3.3.3.3 ?
  backdoor  Specify a BGP backdoor route
  mask      Network mask
  route-map Route-map to modify the attributes
<cr>

R3(config-router)#network 3.3.3.3 mask ?
  A.B.C.D Network mask

R3(config-router)#network 3.3.3.3 mask 255.255.255.255 ?
  backdoor  Specify a BGP backdoor route
  route-map Route-map to modify the attributes
<cr>

R3(config-router)#network 3.3.3.3 mask 255.255.255.255
R3(config-router)#

```

```
R1#show ip bgp
BGP table version is 6, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*> 3.3.3.3/32        3.3.3.3                  0          0 200 i
R1#
```

```

R3(config-router)#
R3(config-router)#
R3(config-router)#no network 33.3.3.3 mask 255.255.255.255
R3(config-router)#network 33.3.3.3 mask 255.255.255.0
% BGP: Incorrect network or mask/prefix-length configured
R3(config-router)#network 33.3.3.0 mask 255.255.255.0
R3(config-router)#^Z
R3#wr
Building configuration...

BRYANT_ADV_1#1
[Resuming connection 1 to r1 ... ]

R1#
R1#
R1#show ip bgp
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*> 3.3.3.3/32        3.3.3.3              0        0 200 i
*> 33.3.3.0/24       3.3.3.3              0        0 200 i
R1#

```

4/5/2018 9:33 AM - Screen Clipping

ATTRIBUTES:

With our first two routes in the bank, let's turn our attention to the all-important BGP path attributes. To truly understand BGP and earn your CCNP, you have to know exactly what these attributes do and how they affect your BGP deployment. Let's jump right in!

The well-known mandatory attributes: AS_PATH, origin, next-hop.

The well-known discretionary attributes: local preference ("LOCAL PREF"), atomic aggregate.

Optional, transitive attributes: aggregator, community.

The optional non-transitive attribute: MED, the multi-exit discriminator.

The three mandatory attributes will appear in all BGP update messages sent to neighbors. These are the only three attributes all BGP speakers must understand.

AS_PATH, Origin, and Next-hop are the most important to remember.

The optional attributes can be a bit of a pain in the royal tuckus for BGP operation, since not every BGP speaker is going to understand every optional attribute. That's where the difference between "optional transitive" and "optional non-transitive" comes into play. A BGP path carrying an unrecognized transitive optional attribute will be accepted, and should this path be advertised to other routers, the Partial bit will be set and the attribute advertised to the neighbor.

Marking an attribute as Partial is the equivalent of the advertising router saying "I didn't understand this attribute, but maybe you will, so here it is."

An unrecognized non-transitive optional attribute will not be passed on to other BGP speakers. With that said, let's start examining the individual attributes.

BGP 6 : Origin and Next-hop Attributes and Best Path Selection Process

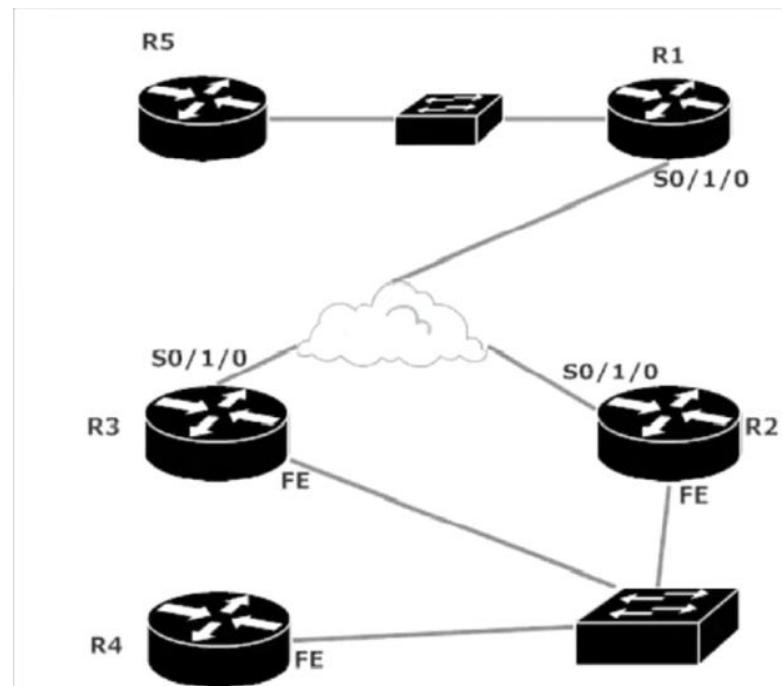
Thursday, April 5, 2018 9:36 AM

The networks:

R1 – R5 Ethernet: 10.1.1.0 /24

R1 – R2 – R3 Serial: 172.12.123.0 /24

R2 – R3 – R4 Ethernet: 172.12.234.0 0/24



The Origin Attribute

You'll see this attribute at the far right of the output of *show ip bgp*. The actual origin codes are shown as well, but I'd have them memorized for the CCNP exams.

```
R1#show ip bgp
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
*> 3.3.3.3/32    172.12.123.3      0        0 200 i
*> 33.3.3.0/24   172.12.123.3      0        0 200 i
```

The letter "i" indicates a route that originated from an IGP via the *network* command, where the letter "e" indicates a path that originated from an External Gateway Protocol. The question mark tells us the true origin of the route is unclear, since it was learned via route redistribution.

In the best-path selection process, "i" is preferred over "e", which in turn is preferred over the question mark. Much more about this process later in this section.

Memorize Origin Codes.

The AS_PATH Attribute And The Best Path Selection Process

The AS_PATH attribute shows the autonomous systems along the path to the destination network, including the AS the destination network resides in. The shorter the path, the more preferred the path is during the best-path selection process.

The AS_PATH attribute helps to prevent routing loops. Should a BGP speaker receive an update that has its own AS number in the path to a destination, that route is discarded. In this example, the only AS shown in the path is the AS containing the networks. Note that on R1, the Path column shows an AS of 200, followed by the origin code...

```
R1#show ip bgp

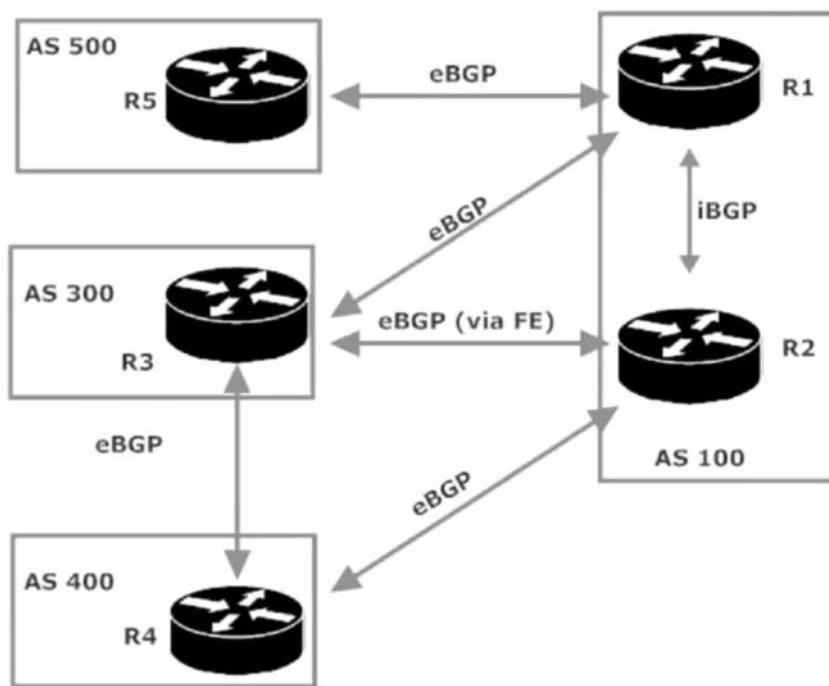
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
*> 3.3.3.3/32    172.12.123.3      0        0 200 i
*> 33.3.3.0/24   172.12.123.3      0        0 200 i
```

Let's run a lab that allows us to see the best-path selection process in action. In this lab, every router will advertise its loopback address into BGP, and every router's loopback is its router number for each octet.

The physical network setup follows. For the sake of clarity and sanity (yours and mine), the physical interfaces and switches will not be included in the BGP diagrams. Also note that R2 is on the right side of the diagram; I mention that only because it's on the left in most of the other diagrams in the book.

BGP adjacencies are as follows.



4/5/2018 9:41 AM - Screen Clipping

R1 has at least one entry for each loopback in our network, and multiple paths for the loopbacks on R3 and R4.

```

R1#show ip bgp
BGP table version is 6, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inte
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 1.1.1.1/32        0.0.0.0              0        32768  i
*->i2.2.2.2/32       172.12.123.2          0        100     0 i
* i3.3.3.3/32       172.12.234.3          0        100     0 300 i
*>                  172.12.123.3          0        0       0 300 i
*> 4.4.4.4/32       172.12.123.3          0        0       0 300 400 i
* i                  172.12.234.4          0        100     0 400 i
*> 5.5.5.5/32       10.1.1.5             0        0       0 500 i
R1#
  
```

Router 1 has one entry for its loopback. 1 for rtr5 loopback. 3 and 4 have two entry's? These two 172 IP's for each loopback. Pay attention the **valid** and **best**.

Every path is marked with an asterisk, and in BGP-speak that means every path is valid. BGP had some decisions to make as to which paths were valid and best when it came to the loopbacks on R3 and R4, and here's the rather long-winded decision-making process in all its splendor. When deciding between multiple paths for the same destination, we start at the top of this list and keep going until the tie is broken.

When it comes to making a decision for the best routes, multiple paths have a long decision making process. **MEMORIZE THESE.**

1. Highest weight preferred. (BGP weight is a Cisco-proprietary attribute)
2. If tie or non-Cisco routers involved, highest local preference is preferred.
3. Locally originated path preferred
4. Shortest AS_PATH preferred
5. Best origin code (i, then e, then ?)
6. Lowest MED
7. eBGP path preferred over iBGP path
8. lowest IGP metric to BGP next-hop address
9. oldest path
10. path from BGP router with lowest BGP RID

4/5/2018 9:46 AM - Screen Clipping

Weight
local pref
local origin path
shortest AS
best origin code (i,e,?)
MED
ebgp over ibgp path
lowest IGP metric to BGP next hop address
oldest path
path from BGP router with lowest BGP RID.

BGP 7: Examining the Selection of One BGP Path over another

Monday, April 9, 2018 8:24 AM

With this list in mind, let's have another look at R1's BGP table.

```
R1#show ip bgp
BGP table version is 6, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.1.1.1/32	0.0.0.0	0		32768	i
*>i2.2.2.2/32	172.12.123.2	0	100	0	i
* i3.3.3.3/32	172.12.234.3	0	100	0	300 i
*>	172.12.123.3	0		0	300 i
*> 4.4.4.4/32	172.12.123.3			0	300 400 i
* i	172.12.234.3	0	100	0	300 400 i
*> 5.5.5.5/32	10.1.1.5	0		0	500 i

The two networks with multiple paths in the table are 3.3.3.3 /32 and 4.4.4.4 /32. Let's take a closer look at those two paths and how BGP determined the best path for each, starting with 3.3.3.3 /32.

```
R1#show ip bgp
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i3.3.3.3/32	172.12.234.3	0	100	0	300 i
*>	172.12.123.3	0		0	300 i

4/9/2018 10:49 AM - Screen Clipping

The first value BGP considers is **weight**. Both paths here have a weight of 0. The second value is **local pref** but the path chosen has no local pref listed. The other path not chosen has a local pref of **100 by default**. So why was one path chosen over the other?

We'll come back to that after taking a closer look at the paths for 4.4.4.4 /32.

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 4.4.4.4/32	172.12.123.3			0	300 400 i
* i	172.12.234.4	0	100	0	400 i

You can see that the route is inaccessible. You can also see the number of available paths.

```

R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 3
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.5 172.12.123.2
      300 400
        172.12.123.3 from 172.12.123.3 (172.12.234.3)
          Origin IGP, localpref 100, valid, external, best
      400
        172.12.234.4 (inaccessible) from 172.12.123.2 (172.12.234.2)
          Origin IGP, metric 0, localpref 100, valid, internal
R1#show ip bgp
BGP table version is 6, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - in
Origin codes: i - IGP, e - EGP, ? - incomplete
      Network           Next Hop            Metric LocPrf Weight Path
*-> 1.1.1.1/32       0.0.0.0             0        32768  i
*>i2.2.2.2/32       172.12.123.2         0        100    0 i
* i3.3.3.3/32       172.12.234.3         0        100    0 300 i
*>
*> 4.4.4.4/32       172.12.123.3         0        300    0 400 i
* i
*> 5.5.5.5/32       172.12.234.4         0        100    0 400 i
*-> 10.1.1.5         10.1.1.5            0        500    0 500 i
R1#

```

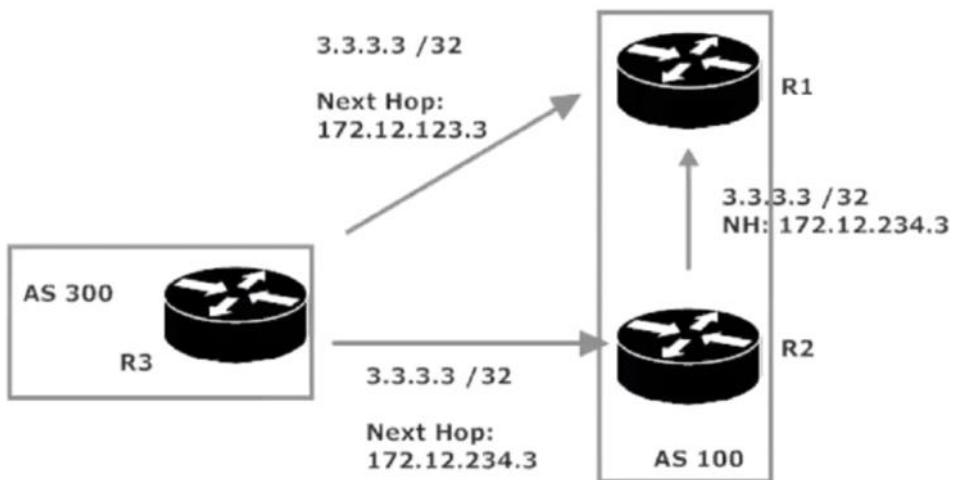
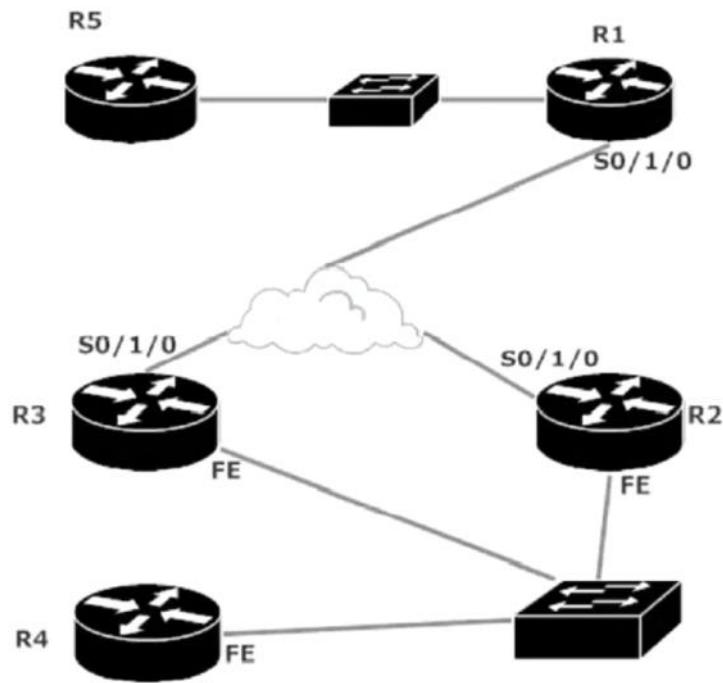
The Nexthop for the given path to 172.12.234.3 and the router doesn't have a path to get to it. We're not running any protocols or static routes to tell the router how to get there, so the route isn't even considered.

The networks:

R1 – R5 Ethernet: 10.1.1.0 /24

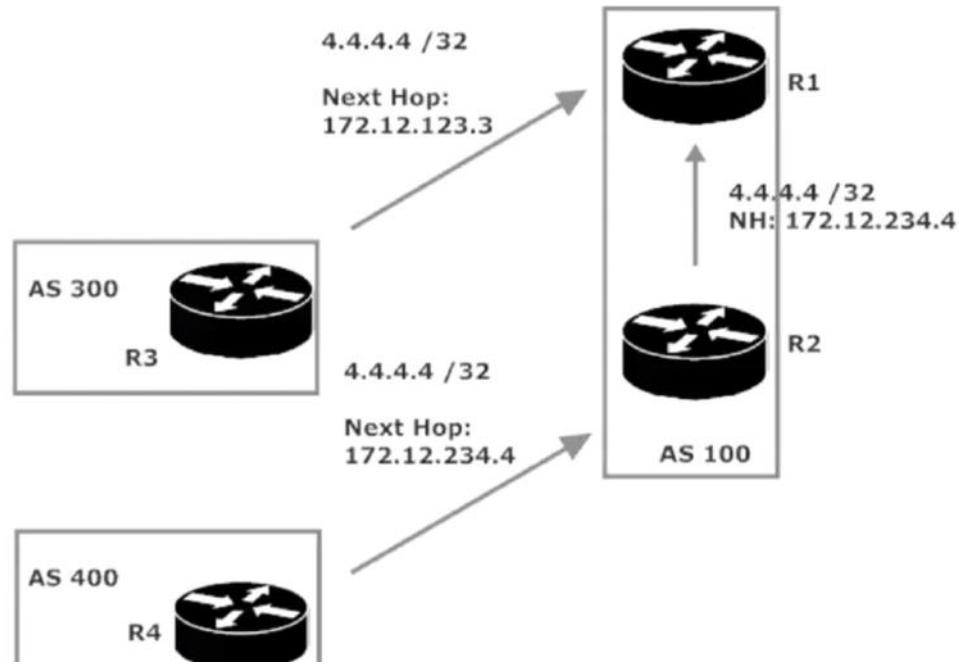
R1 – R2 – R3 Serial: 172.12.123.0 /24

R2 – R3 – R4 Ethernet: 172.12.234.0 0/24



4/9/2018 10:57 AM - Screen Clipping

Same goes for the route to 4.4.4.4. When R2 learns that route from R4, the next-hop is 172.12.234.4. When R1 learns that route from R2, the next-hop address is still 172.12.234.4. Why didn't the next-hop address change to R2's advertising interface (172.12.123.2)?



4/9/2018 10:57 AM - Screen Clipping

This is what's causing the inaccessibility because for some reason the IP address isn't changing the next hop address. Solution on next video.

BGP 8: Next Hop Self Command

Monday, April 9, 2018 10:58 AM

Before we give the solution, we need to identify a tool. The next hop self.

R2(config-router)#neighbor 172.12.123.1 ?	Enable the Address Family for this Neighbor
activate	specify route-map for conditional advertisement
advertise-map	Minimum interval between sending BGP routing update
advertisement-interval	Accept as-path with my AS present in it
allowas-in	Advertise capability to the peer
capability	Originate default route to this neighbor
default-originate	Neighbor specific description
description	one-hop away EBGP peer using loopback address
disable-connected-check	Filter updates to/from this neighbor
distribute-list	Propagate the DMZ link bandwidth
dmzlink-bw	Allow EBGP neighbors not on directly connected networks
ebgp-multihop	session fall on peer route lost
fall-over	Establish BGP filters
filter-list	high availability mode
ha-mode	Inherit a template
inherit	Specify a local-as number
local-as	Maximum number of prefixes accepted from this peer
maximum-prefix	Disable the next hop calculation for this neighbor
next-hop-self	

This command disables the next hop calculation for its neighbor.

```
R2(config-router)#neighbor 172.12.123.1 next-hop-self
R2(config-router)#^Z
R2#
*Oct 24 17:19:11.831: %SYS-5-CONFIG_I: Configured from console by c
R2#
```

Now you have to do a soft reset to refresh the routing table.

```
R2#clear ip bgp * soft out
R2#
```

```
R1#show ip bgp
BGP table version is 18, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 1.1.1.1/32        0.0.0.0              0        32768  i
* i3.3.3.3/32        172.12.123.2          0       100    0 300  i
*->
*->i4.4.4.4/32        172.12.123.3          0       100    0 400  i
*-
*-> 5.5.5.5/32        172.12.123.3          0       100    0 300  400
*-> 5.5.5.5/32        10.1.1.5             0       100    0 500  i
R1#
```

4/9/2018 11:01 AM - Screen Clipping

Notice route selection has changed to 172.12.123.2. Router 3 didn't change, but we don't see

inaccessible

```
R1#show ip bgp 3.3.3.3
BGP routing table entry for 3.3.3.3/32, version 13
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.5 172.12.123.2
    300
      172.12.123.2 from 172.12.123.2 (172.12.234.2)
        Origin IGP, metric 0, localpref 100, valid, internal
    300
      172.12.123.3 from 172.12.123.3 (3.3.3.3)
        Origin IGP, metric 0, localpref 100, valid, external, best
R1#
```

The path for router 4.4.4.4 changed because the ASpath is shorter.

```
R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 18
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.5 172.12.123.3
    400
      172.12.123.2 from 172.12.123.2 (172.12.234.2)
        Origin IGP, metric 0, localpref 100, valid, internal, best
    300 400
      172.12.123.3 from 172.12.123.3 (3.3.3.3)
        Origin IGP, localpref 100, valid, external
R1#
```

400 was chosen because it has a high ASpath.

The valid and best route to 3.3.3.3 still has a next-hop of 172.12.123.3, but the next-hop for the other route has changed to 172.12.123.2.

```
R1#show ip bgp 3.3.3.3
BGP routing table entry for 3.3.3.3/32, version 4
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
    1          2
    300
      172.12.123.2 from 172.12.123.2 (2.2.2.2)
        Origin IGP, metric 0, localpref 100, valid, internal
    300
      172.12.123.3 from 172.12.123.3 (33.3.3.3)
        Origin IGP, metric 0, localpref 100, valid, external, best
```

For 4.4.4.4 /32, the path now in use is the one with the next-hop of 172.12.123.2, since its AS_PATH is shorter than the other valid path.

The best-path selection process was just a little longer for 3.3.3.3 /32. The weights are the same, the local pref is the same, none of the routes originated on R1, the AS_PATH length is the same, the origin code is the same (IGP), and the MED is the same ("metric"). Next on the list is eBGP routes being preferred over iBGP routes, and that's why the path with a next-hop of 172.12.123.3 was chosen as the best path - external over internal!

BGP 9: Next Hop Address Rules

Monday, April 9, 2018 11:05 AM

Why didn't the next-hop addresses change? Let's find out with a closer look at the rules for the default BGP next-hop address.

The Next-Hop Attribute Defaults

Let's go back to a simpler time and a simpler network. R3 is advertising its two loopback interfaces, 3.3.3.3 /32 and 33.3.3.3 /32, via an eBGP adjacency with R1. All adjacencies in this lab are over the 172.12.123.0 /24 network.



4/9/2018 11:09 AM - Screen Clipping

```
R1#show ip bgp
BGP table version is 3, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 3.3.3.3/32      172.12.123.3        0       0 300 i
*-> 33.3.3.3/32     172.12.123.3        0       0 300 i
R1#
```

4/9/2018 11:09 AM - Screen Clipping

When a BGP speaker advertises a route to an eBGP neighbor, the next-hop address is that of the advertising interface. R1's BGP table shows the address of the Serial interface on R3 (172.12.123.3) as the next-hop address for both of those routes.

```
R1#show ip bgp
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
*> 3.3.3.3/32    172.12.123.3      0        0 200 i
*> 33.3.3.0/24   172.12.123.3      0        0 200 i
```

Makes perfect sense, right? Right! Now, here's the *slightly* odd rule. With iBGP route updates, if the route was originated outside the local AS, the next-hop address is *still* the source address of the router in the remote AS.

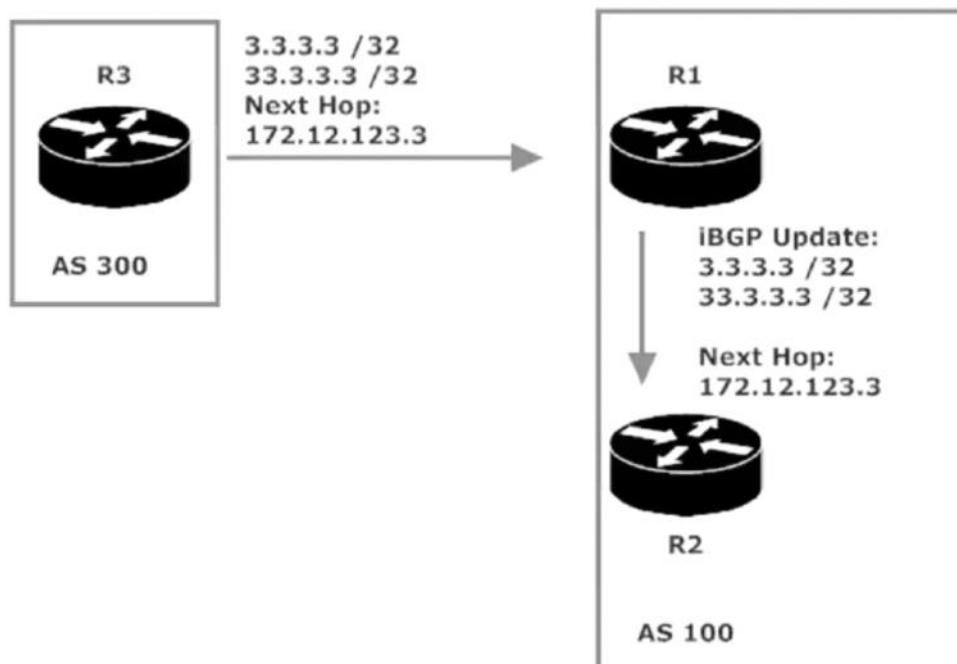
Weird, right? Right!

Let's prove this theory by adding R2 back to AS 100 and then advertising those two routes from R1 to R2.

4/9/2018 11:10 AM - Screen Clipping

In iBGP, if the original route was created outside the local AS, the nexthop is still the source address of the router in the remote AS. This is why you need to pay attention to "I" and "e" originations.

Let's prove this theory by adding R2 back to AS 100 and then advertising those two routes from R1 to R2.



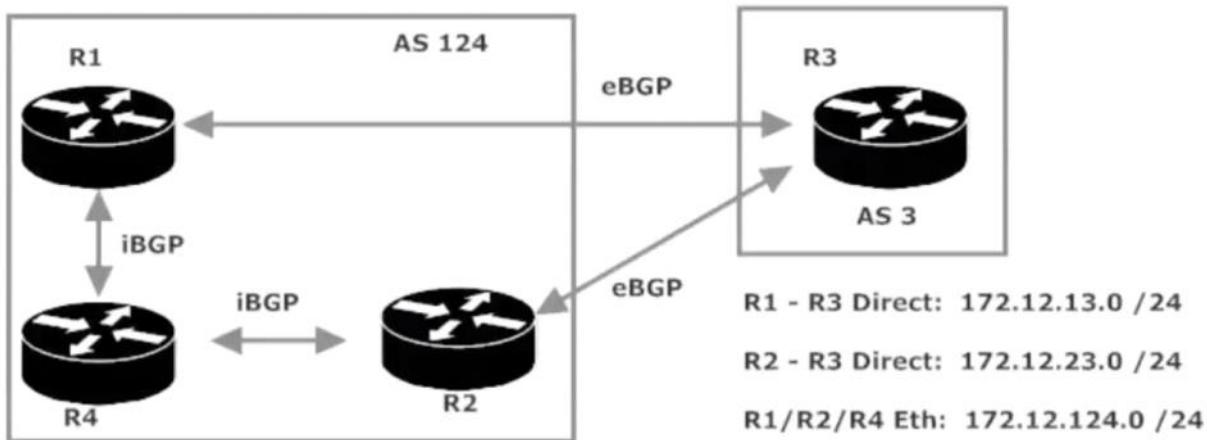
|
When a router has learned a route from the external neighbor, the nexthop remains, it won't change.

BGP 10: Multi-exit Discriminator

Monday, April 9, 2018 11:12 AM

This is an optional attribute for BGP.

It sounds like something we tried to stop Godzilla with back in the day ("*Fire the Multi-Exit Discriminator!*"), but this optional attribute really comes in handy when there are multiple entrance points for a single AS, as in the following lab. Note the direct connections between R1 – R3 and R2 – R3; the frame network is not in use in this lab.



R3 can enter AS 124 via R1 or R2, and there may be one path in particular we *really* want R3 to use. We can use the MED to tell R3 which of those two paths is the one we'd prefer it to use. The path with the lowest MED is preferred. The MED

4/9/2018 11:14 AM - Screen Clipping

Best path selection may not give us our preferred path. You can also differentiate these connection based on traffic.

For this lab, I'll advertise two loopbacks that just happen to be sitting on R4.

```
R4(config)#router bgp 124
R4(config-router)#network 4.4.4.4 mask 255.255.255.255
R4(config-router)#network 44.4.4.4 mask 255.255.255.255
```

4/9/2018 11:16 AM - Screen Clipping

```
R3#show ip bgp
BGP table version is 5, local router ID is 172.12.23.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
                r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*  4.4.4.4/32        172.12.23.2          0    124   i
*->                    172.12.13.1          0    124   i
*  44.4.4.4/32       172.12.23.2          0    124   i
*->                    172.12.13.1          0    124   i
```

4/9/2018 11:16 AM - Screen Clipping

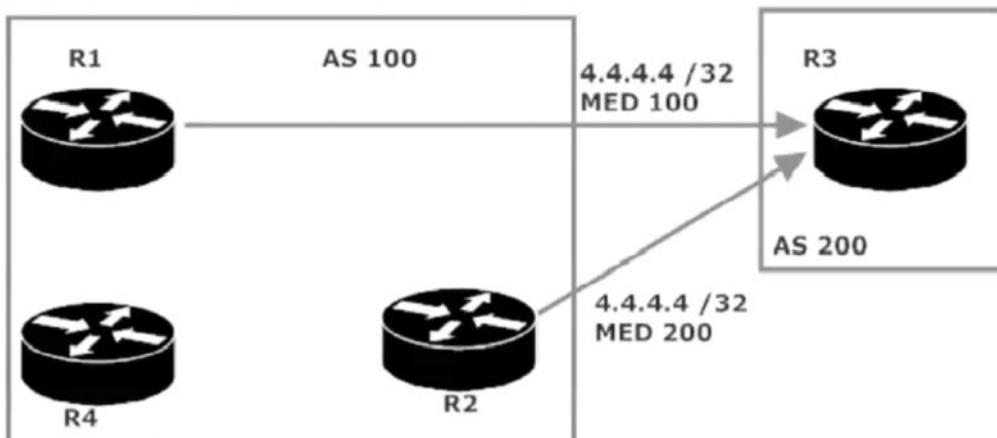
```
R3#show ip bgp 44.4.4.4
BGP routing table entry for 44.4.4.4/32, version 5
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
    1
  124
    172.12.23.2 from 172.12.23.2 (172.12.124.2)
      Origin IGP, localpref 100, valid, external
  124
    172.12.13.1 from 172.12.13.1 (172.12.124.1)
      Origin IGP, localpref 100, valid, external, best
```

4/9/2018 11:17 AM - Screen Clipping

Remember the MED is optional so you first have to create it.

With everything else equal, it all comes down to those BGP IDs. The paths through R1 are being selected as best because R1's ID is lower than R2's. Let's balance that load a bit by having R3 use 172.12.13.1 as the next hop for traffic headed for 4.4.4.4, and 172.12.23.2 as the next hop for 44.4.4.4.

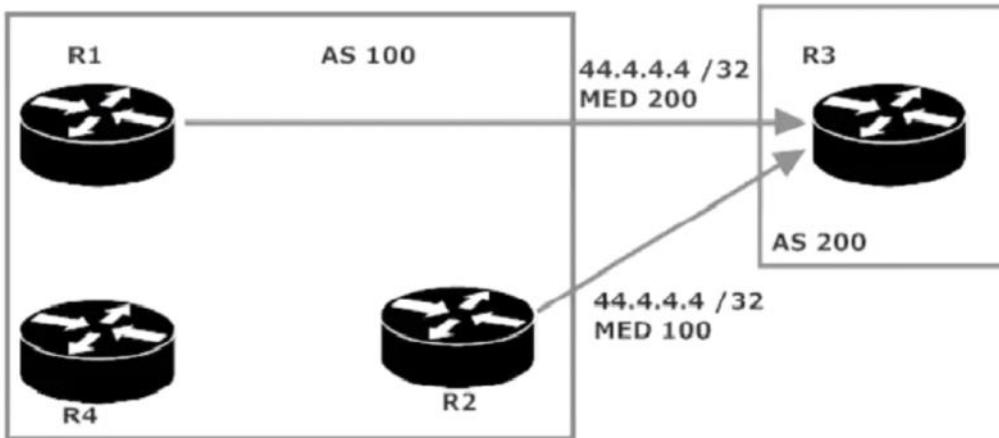
We want R3 to go through R1 to reach 4.4.4.4, so we'll have R1 advertise a MED of 100 for that route while R2 advertises a MED of 200. Lowest MED wins!



We'll do the reverse/inverse/universe for the other path, with R2 advertising the lowest MED for the 44.4.4.4 network.

4/9/2018 11:18 AM - Screen Clipping

This is similar to HSRP, where two values are set in the MED where the **lowest MED wins**.



We'll make this happen with a simple access list and a route map to apply the MEDs, first on R1. There is no "MED" available to set – the value we want to set is *metric*.

```
R1(config)#access-list 4 permit 4.4.4.4
R1(config)#access-list 44 permit 44.4.4.4
```

1st create you access lists to identify where the route maps will refer to.

```
R1#conf t
Enter configuration commands, one per line.
R1(config)#access-list 4 permit 4.4.4.4
R1(config)#
R1(config)#access-list 44 permit 44.4.4.4
R1(config)#

```

2nd create route map.

```
R1(config)#route-map SETMED permit 10
R1(config-route-map)#match ip address 4
```

3rd create MED

metric	Metric value for destination routing protocol
metric-type	Type of metric for destination routing protocol

```
|R1(config-route-map)#set metric 100
```

```
R1(config-route-map)#route-map SETMED permit 20
R1(config-route-map)#match ip address 44
R1(config-route-map)#set metric 200
R1(config-route-map)#

```

Now lets go to router 2.

```
R2#conf t  
Enter configuration commands, one per line  
R2(config)#access-list 4 permit 4.4.4.4  
R2(config)#  
R2(config)#access-list 44 permit 44.4.4.4  
R2(config)#  
R2(config)#route-map SETMED permit 10  
R2(config-route-map)#match ip add 4  
R2(config-route-map)#set met 200
```

```
R2(config-route-map)#route-map SETMED permit 20  
R2(config-route-map)#match ip address 44  
R2(config-route-map)#set met 100  
R2(config-route-map)#^Z
```

ACLs, route map, and MED written. Now we must apply via BGP config.

```
R2(config)#router bgp 124
```

```
R2(config-router)#neighbor 172.12.23.3 route-map SETMED out  
R2(config-router)#^Z  
R2#wr  
Building configuration
```

```
R1(config)#router bgp 124  
R1(config-router)#neighbor 172.12.13.3 route-map SETMED out  
R1(config-router)#  
R1(config-router)#{
```

Clear the route table.

```
R1#clear ip bgp * soft out  
R1#^Z
```

Verify.

```
R3#show ip bgp  
BGP table version is 9, local router ID is 172.12.23.3  
Status codes: s suppressed, d damped, h history, * valid, > best, i  
r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
      Network          Next Hop           Metric LocPrf Weight Path  
* 4.4.4.4/32        172.12.23.2       200    0 124 i  
*>                 172.12.13.1       100    0 124 i  
*> 44.4.4.4/32      172.12.23.2       100    0 124 i  
*                  172.12.13.1       200    0 124 i
```

You can now see how the MED is choosing .2 for the 44 network instead of .1, due to the MED.

```
R3#show ip bgp 44.4.4.4
BGP routing table entry for 44.4.4.4/32, version 8
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Flag: 0x4840
    Advertised to update-groups:
        1
        124
            172.12.23.2 from 172.12.23.2 (172.12.124.2)
                Origin IGP, metric 100, localpref 100, valid, external, best
        124
            172.12.13.1 from 172.12.13.1 (172.12.124.1)
                Origin IGP, metric 200, localpref 100, valid, external
```

The metrics are the MED.

BGP 12: Local Pref Changing all Routes

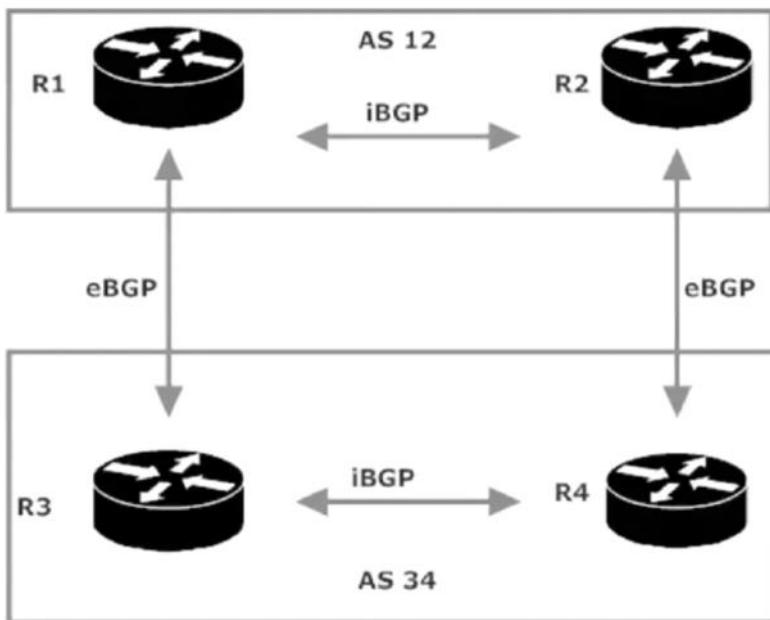
Monday, April 9, 2018 4:48 PM

The Local Preference Attribute

Also known as LOCAL_PREF, this well-known attribute also comes into play when multiple paths between ASes exist. However, the local pref attribute is just that... *local*. Where the MED tells routers outside the AS what entrance path is preferred, local preference tells routers inside the local AS which exit path to use when multiple paths exist. The path with the highest local preference is preferred.

The local preference value is passed only between iBGP peers, and local pref is never advertised outside the AS. Yes, the dreaded phrase "locally significant only" is back!

We'll tweak the local pref in our next lab. Just for fun, all four routers are on the same ethernet segment (10.1.1.0 /24), and all adjacencies are formed using addresses from that segment.



R1:

```
router bgp 12
neighbor 10.1.1.2 remote-as 12
neighbor 10.1.1.3 remote-as 34
```

R2:

```
router bgp 12
neighbor 10.1.1.1 remote-as 12
neighbor 10.1.1.4 remote-as 34
```

R3:

```
router bgp 34
```

```
network 172.12.34.0 mask 255.255.255.0
neighbor 10.1.1.1 remote-as 12
neighbor 10.1.1.4 remote-as 34
```

R4:

```
router bgp 34
network 172.12.34.0 mask 255.255.255.0
neighbor 10.1.1.2 remote-as 12
neighbor 10.1.1.3 remote-as 34
```

R3 and R4 are also connected via a direct serial link using the 172.12.34.0 /24 network. We're not using addresses from that link for adjacencies, but we are going to advertise the network into BGP on both R3 and R4.

```
R3(config)#router bgp 34
R3(config-router)#network 172.12.34.0 mask 255.255.255.0
```

```
R4(config)#router bgp 34
R4(config-router)#network 172.12.34.0 mask 255.255.255.0
```

R1 and R2 both have two paths to that network in their BGP tables, but there's a different next-hop address for each.

```

R1#show ip bgp
BGP table version is 2, local router ID is 10.
Status codes: s suppressed, d damped, h histor
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric
* i172.12.34.0/24    10.1.1.4           0
*>                  10.1.1.3           0
R1#
R1#
R1#
BRYANT_ADV_1#2
[Resuming connection 2 to r2 ... ]

R2#
R2#show ip bgp
BGP table version is 3, local router ID is 10.
Status codes: s suppressed, d damped, h histor
              r RIB-failure, S Stale, m multip
1
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric
* i172.12.34.0/24    10.1.1.3           0
*>                  10.1.1.4           0
R2#

```

```

R2#show ip bgp
BGP table version is 3, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best,
              r RIB-failure, S Stale, m multipath, b backup-path,
1
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric  LocPrf  Weight Path
* i172.12.34.0/24    10.1.1.3           0        100     0 34 i
*>                  10.1.1.4           0        100     0 34 i
R2#

```

So does .4 have a local pref of 0?

No. You just ned to set the network in the command.

```
R2#show ip bgp 172.12.34.0
BGP routing table entry for 172.12.34.0/24, version 3
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    1
  34
    10.1.1.3 from 10.1.1.1 (10.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal
  34
    10.1.1.4 from 10.1.1.4 (10.1.1.4)
      Origin IGP, metric 0, localpref 100, valid, external, best
R2#
```

The local pref for both routes is 100, so raising the local pref on R2 for the path with the next-hop of 10.1.1.3 should change the route selection. We can either raise the local preference for all routes advertised by a given router, or we can change the local pref of a particular router. First, we'll look at the all-or-nothing approach.

We know the route for 172.12.34.0 /24 with the next-hop of 10.1.1.3 is coming from R1, so the all-or-nothing approach requires us to change the BGP default local preference on R1 with *bgp default local-preference*.

```
R1(config-router)#bgp default ?
  ipv4-unicast      Activate ipv4-unicast for a peer by default
  local-preference  local preference (higher=more preferred)
  route-target      Control behavior based on Route-Target attributes

R1(config-router)#bgp default local-preference ?
<0-4294967295>  Configure default local preference value

R1(config-router)#bgp default local-preference 200
R1(config-router)#^Z
R1#clear ip
*Jun 12 21:22:01.421: %SYS-5-CONFIG_I: Configured from console by console

R1#clear ip bgp * soft out
```

R2 now has a local preference of 200 for the path advertised by R1, and has now selected that path to reach 172.12.34.0 /24.

4/11/2018 3:34 PM - Screen Clipping

```
R1#conf t
Enter configuration command
R1(config)#router bgp 12
R1(config-router)#
```

```
R1(config-router)#bgp default local-preference ?
<0-4294967295> Configure default local preference value
R1(config-router)#bgp default local-preference 200
R1(config-router)#^Z
```

```
R1#clear ip b
01:19:28: %SYS-5-CONFIG_I: Co
R1#clear ip bgp * soft out
R1#
```

```
R2#show ip bgp
BGP table version is 4, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
                r RIB-failure, S Stale, m multipath, b backup-path, x
                l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*>i172.12.34.0/24    10.1.1.3          0       200      0 34 i
*                   10.1.1.4          0           0 34 i
R2#
```

4/11/2018 3:36 PM - Screen Clipping

Router 2 is now using the path through router 1 down to router 3 with the .3 address because of the higher local pref. This has changed the local pref on all routes from Router 1.

4/11/2018 3:25 PM - Screen Clipping

BGP 13: Local Pref Changing Selected Routes

Wednesday, April 11, 2018 3:38 PM

Lets use route map to change local pref. Ive taken the local pref command off and a soft reset to update router 2.

```
R2#show ip bgp
BGP table version is 4, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale, m multipath, b backup-path,
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i172.12.34.0/24    10.1.1.3          0        100      0 34 i
* >                  10.1.1.4          0        100      0 34 i
* i210.3.3.0        10.1.1.3          0        100      0 34 i
* >                  10.1.1.4          0        100      0 34 i
* >
```

Lets change the local pref of just one route.

Create ACL. 7 is a random number.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 7 permit 172.12.34.0 0.0.0.255
R1(config)#
-->
```

Create Route Map.

```
R1(config)#route-map DOUBLEPREF permit 10
R1(config-route-map)#match ip add 7
```

```
|R1(config-route-map)#set local-pref 200
```

```
R1(config-route-map)#set local-pref 200
R1(config-route-map)#route-map DOUBLEPREF permit 20
R1(config-route-map)#set local-pref 100
R1(config-route-map)#
R1(config-route-map)#
R1(config-route-map)#exit
```

Apply it to the neighbor.

```
R1(config-router)#neighbor 10.1.1.2 route-map DOUBLEPREF out
R1(config-router)#^Z
```

Soft reset.

```

R2#show ip bgp
BGP table version is 5, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale, m multipath, b backup-path, x
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*->i172.12.34.0/24    10.1.1.3           0       200      0 34 i
*                  10.1.1.4           0       200      0 34 i
* i210.3.3.0         10.1.1.3           0       100      0 34 i
*>                  10.1.1.4           0       200      0 34 i
R2#show ip bgp 172.12.34.0
BGP routing table entry for 172.12.34.0/24, version 5
Paths: (2 available, best #1, table default)
      Advertised to update-groups:
          2
      34
          10.1.1.3 from 10.1.1.1 (10.1.1.1)
              Origin IGP, metric 0, localpref 200, valid, internal, best
      34
          10.1.1.4 from 10.1.1.4 (172.12.34.4)
              Origin IGP, metric 0, localpref 100, valid, external
R2#

```

4/11/2018 3:42 PM - Screen Clipping

Both paths are valid and we now see the change in local pref. When you get fluent with route maps, youll be able to do anything with BGP.

BGP 14: Weight Attribute

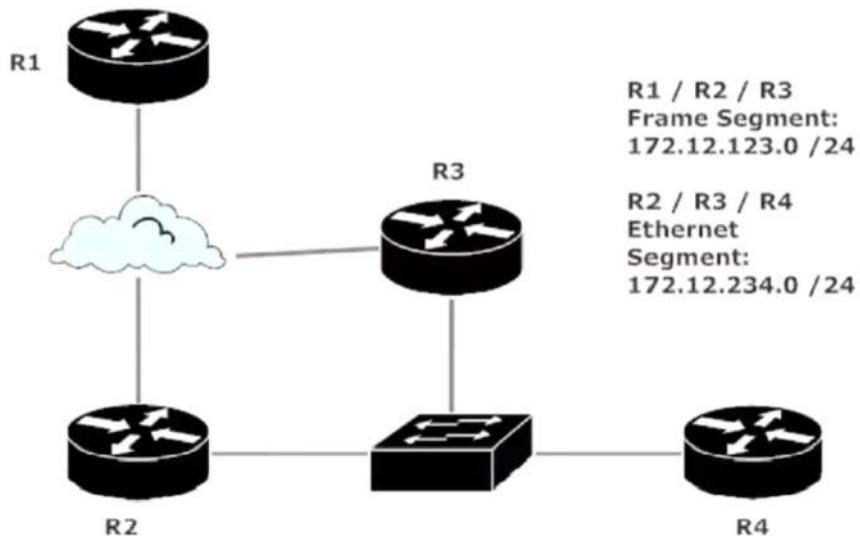
Wednesday, April 11, 2018 3:44 PM

The Weight Attribute

Quick weight facts: This attribute is Cisco-proprietary, is locally significant only, and is never advertised to any other router (iBGP or eBGP). The path with the larger weight is preferred. The default for this attribute is a bit odd. The default weight for a route originated on the local router is 32768, and for all other routes, it's zero.

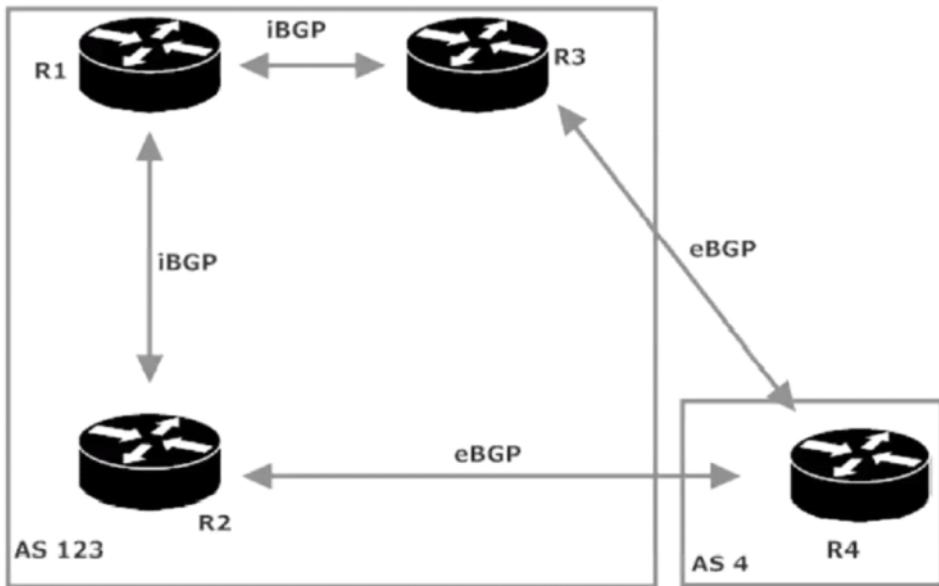
We'll manipulate the weight attribute in the following lab, which has two networks. The first is the very familiar 172.12.123.0 /24 network, connecting R1, R2, and R3. In this lab, R2, R3, and R4 are on the 172.12.234.0 /24 network.

WEIGHT ATTRIBUTE LARGER IS PREFERRED. 32768 IS THE DEFAULT.



4/11/2018 3:46 PM - Screen Clipping

The ASes:



4/11/2018 3:46 PM - Screen Clipping

```
R4#show ip bgp
BGP table version is 2, local router ID is 172.12.234.4
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S stale, m multipath, b backup-path,
               l
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
*> 4.4.4.4/32    0.0.0.0                  0        32768  i
```

4/11/2018 3:47 PM - Screen Clipping

Lets go to routers 2 and 3 to check the peerings.

```
R4#show ip bgp summ
BGP router identifier 172.12.234.4, local AS number 4
BGP table version is 2, main routing table version 2
1 network entries using 136 bytes of memory
1 path entries using 52 bytes of memory
1/1 BGP path/bestpath attribute entries using 124 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 312 total bytes of memory
BGP activity 1/0 prefixes, 1/0 paths, scan interval 60 secs
```

Neighbor /PfxRcd	V	AS	MsgRcvd	MsgSent	TblVer	InQ	outQ	Up/Down
172.12.234.2 0	4	123	40	41	2	0	0	00:33:39
172.12.234.3 0	4	123	36	40	2	0	0	00:33:29

R4#
BRYANT ADV 1#2

```
R2#show ip bgp
BGP table version is 2, local router ID is 172.12.234.2
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale, m multipath, b backup-path, >
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 4.4.4.4/32      172.12.234.4        0        0 4 i
R2#
```

```
R3#show ip bgp
BGP table version is 2, local router ID is 172.12.234.3
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 4.4.4.4/32      172.12.234.4        0        0 4 i
R3#
```

Notice the weights are 0.

```
R1#
R1#show ip bgp
BGP table version is 6, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i4.4.4.4/32       172.12.234.4        0      100      0 4 i
* i                  172.12.234.4        0      100      0 4 i
R1#
```

Notice there's no best route!

How could you remedy this if you could not allow router one to resolve this issue?

```
R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 6
Paths: (2 available, no best path)
  Not advertised to any peer
  4
    172.12.234.4 (inaccessible) from 172.12.123.2 (172.12.234.2)
      Origin IGP, metric 0, localpref 100, valid, internal
  4
    172.12.234.4 (inaccessible) from 172.12.123.3 (172.12.234.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

We see that the route is inaccessible. Because of the **next hop** router 1 has no idea by default where 234 is because you CANT assume were running a routing protocol to advertise. Router 1 gets the next hop address from direct connections. What if you could only use BGP connection to make this happen? **Next hop self**.

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router bgp 123
R2(config-router)#neighbor 172.12.123.1 next-hop-self
R2(config-router)#^Z
R2#wr
Building configuration...
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 123
R3(config-router)#neighbor 172.12.123.1 next-hop-self
R3(config-router)#^Z
R3#wr
Building configuration...
```

```
R1#clear ip bgp * soft in
R1#
R1#show ip bgp
BGP table version is 8, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*>i4.4.4.4/32      172.12.123.2        0     100      0 4 i
* i                  172.12.123.3        0     100      0 4 i
```

Notice now both next hops have changed. You can see that the nh info from r3 updated, but not r2. That's why we did the soft clear.

```
R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 8
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  4
    172.12.123.2 from 172.12.123.2 (172.12.234.2)
      Origin IGP, metric 0, localpref 100, valid, internal, b
  4
    172.12.123.3 from 172.12.123.3 (172.12.234.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

With this we went all the way down to the rid because so many attributes are the same. But what if we wanted to use the weight attribute to assign paths.

You'll use the neighbor command, but you need to use it on the router since Weight is locally significant only. You have to do it on each router.

```
R1(config)#router bgp 123
```

```
|R1(config-router)#neighbor 172.12.123.3 weight 200  
<cr>
```

This will be the default route for all neighbors.

```
R1#show ip bgp  
BGP table version is 9, local router ID is 172.12.123.1  
Status codes: s suppressed, d damped, h history, * valid, > best,  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
      Network          Next Hop            Metric LocPrf Weight Path  
* i4.4.4.4/32      172.12.123.2      0       100      0 4 i  
*>i               172.12.123.3      0       100      200 4 i  
R1#
```

Notice the valid and best path has now changed to the higher weight.

-----End of
1/2-----

Same configs from last video.

```
R1#conf t  
Enter configuration commands, one per line. End  
R1(config)#router bgp 123  
R1(config-router)#neighbor 10.1.1.5 remote-as 5  
R1(config-router)#^Z
```

```
R5#conf t  
Enter configuration commands, one per line. End w  
R5(config)#router bgp 5  
R5(config-router)#neighbor 10.1.1.1 remote-as 123  
R5(config-router)#^Z
```

```
R5#show ip bgp  
BGP table version is 2, local router  
Status codes: s suppressed, d damped,  
              r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - i  
  
      Network          Next Hop  
*> 4.4.4.4/32      10.1.1.1  
R5#
```

Let's take the weight command off.

```
R1(config-router)#no neighbor 172.12.123.3 weight 200  
R1(config-router)#^Z
```

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 44 permit 44.4.4.4
R1(config)#
R1(config)#route-map WEIGHT200 permit 10
R1(config-route-map)#match ip add 44
R1(config-route-map)#set weight 200
R1(config-route-map)#route-map WEIGHT200 permit 20
R1(config-route-map)#^Z
R1#
```

Create an ACL and Route Map.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 123
R1(config-router)#route-map ?
% Unrecognized command
R1(config-router)#neighbor 172.12.123.3 route-map WEIGHT200 ?
  in  Apply map to incoming routes
  out Apply map to outbound routes

R1(config-router)#neighbor 172.12.123.3 route-map WEIGHT200 in
R1(config-router)#^Z
R1#
R1#
R1#cle
01:49:32: %SYS-5-CONFIG_I: Configured from console by console
R1#clear ip bgp * soft in
R1#
R1#show
```

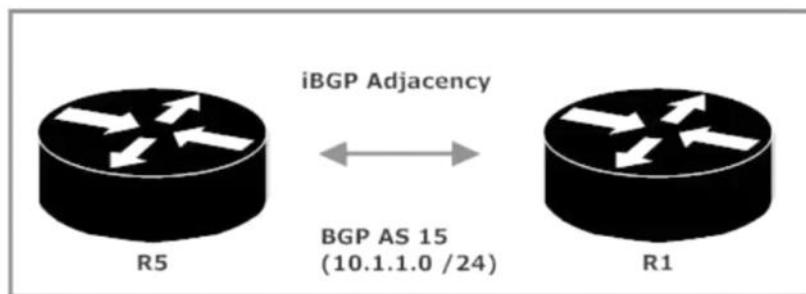
```
R1#show ip bgp
BGP table version is 17, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i4.4.4.4/32        172.12.123.3      0       100      0 4 i
*>i                 172.12.123.2      0       100      0 4 i
*>i4.4.4.4/32        172.12.123.3      0       100     200 4 i
* i                 172.12.123.2      0       100      0 4 i
R1#
```

Summarizing / Aggregating BGP Routes

The “ones and zeroes” part of summarizing BGP routes works exactly the same way as does summarization with EIGRP and OSPF. We just need to write the routes out in binary, identify the common bits, and add those bits up to get the route and mask.

There are some choices with BGP route aggregation that aren’t present with OSPF and EIGRP. When we configured summarization with OSPF and EIGRP, the interface sent out only the summary route and mask. We can do just that with BGP, or we can send out the aggregate route *and* the more-specific routes. We’ll see that in action in the following lab, using the link between R1 and R5 from the previous lab.



```
R1#show ip bgp
BGP table version is 17, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*->i16.0.0.0        10.1.1.5           0       100    0 i
*->i17.0.0.0        10.1.1.5           0       100    0 i
*->i18.0.0.0        10.1.1.5           0       100    0 i
*->i19.0.0.0        10.1.1.5           0       100    0 i
R1#
```

```
16 00010000
17 00010001
18 00010010
19 00010011
```

We know the drill to get the summary route – just add the common bits, which gives us 16. The summary route is 16.0.0.0, but what of the mask? Just put a “1” in the mask for every common bit and a “0” for the others, which gives us 11111100 00000000 00000000 00000000. In dotted decimal, that’s 252.0.0.0. Now we just need to introduce the aggregate route into our BGP domain and we’re gold. We’ll do that on R5 with the aggregate-address command.

```
R5(config)#router bgp 15
```

```
address-family
```

```
address-family      Enter Address Family Command mode
aggregate-address  Configure BGP aggregate entries
auto-summary       Enable automatic network number summary
```

```
R5(config-router)#aggregate-address 16.0.0.0 252.0.0.0
```

```
R1#show ip bgp
BGP table version is 18, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*>i16.0.0.0      10.1.1.5          0        100    0 i
*>i16.0.0.0/6    10.1.1.5          0        100    0 i
*>i17.0.0.0      10.1.1.5          0        100    0 i
*>i18.0.0.0      10.1.1.5          0        100    0 i
*>i19.0.0.0      10.1.1.5          0        100    0 i
R1#
```

The individual routes don't go away just because you summarize. This is the BGP default.

```
R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#router bgp 15
R5(config-router)#no aggregate-address 16.0.0.0 252.0.0.0
R5(config-router)#aggregate-address 16.0.0.0 252.0.0.0 ?
```

```
suppress-map     Conditionally filter more specific routes from updates
<cr>
```

If you ever need to filter some of the routes, but not all, you can use suppress map.

```
R5(config-router)#aggregate-address 16.0.0.0 252.0.0.0 summary-only
R5(config-router)#^Z
R5#w
*Nov  2 14:14:55.191: %SYS-5-CONFIG_I: Configured from console by console
R5#wr
Building configuration...
```

```
R1#show ip bgp
BGP table version is 24, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*>i16.0.0.0/6    10.1.1.5          0        100    0 i
```

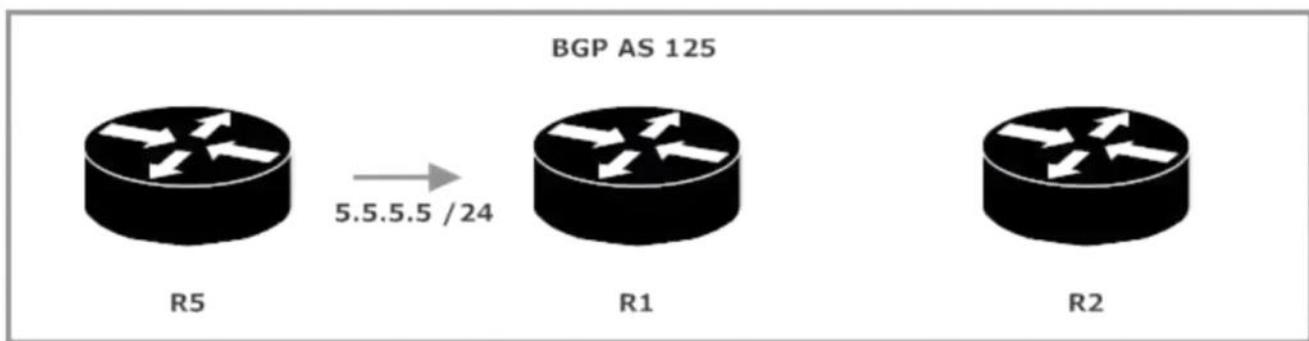
Using summary only removes all of the additional routes.

Internal BGP: Synchronization, Route Reflectors, Full Meshes, And Lack Thereof

The only circumstances under which a BGP speaker will advertise a route to an internal neighbor is if the route was created by the advertising router via the *network* command, static route redistribution, IGP route redistribution, or if the advertised route is a connected route.

That sounds like a lot of circumstances, but one common circumstance is missing from that list. When a BGP router learns a route from an internal neighbor, *that same router cannot advertise the route to another internal neighbor*.

Let's put that theory to the test with this simple network. R5 is advertising 5.5.5.0 /24 to R1...



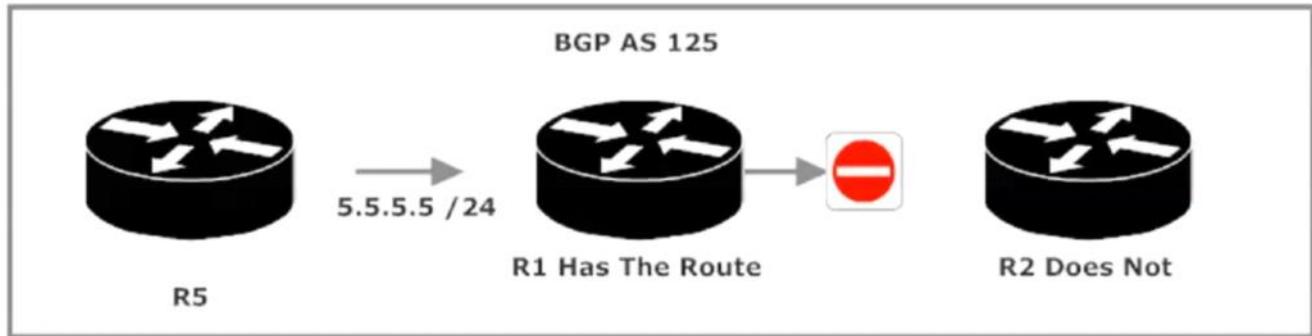
... verified with *show ip bgp 5.5.5.0* on R1.

```
R1#show ip bgp 5.5.5.0
BGP routing table entry for 5.5.5.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x208
    Not advertised to any peer
    Local
        10.1.1.5 from 10.1.1.5 (19.5.5.5)
            Origin IGP, metric 0, localpref 100, valid, internal, best
```

Looks like the theory is correct – check out “not advertised to any peer”, verified by *show ip bgp 5.5.5.0* on R2.

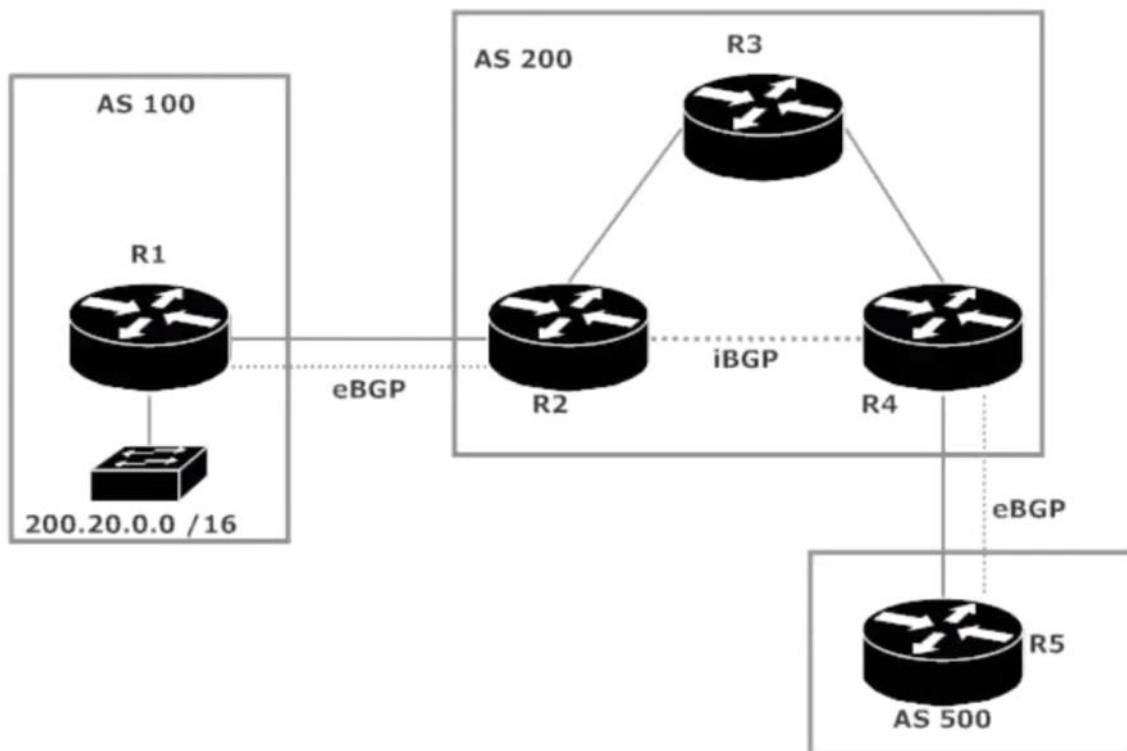
```
R2#show ip bgp 5.5.5.0
% Network not in table
```

Router doesn't have the route.



This looks to be an *extremely* restrictive rule. In theory, this would mean we'd need a full mesh in every AS in order for routes to be propagated properly and punctually. (Period.) In real-world networking, that would be an unbelievable amount of overhead. Luckily, BGP gives us a way around that literally logical nightmare. We'll see that solution in action shortly.

Right now, let's have a look at BGP's *rule of synchronization*. This rule only comes into play when an AS is a transit area and if there are non-BGP speakers in the transit area... like this!



AS 200 is serving as a transit area between AS 100 and AS 500; the only iBGP neighbor relationship in AS 200 is between R2 and R4. Problem is, AS 200 is a classic hub-and-spoke config where all data sent from spoke to spoke (R2-R4, R4-R2) must go through the hub (R3). Since R3 is not running BGP, it can't possibly know about that network, so R3 will drop packets destined for 200.20.0.0 /16.

Without the synch rule, R4 would advertise a path to 200.20.0.0 over its eBGP connection to R5. As you'd expect, packets sent by R5 to 200.20.0.0 would be

4/17/2018 8:55 AM - Screen Clipping

dropped at R3.

The BGP rule of synchronization: A transit AS will not advertise a route until every router in the transit AS has that same route in its IGP routing table. In this case, R4 will not send an advertisement for 200.20.0.0 /16 to R5 until R4 hears an advertisement for that network from R3 via an IGP. That advertisement indicates the non-BGP-speaking R3 has a route for that network.

Synchronization's major benefit is that packets that can't possibly reach the desired destination will not even be sent, reducing both the amount of unnecessary traffic and the unnecessary strain on router resources. Why send packets if they can't get where they need to go?

As you've likely noticed, BGP synchronization has been turned off by default in our configs. That's the case as of IOS 12.2(8). If BGP synch is on, it's safe to turn it off if all the routers in the AS are running BGP, if there's a full mesh in the AS, or if the AS in question isn't a transit AS. To disable BGP synch, just run the *no synch* command.

```
R5(config)#router bgp 5
R5(config-router)#no synch
```

BGP Split Horizon And Full Mesh Deployments

You'd expect BGP split horizon to work just a little differently than EIGRP split horizon. You'd be right. BGP split horizon states that one a BGP speaker cannot learn a route from an iBGP peer and then advertise it to another iBGP peer. (Sounds familiar!) To work with that rule, we'd need a logical full mesh among all iBGP peers in every BGP AS, which is not a practical idea.

The reason we don't see many BGP full meshes is really the same reason we don't see many Frame Relay full mesh networks, and that's one simple word – *overhead*. Any full-mesh deployment of BGP is going to hammer your router resources. A full mesh will also need a ton of TCP connections, and the more routers you have, the more connections you'll need. The formula for determining the number of TCP connections needed for a full mesh:

$X (X-1) / 2$, with "x" being the number of routers

Consider an AS with 20 routers. $20 (20 - 1) / 2 = 190$. BGP requires 190 separate TCP connections for a 20-router AS. Add that to the administrative nightmare involved in creating and maintaining the full mesh, and you have quite the labor-intensive situation. In short, there are three really good reasons to avoid full-mesh iBGP deployments:

An unnecessarily large number of TCP connections are needed.

Those sessions suck up a lot of bandwidth.

Creating and maintaining the full mesh is time-intensive and a logical landmine which is likely to result in a lot more troubleshooting than you or I would care to do.

Having analyzed the problem, let's apply the solution... *route reflectors!*

Route Reflectors

A router configured as a BGP route reflector can take a route learned from an iBGP peer and advertise it to another iBGP peer. Take *that*, split horizon!

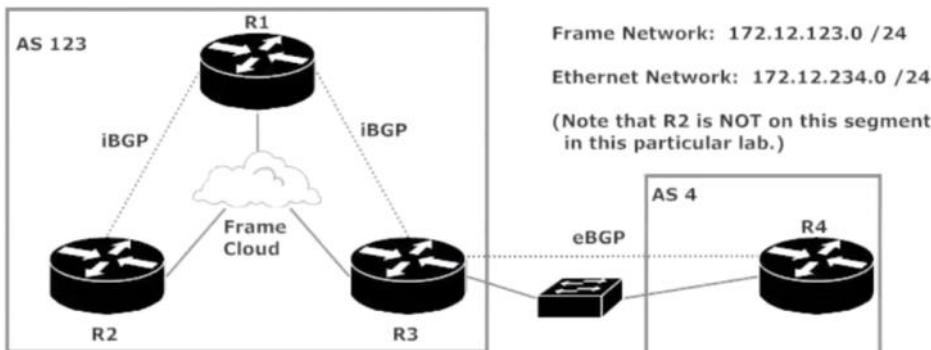
The iBGP peers that send routes to the route reflector are *clients*. When the RR receives a route from a client, the RR does just what you think it would do – it reflects the route to the other clients. The clients have no idea this is going on. The clients don't even know they are clients, and they require no additional configuration to make this happen. The only config you'll write is on the route reflector itself. Clients will have a peering with the RR, but not with each other, avoiding the full mesh mess.

A BGP speaker with a peering to a route reflector does not have to be a client. These speakers that are not clients are technically referred to as "nonclients". Nonclients *do* need a TCP connection to every other router in the AS.

Enough talk – let's reflect some routes!

Now a BGP speaker with a peering to a reflector does not have to be a client and we have a really clever

name for the ones that are not clients we call them non clients.



```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#router bgp 4
R4(config-router)#network 4.4.4.4 mask 255.255.255.255
R4(config-router)#^Z
R4#
```

Advertise this route to BGP.

```
R4#show ip bgp
BGP table version is 4, local router ID is 172.12.234.4
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale, m multipath, b backup-path,
]
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 4.4.4.4/32        0.0.0.0                  0        32768 i
```

Router 4 is now advertising it.

```
R3#show ip bgp
BGP table version is 4, local router ID is 172.12.234.3
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale, m multipath, b backup-path,
]
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 4.4.4.4/32        172.12.234.4          0        0 4 i
```

Router 3 has also recognized this is the best and valid path.

```
R1#show ip bgp
BGP table version is 1, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 4.4.4.4/32        172.12.234.4          0        100      0 4 i
```

On router 1 however, the entry is valid but not best. The next hop is still 234.4 because the route is inaccessible.

```
R1#show ip bgp
BGP table version is 1, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 4.4.4.4/32        172.12.234.4          0        100      0 4 i
R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  4
    172.12.234.4 (inaccessible) from 172.12.123.3 (172.12.234.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

In order to fix this we use **NEXTHOP SELF**

```
R3#
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 123
R3(config-router)#neighbo 172.12.123.1 next-hop-self
R3(config-router)#^Z
R3#wr
Building configuration...
```

```
R1#show ip bgp
BGP table version is 1, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i4.4.4.4/32      172.12.234.4        0       100      0 4 i
R1#clear ip bgp * soft in
R1#show ip bgp
BGP table version is 2, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best,
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*>i4.4.4.4/32     172.12.123.3        0       100      0 4 i
R1#
```

You will need to do this for most routes.

Router 2 doesn't have any BGP routes. That's because the routes are not being advertised from router 1.

```
*N
R2#
R2#show ip bgp
R2#
```

```
R1#show ip bgp 4.4.4.4
BGP routing table entry for 4.4.4.4/32, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
    4
      172.12.123.3 from 172.12.123.3 (172.12.234.3)
        Origin IGP, metric 0, localpref 100, valid, internal, best
R1#
```

Lets set a loopback on router 2 to be advertised via BGP.

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/D.
R2(config)#int loopback2
R2(config-if)#ip address 2.2.2
*Nov 5 13:19:36.883: %LINEPROTO-5-UPDOWN: Line protocol or
    changed state to up
R2(config-if)#ip address 2.2.2.2 255.255.255.255
R2(config-if)#router bgp 123
R2(config-router)#network 2.2.2.2 mask 255.255.255.255
R2(config-router)#^Z
R2#wr
Building configuration...

```

Now router 1 has the entry.

```

R1#show ip bgp
BGP table version is 3, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*->i2.2.2.2/32      172.12.123.2      0        100      0 i
*->i4.4.4.4/32      172.12.123.3      0        100      0 4 i
R1#
BRYANT_ADV_1#3
[Resuming connection 3 to r3 ... ]

```

Router 3 is still missing it though due to BGP split horizon. We want use a route reflector instead of physical route. Make Router the route reflector since it is the hub of the AS.

```

R1(config)#router bgp 123
R1(config-router)#neighbor 172.12.123.2 ?

```

<code>route-reflector-client</code>	Configure a neighbor as Route Reflector client
<code>send-community</code>	Send Community attribute to this neighbor

```

R1(config-router)#neighbor 172.12.123.2 route-reflector-client
R1(config-router)#neighbor 172.12.123.3
01:36:43: %BGP-5-ADJCHANGE: neighbor 172.12.123.2 Down RR client con
R1(config-router)#neighbor 172.12.123.3 route-reflector-client
R1(config-router)#
01:36:49: %BGP-5-ADJCHANGE: neighbor 172.12.123.2 Up
01:36:50: %BGP-5-ADJCHANGE: neighbor 172.12.123.3 Down RR client con
R1(config-router)#

```

You're adjacency will go down and then come back up.

```
R2#show ip bgp
BGP table version is 3, local router ID is 172.12.123.2
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S stale, m multipath, b backup-path,
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 2.2.2.2/32        0.0.0.0                  0          32768 i
*->i4.4.4.4/32       172.12.123.3            0         100      0 4 i
R2#
```

```
R1#show ip bgp summ
BGP router identifier 172.12.123.1, local AS number 123
BGP table version is 7, main routing table version 7
2 network entries and 2 paths using 266 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 3/17 prefixes, 5/3 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	Tblver	InQ	OutQ	Up/Down
172.12.123.2	4	123	36	35	7	0	0	00:01:26
172.12.123.3	4	123	39	35	7	0	0	00:01:17

```
R1#show ip bgp neighbor
BGP neighbor is 172.12.123.2, remote AS 123, internal link
  BGP version 4, remote router ID 172.12.123.2
  BGP state = Established, up for 00:01:43
  Last read 00:00:01, hold time is 180, keepalive interval 60
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Received 37 messages, 0 notifications, 0 in queue
  Sent 35 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 1
  Default minimum time between advertisement runs is 5 seconds

  For address family: IPv4 Unicast
  BGP table version 7, neighbor version 7
  Index 1, offset 0, Mask 0x2
  Route-Reflector Client
  1 accepted prefixes consume 36 bytes
  Prefix advertised 1, suppressed 0, withdrawn 0
  Number of NLRI's in the update sent: max 1, min 0

  Connections established 2; dropped 1
  Last reset 00:01:49, due to RR client config change
```

That's it for route reflector. All you need to know is WHERE to configure it.

Now, route reflectors don't just reflect to *everyone*, mind you. How a RR handles a routing update depends on the type of BGP router that sent that update. Save yourself some serious headaches from unnecessary troubleshooting and memorize this list!

Updates from RR clients are sent to all client and nonclient peers.

Updates from eBGP peers are sent to all client and nonclient peers. (I detect a pattern.)

Updates from nonclient peers are sent to all clients.

Once you have your BGP config in place, you may want to fine-tune the routes to be advertised. Or, perhaps, the routes *not* to be advertised. Either way, prefix lists

Prefix Lists And BGP

Cisco loves prefix lists for their high flexibility, their support for incremental updates, and the fact that writing BGP prefix lists is much more efficient than writing ACLs that filter BGP updates. (They're sure right on that last point!) BGP tables can be MUCH larger than any IGP table you'll ever see, and since prefix lists match only on the prefix of the address, the overall process is much faster than using ACLs.

A prefix list has several things in common with an ACL:

With both, if a route is not expressly permitted, it's denied.

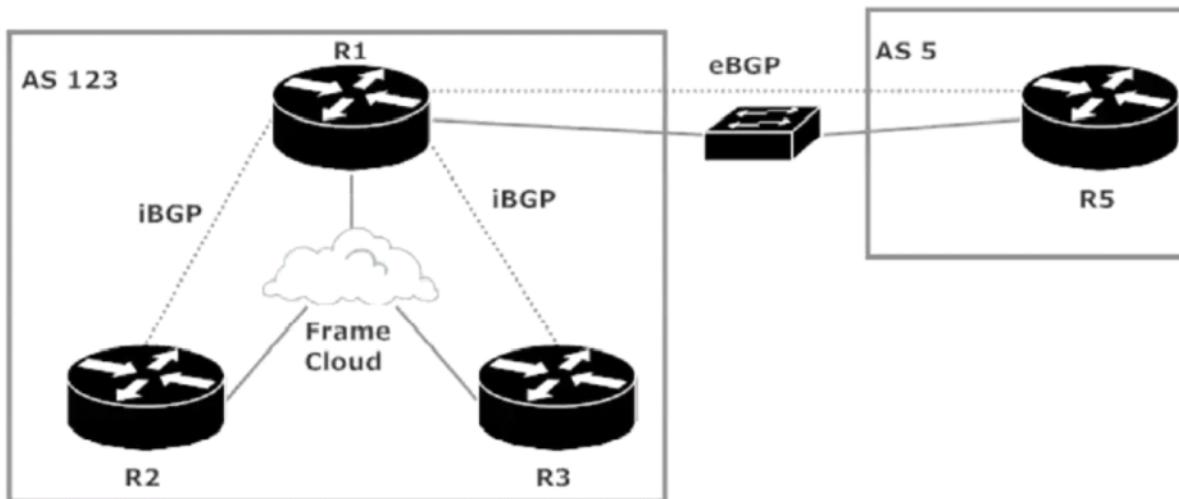
At the bottom of both a prefix list and an ACL, you'll find the implicit deny.

Explicit deny statements do not override the implicit deny.

Prefix lists work from top to bottom, and when a match is found, the process stops. It's vital the lines are in the correct order to do the job you want the prefix list to do.

Prefix list lines are numbered, with the lowest numbers at the top. Even if you and I (the network admins!) don't number the statements manually, the IOS will number them for you, incrementing by 5. This makes it easy for you to go back and add lines exactly where you need them.

Let's see prefix lists in action with the following network. The frame network is 172.12.123.0 /24 and the Ethernet segment is 10.1.1.0 /24.



On R5, we'll advertise the usual loopback of 5.5.5.5/32 along with four other networks.

```

R5#show ip bgp
BGP table version is 6, local router ID is 55.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 5.5.5.5/32        0.0.0.0                  0       32768  i
*-> 16.0.0.0          0.0.0.0                  0       32768  i
*-> 17.0.0.0          0.0.0.0                  0       32768  i
*-> 18.0.0.0          0.0.0.0                  0       32768  i
*-> 19.0.0.0          0.0.0.0                  0       32768  i
R5#
R5#

```

```

R1#show ip bgp
BGP table version is 6, local router ID is 172.12.123.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 5.5.5.5/32        10.1.1.5                 0       0 5 i
*-> 16.0.0.0          10.1.1.5                 0       0 5 i
*-> 17.0.0.0          10.1.1.5                 0       0 5 i
*-> 18.0.0.0          10.1.1.5                 0       0 5 i
*-> 19.0.0.0          10.1.1.5                 0       0 5 i
R1#

```

Routers 5 and 1 have valid and best routes. (NEXT HOP SELF)

```

R2#show ip bgp
BGP table version is 11, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S stale, m multipath, b backup-path, x
]
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i5.5.5.5/32         10.1.1.5                 0       100 0 5 i
* i16.0.0.0           10.1.1.5                 0       100 0 5 i
* i17.0.0.0           10.1.1.5                 0       100 0 5 i
* i18.0.0.0           10.1.1.5                 0       100 0 5 i
* i19.0.0.0           10.1.1.5                 0       100 0 5 i
R2#

```

```
R3#show ip bgp
BGP table version is 11, local router ID is 172.12.234.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale, m multipath, b backup-path, x
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop          Metric LocPrf Weight Path
* i5.5.5.5/32        10.1.1.5          0       100      0 5 i
* i16.0.0.0          10.1.1.5          0       100      0 5 i
* i17.0.0.0          10.1.1.5          0       100      0 5 i
* i18.0.0.0          10.1.1.5          0       100      0 5 i
* i19.0.0.0          10.1.1.5          0       100      0 5 i

```

Router 2 and 3 do not have a valid and best route.
Let's do next hop self.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 123
R1(config-router)#neighbor 172.12.123.2 next-hop-self
R1(config-router)#neighbor 172.12.123.3 next-hop-self
R1(config-router)#^Z
R1#wr
```

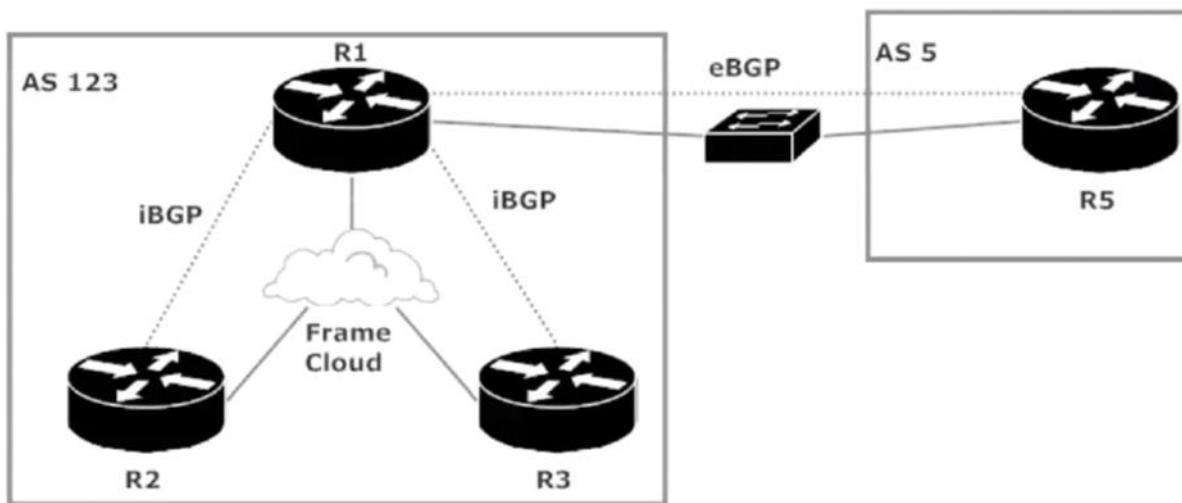
```
R2#show ip bgp
BGP table version is 16, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale, m multipath, b backup-path, x
               l
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop          Metric LocPrf Weight Path
*>i5.5.5.5/32        172.12.123.1        0       100      0 5 i
*>i16.0.0.0          172.12.123.1        0       100      0 5 i
*>i17.0.0.0          172.12.123.1        0       100      0 5 i
*>i18.0.0.0          172.12.123.1        0       100      0 5 i
*>i19.0.0.0          172.12.123.1        0       100      0 5 i
R2#
```

Now both 2 and 3 have valid and best. So how can we make 2 and 3 to not know of the existence of the 16, 17, 18, 19 routes, but we do want the 55 network and all networks added to router 5/advertised in the future.

The first problem is finding where we put the prefix list.

Let's see prefix lists in action with the following network. The frame network is 172.12.123.0 /24 and the Ethernet segment is 10.1.1.0 /24.



On R5, we'll advertise the usual loopback of 5.5.5.5/32 along with four other networks.

Router 1 is the HUB so the only place to put the filter is router 1.

First write your prefix list.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip prefix-list ?
WORD                  Name of a prefix list
sequence-number       Include/exclude sequence numbers in NVGEN

R1(config)#ip prefix-list NET5 ?
deny                 Specify packets to reject
description          Prefix-list specific description
permit               Specify packets to forward
seq                 sequence number of an entry
```

The purpose is to deny. Remember the implicit deny that exists as well.

```
R1(config)#ip prefix-list NET5 deny ?
A.B.C.D  IP prefix <network>/<length>, e.g., 35.0.0.0/8
```

```
R1(config)#ip prefix-list NET5 deny 16.0.0.0 /8
^
% Invalid input detected at '^' marker.

R1(config)#ip prefix-list NET5 deny 16.0.0.0/8
R1(config)#ip prefix-list NET5 deny 17.0.0.0/8
R1(config)#ip prefix-list NET5 deny 18.0.0.0/8
R1(config)#ip prefix-list NET5 deny 19.0.0.0/8
R1(config)#

```

Now we need to put in the prefix list equivalent of permit any.

```
R1(config)#ip prefix-list NET5 perm 0.0.0.0/0 ?
  ge Minimum prefix length to be matched
  le Maximum prefix length to be matched
<cr>

R1(config)#ip prefix-list NET5 perm 0.0.0.0/0 le ?
  <1-32> Maximum prefix length

R1(config)#ip prefix-list NET5 perm 0.0.0.0/0 le 32
R1(config)#

```

This rule matches everything other than the denys we've created.

-1/2-----

Lets now apply this prefix list

```
R1(config)#
R1(config)#router bgp 123
R1(config-router)#neighbor 172.12.123.2 ?

R1(config-router)#neighbor 172.12.123.2 prefix-list ?
  WORD  Name of a prefix list

R1(config-router)#neighbor 172.12.123.2 prefix-list NET5 ?
  in   Filter incoming updates
  out  Filter outgoing updates

R1(config-router)#neighbor 172.12.123.2 prefix-list NET5 out ?
<cr>

R1(config-router)#neighbor 172.12.123.2 prefix-list NET5 out
R1(config-router)#neighbor 172.12.123.3 prefix-list NET5 out
R1(config-router)#^Z
R1#
```

```
R2#show ip bgp
BGP table version is 20, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale, m multipath, b backup-path, x
               ]
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*->i5.5.5.5/32      172.12.123.1        0       100      0 5 i
R2#
```

Now the routes are filtered.

```
R3#show ip bgp
BGP table version is 20, local router ID is 172.12.234.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale, m multipath, b backup-path, x
               ]
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*->i5.5.5.5/32      172.12.123.1        0       100      0 5 i
R3#
```

Routes 16,17,18,19 are all gone but 5 is being advertised

BGP 19: Final Tips

Tuesday, April 17, 2018 9:41 AM

Private AS Numbers

You've surely noticed the rather large range of numbers available to us for an AS:

```
R1(config)#router bgp ?  
<1-65535> Autonomous system number
```

Some of those numbers are private AS numbers, or reserved for other reasons. The numbers 64496 – 65535 are considered private ASes, and just as private IP addresses should not be advertised to external networks, neither should private AS numbers. And no matter how hard you try, you can't assign AS Zero.

```
R1(config)#router bgp 0  
^  
% Invalid input detected at '^' marker.
```

By the way, some Cisco documentation uses the term "ASN" to refer to an AS number, and some doesn't. Be prepared to see the term "ASN" on your exam as well as just plain ol' "AS number".

4/17/2018 9:41 AM - Screen Clipping

```
--  
BGP table version is 2, local router ID is 172.12.123.1  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
              r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i5.5.5.5/32	10.1.1.5	0	100	0	i

The RID of the advertising router for a particular route can be seen with *show ip bgp X.X.X.X*.

```
R1#show ip bgp 5.5.5.5  
BGP routing table entry for 5.5.5.5/32, version 2  
Paths: (1 available, best #1, table Default-IP-Routing-Table)  
Flag: 0x820  
      Not advertised to any peer  
      Local  
      10.1.1.5 from 10.1.1.5 (55.55.55.55)  
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

The BGP RID follows much the same rules as the EIGRP and OSPF RIDs. The highest IP address assigned to a loopback is used as the BGP RID. If there's no loopback, the highest IP address assigned to an up/up physical interface is used. To hardcode the BGP RID, use the *bgp router-id* command. Yes, even though you're in BGP configuration mode, you still have to specify "bgp" in this command.

4/17/2018 9:44 AM - Screen Clipping

Security 1: Standard and Extended ACLs

Thursday, April 19, 2018 8:56 AM

Network Security Fundamentals

Just to be sure that everyone's totally up-to-speed on ACLs, I've included this review of basic ACL logic, standard ACLs, and extended ACLs. Even if you're comfortable with ACLs, I suggest strongly you review this material before moving ahead. We'll also have a look at time-based ACLs. Let's get started!

ACL Logic

When a packet enters or exits an interface with an ACL applied in the direction the packet is traveling, the packet is compared against the criteria of the ACL's initial line. If the packet matches, the appropriate action is taken, and the process is over. If there is no match, the second line is examined, and if it matches, the named action is taken and the process is done. If there's no match on the second line, the router keeps looking through the ACLs lines until a match is found.

If no explicit match is found in any line, the packet is denied via the implicit deny. If a packet is not expressly permitted, it's implicitly denied.

Configuring Standard ACLs and Extended ACLs

A standard ACL is concerned only with the source IP address of the packet. That's literally the only value that can be configured when writing a standard ACL. IOS Help

will make no mention of which address you're matching on (source or destination, that is), so you better have it down cold.

```
R5(config)#access-list 5 permit ?
  Hostname or A.B.C.D  Address to match
    any                Any source host
    host               A single host address
```

Extended ACLs consider both the source and destination IP address, and can consider the port number as well. Even if you don't care about the source and just want to match on destination, you'll have to put "any" in for the source. Plenty of practice coming up with that soon!

Standard and extended ACLs use separate numeric ranges. Standard ACLs use 1 – 99 and 1300 – 1999; extended ACLs use 100 – 199 and 2000 – 2699.

```
R5(config)#access-list ?
<1-99>          IP standard access list
<100-199>        IP extended access list
<1000-1099>      IPX SAP access list
<1100-1199>      Extended 48-bit MAC address access list
```

Standard ACLs are only concerned with the **source ip** of packet.

Extended ACLs consider source, destination, and port number if desired.

We have the two ranges because Cisco at one point believed this would be all we would ever need.
What was fine at one point isn't fine now.

<700-799>	48-bit MAC address access list
<800-899>	IPX standard access list
<900-999>	IPX extended access list
dynamic-extended	Extend the dynamic ACL absolute timer
rate-limit	Simple rate-limit specific access list

ACLs are applied to Cisco router interfaces with the *ip access-group* command, along with two important values. Let's say we've written ACL 5, permitting packets sourced from 172.12.12.0 /24 and denying all other sources. Here's what that ACL looks like:

```
R5(config)#access-list 5 permit 172.12.12.0 ?
  A.B.C.D  Wildcard bits
  log      Log matches against this entry
<cr>

R5(config)#access-list 5 permit 172.12.12.0 0.0.0.255
```

When applying the ACL, you must specify the ACL number and the direction in which the packets will be checked against the ACL.

```
R5(config)#int fast 0/0
R5(config-if)#ip access-group ?
  <1-199>    IP access list (standard or extended)
  <1300-2699> IP expanded access list (standard or extended)
```

ACLs are applied per **interface**.

ALWAYS USE WILDCARD MASKS WITH ACLS.

Acl>access group>interface.

And that's it! Verify with *show ip access-list*, *show access-list*, and / or *show ip interface* (a handy and often overlooked command) and you're gold!

```
R5(config)#int gig 0/0
R5(config-if)#ip access-group 5 out

R5#show ip int gig 0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is 5
  Inbound access list is not set
  (output truncated)
```

Using "Host" and "Any" in Wildcard Masks

A wildcard mask of all zeroes (0.0.0.0) means the address specified in the ACL line must be matched exactly in order for the specified action to be taken. You've likely used 0.0.0.0 as a wildcard mask in OSPF in order to enable only a specific address with OSPF. With ACLs, you can use the word host to represent a mask of 0.0.0.0 in either a standard or extended ACL.

Using host in a standard ACL:

```
R5(config)#access-list 7 permit ?
    Hostname or A.B.C.D  Address to match
    any                  Any source host
    host                 A single host address

R5(config)#access-list 7 permit host ?
    Hostname or A.B.C.D  Host address
```

Using host in an extended ACL:

```
R5(config)#access-list 7 permit host 172.12.12.3
R5(config)#access-list 177 permit ip host 172.12.12.3 ?
    A.B.C.D      Destination address
    any          Any destination host
    host         A single destination host
    object-group Destination network object group
```

At the other end of the spectrum, we have 255.255.255.255, which matches any address. Lines using this address are often written to negate the implicit deny..

```
R5(config)#access-list 8 permit 0.0.0.0 255.255.255.255
```

.. or to log packets that are denied.

```
R5(config)#access-list 9 deny 0.0.0.0 255.255.255.255 log
```

You can use the word any in place of that address and mask. It'll save you some typing, too! The following two ACLs do the exact same thing as ACLs 8 and 9 above.

```
R5(config)#access-list 8 permit any
R5(config)#access-list 9 deny any log
```

Watch The Order Of Your ACL Lines!

Getting just one line out of place in an ACL can wreck everything you're trying to do. Let's say we need an ACL that denies traffic from 172.18.18.0 /24 while allowing traffic from any other subnet. Some enterprising soul (me) has presented you with four different ACLs. Which is correct, and exactly why are the other three wrong?

Security 2: Host, Any, and Seeing Dollar Signs

Thursday, April 19, 2018 9:27 AM

Watch The Order Of Your ACL Lines!

Getting just one line out of place in an ACL can wreck everything you're trying to do. Let's say we need an ACL that denies traffic from 172.18.18.0 /24 while allowing traffic from any other subnet. Some enterprising soul (me) has presented you with four different ACLs. Which is correct, and exactly why are the other three wrong?

```
R5(config)#access-list 17 deny 172.18.18.0 0.0.0.255
R5(config)#access-list 17 perm any

R5(config)#access-list 18 perm any
R5(config)#access-list 18 deny 172.18.18.0 0.0.0.255

R5(config)#access-list 19 deny 172.18.18.0 255.0.0.0
R5(config)#access-list 19 perm any

R5(config)#access-list 20 perm any
R5(config)#access-list 20 deny 172.18.18.0 255.0.0.0
```

We know ACLs 19 and 20 can't be right, since the masks on each would match the last three octets while not caring what the first octet is. You're not going to want a wildcard mask like that very often.

That leaves ACLs 17 and 18. ACL 18 will match all traffic with its very first line ("*Hey, permit anything!*"). The second line denying the specific traffic will never be read. The ACL we want is 17, where the specific traffic is denied and then the remaining traffic is allowed.

More On Extended ACLs

Extended ACLs not only allow matches against the IP source and destination IP addresses, they demand both. Even if you're only matching against the destination, you still have to put something in for the source, even if it's any. (Yes, I told you this before. I'm just telling you again!)

Matching on source port, destination port, and protocol type are optional.

Let's write an extended ACL that denies traffic sourced from 172.50.50.0 /24 if it's destined for 172.50.100.0 /24. All other packets should be allowed.

```
R5(config)#access-list 100 deny ip ?
  A.B.C.D      Source address
  any          Any source host
```

```

host           A single source host
object-group  Source network object group

R5(config)#access-list 100 deny ip 172.50.50.0 ?
  A.B.C.D  Source wildcard bits

R5(config)#access-list 100 deny ip 172.50.50.0 0.0.0.255 ?
  A.B.C.D      Destination address
  any          Any destination host
  host         A single destination host
  object-group Destination network object group

R5(config)#access-list 100 deny ip 172.50.50.0 0.0.0.255 172.50.100.0 ?
  A.B.C.D  Destination wildcard bits

R5(config)#${ 100 deny ip 172.50.50.0 0.0.0.255 172.50.100.0 0.0.0.255 ?
  dscp        Match packets with given dscp value
  fragments   Check non-initial fragments
  log         Log matches against this entry
  log-input   Log matches against this entry, including input interface
  option      Match packets with given IP Options value
  precedence  Match packets with given precedence value
  time-range Specify a time-range
  tos         Match packets with given TOS value
<cr>

```

There are two odd things happening in that ACL config. Did you notice the dollar sign in front of the "100"? That symbol indicates that you've typed a command longer than the console screen can show you at once. In the last line, there are two "any" statements, and that's how you get rid of the implicit deny. The "permit any" statement in an extended ACL will have two anys – one for the source address, the next for the destination address. If you try to put just one any in, the router won't let you get away with it!

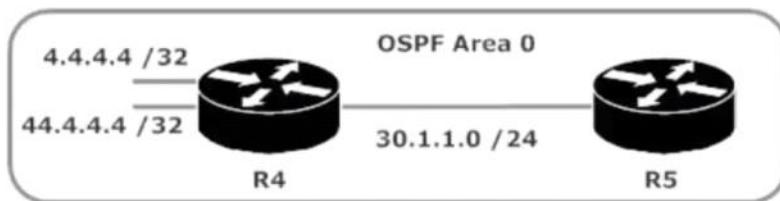
```
R5(config)#access-list 101 permit ip any
% Incomplete command.
```

Security 3: Extended ACL Lab

Thursday, April 19, 2018 9:40 AM

Let's get a log going with our ACLs! In the following lab, R4 has two loopbacks and has an OSPF adjacency with R5 via its FE interface. R5 sees both loopbacks in its OSPF route table and can ping both... for now!

Let's see extended ACLs in action! R4 has two loopbacks and has an OSPF adjacency with R5 via its FE interface. R5 sees both loopbacks in its OSPF route table and can ping both... for now!



```
R5#show ip route ospf
```

```
4.0.0.0/32 is subnetted, 1 subnets
O      4.4.4.4 [110/2] via 30.1.1.4, 00:11:17, GigabitEthernet0/0
        44.0.0.0/32 is subnetted, 1 subnets
O      44.4.4.4 [110/2] via 30.1.1.4, 00:11:07, GigabitEthernet0/0
R5#ping 4.4.4.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Lets deny traffic sourced from 30.1.1.0 that's destined for 4.4.4.4, permit everything else.

1. Create the ACL to deny and permit

```
R4(config)#access-list 144 deny ip 30.1.1.0 0.255.255.255 host 4.4.4.4
R4(config)#access-list 144 perm ip any any
R4(config)#

```

2. Apply group to interface. If you 4.4.4.4, it should time out.

```
R4(config)#int fast 0/0
R4(config-if)#ip access-group 144 ?
  in  inbound packets
  out outbound packets

R4(config-if)#ip access-group 144 in
R4(config-if)#^Z
R4#wr
Building configuration...

*Jan  1 01:55:27.043: %SYS-5-CONFIG_I: Co
R4#
```

```
R5#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4,
U.U.U
Success rate is 0 percent (0/5)
```

3. Verify

```
R4#show ip access-list
Extended IP access list 144
    10 deny ip 30.0.0.0 0.255.255.255 host 4.4.4.4 (5 matches)
    20 permit ip any any (15 matches)
R4#show ip access-list
Extended IP access list 144
    10 deny ip 30.0.0.0 0.255.255.255 host 4.4.4.4 (5 matches)
    20 permit ip any any (18 matches)
```

```
R4#show ip int fast 0/0
FastEthernet0/0 is up, line protocol is up
    Internet address is 30.1.1.4/24
    Broadcast address is 255.255.255.255
    Address determined by setup command
    MTU is 1500 bytes
    Helper address is not set
    Directed broadcast forwarding is disabled
    Multicast reserved groups joined: 224.0.0.5 224.0.0.6
    Outgoing access list is not set
    Inbound access list is 144
    Proxy ARP is enabled
    Local Proxy ARP is disabled
    Security level is default
    Split horizon is enabled
    ICMP redirects are always sent
    ICMP unreachables are always sent
    ICMP mask replies are never sent
    IP fast switching is enabled
    IP fast switching on the same interface is disabled
    IP Flow switching is disabled
    IP CEF switching is enabled
    IP CEF switching turbo vector
    IP multicast fast switching is enabled
```

You can see the inbound access list set. These are good general commands to know.

Security 4: Named ACL Lab

Thursday, April 19, 2018 9:47 AM

```
R4(config-ext-nacl)#deny ip 30.1.1.0 0.0.0.255 host 4.4.4.4
R4(config-ext-nacl)#perm ip any any
R4(config-ext-nacl)#exit
R4(config)#int fast 0/0
R4(config-if)#ip access-group ?
<1-199>      IP access list (standard or extended)
<1300-2699>   IP expanded access list (standard or extended)
WORD          Access-list name

R4(config-if)#ip access-group NO4 ?
    in  inbound packets
    out outbound packets

R4(config-if)#ip access-group NO4 in
R4(config-if)#^Z
R4#
Aug 29 13:32:44.687: %SYS-5-CONFIG_I: Configured from console by console
R4#show ip access-list
Extended IP access list NO4
  10 deny ip 30.1.1.0 0.0.0.255 host 4.4.4.4
  20 permit ip any any (1 match)
```

4/19/2018 9:58 AM - Screen Clipping

Security 5: Time-Based ACLs

Thursday, April 19, 2018 10:00 AM

Time based ACL's allow the ACL to take effect at a chosen time.

The ACL and time range command separate. You must reference the ACL name via time range commands.

```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#time-range ?
WORD  Time range name

R4(config)#time-range CCNP
R4(config-time-range)#?
Time range configuration commands:
 absolute    absolute time and date
 default     Set a command to its defaults
 exit        Exit from time-range configuration mode
 no          Negate a command or set its defaults
 periodic   Periodic time and date

R4(config-time-range)#

```

Absolute chooses when it starts and when it ends. Settings are in military time.

```
R4(config-time-range)#absolute ?
end      ending time and date
start    starting time and date

R4(config-time-range)#absolute start ?
hh:mm   Starting time

R4(config-time-range)#absolute start 09:00 ?
<1-31>  Day of the month

R4(config-time-range)#absolute start 09:00 31 ?
MONTH   Month of the year [eg: Jan for January, Jun for June]

R4(config-time-range)#absolute start 09:00 31 Aug ?
<1993-2035> Year

R4(config-time-range)#absolute start 09:00 31 Aug 2015 ?
end      ending time and date
<cr>
```

Periodic chooses to take effect per day weekday and weekend options

```

R4(config-time-range)#periodic mon ?
  Friday      Friday
  Saturday    Saturday
  Sunday      Sunday
  Thursday    Thursday
  Tuesday     Tuesday
  Wednesday   Wednesday
  hh:mm       Starting time

R4(config-time-range)#periodic weekdays ?
  hh:mm       Starting time

R4(config-time-range)#periodic weekdays 10:00 ?
  to          ending day and time

R4(config-time-range)#periodic weekdays 10:00 to ?
  hh:mm       Ending time - stays valid until beginning

R4(config-time-range)#periodic weekdays 10:00 to _

```

- ⌚ Verify. We see that the entry is **inactive**. The time is Aug 31 09:02. But the periodic time isn't within that range. This is one reason we need synched time (NTP).

```

R4#show time-
Aug 31 09:02:33.307: %SYS-5-CONFIG_I: Configured from console by con
R4#show time-range
time-range entry: CCNP (inactive)
  periodic weekdays 10:00 to 17:00
R4#
R4#
R4#

```

Apply to ACL.

```

R4(config)#access-list 145 permit tcp any any eq telnet time-range CCNP
R4(config)#^Z

```

```

R4#show ip access-list
Extended IP access list 145
  10 permit tcp any any eq telnet time-range CCNP (inactive)

```

Apply to telnet

```
R4#conf t
Enter configuration commands, one per line.  E
R4(config)#line vty 0 4
R4(config-line)#access-?
access-class

R4(config-line)#access-class ?
<1-199>      IP access list
<1300-2699>   IP expanded access list
WORD          Access-list name

R4(config-line)#access-class 145 ?
in    Filter incoming connections
out   Filter outgoing connections

R4(config-line)#access-class 145 in
R4(config-line)#^Z
R4#
Aug 31 09:05:39.631: %SYS-5-CONFIG_I: Configuration change detected, saving...
R4#wr
Building configuration...
```

Implicit deny will refuse the connection since its outside of the time frame.

```
R5#telnet 30.1.1.4
Trying 30.1.1.4 ...
% Connection refused by remote host

R5#
```

Set your clock within range to see this change.

```
R4#clock set ?
hh:mm:ss  Current Time

R4#clock set 11:00:00 Aug 31 2015
R4#
R4#
R4#
Aug 31 11:00:00.000: %SYS-6-CLOCKUPDA
6:40 UTC Mon Aug 31 2015 to 11:00:00
e by console.

R4#
R4#
R4#show time-range
time-range entry: CCNP (active)
  periodic weekdays 10:00 to 17:00
  used in: IP ACL entry
```

```
R4#show ip access-list
Extended IP access list 145
    10 permit tcp any any eq telnet time-range CCNP (active)
R4#
BRYANT_ADV_1#5
[Resuming connection 5 to r5 ... ]

R5#
R5#telnet 30.1.1.4
Trying 30.1.1.4 ... Open
```

Security 6 : Password Review and Telnet Lab

Monday, April 23, 2018 3:50 PM

```
R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#enable ?
  last-resort Define enable action if no TACACS servers respond
  password     Assign the privileged level password (MAX of 25
  secret       Assign the privileged level secret (MAX of 25 characters)
  use-tacacs   Use TACACS to check enable passwords

R5(config)#enable password CCNA
R5(config)#enable secret CCNP
R5(config)#^Z
```

Secret is the that's encrypted by default.

```
hostname R5
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$HFT3$hPg1U8EvigYkgU0rhs/cx0
enable password CCNA
!
no aaa new-model
```

Password is obsolete because it has no encryption.

```
line con 0
  exec-timeout 0 0
  logging synchronous
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
end

R5#
```

Logging Synch makes router hold onto console commands until you're done with input
Exec timeout stops from timing out of exec mode. Good for labs.

```
R4#  
R4#telnet 30.1.1.5  
Trying 30.1.1.5 ... Open
```

Password required, but none set

```
[Connection to 30.1.1.5 closed by foreign host]  
R4#
```

With login you have to have a password.

Your login is disabled until password is set.

```
R5#conf t  
Enter configuration commands, one per line. End with  
R5(config)#line vty 0 4  
R5(config-line)#no login  
R5(config-line)#  
R5(config-line)#  
R5(config-line)#login  
% Login disabled on line 578, until 'password' is set  
% Login disabled on line 579, until 'password' is set  
% Login disabled on line 580, until 'password' is set  
% Login disabled on line 581, until 'password' is set  
% Login disabled on line 582, until 'password' is set  
R5(config-line)#password CCNP  
R5(config-line)#^Z  
R5#wr  
Building configuration...
```

You are now in user exec mode.

```
R4#telnet 30.1.1.5  
Trying 30.1.1.5 ... Open
```

```
User Access Verification
```

Security 7 : Username/Password Database and Telnet

Monday, April 23, 2018 4:28 PM

```
R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#username chris password bryant
R5(config)#
```

These creds are found locally. When set, instead of prompting the telnet for password, it will first go to the local db. We can tell the VTY lines to look here instead of the telnet session. We can reference it with the local command.

```
R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#username chris password bryant
R5(config)#username lou password thebigsecret
R5(config)#
R5(config)#
R5(config)#line vty 0 4
R5(config-line)#login ?
  local  Local password checking
<cr>
R5(config-line)#login local
R5(config-line)#
```

We are now into the device, but only in user mode. How do we get to privileged access?

```
R4#telnet 30.1.1.5
Trying 30.1.1.5 ... open

User Access Verification

Username: chris
Password: _____
R5>
```

If we wanted to automatically log into a higher privilege level, we have to set it via the local database for the specified user.

```
R5(config)#username paul ?
  aaa                      AAA directive
  access-class              Restrict acce
  autocmd                  Automatically
  callback-dialstring      Callback dial
  callback-line             Associate a s
  callback-rotary           Associate a r
  dnis                      Do not requir
  nocallback-verify         Do not requir
  noescape                  Prevent the u
  nohangup                 Do not discon
  nopassword                No password i
  one-time                  Specify that
                            time
  password                 Specify the p
  privilege                Set user priv
  secret                   Specify the s
  user-maxlinks            Limit the use
  view                     Set view name
<cr>
```

```
R5(config)#username paul privilege ?
<0-15> User privilege level
```

```
R5(config)#username paul privilege 15 password ?
  0    Specifies an UNENCRYPTED password will follow
  7    Specifies a HIDDEN password will follow
  LINE The UNENCRYPTED (cleartext) user password
```

```
R5(config)#username paul privilege 15 password jones
R5(config)#^Z
```

```
R5#
*Aug 31 17:18:40.491: %SYS-5-CONFIG_I: Configured from console by console
R5#
BRYANT_ADV_1#4
[Resuming connection 4 to r4 ... ]
```

```
R4#
R4#
R4#telnet 30.1.1.5
Trying 30.1.1.5 ... open
```

```
User Access Verification
```

```
R4#telnet 30.1.1.5
Trying 30.1.1.5 ... Open

User Access Verification

Username: chris
Password:
R5>exit

[Connection to 30.1.1.5 closed]
R4#telnet 30.1.1.5
Trying 30.1.1.5 ... Open

User Access Verification

Username: paul
Password:
R5#
```

Notice the prompt difference after login. Paul is at a higher privilege.

Of course you can use the **service password encryption** to hide all of your plain text passwords.

```
R5(config)#service password-encryption
R5(config)#^Z
R5#
!
license udi pid CISCO3825 sn FTX1003C2XM
username chris password 7 121B170E130518
username lou password 7 120D0D120111
username paul privilege 15 password 7 045104080A32
!
redundancy
!
```

Security 8: Secure Shell

Monday, April 23, 2018 4:37 PM

Putting The "Secure" In Secure Shell

There's a huge problem with Telnet in that it sends all data in an unencrypted format, including any and all passwords involved.



Any would-be network intruder who gets their hands on that info is then a will-be network intruder, and the trouble has just begun. Secure Shell is basically encrypted Telnet. The basic operation of each is similar, but Secure Shell encrypts all the data involved in the transaction, including the password.



All info over a telnet session is unencrypted, thus why we use SSH which encrypts all data including the passwords.

The obvious question: "Why do we still use Telnet?" Telnet is easier to set up, and that's part of its appeal, but the real issue tends to be with hardware. SSH takes a little more work, which we don't mind, but it may require a hardware upgrade that's not in your budget. Should your routers be up to speed on SSH, you can allow only SSH logins with the *transport input ssh* command on your VTY lines.

```
R1(config-line)#transport input ?
  all      All protocols
  lat      DEC LAT protocol
  mop      DEC MOP Remote Console Protocol
  none    No protocols
  pad      X.3 PAD
  rlogin   Unix rlogin protocol
  ssh      TCP/IP SSH protocol
  telnet   TCP/IP Telnet protocol
  udptn    UDPTN async via UDP protocol
  v120    Async over ISDN
```

```
R1(config-line)#transport input ssh
```

```
R5#conf t
Enter configuration commands, one per line
R5(config)#line vty 0 4
R5(config-line)#transport input ?
  all      All protocols
  lapb-ta  LAPB Terminal Adapter
  lat      DEC LAT protocol
  mop      DEC MOP Remote Console Protocol
  none    No protocols
  pad      X.3 PAD
  rlogin   Unix rlogin protocol
  ssh      TCP/IP SSH protocol
  telnet   TCP/IP Telnet protocol
  udptn   UDPTN async via UDP protocol
  v120    Async over ISDN
```

Putting SSH means it will only accept an SSH connection.

```
R5(config-line)#transport input ssh
R5(config-line)#^Z
```

```
R4#
R4#telnet 30.1.1.5
Trying 30.1.1.5 ...
% Connection refused by remote host
```

At the very least, your SSH config will require a username/password database on the local router. You can set up exterior devices to handle this authentication, including an AAA server. If you're using the local database, configure *login local* on your VTY lines.

You'll also need to specify the router's domain name with *ip domain-name*. Once that's in place, run *crypto key generate rsa*, and you're good when you see the message regarding SSH being enabled.

```
R1(config)#ip domain-name BRYANTADVANTAGE.COM
```

```
R1(config)#crypto key generate rsa
The name for the keys will be: R1.BRYANTADVANTAGE.COM
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
```

```
Jul 12 18:59:29.355: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

A quick note before we proceed: There's some cutover between the SWITCH and ROUTE exams when it comes to SNMP and the Network Time Protocol (NTP). When it comes to NTP, the ROUTE exam seems to be only concerned with securing it. Instead of just telling you how to secure NTP, I've included my full NTP lesson from my SWITCH book, which is more than enough for ROUTE. I'd rather you knew all about NTP than just how to secure it. Both of the following sections will show configs on 13 switches, and the commands are exactly the same on routers

```
R5(config)#ip domain-name BRYANTADVANTAGE.COM
R5(config)#
R5(config)#crypto key generate rsa ?
  encryption      Generate a general purpose RSA key pair for signing and
                    encryption
  exportable       Allow the key to be exported
  general-keys    Generate a general purpose RSA key pair for signing and
                    encryption
  label            Provide a label
  modulus          Provide number of modulus bits on the command line
  on               create key on specified device.
  redundancy      Allow the key to be synced to high-availability peer
  signature        Generate a general purpose RSA key pair for signing and
                    encryption
  storage          Store key on specified device
  usage-keys      Generate separate RSA key pairs for signing and encryption
<cr>
```

```
R5(config)#crypto key generate rsa
The name for the keys will be: R5.BRYANTADVANTAGE.COM
Choose the size of the key modulus in the range of 360 to 2048 for
General Purpose Keys. Choosing a key modulus greater than 512 may
a few minutes.
```

```
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```
R5(config)#[
```

```
R5(config)#
*Aug 31 17:54:10.355: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

SSH is complete.

A quick note before we proceed: There's some cutover between the SWITCH and ROUTE exams when it comes to SNMP and the Network Time Protocol (NTP). When it comes to NTP, the ROUTE exam seems to be only concerned with securing it. Instead of just telling you how to secure NTP, I've included my full NTP lesson from my SWITCH book, which is more than enough for ROUTE. I'd rather you knew all about NTP than just how to secure it. Both of the following sections will show configs on L3 switches, and the commands are exactly the same on routers.

In short, I'd rather give you full information on a topic than just a little. Having said that, let's have at it!

Security 8.1 : SNMP

Monday, April 23, 2018 4:45 PM

SNMP

The Simple Network Management Protocol is used to carry network management info from one network device to another, and you'll find it in just about every network out there today. An SNMP deployment has three main parts:

The SNMP Manager, the actual monitoring device

The SNMP Agents, the devices being monitored (and running an SNMP instance)

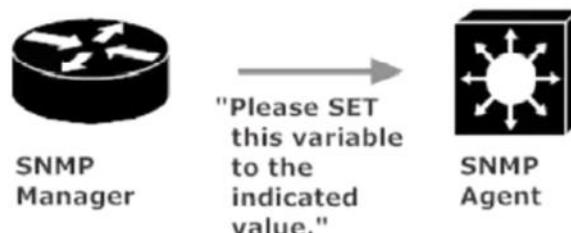
The Management Information Base (MIB), the database on the Agent that contains important information ("variables") about the Agent.

SNMP Managers *poll* Agents over UDP port 161, and these messages take the form of GETs and SETs. A "GET" is a request for information...



... and a "SET" is a request from the Manager to the Agent, requesting a certain

variable be set to the value indicated in the SET.

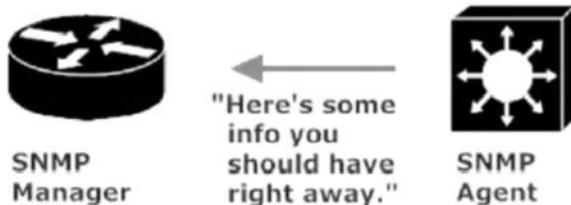


Seems like a good approach, but there's one glaring issue. The only way for the Manager to receive immediate or even near-immediate notice of a critical network event is to poll the Agents quite often, which in turn sucks up bandwidth and is a hit on the Manager's CPU.

Let's say our Manager is polling our Agent every 10 minutes regarding one particular variable. Three seconds after the Agent answers one such GET, that variable undergoes a critical change. It would then take 9 minutes and 57 seconds for the Manager to find out about the change!

To get a quick notification on such an event without overloading the Manager, we configure *SNMP traps* on the managed devices, allowing the Agents to send a message to the Manager when such a variable changes.

To get a quick notification on such an event without overloading the Manager, we configure *SNMP traps* on the managed devices, allowing the Agents to send a message to the Manager when such a variable changes.



We still have three versions of SNMP out there – versions 1, 2c, and 3 – and there are some serious security concerns with the earlier versions. V3 has both authentication and encryption capabilities; the earlier versions do not. For that reason alone, you should use V3 whenever possible, and the use of the other versions should be restricted to allowing read-only access via the use of *community strings*.

SNMP community strings, found in SNMP v1 and 2c, are a kind of password / authority level combination that allow you to set the strings as read-only or read-write.

```
MLS_1(config)#snmp-server community ?  
WORD  SNMP community string
```

```
MLS_1(config)#snmp-server community CCNP ?
```

```
MLS_1(config)#snmp-server community ?
WORD SNMP community string

MLS_1(config)#snmp-server community CCNP ?
<1-99> Std IP accesslist allowing access with this community string
<1300-1999> Expanded IP accesslist allowing access with this community
string
WORD Access-list name

ro Read-only access with this community string
rw Read-write access with this community string
view Restrict this community to a named MIB view
<cr>

MLS_1(config)#snmp-server community CCNP ro ?
<1-99> Std IP accesslist allowing access with this community string
<1300-1999> Expanded IP accesslist allowing access with this community
string
WORD Access-list name
ipv6 Specify IPv6 Named Access-List
<cr>

MLS_1(config)#snmp-server community CCNP ro 15
```

This configuration would allow hosts identified by ACL 15 to have read-only access to all SNMP objects specified by this community string.

With SNMP v3, things are much more secure and just a tad more complex. Let's use IOS Help to venture through some of the most long-winded commands you're ever going to see. Let's start with creating an SNMP group and then assigning a user to that group.

With SNMP v3, things are much more secure and just a tad more complex. Let's use IOS Help to venture through some of the most long-winded commands you're ever going to see. Let's start with creating an SNMP group and then assigning a user to that group.

```
MLS_1(config)#snmp-server group BULLDOGS ?
v1    group using the v1 security model
v2c   group using the v2c security model
v3    group using the User Security Model (SNMPv3)
```

```
MLS_1(config)#snmp-server group BULLDOGS v3 ?
auth   group using the authNoPriv Security Level
noauth group using the noAuthNoPriv Security Level
priv   group using SNMPv3 authPriv security level
```

A quick word about those three security levels – they look intimidating, but when you break them down they're easy to remember.

authNoPriv – You have **authentication**, but no **privacy** (no encryption)

noAuthNoPriv – You're really asking for it. You have no **authentication** and no **privacy** (encryption).

authPriv – Your SNMP packets are both **authenticated** and **privacy** is assured via encryption.

```
MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth ?
md5  Use HMAC MD5 algorithm for authentication
sha  Use HMAC SHA algorithm for authentication
```

```
MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth sha ?
WORD authentication password for user
```

```
MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth sha CCNP ?
access specify an access-list associated with this group
priv   encryption parameters for the user
<cr>
```

```
MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth sha CCNP priv ?
3des  Use 168 bit 3DES algorithm for encryption
aes   Use AES algorithm for encryption
des   Use 56 bit DES algorithm for encryption
```

```
MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth sha CCNP priv aes ?
128   Use 128 bit AES algorithm for encryption
192   Use 192 bit AES algorithm for encryption
256   Use 256 bit AES algorithm for encryption
```

```
? MLS_1(config)#snmp-server user CHRIS BULLDOGS v3 auth sha CCNP priv aes 128
WORD privacy password for user
```

```
MLS_1(config)#$S BULLDOGS v3 auth sha CCNP priv aes 128 TIREDOFTYPING ?
access specify an access-list associated with this group
<cr>

MLS_1(config)#$S BULLDOGS v3 auth sha CCNP priv aes 128 TIREDOFTYPING
MLS_1(config)#+Z
MLS_1#
Mar 26 10:16:25.467: Configuring snmpv3 USM user, persisting snmpEngineBoots.
```

Finally, we'll define the host to which to send traps.

```
MLS_1(config)#snmp-server host ?
WORD IP/IPv6 address of
SNM notification host
http://<Hostname or A.B.C.D>[:<port number>] [/<uri>] HTTP address of XML
notification host

MLS_1(config)#snmp-server host 10.1.1.3 ?
WORD SNMPv1/v2c community string or SNMPv3 user name
informs Send Inform messages to this host
traps Send Trap messages to this host
version SNMP version to use for notification messages
vrf VPN Routing instance for this host
```

```
MLS_1(config)#snmp-server host 10.1.1.3 ?
WORD      SNMPv1/v2c community string or SNMPv3 user name
informs   Send Inform messages to this host
traps     Send Trap messages to this host
version   SNMP version to use for notification messages
vrf       VPN Routing instance for this host

MLS_1(config)#snmp-server host 10.1.1.3 traps ?
WORD      SNMPv1/v2c community string or SNMPv3 user name
version   SNMP version to use for notification messages

MLS_1(config)#snmp-server host 10.1.1.3 traps version ?
1    Use SNMPv1
2c   Use SNMPv2c
3    Use SNMPv3

MLS_1(config)#snmp-server host 10.1.1.3 traps version 3 ?
auth    Use the SNMPv3 authNoPriv Security Level
noauth  Use the SNMPv3 noAuthNoPriv Security Level
priv    Use the SNMPv3 authPriv Security Level

MLS_1(config)#snmp-server host 10.1.1.3 traps version 3 priv ?
WORD    SNMPv1/v2c community string or SNMPv3 user name

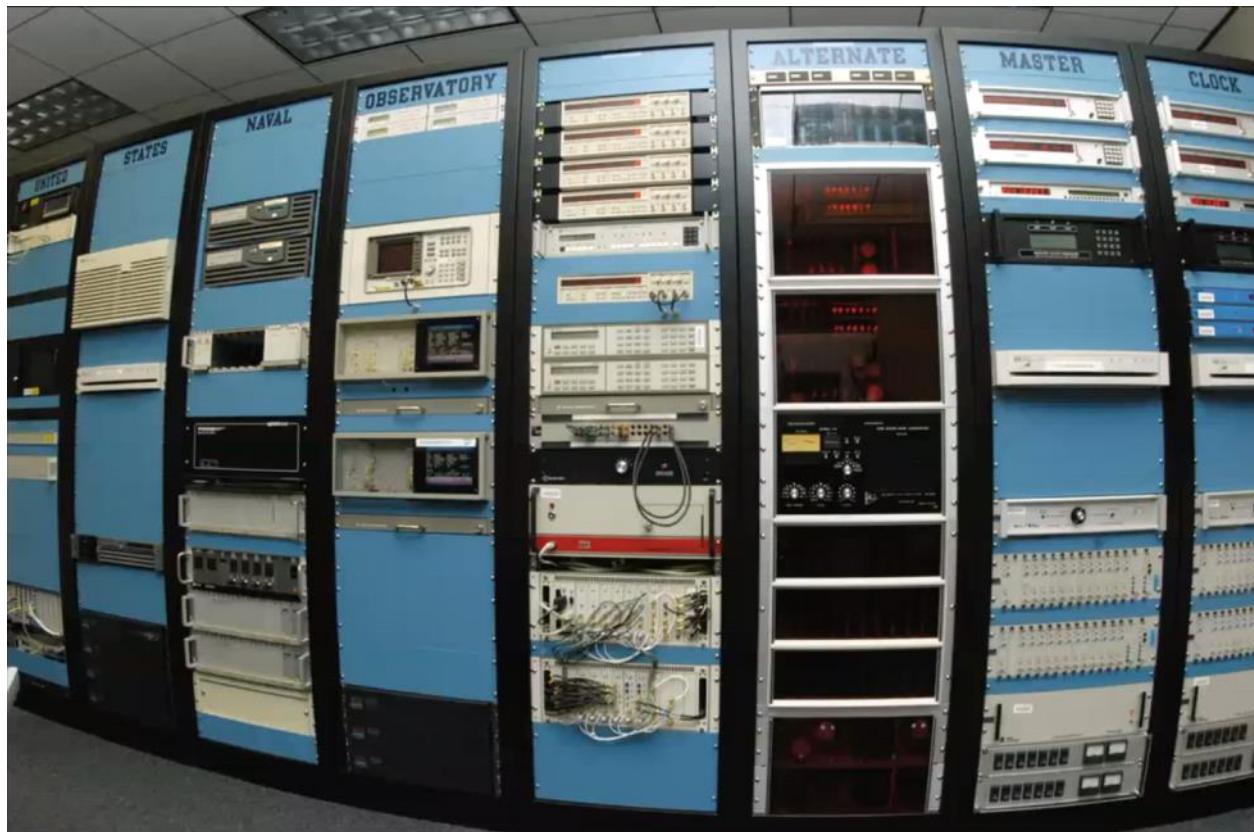
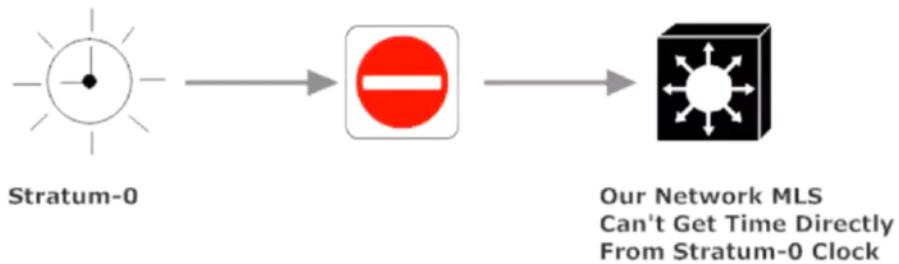
MLS_1(config)#snmp-server host 10.1.1.3 traps version 3 priv CHRIS ?
```

The Network Time Protocol

It's vital for our routers and switches to have a central time source that allows our network devices to synchronize their clocks. Doing so allows our syslog timestamps to have accurate and synched time throughout the network, making troubleshooting a lot less frustrating. Synced time is critical for digital certificate operation as well. If your certificate is good from 2015 – 2017 and your device thinks it's 2010, there's a problem!

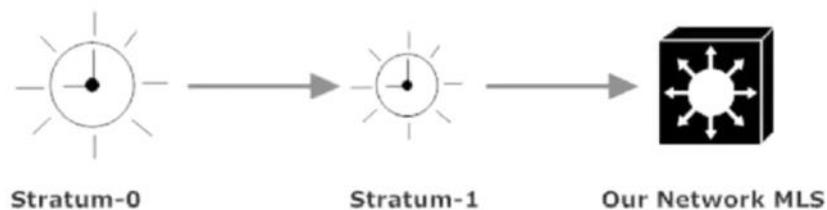
NTP allows us to specify time sources for our switches and routers, whether that time source be another router in the same network or an external time source.

At the very top of our NTP hierarchy are stratum-0 devices, typically atomic clocks. You can't configure a Cisco router to get its time directly from a stratum-0 server.



US Naval Observatory Alternate Time Clock

Stratum-1 servers are generally referred to as *time servers*, and we can configure a Cisco router to get its time from a stratum-1 device.



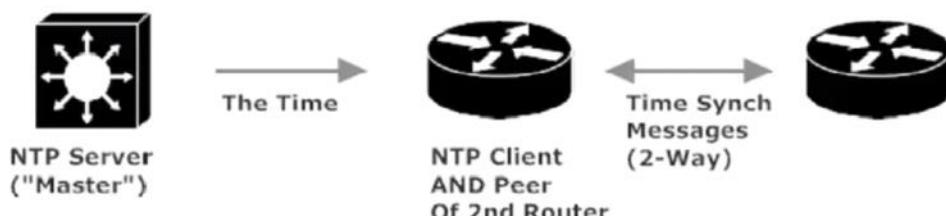
It's strongly recommended that your network's "outside" router receive its time from a public NTP timeserver. For the latest IP addresses of these servers, just run a search on the term *public NTP servers*. Be sure not to block UDP port 123 on that or other routers in your network – that's the port NTP uses.

Cisco routers can serve as NTP servers, clients, or peers. They can also depend on NTP broadcasts for the correct time. The NTP server-client relationship is as you'd expect, with the server giving the correct time to clients.



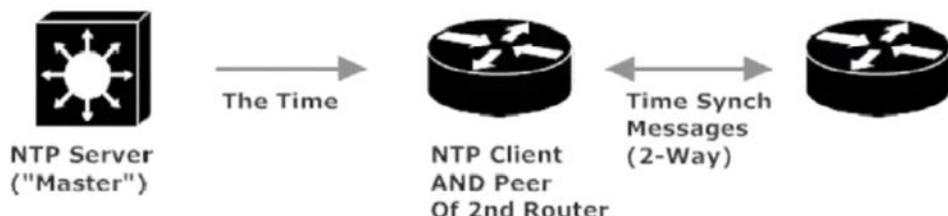
Clients accept the time synch message from the server and set their internal clock accordingly. Clients do NOT send NTP time synch messages back to the server.

We're not limited to the traditional Server/Client relationship with NTP. *NTP peers* send NTP messages to each other, and either peer can send time synch messages to the other.



Clients accept the time synch message from the server and set their internal clock accordingly. Clients do NOT sent NTP time synch messages back to the server.

We're not limited to the traditional Server/Client relationship with NTP. *NTP peers* send NTP messages to each other, and either peer can send time synch messages to the other.



We can choose to run NTP in *broadcast mode* or *multicast mode*. With these methods, the server broadcasts or multicasts its NTP messages, which the clients must be able to receive – otherwise, we're wasting our time!

It's *highly recommended* an NTP public timeserver be used as your NTP Master time source. Should you choose to use one of your network routers as the NTP Master, it's imperative you use NTP authentication and/or ACLs to prevent routers from outside your network from attempting to synch with one of your routers.

```
R4#show ntp assoc
      address          ref clock      st  when  poll reach delay offset
  * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configur
R4#
R4#show clock
.08:40:05.951 UTC Fri Sep 4 2015
R4#
R4#conf t
Enter configuration commands,
R4(config)#ntp master ?
<1-15> Stratum number
<cr>
R4(config)#ntp master 4 ?
<cr>
R4(config)#ntp master 4
R4(config)#

```

Router 4 will be the master.

```
R3(config)#ntp server 10.1.1.4 ?
  burst      Send a burst when peer is reachable
  iburst     Send a burst when peer is unreachable
  key        Configure peer authentication key
  maxpoll    Maximum poll interval
  minpoll    Minimum poll interval
  prefer     Prefer this peer when possible
  source     Interface for source address
  version    Configure NTP version
<cr>
```

We want router 3 to get its time from router 4. You can use prefer to create a primary and failover.

You need to see synchronization. It may take a minute to synch. Up to six minutes.

```
R4#show
% Type "show ?" for a list of subcommands
R4#show ntp assoc
  address          ref clock          st  when  poll  reach  delay  offset  o
*~127.127.1.1    .LOCL.            3    8    16   377  0.000  0.000  0.
 * sys.peer. # selected. + candidate. - outlyer. x falseticker. ~ configu
Notice the ref clock is local.
```

```
R3#show ntp status
Clock is synchronized, stratum 5, reference is 10.1.1.4
nominal freq is 250.0000 Hz, actual freq is 250.0017 Hz, precision is 2***24
reference time is D993DBBF.E7C73C22 (08:58:39.905 UTC Fri Sep 4 2015)
clock offset is -60.1169 msec, root delay is 1.91 msec
root dispersion is 1004.57 msec, peer dispersion is 189.73 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000006859
system poll interval is 64, last update was 178 sec ago.
R3# udemy
```

Has higher stratum therefore is a client to the server on router 4.

```
R3#show ntp assoc
  address          ref clock          st  when  poll  reach  delay  offset  o
*~10.1.1.4        127.127.1.1      4    17    64   177  1.912 -60.116 65.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configu
R3# udemy
```

Hardcode is like BGP (best,valid) symbols Systm peer and configured

Security 10 : NTP Server/Client Lab

Monday, April 23, 2018 5:09 PM

```
R3#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R3(config)#ntp authenticate
```

Enable NTP Authentication

```
R3#conf t
Enter configuration commands, one per line.
R3(config)#ntp authenticate
R3(config)#
R3(config)#ntp authentication-key ?
<1-4294967295> Key number

R3(config)#ntp authentication-key 1 ?
md5 MD5 authentication

R3(config)#ntp authentication-key 1 md5 ?
WORD Authentication key

R3(config)#ntp authentication-key 1 md5 CCNP
```

Create authentication key

```
R3(config)#ntp trusted-key ?  
    <1-4294967295>  Key number  
  
R3(config)#ntp trusted-key 1  
R3(config)#{
```

```
R3(config)#ntp server 10.1.1.4 key ?  
    <0-4294967295>  Peer key number  
  
R3(config)#ntp server 10.1.1.4 key 1  
R3(config)#^Z  
R3#wr
```

Create key number

Apply to client.

```

R3#show ntp assoc
  address          ref clock      st  when   poll reach delay offset d
*~10.1.1.4        127.127.1.1    4    9     64   37  1.934 -44.765 440
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configure
R3#
R3#
R3#show ntp status
Clock is synchronized, stratum 5, reference is 10.1.1.4
nominal freq is 250.0000 Hz, actual freq is 250.0020 Hz, precision is 2**24
reference time is D993E5A2.C89240EF (09:40:50.783 UTC Fri Sep 4 2015)
clock offset is -44.7658 msec, root delay is 1.93 msec
root dispersion is 985.98 msec, peer dispersion is 440.06 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000007993
system poll interval is 64, last update was 77 sec ago.
R3#

```

```

R3#show ntp assoc detail
10.1.1.4 configured, authenticated, our_master, sane, valid, stratum 4
ref ID 127.127.1.1 , time D993E613.0B366529 (09:42:43.043 UTC Fri Sep 4 2015)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.48, reach 77, sync dist 194.74
delay 1.93 msec, offset -44.7658 msec, dispersion 191.00
precision 2**24, version 4
org time D993E623.BB440FC6 (09:42:59.731 UTC Fri Sep 4 2015)
rec time D993E623.C6F5322F (09:42:59.777 UTC Fri Sep 4 2015)
xmt time D993E623.C66AFD56 (09:42:59.775 UTC Fri Sep 4 2015)
filtdelay = 1.96 1.94 1.93 1.95 1.93 1.93 0.00 0.00
filtoffset = -44.68 -45.08 -45.61 -45.15 -44.76 -44.50 0.00 0.00
filterror = 0.00 0.97 1.93 2.88 3.82 4.81 16000.0 16000.0
minpoll = 6, maxpoll = 10

```

4 is still master and we verify the NTP authentication.

```

R4#debug ntp packet
NTP packets debugging is on
R4#
Sep 4 09:45:08.723: NTP message received from 10.1.1.3 on interface 'FastEthernet0/0' (10.1.1.4).
Sep 4 09:45:08.723: NTP message sent to 10.1.1.3, from interface 'FastEthernet0/0' (10.1.1.4).
R4#
Sep 4 09:45:26.779: NTP message received from 10.1.1.2 on interface 'FastEthernet0/0' (10.1.1.4).
Sep 4 09:45:26.779: NTP message sent to 10.1.1.2, from interface 'FastEthernet0/0' (10.1.1.4).
R4#

```

```

R4#u all
All possible debugging has been turned off
R4#

```

Build an NTP Access List.

```
R4(config)#access-list 33 permit host 10.1.1.3
R4(config)#
R4(config)#ntp ?
 access-group          Control NTP access
 authenticate          Authenticate time sources
 authentication-key    Authentication key for trusted time sources
 broadcastdelay        Estimated round-trip delay
 clock-period          Length of hardware clock tick
 logging               Enable NTP message logging
 master                Act as NTP master clock
 max-associations     Set maximum number of associations
 peer                 Configure NTP peer
 server               Configure NTP server
 source               Configure interface for source address
 trusted-key          Key numbers for trusted time sources
 update-calendar      Periodically update calendar with NTP time
```

```
R4(config)#ntp access-group ?
 peer                 Provide full access
 query-only           Allow only control queries
 serve                Provide server and query access
 serve-only            Provide only server access
```

```
R4(config)#ntp access-group ?
 peer                 Provide full access
 query-only           Allow only control queries
 serve                Provide server and query access
 serve-only            Provide only server access
```

```
R4(config)#ntp access-group serve ?
 <1-99>               Standard IP access list
 <1300-1999>          Standard IP access list (expanded range)
 WORD                 Named access list
```

```
R4(config)#ntp access-group serve 33
R4(config)#
R4(config)#
R4(config)#^Z
```

Security 11: Unicast Reverse Path Forwarding

Wednesday, April 25, 2018 6:33 PM

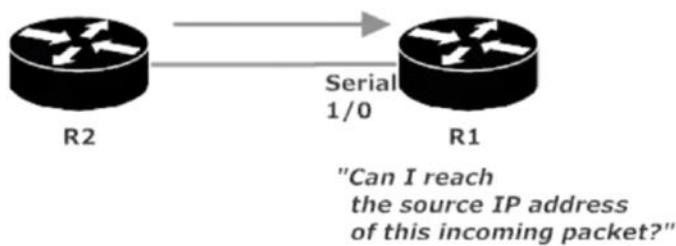
Man, that name has *everything*. We got unicasts, we got something going in reverse, and then we got forwarding! This should be something!

Actually, it is something, and something you shouldn't get confused with plain old *reverse path forwarding* (RPF). RPF is a multicasting feature, and Unicast RPF is, well, a unicast feature. Unicast RPF enables a router to verify an incoming packet by ensuring its source IP address is reachable.

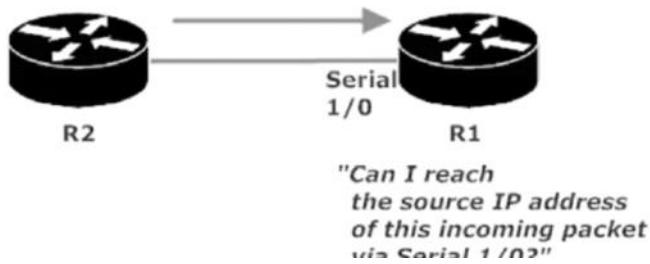
Unicast RPF will use its FIB to perform this check, so you know what that means...

```
R1(config)#ip cef
```

Cisco Express Forwarding must be up and running before you even get Unicast RPF started! After that, your choice is between *loose* and *strict* mode. With loose mode, the router will only check to be sure the source IP address of the incoming packet is reachable.



With strict mode, the verification is much tighter. The router must consider the source IP to be reachable by the same interface the packet rode in on.



For either mode, the command is *ip verify unicast source reachable-via*. Follow that with *any* for loose mode and *rx* for strict mode. There was an *ip verify unicast reverse-path* command that's still present on many routers, but as IOS Help notes, that's the old command format.

```
R1(config)#int serial 1/0
R1(config-if)#ip verify unicast ?
    reverse-path  Reverse path validation of source address (old command format)
    source        Validation of source address

R1(config-if)#ip verify unicast source ?
```

```

reachable-via Specify reachability check to apply to the source address

R1(config-if)#ip verify unicast source reachable-via ?
any Source is reachable via any interface
rx Source is reachable via interface on which packet was received

```

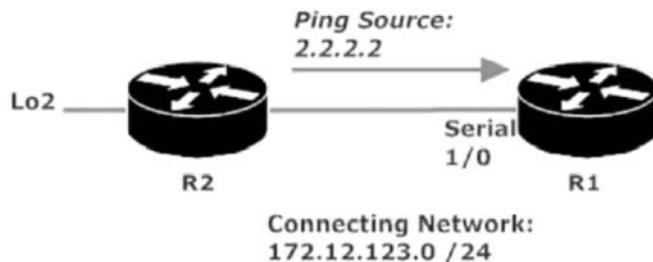
Before we run a lab with this feature, check out *these* interesting options, particularly the bottom two.

```

R1(config-if)#ip verify unicast source reachable-via rx ?
<1-199> A standard IP access list number
<1300-2699> A standard IP expanded access list number
allow-default Allow default route to match when checking source address
allow-self-ping Allow router to ping itself (opens vulnerability in
verification)

```

The *allow-default* option exists because the default behavior of Unicast RPF is to drop packets that could meet verification only through the use of a default route. The *allow-self-ping* option means just what it says, but when IOS Help says "opens vulnerability in verification", I'd think thrice about using it.



R2 has a loopback interface with the IP address 2.2.2.2, and we'll source the pings from that address to 172.12.123.1, R1's Serial 1/0 interface. Right now, R1 has no entry in its FIB for 2.2.2.2, but it *does* have a default route using 172.12.123.2 as the next-hop address.

```

R1#show ip cef
Prefix          Next Hop           Interface
0.0.0.0/0       172.12.123.2      Serial1/0
0.0.0.0/8       drop
0.0.0.0/32      receive
127.0.0.0/8     drop
(output truncated)

```

No trouble pinging 172.12.123.1 from 2.2.2.2... for now.

```
R2#ping 172.12.123.1 source 2.2.2.2
```

No default route.

```
R1#show ip cef
Prefix          Next Hop           Interface
0.0.0.0/0       drop               Null0 (default route handler entry)
0.0.0.0/8        drop
0.0.0.0/32      receive
127.0.0.0/8     drop
172.12.123.0/24 attached           Serial1/0
172.12.123.0/32 receive
172.12.123.1/32 receive
172.12.123.2/32 172.12.123.2    Serial1/0
172.12.123.3/32 172.12.123.3    Serial1/0
172.12.123.255/32 receive
224.0.0.0/4      drop
224.0.0.0/24     receive
240.0.0.0/4      drop
255.255.255.255/32 receive
```

Lets put a default route

```
R1(config)#ip route 0.0.0.0 0.0.0.0 172.12.123.2
R1(config)#^Z
R1#show ip cef
```

Now check

```
R1#show ip cef
*Aug 20 05:35:50.023: %SYS-5-CONFIG_I: Configured
R1#show ip cef
Prefix          Next Hop           Interface
0.0.0.0/0       172.12.123.2    Serial1/0
0.0.0.0/8        drop
0.0.0.0/32      receive
127.0.0.0/8     drop
172.12.123.0/24 attached           Serial1/0
172.12.123.0/32 receive
172.12.123.1/32 receive
172.12.123.2/32 172.12.123.2    Serial1/0
172.12.123.3/32 172.12.123.3    Serial1/0
172.12.123.255/32 receive
224.0.0.0/4      drop
224.0.0.0/24     receive
240.0.0.0/4      drop
255.255.255.255/32 receive
```

Now ping and specify the source on router 2 to 172.123

This is what happens if the address isn't on an up interface.

```
R2#ping 172.12.123.1 source 2.2.2.2
% Invalid source address- IP address not on any of our up interfaces
R2#
```

Lets set one.

```
R2# Configuration commands, one per line. End with Ctrl-Z.  
R2(config)#int loopback2  
R2(config-if)#ip address 2.2.2.2 25.  
*Sep 7 14:10:47.711: %LINK-3-UPDOWN: Interface Loopback2, changed state to  
*Sep 7 14:10:48.711: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopb  
changed state to up  
R2(config-if)#ip address 2.2.2.2 255.255.255.255  
R2(config-if)#^Z  
R2#
```

Ping will now go through.

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.123.1, timeout is 2  
Packet sent with a source address of 2.2.2.2  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max =  
R2#
```

Lets change verification.

```
R1(config-if)#ip verify unicast source r any
```

Verify. By default unicast rpf doesn't allow default address to match.

```
R1#show cef int serial 1/0  
Serial1/0 is up (if_number 6)  
  Corresponding hwidb fast_if_number 6  
  Corresponding hwidb firstsw->if_number 6  
  Internet address is 172.12.123.1/24  
  ICMP redirects are always sent  
  Per packet load-sharing is disabled  
  IP unicast RPF check is enabled  
  Inbound access list is not set  
  Outbound access list is not set  
  Hardware idb is Serial1/0  
  Fast switching type 5, interface type 71  
  IP CEF switching enabled  
  IP CEF Feature Fast switching turbo vector  
  Input fast flags 0x4000, Input fast flags2 0x0,  
fast flags2 0x0  
  ifindex 4(4)  
  Slot 1 slot unit 0 Unit 0 VC -1  
  Transmit limit accumulator 0x0 (0x0)  
  TP MTU 1500
```

Now the ping will fail.

```
R2#ping 172.12.123.1 source 2.2.2.2  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.123.1,  
Packet sent with a source address of 2.2.2.2  
....  
Success rate is 0 percent (0/5)  
R2#  
DRAFT ADV 1#
```

Verify

```
R1#show ip traffic
IP statistics:
Rcvd: 10 total, 5 local destination
      0 format errors, 0 checksum errors, 0 bad h
      0 unknown protocol, 0 not a gateway
      0 security failures, 0 bad options, 0 with
Opts: 0 end, 0 nop, 0 basic security, 0 loose sou
      0 timestamp, 0 extended security, 0 record
      0 stream ID, 0 strict source route, 0 alert
      0 other
Frags: 0 reassembled, 0 timeouts, 0 couldn't reass
      0 fragmented, 0 fragments, 0 couldn't fragm
Bcast: 0 received, 0 sent
Mcast: 0 received, 0 sent
Sent: 5 generated, 0 forwarded
Drop: 0 encapsulation failed, 0 unresolved, 0 no
      0 no route, 5 unicast RPF ↴ 0 forced drop
      0 options denied
```

Security 12 : Helper Address

Wednesday, April 25, 2018 6:47 PM

This n' That: The IP Helper Address Command

Cisco routers can't forward broadcasts, but this command enables a router to take an incoming UDP broadcast and forward it in unicast fashion to the address specified in the command. The *ip helper-address* command must be configured on the interface receiving the broadcasts that need forwarding.



```
R1(config)#int fast 0/0
R1(config-if)#ip helper-address 172.12.123.2
```

```
R1#show ip int fast 0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is 172.12.123.2
```

This command is often used to allow a network segment with no DHCP server to successfully obtain IP addressing from a DHCP server on another segment. By default, this command actually forwards eight UDP broadcasts by default, and 'tis a good idea to know them all.

TIME 37

TACACS 49

DNS 53

BOOTP (DHCP Server) 67

BOOTP (DHCP Client) 68

TFTP 69

NetBIOS Name Service 137

NetBIOS Datagram Service 138

If the UDP broadcast that needs help isn't in that list, add it with the *ip forward-protocol udp* command. To remove a broadcast type from the list – you guessed it! -- run *no ip forward-protocol*. This one's a global command, not an interface-level command. IOS Help will show you some of the more common UDP port numbers, but this is not an all-inclusive list.

```
R1(config)#ip forward-protocol ?
  nd                  Sun's Network Disk protocol
  sdns                Network Security Protocol
  spanning-tree        Use transparent bridging to flood UDP broadcasts
  turbo-flood          Fast flooding of UDP broadcasts
  udp                 Packets to a specific UDP port
```

```
R1(config)#ip forward-protocol udp ?
<0-65535>      Port number
biff              Biff (mail notification, comsat, 512)
bootpc            Bootstrap Protocol (BOOTP) client (68)
bootps            Bootstrap Protocol (BOOTP) server (67)
discard           Discard (9)
dnsix             DNSIX security protocol auditing (195)
domain            Domain Name Service (DNS, 53)
echo               Echo (7)
isakmp             Internet Security Association and Key Man
(500)
mobile-ip          Mobile IP registration (434)
nameserver         IEN116 name service (obsolete, 42)
netbios-dgm        NetBios datagram service (138)
netbios-ns          NetBios name service (137)
netbios-ss          NetBios session service (139)
non500-isakmp     Internet Security Association and Key Man
(4500)
ntp                Network Time Protocol (123)
pim-auto-rp        PIM Auto-RP (496)
rip                Routing Information Protocol (router, in
Simple Network Management Protocol (161)
snmp               SNMP Traps (162)
snmptrap
More
```

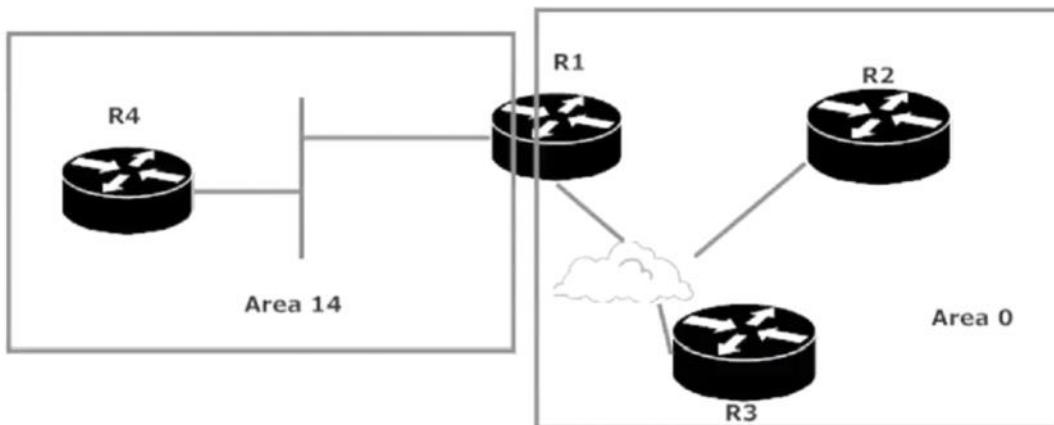
4/25/2018 6:52 PM - Screen Clipping

Security 13 : Using and Filtering "Debug IP Packet"

Wednesday, April 25, 2018 6:52 PM

Giving "debug ip packet" A Little SUJ (Shut-Up Juice)

The *debug ip packet* command is a great troubleshooting tool, but it gives you so much information that it can be hard to spot the info you need. On top of that, running this command in a production network (and some larger lab environments) can overwhelm the router to the point where the router cannot do its job. Here's the output of *show ip route* on R1, which at this point has a few OSPF adjacencies but isn't sending or receiving production traffic.



```
R1#debug ip packet
IP packet debugging is on
R1#
*Aug 20 08:01:11: IP: s=172.12.123.3 (Serial1/0), d=172.12.123.1, len 80, r
R1#
*Aug 20 08:01:13: IP: s=172.12.123.1 (local), d=172.12.123.2 (Serial1/0), 1
, sending
*Aug 20 08:01:13: IP: s=172.12.123.1 (local), d=172.12.123.3 (Serial1/0), 1
, sending
R1#
*Aug 20 08:01:15: IP: s=172.12.123.2 (Serial1/0), d=172.12.123.1, len 80, r
R1#
*Aug 20 08:01:16: IP: s=172.12.123.3 (Serial1/0), d=172.12.123.1, len 80, r
```

```
R1#u all
All possible debugging has been turned off
R1#
*Aug 20 08:01:28: IP: s=10.1.1.4 (FastEthernet0/0), d=224.0.0.5, len 80, rc
*Aug 20 08:01:28: IP: s=10.1.1.1 (local), d=224.0.0.5 (FastEthernet0/0), le
, sending broad/multicast
*Aug 20 08:01:28: IP: s=172.12.123.1 (local), d=172.12.123.2 (Serial1/0), 1
, sending
*Aug 20 08:01:28: IP: s=172.12.123.1 (local), d=172.12.123.3 (Serial1/0), 1
, sending
```

Generate traffic

```
R1#debug ip packet
IP packet debugging is on
R1#
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.2
ial1/0), routed via RIB
*Aug 20 08:02:06: IP: s=172.12.123.2 (Serial1/0
n 100, rcvd 3
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.1
/0), routed via FIB
*Aug 20 08:02:06: IP: s=172.12.123.1 (local), d
0, sending
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.2
ial1/0), routed via RIB
*Aug 20 08:02:06: IP: s=172.12.123.2 (Serial1/0
n 100, rcvd 3
R1#
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.1
/0), routed via FIB
*Aug 20 08:02:06: IP: s=172.12.123.1 (local), d
0, sending
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.2
ial1/0), routed via RIB
*Aug 20 08:02:06: IP: s=172.12.123.2 (Serial1/0
n 100, rcvd 3
*Aug 20 08:02:06: IP: tableid=0, s=172.12.123.1
```

Write an Acl that identifies the traffic from the debug IP command that matches the ACL only. So traffic only destined for router 4 or sourced from router 4.

10.1.1.4

```
R1#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.12.123.2	0	FULL/DROTHER	00:00:16	172.12.123.2	Serial1/0
172.12.123.3	0	FULL/DROTHER	00:00:17	172.12.123.3	Serial1/0
10.1.1.4	1	FULL/DR	00:00:30	10.1.1.4	FastEther

You need an ACL with two lines (extended ACL) because we need one line to match the source and a second that matches destination.

```
R1(config)#access-list 124 permit ip host 10.1.1.4 any
R1(config)#access-list 124 permit ip any host 10.1.1.4
```

So how do we apply this ACL to a debug? This way it will only ID traffic for ACL 124.

```
R1#debug ip packet ?
<1-199>      Access list
<1300-2699>  Access list (expanded range)
detail        Print more debugging detail
<cr>
```

```
R1#debug ip packet 124
IP packet debugging is on for access list 124
```

Ping from router 2.

```
R2#ping ip 172.12.123.1 repeat 1000
Type escape sequence to abort.
Sending 1000, 100-byte ICMP Echos to 172.12.123.1
!!!!!!!!!!!!!!
```

Now check router 1.

```
*Aug 20 08:08:45: IP: tableid=0, s=10.1.1.4 (FastEthernet0/0), d=10.1.1.1 (FastEthernet0/0), routed via RIB
*Aug 20 08:08:45: IP: s=10.1.1.4 (FastEthernet0/0), d=10.1.1.1 (FastEthernet0/0), len 100, rcvd 3
*Aug 20 08:08:45: IP: tableid=0, s=10.1.1.1 (local), d=10.1.1.4 (FastEthernet0/0), routed via FIB
*Aug 20 08:08:45: IP: s=10.1.1.1 (local), d=10.1.1.4 (FastEthernet0/0), sending
```

Security 14 : Spotting Memory Issues and Core Dumps!

Wednesday, April 25, 2018 7:06 PM

Spotting Memory Issues (Router Memory, That Is)

I often mention being careful about the load on your router's memory, but how do you know when you're starting to overload your router in that department? Cisco's website lists several signs that you just might have a problem:

If your router rejects Telnet sessions... ~~you might be a redneck~~ you might have a memory problem.

If your router shows you the output of *show processor memory* regardless of the command you're actually entering... that, my friend, is a cry for help.

If your router literally tells you "low on memory" or "Unable to create EXEC – no memory or too many processes" ... you DO have a memory problem.

Other possible signs of memory issues include your *show* commands not showing you anything when you know darn well there is something to show you, and your router just hanging when you try to connect via the console connection.

If you can't connect via the console port, Cisco recommends you disconnect both the WAN and LAN cables connected to the router (ouch!) and then try to get in. Since the router won't be processing packets, there's an excellent chance you'll be able to connect successfully. Cisco recommends you then run *show memory allocating-*

process totals and *show logging*. If your router doesn't support *show memory allocating-process totals*, run *show memory summary*. Many routers support both.

R1#show memory ?	
allocating-process	Show allocating process name
dead	Memory owned by dead processes
debug	Memory debugging commands
failures	Memory failures
fast	Fast memory stats
fragment	Summary of memory fragment information
free	Free memory stats
io	IO memory stats
multibus	Multibus memory stats
overflow	memory overflow corrections
pci	PCI memory stats
processor	Processor memory stats
statistics	Mempool Statistics
summary	Summary of memory usage per alloc PC
transient	Transient memory stats

R1#show memory allocating-process totals						
	Head	Total(b)	Used(b)	Free(b)	Lowest(b)	Largest
Processor	84A738A0	31419072	14150280	17268792	15689128	1567
I/O	6700000	26214400	2127016	24087384	24087384	2408

Allocator PC Summary for: Processor

PC	Total	Count	Name
0x802D6124	1719452	201	Process Stack
0x822CB508	1469296	9	pak subblock chunk
0x821C9C14	526324	48	TCL Chunks
0x8055CCAC	456416	507	*Packet Header*
0x800E43E8	440680	6	MallocLite
0x822CA730	351768	30	TW Buckets
0x826E6BE4	208144	1	epa crypto blk
0x822A9BA0	152036	199	Process
0x822B1A40	148996	716	*Init*
0x814F6C20	133212	4	CEF: 1 path chunk pool
0x815099DC	131124	1	Init
0x8055CCFC	121200	150	*Packet Data*
0x82BB4A10	116380	16	IPv6 CEF fib tables
0x8231EB34	105692	3	CCH323_BE_STATIC_DESCRIPTOR_HASH
0x81640914	98356	1	Init
0x821C9A60	97384	17	TCL Chunks
0x80BB8E08	88736	1163	Init
0x821C9BC0	83164	12	TclCreateExecEnv
0x8187C594	82564	1	Init

Making A Core Dump

Defining a core dump is simple enough; it's a copy of the router's memory contents. Simple enough, indeed!

Generating a core dump is another matter. Here's what Cisco has to say about creating core dumps:

"CAUTION: Core dumps are not necessary to solve most crash cases. Creation of a core dump when the router is functioning in a network can disrupt network operation. Use...only under the direction of a technical support representative."

A core dump can be created via FTP, TFTP, RCP (Remote Copy Protocol), or via a flash disk. Obviously, the procedures are different depending on which method you use, but each will use the *exception core-file* and *exception region-size* commands.

```
R1(config)#exception ?
  core-file           Set name of core dump file
  crashinfo          Crashinfo collection
  data-corruption    Data error exception handling
  delay-dump         Pause dump (in the case of dump via peer)
  dump               Set name of host to dump to
  flash              Set the device and erase permission
  memory             Memory leak debugging
```

```
R1(config)#exception core-file ?  
WORD Name of the core file
```

Use *exception core-file* to change the default name given to the core dump, which is "R1-core" in this case. The hostname always begins the core dump filename by default, followed by "-core".

```
R1(config)#exception region-size ?  
<1024-65536> Region size
```

Speaking of memory (and we just were!), *exception region-size* reserves just a bit of memory to be used by the core dump in case the larger memory pool is corrupted. Otherwise, you have a good chance of a memory issue arising during the core dump creation, and that's the last thing you need. The default is 16384 bytes, and the larger this value, the better the chances of having an issue-free core dump.

You can test with *write core*, which creates a core dump and saves the file to the location specified.

```
R4#write core 10.1.1.1  
Base name of core files to write [R4-core]?
```

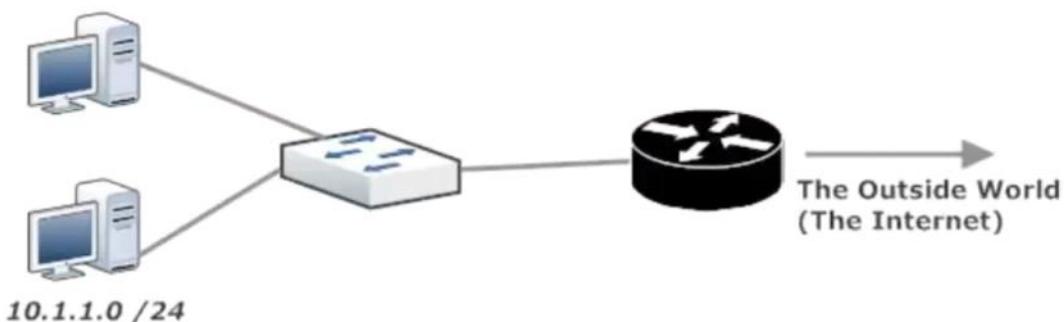
4/25/2018 7:14 PM - Screen Clipping

NAT 1: Static NAT

Wednesday, April 25, 2018 7:15 PM

NAT and PAT

Network Address Translation takes a host's private IP address and translates it to a non-private, routable address. A simple but important job! Without NAT, a host such as this one on the 10.1.1.0 /24 network couldn't communicate with the outside world.



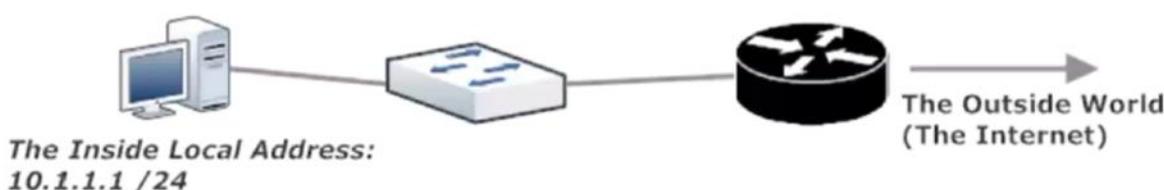
The addresses we'll be translating are from the RFC 1918 range of private addresses. Note the masks for these address ranges are not the same as those for the full Class A (/8), Class B (/16), and Class C (/24) address ranges.

Class A: 10.0.0.0 /8

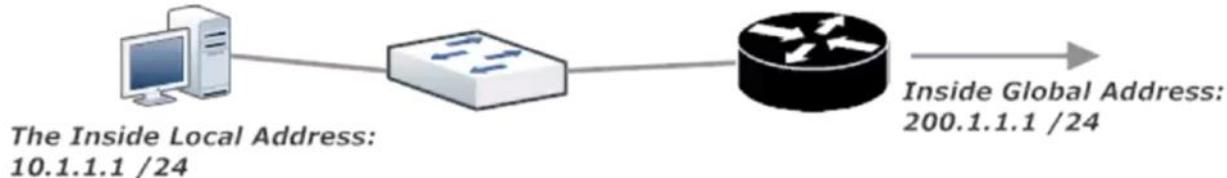
Class B: 172.16.0.0 /12

Class C: 192.168.0.0 /16

The only thing that's even the *slightest* bit tricky about NAT are the names given to the addresses in the overall NAT process. We start with an *inside local address*, the address used by hosts on the local network to communicate with other hosts on the local network. The inside local address is the address being translated – locally. In this network, the inside local address is 10.1.1.1 /24.



The inside local address is translated to an *inside global address*. In this case, that'll be 200.1.1.1 /24.



Outside local addresses are the non-routable addresses of hosts on the remote network, while *outside global addresses* are the routable addresses assigned to hosts on a remote network. The terms “inside” and “outside” really depend on your perspective. If the address is in use on your network (global or local), it’s an inside address. If it’s in use by the other involved network, it’s a global address.

"This address, 10.1.1.1 /24, is an inside local address."



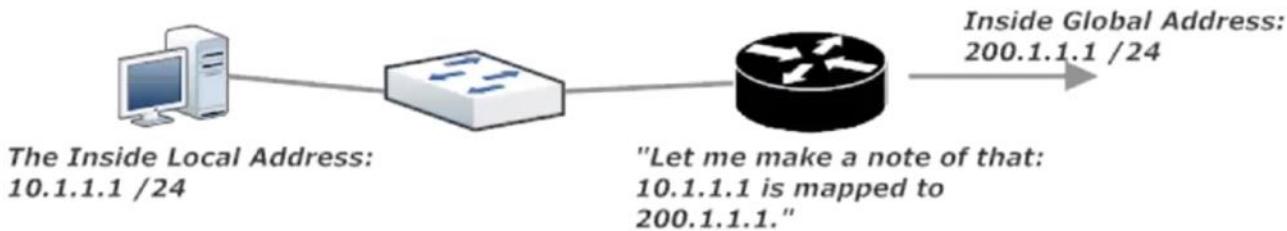
"That address, 10.1.1.1 /24, outside local address."



When a router performs NAT, that router makes an entry in its NAT translation table, mapping the inside local address to the assigned inside global address.

ITS ALL ABOUT PERSPECTIVE

When a router performs NAT, that router makes an entry in its NAT translation table, mapping the inside local address to the assigned inside global address.



The private address is never seen outside the local network, and the host receiving these packets has no idea NAT has occurred. Actually, the host sending the packets doesn't know about NAT either! The only device that even knows this is going on is the NAT router.

When packets come back in with a routable address, the router checks its NAT table to see if another translation is in order. If so, the router translates the inside global address back to the appropriate inside local address and routes the packets accordingly.

NAT 2: Dynamic NAT

Wednesday, April 25, 2018 7:15 PM

NAT 3 : Port Address Translation

Wednesday, April 25, 2018 7:15 PM

Ipsec over GRE

Saturday, May 4, 2019 11:32 PM

You want to configure a GRE over IPSec tunnel between RouterA and RouterC. You should complete the following tasks:

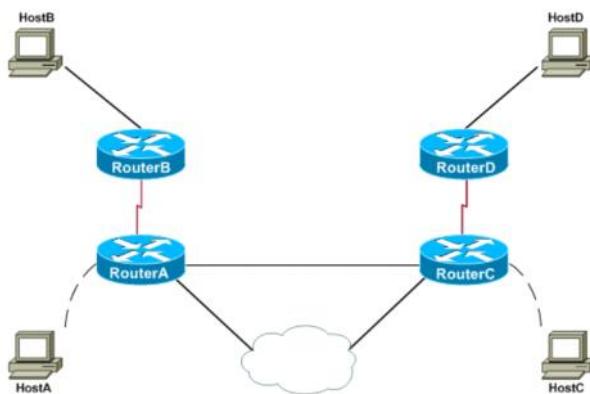
- Use tunnel interface Tunnel 0.
- Assign the tunnel interface on RouterA an IP address of 192.168.100.1/30.
- Assign the tunnel interface on RouterC an IP address of 192.168.100.2/30.
- Name the transform set boson. Specify the encryption transform before the authentication transform.
- Name the crypto map exsim, and use sequence number 1.
- Use access list 101.
- Specify the tunnel source and destination by using the IP addresses in the topology.
- Create a static route that routes packets destined for 10.1.1.0/24 and 10.2.2.0/24 through the tunnel. Specify the next hop as an interface, not as an IP address.

Use the following IKE configuration parameters:

- Priority = 10
- ESP with 3DES encryption
- ESP with MD5 authentication
- DH group 2 key exchange
- PSK = abcd1234

You can click HostA and HostC to establish console cable connections to RouterA and RouterC, respectively.

5/4/2019 11:32 PM - Screen Clipping



5/4/2019 11:33 PM - Screen Clipping

Answer:

On RouterA:

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#crypto isakmp policy 10
RouterA(config-isakmp)#authentication pre-share
RouterA(config-isakmp)#hash md5
RouterA(config-isakmp)#encryption 3des
RouterA(config-isakmp)#group 2
RouterA(config-isakmp)#exit
RouterA(config)#crypto isakmp key abcd1234 address 192.168.1.2
RouterA(config)#crypto ipsec transform-set boson esp-3des esp-md5-hmac
RouterA(config)#crypto map exsim 1 ipsec-isakmp
RouterA(config-crypto-map)#set peer 192.168.1.2
RouterA(config-crypto-map)#set transform-set boson
RouterA(config-crypto-map)#match address 101
RouterA(config-crypto-map)#exit
RouterA(config)#access-list 101 permit gre host 192.168.1.1 host 192.168.1.2
RouterA(config)#interface fastethernet 0/0
RouterA(config-if)#crypto map exsim
RouterA(config-if)#exit
RouterA(config)#interface tunnel 0
RouterA(config-if)#ip address 192.168.100.1 255.255.255.252
RouterA(config-if)#tunnel source 192.168.1.2
RouterA(config-if)#exit
RouterA(config)#ip route 10.1.1.0 255.255.255.0 tunnel 0
```

5/4/2019 11:33 PM - Screen Clipping

```

On RouterC:
RouterC>enable
RouterC#configure terminal
RouterC(config)#crypto isakmp policy 10
RouterC(config-isakmp)#authentication pre-share
RouterC(config-isakmp)#hash md5
RouterC(config-isakmp)#encryption 3des
RouterC(config-isakmp)#group 2
RouterC(config-isakmp)#exit
RouterC(config)#crypto isakmp key abcd1234 address 192.168.1.1
RouterC(config)#crypto ipsec transform-set boson esp-3des esp-md5-hmac
RouterC(config)#crypto map exsim 1 ipsec-isakmp
RouterC(config-crypto-map)#set peer 192.168.1.1
RouterC(config-crypto-map)#set transform-set boson
RouterC(config-crypto-map)#match address 101
RouterC(config-crypto-map)#exit
RouterC(config)#access-list 101 permit gre host 192.168.1.2 host 192.168.1.1
RouterC(config)#interface fastethernet 0/0
RouterC(config-if)#crypto map exsim
RouterC(config-if)#exit
RouterC(config)#interface tunnel 0
RouterC(config-if)#ip address 192.168.100.2 255.255.255.252
RouterC(config-if)#tunnel source 192.168.1.2
RouterC(config-if)#tunnel destination 192.168.1.1
RouterC(config-if)#exit
RouterC(config)#ip route 10.1.1.0 255.255.255.0 tunnel 0

```

5/4/2019 11:33 PM - Screen Clipping

Explanation:

You must perform the following tasks to configure a Generic Routing Encapsulation (GRE) over IP Security (IPSec) tunnel between RouterA and RouterC:

- Configure Internet Key Exchange (IKE) peering.
- Configure an IPSec transform set.
- Configure an IPSec crypto map.
- Create an access list that permits the GRE traffic.
- Apply the IPSec crypto map to the physical interface.
- Configure a tunnel interface.
- Configure routing through the tunnel.

To access RouterA, you should click the connected host computer. Issue the **configure terminal** command to enter global configuration mode. To configure IKE peering, you must create an IKE policy. The scenario indicates that you should use priority 10 for the policy. Therefore, you should issue the **crypto isakmp policy 10** command. Issuing the **crypto isakmp policy 10** command will place the router into Internet Security Association and Key Management Protocol (ISAKMP) policy configuration mode, where you can specify several IKE configuration parameters. To configure the router to use a preshared key (PSK), you should issue the **authentication pre-share** command. To specify Triple Data Encryption Standard (3DES) encryption, you should issue the **encryption 3des** command. To specify Message Digest 5 (MD5), you should issue the **hash md5** command. Finally, to specify Diffie-Hellman (DH) group 2 key exchange, you should issue the **group 2** command. Issuing the **exit** command will return the router to global configuration mode.

To configure the PSK, you should issue the **crypto isakmp key abcd1234 address 192.168.1.2** command in global configuration mode. The syntax of the **crypto isakmp key** command is **crypto isakmp key key address peer-address**. The scenario indicates that the PSK is abcd1234, and the network topology indicates that the remote peer address is 192.168.1.2.

5/4/2019 11:34 PM - Screen Clipping

To configure an IPSec transform set named boson, you should issue the **crypto ipsec transform-set boson esp-3des esp-md5-hmac** command. The syntax of the **crypto ipsec transform-set** command is **crypto ipsec transform-set transform-name transform1 [transform2] [transform3] [transform4]**. Up to four transforms can be specified in an IPSec transform set: one Encapsulating Security Payload (ESP) authentication transform, one ESP encryption transform, one Authentication Header (AH) transform, and one IP compression transform.

To configure an IPSec crypto map named exsim with sequence number 1, you should issue the **crypto map exsim 1 ipsec-isakmp** command. Issuing this command will place the router into crypto map configuration mode where you must specify the peer address, the transform set, and the access list. To specify the peer address, you should issue the **set peer 192.168.1.2** command. To specify the transform set, you should issue the **set transform-set boson** command. To specify the access list, you should issue the **match address 101** command. Issuing the **exit** command will return the router to global configuration mode. To configure an access list that will allow the GRE traffic, you should issue the **access-list 101 permit gre host 192.168.1.1 host 192.168.1.2** command.

To apply the crypto map to the FastEthernet 0/1 interface, you should issue the **interface fastethernet 0/0** command to place the router into interface configuration mode for FastEthernet 0/0. Issuing the **crypto map exsim** command will apply the crypto map to the interface. Issuing the **exit** command will return the router to global configuration mode.

To create the Tunnel 0 interface, you should issue the **interface tunnel 0** command. Issuing this command will place the router into interface configuration mode for Tunnel 0. To configure Tunnel 0 to use IP address 192.168.100.1/30, you should issue the **ip address 192.168.100.1 255.255.255.252** command. To configure the tunnel source, you should issue the **tunnel source 192.168.1.1** command. To configure the tunnel destination, you should issue the **tunnel destination 192.168.1.2** command. Issuing the **exit** command will return the router to global configuration mode.

To create a static route that routes traffic destined for the 10.2.2.0/24 network through Tunnel 0, you should issue the **ip route 10.2.2.0 255.255.255.0 tunnel 0** command; had the scenario allowed it, you could have also issued the **ip route 10.2.2.0 255.255.255.0 192.168.100.1** command. You should then repeat these steps for RouterC by using the appropriate IP addresses for the ISAKMP key, the peer, the access list, the tunnel address, the tunnel source and destination, and the static route.

5/4/2019 11:34 PM - Screen Clipping

IPv6 Communications

Wednesday, May 8, 2019 3:31 PM

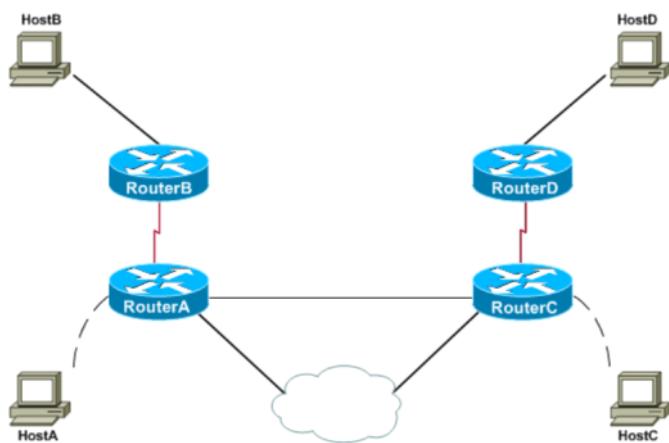
You want to configure IPv6 communication between RouterA and RouterC. You should complete the following tasks:

- Create a manual IPv6 over IPv4 tunnel between RouterA and RouterC.
- Configure the tunnel source and destination by using IP addresses.
- Configure the Tunnel 0 interface of RouterA with the IPv6 address 2001:1:1:1::1/64.
- Configure the Tunnel 0 interface of RouterC with the IPv6 address 2001:1:1:1::2/64.

After you complete these tasks, you should be able to ping the IPv6 address of RouterC from RouterA and you should be able to ping the IPv6 address of RouterA from RouterC.

You can click HostA and HostC to establish console cable connections to RouterA and RouterC, respectively. All passwords are set to cisco.

Screen clipping taken: 5/8/2019 3:31 PM



Screen clipping taken: 5/8/2019 3:32 PM

Partially Correct 62/74

Answer:

On RouterA:

```
RouterA>enable  
RouterA#configure terminal  
RouterA(config)#interface tunnel 0  
RouterA(config-if)#ipv6 address 2001:1:1:1::1/64  
RouterA(config-if)#tunnel source 10.1.1.1  
RouterA(config-if)#tunnel destination 10.1.1.2  
RouterA(config-if)#tunnel mode ipv6ip
```

On RouterC:

```
RouterC>enable  
RouterC#configure terminal  
RouterC(config)#interface tunnel 0  
RouterC(config-if)#ipv6 address 2001:1:1:1::2/64  
RouterC(config-if)#tunnel source 10.1.1.2  
RouterC(config-if)#tunnel destination 10.1.1.1  
RouterC(config-if)#tunnel mode ipv6ip
```

Screen clipping taken: 5/8/2019 3:32 PM

To access RouterA and RouterC, you should click the connected host computer. On each router, you should issue the **configure terminal** command to enter global configuration mode. You should then issue the **interface tunnel 0** command to create the tunnel interface that will be used to connect RouterA to RouterC. Then you should issue the **ipv6 address 2001:1:1:1::1/64** command on RouterA and the **ipv6 address 2001:1:1:1::2/64** command on RouterC to configure the Tunnel 0 interface on each router with the appropriate IP version 6 (IPv6) address.

On RouterA, you should then issue the **tunnel source 10.1.1.1** command to bind the virtual tunnel interface to the physical interface associated with the 10.1.1.1 IP address, which in this case is the FastEthernet 0/0 interface. Because a manual IPv6 over IPv4 tunnel requires a source address and a destination address, you should next issue the **tunnel destination 10.1.1.2** command to specify the external IP address of RouterC as the destination of the tunnel.

By default, a tunnel interface is created as a Generic Routing Encapsulation (GRE) tunnel. Although GRE is capable of tunneling IPv6 over IPv4, this scenario requires you to configure a manual IPv6 over IPv4 tunnel, not a GRE tunnel. Therefore, you will need to modify the tunnel mode. You should issue the **tunnel mode ipv6ip** command to configure manual IPv6 over IPv4 tunnel mode.

Similarly, on RouterC you should issue the **tunnel source 10.1.1.2** command to bind the virtual tunnel interface to the physical interface associated with the 10.1.1.2 IP address, which in this case is the FastEthernet 0/0 interface. Next, you should issue the **tunnel destination 10.1.1.1** command to specify the external IP address of RouterA as the destination of the tunnel. Finally, you should issue the **tunnel mode ipv6ip** command to configure manual IPv6 over IPv4 tunnel mode.

You can test the tunnel configuration by issuing the **ping ipv6 2001:1:1:1::2** command from privileged EXEC mode on RouterA or by issuing the **ping ipv6 2001:1:1:1::1** command from privileged EXEC mode on RouterC. If the ping is successful, RouterA and RouterC are configured correctly and are routing IPv6 packets properly.

Although you are not required to do so in this simulation, you should always save the configuration by issuing the **copy running-config startup-config** command on each router. If you make changes to the running configuration and do not save it to the startup configuration by issuing the **copy running-config startup-config** command, you will lose those changes if the router is restarted.

Screen clipping taken: 5/8/2019 3:32 PM

EIGRP Classic Config

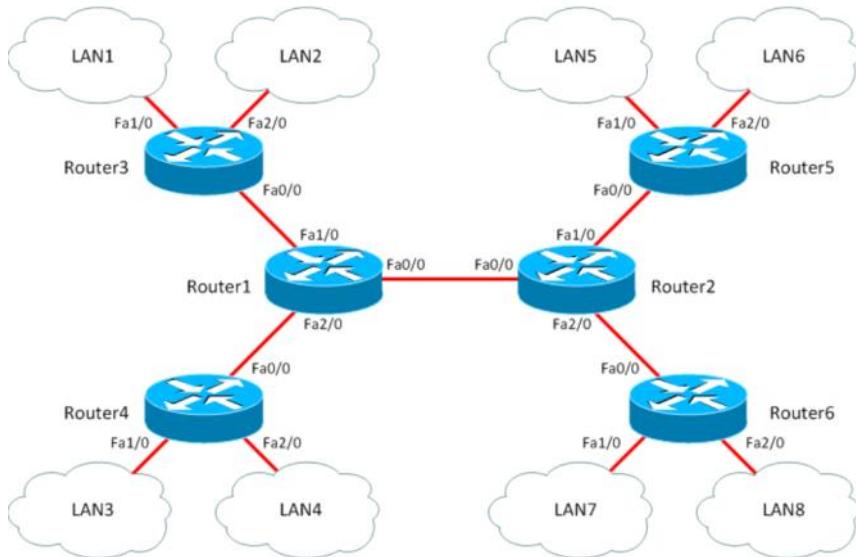
Wednesday, May 8, 2019 3:33 PM

You are a network administrator for the corporate headquarters at yourcorp.tv. You have been asked to connect four remote sites to your company's EIGRP configuration. Router1 and Router2 are both located at yourcorp headquarters and have already been configured by using an EIGRP named mode configuration. The administrators at the remote sites are not yet familiar with EIGRP named mode configurations. Therefore, you have been asked to deploy a classic configuration on the remote routers until the administrators at the remote sites can receive appropriate training on EIGRP named mode conversion and configuration.

Click on each router in the topology to access its console and complete the configuration. The completed configuration must meet the following specifications:

- Router3, Router4, Router5, and Router6 must use a classic EIGRP configuration
- All interfaces on Router3, Router4, Router5, and Router6 should operate in EIGRP AS 1.
- Any wildcard masks you configure should encompass as many networks as reasonably possible.
- On each router in the topology, you should configure manual summarization for as many networks as possible.

Screen clipping taken: 5/8/2019 3:33 PM



Screen clipping taken: 5/8/2019 3:33 PM

Partially Correct 4/32

Answer:

On Router1:

```
Router1>enable
Router1#configure terminal
Router1(config)#router eigrp yourcorp
Router1(config-router)#address-family ipv4 unicast autonomous-system 1
Router1(config-router-af)#af-interface fastethernet 0/0
Router1(config-router-af-interface)#summary-address 198.51.100.0 255.255.255.240
Router1(config-router-af-interface)#summary-address 192.168.1.0 255.255.255.192
```

On Router2:

```
Router2>enable
Router2#configure terminal
Router2(config)#router eigrp yourcorp
Router2(config-router)#address-family ipv4 unicast autonomous-system 1
Router2(config-router-af)#af-interface fastethernet 0/0
Router2(config-router-af-interface)#summary-address 198.51.100.16 255.255.255.248
Router2(config-router-af-interface)#summary-address 192.168.1.64 255.255.255.192
```

- - - -

Screen clipping taken: 5/8/2019 3:34 PM

```
On Router3:  
Router3>enable  
Router3#configure terminal  
Router3(config)#router eigrp 1  
Router3(config-router)#network 3.3.3.3 0.0.0.0  
Router3(config-router)#network 192.168.1.0 0.0.0.31  
Router3(config-router)#network 198.51.100.4 0.0.0.3  
Router3(config-router)#exit  
Router3(config)#interface fastethernet 0/0  
Router3(config-if)#ip summary-address eigrp 1 192.168.1.0 255.255.255.224
```

```
On Router4:  
Router4>enable  
Router4#configure terminal  
Router4(config)#router eigrp 1  
Router4(config-router)#network 4.4.4.4 0.0.0.0  
Router4(config-router)#network 192.168.1.32 0.0.0.31  
Router4(config-router)#network 198.51.100.8 0.0.0.3  
Router4(config-router)#exit  
Router4(config)#interface fastethernet 0/0  
Router4(config-if)#ip summary-address eigrp 1 192.168.1.32 255.255.255.224
```

Screen clipping taken: 5/8/2019 3:34 PM

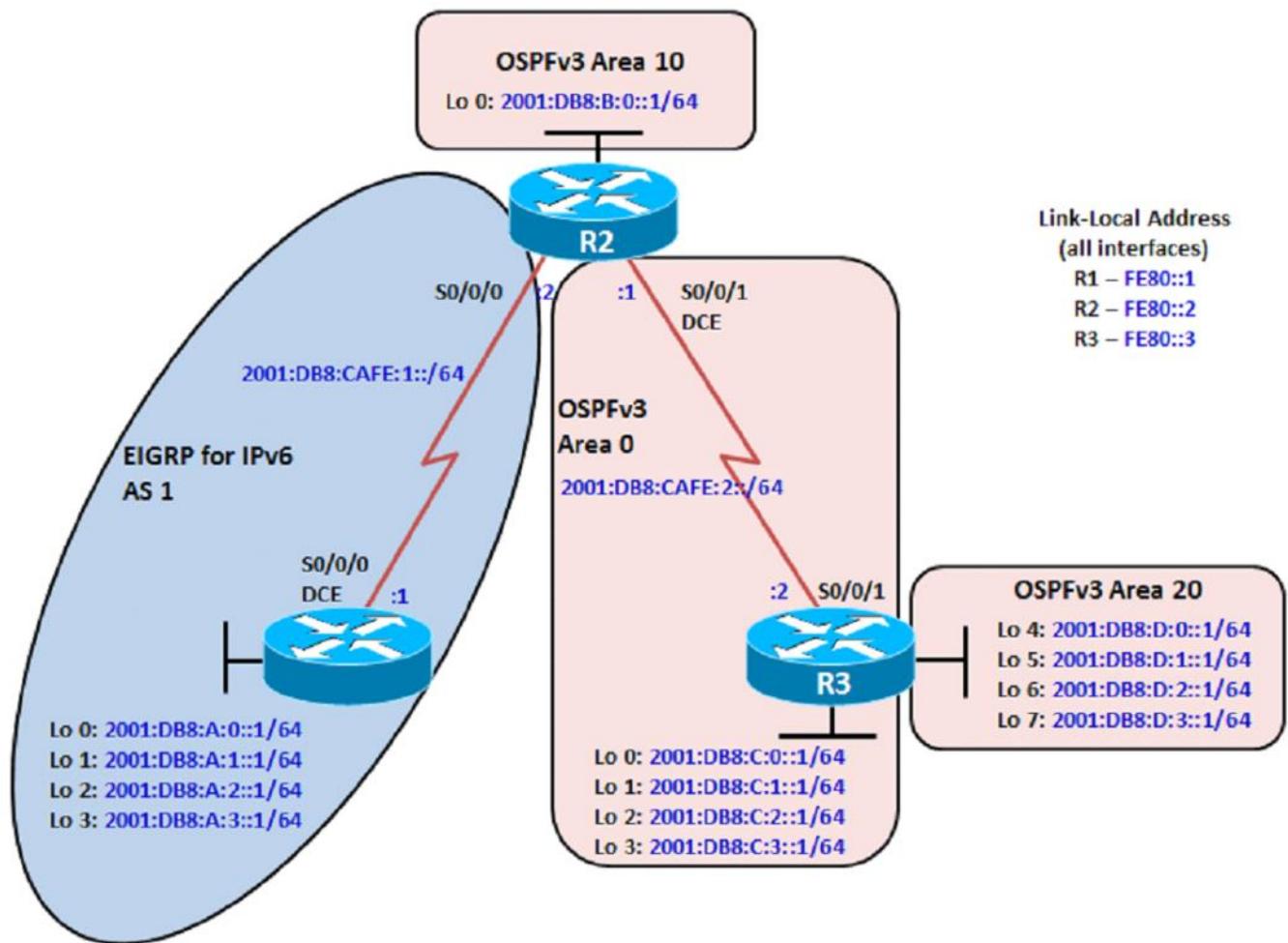
```
On Router5:  
Router5>enable  
Router5#configure terminal  
Router5(config)#router eigrp 1  
Router5(config-router)#network 5.5.5.5 0.0.0.0  
Router5(config-router)#network 192.168.1.64 0.0.0.31  
Router5(config-router)#network 198.51.100.20 0.0.0.3  
Router5(config-router)#exit  
Router5(config)#interface fastethernet 0/0  
Router5(config-if)#ip summary-address eigrp 1 192.168.1.64 255.255.255.224
```

```
On Router6:  
Router6>enable  
Router6#configure terminal  
Router6(config)#router eigrp 1  
Router6(config-router)#network 6.6.6.6 0.0.0.0  
Router6(config-router)#network 192.168.1.96 0.0.0.31  
Router6(config-router)#network 198.51.100.16 0.0.0.3  
Router6(config-router)#exit  
Router6(config)#interface fastethernet 0/0  
Router6(config-if)#ip summary-address eigrp 1 192.168.1.96 255.255.255.224
```

Screen clipping taken: 5/8/2019 3:34 PM

EIGRP ipv6 redis into OSPFv3

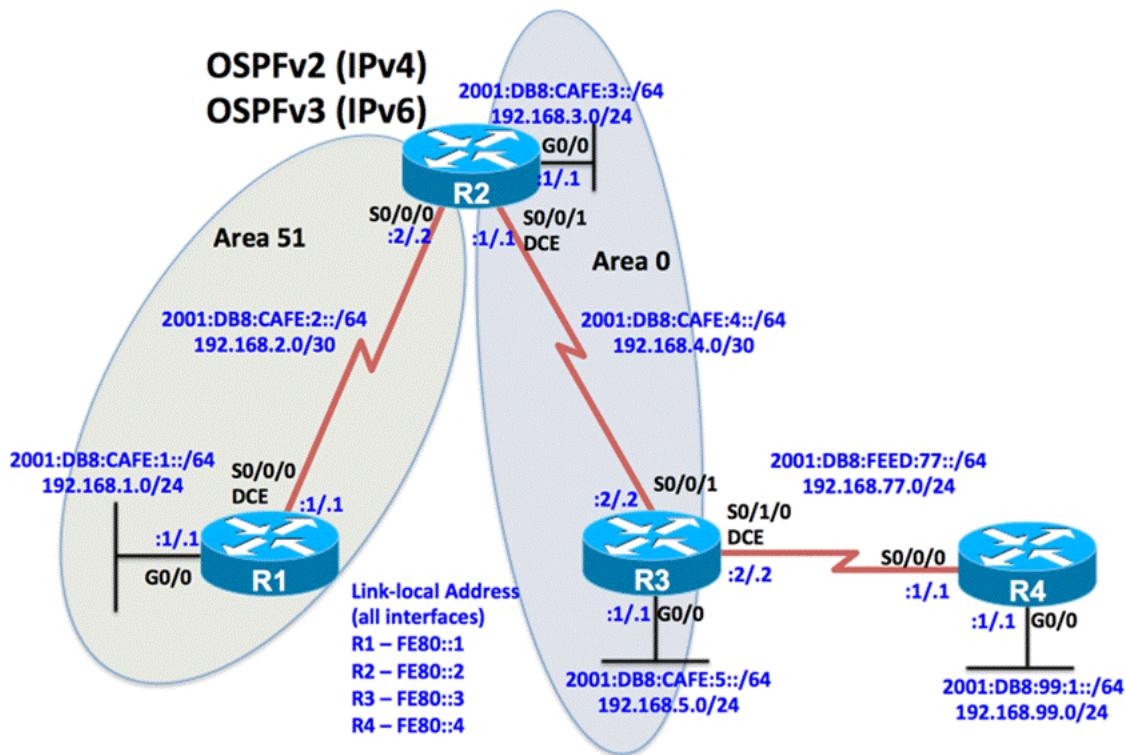
Wednesday, May 8, 2019 8:59 PM



Screen clipping taken: 5/8/2019 9:00 PM

Multilayer OSPF

Thursday, May 9, 2019 3:07 PM



ip dhcp relay information option → automatically add the circuit identifier suboption and the remote ID suboption

ip dhcp relay information check → check that the relay agent information option in forwarded BOOTREPLY messages is valid

ip dhcp relay information policy → Configures the reforwarding policy for a DHCP relay agent

ip dhcp relay information subscriber-id → enable an ISP to add a unique identifier

ip dhcp relay information trusted-sources → configures interfaces on a router as trusted sources

ip dhcp relay information → configured in global configuration mode applies to all interfaces

NAT64:

- + Use Network-specific prefix
- + Modify session during translation

NPTv6:

- + Modify IP header in transit
- + Map one IPv6 address prefix to another IPv6 prefix

+ OpenSent: wait for an OPEN message

+ OpenConfirm: wait for a KEEPALIVE or NOTIFICATION message

+ Established: UPDATE, NOTIFICATION and KEEPALIVE messages are exchanged with peers

+ Idle: refuse connections

+ Active: listen for and accept connection

+ Connect: wait for the connection to be completed

+ Target 1: When the LCP phase is complete and CHAP is negotiated between both devices, the authenticator sends a challenge message to the peer

- + Target 2: The peer responds with a value calculated through a one-way hash function (MD5)
 - + Target 3: The authenticator checks the response against its own calculation of the expected hash value if the values match the authentication is successful. Otherwise, the connection is terminated
-
- + SVC: A circuit that provides temporary on-demand connections between DTEs
 - + LMI: A signaling mechanism for Frame Relay devices
 - + DLCI: A locally significant ID
 - + FECN: An indicator edof congestion on the network
 - + PVC: A logical connection comprising two endpoints and a CIR