

Copyright Information

Copyright © 2008 Internetwork Expert, Inc. All rights reserved.

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, was developed by Internetwork Expert, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means without the prior written permission of Internetwork Expert, Inc.

Cisco®, Cisco® Systems, CCIE, and Cisco Certified Internetwork Expert, are registered trademarks of Cisco® Systems, Inc. and/or its affiliates in the U.S. and certain countries.

All other products and company names are the trademarks, registered trademarks, and service marks of the respective owners. Throughout this manual, Internetwork Expert, Inc. has used its best efforts to distinguish proprietary trademarks from descriptive names by following the capitalization styles used by the manufacturer.

Disclaimer

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab Exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE lab material is completely coincidental.

Table of Contents

| | |
|--|----|
| EIGRP | 1 |
| 5.1 EIGRP Network Statement..... | 1 |
| 5.2 EIGRP Auto-Summary | 1 |
| 5.3 EIGRP Split Horizon..... | 1 |
| 5.4 EIGRP MD5 Authentication..... | 2 |
| 5.5 EIGRP Key Chain Rotation | 2 |
| 5.6 EIGRP Unicast Updates..... | 2 |
| 5.7 EIGRP Default Network..... | 2 |
| 5.8 EIGRP Summarization | 3 |
| 5.9 EIGRP Summarization with Default Routing | 3 |
| 5.10 EIGRP Summarization with Leak Map | 3 |
| 5.11 EIGRP Floating Summarization | 3 |
| 5.12 EIGRP Poisoned Floating Summarization..... | 3 |
| 5.13 EIGRP Metric Weights | 4 |
| 5.14 EIGRP Traffic Engineering with Metric..... | 4 |
| 5.15 EIGRP Unequal Cost Load Balancing..... | 4 |
| 5.16 EIGRP Convergence Timers..... | 4 |
| 5.17 EIGRP Stub Routing | 4 |
| 5.18 EIGRP Stub Routing with Leak Map | 5 |
| 5.19 EIGRP Filtering with Passive Interface..... | 5 |
| 5.20 EIGRP Filtering with Prefix-Lists | 5 |
| 5.21 EIGRP Filtering with Standard Access-Lists | 5 |
| 5.22 EIGRP Filtering with Extended Access-Lists..... | 5 |
| 5.23 EIGRP Filtering with Offset Lists | 6 |
| 5.24 EIGRP Filtering with Administrative Distance..... | 6 |
| 5.25 EIGRP Filtering with Per Neighbor AD | 6 |
| 5.26 EIGRP Filtering with Route Maps..... | 6 |
| 5.27 EIGRP Bandwidth Pacing | 6 |
| 5.28 EIGRP Default Metric | 7 |
| 5.29 EIGRP Neighbor Logging..... | 7 |
| 5.30 EIGRP Router-ID | 7 |
| 5.31 EIGRP Maximum Hops | 7 |
| EIGRP Solutions | 9 |
| 5.1 EIGRP Network Statement..... | 9 |
| 5.2 EIGRP Auto-Summary | 15 |
| 5.3 EIGRP Split Horizon..... | 16 |
| 5.4 EIGRP MD5 Authentication..... | 18 |
| 5.5 EIGRP Key Chain Rotation | 20 |
| 5.6 EIGRP Unicast Updates..... | 22 |
| 5.7 EIGRP Default Network..... | 24 |
| 5.8 EIGRP Summarization | 26 |
| 5.9 EIGRP Summarization with Default Routing | 28 |

| | |
|--|----|
| 5.10 EIGRP Summarization with Leak Map | 29 |
| 5.11 EIGRP Floating Summarization | 31 |
| 5.12 EIGRP Poisoned Floating Summarization..... | 35 |
| 5.13 EIGRP Metric Weights | 36 |
| 5.14 EIGRP Traffic Engineering with Metric..... | 38 |
| 5.15 EIGRP Unequal Cost Load Balancing..... | 42 |
| 5.16 EIGRP Convergence Timers..... | 49 |
| 5.17 EIGRP Stub Routing | 55 |
| 5.18 EIGRP Stub Routing with Leak Map | 58 |
| 5.19 EIGRP Filtering with Passive Interface..... | 61 |
| 5.20 EIGRP Filtering with Prefix-Lists | 63 |
| 5.21 EIGRP Filtering with Standard Access-Lists | 66 |
| 5.22 EIGRP Filtering with Extended Access-Lists..... | 67 |
| 5.23 EIGRP Filtering with Offset Lists | 71 |
| 5.24 EIGRP Filtering with Administrative Distance..... | 74 |
| 5.25 EIGRP Filtering with Per Neighbor AD | 75 |
| 5.26 EIGRP Filtering with Route Maps..... | 78 |
| 5.27 EIGRP Bandwidth Pacing | 83 |
| 5.28 EIGRP Default Metric | 84 |
| 5.29 EIGRP Neighbor Logging..... | 85 |
| 5.30 EIGRP Router-ID | 86 |
| 5.31 EIGRP Maximum Hops | 88 |

EIGRP

 **Note**

Load the *Initial EIGRP* initial configurations prior to starting. Note that R4's link to VLAN 146 and the point-to-point link between R2 and R3 are disabled.

5.1 EIGRP Network Statement

- Configure EIGRP AS 100 on all devices in the internal topology.
- Enable EIGRP on all interfaces in the 150.X.0.0 and 155.X.0.0 networks on all devices.
- Any new interfaces added should not automatically have EIGRP enabled on them regardless of their IP addresses.
- Do not disable auto-summary.
- Note any reachability problems throughout the network.

5.2 EIGRP Auto-Summary

- Disable auto-summary on all devices running EIGRP.
- Note any changes in reachability throughout the network.

5.3 EIGRP Split Horizon

- Disable split-horizon for EIGRP on R5's connection to the Frame Relay network.
- Note any changes in reachability throughout the network.

5.4 EIGRP MD5 Authentication

- Configure EIGRP 10 on the link between R6 and BB1.
- Authenticate this adjacency with the MD5 key 1 using the password CISCO.
- Use a key-chain named MD5_KEYS.

5.5 EIGRP Key Chain Rotation

- Authenticate the EIGRP adjacencies on the Frame Relay network between R1, R2, R3, R4, and R5 using key 10 and the password CISCO10.
- Key 10 should be sent until midnight on Dec 31st 2030, and should be accepted for 15 minutes past this time.
- Configure a new key 20 with the password CISCO20 that is sent starting Jan 1st 2030, and is accepted any time after this time.
- Use a key-chain named KEY_ROTATION.

5.6 EIGRP Unicast Updates

- Configure R5 and SW2 so that they exchange EIGRP packets only as unicasts on their connection to VLAN 58.

5.7 EIGRP Default Network

- Redistribute between EIGRP AS 10 and EIGRP AS 100 on R6.
- Configure R6 to advertise the network 200.0.0.0/24 as the default network to all devices in EIGRP AS 100.

5.8 EIGRP Summarization

- Redistribute between RIP and EIGRP AS 100 on R4.
- Use the metric of R4's connection to VLAN 43 to translate RIP metrics into EIGRP metrics.
- Configure R4 to summarize the 30.0.0.0 subnets to R5 out the Frame Relay link, and the 31.0.0.0 subnets out the point-to-point link.
- Do not overlap any address space that R4 does not have a longer match to.
- If R4's point-to-point link is down traffic for the 30.0.0.0 subnets should be rerouted out the Frame Relay link.
- If R4's Frame Relay link is down traffic for the 31.0.0.0 subnets should be rerouted out the point-to-point link.

5.9 EIGRP Summarization with Default Routing

- Remove R6's default network advertisement.
- Remove R4's previous summarization.
- Configure summarization on R4's connections to R5 so that it only advertises a default route out to R5 via EIGRP.

5.10 EIGRP Summarization with Leak Map

- Configure a leak-map on R4 so that traffic going to R4's Loopback0 network is routed out the point-to-point link between R4 and R5.
- If this link is down traffic should still be rerouted out the Frame Relay connection between these devices.

5.11 EIGRP Floating Summarization

- Shutdown the point-to-point link between R4 and R5.
- Configure R5 to summarize the Loopback0 networks of R4 and R5 out to SW2; this route should not overlap any additional networks.
- Configure an equal longest match static route on R5 so that SW2 has reachability to both the Loopback0 networks of R4 and R5.

5.12 EIGRP Poisoned Floating Summarization

- Remove the previously configured static route on R5.
- Modify the administrative distance of the summary that R5 is generating to SW2 so that a route to Null0 is not installed.

 **Note**

Erase and reload all devices and load the *Basic EIGRP Routing* initial configurations before continuing.

5.13 EIGRP Metric Weights

- Configure all devices in EIGRP AS 100 so that only delay is used in the composite metric calculation.

5.14 EIGRP Traffic Engineering with Metric

- Configure a metric manipulation on SW1 so that traffic from SW3 to the Loopback0 network of R6 transits the link between R3 and R1.

5.15 EIGRP Unequal Cost Load Balancing

- Configure unequal cost load balancing so that traffic from R6 going to VLAN 9 is load balanced between R1 and SW1.
- The traffic share should be configured in such a way that the link to SW1 is used five times as much as the link to R1.
- Verify this by configuring per-packet load balancing on R6.

5.16 EIGRP Convergence Timers

- Configure R1 through R6 so that EIGRP hellos are sent every one second; these devices should inform their neighbors to declare them down if subsequent hellos are not received within three seconds.
- Configure SW1 through SW4 so that EIGRP hellos are sent every ten seconds; these devices should inform their neighbors to declare them down if subsequent hellos are not received within thirty seconds.
- Additionally configure AS 100 so that lost routes are considered Stuck In Active if a query response has not been heard within one minute.

5.17 EIGRP Stub Routing

- Configure the EIGRP stub feature in such a way that SW2 does not receive EIGRP query messages.
- Ensure that all devices in AS 100 still have IP reachability to VLAN 8.

5.18 EIGRP Stub Routing with Leak Map

- Configure the EIGRP stub feature in such a way that R5 does not receive EIGRP query messages.
- R5 should continue to advertise all learned routes with the exception of SW2's Loopback0 network.

5.19 EIGRP Filtering with Passive Interface

- Configure the passive-interface feature on R5, SW2, and SW4 so that EIGRP hello packets are not sent out the LAN segments without routers attached.
- Configure the passive-interface default feature on SW1 and SW3 so that EIGRP hello packets are not sent out the LAN segments without routers attached; ensure that full reachability is maintained after this change is made.

5.20 EIGRP Filtering with Prefix-Lists

- Configure a prefix-list on R4 so that it does not advertise the 30.0.0.0 and 31.0.0.0 subnets learned from BB3 out the point-to-point link to R5; use the most efficient list to accomplish this that will not deny any other networks than those subnets R4 is learning.
- Configure a prefix-list on R1 so that it does not install any updates received from R4 on the VLAN 146 segment.

5.21 EIGRP Filtering with Standard Access-Lists

- Configure a one line standard access-list on R6 to filter out all routes coming from BB1 that have an odd number in the third octet.

5.22 EIGRP Filtering with Extended Access-Lists

- Shutdown R5's point-to-point link to R4.
- Configure an extended access-list filter on R5 so that traffic for the Loopback0 networks of R4 and R6 is sent to R2.
- Traffic for the Loopback0 networks of R1 and R2 should be sent to R3.
- Traffic for the Loopback0 networks of SW1 and SW3 should be sent to R1.
- This filter should not affect any other updates on this segment.

5.23 EIGRP Filtering with Offset Lists

- Configure an offset-list on SW1 so traffic destined for R3's Loopback0 network is sent to R6.
- If the link to R6 is down traffic should be rerouted directly to R3.

5.24 EIGRP Filtering with Administrative Distance

- Configure administrative distance filtering on R6 so that it does not install the route to R4's Loopback0 network.

5.25 EIGRP Filtering with Per Neighbor AD

- Configure administrative distance filtering on R3 so that traffic destined for SW1's Loopback0 network is sent towards R1.

5.26 EIGRP Filtering with Route Maps

- Configure R4 to redistribute the VLAN 43 subnet into EIGRP with the tag value of 4.
- Configure a route-map filter on R2 that matches this tag value and denies the route from being installed in the routing table.
- Configure a route-map filter on R3 that denies EIGRP routes with a metric in the range of 500,000 – 750,000 from entering the routing table.
- These filters should not impact any other networks advertised by R4 or learned by R2 and R3.

5.27 EIGRP Bandwidth Pacing

- Configure R2 and R3 so that EIGRP can not use more than 154Kbps of bandwidth on the point-to-point link between them, assuming that the link speed is 1544Kbps.

5.28 EIGRP Default Metric

- Configure a static route on R2 for the prefix 222.22.2.2/32 that is reachable via BB2.
- Advertise this prefix into EIGRP as external routes using a default metric of 100Mbps, 100 microseconds of delay, maximum reliability, minimum load, and an MTU of 1500 bytes.

5.29 EIGRP Neighbor Logging

- Configure SW3 so that it does not log EIGRP neighbor adjacency events.
- Additionally EIGRP warning logs should not be generated more often than every 20 seconds.

5.30 EIGRP Router-ID

- Modify the EIGRP Router-ID on SW2 so that external EIGRP routes generated by R2 are ignored.

5.31 EIGRP Maximum Hops

- Configure all devices in EIGRP AS 100 so that routes with a hop count of greater than 10 are considered invalid.

EIGRP Solutions

5.1 EIGRP Network Statement

- Configure EIGRP AS 100 on all devices in the internal topology.
- Enable EIGRP on all interfaces in the 150.X.0.0 and 155.X.0.0 networks on all devices.
- Any new interfaces added should not automatically have EIGRP enabled on them regardless of their IP addresses.
- Do not disable auto-summary.
- Note any reachability problems throughout the network.

Configuration

R1:

```
router eigrp 100
 network 150.1.1.1 0.0.0.0
 network 155.1.0.1 0.0.0.0
 network 155.1.146.1 0.0.0.0
 network 155.1.13.1 0.0.0.0
```

R2:

```
router eigrp 100
 network 150.1.2.2 0.0.0.0
 network 155.1.0.2 0.0.0.0
```

R3:

```
router eigrp 100
 network 150.1.3.3 0.0.0.0
 network 155.1.0.3 0.0.0.0
 network 155.1.13.3 0.0.0.0
 network 155.1.37.3 0.0.0.0
```

R4:

```
router eigrp 100
 network 150.1.4.4 0.0.0.0
 network 155.1.0.4 0.0.0.0
 network 155.1.45.4 0.0.0.0
```

R5:

```
router eigrp 100
 network 150.1.5.5 0.0.0.0
 network 155.1.0.5 0.0.0.0
 network 155.1.5.5 0.0.0.0
 network 155.1.45.5 0.0.0.0
 network 155.1.58.5 0.0.0.0
```

R6:

```
router eigrp 100
 network 150.1.6.6 0.0.0.0
 network 155.1.67.6 0.0.0.0
 network 155.1.146.6 0.0.0.0
```

SW1:

```
ip routing
!
router eigrp 100
 network 150.1.7.7 0.0.0.0
 network 155.1.7.7 0.0.0.0
 network 155.1.37.7 0.0.0.0
 network 155.1.67.7 0.0.0.0
 network 155.1.79.7 0.0.0.0
```

SW2:

```
ip routing
!
router eigrp 100
 network 150.1.8.8 0.0.0.0
 network 155.1.8.8 0.0.0.0
 network 155.1.58.8 0.0.0.0
 network 155.1.108.8 0.0.0.0
```

SW3:

```
ip routing
!
router eigrp 100
 network 150.1.9.9 0.0.0.0
 network 155.1.9.9 0.0.0.0
 network 155.1.79.9 0.0.0.0
```

SW4:

```
ip routing
!
router eigrp 100
 network 150.1.10.10 0.0.0.0
 network 155.1.10.10 0.0.0.0
 network 155.1.108.10 0.0.0.0
```

Verification

Note

The network statement in EIGRP, like in OSPF, does not control what networks are being advertised, but instead controls what interfaces are running the EIGRP process. By using a wildcard address of 0.0.0.0 in the EIGRP network statement this means that only the interface with that particular IP address will have the EIGRP process enabled. By using all zeros in the wildcard mask there is no question as to which interfaces are running the process, and new interfaces added to the device will not automatically be running the EIGRP process.

Once the network statement is configured the first verification you should always do is to check the neighbor adjacencies with the **show ip eigrp neighbors** command. A “Q Cnt” (queue count) of zero means that there are no updates waiting to be sent and the network is converged.

Rack1R1#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|-------------|-----------|------------|----------|-----------|-----|-------|---------|
| 2 | 155.1.146.6 | Fa0/0 | 13 | 00:05:10 | 4 | 200 | 0 | 17 |
| 1 | 155.1.0.5 | Se0/0.1 | 177 | 00:05:19 | 44 | 264 | 0 | 22 |
| 0 | 155.1.13.3 | Se0/1 | 13 | 00:05:31 | 23 | 200 | 0 | 44 |

Rack1R2#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|-----------|-----------|------------|----------|-----------|-----|-------|---------|
| 0 | 155.1.0.5 | Se0/0.1 | 174 | 00:05:22 | 44 | 264 | 0 | 22 |

Rack1R3#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|------------|----------|-----------|------|-------|---------|
| 2 | 155.1.37.7 | Fa0/0 | 12 | 00:05:08 | 3 | 200 | 0 | 17 |
| 1 | 155.1.0.5 | Se1/0.1 | 167 | 00:05:29 | 43 | 1140 | 0 | 22 |
| 0 | 155.1.13.1 | Se1/2 | 11 | 00:05:41 | 20 | 1140 | 0 | 37 |

Rack1R4#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|------------|----------|-----------|------|-------|---------|
| 1 | 155.1.0.5 | Se0/0.1 | 163 | 00:02:59 | 830 | 4980 | 0 | 28 |
| 0 | 155.1.45.5 | Se0/1 | 10 | 00:02:59 | 54 | 324 | 0 | 27 |

Rack1R5#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|------------|----------|-----------|------|-------|---------|
| 5 | 155.1.0.4 | Se0/0 | 12 | 00:03:02 | 88 | 528 | 0 | 10 |
| 4 | 155.1.45.4 | Se0/1 | 13 | 00:03:02 | 43 | 258 | 0 | 8 |
| 3 | 155.1.58.8 | Fa0/0 | 14 | 00:05:00 | 9 | 200 | 0 | 7 |
| 2 | 155.1.0.1 | Se0/0 | 11 | 00:05:36 | 240 | 1440 | 0 | 35 |
| 1 | 155.1.0.2 | Se0/0 | 12 | 00:05:36 | 242 | 1452 | 0 | 3 |
| 0 | 155.1.0.3 | Se0/0 | 14 | 00:05:36 | 207 | 1242 | 0 | 46 |

Rack1R6#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|-------------|-----------|---------------|----------|--------------|-----|----------|------------|
| 1 | 155.1.67.7 | Fa0/0.67 | 14 | 00:05:20 | 7 | 200 | 0 | 16 |
| 0 | 155.1.146.1 | Fa0/0.146 | 10 | 00:05:33 | 2 | 200 | 0 | 36 |

Rack1SW1#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|---------------|----------|--------------|------|----------|------------|
| 2 | 155.1.79.9 | Vl79 | 13 | 00:05:01 | 13 | 200 | 0 | 3 |
| 1 | 155.1.67.6 | Vl67 | 12 | 00:05:24 | 521 | 3126 | 0 | 18 |
| 0 | 155.1.37.3 | Fa0/3 | 13 | 00:05:24 | 418 | 2508 | 0 | 45 |

Rack1SW2#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|--------------|-----------|---------------|----------|--------------|------|----------|------------|
| 1 | 155.1.108.10 | Po1 | 14 | 00:04:48 | 8 | 200 | 0 | 2 |
| 0 | 155.1.58.5 | Vl58 | 13 | 00:05:16 | 814 | 4884 | 0 | 23 |

Rack1SW3#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|---------------|----------|--------------|------|----------|------------|
| 0 | 155.1.79.7 | Vl79 | 11 | 00:05:11 | 509 | 3054 | 0 | 18 |

Rack1SW4#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|-------------|-----------|---------------|----------|--------------|------|----------|------------|
| 0 | 155.1.108.8 | Po1 | 12 | 00:04:54 | 1417 | 5000 | 0 | 6 |

Two separate design problems exist in the current network topology as configured in this section. The first is discontinuous networks, and the second is split-horizon.

The first problem can be seen from the routing table output of any device in the network. Each device has a Loopback0 interface in the major network 150.1.0.0/16, while the transit network between the devices is 155.1.0.0/16. When auto-summary is on in EIGRP networks are summarized as they pass through the major network boundary. This is unlike RIP, which automatically summarizes networks anytime there is a different major network or different subnet masks.

The result of auto-summary being on and the discontinuous networks is that the subnets of the 150.1.0.0/16 network cannot be advertised. This is due to the fact that all routers generate a local route for 150.1.0.0/16 to Null0, and advertise this into the EIGRP topology. When another device receive the advertisement 150.1.0.0/16, which it already has installed locally, the local route is preferred over the received route. This means that since the subnets of 150.1.0.0/16 are not exchanged, no device will have reachability to the Loopback0 networks of the other devices in the topology.

```
Rack1SW4#show ip route eigrp
 155.1.0.0/16 is variably subnetted, 15 subnets, 2 masks
D    155.1.146.0/24 [90/2175232] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.8.0/24 [90/15616] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.9.0/24 [90/2175744] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.13.0/24 [90/2684672] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.0.0/24 [90/2172672] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.0.0/16 is a summary, 00:05:06, Null0
D    155.1.7.0/24 [90/2175488] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.5.0/24 [90/30976] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.58.0/24 [90/15616] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.45.0/24 [90/2172672] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.37.0/24 [90/2175232] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.79.0/24 [90/2175488] via 155.1.108.8, 00:05:01, Port-channel1
D    155.1.67.0/24 [90/2175488] via 155.1.108.8, 00:05:01, Port-channel1
 150.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
D    150.1.0.0/16 is a summary, 00:05:02, Null0
```

The second problem related to auto-summary can be seen on R2. Since R2's only connection to the rest of the EIGRP network is through the Frame Relay network, all advertisements that R5 receives in the Frame Relay interface cannot be sent back out to R2. This is similar to the RIP split-horizon problem previously introduced, however EIGRP split-horizon is enabled on all interfaces, regardless if they are main interfaces or subinterfaces. To resolve this issue R5 needs to disabled split-horizon for this EIGRP process by using the command `no ip split-horizon eigrp 100` under the Frame Relay interface.

```
Rack1R2#show ip route eigrp
 155.1.0.0/16 is variably subnetted, 8 subnets, 2 masks
D    155.1.10.0/24 [90/2175232] via 155.1.0.5, 00:06:17, Serial0/0.1
D    155.1.8.0/24 [90/2172672] via 155.1.0.5, 00:06:49, Serial0/0.1
D    155.1.0.0/16 is a summary, 00:07:50, Null0
D    155.1.5.0/24 [90/2172416] via 155.1.0.5, 00:07:24, Serial0/0.1
D    155.1.58.0/24 [90/2172416] via 155.1.0.5, 00:07:24, Serial0/0.1
D    155.1.45.0/24 [90/2681856] via 155.1.0.5, 00:07:24, Serial0/0.1
D    155.1.108.0/24 [90/2174976] via 155.1.0.5, 00:06:47, Serial0/0.1
 150.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
D    150.1.0.0/16 is a summary, 00:07:24, Null0
```

5.2 EIGRP Auto-Summary

- Disable auto-summary on all devices running EIGRP.
- Note any changes in reachability throughout the network.

Configuration

```
R1 - R6, SW1 - SW4:
router eigrp 100
  no auto-summary
```

Verification

Note

With EIGRP auto-summary disabled the subnets of the discontinuous network 150.1.0.0/16 can be advertised to all neighbors. The result of this can be seen anywhere in the topology through the `show ip route eigrp` output, as now the individual /24 subnets of 150.1.0.0 are installed.

As a general rule auto-summary for EIGRP would always be disabled in a real design in order for the protocol to function in a truly “classless” fashion. Within the scope of the lab exam it would be safe to assume that auto-summary for EIGRP could be disabled unless there is a specific question telling you to leave auto-summary on.

```
Rack1SW4#show ip route eigrp
 155.1.0.0/24 is subnetted, 14 subnets
D    155.1.146.0 [90/2175232] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.8.0 [90/15616] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.9.0 [90/2175744] via 155.1.108.8, 00:00:11, Port-channel1
D    155.1.13.0 [90/2684672] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.0.0 [90/2172672] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.7.0 [90/2175488] via 155.1.108.8, 00:00:11, Port-channel1
D    155.1.5.0 [90/30976] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.58.0 [90/15616] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.45.0 [90/2172672] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.37.0 [90/2175232] via 155.1.108.8, 00:00:15, Port-channel1
D    155.1.79.0 [90/2175488] via 155.1.108.8, 00:00:12, Port-channel1
D    155.1.67.0 [90/2175488] via 155.1.108.8, 00:00:15, Port-channel1
 150.1.0.0/24 is subnetted, 10 subnets
D    150.1.7.0 [90/2303232] via 155.1.108.8, 00:00:12, Port-channel1
D    150.1.6.0 [90/2303232] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.5.0 [90/143616] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.4.0 [90/2300672] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.3.0 [90/2300672] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.2.0 [90/2300672] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.1.0 [90/2300672] via 155.1.108.8, 00:00:16, Port-channel1
D    150.1.9.0 [90/2303488] via 155.1.108.8, 00:00:13, Port-channel1
D    150.1.8.0 [90/143360] via 155.1.108.8, 00:00:16, Port-channel1
```

5.3 EIGRP Split Horizon

- Disable split-horizon for EIGRP on R5's connection to the Frame Relay network.
- Note any changes in reachability throughout the network.

Configuration

```
R5:
interface Serial0/0
no ip split-horizon eigrp 100
```

Verification

Note

Once split-horizon for EIGRP is disabled on R5's link to the Frame Relay network updates can be exchanged to R2 as they come from other neighbors on the Frame Relay segment, such as R1. The result of this can be seen in the routing table of R2.

```
Rack1R2#show ip route eigrp
155.1.0.0/24 is subnetted, 14 subnets
D    155.1.146.0 [90/2684416] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.10.0 [90/2175232] via 155.1.0.5, 00:04:15, Serial0/0.1
D    155.1.8.0 [90/2172672] via 155.1.0.5, 00:04:19, Serial0/0.1
D    155.1.9.0 [90/2684928] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.13.0 [90/3193856] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.7.0 [90/2684672] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.5.0 [90/2172416] via 155.1.0.5, 00:27:31, Serial0/0.1
D    155.1.58.0 [90/2172416] via 155.1.0.5, 00:27:31, Serial0/0.1
D    155.1.45.0 [90/2681856] via 155.1.0.5, 00:27:31, Serial0/0.1
D    155.1.37.0 [90/2684416] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.79.0 [90/2684672] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.67.0 [90/2684672] via 155.1.0.5, 00:00:40, Serial0/0.1
D    155.1.108.0 [90/2174976] via 155.1.0.5, 00:04:19, Serial0/0.1
150.1.0.0/24 is subnetted, 10 subnets
D    150.1.7.0 [90/2812416] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.6.0 [90/2812416] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.5.0 [90/2297856] via 155.1.0.5, 00:04:23, Serial0/0.1
D    150.1.4.0 [90/2809856] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.3.0 [90/2809856] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.1.0 [90/2809856] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.10.0 [90/2302976] via 155.1.0.5, 00:04:17, Serial0/0.1
D    150.1.9.0 [90/2812672] via 155.1.0.5, 00:00:42, Serial0/0.1
D    150.1.8.0 [90/2300416] via 155.1.0.5, 00:04:21, Serial0/0.1
```

Although disabling split-horizon on R5 does not cause a routing loop, it does add additional route replication into the topology. For example if we look at the EIGRP topology table on R2 for the prefix 150.1.2.0/24, which is directly connected, we can see that this prefix is originated from the connected route, but is also learned back in from R5.

```
Rack1R2#show ip eigrp topology 150.1.2.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.2.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
  0.0.0.0 (Loopback0), from Connected, Send flag is 0x0
    Composite metric is (128256/0), Route is Internal
  Vector metric:
    Minimum bandwidth is 10000000 Kbit
    Total delay is 5000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1514
    Hop count is 0
  155.1.0.5 (Serial0/0.1), from 155.1.0.5, Send flag is 0x0
    Composite metric is (2809856/2297856), Route is Internal
  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 45000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 2
```

Even though there are multiple paths to the same destination a loop cannot occur based on the EIGRP feasibility condition. The feasibility condition determines which routes from the EIGRP topology will actually be used for forwarding in the IP routing table.

First off the complete end-to-end composite metric is compared between routes. In this case R2's local route has a metric of 128,256, while R5's route has a metric of 2,809,856. This value is seen as the first value in parenthesis before the slash. The lower of these values, 128,256, is considered the Feasible Distance, and is the end-to-end metric of the active forwarding path. This active forwarding path is called the Successor.

Next, R2 compares the metric that the upstream neighbor is advertising for the destination. In this case R5 is advertising a metric of 2,297,856. This value seen as the second number inside parenthesis, the Advertised Distance, is used to find alternate loop-free forwarding paths. If the Advertised Distance of a neighbor's route is lower than the current Feasible Distance of the Successor, the route is considered an alternate path, or a Feasible Successor.

In this case R5's Advertised Distance is higher than R2's Feasible Distance, so the route is discarded and not considered a valid path.

5.4 EIGRP MD5 Authentication

- Configure EIGRP 10 on the link between R6 and BB1.
- Authenticate this adjacency with the MD5 key 1 using the password CISCO.
- Use a key-chain named MD5_KEYS.

Configuration

```
R6:
key chain MD5_KEYS
  key 1
    key-string CISCO
!
interface Serial0/0
  ip authentication mode eigrp 10 md5
  ip authentication key-chain eigrp 10 MD5_KEYS
!
router eigrp 10
  network 54.1.1.6 0.0.0.0
  no auto-summary
```

Verification

Note

Without authentication configured hello packets from BB1 are ignored.

Rack1R6#debug eigrp packet

```
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB,
  SIAQUERY, SIAREPLY)
```

```
EIGRP: Serial0/0: ignored packet from 54.1.1.254, opcode = 5
(authentication off)
```

If authentication is successful the neighbor relationship should appear immediately.

```
Rack1R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#
Rack1R6(config)#interface Serial0/0
Rack1R6(config-if)#ip authentication mode eigrp 10 md5
Rack1R6(config-if)#end
Rack1R6#
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 10: Neighbor 54.1.1.254 (Serial0/0) is
up: new adjacency

Rack1R6#show ip route eigrp 10
D    200.0.0.0/24 [90/2297856] via 54.1.1.254, 00:00:10, Serial0/0
D    200.0.1.0/24 [90/2297856] via 54.1.1.254, 00:00:10, Serial0/0
D    200.0.2.0/24 [90/2297856] via 54.1.1.254, 00:00:10, Serial0/0
D    200.0.3.0/24 [90/2297856] via 54.1.1.254, 00:00:10, Serial0/0
```

Pitfall

Like EIGRP a white space in the key-string can cause authentication failure.

```
Rack1R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#key chain MD5_KEYS
Rack1R6(config-keychain)#key 1
Rack1R6(config-keychain-key)#key-string CISCO ?
LINE    <cr>

Rack1R6(config-keychain-key)#key-string CISCO
Rack1R6(config-keychain-key)#interface Serial0/0
Rack1R6(config-if)#ip authentication mode eigrp 10 md5
Rack1R6(config-if)#ip authentication key-chain eigrp 10 MD5_KEYS
Rack1R6(config-if)#end
Rack1R6#
%SYS-5-CONFIG_I: Configured from console by console
Rack1R6#show key chain
Key-chain MD5_KEYS:
  key 1 -- text "CISCO "
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]

Rack1R6#debug eigrp packets
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB,
  SIAQUERY, SIAREPLY)
Rack1R6#
EIGRP: pkt key id = 1, authentication mismatch
EIGRP: Serial0/0: ignored packet from 54.1.1.254, opcode = 5 (invalid
authentication)
```

5.5 EIGRP Key Chain Rotation

- Authenticate the EIGRP adjacencies on the Frame Relay network between R1, R2, R3, R4, and R5 using key 10 and the password CISCO10.
- Key 10 should be sent until 12:05 AM on Jan 1st 2030, and should be accepted for 10 minutes past this time.
- Configure a new key 20 with the password CISCO20 that is sent starting 12:00 AM on Jan 1st 2030, and is accepted any time after this time.
- Use a key-chain named KEY_ROTATION.

Configuration

R1, R2, R4:

```
key chain KEY_ROTATION
  key 10
    key-string CISCO10
    accept-lifetime 00:00:00 Jan 1 1993 00:15:00 Jan 1 2030
    send-lifetime 00:00:00 Jan 1 1993 00:05:00 Jan 1 2030
  key 20
    key-string CISCO20
    accept-lifetime 00:00:00 Jan 1 2030 infinite
    send-lifetime 00:00:00 Jan 1 2030 infinite
!
interface Serial0/0.1 point-to-point
  ip authentication mode eigrp 100 md5
  ip authentication key-chain eigrp 100 KEY_ROTATION
```

R3:

```
key chain KEY_ROTATION
  key 10
    key-string CISCO10
    accept-lifetime 00:00:00 Jan 1 1993 00:15:00 Jan 1 2030
    send-lifetime 00:00:00 Jan 1 1993 00:05:00 Jan 1 2030
  key 20
    key-string CISCO20
    accept-lifetime 00:00:00 Jan 1 2030 infinite
    send-lifetime 00:00:00 Jan 1 2030 infinite
!
interface Serial1/0.1 point-to-point
  ip authentication mode eigrp 100 md5
  ip authentication key-chain eigrp 100 KEY_ROTATION
```



```

R5:
key chain KEY_ROTATION
  key 10
    key-string CISCO10
    accept-lifetime 00:00:00 Jan 1 1993 00:15:00 Jan 1 2030
    send-lifetime 00:00:00 Jan 1 1993 00:05:00 Jan 1 2030
  key 20
    key-string CISCO20
    accept-lifetime 00:00:00 Jan 1 2030 infinite
    send-lifetime 00:00:00 Jan 1 2030 infinite
!
interface Serial0/0 point-to-point
 ip authentication mode eigrp 100 md5
 ip authentication key-chain eigrp 100 KEY_ROTATION

```

Verification

☠ Pitfall

Anytime time based authentication is configured ensure that all devices agree on the same time. This can be manually configured with the `clock set` command or through NTP. Also the additional overlap of sending/receiving keys ensures that a drift away from the accurate time will not cause routing adjacencies to be lost.

```

Rack1R2#show clock
00:04:55.223 UTC Tue Jan 1 2030

```

```

Rack1R2#show key chain KEY_ROTATION
Key-chain KEY_ROTATION:
  key 10 -- text "CISCO10"
    accept lifetime (00:00:00 UTC Jan 1 1993) - (00:15:00 UTC Jan 1 2030)
    [valid now]
    send lifetime (00:00:00 UTC Jan 1 1993) - (00:05:00 UTC Jan 1 2030)
    [valid now]
  key 20 -- text "CISCO20"
    accept lifetime (00:00:00 UTC Jan 1 2030) - (infinite) [valid now]
    send lifetime (00:00:00 UTC Jan 1 2030) - (infinite) [valid now]

```

```

Rack1R2#show clock
00:05:30.954 UTC Tue Jan 1 2030

```

```

Rack1R2#show key chain KEY_ROTATION
Key-chain KEY_ROTATION:
  key 10 -- text "CISCO10"
    accept lifetime (00:00:00 UTC Jan 1 1993) - (00:15:00 UTC Jan 1 2030)
    [valid now]
    send lifetime (00:00:00 UTC Jan 1 1993) - (00:05:00 UTC Jan 1 2030)
  key 20 -- text "CISCO20"
    accept lifetime (00:00:00 UTC Jan 1 2030) - (infinite) [valid now]
    send lifetime (00:00:00 UTC Jan 1 2030) - (infinite) [valid now]

```

5.6 EIGRP Unicast Updates

- Configure R5 and SW2 so that they exchange EIGRP packets only as unicasts on their connection to VLAN 58.

Configuration

```
R5:  
router eigrp 100  
  neighbor 155.1.58.8 FastEthernet0/0
```

```
SW2:  
router eigrp 100  
  neighbor 155.1.58.5 Vlan58
```

Verification

Note

By default EIGRP hello packets are sent to the multicast address 224.0.0.10, while topology synchronization between two neighbors is unicast. Like RIP the **neighbor** statement under the EIGRP process is used to send hello packets as unicasts. However unlike RIP the **passive-interface** command is not needed to suppress the sending of the multicast hello. This means that if the **neighbor** statement is configured on one end of the adjacency it is required that the neighbor statement be configured on the other end.

Rack1R5#debug ip packet detail

```
IP packet debugging is on (detailed)
IP: s=155.1.58.5 (local), d=224.0.0.10 (FastEthernet0/0), len 60,
sending broad/multicast, proto=88
IP: s=155.1.58.8 (FastEthernet0/0), d=224.0.0.10, len 60, rcvd 2,
proto=88
```

```
Rack1R5#config t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Rack1R5(config)#router eigrp 100
```

```
Rack1R5(config-router)#neighbor 155.1.58.8 FastEthernet0/0
```

```
Rack1R5(config-router)#end
```

```
Rack1R5#
```

```
Rack1SW2#config t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Rack1SW2(config)#router eigrp 100
```

```
Rack1SW2(config-router)#neighbor 155.1.58.5 Vlan58
```

```
Rack1SW2(config-router)#end
```

```
Rack1SW2#
```

Rack1R5#debug ip packet detail

```
IP packet debugging is on (detailed)
IP: s=155.1.58.5 (local), d=155.1.58.8 (FastEthernet0/0), len 60,
sending, proto=88
IP: tableid=0, s=155.1.58.8 (FastEthernet0/0), d=155.1.58.5
(FastEthernet0/0), routed via RIB
IP: s=155.1.58.8 (FastEthernet0/0), d=155.1.58.5 (FastEthernet0/0), len
60, rcvd 3, proto=88
```

5.7 EIGRP Default Network

- Redistribute between EIGRP AS 10 and EIGRP AS 100 on R6.
- Configure R6 to advertise the network 200.0.0.0/24 as the default network to all devices in EIGRP AS 100.

Configuration

```
R6:
router eigrp 100
 redistribute eigrp 10
!
router eigrp 10
 redistribute eigrp 100
!
ip default-network 200.0.0.0
```

Verification

Note

The original implementation of IGRP did not support the advertisement of the network 0.0.0.0/0, so the `ip default-network` command was used as a workaround. Although EIGRP does support the direct advertisement of 0.0.0.0/0, it also inherits the default network behavior from IGRP. A default network is a classful major network that is advertised as the candidate destination for unknown traffic to be forwarded towards. This network is denoted with an * in the routing table, as seen from the below output on SW4.

Rack1SW4#show ip route

```
Codes: 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, E - EGP
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is 155.1.108.8 to network 200.0.0.0
```

```
155.1.0.0/24 is subnetted, 14 subnets
D    155.1.146.0 [90/2175232] via 155.1.108.8, 00:16:23, Port-channel1
C    155.1.10.0 is directly connected, Vlan10
D    155.1.7.0 [90/2175488] via 155.1.108.8, 00:16:24, Port-channel1
D    155.1.5.0 [90/30976] via 155.1.108.8, 00:16:24, Port-channel1
D    155.1.58.0 [90/15616] via 155.1.108.8, 01:34:41, Port-channel1
D    155.1.45.0 [90/2172672] via 155.1.108.8, 00:16:24, Port-channel1
D    155.1.37.0 [90/2175232] via 155.1.108.8, 00:16:24, Port-channel1
D    155.1.79.0 [90/2175488] via 155.1.108.8, 00:16:24, Port-channel1
D    155.1.67.0 [90/2175488] via 155.1.108.8, 00:16:24, Port-channel1
C    155.1.108.0 is directly connected, Port-channel1
D*EX 200.0.0.0/24 [170/2815232] via 155.1.108.8, 00:08:51, Port-channel1
<output omitted>
```

If we trace the path of the default network back to the source we can see that the gateway of last resort (the default next-hop) changes on a per router basis.

```
Rack1SW4#show ip route | include last resort|D\*
```

```
Gateway of last resort is 155.1.108.8 to network 200.0.0.0
D*EX 200.0.0.0/24 [170/2815232] via 155.1.108.8, 00:16:04, Port-channel1
```

```
Rack1SW2#show ip route | include last resort|D\*
```

```
Gateway of last resort is 155.1.58.5 to network 200.0.0.0
D*EX 200.0.0.0/24 [170/2812672] via 155.1.58.5, 00:16:15, Vlan58
```

```
Rack1R5#show ip route | include last resort|D\*
```

```
Gateway of last resort is 155.1.0.1 to network 200.0.0.0
D*EX 200.0.0.0/24 [170/2812416] via 155.1.0.1, 00:16:18, Serial0/0
```

```
Rack1R1#show ip route | include last resort|D\*
```

```
Gateway of last resort is 155.1.146.6 to network 200.0.0.0
D*EX 200.0.0.0/24 [170/2300416] via 155.1.146.6, 00:16:21, FastEthernet0/0
```

```
Rack1R6#show ip route | include last resort|D\*
```

```
Gateway of last resort is 54.1.1.254 to network 200.0.0.0
D* 200.0.0.0/24 [90/2297856] via 54.1.1.254, 00:53:29, Serial0/0
```

While technically not a “default route”, the result of the default network is the same. Traffic for unknown destinations is forwarded towards the device that originates the default network.

```
Rack1SW4#show ip route 1.2.3.4
```

```
% Network not in table
```

```
Rack1SW4#traceroute 1.2.3.4
```

```
Type escape sequence to abort.
```

```
Tracing the route to 1.2.3.4
```

```
 1 155.1.108.8 0 msec 0 msec 0 msec
 2 155.1.58.5 0 msec 4 msec 0 msec
 3 155.1.0.1 28 msec 28 msec 28 msec
 4 155.1.146.6 28 msec 28 msec 28 msec
 5 54.1.1.254 44 msec 48 msec 44 msec
 6 54.1.1.254 !H * !H
```

5.8 EIGRP Summarization

- Redistribute between RIP and EIGRP AS 100 on R4.
- Use the metric of R4's connection to VLAN 43 to translate RIP metrics into EIGRP metrics.
- Configure R4 to summarize the 30.0.0.0 subnets to R5 out the Frame Relay link, and the 31.0.0.0 subnets out the point-to-point link.
- Do not overlap any address space that R4 does not have a longer match to.
- If R4's point-to-point link is down traffic for the 30.0.0.0 subnets should be rerouted out the Frame Relay link.
- If R4's Frame Relay link is down traffic for the 31.0.0.0 subnets should be rerouted out the point-to-point link.

Configuration

```
R4:
interface Serial0/0.1 point-to-point
 ip summary-address eigrp 100 30.0.0.0 255.252.0.0 5
!
interface Serial0/1
 ip summary-address eigrp 100 31.0.0.0 255.252.0.0 5
!
router eigrp 100
 redistribute rip metric 100000 10 255 1 1500
!
router rip
 redistribute eigrp 100 metric 1
```

Verification

Note

Like EIGRP, EIGRP supports summarization at the interface level anywhere throughout the topology, but does not have the limitation of not being able to summarize beyond the classful boundary. When a summary is configured in EIGRP all subnets that make up the summary are suppressed from being advertised out the link. Design-wise this feature can be used to both reduce the size of the routing table and to limit the scope of EIGRP query messages.

In the below output we can see that R5 learns the summary 30.0.0.0/14 in the Frame Relay network, and the /16 subnets in the point-to-point link. Based on longest match routing we can infer that R5 will send traffic for any subnet of the aggregate out the point-to-point link.

```
Rack1R5#show ip route | include 30\.|31\.
  31.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D EX   31.3.0.0/16 [170/2172416] via 155.1.0.4, 00:01:22, Serial0/0
D EX   31.2.0.0/16 [170/2172416] via 155.1.0.4, 00:01:22, Serial0/0
D EX   31.1.0.0/16 [170/2172416] via 155.1.0.4, 00:01:22, Serial0/0
D EX   31.0.0.0/16 [170/2172416] via 155.1.0.4, 00:01:22, Serial0/0
D      31.0.0.0/14 [90/2172416] via 155.1.45.4, 00:01:21, Serial0/1
  30.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D EX   30.2.0.0/16 [170/2172416] via 155.1.45.4, 00:01:20, Serial0/1
D EX   30.3.0.0/16 [170/2172416] via 155.1.45.4, 00:01:20, Serial0/1
D EX   30.0.0.0/16 [170/2172416] via 155.1.45.4, 00:01:20, Serial0/1
D      30.0.0.0/14 [90/2172416] via 155.1.0.4, 00:01:27, Serial0/0
D EX   30.1.0.0/16 [170/2172416] via 155.1.45.4, 00:01:20, Serial0/1
```

```
Rack1R5#traceroute 30.0.0.1
```

```
Type escape sequence to abort.
Tracing the route to 30.0.0.1
```

```
 1 155.1.45.4 16 msec 12 msec 16 msec
 2 204.12.1.254 16 msec * 16 msec
```

If the point-to-point link fails the longest match for these destinations becomes the /14 summary, and traffic is routed out the Frame Relay network.

```
Rack1R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface Serial0/1
Rack1R4(config-if)#shutdown
Rack1R4(config-if)#
```

```
Rack1R5#traceroute 30.0.0.1
```

```
Type escape sequence to abort.
Tracing the route to 30.0.0.1
```

```
 1 155.1.0.4 32 msec 28 msec 32 msec
 2 204.12.1.254 32 msec * 32 msec
```

```
Rack1R5#show ip route | include 30\.|31\.
  31.0.0.0/16 is subnetted, 4 subnets
D EX   31.3.0.0 [170/2172416] via 155.1.0.4, 00:01:55, Serial0/0
D EX   31.2.0.0 [170/2172416] via 155.1.0.4, 00:01:55, Serial0/0
D EX   31.1.0.0 [170/2172416] via 155.1.0.4, 00:01:55, Serial0/0
D EX   31.0.0.0 [170/2172416] via 155.1.0.4, 00:01:55, Serial0/0
  30.0.0.0/14 is subnetted, 1 subnets
D      30.0.0.0 [90/2172416] via 155.1.0.4, 00:02:00, Serial0/0
```

5.9 EIGRP Summarization with Default Routing

- Remove R6's default network advertisement.
- Remove R4's previous summarization.
- Configure summarization on R4's connections to R5 so that it only advertises a default route out to R5 via EIGRP.

Configuration

```
R4:
interface Serial0/0.1 point-to-point
 ip summary-address eigrp 100 0.0.0.0 0.0.0.0 5
!
interface Serial0/1
 ip summary-address eigrp 100 0.0.0.0 0.0.0.0 5
```

Verification

Note

Summarization can also be used to originate a default route in EIGRP. The disadvantage of this configuration however is that all subnets previously advertised out an interface will be suppressed, since all IPv4 networks are a subnet of the aggregate 0.0.0.0/0.

```
Rack1R5#show ip route | include via 155.1.(0|45).4
D*  0.0.0.0/0 [90/2172416] via 155.1.45.4, 00:00:36, Serial0/1
    [90/2172416] via 155.1.0.4, 00:00:36, Serial0/0
```

```
Rack1R5#show ip route 30.0.0.1
% Network not in table
```

```
Rack1R5#traceroute 30.0.0.1
```

```
Type escape sequence to abort.
Tracing the route to 30.0.0.1
```

```
 1 155.1.45.4 24 msec
   155.1.0.4 28 msec
   155.1.45.4 20 msec
 2 204.12.1.254 32 msec * 32 msec
```


5.10 EIGRP Summarization with Leak Map

- Configure a leak-map on R4 so that traffic going to R4's Loopback0 network is routed out the point-to-point link between R4 and R5.
- If this link is down traffic should still be rerouted out the Frame Relay connection between these devices.

Configuration

```
R4:
interface Serial0/1
 ip summary-address eigrp 100 0.0.0.0 0.0.0.0 5 leak-map LEAK_LOOPBACK0
!
ip prefix-list LOOPBACK0 seq 5 permit 150.1.4.0/24
!
route-map LEAK_LOOPBACK0 permit 10
 match ip address prefix-list LOOPBACK0
```

Verification

Note

The EIGRP **leak-map** feature of the **summary-address** allows the advertisement of specific subnets encompassed by the interface level summary, similar to the **unsuppress-map** feature of BGP aggregation. Routes match in the leak-map route-map will be advertised in addition to the summary. If the route-map matches all routes, all subnets of the aggregate will be advertised in addition to the aggregate. This is useful in cases where you want to originate a default route with the interface summary-address, but don't want to stop the advertisement of any subnets.

In this particular design the leak-map is used to enforce longest match routing traffic engineering. Since R5 has a longer match for the prefix 150.1.4.0/24 via the Serial0/1 interface, traffic for this prefix will never get routed over the Frame Relay network unless the point-to-point link is down.

```
Rack1R5#show ip route | include via 155.1.(0|45).4
D      150.1.4.0 [90/2297856] via 155.1.45.4, 00:00:04, Serial0/1
D*    0.0.0.0/0 [90/2172416] via 155.1.45.4, 00:00:04, Serial0/1
      [90/2172416] via 155.1.0.4, 00:00:04, Serial0/0
```

```
Rack1R5#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.45.4 16 msec * 12 msec
```

```
Rack1R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface Serial0/1
Rack1R4(config-if)#shutdown
Rack1R4(config-if)#
```

```
Rack1R5#show ip route | include via 155.1.(0|45).4
D*    0.0.0.0/0 [90/2172416] via 155.1.0.4, 00:00:03, Serial0/0
```

```
Rack1R5#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.0.4 32 msec * 28 msec
```

5.11 EIGRP Floating Summarization

- Shutdown the point-to-point link between R4 and R5.
- Configure R5 to summarize the Loopback0 networks of R4 and R5 out to SW2; this route should not overlap any additional networks.
- Configure an equal longest match static route on R5 so that SW2 has reachability to both the Loopback0 networks of R4 and R5.

Configuration

```
R5:
interface FastEthernet0/0
 ip summary-address eigrp 100 150.1.4.0 255.255.254.0 5
!
ip route 150.1.4.0 255.255.254.0 155.1.0.4
```

Verification

Note

When summaries are created in EIGRP, OSPF, and BGP the router automatically installs a route to Null0 to match the summary. This is used to prevent the router from forwarding traffic for destinations inside the summary that it does not have a longer match for. However in certain designs this can be an undesirable behavior. To resolve this EIGRP sets its interface level summaries to have an administrative distance of 5 by default. This means that any other route with a distance of 1 – 4 will take precedence over the summary.

In this particular case before summarization is configured on R5, SW2 has the subnet route 150.1.5.0/24, and a default route to reach 150.1.4.4. This is because R4 is generating a default route and suppressing its subnet advertisements.

```
Rack1SW2#show ip route 150.1.4.4
```

```
% Subnet not in table
```

```
Rack1SW2#show ip route 150.1.5.5
```

```
Routing entry for 150.1.5.0/24
```

```
Known via "eigrp 100", distance 90, metric 130816, type internal
```

```
Redistributing via eigrp 100
```

```
Last update from 155.1.58.5 on Vlan58, 01:19:03 ago
```

```
Routing Descriptor Blocks:
```

```
* 155.1.58.5, from 155.1.58.5, 01:19:03 ago, via Vlan58
```

```
Route metric is 130816, traffic share count is 1
```

```
Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
```

```
Reliability 255/255, minimum MTU 1500 bytes
```

```
Loading 1/255, Hops 1
```

Likewise R5 only has a default route to 150.1.4.4, while 150.1.5.5 is directly connected.

```
Rack1R5#show ip route 150.1.4.4
```

```
% Subnet not in table
```

```
Rack1R5#show ip route 150.1.5.5
```

```
Routing entry for 150.1.5.0/24
```

```
Known via "connected", distance 0, metric 0 (connected, via interface)
```

```
Redistributing via eigrp 100
```

```
Routing Descriptor Blocks:
```

```
* directly connected, via Loopback0
```

```
Route metric is 0, traffic share count is 1
```

Based on this current routing information SW2 has reachability to both of these destinations.

```
Rack1SW2#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
```

```
Tracing the route to 150.1.4.4
```

```
1 155.1.58.5 0 msec 0 msec 0 msec
```

```
2 155.1.0.4 34 msec * 25 msec
```

```
Rack1SW2#traceroute 150.1.5.5
```

```
Type escape sequence to abort.
```

```
Tracing the route to 150.1.5.5
```

```
1 155.1.58.5 0 msec * 0 msec
```

Once R5 advertises the summary 150.1.4.0/23 SW2 loses its more specific route to 150.1.5.0/24, but gains a longer match to 150.1.4.4.

```
Rack1R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
Rack1R5(config)#interface FastEthernet0/0
Rack1R5(config-if)#ip summary-address eigrp 100 150.1.4.0 255.255.254.0
```

```
Rack1SW2#show ip route 150.1.4.4
Routing entry for 150.1.4.0/23
  Known via "eigrp 100", distance 90, metric 130816, type internal
  Redistributing via eigrp 100
  Last update from 155.1.58.5 on Vlan58, 00:00:12 ago
  Routing Descriptor Blocks:
  * 155.1.58.5, from 155.1.58.5, 00:00:12 ago, via Vlan58
    Route metric is 130816, traffic share count is 1
    Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

```
Rack1SW2#show ip route 150.1.5.5
Routing entry for 150.1.4.0/23
  Known via "eigrp 100", distance 90, metric 130816, type internal
  Redistributing via eigrp 100
  Last update from 155.1.58.5 on Vlan58, 00:00:16 ago
  Routing Descriptor Blocks:
  * 155.1.58.5, from 155.1.58.5, 00:00:16 ago, via Vlan58
    Route metric is 130816, traffic share count is 1
    Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

Since R5 previously only had a default route to reach 150.1.4.4, the longer match is now the summary to Null0.

```
Rack1R5#show ip route 150.1.4.4
Routing entry for 150.1.4.0/23
  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
```

The longer match for 150.1.5.5 remains the connected interface.

```
Rack1R5#show ip route 150.1.5.5
Routing entry for 150.1.5.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via eigrp 100
  Routing Descriptor Blocks:
  * directly connected, via Loopback0
    Route metric is 0, traffic share count is 1
```

This implies that R5 can forward traffic for 150.1.5.5, but traffic for 150.1.4.4 will be Null routed (dropped).

```
Rack1SW2#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.58.5 0 msec 0 msec 8 msec
 2 155.1.58.5 !H * !H
```

```
Rack1SW2#traceroute 150.1.5.5
```

```
Type escape sequence to abort.
Tracing the route to 150.1.5.5
```

```
 1 155.1.58.5 8 msec * 0 msec
```

To resolve this a static route with a lower administrative distance than the summary is installed in the routing table of R5. This static route tells R5 to forward traffic that matches the summary towards R4.

```
Rack1R5#show ip route | include 150.1.4.0
D 150.1.4.0/23 is a summary, 00:01:14, Null0
```

```
Rack1R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
Rack1R5(config)#ip route 150.1.4.0 255.255.254.0 155.1.0.4
Rack1R5(config)#end
```

```
Rack1R5#show ip route | include 150.1.4.0
S 150.1.4.0/23 [1/0] via 155.1.0.4
```

```
Rack1SW2#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.58.5 0 msec 8 msec 0 msec
 2 155.1.0.4 25 msec * 25 msec
```

5.12 EIGRP Poisoned Floating Summarization

- Remove the previously configured static route on R5.
- Modify the administrative distance of the summary that R5 is generating to SW2 so that a route to Null0 is not installed.

Configuration

```
R5:
interface FastEthernet0/0
 ip summary-address eigrp 100 150.1.4.0 255.255.254.0 255
```

Verification

Note

Routes with an administrative distance of 255 are not candidate to be installed in the routing table. By poisoning the interface level summary on R5 with a distance of 255, the route to Null0 cannot be installed locally in the routing table, but the summary itself can be advertised out the interface. The use of this configuration design-wise is in cases where you want the router to forward traffic for destinations inside the summary that it does not have a longer match for.

In this case we can see that SW2 has the route 150.1.4.0/23 to reach 150.1.4.4. However since R4 is only advertising a default route to R5, R5 has no longer match for 150.1.4.4. In the previous case R5's longer match to 150.1.4.4 was its own summary to Null0, meaning that all traffic going to 150.1.4.4 was dropped. By poisoning the summary with a distance of 255 R5 can now use the default route to reach 150.1.4.4.

```
Rack1SW2#show ip route 150.1.4.4
```

```
Routing entry for 150.1.4.0/23
  Known via "eigrp 100", distance 90, metric 130816, type internal
  Redistributing via eigrp 100
  Last update from 155.1.58.5 on Vlan58, 00:00:46 ago
  Routing Descriptor Blocks:
  * 155.1.58.5, from 155.1.58.5, 00:00:46 ago, via Vlan58
    Route metric is 130816, traffic share count is 1
    Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

```
Rack1R5#show ip route 150.1.4.4
```

```
% Subnet not in table
```

```
Rack1SW2#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.58.5 0 msec 0 msec 9 msec
 2 155.1.0.4 25 msec * 25 msec
```

5.13 EIGRP Metric Weights

- Configure all devices in EIGRP AS 100 so that only delay is used in the composite metric calculation.

Configuration

```
R1 - R6, SW1 - SW4:
router eigrp 100
 metric weights 0 0 0 1 0 0
```

Verification

Note

By default EIGRP uses bandwidth and load to calculate its composite metric. Load and reliability can also be used, or the ratio at which bandwidth and delay are used can be changed, by modifying the **metric weights**. The default weighting of K1 and K3 mean that only bandwidth and delay are used. Specifically the calculation is as follows:

$$\text{metric} = [k1 * \text{bandwidth} + (k2 * \text{bandwidth}) / (256 - \text{load}) + k3 * \text{delay}] * [k5 / (\text{reliability} + k4)]$$

If k5 equals zero the second half of the equation is ignored. Bandwidth is the inverse minimum bandwidth along the path scaled by $2.56 * 10^{12}$. Delay is 10s of microseconds scaled by 256.

The weighting of the metrics can be seen from the **show ip protocols** command.

```
Rack1SW3#show ip protocols
```

```
*** IP Routing is NSF aware ***
```

```
Routing Protocol is "eigrp 100"
```

```
  Outgoing update filter list for all interfaces is not set
```

```
  Incoming update filter list for all interfaces is not set
```

```
  Default networks flagged in outgoing updates
```

```
  Default networks accepted from incoming updates
```

```
  EIGRP metric weight K1=0, K2=0, K3=1, K4=0, K5=0
```

```
  EIGRP maximum hopcount 100
```

```
  EIGRP maximum metric variance 1
```

```
  Redistributing: eigrp 100
```

```
  EIGRP NSF-aware route hold timer is 240s
```

```
  Automatic network summarization is not in effect
```

```
<output omitted>
```


show ip eigrp topology shows the individual vector metrics that are used in the composite calculation.

```
Rack1SW3#show ip eigrp topology 150.1.9.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.9.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
128000
  Routing Descriptor Blocks:
  0.0.0.0 (Loopback0), from Connected, Send flag is 0x0
    Composite metric is (128000/0), Route is Internal
    Vector metric:
      Minimum bandwidth is 10000000 Kbit
      Total delay is 5000 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1514
      Hop count is 0
```

For the connected Loopback0 network of SW3 the total delay is 5000 microseconds. 500 tens of microseconds scaled by 256 equals the total composite metric of 128,000. This indicates that only delay is weighted in the calculation.

Pitfall

The metric weights must match in order for EIGRP adjacency to form.

```
Rack1SW3#config t
Enter configuration commands, one per line. End with CNTL/Z.
Rack1SW3(config)#router eigrp 100
Rack1SW3(config-router)#metric weights 0 1 1 1 1 1
Rack1SW3(config-router)#end
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.79.7 (Vlan79) is
down: metric changed
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.79.7 (Vlan79) is
down: K-value mismatch
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.79.7 (Vlan79) is
down: K-value mismatch
```

5.14 EIGRP Traffic Engineering with Metric

- Configure a metric manipulation on SW1 so that traffic from SW3 to the Loopback0 network of R6 transits the link between R3 and R1.

Configuration

```
SW1:
interface Vlan67
  delay 100000
```

Verification

Note

Before any metric manipulation, SW3's traffic to R6 is sent to SW1, then directly to R6.

```
Rack1SW3#traceroute 150.1.6.6
```

```
Type escape sequence to abort.
Tracing the route to 150.1.6.6
```

```
 1 155.1.79.7 4 msec 0 msec 0 msec
 2 155.1.67.6 4 msec * 0 msec
```

This is based on the fact that SW1 installs the route to 150.1.6.6 via 155.1.67.6.

```
Rack1SW1#show ip route 150.1.6.6
```

```
Routing entry for 150.1.6.0/24
  Known via "eigrp 100", distance 90, metric 128256, type internal
  Redistributing via eigrp 100
  Last update from 155.1.67.6 on Vlan67, 00:14:50 ago
  Routing Descriptor Blocks:
  * 155.1.67.6, from 155.1.67.6, 00:14:50 ago, via Vlan67
    Route metric is 128256, traffic share count is 1
    Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

Also note that SW1 does not know the alternate route through R3.

```
Rack1SW1#show ip eigrp topology 150.1.6.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.6.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
    155.1.67.6 (Vlan67), from 155.1.67.6, Send flag is 0x0
      Composite metric is (128256/128000), Route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 5010 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

This is due to the fact that R3 is choosing SW1's route as the successor. Since only the successor is candidate to be advertised, and since split-horizon is enabled on R3's link to SW1, this route will not advertise this route back to SW1.

```
Rack1R3#show ip eigrp topology 150.1.6.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.6.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 130816
  Routing Descriptor Blocks:
    155.1.37.7 (FastEthernet0/0), from 155.1.37.7, Send flag is 0x0
      Composite metric is (130816/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 5110 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
    155.1.0.5 (Serial1/0.1), from 155.1.0.5, Send flag is 0x0
      Composite metric is (1154560/642560), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 3
    155.1.13.1 (Serial1/2), from 155.1.13.1, Send flag is 0x0
      Composite metric is (642560/130560), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 25100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

In order for R3 to advertise the alternate path to SW1, R3 must see a better composite metric through R1 than it does through SW1. This can be accomplished by altering the advertised distance of the route from SW1 to R3 by changing the delay.

```
Rack1SW1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#interface Vlan 67
Rack1SW1(config-if)#delay 100000
Rack1SW1(config-if)#end
```

The EIGRP neighbors must then be cleared to recalculate DUAL.

```
Rack1SW1#clear ip eigrp neighbors
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.37.3
(FastEthernet0/3) is down: manually cleared
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.67.6 (Vlan67) is
down: manually cleared
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.79.9 (Vlan79) is
down: manually cleared
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.79.9 (Vlan79) is up:
new adjacency
```

SW1 now chooses R3's route as the successor, as the composite result 645,120 is lower than 25,728,000.

```
Rack1SW1#show ip eigrp topology 150.1.6.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.6.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 645120
  Routing Descriptor Blocks:
  155.1.37.3 (FastEthernet0/3), from 155.1.37.3, Send flag is 0x0
    Composite metric is (645120/642560), Route is Internal
  Vector metric:
    Minimum bandwidth is 128 Kbit
    Total delay is 25200 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 3
  155.1.67.6 (Vlan67), from 155.1.67.6, Send flag is 0x0
    Composite metric is (25728000/128000), Route is Internal
  Vector metric:
    Minimum bandwidth is 1000000 Kbit
    Total delay is 1005000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
```

SW1 now routes through R3 to reach 150.1.6.6, which is reflected in both the routing table output of SW1 and the traceroute output of SW3.

```
Rack1SW1#show ip route 150.1.6.6
```

```
Routing entry for 150.1.6.0/24
```

```
Known via "eigrp 100", distance 90, metric 645120, type internal
```

```
Redistributing via eigrp 100
```

```
Last update from 155.1.37.3 on FastEthernet0/3, 00:02:14 ago
```

```
Routing Descriptor Blocks:
```

```
* 155.1.37.3, from 155.1.37.3, 00:02:14 ago, via FastEthernet0/3
```

```
Route metric is 645120, traffic share count is 1
```

```
Total delay is 25200 microseconds, minimum bandwidth is 128 Kbit
```

```
Reliability 255/255, minimum MTU 1500 bytes
```

```
Loading 1/255, Hops 3
```

```
Rack1SW3#traceroute 150.1.6.6
```

```
Type escape sequence to abort.
```

```
Tracing the route to 150.1.6.6
```

```
1 155.1.79.7 0 msec 0 msec 0 msec
```

```
2 155.1.37.3 4 msec 0 msec 0 msec
```

```
3 155.1.13.1 4 msec 8 msec 8 msec
```

```
4 155.1.146.6 8 msec * 4 msec
```

5.15 EIGRP Unequal Cost Load Balancing

- Configure unequal cost load balancing so that traffic from R6 going to VLAN 9 is load balanced between R1 and SW1.
- The traffic share should be configured in such a way that the link to SW1 is used five times as much as the link to R1.
- Verify this by configuring per-packet load balancing on R6.

Configuration

```
R1:
interface Serial0/1
  delay 1

R3:
interface FastEthernet0/0
  delay 1

R6:
interface FastEthernet0/0.146
  delay 56
!
router eigrp 100
  variance 128
```

Verification

Note

Previously the `metric weights` command was updated on all devices in EIGRP AS 100 so that only delay was weighted. Therefore based on the interface delay values from R6 outbound towards VLAN 9 we can calculate how traffic will be routed. Recall that the delay value used in the composite calculation is *tens of microseconds scaled by 256*.

To start, without any configuration changes, the path from R6 to SW3 has the following delays.

```
Rack1R6#show interface FastEthernet0/0.67 | include DLY
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
Rack1SW1#show interface Vlan79 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

```
Rack1SW3#show interface Vlan9 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

The path from R6 -> SW1 -> SW3 therefore has a total delay of 120 microseconds. *12 tens of microseconds* scaled by 256 gives us a composite metric of 3,072.

This path is then compared to the one R6 -> R1 -> R3 -> SW1 -> SW3 with the following delays.

```
Rack1R6#show interface FastEthernet0/0.146 | include DLY
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
Rack1R1#show interface Serial0/1 | include DLY
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
```

```
Rack1R3#show interface FastEthernet0/0 | include DLY
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
Rack1SW1#show interface Vlan79 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

```
Rack1SW3#show interface Vlan9 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

This path has a total delay of 20,220 microseconds. *2,022 tens of microseconds* scaled by 256 gives us a composite metric of 517,632. Since 3,072 is lower than 517,632, the Successor is the route from R6 to SW1. This can be verified from the topology view of R6.

```

Rack1R6#show ip eigrp topology 155.1.9.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 155.1.9.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3072
  Routing Descriptor Blocks:
    155.1.67.7 (FastEthernet0/0.67), from 155.1.67.7, Send flag is 0x0
      Composite metric is (3072/512), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 120 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2

```

In order to consider the route from R6 to R1 for load balancing, the route first must pass the Feasibility Condition. Again the Feasibility Condition states that if the Advertised Distance of an alternate route is lower than the Feasible Distance of the Successor, the route is a loop free path and can be considered for load balancing. In other words if R1's metric to reach SW3 is lower than R6's metric to reach SW3, R6 can assume that R1 is closer to SW3, and is a loop free path.

The Advertised Distance that R1 would be sending to R6 is based on these interfaces in the transit path.

```

Rack1R1#show interface Serial0/1 | include DLY
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

```

```

Rack1R3#show interface FastEthernet0/0 | include DLY
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

```

```

Rack1SW1#show interface Vlan79 | include DLY
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,

```

```

Rack1SW3#show interface Vlan9 | include DLY
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,

```

The total delay of this path is 20,120 microseconds, or 2,012 *tens of microseconds*. Scaled by 256 R1 would be advertising 515,072. Since 515,072 is greater than 3,072, R6's Feasible Distance, this path cannot be considered a Feasible Successor. Therefore the first step in doing unequal cost load balancing is to lower what R1 is advertising as its metric. In this example this is accomplished by changing the delay of R1's link to R3 and R3's link to SW1 to 10 microseconds.


```
Rack1R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R1(config)#interface Serial0/1
Rack1R1(config-if)#delay 1
Rack1R1(config-if)#
```

```
Rack1R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R3(config)#interface FastEthernet0/0
Rack1R3(config-if)#delay 1
Rack1R3(config-if)#
```

These new delay values update the entire path as follows.

```
Rack1R6#show interface FastEthernet0/0.146 | include DLY
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
Rack1R1#show interface Serial0/1 | include DLY
MTU 1500 bytes, BW 1544 Kbit, DLY 10 usec,
```

```
Rack1R3#show interface FastEthernet0/0 | include DLY
MTU 1500 bytes, BW 100000 Kbit, DLY 10 usec,
```

```
Rack1SW1#show interface Vlan79 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

```
Rack1SW3#show interface Vlan9 | include DLY
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

The total delay of the path is now 140 microseconds. 14 *tens of microseconds* scaled by 256 equals a total composite of 3,584. Since this is still higher than 3,072, this path is not the Successor. However R1's Advertised Distance is now a total delay of 40 microseconds. 4 *tens of microseconds* scaled by 256 equals a total Advertised Distance of 1024. Since 1,024 is lower than 3,072, R6's Feasible Distance, this route is now a Feasible Successor.

If the variance command were configured on R6 this path would now be installed in the routing table for load balancing. The actual value of variance is arbitrary, as long as the Feasible Distance, 3,072, times the variance is greater than the total composite metric through R1.

```
Rack1R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#router eigrp 100
Rack1R6(config-router)#variance 128
Rack1R6(config-router)#end

Rack1R6#show ip route 155.1.9.9
Routing entry for 155.1.9.0/24
  Known via "eigrp 100", distance 90, metric 3072, type internal
  Redistributing via eigrp 100, eigrp 10
  Advertised by eigrp 10
  Last update from 155.1.146.1 on FastEthernet0/0.146, 00:00:03 ago
  Routing Descriptor Blocks:
    155.1.146.1, from 155.1.146.1, 00:00:03 ago, via FastEthernet0/0.146
      Route metric is 3584, traffic share count is 103
      Total delay is 140 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 4
    * 155.1.67.7, from 155.1.67.7, 00:00:03 ago, via FastEthernet0/0.67
      Route metric is 3072, traffic share count is 120
      Total delay is 120 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

These paths are now balanced 103:120. To achieve the desired 1:5 traffic share, R6's delay on the link to R1 must be updated. The actual values used on R1, R3, and R6 for delay can have multiple valid options as long as two conditions are true. First, the Advertised Distance R1 sends to R6 must be lower than R6's Feasible Distance. Secondly the entire composite result R6 calculates through R1 should be five times the Feasible Distance.

In our case R1's Advertised Distance is 40 microseconds, or 4 *tens of microseconds*. This specifically means the following must be true if we want a traffic share of 1:5.

$$3072 * 5 = (R6_TO_R1_DLY + 4) * 256$$

Therefore R6's delay to R1 should be 56 *tens of microseconds*.

```

Rack1R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface FastEthernet0/0.146
Rack1R6(config-subif)#delay 56
Rack1R6(config-subif)#end

Rack1R6#show ip route 155.1.9.9
Routing entry for 155.1.9.0/24
  Known via "eigrp 100", distance 90, metric 3072, type internal
  Redistributing via eigrp 100, eigrp 10
  Advertised by eigrp 10
  Last update from 155.1.146.1 on FastEthernet0/0.146, 00:00:05 ago
  Routing Descriptor Blocks:
    155.1.146.1, from 155.1.146.1, 00:00:05 ago, via FastEthernet0/0.146
      Route metric is 15360, traffic share count is 1
      Total delay is 600 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 4
    * 155.1.67.7, from 155.1.67.7, 00:00:05 ago, via FastEthernet0/0.67
      Route metric is 3072, traffic share count is 5
      Total delay is 120 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2

```

To test that traffic is actually sent in this distribution CEF is disabled on R6, and traffic is load balanced per packet. Next access-lists are used inbound on R1 and SW1 to count how many ICMP packets they receive from R6 going to VLAN 9.

```

Rack1R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface FastEthernet0/0.67
Rack1R6(config-subif)#no ip route-cache
Rack1R6(config-subif)#ip load-sharing per-packet
Rack1R6(config-subif)#interface FastEthernet0/0.146
Rack1R6(config-subif)#no ip route-cache
Rack1R6(config-subif)#ip load-sharing per-packet

```

```

Rack1R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R1(config)#access-list 100 permit icmp any host 155.1.9.9 log
Rack1R1(config)#access-list 100 permit ip any any
Rack1R1(config)#interface FastEthernet0/0
Rack1R1(config-if)#ip access-group 100 in

```

```

Rack1SW1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#access-list 100 permit icmp any host 155.1.9.9 log
Rack1SW1(config)#access-list 100 permit ip any any
Rack1SW1(config)#interface Vlan67
Rack1SW1(config-if)#ip access-group 100 in

```


5.16 EIGRP Convergence Timers

- Configure R1 through R6 so that EIGRP hellos are sent every one second; these devices should inform their neighbors to declare them down if subsequent hellos are not received within three seconds.
- Configure SW1 through SW4 so that EIGRP hellos are sent every ten seconds; these devices should inform their neighbors to declare them down if subsequent hellos are not received within thirty seconds.
- Additionally configure AS 100 so that lost routes are considered Stuck In Active if a query response has not been heard within one minute.

Configuration

```
R1:
interface FastEthernet0/0
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/0.1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
router eigrp 100
 timers active-time 1
```

```
R2:
interface FastEthernet0/0
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/0.1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
router eigrp 100
 timers active-time 1
```

```
R3:
interface FastEthernet0/0
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial1/0.1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial1/2
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial1/3
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
router eigrp 100
 timers active-time 1
```

```
R4:
interface FastEthernet0/1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/0.1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
router eigrp 100
 timers active-time 1
```

```
R5:
interface FastEthernet0/0
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/0
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface Serial0/1
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
```

```
R6:
interface FastEthernet0/0.67
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
interface FastEthernet0/0.146
 ip hello-interval eigrp 100 1
 ip hold-time eigrp 100 3
!
router eigrp 100
 timers active-time 1

SW1:
interface FastEthernet0/3
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
interface Vlan67
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
interface Vlan79
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
router eigrp 100
 timers active-time 1

SW2:
interface Vlan58
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
interface Port-channel1
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
router eigrp 100
 timers active-time 1

SW3:
interface Vlan79
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
router eigrp 100
 timers active-time 1

SW4:
interface Port-channel1
 ip hello-interval eigrp 100 10
 ip hold-time eigrp 100 30
!
router eigrp 100
 timers active-time 1
```

Verification **Note**

Unlike OSPF, EIGRP hello intervals do not need to match in order to form adjacency. Instead, the neighbor sending the hello packet tells the adjacent router what its hold time is for that particular hello.

In this case R3 has its hello and dead intervals configured as 1 and 3, while SW1 has them configured as 10 and 30. This means that R3 will be expecting a hello to come in from SW1 within 3 seconds, while R3 will be expecting a hello to come in from SW1 within 30 seconds. In most designs the hello and dead intervals will be set identical on both ends of the link, however as we can see from this example it is not technically required.

Rack1R3#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|------------|----------|-----------|------|-------|---------|
| 0 | 155.1.0.5 | Se1/0.1 | 2 | 00:40:50 | 422 | 2532 | 0 | 343 |
| 1 | 155.1.13.1 | Se1/2 | 2 | 00:42:22 | 41 | 1140 | 0 | 208 |
| 3 | 155.1.23.2 | Se1/3 | 2 | 00:45:58 | 101 | 1140 | 0 | 133 |
| 2 | 155.1.37.7 | Fa0/0 | 23 | 00:46:01 | 3 | 200 | 0 | 194 |

Rack1SW1#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|---|------------|-----------|------------|----------|-----------|-----|-------|---------|
| 2 | 155.1.67.6 | Vl67 | 2 | 00:00:53 | 17 | 200 | 0 | 162 |
| 0 | 155.1.79.9 | Vl79 | 28 | 00:46:06 | 3 | 200 | 0 | 62 |
| 1 | 155.1.37.3 | Fa0/3 | 2 | 00:46:08 | 1 | 200 | 0 | 248 |

The `timers active-time` command controls how long an EIGRP router will wait for a reply to a query message before considering the route Stuck In Active (SIA), and declaring the neighbor down that a reply was not received from.

The query and reply process is used to discover alternate paths to a route for which the successor is lost. In the below case SW4 loses the successor for 155.1.10.0/24 when the Vlan10 interface is shutdown. This causes it to send an EIGRP query message out to its neighbor, SW2.

```
Rack1SW3#debug eigrp packet terse
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
  SIAREPLY)

Rack1SW4#debug eigrp packet terse
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
  SIAREPLY)
Rack1SW4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW4(config)#interface Vlan10
Rack1SW4(config-if)#shutdown
Rack1SW4(config-if)#
Jul 10 10:22:16: EIGRP: Enqueueing QUERY on Port-channell iidbQ un/rely 0/1
serno 453-453
Jul 10 10:22:16: EIGRP: Sending QUERY on Port-channell
Jul 10 10:22:16: AS 100, Flags 0x0, Seq 69/0 idbQ 0/0 iidbQ un/rely 0/0 serno
453-453
```

SW2 acknowledges the reception of the query with an ACK to SW4, and the query is continued to be forwarded.

```
Rack1SW4(config-if)#
Jul 10 10:22:16: EIGRP: Received ACK on Port-channell nbr 155.1.108.8
Jul 10 10:22:16: AS 100, Flags 0x0, Seq 0/69 idbQ 0/0 iidbQ un/rely 0/0 peerQ
un/rely 0/1
Jul 10 10:22:16: EIGRP: Port-channell multicast flow blocking cleared
```

Within one second the query reaches the far end of the network at SW3. SW3 then acknowledges to SW1 that it received the query.

```
Rack1SW3#
Jul 10 10:22:17: EIGRP: Received QUERY on Vlan79 nbr 155.1.79.7
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 209/0 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/0
Jul 10 10:22:17: EIGRP: Enqueueing ACK on Vlan79 nbr 155.1.79.7
Jul 10 10:22:17: Ack seq 209 iidbQ un/rely 0/0 peerQ un/rely 1/0
Jul 10 10:22:17: EIGRP: Sending ACK on Vlan79 nbr 155.1.79.7
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 0/209 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 1/0
```

Since SW3 does not have any other neighbors to send the query to, and it does not have an alternate route to 155.1.10.0/24, it replies to SW1 telling it that it does not have another path. SW1 then acknowledges the query reply to SW3, and sends its own replies back to its other neighbors.

```
Rack1SW3#
Jul 10 10:22:17: EIGRP: Enqueueing REPLY on Vlan79 nbr 155.1.79.7 iidbQ un/rely
0/1 peerQ un/rely 0/0 serno 542-542
Jul 10 10:22:17: EIGRP: Requeued unicast on Vlan79
Jul 10 10:22:17: EIGRP: Sending REPLY on Vlan79 nbr 155.1.79.7
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 66/209 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/1 serno 542-542
Jul 10 10:22:17: EIGRP: Received ACK on Vlan79 nbr 155.1.79.7
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 0/66 idbQ 0/0 iidbQ un/rely 0/0 peerQ
un/rely 0/1
```

Within two seconds the entire query process is completed on SW4 with the reply coming back from SW2. Incredibly the query and replies are received even before the link up/down message is generated, indicating the immensely fast convergence capability of EIGRP. If a reply had not come back from SW2, SW4 would wait for the **timers active-time** to expire. If this timer had expired the route would have been considered SIA, and the neighbor relationship to SW2 would have been reset.

```
Rack1SW4(config-if)#
Jul 10 10:22:17: EIGRP: Received REPLY on Port-channell nbr 155.1.108.8
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 131/69 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/0
Jul 10 10:22:17: EIGRP: Enqueueing ACK on Port-channell nbr 155.1.108.8
Jul 10 10:22:17: Ack seq 131 iidbQ un/rely 0/0 peerQ un/rely 1/0
Jul 10 10:22:17: EIGRP: Sending ACK on Port-channell nbr 155.1.108.8
Jul 10 10:22:17: AS 100, Flags 0x0, Seq 0/131 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 1/0
Jul 10 10:22:18: %LINK-5-CHANGED: Interface Vlan10, changed state to
administratively down
Jul 10 10:22:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,
changed state to down
```

5.17 EIGRP Stub Routing

- Configure the EIGRP stub feature in such a way that SW2 does not receive EIGRP query messages.
- Ensure that all devices in AS 100 still have IP reachability to VLAN 8.

Configuration

```
SW2:
router eigrp 100
  eigrp stub connected
```

Verification

Note

The EIGRP stub feature is used to limit the scope of EIGRP query messages, and to limit what routes a neighbor advertises.

Rack1R5#show ip eigrp neighbors detail

```
IP-EIGRP neighbors for process 100
H   Address                Interface          Hold Uptime    SRTT   RTO  Q  Seq
                               (sec)           (ms)          Cnt  Num
 4   155.1.58.8              Fa0/0             27 00:19:42    6    200  0  172
    Version 12.2/1.2, Retrans: 4, Retries: 0, Prefixes: 3
    Stub Peer Advertising ( CONNECTED ) Routes
    Suppressing queries
 5   155.1.45.4              Se0/1             225 02:10:18   22    200  0  355
    Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 33
 3   155.1.0.4               Se0/0              2 02:41:35    41    246  0  356
    Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 33
 2   155.1.0.2               Se0/0              2 02:41:35    50    300  0  189
    Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 13
 1   155.1.0.3               Se0/0              2 02:41:35    55    330  0  346
    Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 32
 0   155.1.0.1               Se0/0              2 02:41:35    44    264  0  295
    Version 12.4/1.2, Retrans: 1, Retries: 0, Prefixes: 33
```

In this case SW2 is configured to only advertise its connected routes to other EIGRP neighbors. This implies that SW4 will not have reachability to any destinations behind SW2, and destinations behind SW2 will not have reachability to SW4.

```
Rack1SW4#show ip route eigrp
 155.1.0.0/24 is subnetted, 4 subnets
D    155.1.8.0 [90/2816] via 155.1.108.8, 00:02:00, Port-channel1
D    155.1.58.0 [90/2816] via 155.1.108.8, 00:02:00, Port-channel1
 150.1.0.0/24 is subnetted, 2 subnets
D    150.1.8.0 [90/130560] via 155.1.108.8, 00:02:00, Port-channel1
```

```
Rack1R5#show ip route | include via 155.1.58.8
D    155.1.8.0 [90/2816] via 155.1.58.8, 00:02:28, FastEthernet0/0
D    155.1.108.0 [90/5120] via 155.1.58.8, 00:02:28, FastEthernet0/0
D    150.1.8.0 [90/130560] via 155.1.58.8, 00:02:28, FastEthernet0/0
```

Output from the `debug eigrp packet terse` shows the progression of a QUERY message in the network and its REPLY. First, SW3 shuts down its Vlan9 interface, withdrawing 155.1.9.0/24 and generating a QUERY.

```
Rack1SW3#config t
Enter configuration commands, one per line. End with CNTL/Z.
Rack1SW3(config)#interface Vlan9
Rack1SW3(config-if)#shutdown
Rack1SW3(config-if)#
EIGRP: Enqueueing QUERY on Vlan79 idbQ un/rely 0/1 serno 574-574
EIGRP: Sending QUERY on Vlan79
  AS 100, Flags 0x0, Seq 84/0 idbQ 0/0 iidbQ un/rely 0/0 serno 574-574
```

R5 receives the QUERY in from R1, R2, R3, and R4.

```
Rack1R5#
EIGRP: Received QUERY on Serial0/0 nbr 155.1.0.3
  AS 100, Flags 0x0, Seq 344/437 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Received QUERY on Serial0/0 nbr 155.1.0.2
  AS 100, Flags 0x0, Seq 187/437 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Received QUERY on Serial0/1 nbr 155.1.45.4
  AS 100, Flags 0x0, Seq 353/438 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
EIGRP: Received QUERY on Serial0/0 nbr 155.1.0.1
  AS 100, Flags 0x0, Seq 292/437 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Received QUERY on Serial0/0 nbr 155.1.0.4
  AS 100, Flags 0x0, Seq 352/437 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
```

QUERY is forwarded on to all neighbors except the stub neighbor, SW2.

```
Rack1R5#
EIGRP: Sending QUERY on Serial0/0 nbr 155.1.0.4
  AS 100, Flags 0x0, Seq 449/352 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
serno 1262-1263
EIGRP: Sending QUERY on Serial0/0 nbr 155.1.0.2
  AS 100, Flags 0x0, Seq 449/187 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
serno 1262-1263
EIGRP: Sending QUERY on Serial0/0 nbr 155.1.0.3
  AS 100, Flags 0x0, Seq 449/344 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
serno 1262-1263
EIGRP: Sending QUERY on Serial0/1 nbr 155.1.45.4
  AS 100, Flags 0x0, Seq 445/353 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
serno 1262-1263
EIGRP: Sending QUERY on Serial0/0 nbr 155.1.0.1
  AS 100, Flags 0x0, Seq 449/292 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
serno 1262-1263
```

5.18 EIGRP Stub Routing with Leak Map

- Configure the EIGRP stub feature in such a way that R5 does not receive EIGRP query messages.
- R5 should continue to advertise all learned routes with the exception of SW2's Loopback0 network.

Configuration

```
R5:
ip prefix-list SW2_LOOPBACK seq 5 permit 150.1.8.0/24
!
route-map STUB_LEAK_MAP deny 10
  match ip address prefix-list SW2_LOOPBACK
!
route-map STUB_LEAK_MAP permit 20
!
router eigrp 100
  eigrp stub connected leak-map STUB_LEAK_MAP
```

Verification

Note

The `leak-map` feature of EIGRP stub, like the leak-map for EIGRP summarization, allows the advertisement of routes that would normally be suppressed. When R5 is configured with only the `eigrp stub` command, it cannot be used as transit. This can be seen from the routing table views of SW2 and R3.

```
Rack1R3#show ip route | include via 155.1.0.5
```

```
D      155.1.5.0 [90/514560] via 155.1.0.5, 00:00:16, Serial1/0.1
D      155.1.58.0 [90/514560] via 155.1.0.5, 00:00:16, Serial1/0.1
D      155.1.45.0 [90/1024000] via 155.1.0.5, 00:00:17, Serial1/0.1
D      150.1.5.0 [90/640000] via 155.1.0.5, 00:00:16, Serial1/0.1
```

```
Rack1SW2#show ip route | include via 155.1.58.5
```

```
D      155.1.0.0 [90/512256] via 155.1.58.5, 00:00:19, Vlan58
D      155.1.5.0 [90/2816] via 155.1.58.5, 00:00:19, Vlan58
D      155.1.45.0 [90/512256] via 155.1.58.5, 00:00:19, Vlan58
D      150.1.5.0 [90/128256] via 155.1.58.5, 00:00:19, Vlan58
```

In this design R5 is configured to leak all dynamically learned routes with the exception of SW2's Loopback. If a failure in the network occurs however, R5 will not receive EIGRP QUERY messages.

Rack1R3#show ip eigrp neighbors detail

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold (sec) | Uptime | SRTT (ms) | RTO | Q Cnt | Seq Num |
|--|------------|-----------|------------|----------|-----------|------|-------|---------|
| 0 | 155.1.0.5 | Se1/0.1 | 2 | 00:00:05 | 911 | 5000 | 0 | 626 |
| Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 37 | | | | | | | | |
| Stub Peer Advertising (CONNECTED) Routes | | | | | | | | |
| Suppressing queries | | | | | | | | |
| 1 | 155.1.13.1 | Se1/2 | 2 | 02:57:20 | 27 | 1140 | 0 | 383 |
| Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 24 | | | | | | | | |
| 3 | 155.1.23.2 | Se1/3 | 2 | 03:00:57 | 30 | 1140 | 0 | 285 |
| Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 20 | | | | | | | | |
| 2 | 155.1.37.7 | Fa0/0 | 29 | 03:01:00 | 1 | 200 | 0 | 407 |
| Version 12.2/1.2, Retrans: 5, Retries: 0, Prefixes: 5 | | | | | | | | |

Rack1R3#

Rack1R3#show ip route | include via 155.1.0.5

```
D 155.1.8.0 [90/514816] via 155.1.0.5, 00:00:00, Serial1/0.1
D 155.1.5.0 [90/514560] via 155.1.0.5, 00:00:02, Serial1/0.1
D 155.1.58.0 [90/514560] via 155.1.0.5, 00:00:02, Serial1/0.1
D 155.1.45.0 [90/1024000] via 155.1.0.5, 00:00:02, Serial1/0.1
D 155.1.108.0 [90/517120] via 155.1.0.5, 00:00:00, Serial1/0.1
D 150.1.5.0 [90/640000] via 155.1.0.5, 00:00:02, Serial1/0.1
```

Rack1R3#show ip route 150.1.8.8

% Subnet not in table

Rack1SW2#show ip eigrp neighbors detail

IP-EIGRP neighbors for process 100

| H | Address | Interface | Hold Uptime (sec) | SRTT (ms) | RTO | Q Cnt | Seq Num | Type |
|--|--------------|-----------|----------------------|--------------|-----|----------|------------|------|
| 0 | 155.1.58.5 | Vl58 | 2 00:00:25 | 1 | 200 | 0 | 633 | |
| Version 12.4/1.2, Retrans: 3, Retries: 0 | | | | | | | | |
| Stub Peer Advertising (CONNECTED REDISTRIBUTED) Routes | | | | | | | | |
| Suppressing queries | | | | | | | | |
| 1 | 155.1.108.10 | Pol | 20 00:34:21 | 1 | 450 | 0 | 90 | |
| Version 12.2/1.2, Retrans: 12, Retries: 0 | | | | | | | | |

Rack1SW2#show ip route | include via 155.1.58.5

```

D EX 222.22.2.0/24 [170/537856] via 155.1.58.5, 00:00:18, Vlan58
D EX 204.12.1.0/24 [170/537856] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.146.0 [90/514816] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.23.0 [90/1024256] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.13.0 [90/512512] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.0.0 [90/512256] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.7.0 [90/512768] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.5.0 [90/2816] via 155.1.58.5, 00:00:18, Vlan58
D    155.1.45.0 [90/512256] via 155.1.58.5, 00:00:19, Vlan58
D    155.1.37.0 [90/512512] via 155.1.58.5, 00:00:19, Vlan58
D    155.1.79.0 [90/512768] via 155.1.58.5, 00:00:19, Vlan58
D    155.1.67.0 [90/517376] via 155.1.58.5, 00:00:14, Vlan58
D EX 220.20.3.0/24 [170/537856] via 155.1.58.5, 00:00:19, Vlan58
D EX 200.0.0.0/24 [170/1154816] via 155.1.58.5, 00:00:14, Vlan58
D EX   54.1.1.0 [170/1026816] via 155.1.58.5, 00:00:14, Vlan58
D EX 200.0.1.0/24 [170/1154816] via 155.1.58.5, 00:00:14, Vlan58
D EX 200.0.2.0/24 [170/1154816] via 155.1.58.5, 00:00:14, Vlan58
D EX 200.0.3.0/24 [170/1154816] via 155.1.58.5, 00:00:14, Vlan58
D EX 192.10.1.0/24 [170/537856] via 155.1.58.5, 00:00:19, Vlan58
D EX   31.3.0.0 [170/537856] via 155.1.58.5, 00:00:14, Vlan58
D EX   31.2.0.0 [170/537856] via 155.1.58.5, 00:00:14, Vlan58
D EX   31.1.0.0 [170/537856] via 155.1.58.5, 00:00:14, Vlan58
D EX   31.0.0.0 [170/537856] via 155.1.58.5, 00:00:14, Vlan58
D    150.1.7.0 [90/640512] via 155.1.58.5, 00:00:21, Vlan58
D    150.1.6.0 [90/642816] via 155.1.58.5, 00:00:16, Vlan58
D    150.1.5.0 [90/128256] via 155.1.58.5, 00:00:21, Vlan58
D    150.1.4.0 [90/640256] via 155.1.58.5, 00:00:16, Vlan58
D    150.1.3.0 [90/640256] via 155.1.58.5, 00:00:21, Vlan58
D    150.1.2.0 [90/640256] via 155.1.58.5, 00:00:21, Vlan58
D    150.1.1.0 [90/640256] via 155.1.58.5, 00:00:21, Vlan58
D    150.1.9.0 [90/640768] via 155.1.58.5, 00:00:21, Vlan58
D EX 205.90.31.0/24 [170/537856] via 155.1.58.5, 00:00:21, Vlan58
D EX   30.2.0.0 [170/537856] via 155.1.58.5, 00:00:21, Vlan58
D EX   30.3.0.0 [170/537856] via 155.1.58.5, 00:00:21, Vlan58
D EX   30.0.0.0 [170/537856] via 155.1.58.5, 00:00:21, Vlan58
D EX   30.1.0.0 [170/537856] via 155.1.58.5, 00:00:21, Vlan58

```


5.19 EIGRP Filtering with Passive Interface

- Configure the passive-interface feature on R5, SW2, and SW4 so that EIGRP hello packets are not sent out the LAN segments without routers attached.
- Configure the passive-interface default feature on SW1 and SW3 so that EIGRP hello packets are not sent out the LAN segments without routers attached; ensure that full reachability is maintained after this change is made.

Configuration

R5:

```
router eigrp 100
  passive-interface FastEthernet0/1
```

SW1:

```
router eigrp 100
  passive-interface default
  no passive-interface Vlan67
  no passive-interface Vlan79
  no passive-interface FastEthernet0/3
```

SW2:

```
router eigrp 100
  passive-interface Vlan8
```

SW3:

```
router eigrp 100
  passive-interface default
  no passive-interface Vlan79
```

SW4:

```
router eigrp 100
  passive-interface Vlan10
```

Verification

Note

The **passive-interface** command in EIGRP, like in RIPv2, stops the sending of updates out an interface. Unlike RIPv2 however, passive-interface in EIGRP will stop the forming of an adjacency on the interface, and hence the learning of any updates on the link.

The **passive-interface default** command can be used to make all interfaces passive, and then interfaces can have the passive feature selectively disabled with the no passive-interface command.

Rack1SW1#show ip protocols

*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=0, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 100

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

150.1.0.0

155.1.0.0

Passive Interface(s):

Vlan1

Vlan7

FastEthernet0/1

FastEthernet0/2

FastEthernet0/4

FastEthernet0/5

FastEthernet0/6

FastEthernet0/7

FastEthernet0/8

FastEthernet0/9

FastEthernet0/10

FastEthernet0/11

FastEthernet0/12

FastEthernet0/13

FastEthernet0/14

FastEthernet0/15

FastEthernet0/16

FastEthernet0/17

FastEthernet0/18

FastEthernet0/19

FastEthernet0/20

FastEthernet0/21

FastEthernet0/22

FastEthernet0/23

FastEthernet0/24

GigabitEthernet0/1

GigabitEthernet0/2

Loopback0

Routing Information Sources:

| Gateway | Distance | Last Update |
|---------|----------|-------------|
|---------|----------|-------------|

| | | |
|------------|----|----------|
| 155.1.37.3 | 90 | 00:01:39 |
|------------|----|----------|

| | | |
|------------|----|----------|
| 155.1.79.9 | 90 | 00:01:39 |
|------------|----|----------|

| | | |
|------------|----|----------|
| 155.1.67.6 | 90 | 00:01:39 |
|------------|----|----------|

Distance: internal 90 external 170

5.20 EIGRP Filtering with Prefix-Lists

- Configure a prefix-list on R4 so that it does not advertise the 30.0.0.0 and 31.0.0.0 subnets learned from BB3 out the point-to-point link to R5; use the most efficient list to accomplish this that will not deny any other networks than those subnets R4 is learning.
- Configure a prefix-list on R1 so that it does not install any updates received from R4 on the VLAN 146 segment.

Configuration

```
R1:
router eigrp 100
  distribute-list prefix PERMIT_ALL gateway NOT_FROM_R4 in
  !
ip prefix-list NOT_FROM_R4 seq 5 deny 155.1.146.4/32
ip prefix-list NOT_FROM_R4 seq 10 permit 0.0.0.0/0 le 32
  !
ip prefix-list PERMIT_ALL seq 5 permit 0.0.0.0/0 le 32

R4:
router eigrp 100
  distribute-list prefix STOP_RIP_SUBNETS out Serial0/1
  !
ip prefix-list STOP_RIP_SUBNETS seq 5 deny 30.0.0.0/14 ge 16 le 16
ip prefix-list STOP_RIP_SUBNETS seq 10 deny 31.0.0.0/14 ge 16 le 16
ip prefix-list STOP_RIP_SUBNETS seq 15 permit 0.0.0.0/0 le 32
```

Verification

Note

Before filtering:

```
Rack1R1#show ip route | include 3(0|1).[0-3].0.0
 31.0.0.0/16 is subnetted, 4 subnets
D EX   31.3.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   31.2.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   31.1.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   31.0.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
 30.0.0.0/16 is subnetted, 4 subnets
D EX   30.2.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   30.3.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   30.0.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
D EX   30.1.0.0 [170/28160] via 155.1.146.4, 00:17:02, FastEthernet0/0
```

```

Rack1R5#show ip route | include via 155.1.(0|45).4
D EX 204.12.1.0/24 [170/537600] via 155.1.45.4, 00:21:37, Serial0/1
    [170/537600] via 155.1.0.4, 00:21:37, Serial0/0
D      155.1.146.0 [90/514560] via 155.1.45.4, 00:21:37, Serial0/1
    [90/514560] via 155.1.0.4, 00:21:37, Serial0/0
D      155.1.67.0 [90/517120] via 155.1.45.4, 00:21:36, Serial0/1
    [90/517120] via 155.1.0.4, 00:21:36, Serial0/0
D EX 200.0.0.0/24 [170/1154560] via 155.1.45.4, 00:21:36, Serial0/1
    [170/1154560] via 155.1.0.4, 00:21:36, Serial0/0
D EX   54.1.1.0 [170/1026560] via 155.1.45.4, 00:21:37, Serial0/1
    [170/1026560] via 155.1.0.4, 00:21:37, Serial0/0
D EX 200.0.1.0/24 [170/1154560] via 155.1.45.4, 00:21:37, Serial0/1
    [170/1154560] via 155.1.0.4, 00:21:37, Serial0/0
D EX 200.0.2.0/24 [170/1154560] via 155.1.45.4, 00:21:37, Serial0/1
    [170/1154560] via 155.1.0.4, 00:21:37, Serial0/0
D EX 200.0.3.0/24 [170/1154560] via 155.1.45.4, 00:21:37, Serial0/1
    [170/1154560] via 155.1.0.4, 00:21:37, Serial0/0
D EX   31.3.0.0 [170/537600] via 155.1.45.4, 00:03:52, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:52, Serial0/0
D EX   31.2.0.0 [170/537600] via 155.1.45.4, 00:03:52, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:52, Serial0/0
D EX   31.1.0.0 [170/537600] via 155.1.45.4, 00:03:52, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:52, Serial0/0
D EX   31.0.0.0 [170/537600] via 155.1.45.4, 00:03:52, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:52, Serial0/0
D      150.1.6.0 [90/642560] via 155.1.45.4, 00:21:40, Serial0/1
    [90/642560] via 155.1.0.4, 00:21:40, Serial0/0
D      150.1.4.0 [90/640000] via 155.1.45.4, 00:21:41, Serial0/1
    [90/640000] via 155.1.0.4, 00:21:41, Serial0/0
D EX   30.2.0.0 [170/537600] via 155.1.45.4, 00:03:55, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:55, Serial0/0
D EX   30.3.0.0 [170/537600] via 155.1.45.4, 00:03:55, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:55, Serial0/0
D EX   30.0.0.0 [170/537600] via 155.1.45.4, 00:03:55, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:55, Serial0/0
D EX   30.1.0.0 [170/537600] via 155.1.45.4, 00:03:55, Serial0/1
    [170/537600] via 155.1.0.4, 00:03:55, Serial0/0

```

After filtering:

```

Rack1R1#show ip route | include 3(0|1).[0-3].0.0
    31.0.0.0/16 is subnetted, 4 subnets
D EX   31.3.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   31.2.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   31.1.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   31.0.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
    30.0.0.0/16 is subnetted, 4 subnets
D EX   30.2.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   30.3.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   30.0.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1
D EX   30.1.0.0 [170/1049600] via 155.1.0.5, 00:00:04, Serial0/0.1

```

```
Rack1R5#show ip route | include via 155.1.(0|45).4
D EX 204.12.1.0/24 [170/537600] via 155.1.45.4, 00:04:25, Serial0/1
    [170/537600] via 155.1.0.4, 00:04:25, Serial0/0
D     155.1.146.0 [90/514560] via 155.1.45.4, 00:04:25, Serial0/1
    [90/514560] via 155.1.0.4, 00:04:25, Serial0/0
D     155.1.67.0 [90/517120] via 155.1.45.4, 00:04:24, Serial0/1
    [90/517120] via 155.1.0.4, 00:04:24, Serial0/0
D EX 200.0.0.0/24 [170/1154560] via 155.1.45.4, 00:04:24, Serial0/1
    [170/1154560] via 155.1.0.4, 00:04:25, Serial0/0
D EX  54.1.1.0 [170/1026560] via 155.1.45.4, 00:04:25, Serial0/1
    [170/1026560] via 155.1.0.4, 00:04:25, Serial0/0
D EX 200.0.1.0/24 [170/1154560] via 155.1.45.4, 00:04:25, Serial0/1
    [170/1154560] via 155.1.0.4, 00:04:25, Serial0/0
D EX 200.0.2.0/24 [170/1154560] via 155.1.45.4, 00:04:25, Serial0/1
    [170/1154560] via 155.1.0.4, 00:04:25, Serial0/0
D EX 200.0.3.0/24 [170/1154560] via 155.1.45.4, 00:04:25, Serial0/1
    [170/1154560] via 155.1.0.4, 00:04:25, Serial0/0
D EX  31.3.0.0 [170/537600] via 155.1.0.4, 00:04:26, Serial0/0
D EX  31.2.0.0 [170/537600] via 155.1.0.4, 00:04:26, Serial0/0
D EX  31.1.0.0 [170/537600] via 155.1.0.4, 00:04:26, Serial0/0
D EX  31.0.0.0 [170/537600] via 155.1.0.4, 00:04:26, Serial0/0
D     150.1.6.0 [90/642560] via 155.1.45.4, 00:04:26, Serial0/1
    [90/642560] via 155.1.0.4, 00:04:26, Serial0/0
D     150.1.4.0 [90/640000] via 155.1.45.4, 00:04:26, Serial0/1
    [90/640000] via 155.1.0.4, 00:04:26, Serial0/0
D EX  30.2.0.0 [170/537600] via 155.1.0.4, 00:04:28, Serial0/0
D EX  30.3.0.0 [170/537600] via 155.1.0.4, 00:04:28, Serial0/0
D EX  30.0.0.0 [170/537600] via 155.1.0.4, 00:04:28, Serial0/0
D EX  30.1.0.0 [170/537600] via 155.1.0.4, 00:04:28, Serial0/0
```

5.21 EIGRP Filtering with Standard Access-Lists

- Configure a one line standard access-list on R6 to filter out all routes coming from BB1 that have an odd number in the third octet.

Configuration

```
R6:
router eigrp 10
 distribute-list 1 in Serial0/0
!
access-list 1 permit 0.0.0.0 255.255.254.255
```

Verification

Note

Before filter.

```
Rack1R6#show ip route eigrp 10
D   200.0.0.0/24 [90/2297856] via 54.1.1.254, 00:00:04, Serial0/0
D   200.0.1.0/24 [90/2297856] via 54.1.1.254, 00:00:04, Serial0/0
D   200.0.2.0/24 [90/2297856] via 54.1.1.254, 00:00:04, Serial0/0
D   200.0.3.0/24 [90/2297856] via 54.1.1.254, 00:00:04, Serial0/0
```

After filter.

```
Rack1R6#show ip route eigrp 10
D   200.0.0.0/24 [90/2297856] via 54.1.1.254, 00:00:03, Serial0/0
D   200.0.2.0/24 [90/2297856] via 54.1.1.254, 00:00:03, Serial0/0
```

5.22 EIGRP Filtering with Extended Access-Lists

- Shutdown R5's point-to-point link to R4.
- Configure an extended access-list filter on R5 so that traffic for the Loopback0 networks of SW1 and SW3 is sent to R1.
- Traffic for the Loopback0 networks of R4 and R6 should be sent to R2.
- Traffic for the Loopback0 networks of R1 and R2 should be sent to R3.
- This filter should not affect any other updates on this segment.

Configuration

```
R5:
access-list 100 deny ip host 155.1.0.2 host 150.1.7.0
access-list 100 deny ip host 155.1.0.3 host 150.1.7.0
access-list 100 deny ip host 155.1.0.4 host 150.1.7.0
access-list 100 deny ip host 155.1.0.2 host 150.1.9.0
access-list 100 deny ip host 155.1.0.3 host 150.1.9.0
access-list 100 deny ip host 155.1.0.4 host 150.1.9.0
access-list 100 deny ip host 155.1.0.1 host 150.1.4.0
access-list 100 deny ip host 155.1.0.3 host 150.1.4.0
access-list 100 deny ip host 155.1.0.4 host 150.1.4.0
access-list 100 deny ip host 155.1.0.1 host 150.1.6.0
access-list 100 deny ip host 155.1.0.3 host 150.1.6.0
access-list 100 deny ip host 155.1.0.4 host 150.1.6.0
access-list 100 deny ip host 155.1.0.1 host 150.1.1.0
access-list 100 deny ip host 155.1.0.2 host 150.1.1.0
access-list 100 deny ip host 155.1.0.4 host 150.1.1.0
access-list 100 deny ip host 155.1.0.1 host 150.1.2.0
access-list 100 deny ip host 155.1.0.2 host 150.1.2.0
access-list 100 deny ip host 155.1.0.4 host 150.1.2.0
access-list 100 permit ip any any
!
router eigrp 100
 distribute-list 100 in Serial0/0
```

Verification

Note

Like RIP, extended access-lists when called as a distribute-list in IGP have a different meaning than in redistribution or as in BGP. With BGP and redistribution the “source” field in the ACL represents the network address, and the “destination” field represents the subnet mask. In IGP distribute-list application the “source” field in the ACL matches the update source of the route, and the “destination” field represents the network address. This implementation allows us to control which networks we are receiving, but more importantly who we are receiving them from. Before the filter is applied, R5 routes as follows.

```
Rack1R5#show ip route eigrp | include 150.1.
  150.1.0.0/24 is subnetted, 9 subnets
D       150.1.7.0 [90/640256] via 155.1.0.3, 00:43:14, Serial0/0
D       150.1.6.0 [90/642560] via 155.1.0.4, 00:00:20, Serial0/0
D       150.1.4.0 [90/640000] via 155.1.0.4, 00:00:20, Serial0/0
D       150.1.3.0 [90/640000] via 155.1.0.3, 01:07:01, Serial0/0
D       150.1.2.0 [90/640000] via 155.1.0.2, 01:07:01, Serial0/0
D       150.1.1.0 [90/640000] via 155.1.0.1, 01:07:01, Serial0/0
D       150.1.9.0 [90/640512] via 155.1.0.3, 00:43:15, Serial0/0
D       150.1.8.0 [90/130560] via 155.1.58.8, 01:46:12, FastEthernet0/0
```

```
Rack1R5#traceroute 150.1.7.7
```

```
Type escape sequence to abort.
Tracing the route to 150.1.7.7
```

```
 1 155.1.0.3 32 msec 28 msec 32 msec
 2 155.1.37.7 28 msec * 28 msec
```

```
Rack1R5#traceroute 150.1.9.9
```

```
Type escape sequence to abort.
Tracing the route to 150.1.9.9
```

```
 1 155.1.0.3 28 msec 28 msec 32 msec
 2 155.1.37.7 28 msec 32 msec 32 msec
 3 155.1.79.9 28 msec * 32 msec
```

```
Rack1R5#traceroute 150.1.4.4
```

```
Type escape sequence to abort.
Tracing the route to 150.1.4.4
```

```
 1 155.1.0.4 32 msec * 28 msec
```

```
Rack1R5#traceroute 150.1.6.6
```

```
Type escape sequence to abort.
Tracing the route to 150.1.6.6
```

```
 1 155.1.0.4 32 msec
   155.1.0.1 28 msec
   155.1.0.4 28 msec
 2 155.1.146.6 28 msec * 28 msec
```

```
Rack1R5#traceroute 150.1.1.1
```

```
Type escape sequence to abort.
Tracing the route to 150.1.1.1
```

```
 1 155.1.0.1 28 msec * 28 msec
```


Rack1R5#traceroute 150.1.2.2

Type escape sequence to abort.
Tracing the route to 150.1.2.2

```
 1 155.1.0.2 32 msec * 28 msec
```

Once the distribute-list is implemented R5 has only one possible way to route to these destinations.

Rack1R5#show ip route eigrp | include 150.1.

```
 150.1.0.0/24 is subnetted, 9 subnets
D    150.1.7.0 [90/640512] via 155.1.0.1, 00:03:20, Serial0/0
D    150.1.6.0 [90/1666560] via 155.1.0.2, 00:03:20, Serial0/0
D    150.1.4.0 [90/26766592] via 155.1.0.2, 00:03:20, Serial0/0
D    150.1.3.0 [90/640000] via 155.1.0.3, 00:03:20, Serial0/0
D    150.1.2.0 [90/1152000] via 155.1.0.3, 00:03:20, Serial0/0
D    150.1.1.0 [90/1152000] via 155.1.0.3, 00:03:20, Serial0/0
D    150.1.9.0 [90/640768] via 155.1.0.1, 00:03:20, Serial0/0
D    150.1.8.0 [90/130560] via 155.1.58.8, 00:03:20, FastEthernet0/0
```

Rack1R5#traceroute 150.1.7.7

Type escape sequence to abort.
Tracing the route to 150.1.7.7

```
 1 155.1.0.1 32 msec 28 msec 28 msec
 2 155.1.13.3 36 msec 32 msec 32 msec
 3 155.1.37.7 36 msec * 32 msec
```

Rack1R5#traceroute 150.1.9.9

Type escape sequence to abort.
Tracing the route to 150.1.9.9

```
 1 155.1.0.1 28 msec 32 msec 93 msec
 2 155.1.13.3 36 msec 32 msec 32 msec
 3 155.1.37.7 36 msec 32 msec 36 msec
 4 155.1.79.9 36 msec * 32 msec
```

Rack1R5#traceroute 150.1.4.4

Type escape sequence to abort.
Tracing the route to 150.1.4.4

```
 1 155.1.0.2 28 msec 28 msec 32 msec
 2 155.1.23.3 32 msec 36 msec 36 msec
 3 155.1.37.7 32 msec 32 msec 32 msec
 4 155.1.67.6 37 msec 36 msec 32 msec
 5 155.1.146.4 36 msec * 32 msec
```

```
Rack1R5#traceroute 150.1.6.6
```

```
Type escape sequence to abort.  
Tracing the route to 150.1.6.6
```

```
 1 155.1.0.2 28 msec 32 msec 28 msec  
 2 155.1.23.3 36 msec 36 msec 32 msec  
 3 155.1.13.1 36 msec 40 msec 40 msec  
 4 155.1.146.6 40 msec * 40 msec
```

```
Rack1R5#traceroute 150.1.1.1
```

```
Type escape sequence to abort.  
Tracing the route to 150.1.1.1
```

```
 1 155.1.0.3 32 msec 28 msec 32 msec  
 2 155.1.13.1 36 msec * 32 msec
```

```
Rack1R5#traceroute 150.1.2.2
```

```
Type escape sequence to abort.  
Tracing the route to 150.1.2.2
```

```
 1 155.1.0.3 28 msec 32 msec 28 msec  
 2 155.1.23.2 41 msec * 32 msec
```

5.23 EIGRP Filtering with Offset Lists

- Configure an offset-list on SW1 so traffic destined for R3's Loopback0 network is sent to R6.
- If the link to R6 is down traffic should be rerouted directly to R3.

Configuration

```
SW1:
router eigrp 100
  offset-list 1 in 2147483647 FastEthernet0/3
!
access-list 1 permit 150.1.3.0
```

Verification

Note

Like in RIP, the offset-list feature in EIGRP is used to modify the metric on a per-route basis or a per-interface basis. Before any metric modifications we can see that SW1 is routing directly to R3 to reach 150.1.3.0/24. There are no additional entries in the EIGRP topology table for this prefix since both R6 and SW3 are routing through SW1 to reach it.

```
Rack1SW1#show ip route 150.1.3.3
```

```
Routing entry for 150.1.3.0/24
  Known via "eigrp 100", distance 90, metric 130560, type internal
  Redistributing via eigrp 100
  Last update from 155.1.37.3 on FastEthernet0/3, 02:16:47 ago
  Routing Descriptor Blocks:
  * 155.1.37.3, from 155.1.37.3, 02:16:47 ago, via FastEthernet0/3
    Route metric is 130560, traffic share count is 1
    Total delay is 5100 microseconds, minimum bandwidth is 100000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

```
Rack1SW1#show ip eigrp topology 150.1.3.0 255.255.255.0
```

```
IP-EIGRP (AS 100): Topology entry for 150.1.3.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 130560
  Routing Descriptor Blocks:
  155.1.37.3 (FastEthernet0/3), from 155.1.37.3, Send flag is 0x0
    Composite metric is (130560/128000), Route is Internal
  Vector metric:
    Minimum bandwidth is 100000 Kbit
    Total delay is 5100 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
```

```
Rack1SW1#traceroute 150.1.3.3
```

```
Type escape sequence to abort.
Tracing the route to 150.1.3.3
```

```
 1 155.1.37.3 9 msec * 0 msec
```

In order for SW1 to route through R6 to reach this destination, the metric must be offset sufficiently so that R6 computes a lower composite metric through R1 than SW1. By offsetting to the maximum value inbound on SW1 from R3 this is ensured. Also since an access-list is used to match just 150.1.3.0, no other prefixes are affected by this traffic engineering.

```
Rack1SW1#show ip eigrp topology 150.1.3.0 255.255.255.0
```

```
IP-EIGRP (AS 100): Topology entry for 150.1.3.0/24
```

```
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
  25742592
```

```
Routing Descriptor Blocks:
```

```
 155.1.67.6 (Vlan67), from 155.1.67.6, Send flag is 0x0
```

```
    Composite metric is (25742592/142592), Route is Internal
```

```
    Vector metric:
```

```
      Minimum bandwidth is 1544 Kbit
```

```
      Total delay is 1005570 microseconds
```

```
      Reliability is 255/255
```

```
      Load is 1/255
```

```
      Minimum MTU is 1500
```

```
      Hop count is 3
```

```
 155.1.37.3 (FastEthernet0/3), from 155.1.37.3, Send flag is 0x0
```

```
    Composite metric is (2147614207/2147611647), Route is Internal
```

```
    Vector metric:
```

```
      Minimum bandwidth is 100000 Kbit
```

```
      Total delay is 83891179 microseconds
```

```
      Reliability is 255/255
```

```
      Load is 1/255
```

```
      Minimum MTU is 1500
```

```
      Hop count is 1
```

```
Rack1SW1#traceroute 150.1.3.3
```

```
Type escape sequence to abort.
Tracing the route to 150.1.3.3
```

```
 1 155.1.67.6 9 msec 0 msec 0 msec
```

```
 2 155.1.146.1 0 msec 0 msec 8 msec
```

```
 3 155.1.13.3 8 msec * 8 msec
```

Since the route through R3 is still installed in the topology table it will be used as a backup route if the path through R6 is lost.

```
Rack1SW1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#interface Vlan67
Rack1SW1(config-if)#shutdown
Rack1SW1(config-if)#end
Rack1SW1#
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 155.1.67.6 (Vlan67) is
down: interface down
%SYS-5-CONFIG_I: Configured from console by console
%LINK-5-CHANGED: Interface Vlan67, changed state to administratively
down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan67, changed state
to down
```

```
Rack1SW1#show ip eigrp topology 150.1.3.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 150.1.3.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
  2147614207
  Routing Descriptor Blocks:
    155.1.37.3 (FastEthernet0/3), from 155.1.37.3, Send flag is 0x0
      Composite metric is (2147614207/2147611647), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 83891179 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

```
Rack1SW1#traceroute 150.1.3.3
```

```
Type escape sequence to abort.
Tracing the route to 150.1.3.3
```

```
 1 155.1.37.3 0 msec * 0 msec
```

5.24 EIGRP Filtering with Administrative Distance

- Configure administrative distance filtering on R6 so that it does not install the route to R4's Loopback0 network.

Configuration

```
R6:
access-list 4 permit 150.1.4.0
!
router eigrp 100
 distance 255 0.0.0.0 255.255.255.255 4
```

Verification

Note

Like in the other IGP protocols, administrative distance can be set on a per-prefix basis in EIGRP. In this example the source address of the route is ignored by matching an address of 0.0.0.0 with the wildcard 255.255.255.255, while access-list 4 matches the route to change the distance for. Since the AD value of 255 is "infinite", the route in question cannot be installed in the routing table or the EIGRP topology.

```
Rack1R6#show ip route 150.1.4.4
Routing entry for 150.1.4.0/24
  Known via "eigrp 100", distance 90, metric 156160, type internal
  Redistributing via eigrp 10, eigrp 100
  Advertised by eigrp 10
  Last update from 155.1.146.4 on FastEthernet0/0.146, 00:27:44 ago
  Routing Descriptor Blocks:
  * 155.1.146.4, from 155.1.146.4, 00:27:44 ago, via FastEthernet0/0.146
    Route metric is 156160, traffic share count is 1
    Total delay is 5100 microseconds, minimum bandwidth is 100000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

```
Rack1R6#config t
Enter configuration commands, one per line. End with CNTL/Z.
Rack1R6(config)#access-list 4 permit 150.1.4.0
Rack1R6(config)#router eigrp 100
Rack1R6(config-router)#distance 255 0.0.0.0 255.255.255.255 4
Rack1R6(config-router)#end
Rack1R6#clear ip route *
Rack1R6#show ip route 150.1.4.4
% Subnet not in table
```

```
Rack1R6#show ip eigrp 100 topology 150.1.4.0 255.255.255.0
% IP-EIGRP (AS 100): Route not in topology table
```

5.25 EIGRP Filtering with Per Neighbor AD

- Configure administrative distance filtering on R3 so that traffic destined for SW1's Loopback0 network is sent towards R1.

Configuration

```
R3:
access-list 7 permit 150.1.7.0
!
router eigrp 100
 distance 255 155.1.37.7 0.0.0.0 7
```

Verification

Note

Prior to any distance modifications R3 routes directly to SW1 to reach 150.1.7.0/24. Based on the routing table and EIGRP topology table we can see that the Feasible Distance is 128256, and the neighbor the route is learned from is 155.1.37.7.

```
Rack1R3#show ip route 150.1.7.7
```

```
Routing entry for 150.1.7.0/24
  Known via "eigrp 100", distance 90, metric 128256, type internal
  Redistributing via eigrp 100
  Last update from 155.1.37.7 on FastEthernet0/0, 00:01:32 ago
  Routing Descriptor Blocks:
  * 155.1.37.7, from 155.1.37.7, 00:01:32 ago, via FastEthernet0/0
    Route metric is 128256, traffic share count is 1
    Total delay is 5010 microseconds, minimum bandwidth is 100000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

```
Rack1R3#show ip eigrp topology 150.1.7.0 255.255.255.0
```

```
IP-EIGRP (AS 100): Topology entry for 150.1.7.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
  155.1.37.7 (FastEthernet0/0), from 155.1.37.7, Send flag is 0x0
    Composite metric is (128256/128000), Route is Internal
    Vector metric:
      Minimum bandwidth is 100000 Kbit
      Total delay is 5010 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 1
  155.1.0.5 (Serial1/0.1), from 155.1.0.5, Send flag is 0x0
    Composite metric is (1152512/640512), Route is Internal
    Vector metric:
      Minimum bandwidth is 128 Kbit
      Total delay is 45020 microseconds
      Reliability is 255/255
      Load is 1/255
```

Minimum MTU is 1500

```
Rack1R3#traceroute 150.1.7.7
```

```
Type escape sequence to abort.  
Tracing the route to 150.1.7.7
```

```
 1 155.1.137.7 0 msec * 0 msec
```

As we saw in the previous example administrative distance can be changed on a per-prefix basis. Based on matching who the route is learned from the distance can also be changed on a per-prefix per-neighbor basis.

```
Rack1R3#config t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Rack1R3(config)#access-list 7 permit 150.1.7.0
```

```
Rack1R3(config)#router eigrp 100
```

```
Rack1R3(config-router)#distance 255 155.1.37.7 0.0.0.0 7
```

```
Rack1R3(config-router)#end
```

```
Rack1R3#clear ip route *
```

```
Rack1R3#show ip route 150.1.7.7
```

```
Routing entry for 150.1.7.0/24
```

```
  Known via "eigrp 100", distance 90, metric 645120, type internal
```

```
  Redistributing via eigrp 100
```

```
  Last update from 155.1.13.1 on Serial1/2, 00:00:01 ago
```

```
  Routing Descriptor Blocks:
```

```
    * 155.1.13.1, from 155.1.13.1, 00:00:01 ago, via Serial1/2
```

```
      Route metric is 645120, traffic share count is 1
```

```
      Total delay is 25200 microseconds, minimum bandwidth is 128 Kbit
```

```
      Reliability 255/255, minimum MTU 1500 bytes
```

```
      Loading 1/255, Hops 3
```


Although the composite metric is higher through R1 than it was originally through SW1, the route through SW1 cannot be installed in the topology table because it has an infinite administrative distance. This implies that R3 must route through R1 to reach the destination.

Rack1R3#show ip eigrp topology 150.1.7.0 255.255.255.0

```
IP-EIGRP (AS 100): Topology entry for 150.1.7.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 645120
  Routing Descriptor Blocks:
  155.1.13.1 (Serial1/2), from 155.1.13.1, Send flag is 0x0
    Composite metric is (645120/133120), Route is Internal
    Vector metric:
      Minimum bandwidth is 128 Kbit
      Total delay is 25200 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 3
  155.1.0.5 (Serial1/0.1), from 155.1.0.5, Send flag is 0x0
    Composite metric is (1157120/645120), Route is Internal
    Vector metric:
      Minimum bandwidth is 128 Kbit
      Total delay is 45200 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 4
```

Rack1R3#traceroute 150.1.7.7

```
Type escape sequence to abort.
Tracing the route to 150.1.7.7
```

```
 1 155.1.13.1 16 msec 16 msec 12 msec
 2 155.1.146.6 12 msec 16 msec 16 msec
 3 155.1.67.7 4 msec * 4 msec
```

☠ Pitfall

The administrative distance for EIGRP internal routes can be changed on a per-prefix basis, but external EIGRP routes cannot.

5.26 EIGRP Filtering with Route Maps

- Configure R4 to redistribute the VLAN 43 subnet into EIGRP with the tag value of 4.
- Configure a route-map filter on R2 that matches this tag value and denies the route from being installed in the routing table.
- Configure a route-map filter on R3 that denies EIGRP routes with a metric in the range of 500,000 – 750,000 from entering the routing table.
- These filters should not impact any other networks advertised by R4 or learned by R2 and R3.

Configuration

```
R2:
router eigrp 100
  distribute-list route-map FILTER_ON_TAGS in
!
route-map FILTER_ON_TAGS deny 10
  match tag 4
!
route-map FILTER_ON_TAGS permit 20

R3:
router eigrp 100
  distribute-list route-map FILTER_ON_METRIC_RANGE in
!
route-map FILTER_ON_METRIC_RANGE deny 10
  match metric 625000 +- 125000
!
route-map FILTER_ON_METRIC_RANGE permit 20

R4:
router eigrp 100
  redistribute rip metric 100000 100 255 1 1500 route-map RIP_TO_EIGRP
!
ip prefix-list VLAN_43 seq 5 permit 204.12.1.0/24
!
route-map RIP_TO_EIGRP permit 10
  match ip address prefix-list VLAN_43
  set tag 4
!
route-map RIP_TO_EIGRP permit 20
```

Verification

Note

Unlike BGP, filtering with route-maps in IGP is usually limited to redistribution filtering only. However EIGRP now supports route-map filtering as a distribute-list with matches on metric and tag.

Route tags are set at the time of redistribution, and can be used like BGP community values to group prefixes together without having to match on the actual route in a prefix-list or access-list. In this example we can see that R2 and R4 see the prefix 204.12.1.0/24 with a tag of 4 in the topology table. R2 installs this in the routing table until the distribute-list is applied which denies routes with that tag value.

```
Rack1R4#show ip eigrp topology | include tag
P 204.12.1.0/24, 1 successors, FD is 25600, tag is 4
```

```
Rack1R2#show ip eigrp topology | include tag
P 204.12.1.0/24, 1 successors, FD is 1049600, tag is 4
```

```
Rack1R2#show ip route 204.12.1.0
Routing entry for 204.12.1.0/24
  Known via "eigrp 100", distance 170, metric 1049600
  Tag 4, type external
  Redistributing via eigrp 100, rip
  Advertised by rip metric 1
  Last update from 155.1.0.5 on Serial0/0.1, 00:05:14 ago
  Routing Descriptor Blocks:
  * 155.1.0.5, from 155.1.0.5, 00:05:14 ago, via Serial0/0.1
    Route metric is 1049600, traffic share count is 1
    Total delay is 41000 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 2
    Route tag 4
```

```
Rack1R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R2(config)#route-map FILTER_ON_TAGS deny 10
Rack1R2(config-route-map)#match tag 4
Rack1R2(config-route-map)#route-map FILTER_ON_TAGS permit 20
Rack1R2(config-route-map)#router eigrp 100
Rack1R2(config-router)#distribute-list route-map FILTER_ON_TAGS in
Rack1R2(config-router)#end
Rack1R2#
```

```
Rack1R2#show ip route 204.12.1.0
```

```
% Network not in table
```

```
Rack1R2#show ip eigrp topology 204.12.1.0 255.255.255.0
```

```
% IP-EIGRP (AS 100): Route not in topology table
```

Other routes learned by EIGRP are not affected by this filter.

```
Rack1R2#show ip route eigrp
```

```
155.1.0.0/24 is subnetted, 13 subnets
D    155.1.146.0 [90/1026560] via 155.1.0.5, 00:05:58, Serial0/0.1
D    155.1.8.0 [90/514816] via 155.1.0.5, 00:09:19, Serial0/0.1
D    155.1.9.0 [90/512768] via 155.1.23.3, 00:09:19, Serial0/1
D    155.1.13.0 [90/1024000] via 155.1.23.3, 00:09:24, Serial0/1
D    155.1.7.0 [90/512512] via 155.1.23.3, 00:09:19, Serial0/1
D    155.1.5.0 [90/514560] via 155.1.0.5, 00:09:19, Serial0/0.1
D    155.1.58.0 [90/514560] via 155.1.0.5, 00:09:19, Serial0/0.1
D    155.1.37.0 [90/512256] via 155.1.23.3, 00:09:19, Serial0/1
D    155.1.9.0 [90/512512] via 155.1.23.3, 00:09:19, Serial0/1
D    155.1.67.0 [90/1029120] via 155.1.0.5, 00:05:58, Serial0/0.1
D    155.1.108.0 [90/517120] via 155.1.0.5, 00:09:19, Serial0/0.1
D EX 200.0.0.0/24 [170/1666560] via 155.1.23.3, 00:09:22, Serial0/1
      [170/1666560] via 155.1.0.5, 00:09:22, Serial0/0.1
54.0.0.0/24 is subnetted, 1 subnets
D EX 54.1.1.0 [170/1538560] via 155.1.23.3, 00:09:22, Serial0/1
      [170/1538560] via 155.1.0.5, 00:09:23, Serial0/0.1
D EX 200.0.2.0/24 [170/1666560] via 155.1.23.3, 00:09:23, Serial0/1
      [170/1666560] via 155.1.0.5, 00:09:23, Serial0/0.1
31.0.0.0/16 is subnetted, 4 subnets
D EX 31.3.0.0 [170/1049600] via 155.1.0.5, 00:05:59, Serial0/0.1
D EX 31.2.0.0 [170/1049600] via 155.1.0.5, 00:05:59, Serial0/0.1
D EX 31.1.0.0 [170/1049600] via 155.1.0.5, 00:05:59, Serial0/0.1
D EX 31.0.0.0 [170/1049600] via 155.1.0.5, 00:06:01, Serial0/0.1
150.1.0.0/24 is subnetted, 7 subnets
D    150.1.7.0 [90/1157120] via 155.1.0.5, 00:06:00, Serial0/0.1
D    150.1.6.0 [90/26240256] via 155.1.23.3, 00:06:00, Serial0/1
D    150.1.5.0 [90/640000] via 155.1.0.5, 00:09:21, Serial0/0.1
D    150.1.3.0 [90/640000] via 155.1.23.3, 00:09:21, Serial0/1
D    150.1.1.0 [90/26254592] via 155.1.23.3, 00:06:00, Serial0/1
D    150.1.9.0 [90/640512] via 155.1.23.3, 00:09:25, Serial0/1
30.0.0.0/16 is subnetted, 4 subnets
D EX 30.2.0.0 [170/1049600] via 155.1.0.5, 00:09:21, Serial0/0.1
D EX 30.3.0.0 [170/1049600] via 155.1.0.5, 00:06:01, Serial0/0.1
D EX 30.0.0.0 [170/1049600] via 155.1.0.5, 00:09:21, Serial0/0.1
D EX 30.1.0.0 [170/1049600] via 155.1.0.5, 00:09:21, Serial0/0.1
```

As a filter for metrics the route-map match can on an absolute metric value, such as with the `match metric 10` command, or on a range of metrics. In this case metrics in the range of 500,000 – 750,000 are filtered out based on matching the value 625,000 plus or minus 125,000.

We can see from the below output the highlighted prefixes are no longer installed in the routing table via the same path once the filter is applied. Note that the route 150.1.2.0/24 is withdrawn completely as there is no other valid path to this destination than the one with a metric value matched by the range.

Rack1R3#show ip route eigrp

```
D EX 222.22.2.0/24 [170/537600] via 155.1.23.2, 00:02:51, Serial1/3
D EX 204.12.1.0/24 [170/1049600] via 155.1.0.5, 00:02:51, Serial1/0.1
    155.1.0.0/24 is subnetted, 13 subnets
D    155.1.146.0 [90/514560] via 155.1.13.1, 00:02:51, Serial1/2
D    155.1.8.0 [90/514816] via 155.1.0.5, 00:02:51, Serial1/0.1
D    155.1.9.0 [90/768] via 155.1.37.7, 00:02:51, FastEthernet0/0
D    155.1.7.0 [90/512] via 155.1.37.7, 00:02:51, FastEthernet0/0
D    155.1.5.0 [90/514560] via 155.1.0.5, 00:02:51, Serial1/0.1
D    155.1.58.0 [90/514560] via 155.1.0.5, 00:02:51, Serial1/0.1
D    155.1.79.0 [90/512] via 155.1.37.7, 00:02:51, FastEthernet0/0
D    155.1.67.0 [90/517120] via 155.1.13.1, 00:02:51, Serial1/2
D    155.1.108.0 [90/517120] via 155.1.0.5, 00:02:51, Serial1/0.1
D EX 220.20.3.0/24 [170/537600] via 155.1.23.2, 00:02:51, Serial1/3
D EX 200.0.0.0/24 [170/1154560] via 155.1.13.1, 00:02:51, Serial1/2
    54.0.0.0/24 is subnetted, 1 subnets
D EX    54.1.1.0 [170/1026560] via 155.1.13.1, 00:02:52, Serial1/2
D EX 200.0.2.0/24 [170/1154560] via 155.1.13.1, 00:02:52, Serial1/2
D EX 192.10.1.0/24 [170/537600] via 155.1.23.2, 00:02:52, Serial1/3
    31.0.0.0/16 is subnetted, 4 subnets
D EX    31.3.0.0 [170/1049600] via 155.1.0.5, 00:02:52, Serial1/0.1
D EX    31.2.0.0 [170/1049600] via 155.1.0.5, 00:02:52, Serial1/0.1
D EX    31.1.0.0 [170/1049600] via 155.1.0.5, 00:02:52, Serial1/0.1
D EX    31.0.0.0 [170/1049600] via 155.1.0.5, 00:02:52, Serial1/0.1
    150.1.0.0/24 is subnetted, 7 subnets
D    150.1.7.0 [90/645120] via 155.1.13.1, 00:02:53, Serial1/2
D    150.1.6.0 [90/642560] via 155.1.13.1, 00:02:53, Serial1/2
D    150.1.5.0 [90/640000] via 155.1.0.5, 00:02:53, Serial1/0.1
D    150.1.2.0 [90/640000] via 155.1.23.2, 00:02:52, Serial1/3
D    150.1.1.0 [90/640000] via 155.1.13.1, 00:02:52, Serial1/2
D    150.1.9.0 [90/128512] via 155.1.37.7, 00:02:53, FastEthernet0/0
D EX 205.90.31.0/24 [170/537600] via 155.1.23.2, 00:02:53, Serial1/3
    30.0.0.0/16 is subnetted, 4 subnets
D EX    30.2.0.0 [170/1049600] via 155.1.0.5, 00:02:53, Serial1/0.1
D EX    30.3.0.0 [170/1049600] via 155.1.0.5, 00:02:53, Serial1/0.1
D EX    30.0.0.0 [170/1049600] via 155.1.0.5, 00:02:53, Serial1/0.1
D EX    30.1.0.0 [170/1049600] via 155.1.0.5, 00:02:53, Serial1/0.1
```

```
Rack1R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R3(config)#router eigrp 100
Rack1R3(config-router)#distribute-list route-map FILTER_ON_METRIC_RANGE
IN
Rack1R3(config-router)#end
```

```
Rack1R3#show ip route eigrp
D EX 222.22.2.0/24 [170/1049600] via 155.1.0.5, 00:00:17, Serial1/0.1
D EX 204.12.1.0/24 [170/1049600] via 155.1.0.5, 00:03:38, Serial1/0.1
  155.1.0.0/24 is subnetted, 13 subnets
D    155.1.146.0 [90/1026560] via 155.1.0.5, 00:00:17, Serial1/0.1
D    155.1.8.0 [90/1026816] via 155.1.23.2, 00:00:17, Serial1/3
      [90/1026816] via 155.1.13.1, 00:00:17, Serial1/2
D    155.1.9.0 [90/768] via 155.1.37.7, 00:03:38, FastEthernet0/0
D    155.1.7.0 [90/512] via 155.1.37.7, 00:03:38, FastEthernet0/0
D    155.1.5.0 [90/1026560] via 155.1.23.2, 00:00:17, Serial1/3
      [90/1026560] via 155.1.13.1, 00:00:17, Serial1/2
D    155.1.58.0 [90/1026560] via 155.1.23.2, 00:00:17, Serial1/3
      [90/1026560] via 155.1.13.1, 00:00:17, Serial1/2
D    155.1.79.0 [90/512] via 155.1.37.7, 00:03:38, FastEthernet0/0
D    155.1.67.0 [90/1029120] via 155.1.0.5, 00:00:18, Serial1/0.1
D    155.1.108.0 [90/1029120] via 155.1.23.2, 00:00:17, Serial1/3
      [90/1029120] via 155.1.13.1, 00:00:18, Serial1/2
D EX 220.20.3.0/24 [170/1049600] via 155.1.0.5, 00:00:19, Serial1/0.1
D EX 200.0.0.0/24 [170/1154560] via 155.1.13.1, 00:00:19, Serial1/2
  54.0.0.0/24 is subnetted, 1 subnets
D EX  54.1.1.0 [170/1026560] via 155.1.13.1, 00:00:19, Serial1/2
D EX 200.0.2.0/24 [170/1154560] via 155.1.13.1, 00:00:19, Serial1/2
D EX 192.10.1.0/24 [170/1049600] via 155.1.0.5, 00:00:19, Serial1/0.1
  31.0.0.0/16 is subnetted, 4 subnets
D EX  31.3.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
D EX  31.2.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
D EX  31.1.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
D EX  31.0.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
  150.1.0.0/24 is subnetted, 6 subnets
D    150.1.7.0 [90/1157120] via 155.1.0.5, 00:00:20, Serial1/0.1
D    150.1.6.0 [90/25728256] via 155.1.37.7, 00:00:20, FastEthernet0/0
D    150.1.5.0 [90/1152000] via 155.1.23.2, 00:00:20, Serial1/3
      [90/1152000] via 155.1.13.1, 00:00:20, Serial1/2
D    150.1.1.0 [90/25742592] via 155.1.37.7, 00:00:20, FastEthernet0/0
D    150.1.9.0 [90/128512] via 155.1.37.7, 00:03:41, FastEthernet0/0
D EX 205.90.31.0/24 [170/1049600] via 155.1.0.5, 00:00:20, Serial1/0.1
  30.0.0.0/16 is subnetted, 4 subnets
D EX  30.2.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
D EX  30.3.0.0 [170/1049600] via 155.1.0.5, 00:03:41, Serial1/0.1
D EX  30.0.0.0 [170/1049600] via 155.1.0.5, 00:03:42, Serial1/0.1
D EX  30.1.0.0 [170/1049600] via 155.1.0.5, 00:03:42, Serial1/0.1
```

5.27 EIGRP Bandwidth Pacing

- Configure R2 and R3 so that EIGRP can not use more than 154Kbps of bandwidth on the point-to-point link between them, assuming that the link speed is 1544Kbps.

Configuration

```
R2:
interface Serial0/1
  bandwidth 1544
  ip bandwidth-percent eigrp 100 10
```

```
R3:
interface Serial1/3
  bandwidth 1544
  ip bandwidth-percent eigrp 100 10
```

5.28 EIGRP Default Metric

- Configure a static route on R2 for the prefix 222.22.2.2/32 that is reachable via BB2.
- Advertise this prefix into EIGRP as external routes using a default metric of 100Mbps, 100 microseconds of delay, maximum reliability, minimum load, and an MTU of 1500 bytes.

Configuration

```
R2:
ip route 222.22.2.2 255.255.255.255 192.10.1.254
!
router eigrp 100
 redistribute static
 default-metric 100000 10 255 1 1500
```

Verification

Note

When redistributing connected into EIGRP or between EIGRP processes, metrics are automatically derived from the source prefix. For all other redistribution the metric must be manually set on the redistribute statement, under a route-map, or from the default metric.

Rack1R2#show ip eigrp topology 222.22.2.2/32

```
IP-EIGRP (AS 100): Topology entry for 222.22.2.2/32
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2560
Routing Descriptor Blocks:
 192.10.1.254, from Rstatic, Send flag is 0x0
   Composite metric is (2560/0), Route is External
   Vector metric:
     Minimum bandwidth is 100000 Kbit
     Total delay is 100 microseconds
     Reliability is 255/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 0
   External data:
     Originating router is 150.1.2.2 (this system)
     AS number of route is 0
     External protocol is Static, external metric is 0
     Administrator tag is 0 (0x00000000)
```


5.29 EIGRP Neighbor Logging

- Configure SW3 so that it does not log EIGRP neighbor adjacency events.
- Additionally EIGRP warning logs should not be generated more often than every 20 seconds.

Configuration

```
SW3:  
router eigrp 100  
  no eigrp log-neighbor-changes  
  eigrp log-neighbor-warnings 20
```

5.30 EIGRP Router-ID

- Modify the EIGRP Router-ID on SW2 so that external EIGRP routes generated by R2 are ignored.

Configuration

```
SW2:
router eigrp 100
 eigrp router-id 150.1.2.2
```

Verification

Note

EIGRP uses the router-id field in external routes as a loop prevention mechanism. The router who originates the external route inserts its EIGRP router-id into the update. If an update is received back in with a router-id in this field matching the local router-id, the update is dropped.

In this case, SW2's router-id is 150.1.8.8, as seen in the topology table.

```
Rack1SW2#show ip eigrp topology | include ID
IP-EIGRP Topology Table for AS(100)/ID(150.1.8.8)
```

```
Rack1SW2#show ip route | include D EX
D EX    222.22.2.2/32 [170/514816] via 155.1.58.5, 00:12:47, Vlan58
D EX    222.22.2.0/24 [170/537856] via 155.1.58.5, 02:19:28, Vlan58
D EX 204.12.1.0/24 [170/537856] via 155.1.58.5, 01:41:23, Vlan58
D EX 220.20.3.0/24 [170/537856] via 155.1.58.5, 02:19:28, Vlan58
D EX 200.0.0.0/24 [170/1154816] via 155.1.58.5, 02:19:28, Vlan58
D EX    54.1.1.0 [170/1026816] via 155.1.58.5, 00:06:25, Vlan58
D EX 200.0.2.0/24 [170/1154816] via 155.1.58.5, 02:19:28, Vlan58
D EX 192.10.1.0/24 [170/514816] via 155.1.58.5, 00:12:45, Vlan58
D EX    31.3.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    31.2.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    31.1.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    31.0.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX 205.90.31.0/24 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    30.2.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    30.3.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    30.0.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
D EX    30.1.0.0 [170/537856] via 155.1.58.5, 02:19:29, Vlan58
```

R2 is originating the external route 222.22.2.0/24, and it is tagged with R2's router-id of 150.1.2.2

```
Rack1SW2#show ip eigrp topology 222.22.2.0/24
IP-EIGRP (AS 100): Topology entry for 222.22.2.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
  537856
  Routing Descriptor Blocks:
  155.1.58.5 (Vlan58), from 155.1.58.5, Send flag is 0x0
    Composite metric is (537856/537600), Route is External
    Vector metric:
      Minimum bandwidth is 1544 Kbit
      Total delay is 21010 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 2
    External data:
      Originating router is 150.1.2.2
      AS number of route is 0
      External protocol is RIP, external metric is 7
      Administrator tag is 0 (0x00000000)
```

If SW2 shares this same router-id, this prefix can no longer be installed, along with any other external routes originated by R2.

```
Rack1SW2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW2(config)#router eigrp 100
Rack1SW2(config-router)#eigrp router-id 150.1.2.2
Rack1SW2(config-router)#end
```

```
Rack1SW2#show ip eigrp topology 222.22.2.0/24
% IP-EIGRP (AS 100): Route not in topology table
```

```
Rack1SW2#show ip route | include D EX
D EX 204.12.1.0/24 [170/537856] via 155.1.58.5, 00:00:51, Vlan58
D EX 200.0.0.0/24 [170/1154816] via 155.1.58.5, 00:00:51, Vlan58
D EX   54.1.1.0 [170/1026816] via 155.1.58.5, 00:00:51, Vlan58
D EX 200.0.2.0/24 [170/1154816] via 155.1.58.5, 00:00:51, Vlan58
D EX   31.3.0.0 [170/537856] via 155.1.58.5, 00:00:51, Vlan58
D EX   31.2.0.0 [170/537856] via 155.1.58.5, 00:00:51, Vlan58
D EX   31.1.0.0 [170/537856] via 155.1.58.5, 00:00:51, Vlan58
D EX   31.0.0.0 [170/537856] via 155.1.58.5, 00:00:51, Vlan58
D EX   30.2.0.0 [170/537856] via 155.1.58.5, 00:00:52, Vlan58
D EX   30.3.0.0 [170/537856] via 155.1.58.5, 00:00:52, Vlan58
D EX   30.0.0.0 [170/537856] via 155.1.58.5, 00:00:52, Vlan58
D EX   30.1.0.0 [170/537856] via 155.1.58.5, 00:00:52, Vlan58
```

5.31 EIGRP Maximum Hops

- Configure all devices in EIGRP AS 100 so that routes with a hop count of greater than 10 are considered invalid.

Configuration

```
R1 - R6, SW1 - SW4:  
router eigrp 100  
  metric maximum-hops 10
```