Copyright Information

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

IP Services

> **Note**
>
> Load the *IP Services* initial configurations prior to starting.

13.1 Proxy ARP

- Disable IP routing on SW1, configure interface VLAN 146 with the IP address 155.X.146.7/24, and shutdown all other interfaces running IP.
- Disable Proxy-ARP on R4's connection to VLAN 146.
- Test reachability from SW1 to the rest of the devices in the routing domain, and check the ARP cache of SW1.
- Configure R5's interface FastEthernet0/0 in VLAN 146 with the IP address 155.X.146.5/24, and disable Proxy-ARP on the link.
- Configure the layer 2 switchports connecting to R1 and R5 on SW1 as protected ports, and enable Local Proxy ARP on R6's connection to VLAN 146.
- Test reachability from R1 to R5 on VLAN 146 and check the ARP cache of both of these devices.
- Revert to the original configuration on R5 and SW1.

13.2 DHCP Server

- Configure R6 as a DHCP server for hosts on VLAN 146, but to ignore BOOTP packets.
- Store the DHCP binding in the flash memory of R6.
- Clients should use R4 and R6 as the redundant default gateways.
- DHCP clients should use the default-router IP addresses for DNS servers.
- Exclude the address range 155.X.146.100 - 155.X.146.254 from allocation.
- Lease the addresses for 12 hours.

13.3 DHCP Client

- Configure R1 to obtain its IP address for the link to VLAN 146 dynamically via DHCP.
- The DHCP client-id should be based off of the MAC address of the interface connecting to VLAN 146.

13.4 DHCP Relay

- Configure R6 to allocate IP addresses via DHCP to clients on VLAN 58.
- The clients on VLAN 58 should use R5 as the default gateway.
- Configure SW2's VLAN 58 interface to obtain an IP address via DHCP.
- Base the client-id on the MAC address of its VLAN 58 interface.
- Ensure to avoid IP address conflicts with the existing hosts on VLAN 58.

13.5 DHCP Host Pool

- Create a DHCP host pool on R6 corresponding to R1's VLAN 146 interface to lease R1 the IP address 155.X.146.1.

13.6 DHCP On-Demand Pool

- Configure the link between R1 and R3 to use PPP.
- Configure R3 to assign the IP address 155.X.13.1/24 to R1 using IPCP.
- Create a DHCP pool on R1 that imports the subnet, the netmask, and the DNS servers (R4 and R6) from R3 via IPCP.
- Ensure that RIP routing is still functional once this configuration is complete.

13.7 DHCP Proxy

- Configure R2 to request an IP address on its serial connection to R3 via IPCP.
- R3 should relay this request via DHCP to R6, and R6 should allocate the IP address 155.X.23.2 to R2.

13.8 DHCP Information Option

- Configure R5 to insert the DHCP information option field into DHCP requests forwarded to R6 from clients on VLAN 58.
- R5 should set the subscriber-id string for these requests to "VLAN58".
- Configure R6 to allocate IP addresses based on Option 82 and assign a static IP address to SW2 based on the information option contents.

13.9 DHCP Authorized ARP

- Configure R6 to update the ARP cache on its VLAN 146 interface with the information from the DHCP binding database.
- Ensure that R6 does not accept any ARP mappings on its VLAN 146 interface other than those learned from the DHCP binding database.

Note

Revert all devices to the *IP Services* initial configurations prior to continuing.

13.10 IP SLA

- Configure R6 to automatically monitor BB1's availability using ping and telnet every one minute.

13.11 Object Tracking

- Create two enhanced objects on R6 to track the state of previously created IP SLA operations.
- Create a third enhanced object on R6 that tracks the first two objects; this third object should only be "up" if both of the first two objects are "up".
- The first two objects should wait 10 seconds before reporting their state from up to down, or from down to up.

13.12 HSRP

- Configure HSRP on R4 and R6 so that hosts on VLAN 146 can use the virtual IP address 155.X.146.254 as their default gateway.
- Ensure that R6 is the active physical gateway unless it loses connectivity to VLAN 146.
- Set the hello interval to 1 second and dead interval to 3 seconds for faster failed gateway detection.
- Authenticate HSRP message exchanges using an MD5 hash of the key CISCO.

13.13 VRRP

- Configure VRRP on R4 and R6 so that hosts on VLAN 146 can use the virtual IP address 155.X.146.253 as their default gateway.
- Ensure that R4 is the active physical gateway unless it loses connectivity to VLAN 146.
- Set the hello interval to 3 seconds, and authenticate messages using the clear-text password CISCO.

13.14 GLBP

- Configure GLBP for hosts on VLAN 146 so that first-hop redundancy and load balancing occur between R4 and R6.
- Ensure R6 is elected as the Active Virtual Gateway (AVG), and both R4 and R6 are used as Active Virtual Forwarders (AVFs) for the virtual IP address 155.X.146.252.
- Use a 1 second hello interval and a 3 second hold time.
- Distribute the traffic load between R4 and R6 in 2:1 ratio.
- Authenticate GLBP UDP packet exchange using a secure method based on a password value of CISCO.

13.15 Router Redundancy and Objects Tracking

- R6 should be the active HSRP gateway as long as it can ping BB1 and reach it through telnet, but give up the active role once it loses either telnet or ping reachability to BB1.
- Ensure R4 is the active VRRP gateway when it sees the prefix 30.0.0.0/16 received via IGP from BB3.

13.16 IRDP

- R5 and SW2 should advertise themselves as default gateways for hosts on VLAN58 using ICMP messages.
- R5 should be the preferred gateway on the segment.

> **Note**
>
> Revert all devices to the *IP Services* initial configurations prior to continuing.

13.17 Router ICMP Settings

- Configure R1's VLAN146 interface to stop sending ICMP messages about discarded packets, ICMP messages to select a better next-hop, and ICMP messages reporting subnets mask.
- Rate-limit ICMP unreachables to 2 per-second globally.

13.18 Basic NAT

- Ensure that R4 and R6 are not advertising any internal networks (i.e. 155.X.0.0/16 or 150.X.0.0/16) to BB3.
- Configure R4 and R6 to translate IP source addresses for packets from the subnet 155.X.0.0/16 going to BB3 and BB1 respectively using a dynamic pool consisting of the subnets of their BB connections.
- Do not use any route-maps to accomplish this.

13.19 NAT Overload

- Configure R4 and R6 to translate packets sourced from the Loopback0 subnets (150.X.0.0/16) using their backbone connection IP addresses.

13.20 NAT with Route Maps

- Configure R2 with route-map based NAT to support multiple outside interfaces.
- Traffic from R2's Loopback0 network going out the point-to-point link to R3 should be translated to 155.X.23.200.
- All other traffic going out the point-to-point link to R3 should be translated to R2's interface IP address.
- Traffic from R2's Loopback0 network going out the Frame Relay link to R5 should be translated to 150.X.2.200.
- All other traffic going out the Frame Relay link to R5 should be translated to R2's interface IP address.

13.21 Static NAT

- Configure R5 so that R4 can reach SW2's VLAN 58 IP address using the address 155.X.45.8.
- Additionally SW2 should be able reach R4's point-to-point link using the VLAN 58 IP address 155.X.58.4.

13.22 Static PAT

- Remove the previous static NAT entries on R5.
- Configure R5 so that when R4 telnets to 155.X.45.44 at port 8023 they get connected to SW2's VLAN 8 interface at port 23.
- Configure R5 so that when R4 telnets to 155.X.45.44 at port 10023 they get connected to SW4's VLAN 10 interface at port 23.

13.23 NAT and IP Aliasing

- Configure R5 so that it does not add a local alias for the static mapping of the IP address 155.X.45.44.

13.24 Static Policy NAT

- Remove all previous NAT configurations from R2.
- Configure a static translation on R3 so that when traffic is received from SW1's VLAN 7 IP address and it is going out the point-to-point link to R1 it is translated to the address 155.X.13.7.
- Configure a static translation on R3 so that when traffic is received from SW1's VLAN 7 IP address and it is going out the point-to-point link to R2 it is translated to the address 155.X.23.7.

13.25 NAT with Overlapping Subnets

- Create a new Loopback1 interface on R1 and R2 using the IP addresses 10.0.0.1/24 and 10.0.0.2/24 respectively.
- Advertise the 10.0.0.0/24 network into RIP on R2.
- Configure NAT on R1 so that when R2 telnets to the address 11.0.0.Z from its Loopback1 address it is redirected to the address 10.0.0.Z attached to R1, where "Z" is any number.
- The host 10.0.0.Z on R1 should see the packets from R2 as if they were sourced from the network 22.0.0.0/24.

13.26 TCP Load Distribution with NAT

- Remove the previous NAT configurations on R5.
- Configure rotary NAT on R5 so that when SW2 telnets to the address 155.X.58.55 it is redirected to R1, R2, and R3 in an even distribution.

13.27 Stateful NAT with HSRP

- Remove all previous NAT configurations on R4 and R6.
- Configure HSRP on R4 and R6 with the group name “VLAN146” so that hosts on VLAN 146 can use the virtual IP address 155.X.146.254 as their default gateway.
- Confgure R6 to be the active physical gateway unless the line protocol of the link connecting to BB1 goes down.
- Set the hello interval to 1 second and dead interval to 3 seconds for faster failed gateway detection.
- Authenticate HSRP message exchanges using an MD5 hash of the key CISCO.
- Create a single NAT pool on R4 and R6 with the IP addresses in range 155.X.254.1-155.X.254.254.
- Configure BGP AS 100 on R4 and R6, and peer these routers with BB3 and BB1 respectively, who are in AS 54.
- Advertise the NAT pool via EBGP to BB1 and BB3.
- Redistribute all BGP routes into RIP on R4 and R6, and ensure that R6 is the preferred exit point.
- Configure dynamic NAT on R4 and R6 so that traffic from the 155.X.0.0/16 network is port translated to the previously defined NAT pool.
- Ensure R4 and R6 synchronize their NAT entries using HSRP so that TCP sessions from the inside network connected to devices in AS 54 are not dropped if R6 loses connectivity to BB1.

13.28 Stateful NAT Primary/Backup

- Remove the HSRP configuration on R4 and R6.
- Modify the SNAT configuration on these devices so it does not rely on HSRP status for tracking of the active SNAT router.
- R6 should be the primary and R4 should be the backup router.

13.29 NAT Virtual Interface

- Remove all previous NAT configurations on R5.
- Configure R5 so that traffic sourced from SW2's VLAN 8 network is translated to the source address range 155.X.188.0/24.
- Do not use the **`ip nat inside`** or **`ip nat outside`** commands or static routing to accomplish this.

13.30 NAT Default Interface

- Remove all previous NAT configurations on R5.
- Disable R5's connection to the Frame Relay network.
- Configure R5's point-to-point link to R4 as a passive-interface under RIP.
- Configure R5 so that all connections destined to its point-to-point Serial link are redirected to SW2's Loopback0 interface.
- All other inside to outside traffic on R5 should be dynamically overloaded to the point-to-point Serial link's IP address.

13.31 Reversible NAT

- Remove all previous NAT configurations on R5.
- Configure R5 so that all traffic received on the inside interface connecting to VLAN 58 is translated to the pool 155.X.45.100 – 155.X.45.200 as it exits out the point-to-point link to R4.
- Use route-map based NAT for this configuration to allow outside hosts to initiate connections back to hosts on the inside network once an inside to outside dynamic entry is created in the NAT table.

13.32 Static Extendable NAT

- Configure R5 so that SW4's Loopback0 address is reachable from the outside of the network using the IP addresses 155.1.45.201 and 155.1.45.202 at the same time.

Note

Revert all devices to the *IP Services* initial configurations prior to continuing.

13.33 IP Precedence Accounting

- Configure R6 to account for the number of input and output packets based on their IP Precedence values on it's connection to BB1.

13.34 IP Output Packets Accounting

- Configure R1 to account for the number of output packets on all physical interfaces.
- Only account for packets going to subnets of 155.X.0.0/16.
- Limit the accounting database size to 4096 entries.
- The number of records for packets not matching the account list should be set to 1.

13.35 IP Access Violation Accounting

- Configure R6 to deny packets with an IP precedence of 4 coming from BB1.
- Account for the packets denied by this policy.

13.36 MAC Address Accounting

- Configure R1 to account for the MAC addresses in IP packets entering or leaving its connection to VLAN 146.

13.37 TCP Optimization

- Modify R1's TCP configuration to meet the following requirements:
    - Avoid the TCP "silly window syndrome".
    - Enable high-performance TCP options.
    - Set the TCP window size to twice the standard 16 bit maximum value.
    - Limit the wait time for a TCP SYN response to the minimum.
    - Enable the feature to avoiding fragmentation with TCP sessions.
    - Hold no more that 16 packets in outgoing TCP queue.

13.38 IOS Small Services & Finger

- Configure R1 to meet the following requirements:
  - Users who telnet or send UDP packets to R1 at port 7 should have any typed characters echoed back to them.
  - Users who telnet or send UDP packets to R1 at port 9 should have any typed characters thrown away.
  - Users who telnet to R1 at port 13 should be returned the date and time.
  - Users who telnet or send UDP packets to R1 at port 19 should be sent a stream of ASCII characters.
  - Users who telnet to R1 at port 79 should be sent a list of currently logged in users.

13.39 Directed Broadcasts & UDP Forwarding

- Configure R5 so that if it receives packets for the destination address 155.X.58.255 they are forwarded out the link to SW2, but the destination should not be changed to 255.255.255.255.
- Configure R5 to listen to UDP broadcasts on the point-to-point link to R4, and forward them to the address 155.X.58.255.
- This forwarding should only occur for DNS packets.
- Verify this configuration by sending ICMP pings to the address 155.X.58.255 and DNS broadcast requests from R4.

13.40 DRP Server Agent

- Configure R1 to support reporting of information to a Distributed Director.
- Only allow connections from Directors from VLAN 146, authenticate the communication using the key value CISCO.

13.41 WCCPv1 Web-Cache

- Configure R4 and R6 for WCCPv1 web-cache redirection.
- R4 should redirect outgoing packets on its VLAN 43 interface, and R6 should redirect incoming packets on its VLAN 146 connection.
- Exclude packets coming in R4's Serial connection to R5 from redirection.
- Exclude connections sourced from the VLAN 146 subnet from redirection.
- Do not redirect flows excluded by outbound ACLs.

13.42 WCCPv2 Services

- Configure R5 to listen for WCCPv2 cache engines on its FastEthernet0/0 link using a dynamic service-group numbered 50.
- The cache engines will be sending packets to the multicast address 224.0.1.100.
- Authenticate them with the password CISCO.
- Only accept packets from the cache engines 155.X.58.100 and 155.X.58.200.
- Redirect traffic from hosts in the VLAN 58 subnet to these cache engines.

> **Note**
>
> Revert all devices to the *IP Services* initial configurations prior to continuing.

13.43 SLB Directed Mode

- Configure Server Load Balancing on R5 to distribute connections going to virtual the HTTP server IP address 155.X.58.55.
- The real hosts should consist of the Loopback0 IP addresses of R1, R2 and R3.
- Configure R5 so that traffic to the virtual address is NAT translated, and load balanced in a round robin fashion with a ratio of 1:2:3 between R1, R2, and R3 respectively.
- Reassign a client to another server after two failed initial SYN packets.
- Detect a failed server by counting to 3 failed connections, but retry a failed server after 120 seconds.

13.44 SLB Dispatched Mode

- Create additional Loopback1 interfaces on R1 and R6 with the same IP address 100.X.100.100/32 but do not advertise them into IGP.
- Configure an SLB server farm on R4 so that packets sent to the address 100.X.100.100 are redirected to the VLAN 146 IP addresses of R1 and R6.
- Load balance the connections equally, and limit the number of maximum connections to two per physical server.

13.45 NBAR Protocol Discovery

- Configure R4 and R6 to collect protocol statistics on their connections to backbone routers using NBAR.
- Both routers should be able to classify connections to HTTP proxy ports 3128 and 8080.
- Change the SOCKS protocol mapping to port 2080.
- Create a new custom protocol mapping called TEST that matches the ASCII character “A” in the beginning of a TCP segment flowing to the destination port 3001.

13.46 Netflow Ingress & Egress

- Configure R4 and R6 to export Netflow version 5 information to a host 155.X.146.100 in VLAN 146 using port 9999.
- R4 should collect statistics on incoming flows on all interfaces, while R6 should collect statistics on outgoing flows on all interfaces.
- Limit the size of the flow cache to 4096 entries and export the BGP AS of Origin if available.
- Collect additional information about ICMP message information as well as layer 2 VLAN numbers where applicable.

13.47 Netflow Top Talkers

- Configure R4 and R6 to collect information about 10 most active ICMP flows.
- R4 should count packets, while R6 counts bytes.
- Match only TCP and ICMP flows coming from the subnet 155.X.0.0/16.

13.48 Netflow Aggregation Cache

- Create an aggregation cache that is destination-prefix based on R4 and source-based on R6.
- Export the new information to 155.X.146.100 at port 9998 with Netflow version 8.
- Do not summarize addresses to more than a /8 prefix length.
- Store no more than 1024 aggregated entries.

13.49 Netflow Random Sampling

- Modify R6's Netflow configuration so that it only samples every 10th packet in each flow.

13.50 Netflow Input Filters

- Modify R6's Netflow sampling so that every packet is sampled for sources on VLAN 146.
- Other packets should still be randomly sampled.

> **Note**
>
> Revert all devices to the *IP Services* initial configurations prior to continuing.

13.51 IOS Authoritative DNS Server

- Configure R4 and R6 as primary DNS servers for the domain cisco.com.
- The primary name server should be called ns.cisco.com with a contact mailbox of ccie.cisco.com.
- Both R4 and R6 should resolve the names R4.cisco.com and R6.cisco.com to all connected IP addresses of the respective router.
- R1 should use both R4 and R6 as its DNS servers in a round-robin fashion.
- R1's DNS client should complete all unqualified domain-names with the name "cisco.com".

13.52 IOS Caching DNS Server

- Configure R3 as a DNS client of R1.
- When R1 receives a DNS query from R3 it should check its local cache.
- If the entry is not cached a request should be sent from R1 to R4 or R6 on behalf of R3.

13.53 IOS DNS Spoofing

- Configure R5 as a DNS client of R4.
- Configure SW2 as a DNS client of R5.
- If R5 loses connectivity to R4 it should respond to all DNS queries with the IP address of its Loopback0 interface.

13.54 IP Event Dampening

- Configure dampening on R6 so that after a reload its connection to BB1 is not advertised into IGP for 30 seconds.
- Additionally configure R6 so that when this connection flaps it does not disappear for more that 60 seconds from the routing table no matter how much penalty it accumulates.

IP Services Solutions

13.1 Proxy ARP

- Disable IP routing on SW1, configure interface VLAN 146 with the IP address 155.X.146.7/24, and shutdown all other interfaces running IP.
- Disable Proxy-ARP on R4's connection to VLAN 146.
- Test reachability from SW1 to the rest of the devices in the routing domain, and check the ARP cache of SW1.
- Configure R5's interface FastEthernet0/0 in VLAN 146 with the IP address 155.X.146.5/24, and disable Proxy-ARP on the link.
- Configure the layer 2 switchports on SW1 connecting to R1 and R5 as protected ports, and enable Local Proxy ARP on R6's connection to VLAN 146.
- Test reachability from R1 to R5 on VLAN 146 and check the ARP cache of both of these devices.
- Revert to the original configuration on R5 and SW1.

Configuration

```
R4:
interface FastEthernet0/1
 no ip proxy-arp

R5:
interface FastEthernet0/0
 ip address 155.1.146.5 255.255.255.0
 no ip proxy-arp

R6:
interface FastEthernet0/0.146
 ip proxy-arp
 ip local-proxy-arp

SW1:
interface Vlan 146
 ip address 155.1.146.7 255.255.255.0
!
interface Vlan 67
 shutdown
!
interface Vlan 79
 shutdown
!
interface FastEthernet0/3
 shutdown
!
no ip routing
!
```



```
interface FastEthernet 0/5
 switchport access vlan 146
 switchport mode access
 switchport protected
!
interface FastEthernet 0/1
 switchport protected
```

Verification

```
Rack5SW1#sh ip route
Default gateway is not set

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

> **Note**
>
> With IP routing disabled and no default-gateway is configured on SW1 it will ARP for all destinations, including non-local destinations.

```
Rack5SW1#clear arp-cache
Rack5SW1#ping 155.1.58.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.58.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 58/260/1065 ms

Rack5SW1#ping 155.1.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 58/258/1057 ms

Rack5SW1#show ip arp 155.1.146.6
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.6              1   0011.5cf9.97c0  ARPA   Vlan146

Rack5SW1#show ip arp 155.1.0.2
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.58.5               1   0011.5cf9.97c0  ARPA   Vlan146
```

The MAC addresses in SW1's ARP cache for R2 and R6 are the same, since R6 performs Proxy-ARP for all of SW1's requests.

Next, protected ports would normally prohibit R5 from communicating with R1. However with Local Proxy-ARP enabled on R6 it will reply to R1 and R5 with its own MAC address for them to reach each other.


```
Rack1R1#ping 155.1.146.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.5, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/3/4 ms

Rack1R1#ping 155.1.146.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Rack1R1#show ip arp 155.1.146.5
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.5              1   0011.5cf9.97c0  ARPA   FastEthernet0/0

Rack1R1#show ip arp 155.1.146.6
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.6              1   0011.5cf9.97c0  ARPA   FastEthernet0/0
```

13.2 DHCP Server

- Configure R6 as a DHCP server for hosts on VLAN 146, but to ignore BOOTP packets.
- Store the DHCP binding in the flash memory of R6.
- Clients should use R4 and R6 as the redundant default gateways.
- DHCP clients should use the default-router IP addresses for DNS servers.
- Exclude the address range 155.X.146.100 - 155.X.146.254 from allocation.
- Lease the addresses for 12 hours.

Configuration

```
R6:
ip dhcp pool VLAN146
 network 155.1.146.0 /24
 default-router 155.1.146.6 155.1.146.4
 dns-server 155.1.146.6 155.1.146.4
 lease 0 12
!
no ip bootp server
!
ip dhcp excluded-address 155.1.146.100 155.1.146.254
!
ip dhcp database flash:/bindings
```

Verification

```
Rack1R6#show ip dhcp pool

Pool VLAN146 :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased
addresses
 155.1.146.1          155.1.146.1      - 155.1.146.254     0
```


Note

Storing the DHCP database in flash allows R6 to retain host allocations in the case that the router reboots.

```
Rack1R6#show ip dhcp database
URL      : flash:/bindings
Read     : Never
Written  : Mar 01 2002 01:26 AM
Status   : Last write succeeded. Agent information is up-to-date.
Delay    : 300 seconds
Timeout  : 300 seconds
Failures : 0
Successes: 1
```

Pitfall

Configure the excluded IP address ranges before configuring the DHCP pool itself. If the pool is configure first clients may request and be allocated IP addresses faster than the exclusion is applied.

13.3 DHCP Client

- Configure R1 to obtain its IP address for the link to VLAN 146 dynamically via DHCP.
- The DHCP client-id should be based off of the MAC address of the interface connecting to VLAN 146.

Configuration

```
R1:
interface FastEthernet 0/0
 ip address dhcp client-id FastEthernet 0/0
```

Verification

> **Note**
>
> Enable DHCP packets debugging on the client before requesting the IP address via DHCP.

```
Rack1R1#debug dhcp detail

RAC: Starting DHCP discover on FastEthernet0/0
DHCP: Try 1 to acquire address for FastEthernet0/0
DHCP: allocate request
DHCP: zapping entry in DHC_PURGING state for Fa0/0
DHCP: new entry. add to queue
```

R1 sends a DHCP DISCOVER on the interface with a source IP address of 0.0.0.0. Note that the client-ID contains the MAC address of the FastEthernet interface.

```
DHCP: SDiscover attempt # 1 for entry:
Temp IP addr: 0.0.0.0  for peer on Interface: FastEthernet0/0
Temp  sub net mask: 0.0.0.0
   DHCP Lease server: 0.0.0.0, state: 1 Selecting
   DHCP transaction id: 6AAFAD
   Lease: 0 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:03
   Retry count: 1   Client-ID: 0007.8500.27c0
   Hostname: Rack1R1
DHCP: SDiscover: sending 279 byte length DHCP packet
DHCP: SDiscover 279 bytes
            B'cast on FastEthernet0/0 interface from 0.0.0.0
```


R1 receives the OFFER message from R6, as seen in the Server ID field. Additional DHCP options, such as DNS, are also seen here inside the offer.

```
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Offer
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B059206
DHCP: Scan: Lease Time: 43200
DHCP: Scan: Renewal time: 21600
DHCP: Scan: Rebind time: 37800
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: Scan: Router Option: 155.1.146.6, 155.1.146.4
DHCP: Scan: DNS Name Server Option: 155.1.146.6, 155.1.146.4
DHCP: rcvd pkt source: 155.1.146.6,  destination:  255.255.255.255
   UDP  sport: 43,  dport: 44,  length: 308
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 6AAFAD, secs: 0, flags: 8000
        client: 0.0.0.0, your: 155.1.146.3
        srvr:   0.0.0.0, gw: 0.0.0.0
        options block length: 60

DHCP Offer Message   Offered Address: 155.1.146.3
DHCP: Lease Seconds: 43200    Renewal secs:  21600    Rebind secs:
37800
DHCP: Server ID Option: 155.1.146.6
DHCP: offer received from 155.1.146.6
```

R1 sends a DHCP REQUEST for the IP address it was offered.

```
DHCP: SRequest attempt # 1 for entry:
Temp IP addr: 155.1.146.3  for peer on Interface: FastEthernet0/0
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 155.1.146.6, state: 2 Requesting
   DHCP transaction id: 6AAFAD
   Lease: 43200 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:01
   Retry count: 1   Client-ID: 0007.8500.27c0
   Hostname: Rack1R1
DHCP: SRequest- Server ID option: 155.1.146.6
DHCP: SRequest- Requested IP addr option: 155.1.146.3
DHCP: SRequest placed lease len option: 43200
DHCP: SRequest: 297 bytes
DHCP: SRequest: 297 bytes
            B'cast on FastEthernet0/0 interface from 0.0.0.0
```


R1 receives a DHCP ACK message from the server, and the allocation is complete.

```
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Ack
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B059206
DHCP: Scan: Lease Time: 43200
DHCP: Scan: Renewal time: 21600
DHCP: Scan: Rebind time: 37800
DHCP: Scan: Host Name: Rack1R1
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: Scan: Router Option: 155.1.146.6, 155.1.146.4
DHCP: Scan: DNS Name Server Option: 155.1.146.6, 155.1.146.4
DHCP: rcvd pkt source: 155.1.146.6,  destination:  255.255.255.255
   UDP  sport: 43,  dport: 44,  length: 311
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 6AAFAD, secs: 0, flags: 8000
        client: 0.0.0.0, your: 155.1.146.3
        srvr:   0.0.0.0, gw: 0.0.0.0
        options block length: 63
```

The binding can be verified on the DHCP server as seen below.

```
Rack1R6#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/                Lease expiration
Type
                    Hardware address/
                    User name
155.1.146.3             0100.0785.0027.c0       Mar 01 2002 01:53 PM
Automatic
```

Finally, verify that the agent database stores the bindings in the file on flash.

```
Rack1R6#more flash:bindings
*time* Mar 01 2002 01:58 AM
*version* 2
!IP address     Type  Hardware address   Lease expiration   VRF
155.1.146.3     id    0100.0785.0027.c0  Mar 01 2002 01:53 PM

!IP address      Type     Hardware address  Interface-index


!IP address     Interface-index  Lease expiration      Server IP
address  Hardware address  Vrf
*end*
```


13.4 DHCP Relay

- Configure R6 to allocate IP addresses via DHCP to clients on VLAN 58.
- The clients on VLAN 58 should use R5 as the default gateway.
- Configure SW2's VLAN 58 interface to obtain an IP address via DHCP.
- Base the client-id on the MAC address of its VLAN 58 interface.
- Ensure to avoid IP address conflicts with the existing hosts on VLAN 58.

Configuration

```
R6:
ip dhcp pool VLAN58
 network 155.1.58.0 /24
 default-router 155.1.58.5
!
ip dhcp excluded-address 155.1.58.5

R5:
interface FastEthernet
 ip helper-address 155.1.146.6

SW2:
interface Vlan 58
 ip address dhcp client-id Vlan79
```

Verification

Note

DHCP relay acts as a proxy agent, forwarding all broadcast packets from the clients to the unicast IP address of the actual DHCP server. The server uses a special "giaddr" field in the DHCP packets which contains the IP address of the DHCP relay interface. This field allows the server to select the proper DHCP pool and then unicast a reply back to the DHCP relay agent.

To verify the configuration enable DHCP debugging on the server and request a new IP address on SW2.


```
Rack1R6#debug ip dhcp server packet
Rack1R6#debug ip dhcp server events
```

R6 receives a DHCP DISCOVER packet from the relay IP address (R5) with the Client-ID is based on the MAC address of SW2's VLAN 58 interface.

```
DHCPD: DHCPDISCOVER received from client 0100.1bd4.d47a.42 through
relay 155.1.58.5.
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: remote id 020a00009b01920600000092
  DHCPD: circuit id 00000000
DHCPD: Allocate an address without class information (155.1.58.0)
DHCPD: Adding binding to radix tree (155.1.58.9)
DHCPD: Adding binding to hash tree
```

R6 sends a DHCP OFFER packet to the IP address of the relay (R5).

```
DHCPD: assigned IP address 155.1.58.9 to client 0100.1bd4.d47a.42.
DHCPD: Sending DHCPOFFER to client 0100.1bd4.d47a.42 (155.1.58.9).
DHCPD: unicasting BOOTREPLY for client 001b.d4d4.7a42 to relay
155.1.58.5.
```

SW2 requests the IP address using a REQUEST packet.

```
DHCPD: DHCPREQUEST received from client 0100.1bd4.d47a.42.
DHCPD: Sending notification of ASSIGNMENT:
 DHCPD: address 155.1.58.9 mask 255.255.255.0
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: lease time remaining (secs) = 86400
DHCPD: No default domain to append - abort update
```

The server acknowledges the assignment by sending ACK to the relay.

```
DHCPD: Sending DHCPACK to client 0100.1bd4.d47a.42 (155.1.58.9).
DHCPD: unicasting BOOTREPLY for client 001b.d4d4.7a42 to relay
155.1.58.5.
```

Allocation can be verified by checking the DHCP binding database on R6.

```
Rack1R6#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/                Lease expiration
Type
                Hardware address/
                User name
155.1.58.9          0100.1bd4.d47a.42       Mon DD YYYY 10:21 PM
Automatic
```


13.5 DHCP Host Pools

- Create a DHCP host pool on R6 corresponding to R1's VLAN 146 interface to lease R1 the IP address 155.X.146.1.

Configuration

> **Note**
>
> To find the DHCP Client-ID corresponding to R1 view the binding database on R6. You can easily distinguish a Client-ID from the Hardware Address from additional byte value "01" in the beginning of the string. You may need to release the IP address of R1 before adding a static host pool in R6.

```
Rack1R6#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/                Lease expiration
Type
                Hardware address/
                User name
155.1.58.9          0100.1bd4.d47a.42       Jul 14 2008 10:21 PM
Automatic
155.1.146.1         0100.0c85.96a4.a0       Jul 13 2008 10:35 PM
Automatic

R6:
ip dhcp pool R1_HOST
 client-id 0100.0c85.96a4.a0
 host 155.1.146.1 /24
```


Verification

Note

Issue **no ip address dhcp** command on R1's interface followed by **ip address dhcp client-id** <Interface>. Ensure the **debug dhcp detail** has been enabled on R1 prior to that.

```
Rack1R1#debug dhcp detail

DHCP: DHCP client process started: 10
RAC: Starting DHCP discover on FastEthernet0/0
DHCP: Try 1 to acquire address for FastEthernet0/0
DHCP: allocate request
DHCP: new entry. add to queue
DHCP: SDiscover attempt # 1 for entry:
Temp IP addr: 0.0.0.0  for peer on Interface: FastEthernet0/0
Temp  sub net mask: 0.0.0.0
   DHCP Lease server: 0.0.0.0, state: 1 Selecting
   DHCP transaction id: 62C016
   Lease: 0 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:03
   Retry count: 1   Client-ID: 000c.8596.a4a0
   Hostname: Rack1R1
DHCP: SDiscover: sending 279 byte length DHCP packet
DHCP: SDiscover 279 bytes
            B'cast on FastEthernet0/0 interface from 0.0.0.0
```

In addtion to allocating a static IP address to R1 per the host pool, the DHCP server (R6) also imports the other settings (DNS servers, default routers, etc.) from the more specific subnet-wide DHCP pool.

```
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Offer
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B019206
DHCP: Scan: Lease Time: 86400
DHCP: Scan: Renewal time: 43200
DHCP: Scan: Rebind time: 75600
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: Scan: Router Option: 155.1.146.6, 155.1.146.4
DHCP: Scan: DNS Name Server Option: 155.1.146.6, 155.1.146.4
DHCP: rcvd pkt source: 155.1.146.6,  destination:  255.255.255.255
   UDP  sport: 43,  dport: 44,  length: 308
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 62C016, secs: 0, flags: 8000
        client: 0.0.0.0, your: 155.1.146.1
        srvr:   0.0.0.0, gw: 0.0.0.0
        options block length: 60

DHCP Offer Message   Offered Address: 155.1.146.1
```



```
DHCP: Lease Seconds: 86400    Renewal secs:  43200    Rebind secs:
75600
DHCP: Server ID Option: 155.1.146.6
DHCP: offer received from 155.1.146.6
```

R1 requests the IP address offered from R6.

```
DHCP: SRequest attempt # 1 for entry:
Temp IP addr: 155.1.146.1  for peer on Interface: FastEthernet0/0
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 155.1.146.6, state: 2 Requesting
   DHCP transaction id: 62C016
   Lease: 86400 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:01
   Retry count: 1   Client-ID: 000c.8596.a4a0
   Hostname: Rack1R1
DHCP: SRequest- Server ID option: 155.1.146.6
DHCP: SRequest- Requested IP addr option: 155.1.146.1
DHCP: SRequest placed lease len option: 86400
DHCP: SRequest: 297 bytes
DHCP: SRequest: 297 bytes
            B'cast on FastEthernet0/0 interface from 0.0.0.0
```

The new IP address has been acknowledged by R6.

```
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Ack
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B019206
DHCP: Scan: Lease Time: 86400
DHCP: Scan: Renewal time: 43200
DHCP: Scan: Rebind time: 75600
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: Scan: Router Option: 155.1.146.6, 155.1.146.4
DHCP: Scan: DNS Name Server Option: 155.1.146.6, 155.1.146.4
DHCP: rcvd pkt source: 155.1.146.6,  destination:
255.255.255.2Interface FastEthernet0/0 assigned DHCP address
155.1.146.1, mask 255.255.255.0
55
   UDP  sport: 43,  dport: 44,  length: 308
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 62C016, secs: 0, flags: 8000
        client: 0.0.0.0, your: 155.1.146.1
        srvr:   0.0.0.0, gw: 0.0.0.0
        options block length: 60

DHCP Ack Message
DHCP: Lease Seconds: 86400    Renewal secs:  43200    Rebind secs:
75600
DHCP: Server ID Option: 155.1.146.6-if)#
Rack1R1(config-if)#
DHCP Client Pooling: ***Allocated IP address: 155.1.146.1
```



```
Allocated IP address = 155.1.146.1  255.255.255.0
```

When you're done with R1, verify the binding database on R6.

```
Rack1R6#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/                Lease expiration
Type
                Hardware address/
                User name
155.1.58.9          0100.1bd4.d47a.42       Jul 14 2008 10:21 PM
Automatic
155.1.146.1         0100.0c85.96a4.a0       Infinite
Manual
```

13.6 DHCP on-Demand Pool

- Configure the link between R1 and R3 to use PPP.
- Configure R3 to assign the IP address 155.X.13.1/24 to R1 using IPCP.
- Create a DHCP pool on R1 that imports the subnet, the netmask, and the DNS servers (R4 and R6) from R3 via IPCP.
- Ensure that RIP routing is still functional once this configuration is complete.

Configuration

```
R1:
interface Serial 0/1
 encapsulation ppp
 ip address negotiated
 ppp ipcp mask request
 ppp ipcp dns request
 no peer neighbor-route
!
ip dhcp pool ODAP_POOL
 import all
 origin ipcp
!
router rip
 no validate-update-source

R3:
interface Serial 1/2
 encapsulation ppp
 ip address 155.1.13.3 255.255.255.0
 peer default ip address 155.1.13.1
 ppp ipcp mask 255.255.255.0
 ppp ipcp dns 155.1.146.4 155.1.146.6
 no peer neighbor-route
```


Verification

Note

ODAP allows importing DHCP settings using various methods, one of them being IPCP (Internet Protocol Control Protocol for PPP). IPCP allows importing an IP address along with the subnet mask (while PPP ignores the mask, DHCP can use it) as well as DNS/WINS servers. This is useful in PPPoE deployments, such as with DSL, where the DSL CPE router gets an address assignments through DHCP from the provider, and forwards options such as DNS to the clients behind the CPE.

To verify this check the DHCP pool settings and verify the imported parameters.

```
Rack1R1#show ip dhcp import

Address Pool Name: ODAP_POOL
Domain Name Server(s): 155.1.146.4 155.1.146.6

Rack1R1#show ip dhcp pool

Pool ODAP_POOL :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased
addresses
 155.1.13.1           155.1.13.1       - 155.1.13.254      0
```

Pitfall

When R1 receives RIP updates from R3 they are discarded because they are on different networks. This is due to the fact that R1 uses the host route 155.X.13.1/32, and R3 is outside this range. To resolve this R1 must disable the update source validation feature. Additionally the peer-neighbor-route feature is disabled so that R1 does not inject R3's IP address as a host route into the routing domain.

13.7 DHCP Proxy

- Configure R2 to request an IP address on its serial connection to R3 via IPCP.
- R3 should relay this request via DHCP to R6, and R6 should allocate the IP address 155.X.23.2 to R2.

Configuration

```
R1:
ip route 155.1.23.0 255.255.255.0 155.1.13.3

R2:
interface Serial 0/1
 encapsulation ppp
 ip address negotiated
 no peer neighbor-route

R3:
interface Serial 1/3
 encapsulation ppp
 ip address 155.1.23.3 255.255.255.0
 peer default ip address dhcp
 no peer neighbor-route
!
ip address-pool dhcp-proxy-client
ip dhcp-server 155.1.146.6

R6:
ip dhcp pool R2_SERIAL
 network 155.1.23.0 /24
!
ip dhcp excluded-address 155.1.23.1
ip dhcp excluded-address 155.1.23.3 155.1.23.254
!
ip route 155.1.23.0 255.255.255.0 155.1.146.1
```


Verification

Pitfall

The static route is needed in this exercise because R3's link to R2 isn't in the UP/UP state until IP is actually negotiated. This means that when the proxy request is received at the server, R6 does not have a route back to the relay in order to send the reply back.

Note

DHCP Proxy is an address allocation solution for PPP. When a client requests an IP address via IPCP, the server may issue a DHCP request (act as a proxy) using its own IP address on the PPP interface as the "giaddr" field.

To verify the task, enable debugging of DHCP events in R3 and R6.

```
Rack1R3#debug dhcp detail
Rack1R6#debug ip dhcp server packet
Rack1R6#debug ip dhcp server event
```

Shutdown and no-shutdown R3's connection to R2 and observe the debugging output. Ensure R3 released the IP address before no-shutting the interface.

```
Rack1R3#
```

R3 sends a DHCP discover packet to the globally configured DHCP server as a proxy IPCP address request.

```
DHCP: proxy allocate request
DHCP: zapping entry in DHC_PURGING state for Se1/3
DHCP: deleting entry 64F20E80 155.1.23.2 from list
Temp IP addr: 155.1.23.2  for peer on Interface: Serial1/3
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 155.1.146.6, state: 9 Purging
   DHCP transaction id: 152B
   Lease: 86400 secs,  Renewal: 43200 secs,  Rebind: 75600 secs
   Next timer fires after: 00:00:21
   Retry count: 0   Client-ID: cisco-155.1.23.3-Serial1/3
   Client-ID hex dump: 636973636F2D3135352E312E32332E33
                       2D53657269616C312F33
DHCP: new entry. add to queue, interface Serial1/3
DHCP: SDiscover attempt # 1 for entry:
```



```
Rack1R3(config-if)#
Temp IP addr: 0.0.0.0  for peer on Interface: Serial1/3
Temp  sub net mask: 0.0.0.0
   DHCP Lease server: 0.0.0.0, state: 1 Selecting
   DHCP transaction id: 152C
   Lease: 0 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:04
   Retry count: 1   Client-ID: cisco-155.1.23.3-Serial1/3
   Client-ID hex dump: 636973636F2D3135352E312E32332E33
                       2D53657269616C312F33
   Hostname: Rack1R3
DHCP: SDiscover: sending 292 byte length DHCP packet
DHCP: SDiscover 292 bytes
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/3, changed
state to up
```

R3 receives an OFFER from the DHCP server with the client IP address.

```
DHCP: XID MATCH in dhcpc_for_us()
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Offer
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B019206
DHCP: Scan: Lease Time: 86400
DHCP: Scan: Renewal time: 43200
DHCP: Scan: Rebind time: 75600
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: rcvd pkt source: 155.1.146.6,  destination:  155.1.23.3
   UDP  sport: 43,  dport: 43
Rack1R3(config,  length: 308
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 152C, secs: 0, flags: 0
        client: 0.0.0.0, your: 155.1.23.2
        srvr:   0.0.0.0, gw: 155.1.23.3
        options block length: 60

DHCP Offer Message   Offered Address: 155.1.23.2
DHCP: Lease Seconds: 86400    Renewal secs:  43200    Rebind secs:
75600
DHCP: Server ID Option: 155.1.146.6
DHCP: offer received from 155.1.146.6
DHCP: SRequest attempt # 1 for entry:
Temp IP addr: 155.1.23.2  for peer on Interface: Serial1/3
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 155.1.146.6, state: 2 Requesting
   DHCP transaction id: 152C
   Lease: 86400 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:03
   Retry count: 1   Client-ID: cisco-155.1.23.3-Serial1/3
   Client-ID hex dump: 636973636F2D3135352E312E32332E33
                       2D53657269616C312F33
   Hostname: Rack1R3
```


A request is sent for the IP address offered by the DHCP server.

```
DHCP: SRequest- Server ID option: 155.1.146.6
DHCP: SRequest- Requested IP addr option: 155.1.23.2
DHCP: SRequest placed lease len option: 86400
DHCP: SRequest: 310 bytes
DHCP: SRequest: 310 bytes
DHCP: XID MATCH in dhcpc_for_us()
DHCP: Received a BOOTREP pkt
DHCP: Scan: Message type: DHCP Ack
DHCP: Scan: Server ID Option: 155.1.146.6 = 9B019206
DHCP: Scan: Lease Time: 86400
DHCP: Scan: Renewal time: 43200
DHCP: Scan: Rebind time: 75600
DHCP: Scan: Subnet Address Option: 255.255.255.0
DHCP: rcvd pkt source: 155.1.146.6,  destination:  155.1.23.3
   UDP  sport: 43,  dport: 43,  length: 308
   DHCP op: 2, htype: 1, hlen: 6, hops: 0
   DHCP server identifier: 155.1.146.6
        xid: 152C, secs: 0, flags: 0
        client: 0.0.0.0, your: 155.1.23.2
        srvr:   0.0.0.0, gw: 155.1.23.3
        options block length: 60
```

The server ACKed the request and the IP address is now assigned to R2.

```
DHCP Ack Message
DHCP: Lease Seconds: 86400    Renewal secs:  43200    Rebind secs:
75600
DHCP: Server ID Option: 155.1.146.6
DHCP Proxy Client Pooling: ***Allocated IP address: 155.1.23.2
```

The debugging output at R6 is as follows. The server receives a DHCP DISCOVER message from the relay, R3.

```
Rack1R6#
DHCPD: DHCPDISCOVER received from client
0063.6973.636f.2d31.3535.2e31.2e32.332e.332d.5365.7269.616c.312f.33
through relay 155.1.23.3.
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr 0007.eb49.04c1
  DHCPD: remote id 020a00009b01920600000092
  DHCPD: circuit id 00000000
DHCPD: Allocate an address without class information (155.1.23.0)
DHCPD: Adding binding to radix tree (155.1.23.2)
DHCPD: Adding binding to hash tree
DHCPD: assigned IP address 155.1.23.2 to client
0063.6973.636f.2d31.3535.2e31.2e32.332e.332d.5365.7269.616c.312f.33.
```


The server sends a DHCP OFFER message to the relay.

```
DHCPD: Sending DHCPOFFER to client
0063.6973.636f.2d31.3535.2e31.2e32.332e.332d.5365.7269.616c.312f.33
(155.1.23.2).
DHCPD: unicasting BOOTREPLY for client 0007.eb49.04c1 to relay
155.1.23.3.
```

The client requests the IP address and the server acknowledges the request.

```
DHCPD: DHCPREQUEST received from client
0063.6973.636f.2d31.3535.2e31.2e32.332e.332d.5365.7269.616c.312f.33.
DHCPD: Sending notification of ASSIGNMENT:
 DHCPD: address 155.1.23.2 mask 255.255.255.0
  DHCPD: htype 1 chaddr 0007.eb49.04c1
  DHCPD: lease time remaining (secs) = 86400
DHCPD: No default domain to append - abort update
DHCPD: Sending DHCPACK to client
0063.6973.636f.2d31.3535.2e31.2e32.332e.332d.5365.7269.616c.312f.33
(155.1.23.2).
DHCPD: unicasting BOOTREPLY for client 0007.eb49.04c1 to relay
155.1.23.3.
DHCPD: checking for expired leases.
DHCPD: checking for expired leases.
DHCPD: writing bindings to flash:/bindings.

Rack1R6#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/                Lease expiration
Type
                Hardware address/
                User name
155.1.23.2          0063.6973.636f.2d31.    Jul 09 2008 10:05 PM
Automatic
                    3535.2e31.2e32.332e.
                    332d.5365.7269.616c.
                    312f.33
```


13.8 DHCP Information Option

- Configure R5 to insert the DHCP information option field into DHCP requests forwarded to R6 from clients on VLAN 58.
- R5 should set the subscriber-id string for these requests to “VLAN58”.
- Configure R6 to allocate IP addresses based on Option 82 and assign a static IP address to SW2 based on the information option contents.

Configuration

```
R5:
ip dhcp relay information option
!
interface FastEthernet0/0
 ip dhcp relay information option subscriber-id VLAN58

R6:
ip dhcp class TEST
   relay agent information
      relay-information hex
020c020a00009b013a05000000000606564c414e3538
!
ip dhcp pool VLAN58
   class TEST
     address range 155.1.58.8 155.1.58.8
```


Verification

Note

The DHCP information option format may vary from vendor to vendor and even from platform to platform. Therefore, IOS matches a variable hex string against an information option (you can use wildcard "*" and bitmask to match Option 82 ranges, e.g. remote-id with multiple subscriber-ids) to define a DHCP class.

DHCP debugging commands will not reveal the information option passed in relay packets. To find out the actual value use the detailed IP packets dumping procedere as follows.

First, create an access-list that matches only DHCP packets.

```
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#access-list 100 permit udp any any eq bootps
```

Then enable DHCP packets debugging using an access-list 100 and hidden **dump** option, and configure SW2 to release/request an IP address via DHCP.

```
Rack1R6#debug ip packet detail 100 dump
IP packet debugging is on (detailed) (dump) for access list 100

Rack1SW2#conf t
Rack1SW2(config)#interface Vlan58
Rack1SW2(config-if)#shutdown
Rack1SW2(config-if)#no shutdown
```

Look at the packet dumps below. The two captured packets are DHCP DISCOVER and DHCP REQUEST packets sent from SW2. Note that the information option (you can locate it by looking for ASCII subscriber-id sring "VLAN58") is located at the very end of UDP packet.

The information option starts with a byte having decimal value 82 (hex value 0x52, information option number), followed by the total option length (e.g. 0x16 = 22 bytes). After that, suboptions follow (e.g. remote-id, circuit-id). The suboptions are always TLVs, i.e. type-length-value blocks, with type and length being one-byte quantities. Cisco may add sub-TLVs into each suboption to distinguish remote-id type etc.

Look at the byte-string extracted from the captured DHCP packet.

|5216|020C|020A|00009B013A0500000000|0606|564C414E35|

This is DHCP option 82 (0x52) of total length 22 (0x16). The first suboption is remote-id (0x02 – suboption type) of total length 12 (0xC). The remote-id suboption starts with yet another TLV (type 0x02 length 0x0A) defining remote-id type, followed by 10 bytes of remote-id value. The next suboption is type 0x06 and length 0x06, which represents ASCII value assigned to R5 VLAN58 interface.

However, in a DHCP class, you need to match either a complete Option 82 or its part, using either a wildcard "*" or bitmask. In our case, the whole content of Option 82 (starting after initial 0x52 0x16) is matched by the class statement.

```
Rack1R6#
<snip>
07C57820: 63825363 35010236 049B0192 06330400  c.Sc5..6.....3..
07C57830: 0151803A 040000A8 C03B0400 01275001  .Q.:...(@;...'P.
07C57840: 04FFFFFF 0003049B 013A0552 16020C02  .........:.R....
07C57850: 0A00009B 013A0500 00000006 06564C41  .....:.......VLA
07C57860: 4E3538FF                             N58.

IP: tableid=0, s=155.1.58.5 (FastEthernet0/0.146), d=155.1.146.6
(FastEthernet0/0.146), routed via RIB
IP: s=155.1.58.5 (FastEthernet0/0.146), d=155.1.146.6
(FastEthernet0/0.146), len 350, rcvd 3
    UDP src=67, dst=67

07E28FE0:                       000F 24093CE0            ..$.<`
07E28FF0: 000D2846 8F210800 4500015E 002E0000  ..(F.!..E..^....
07E29000: FD11BA52 9B013A05 9B019206 00430043  }.:R..:......C.C
07E29010: 014A1983 01010601 000010F3 00008000  .J.........s....
07E29020: 00000000 00000000 00000000 9B013A05  ..............:.
07E29030: 001BD4D4 7A420000 00000000 00000000  ..TTzB..........
07E29040: 00000000 00000000 00000000 00000000  ................
07E29050: 00000000 00000000 00000000 00000000  ................
07E29060: 00000000 00000000 00000000 00000000  ................
07E29070: 00000000 00000000 00000000 00000000  ................
07E29080: 00000000 00000000 00000000 00000000  ................
07E29090: 00000000 00000000 00000000 00000000  ................
07E290A0: 00000000 00000000 00000000 00000000  ................
07E290B0: 00000000 00000000 00000000 00000000  ................
07E290C0: 00000000 00000000 00000000 00000000  ................
07E290D0: 00000000 00000000 00000000 00000000  ................
07E290E0: 00000000 00000000 00000000 00000000  ................
07E290F0: 00000000 00000000 00000000 00000000  ................
07E29100: 63825363 35010339 0204803D 0701001B  c.Sc5..9...=....
07E29110: D4D47A42 36049B01 92063204 9B013A08  TTzB6.....2...:.
07E29120: 33040001 51800C08 5261636B 31535732  3...Q...Rack1SW2
07E29130: 37080106 0F2C0321 962B3401 03521602  7....,.!.+4..R..
07E29140: 0C020A00 009B013A 05000000 00060656  .......:.......V
```

```
07E29150: 4C414E35 38FF                          LAN58.

IP: tableid=0, s=155.1.146.6 (local), d=155.1.58.5
(FastEthernet0/0.146), routed via FIB
IP: s=155.1.146.6 (local), d=155.1.58.5 (FastEthernet0/0.146), len 342,
sending
    UDP src=67, dst=67

07E28970:                   45000156 001B0000          E..V....
07E28980: FF11B86D 9B019206 9B013A05 00430043  ..8m......:..C.C
07E28990: 0142546B 02010600 000010F3 00008000  .BTk.......s....
07E289A0: 00000000 9B013A08 00000000 9B013A05  ......:.......:.
07E289B0: 001BD4D4 7A420000 00000000 00000000  ..TTzB..........
07E289C0: 00000000 00000000 00000000 00000000  ................
07E289D0: 00000000 00000000 00000000 00000000  ................
07E289E0: 00000000 00000000 00000000 00000000  ................
07E289F0: 00000000 00000000 00000000 00000000  ................
07E28A00: 00000000 00000000 00000000 00000000  ................
07E28A10: 00000000 00000000 00000000 00000000  ................
07E28A20: 00000000 00000000 00000000 00000000  ................
07E28A30: 00000000 00000000 00000000 00000000  ................
07E28A40: 00000000 00000000 00000000 00000000  ................
07E28A50: 00000000 00000000 00000000 00000000  ................
07E28A60: 00000000 00000000 00000000 00000000  ................
07E28A70: 00000000 00000000 00000000 00000000  ................
07E28A80: 63825363 35010536 049B0192 06330400  c.Sc5..6.....3..
07E28A90: 0151803A 040000A8 C03B0400 0127500C  .Q.:...(@;...'P.
07E28AA0: 08526163 6B315357 320104FF FFFF0003  .Rack1SW2.......
07E28AB0: 049B013A 05521602 0C020A00 009B013A  ...:.R.........:
07E28AC0: 05000000 00060656 4C414E35 38FF      .......VLAN58.
```

Observe how R6 allocates a DHCP address based on the class. Enable DHCP server debugging and let SW2 renew its IP address.

```
Rack1R6#debug ip dhcp server packet
Rack1R6#debug ip dhcp server event
Rack1R6#debug ip dhcp server class

Rack1SW2#conf t
Rack1SW2(config)#interface Vlan58
Rack1SW2(config-if)#shutdown
Rack1SW2(config-if)#no shutdown

Rack1R6#
DHCPD: DHCPRELEASE message received from client 0100.1bd4.d47a.42
(155.1.58.8).
DHCPD: Sending notification of TERMINATION:
 DHCPD: address 155.1.58.8 mask 255.255.255.0
 DHCPD: reason flags: RELEASE
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: lease time remaining (secs) = 84898
DHCPD: returned 155.1.58.8 to address pool VLAN58.
DHCPD: checking for expired leases.
DHCPD: DHCPRELEASE message received from client 0100.1bd4.d47a.42
(155.1.58.8).
```



```
DHCPD: Finding a relay for client 0100.1bd4.d47a.42 on interface
FastEthernet0/0.146.
```

Note that remote-id and circuit-id shown in DHCP debugging output do not represent the information received from R5.

```
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: remote id 020a00009b01920600000092
  DHCPD: circuit id 00000000
DHCPD: Sending notification of DISCOVER:
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: remote id 020a00009b01920600000092
  DHCPD: circuit id 00000000
DHCPD: DHCPDISCOVER received from client 0100.1bd4.d47a.42 through
relay 155.1.58.5.
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: remote id 020a00009b01920600000092
  DHCPD: circuit id 00000000
```

The configured DHCP class matched the information option received, and therefore the configured address subrange is used for allocation.

```
DHCPD: Class 'TEST' matched by default
DHCPD: Searching for a match to 'relay-information
020c020a00009b013a05000000000606564c414e3538' in class TEST
DHCPD: input pattern 'relay-information
020c020a00009b013a05000000000606564c414e3538' matches class TEST
DHCPD: input matches class TEST
DHCPD: Adding binding to radix tree (155.1.58.8)
DHCPD: Adding binding to hash tree
DHCPD: assigned IP address 155.1.58.8 to client 0100.1bd4.d47a.42.
DHCPD: Sending DHCPOFFER to client 0100.1bd4.d47a.42 (155.1.58.8).
DHCPD: unicasting BOOTREPLY for client 001b.d4d4.7a42 to relay
155.1.58.5.
DHCPD: DHCPREQUEST received from client 0100.1bd4.d47a.42.
DHCPD: Sending notification of ASSIGNMENT:
 DHCPD: address 155.1.58.8 mask 255.255.255.0
  DHCPD: htype 1 chaddr 001b.d4d4.7a42
  DHCPD: lease time remaining (secs) = 86400
DHCPD: No default domain to append - abort update
DHCPD: Sending DHCPACK to client 0100.1bd4.d47a.42 (155.1.58.8).
DHCPD: unicasting BOOTREPLY for client 001b.d4d4.7a42 to relay
155.1.58.5.
```


13.9 DHCP Authorized ARP

- Configure R6 to update the ARP cache on its VLAN 146 interface with the information from the DHCP binding database.
- Ensure that R6 does not accept any ARP mappings on its VLAN 146 interface other than those learned from the DHCP binding database.

Configuration

```
R6:
ip dhcp pool VLAN146
 update arp
!
ip dhcp pool R1_HOST
 update arp
!
interface FastEthernet 0/0.146
 arp authorize

R6:
!
! R4's MAC address is statically mapped
!
arp 155.1.146.4 0007.0e9c.dac2 arpa
```


Verification

Note

The Update ARP feature allows the DHCP process to populate the ARP cache with the DHCP-based entries. Authorized ARP enabled at interface level will disable any dynamic ARP learning on that interface. Therefore a static ARP entry for R4 is needed to reach the router (which has static IP address).

```
Rack1R6#ping 155.1.146.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

Rack1R6#ping 155.1.146.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

When the static ARP entry for R4 is removed it can't be reached anymore since ARP learning is disabled.

```
Rack1R6(config)#no arp 155.1.146.4 0007.0e9c.dac2 arpa
Rack1R6(config)#do ping 155.1.146.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```


13.10 IP SLA

- Configure R6 to automatically monitor BB1's availability using ping and telnet every one minute.

Configuration

```
R6:
ip sla monitor 1
 type echo protocol ipIcmpEcho 54.1.1.254
 timeout 1000
 frequency 1
!
ip sla monitor schedule 1 life forever start-time now
!
ip sla monitor 2
 type tcpConnect dest-ipaddr 54.1.1.254 dest-port 23 control disable
 timeout 5000
!
ip sla monitor schedule 2 life forever start-time now
```


Verification

```
Rack1R6#show ip sla monitor statistics 1
Round trip time (RTT)   Index 1
      Latest RTT: 24 ms
Latest operation start time: 00:00:51.921 UTC Wed Jul 9 2008
Latest operation return code: OK
Number of successes: 117
Number of failures: 0
Operation time to live: Forever

Rack1R6#show ip sla monitor statistics 2
Round trip time (RTT)   Index 2
      Latest RTT: 28 ms
Latest operation start time: 00:00:04.011 UTC Wed Jul 9 2008
Latest operation return code: OK
Number of successes: 2
Number of failures: 0
Operation time to live: Forever
```


13.11 Object Tracking

- Create two enhanced objects on R6 to track the state of previously created IP SLA operations.
- Create a third enhanced object on R6 that tracks the first two objects; this third object should only be “up” if both of the first two objects are “up”.
- The first two objects should wait 10 seconds before reporting their state from up to down, or from down to up.

Configuration

```
R6:
track 1 rtr 1
 delay down 10 up 10
!
track 2 rtr 2
 delay down 10 up 10
!
track 3 list boolean and
 object 1
 object 2
```

Verification

> **Note**
>
> In this example a Boolean AND operation is used to calculate the resulting state of the two member objects. Since AND only returns TRUE if all inputs are TRUE, the state of object 3 is only “up” if both objects 1 and 2 are “up”.
>
> In addition to SLA operations, interface states or IP routes can also be tracked. For IP routes it is possible to track routing metrics and see when they cross thresholds.
>
> To verify this configuration display state of the tracked objects as follows.

```
Rack1R6#show track
Track 1
  Response Time Reporter 1 state
  State is Up
    1 change, last change 00:04:08
  Delay up 10 secs, down 10 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 24
  Tracked by:
    Track-list 3
```



```
Track 2
  Response Time Reporter 2 state
  State is Up
    1 change, last change 00:02:54
  Delay up 10 secs, down 10 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 28
  Tracked by:
    Track-list 3
Track 3
  List boolean and
  Boolean AND is Up
    2 changes, last change 00:02:27
    object 1 Up
    object 2 Up
```

Stop ICMP packets coming back from BB1 and see how that effects tracked objects.

```
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#access-list 100 deny icmp any any
Rack1R6(config)#access-list 100 permit ip any any
Rack1R6(config)#interface Serial 0/0
Rack1R6(config-if)#ip access-group 100 in

Rack1R6#show track
Track 1
  Response Time Reporter 1 state
  State is Down
    2 changes, last change 00:06:19
  Delay up 10 secs, down 10 secs
  Latest operation return code: Timeout
  Tracked by:
    Track-list 3
Track 2
  Response Time Reporter 2 state
  State is Up
    1 change, last change 00:11:35
  Delay up 10 secs, down 10 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 100
  Tracked by:
    Track-list 3
Track 3
  List boolean and
  Boolean AND is Down
    3 changes, last change 00:06:18
    object 1 Down
    object 2 Up
```


13.12 HSRP

- Configure HSRP on R4 and R6 so that hosts on VLAN 146 can use the virtual IP address 155.X.146.254 as their default gateway.
- Ensure that R6 is the active physical gateway unless it loses connectivity to VLAN 146.
- Set the hello interval to 1 second and dead interval to 3 seconds for faster failed gateway detection.
- Authenticate HSRP message exchanges using an MD5 hash of the key CISCO.

Configuration

```
R6:
interface FastEthernet 0/0.146
 standby 146 ip 155.1.146.254
 standby 146 timers 1 3
 standby 146 preempt
 standby 146 authentication md5 key-string CISCO
 standby 146 name VLAN146
 standby 146 priority 110

R4:
interface FastEthernet 0/1
 standby 146 ip 155.1.146.254
 standby 146 timers 1 3
 standby 146 preempt
 standby 146 authentication md5 key-string CISCO
 standby 146 name VLAN146
```


Verification

> **Note**
>
> Issue the **show standby** command to ensure that R6 is the active HSRP gateway, and to verify the backup gateway activation. Note the last octet of the virtual HSRP MAC address which represents the group number (shown in hex).

```
Rack1R6#show standby
FastEthernet0/0.146 - Group 146
  State is Active
    2 state changes, last state change 00:10:37
  Virtual IP address is 155.1.146.254
  Active virtual MAC address is 0000.0c07.ac92
    Local virtual MAC address is 0000.0c07.ac92 (v1 default)
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.539 secs
  Authentication MD5, key-string "CISCO"
  Preemption enabled
  Active router is local
  Standby router is 155.1.146.4, priority 100 (expires in 2.058 sec)
  Priority 110 (configured 110)
  IP redundancy name is "VLAN146" (cfgd)

Rack1R6(config)#interface fastEthernet 0/0.146
Rack1R6(config-subif)#shutdown

Rack1R4#show standby
Ethernet0/1 - Group 146
  State is Active
    2 state changes, last state change 00:00:35
  Virtual IP address is 155.1.146.254
  Active virtual MAC address is 0000.0c07.ac92
    Local virtual MAC address is 0000.0c07.ac92 (v1 default)
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.216 secs
  Authentication MD5, key-string "CISCO"
  Preemption enabled
  Active router is local
  Standby router is unknown
  Priority 100 (default 100)
  IP redundancy name is "VLAN146" (cfgd)
```


13.13 VRRP

- Configure VRRP on R4 and R6 so that hosts on VLAN 146 can use the virtual IP address 155.X.146.253 as their default gateway.
- Ensure that R4 is the active physical gateway unless it loses connectivity to VLAN 146.
- Set the hello interval to 3 seconds, and authenticate messages using the clear-text password CISCO.

Configuration

```
R6:
interface FastEthernet 0/0.146
 vrrp 146 ip 155.1.146.253
 vrrp 146 timers advertise 3
 vrrp 146 authentication CISCO

R4:
interface FastEthernet 0/1
 vrrp 146 ip 155.1.146.253
 vrrp 146 timers advertise 3
 vrrp 146 authentication CISCO
 vrrp 146 priority 110
```

Verification

> **Note**
>
> Verify the status of the virtual router and check the backup activation with the **`show vrrp`** command.

```
Rack1R6#show vrrp
FastEthernet0/0.146 - Group 146
  State is Backup
  Virtual IP address is 155.1.146.253
  Virtual MAC address is 0000.5e00.0192
  Advertisement interval is 3.000 sec
  Preemption enabled
  Priority is 100
  Authentication text "CISCO"
  Master Router is 155.1.146.4, priority is 110
  Master Advertisement interval is 3.000 sec
  Master Down interval is 9.609 sec (expires in 7.709 sec)
```



```
Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface FastEthernet 0/1
Rack1R4(config-if)#shutdown

Rack1R6#show vrrp
FastEthernet0/0.146 - Group 146
  State is Master
  Virtual IP address is 155.1.146.253
  Virtual MAC address is 0000.5e00.0192
  Advertisement interval is 3.000 sec
  Preemption enabled
  Priority is 100
  Authentication text "CISCO"
  Master Router is 155.1.146.6 (local), priority is 100
  Master Advertisement interval is 3.000 sec
  Master Down interval is 9.609 sec
```


13.14 GLBP

- Configure GLBP for hosts on VLAN 146 so that first-hop redundancy and load balancing occur between R4 and R6.
- Ensure R6 is elected as the Active Virtual Gateway (AVG), and both R4 and R6 are used as Active Virtual Forwarders (AVFs) for the virtual IP address 155.X.146.252.
- Use a 1 second hello interval and a 3 second hold time.
- Distribute the traffic load between R4 and R6 in 2:1 ratio.
- Authenticate GLBP UDP packet exchange using a secure method based on a password value of CISCO.

Configuration

```
R6:
interface FastEthernet 0/0.146
 glbp 146 ip 155.1.146.252
 glbp 146 timers 1 3
 glbp 146 priority 110
 glbp 146 preempt
 glbp 146 weighting 10
 glbp 146 name VLAN146
 glbp 146 authentication md5 key-string CISCO
 glbp 146 load-balancing weighted

R4:
interface FastEthernet 0/1
 glbp 146 ip 155.1.146.252
 glbp 146 timers 1 3
 glbp 146 preempt
 glbp 146 weighting 20
 glbp 146 name VLAN146
 glbp 146 authentication md5 key-string CISCO
 glbp 146 load-balancing weighted
```


Verification

Note

GLBP allows multiple active physical gateways serve the same virtual gateway IP address. The AVG (active virtual gateway) responds to ARP requests sent to to virtual gateway IP address, and replies with different virtual MAC addresses that correspond to multiple active virtual forwarders (AVFs). Both the AVG and AVFs are redundant, i.e. if a primary physical router representing the AVG or an AVF fails, another physical router will take its role.

To verify this first check status of the GLBP group and virtual forwarders state with the **`show glbp`** command.

```
Rack1R4#show glbp
FastEthernet0/1 - Group 146
```

R4 is the STANDBY AVG, since R6 has a higher priority.

```
State is Standby
    4 state changes, last state change 00:02:31
  Virtual IP address is 155.1.146.252
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.864 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Authentication MD5, key-string "CISCO"
  Preemption enabled, min delay 0 sec
  Active is 155.1.146.6, priority 110 (expires in 2.268 sec)
  Standby is local
  Priority 100 (default)
  Weighting 20 (configured 20), thresholds: lower 1, upper 20
  Load balancing: weighted
  IP redundancy name is "VLAN146"
  Group members:
```

There are two members in the GLBP group, and R6 (the other member) is authenticated

```
    0007.0e9c.dac2 (155.1.146.4) local
    0012.019a.27c0 (155.1.146.6) authenticated
  There are 2 forwarders (1 active)
```


It is important to understand that each AVF is in fact a virtual gateway consisting of primary and backup physical gateways. From the output below you can see that R4 is the primary physical gateway for Forwarder 1 and is in "Listen" (backup state) for Forwarder 2. However, for each AVF only the primary physical gateway would forwards packets, the other is listening to become active for that AVF if the primary fails. The load between AVFs is distributed based on the amount of times the AVG will respond to ARP requests for the IP address of the virtual gateway with the virtual MAC of a particular AVF. The load is proportional to the weight assigned to an AVF if the load-balancing scheme is "weighted".

```
  Forwarder 1
    State is Active
      1 state change, last state change 00:25:34
    MAC address is 0007.b400.9201 (default)
    Owner ID is 0007.0e9c.dac2
    Preemption enabled, min delay 30 sec
    Active is local, weighting 20
    Arp replies sent: 12
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.9202 (learnt)
    Owner ID is 0012.019a.27c0
    Time to live: 14399.896 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 155.1.146.6 (primary), weighting 10 (expires in 2.892
sec)

Rack1R6#show glbp
State is Active
    1 state change, last state change 00:03:29
  Virtual IP address is 155.1.146.252
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.956 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Authentication MD5, key-string "CISCO"
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is 155.1.146.4, priority 100 (expires in 2.395 sec)
  Priority 110 (configured)
  Weighting 10 (configured 10), thresholds: lower 1, upper 10
  Load balancing: weighted
  IP redundancy name is "VLAN146"
  Group members:
    0007.0e9c.dac2 (155.1.146.4) authenticated
    0012.019a.27c0 (155.1.146.6) local
  There are 2 forwarders (0 active)
  Forwarder 1
    State is Listen
    MAC address is 0007.b400.9201 (learnt)
    Owner ID is 0007.0e9c.dac2
    Redirection enabled, 599.335 sec remaining (maximum 600 sec)
    Time to live: 14399.335 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
```



```
    Active is 155.1.146.4 (primary), weighting 20 (expires in 2.335
sec)
  Forwarder 2
    State is Active
      1 state change, last state change 00:00:00
    MAC address is 0007.b400.9202 (default)
    Owner ID is 0012.019a.27c0
    Redirection enabled
    Preemption enabled, min delay 30 sec
    Active is local, weighting 10
```

The below output illustrates how the AVG responds with virtual MACs based on a client's (R1) ARP requests to the GLBP virtual IP address. Note that virtual MAC 0007.b400.9201 is AVF 1 (R4), and 0007.b400.9202 is AVF 2 (R6).

```
Rack1R1#ping 155.1.146.252

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.252, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms

Rack1R1#show ip arp 155.1.146.252
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.252           0   0007.b400.9201  ARPA   FastEthernet0/0

Rack1R1#clear arp-cache
Rack1R1#ping 155.1.146.252

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.252, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms

Rack1R1#sh ip arp 155.1.146.252
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.252           0   0007.b400.9202  ARPA   FastEthernet0/0

Rack1R1#clear arp-cache
Rack1R1#ping 155.1.146.252

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.252, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

Rack1R1#sh ip arp 155.1.146.252
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.252           0   0007.b400.9201  ARPA   FastEthernet0/0
```



```
Rack1R1#clear arp-cache
Rack1R1#ping 155.1.146.252

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.252, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/7/24 ms

Rack1R1#sh ip arp 155.1.146.252
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  155.1.146.252          0   0007.b400.9201  ARPA   FastEthernet0/0
```

Note that the AVF 1 (R4) virtual MAC appears more often than the AVF 2 MAC address, following the configured 2:1 distribution ratio.

13.15 Router Redundancy and Object Tracking

- R6 should be the active HSRP gateway as long as it can ping BB1 and reach it through telnet, but give up the active role once it loses either telnet or ping reachability to BB1.
- Ensure R4 is the active VRRP gateway when it sees the prefix 30.0.0.0/16 received via IGP from BB3.

Configuration

```
R6:
interface FastEthernet 0/0.146
 standby 146 track 3 decrement 20

R4:
track 1 ip route 30.0.0.0/16 reachability
!
interface FastEthernet 0/1
 vrrp 146 track 1 decrement 20
```


Verification

Note

Originally HSRP was only able to track an interface state (e.g. uplink interface) and decrease its priority when the interface failed. As illustrated in the *IP Routing* section of this series, the interface state of a link is not always a good indication of end-to-end connectivity out the link. Combining enhanced object tracking with first hop redundancy extends functionality to any application that can be tracked through IP SLA in addition to basic routing checks.

The below output verifies how HSRP tracks the object on R6.

```
Rack1R6#show track 3
Track 3
  List boolean and
  Boolean AND is Up
    2 changes, last change 00:02:15
    object 1 Up
    object 2 Up
  Tracked by:
    HSRP FastEthernet0/0.146 146

Rack1R6#show standby
FastEthernet0/0.146 - Group 146
  State is Active
    1 state change, last state change 00:03:36
  Virtual IP address is 155.1.146.254
  Active virtual MAC address is 0000.0c07.ac92
    Local virtual MAC address is 0000.0c07.ac92 (v1 default)
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.940 secs
  Authentication MD5, key-string "CISCO"
  Preemption enabled
  Active router is local
  Standby router is 155.1.146.4, priority 100 (expires in 2.395 sec)
  Priority 110 (configured 110)
    Track object 3 state Up decrement 20
  IP redundancy name is "VLAN146" (cfgd)
```


Shutting the link to BB1 down causes the enhanced objects to go down. This in turn causes R6 to decrement its HSRP priority and stop forwarding packets for that HSRP group.

```
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface Serial 0/0
Rack1R6(config-if)#shutdown

HSRP: Fa0/0.146 Grp 146 Track 3 object changed, state Up -> Down
HSRP: Fa0/0.146 Grp 146 Priority 110 -> 90
HSRP: Fa0/0.146 Grp 146 Ignoring Coup (100/155.1.146.4 <
110/155.1.146.6)
HSRP: Fa0/0.146 Grp 146 Hello  in  155.1.146.4 Active  pri 100 vIP
155.1.146.254
HSRP: Fa0/0.146 Grp 146 Active router is 155.1.146.4, was local
HSRP: Fa0/0.146 Grp 146 Standby router is unknown, was 155.1.146.4
HSRP: Fa0/0.146 Grp 146 Active: g/Hello rcvd from higher pri Active
router (100/155.1.146.4)
HSRP: Fa0/0.146 Grp 146 Active -> Speak
%HSRP-5-STATECHANGE: FastEthernet0/0.146 Grp 146 state Active -> Speak
HSRP: Fa0/0.146 Grp 146 Redundancy "VLAN146" state Active -> Speak
HSRP: Fa0/0.146 Grp 146 Speak: d/Standby timer expired (unknown)
HSRP: Fa0/0.146 Grp 146 Standby router is local
HSRP: Fa0/0.146 Grp 146 Speak -> Standby
%HSRP-5-STATECHANGE: FastEthernet0/0.146 Grp 146 state Speak -> Standby
HSRP: Fa0/0.146 Grp 146 Redundancy "VLAN146" state Speak -> Standby

Rack1R6#show standby
FastEthernet0/0.146 - Group 146
  State is Standby
    3 state changes, last state change 00:02:25
  Virtual IP address is 155.1.146.254
  Active virtual MAC address is 0000.0c07.ac92
    Local virtual MAC address is 0000.0c07.ac92 (v1 default)
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.595 secs
  Authentication MD5, key-string "CISCO"
  Preemption enabled
  Active router is 155.1.146.4, priority 100 (expires in 2.334 sec)
  Standby router is local
  Priority 90 (configured 110)
    Track object 3 state Down decrement 20
  IP redundancy name is "VLAN146" (cfgd)
```


The verification procedure is repeated with VRRP on R4 through the local tracking object.

```
Rack1R4#show track 1
Track 1
  IP route 30.0.0.0 255.255.0.0 reachability
  Reachability is Up (RIP)
    2 changes, last change 00:00:09
  First-hop interface is Ethernet0/0
  Tracked by:
    VRRP FastEthernet0/1 146

Rack1R4#show vrrp
FastEthernet0/1 - Group 146
  State is Master
  Virtual IP address is 155.1.146.253
  Virtual MAC address is 0000.5e00.0192
  Advertisement interval is 3.000 sec
  Preemption enabled
  Priority is 110
    Track object 1 state Up decrement 20
  Authentication text "CISCO"
  Master Router is 155.1.146.4 (local), priority is 110
  Master Advertisement interval is 3.000 sec
  Master Down interval is 9.570 sec

Rack1R4#conf t
Rack1R4(config)#inteface FastEthernet 0/0
Rack1R4(config)#shutdown

Rack1R4(config)#do show vrrp
Ethernet0/1 - Group 146
  State is Backup
  Virtual IP address is 155.1.146.253
  Virtual MAC address is 0000.5e00.0192
  Advertisement interval is 3.000 sec
  Preemption enabled
  Priority is 90  (cfgd 110)
    Track object 1 state Down decrement 20
  Authentication text "CISCO"
  Master Router is 155.1.146.6, priority is 100
  Master Advertisement interval is 3.000 sec
  Master Down interval is 9.570 sec (expires in 8.714 sec)
```


13.16 IRDP

- R5 and SW2 should advertise themselves as default gateways for hosts on VLAN58 using ICMP messages.
- R5 should be the preferred gateway on the segment.

Configuration

```
R5:
interface FastEthernet 0/0
 ip irdp
 ip irdp address 155.1.58.5 1000
 ip irdp maxadvertinterval 20
 ip irdp minadvertinterval 10

SW2:
interface Vlan 58
 ip irdp
 ip irdp address 155.1.58.8 500
 ip irdp maxadvertinterval 20
 ip irdp minadvertinterval 10
```


Verification

Note

IRDP uses ICMP messages to advertise candidate default gateway on a segment. Multiple routers can advertise themselves at the same time, with different priorities.

To verify this configuration disable IP routing on SW1, and enable default gateway discovery via IRDP. Additonally create a VLAN 58 interface on SW1 and assign it an IP address of 155.X.58.7/24.

```
SW1:
no ip routing
ip gdp irdp
!
interface Vlan 58
 ip address 155.1.58.7 255.255.255.0

Rack1SW1#show ip route
Gateway         Using  Interval  Priority   Interface
155.1.58.5      IRDP       30      1000     Vlan58

Default gateway is not set

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty

Rack1R5#show ip irdp ethernet 0/0
Ethernet0/0 has router discovery enabled

Advertisements will occur between every 10 and 20 seconds.
Advertisements are sent with multicasts.
Advertisements are valid for 60 seconds.
Default preference will be 0.
Proxy for 155.1.58.5 with preference 1000.
```


13.17 Router ICMP Settings

- Configure R1's VLAN146 interface to stop sending ICMP messages about discarded packets, ICMP messages to select a better next-hop, and ICMP messages reporting subnets mask.
- Rate-limit ICMP unreachables to 2 per-second globally.

Configuration

```
R1:
interface FastEthernet 0/0
 no ip redirects
 no ip unreachables
 no ip mask-reply
!
ip icmp rate-limit unreachable 500
```

Verification

> **Note**
>
> You can verify the rate limit of unreachable using the **`traceroute`** command on R3. The destination router sends ICMP unreachable packets in response to UDP probes. Note that you can't do traceroute from R6 or R4 since R1 has ICMP unreachables disabled on VLAN146 interface.

```
Rack1R3#traceroute 155.1.13.1

Type escape sequence to abort.
Tracing the route to 155.1.13.1

  1 155.1.13.1 12 msec *  12 msec
```

Change the ICMP unreachables rate to 1 each every 10ms and repeat the traceroute command from R3.

```
Rack1R1(config)#ip icmp rate-limit unreachable 10

Rack1R3#traceroute 155.1.13.1

Type escape sequence to abort.
Tracing the route to 155.1.13.1

  1 155.1.13.1 16 msec 16 msec 16 msec
```


Verify the ICMP messages settings on R1 VLAN146 interface.

```
Rack1R1#show ip interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 155.1.146.1/24
  Broadcast address is 255.255.255.255
  Address determined by DHCP
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are never sent
  ICMP unreachables are never sent
  ICMP mask replies are never sent
 <snip>
```


13.18 Basic NAT

- Ensure that R4 and R6 are not advertising any internal networks (i.e. 155.X.0.0/16 or 150.X.0.0/16) to BB3.
- Configure R4 and R6 to translate IP source addresses for packets from the subnet 155.X.0.0/16 going to BB3 and BB1 respectively using a dynamic pool consisting of the subnets of their BB connections.
- Do not use any route-maps to accomplish this.

Configuration

```
R4:
ip nat pool VLAN43 204.12.1.1 204.12.1.253 prefix-length 24
!
ip access-list extended NAT_TRAFFIC
 permit ip 155.1.0.0 0.0.255.255 any
!
ip nat inside source list NAT_TRAFFIC pool VLAN43
!
interface FastEthernet 0/0
 ip nat outside
!
interface FastEthernet 0/1
 ip nat inside
!
interface Serial 0/0
 ip nat inside
!
interface Serial 0/1
 ip nat inside
!
router rip
 passive-interface FastEthernet 0/0

R6:
ip nat pool SERIAL 54.1.1.1 54.1.1.253 prefix-length 24
!
ip access-list extended NAT_TRAFFIC
 permit ip 155.1.0.0 0.0.255.255 any
!
ip nat inside source list NAT_TRAFFIC pool SERIAL
!
interface Serial 0/0
 ip nat outside
!
interface FastEthernet 0/0.146
 ip nat inside
!
interface FastEthernet 0/0.67
 ip nat inside
!
router rip
 passive-interface Serial 0/0
```


Verification

> **Note**
>
> Source ICMP packets off the internal interfaces of R4 and R6 to verify that NAT translations are created. Send additional packets from R1 to ensure that the internal routers may also reach the networks behind the backbone routers.

```
Rack1R6#ping 212.18.1.1 source fastEthernet 0/0.146

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 212.18.1.1, timeout is 2 seconds:
Packet sent with a source address of 155.1.146.6
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/36 ms

Rack1R6#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 54.1.1.1:2        155.1.146.6:2      212.18.1.1:2       212.18.1.1:2
--- 54.1.1.1           155.1.146.6        ---                ---

Rack1R4#ping 30.0.0.1 source FastEthernet 0/1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 155.1.146.4
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/29/96 ms

Rack1R4#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 204.12.1.1:0      155.1.146.4:0      30.0.0.1:0         30.0.0.1:0
tcp 204.12.1.1:179     155.1.146.4:179    204.12.1.254:13219 204.12.1.254:13219
--- 204.12.1.1         155.1.146.4        ---

Rack1R1#ping 30.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 4/5/8 ms

Rack1R1#ping 212.18.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 212.18.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/36 ms
```


Note that with access-list based NAT, the process creates one “parent” IP to IP address translation entry (e.g. 204.12.1.2 to 155.1.146.1 – inside global/inside local), which is used to multiplex all sessions for this particular inside host. Those IP to IP entries are called non-extendable, for they only have inside local and inside global IP address information. The one-to-one entries also permit outside hosts to connected to the inside using the temporary mapped IP address. Additionally, with non-extendable entries, an inside local address may have only one inside global translation. This effectively prevents multi-homed NAT when using access-lists only for NAT configuration.

```
Rack1R4#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 204.12.1.2:4546   155.1.146.1:4546   30.0.0.1:4546      30.0.0.1:4546
icmp 204.12.1.2:4547   155.1.146.1:4547   30.0.0.1:4547      30.0.0.1:4547
icmp 204.12.1.2:4548   155.1.146.1:4548   30.0.0.1:4548      30.0.0.1:4548
icmp 204.12.1.2:4549   155.1.146.1:4549   30.0.0.1:4549      30.0.0.1:4549
--- 204.12.1.2         155.1.146.1        ---                ---
tcp 204.12.1.1:179     155.1.146.4:179    204.12.1.254:13231 204.12.1.254:13231
--- 204.12.1.1         155.1.146.4        ---                ---

Rack1R6#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 54.1.1.2:692      155.1.146.1:692    212.18.1.1:692     212.18.1.1:692
icmp 54.1.1.2:693      155.1.146.1:693    212.18.1.1:693     212.18.1.1:693
icmp 54.1.1.2:694      155.1.146.1:694    212.18.1.1:694     212.18.1.1:694
icmp 54.1.1.2:695      155.1.146.1:695    212.18.1.1:695     212.18.1.1:695
icmp 54.1.1.2:696      155.1.146.1:696    212.18.1.1:696     212.18.1.1:696
--- 54.1.1.2           155.1.146.1        ---                ---
--- 54.1.1.1           155.1.146.6        ---                ---
```


13.19 NAT Overload

- Configure R4 and R6 to translate packets sourced from the Loopback0 subnets (150.X.0.0/16) using their backbone connection IP addresses.

Configuration

```
R4:
ip access-list extended LOOPBACKS
 permit ip 150.1.0.0 0.0.255.255 any
!
ip nat inside source list LOOPBACKS interface FastEthernet 0/0 overload

R6:
ip access-list extended LOOPBACKS
 permit ip 150.1.0.0 0.0.255.255 any
!
ip nat inside source list LOOPBACKS interface Serial 0/0 overload
```


Verification

Note

NAT Overloading is also known as PAT (Port Address Translations) since it uses the same global IP address for all local IP addresses, changing just the port numbers.

To verify the configuration, source traffic off R1's Loopback0 interface to the networks behind BB1 and BB3.

```
Rack1R1#ping 212.18.1.1 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 212.18.1.1, timeout is 2 seconds:
Packet sent with a source address of 150.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/36 ms

Rack1R1#ping 30.0.0.1 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 150.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
```

Note that "overload" NAT feature does not create any IP to IP (non-extendable) entries, but rather multiplexes all translations in one global IP address, using inside/outside IP addresses and port numbers (IDs) as keys.

```
Rack1R4#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 204.12.1.4:15     150.1.1.1:15       30.0.0.1:15        30.0.0.1:15

Rack1R6#show ip nat traslations
Pro Inside global      Inside local       Outside local      Outside global
icmp 54.1.1.6:14       150.1.1.1:14       212.18.1.1:14      212.18.1.1:14
```


13.20 NAT with Route Maps

- Configure R2 with route-map based NAT to support multiple outside interfaces.
- Traffic from R2's Loopback0 network going out the point-to-point link to R3 should be translated to 155.X.23.200.
- All other traffic going out the point-to-point link to R3 should be translated to R2's interface IP address.
- Traffic from R2's Loopback0 network going out the Frame Relay link to R5 should be translated to 150.X.2.200.
- All other traffic going out the Frame Relay link to R5 should be translated to R2's interface IP address.

Configuration

```
R2:
ip access-list standard FROM_LOOPBACK
 permit 150.1.2.0 0.0.0.255
!
route-map NAT_OUT_PPP_FROM_LOOPBACK permit 10
 match ip address FROM_LOOPBACK
 match interface Serial0/1
!
route-map NAT_OUT_PPP_NOT_FROM_LOOPBACK deny 10
 match ip address FROM_LOOPBACK
 match interface Serial0/1
!
route-map NAT_OUT_PPP_NOT_FROM_LOOPBACK permit 20
 match interface Serial0/1
!
route-map NAT_OUT_FRAME_RELAY_FROM_LOOPBACK permit 10
 match ip address FROM_LOOPBACK
 match interface Serial0/0
!
route-map NAT_OUT_FRAME_RELAY_NOT_FROM_LOOPBACK deny 10
 match ip address FROM_LOOPBACK
 match interface Serial0/0
!
route-map NAT_OUT_FRAME_RELAY_NOT_FROM_LOOPBACK permit 20
 match interface Serial0/0
!
ip nat pool PPP_LOOPBACK_POOL 155.1.23.200 155.1.23.200 prefix-length
24
!
ip nat inside source route-map NAT_OUT_PPP_FROM_LOOPBACK pool
PPP_LOOPBACK_POOL overload
!
ip nat inside source route-map NAT_OUT_PPP_NOT_FROM_LOOPBACK interface
Serial0/1 overload
!
ip nat pool FRAME_RELAY_LOOPBACK_POOL 150.1.2.200 150.1.2.200 prefix-
length 24
!
```



```
ip nat inside source route-map NAT_OUT_FRAME_RELAY_FROM_LOOPBACK pool
FRAME_RELAY_LOOPBACK_POOL overload
!
ip nat inside source route-map NAT_OUT_FRAME_RELAY_NOT_FROM_LOOPBACK
interface Serial0/0 overload
!
interface Serial0/0
 ip nat outside
!
interface Serial0/1
 ip nat outside
```

Verification

> **Note**
>
> To verify this configuration telnet from R2 to destinations out the point-to-point and Frame Relay links while changing the source addresses.

```
Rack1R2#telnet 155.1.23.3 /source-interface FastEthernet0/0
Trying 155.1.23.3 ... Open

User Access Verification

Password:
Rack1R3>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:00:17
* 66 vty 0                 idle                 00:00:00 155.1.23.2

  Interface    User               Mode         Idle     Peer Address

Rack1R2#telnet 155.1.23.3 /source-interface Loopback0
Trying 155.1.23.3 ... Open

User Access Verification

Password:
Rack1R3>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:00:27
* 66 vty 0                 idle                 00:00:00 155.1.23.200

  Interface    User               Mode         Idle     Peer Address
```



```
Rack1R2#telnet 155.1.108.10 /source-interface FastEthernet0/0
Trying 155.1.108.10 ... Open

User Access Verification

Password:
Rack1SW4>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                    3d21h
*  1 vty 0                 idle                 00:00:00 155.1.0.2

  Interface      User        Mode                      Idle     Peer
Address

Rack1R2#telnet 155.1.108.10 /source-interface Loopback0
Trying 155.1.108.10 ... Open


User Access Verification

Password:
Rack1SW4>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                    3d21h
*  1 vty 0                 idle                 00:00:00 150.1.2.200

  Interface      User        Mode                      Idle     Peer
Address

Queued Packets: 0
```

When using route-maps for NAT configuration (not just simple access-lists) the NAT process creates “extendable” NAT entries (fully-extended, with local/global IP addresses and port numbers). This allows for NAT multi-homing, since a particular inside local source IP address is not bound to just one inside global IP address. Also, reverse connections to inside hosts using temporary global IP address mappings are not possible with default route-map configuration.

```
Rack1R2#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 150.1.2.200:37611  150.1.2.2:37611    155.1.108.10:23    155.1.108.10:23
udp 155.1.0.2:520      155.1.0.2:520      224.0.0.9:520      224.0.0.9:520
udp 155.1.23.2:520     155.1.23.2:520     224.0.0.9:520      224.0.0.9:520
tcp 155.1.0.2:35504    192.10.1.2:35504   155.1.108.10:23    155.1.108.10:23
```


13.21 Static NAT

- Configure R5 so that R4 can reach SW2's VLAN 58 IP address using the address 155.X.45.8.
- Additionally SW2 should be able reach R4's point-to-point link using the VLAN 58 IP address 155.X.58.4.

Configuration

```
R5:
interface Serial0/1
 ip nat outside
!
interface FastEthernet0/0
 ip nat inside
!
ip nat inside source static 155.1.58.8 155.1.45.8
ip nat outside source static 155.1.45.4 155.1.58.4
!
ip route 155.1.58.4 255.255.255.255 155.1.45.4
```

Verification

> **Note**
>
> When a packet arrives at the NAT "inside" interface its destination IP address is first used for a routing lookup before it is translated via the NAT table; this is why the static route is needed. When SW2 sends a packet to the address 155.X.58.4, R5 first attempts to route it before consulting the NAT table. Without the static route R5 ends up routing the packet back out the interface connecting to VLAN 58, which was the interface the packet came in on.

```
Rack1SW2#telnet 155.1.58.4
Trying 155.1.58.4 ... Open

User Access Verification

Password:
Rack1R4>sh user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 01:08:09
* 66 vty 0                 idle                 00:00:00 155.1.45.8

  Interface    User               Mode         Idle     Peer Address
```

```
Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- ---                ---                155.1.58.4         155.1.45.4
tcp 155.1.45.8:47235   155.1.58.8:47235   155.1.58.4:23      155.1.45.4:23
--- 155.1.45.8         155.1.58.8         ---                ---
```

To verify translation, route-cache is disabled on R5 to allow for transit packets to be process switched.

```
R5:
access-list 100 permit icmp any any
!
interface FastEthernet 0/0
 no ip route-cache
!
interface Serial 0/1
 no ip route-cache

Rack1R5#debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100

Rack1R5#debug ip nat detail
IP NAT detailed debugging is on

Rack1SW2#ping 155.1.58.4 repeat 2

Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 155.1.58.4, timeout is 2 seconds:
!!
Success rate is 100 percent (2/2), round-trip min/avg/max = 42/42/42 ms
```

The NAT inside translation is denoted by the lowercase “i”. The packet from SW2 reaches R5 and gets translated due to the static route out to R4.

```
Rack1R5#
IP: tableid=0, s=155.1.58.8 (FastEthernet0/0), d=155.1.58.4
(Serial0/1), routed via RIB
NAT: i: icmp (155.1.58.8, 3) -> (155.1.58.4, 3) [12]
NAT: s=155.1.58.8->155.1.45.8, d=155.1.58.4 [12]
NAT: s=155.1.45.8, d=155.1.58.4->155.1.45.4 [12]
IP: s=155.1.45.8 (FastEthernet0/0), d=155.1.45.4 (Serial0/1),
g=155.1.45.4, len 100, forward
    ICMP type=8, code=0
```


The NAT outside translation is denoted by the lowercase “o”. For the “outside” interface the destination IP is first used for translation, and only then for the routing lookup. Based on this there is no need for a static “fixup” route on R5.

```
NAT*: o: icmp (155.1.45.4, 3) -> (155.1.45.8, 3) [12]
NAT*: s=155.1.45.4->155.1.58.4, d=155.1.45.8 [12]
NAT*: s=155.1.58.4, d=155.1.45.8->155.1.58.8 [12]
IP: tableid=0, s=155.1.58.4 (Serial0/1), d=155.1.58.8
(FastEthernet0/0), routed via FIB
IP: s=155.1.58.4 (Serial0/1), d=155.1.58.8 (FastEthernet0/0),
g=155.1.58.8, len 100, forward
    ICMP type=0, code=0

<snip>
```

Try removing the static route from R5 and see what happens.

```
Rack1R5(config)#no ip route 155.1.58.4 255.255.255.255 155.1.45.4

Rack1SW2#ping 155.1.58.4 repeat 2

Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 155.1.58.4, timeout is 2 seconds:
!!
Success rate is 100 percent (2/2), round-trip min/avg/max = 8/8/9 ms

Rack1SW2#telnet 155.1.58.4
Trying 155.1.58.4 ...
% Connection refused by remote host
```

NAT never triggers at R5, but R5 responds to the ICMP echo packets due to the local alias. Note that you can disable the local aliasing (proxy ARP entries creation) for NAT entries using the **no-alias** option for the ip nat command.

```
Rack1R5#show arp | include 155.1.58.(4|5)
Internet  155.1.58.4              -   000c.ce60.9fe0  ARPA   FastEthernet0/0
Internet  155.1.58.5              -   000c.ce60.9fe0  ARPA   FastEthernet0/0
```


13.22 Static PAT

- Remove the previous static NAT entries on R5.
- Configure R5 so that when R4 telnets to 155.X.45.44 at port 8023 they get connected to SW2's VLAN 8 interface at port 23.
- Configure R5 so that when R4 telnets to 155.X.45.44 at port 10023 they get connected to SW4's VLAN 10 interface at port 23.

Configuration

```
R5:
ip nat inside source static tcp 155.1.8.8 23 155.1.45.44 8023
ip nat inside source static tcp 155.1.10.10 23 155.1.45.44 10023
```

Verification

> **Note**
>
> Static PAT allows mapping ports on a global IP address to the ports on a local IP address. To verify this configuration, telnet from R4 to R5 at ports 8023 and 10023.

```
Rack1R4#telnet 155.1.45.44 8023
Trying 155.1.45.44, 8023 ... Open

User Access Verification

Password:
Rack1SW2>

Rack1R4#telnet 155.1.45.44 10023
Trying 155.1.45.44, 10023 ... Open

User Access Verification

Password:
Rack1SW4>
```


Note that the host responding to ICMP echo packets is R4, since it created the IP alias entry for the translated IP address.

```
Rack1R4#ping 155.1.45.44

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.45.44, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

Rack1R5#show ip alias
Address Type             IP Address      Port
Interface                150.1.5.5
Interface                155.1.5.5
Dynamic                  155.1.45.44
Interface                155.1.0.5
Interface                155.1.58.5
Interface                155.1.45.5
```


13.23 Static NAT and IP Aliasing

- Configure R5 so that it does not add a local alias for the static mapping of the IP address 155.X.45.44.

Configuration

```
R5:
ip nat inside source static tcp 155.1.8.8 23 155.1.45.44 8023
extendable no-alias
ip nat inside source static tcp 155.1.10.10 23 155.1.45.44 10023
extendable no-alias
```

Verification

> **Note**
>
> The no-alias feature will prevent a router running NAT from installing a local IP alias entry. However, the router will still respond to ARP requests for the global translated IP address, it just won't terminate any connection on itself. To verify this, telnet to 155.X.45.44 at port 8023 from R4, and ping this address.

```
Rack1R4#telnet 155.1.45.44 10023
Trying 155.1.45.44, 10023 ... Open

User Access Verification

Password:
Rack1SW4>
```

Now R5 does not respond to the ICMP echo packets since it has no local alias generated for the IP address 155.X.45.44.

```
Rack1R4#ping 155.1.45.44

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.45.44, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Rack1R5#show ip aliases
Address Type             IP Address      Port
Interface                150.1.5.5
Interface                155.1.5.5
Interface                155.1.0.5
Interface                155.1.58.5
Interface                155.1.45.5
```


13.24 Static Policy NAT

- Remove all previous NAT configurations from R2.
- Configure a static translation on R3 so that when traffic is received from SW1's VLAN 7 IP address and it is going out the point-to-point link to R1 it is translated to the address 155.X.13.7.
- Configure a static translation on R3 so that when traffic is received from SW1's VLAN 7 IP address and it is going out the point-to-point link to R2 it is translated to the address 155.X.23.7.

Configuration

```
R3:
route-map LINK_TO_R1
 match interface Serial1/2
!
route-map LINK_TO_R2
 match interface Serial1/3
!
ip nat inside source static 155.1.7.7 155.1.13.7 route-map LINK_TO_R1
ip nat inside source static 155.1.7.7 155.1.23.7 route-map LINK_TO_R2
!
interface FastEthernet0/0
 ip nat inside
!
interface Serial1/2
 ip nat outside
!
interface Serial1/3
 ip nat outside
```

Verification

Note

Static policy NAT (static NAT with route-maps) allows the use of mappings to different inside global IP addresses for the same inside local IP. The route-maps are used to classify traffic to be used with which respective static NAT translation. Commonly this type of mapping is used on multi-homed routers where the same inside local server address maps to different inside global IP addresses out to multiple ISPs.

To verify this configuration telnet from SW1 with a source address of VLAN 7 out to R1 and R2, and issue the **`show users`** command.


```
Rack1SW1#telnet 155.1.13.1 /source-interface Vlan7
Trying 155.1.13.1 ... Open

User Access Verification

Password:
Rack1R1>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:31:47
* 66 vty 0                 idle                 00:00:00 155.1.13.7

  Interface    User               Mode         Idle     Peer Address

Rack1R1>exit

[Connection to 155.1.13.1 closed by foreign host]
Rack1SW1#telnet 155.1.23.2 /source-interface Vlan7
Trying 155.1.23.2 ... Open


User Access Verification

Password:
Rack1R2>show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:00:12
* 66 vty 0                 idle                 00:00:00 155.1.23.7

  Interface    User               Mode         Idle     Peer Address
```

Next remove the static statements that call the route-maps and replace them with normal static extendable translations.

```
Rack1R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R3(config)#no ip nat inside source static 155.1.7.7 155.1.13.7
route-map LINK_TO_R1
Rack1R3(config)#no ip nat inside source static 155.1.7.7 155.1.23.7
route-map LINK_TO_R2
Rack1R3(config)#ip nat inside source static 155.1.7.7 155.1.13.7
extendable
Rack1R3(config)#ip nat inside source static 155.1.7.7 155.1.23.7
extendable
Rack1R3(config)#end
Rack1R3#
```

Reverify the operation of the translations by telneting from SW1 to R1 and R2 again, while sourcing the packets from VLAN 7.


```
Rack1SW1#telnet 155.1.13.1 /source-interface Vlan7
Trying 155.1.13.1 ... Open


User Access Verification

Password:
Rack1R1>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:40:52
* 66 vty 0                 idle                 00:00:00 155.1.13.7

  Interface    User               Mode         Idle     Peer Address

Rack1R1>exit

[Connection to 155.1.13.1 closed by foreign host]
Rack1SW1#telnet 155.1.23.2 /source-interface Vlan7
Trying 155.1.23.2 ... Open


User Access Verification

Password:
Rack1R2>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:09:19
* 66 vty 0                 idle                 00:00:00 155.1.13.7

  Interface    User               Mode         Idle     Peer Address
```

Note that in the above output both sessions are translated to the address 155.1.13.7, where in the previous output they were translated based on the exit interface when the route-maps were used for classification.

13.25 NAT with Overlapping Subnets

- Create a new Loopback1 interface on R1 and R2 using the IP addresses 10.0.0.1/24 and 10.0.0.2/24 respectively.
- Advertise the 10.0.0.0/24 network into RIP on R2.
- Configure NAT on R1 so that when R2 telnets to the address 11.0.0.Z from its Loopback1 address it is redirected to the address 10.0.0.Z attached to R1, where “Z” is any number.
- The host 10.0.0.Z on R1 should see the packets from R2 as if they were sourced from the network 22.0.0.0/24.

Configuration

```
R1:
interface Loopback 1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
!
ip nat pool R2_MASQUARADE 22.0.0.1 22.0.0.254 prefix-length 24
!
ip access-list extended R2_LOOPBACK1
 permit ip 10.0.0.0 0.0.0.255 any
!
ip nat outside source list R2_LOOPBACK1 pool R2_MASQUARADE
ip nat inside source static network 10.0.0.0 11.0.0.0 /24
!
ip route 11.0.0.0 255.255.255.0 Null 0
ip route 22.0.0.0 255.255.255.0 Serial 0/1
!
router rip
 redistribute static

R2:
interface Loopback 1
 ip address 10.0.0.2 255.255.255.0
!
router rip
 network 10.0.0.0
```


Verification

Note

This particular lab shows how to deal with overlapping subnets by configuring just one router. A better approach in a real-world design would be to create static network-to-network mappings on both R1 and R2 to mask their overlapping 10.0.0.0/24 segments, and never advertise the overlapping subnet in the first place.

In this case R2 still advertises its subnet 10.0.0.0/24 to the rest of the network, but R1 hides its Loopback1 subnet using a static NAT rule. Based on this R1 needs to be configured so that it never sees packets sourced from the 10.0.0.0/24 subnet on R2.

To verify the scenario enable NAT debugging on R1 and source ping packets off the Loopback1 interface of R2. Note that **ip nat inside** command on the Loopback1 interface of R1 is not necessary in this design, but it is needed in a real design where the are actual hosts behind the masqueraded segment.

```
Rack1R1#debug ip nat detailed
IP NAT detailed debugging is on
Rack1R1#

Rack1R2#ping 11.0.0.1 source loopback 1 repeat 2

Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 11.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.0.2
!!
Success rate is 100 percent (2/2), round-trip min/avg/max = 60/62/64 ms
```

Per the NAT outside rule, 10.0.0.2 is translated to 22.0.0.2 using the dynamic pool configured. Note that 11.0.0.1 translates to 10.0.0.1 thanks to the static subnet mapping. Eventually, R1 sees the original packet from 10.0.0.2 to 11.0.0.1 as a packet from 22.0.0.2 to 10.0.0.1.

```
Rack1R1#
NAT*: o: icmp (10.0.0.2, 20) -> (11.0.0.1, 20) [127]
NAT*: o: icmp (10.0.0.2, 20) -> (11.0.0.1, 20) [127]
NAT*: s=10.0.0.2->22.0.0.2, d=11.0.0.1 [127]
NAT*: s=22.0.0.2, d=11.0.0.1->10.0.0.1 [127]
```


The response packet from 10.0.0.1 to 22.0.0.2 is translated using two rules. First, 10.0.0.1 translates back to 11.0.0.1, and secondly, 22.0.0.2 translates to 10.0.0.2. Note that before the translation occurs R1 will try to look up a route for 22.0.0.2; this is why you need a static route on R1. The rest of the network does not need this static route as they know how to get to the 10.0.0.0/24 subnet on R2 because of the dynamic RIP route.

```
NAT: i: icmp (10.0.0.1, 20) -> (22.0.0.2, 20) [127]
NAT: s=10.0.0.1->11.0.0.1, d=22.0.0.2 [127]
NAT: s=11.0.0.1, d=22.0.0.2->10.0.0.2 [127]
```


13.26 TCP Load Distribution with NAT

- Remove the previous NAT configurations on R5.
- Configure rotary NAT R5 so that when SW2 telnets to the address 155.X.58.55 it is redirected to R1, R2, and R3 in an even distribution.

Configuration

```
R5:
interface FastEthernet0/0
 ip nat outside
!
interface Serial0/0
 ip nat inside
!
ip nat pool ROTARY prefix-length 24 type rotary
 address 155.1.0.1 155.1.0.1
 address 155.1.0.2 155.1.0.2
 address 155.1.0.3 155.1.0.3
!
ip access-list extended LOAD_BALANCE
 permit tcp any host 155.1.58.55 eq telnet
!
ip nat inside destination list LOAD_BALANCE pool ROTARY
!
ip alias 155.1.58.55 23
```

Verification

Note

NAT supports simple load-distribution for TCP based applications using the concept of rotary NAT pools. In this particular case the inside destination translation checks traffic as it comes in the outside interface. If the traffic is matched byt access-list LOAD_BALANCE, which equates to telnet traffic going to 155.1.58.55, the destination address is changed to 155.1.0.1, 155.1.0.2, and 155.1.0.3 in a round-robin (rotary) fashion per flow.

A more scalable approach to this same design is to use IOS Server Load Balancing (SLB) feature and NAT.

To verify the configuration telnet from SW2 to R5 three times in a row and see how the destination changes every time.


```
Rack1SW2#telnet 155.1.58.55
Trying 155.1.58.55 ... Open


User Access Verification

Password: cisco
Rack1R1>exit

[Connection to 155.1.58.55 closed by foreign host]

Rack1SW2#telnet 155.1.58.55
Trying 155.1.58.55 ... Open


User Access Verification

Password: cisco
Rack1R2>exit

[Connection to 155.1.58.55 closed by foreign host]

Rack1SW2#telnet 155.1.58.55
Trying 155.1.58.55 ... Open


User Access Verification

Password: cisco
Rack1R3>exit

[Connection to 155.1.58.55 closed by foreign host]
Rack1SW2#

Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.58.55:23     155.1.0.1:23       155.1.58.8:11004   155.1.58.8:11004
tcp 155.1.58.55:23     155.1.0.2:23       155.1.58.8:11005   155.1.58.8:11005
tcp 155.1.58.55:23     155.1.0.3:23       155.1.58.8:11006   155.1.58.8:11006
```

13.27 Stateful NAT with HSRP

- Remove all previous NAT configurations on R4 and R6.
- Configure HSRP on R4 and R6 with the group name “VLAN146” so that hosts on VLAN 146 can use the virtual IP address 155.X.146.254 as their default gateway.
- Confgure R6 to be the active physical gateway unless the line protocol of the link connecting to BB1 goes down.
- Set the hello interval to 1 second and dead interval to 3 seconds for faster failed gateway detection.
- Authenticate HSRP message exchanges using an MD5 hash of the key CISCO.
- Create a single NAT pool on R4 and R6 with the IP addresses in range 155.X.254.1-155.X.254.254.
- Configure BGP AS 100 on R4 and R6, and peer these routers with BB3 and BB1 respectively, who are in AS 54.
- Advertise the NAT pool via EBGP to BB1 and BB3.
- Redistribute all BGP routes into RIP on R4 and R6, and ensure that R6 is the preferred exit point.
- Configure dynamic NAT on R4 and R6 so that traffic from the 155.X.0.0/16 network is port translated to the previously defined NAT pool.
- Ensure R4 and R6 synchronize their NAT entries using HSRP so that TCP sessions from the inside network connected to devices in AS 54 are not dropped if R6 loses connectivity to BB1.

Configuration

```
R4:
interface FastEthernet 0/1
 standby 146 ip 155.1.146.254
 standby 146 timers 1 3
 standby 146 preempt
 standby 146 authentication md5 key-string CISCO
 standby 146 name VLAN146
 ip nat inside
!
interface FastEthernet0/0
 ip nat outside
!
ip nat stateful id 2
 redundancy VLAN146
  mapping-id 1
!
ip nat pool SHARED_POOL 155.1.254.1 155.1.254.254 prefix-length 24
!
ip route 155.1.254.0 255.255.255.0 Null 0
!
```



```
router bgp 100
 neighbor 204.12.1.254 remote-as 54
 network 155.1.254.0 mask 255.255.255.0
!
router rip
 redistribute bgp 100 metric 5
!
ip access-list standard NAT_LIST
 permit 155.1.0.0 0.0.255.255
!
ip nat inside source list NAT_LIST pool SHARED_POOL mapping 1

R6:
interface FastEthernet 0/0.146
 standby 146 ip 155.1.146.254
 standby 146 timers 1 3
 standby 146 preempt
 standby 146 authentication md5 key-string CISCO
 standby 146 name VLAN146
 standby 146 priority 110
 standby 146 track Serial 0/0
 ip nat inside
!
interface Serial0/0
 ip nat outside
!
ip nat stateful id 1
 redundancy VLAN146
  mapping-id 1
!
ip nat pool SHARED_POOL 155.1.254.1 155.1.254.254 prefix-length 24
!
ip route 155.1.254.0 255.255.255.0 Null 0
!
router bgp 100
 neighbor 54.1.1.254 remote-as 54
 network 155.1.254.0 mask 255.255.255.0
!
router rip
 redistribute bgp 100 metric 1
!
ip access-list standard NAT_LIST
 permit 155.1.0.0 0.0.255.255
!
ip nat inside source list NAT_LIST pool SHARED_POOL mapping 1
```


Verification

Note

The idea of stateful NAT is to allow two redundant exit points using the same NAT policy to exchange their NAT translation tables. This allows for stateful failover when the active forwarder fails.

To verify this check the Stateful NAT status first.

```
Rack1R6#sh ip snat distributed verbose

Stateful NAT Connected Peers

SNAT: Mode IP-REDUNDANCY :: ACTIVE
    : State READY
    : Local Address 155.1.146.6
    : Local NAT id 1
    : Peer Address 155.1.146.4
    : Peer NAT id 2
    : Mapping List 1
    : InMsgs 5, OutMsgs 0, tcb 0x84835C00, listener 0x0

User Access Verification

Rack1R4#sh ip snat distributed verbose

Stateful NAT Connected Peers

SNAT: Mode IP-REDUNDANCY :: STANDBY
    : State READY
    : Local Address 155.1.146.4
    : Local NAT id 2
    : Peer Address 155.1.146.6
    : Peer NAT id 1
    : Mapping List 1
    : InMsgs 39, OutMsgs 0, tcb 0x6408D040, listener 0x0
```

Now verify that SNAT peers change roles as R6 loses its active HSRP status:

```
Rack1R4#show standby
Ethernet0/1 - Group 146
  State is Active
    8 state changes, last state change 00:01:29
  Virtual IP address is 155.1.146.254
  Active virtual MAC address is 0000.0c07.ac92
    Local virtual MAC address is 0000.0c07.ac92 (v1 default)
  Hello time 1 sec, hold time 3 sec
    Next hello sent in 0.576 secs
  Authentication MD5, key-string "CISCO"
  Preemption enabled
```



```
  Active router is local
  Standby router is 155.1.146.6, priority 90 (expires in 2.716 sec)
  Priority 100 (default 100)
  IP redundancy name is "VLAN146" (cfgd)

Rack1R4#show ip snat distributed verbose

Stateful NAT Connected Peers

SNAT: Mode IP-REDUNDANCY :: ACTIVE
    : State READY
    : Local Address 155.1.146.4
    : Local NAT id 2
    : Peer Address 155.1.146.6
    : Peer NAT id 1
    : Mapping List 1
    : InMsgs 6, OutMsgs 0, tcb 0x6408D040, listener 0x0
```

Open a telnet session to a backbone network (BB1) and make sure the translation appears in the NAT table of R4.

```
Rack1R3#telnet 112.0.0.1
Trying 112.0.0.1 ... Open

Rack1R4#show ip nat translations | inc 155.1.0.3
tcp 155.1.254.4:65148  155.1.0.3:65148    112.0.0.1:23       112.0.0.1:23
```

Ensure R4 installed this NAT entry in R6's translation table using SNAT

```
Rack1R6#show ip snat peer 155.1.146.4

Show NAT Entries created by peer: 155.1.146.4

Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.254.4:65148  155.1.0.3:65148    112.0.0.1:23       112.0.0.1:23
```

Make R6 the active router now by enabling its backbone connection and shutting down the backbone connection of R4. Ensure that R3 never loses its telnet connection, but just experiences a temporary delay.

```
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface serial 0/0
Rack1R6(config-if)#no shutdown

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface Fastethernet 0/0
Rack1R4(config-if)#shutdown
```



```
Rack1AS>3
[Resuming connection 3 to R3 ... ]

RS.1.1.BB1>
RS.1.1.BB1>
```

Verify the NAT translation table of R6 and ensure it's now identical to the table used by R4.

```
Rack1R6#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.254.4:65148  155.1.0.3:65148    112.0.0.1:23       112.0.0.1:23
```


13.28 Stateful NAT with Primary/Backup

- Remove the HSRP configuration on R4 and R6.
- Modify the SNAT configuration on these devices so it does not rely on HSRP status for tracking of the active SNAT router.
- R6 should be the primary and R4 should be the backup router.

Configuration

```
R6:
interface FastEthernet0/0.146
 no standby 146
!
ip nat Stateful id 1
 no redundancy VLAN146
     primary 155.1.146.6
         peer 155.1.146.4
         mapping-id 1

R4:
interface FastEthernet0/1
 no standby 146
!
ip nat Stateful id 2
 no redundancy VLAN146
     backup 155.1.146.4
         peer 155.1.146.6
         mapping-id 1
```

Verification

> **Note**
>
> Stateful NAT with Primary/Backup replaces the need for HSRP tracking by using a built in SNAT keepalive. To verify the operation of this without HSRP source a telnet connection from R3 to BB1, and ensure that R6 installs the translation entry in the NAT table of R4.

```
Rack1R6#show ip snat distributed verbose

Stateful NAT Connected Peers

SNAT: Mode PRIMARY
    : State READY
    : Local Address 155.1.146.6
    : Local NAT id 1
    : Peer Address 155.1.146.4
    : Peer NAT id 2
    : Mapping List 1
    : InMsgs 5, OutMsgs 0, tcb 0x857FE654, listener 0x0
```

```
Rack1R4#show ip snat distributed verbose

Stateful NAT Connected Peers

SNAT: Mode BACKUP
    : State READY
    : Local Address 155.1.146.4
    : Local NAT id 2
    : Peer Address 155.1.146.6
    : Peer NAT id 1
    : Mapping List 1
    : InMsgs 15, OutMsgs 0, tcb 0x650FE6E0, listener 0x650FE22C

Rack1R6#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.254.1:26887  155.1.146.1:26887  112.0.0.1:23       112.0.0.1:23

Rack1R4#show ip snat peer 155.1.146.6

Show NAT Entries created by peer: 155.1.146.4

Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.254.1:26887  155.1.146.1:26887  112.0.0.1:23       112.0.0.1:23
```


13.29 NAT Virtual Interface

- Remove all previous NAT configurations on R5.
- Configure R5 so that traffic sourced from SW2's VLAN 8 network is translated to the source address range 155.X.188.0/24.
- Do not use the **ip nat inside** or **ip nat outside** commands or static routing to accomplish this.

Configuration

```
R5:
interface Serial 0/0
 ip nat enable
!
interface Serial 0/1
 ip nat enable
!
interface FastEthernet 0/0
 ip nat enable
!
ip access-list standard VLAN8
 permit 155.1.8.0 0.0.0.255
!
ip nat pool NVI_POOL 155.1.188.1 155.1.188.254 prefix 24 add-route
!
ip nat source list VLAN8 pool NVI_POOL
!
router rip
 redistribute static metric 1
```

Verification

> **Note**
>
> To verify translation using the NVI disable CEF on R5's interfaces (no ip route-cache), generate traffic from SW2's VLAN 8 interface, and issue the following debugs.

```
Rack1R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R5(config)#access-list 100 permit icmp any any
Rack1R5(config)#end
Rack1R5#
Rack1R5#debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100
Rack1R5#debug ip nat detail
IP NAT detailed debugging is on
```



```
Rack1SW2#ping 155.1.45.4 source vlan8 repeat 1

Rack1R5#
IP: tableid=0, s=155.1.8.8 (FastEthernet0/0), d=155.1.45.4 (Serial0/1),
routed via FIB
NAT: i: icmp (155.1.8.8, 10) -> (155.1.45.4, 10) [41]
NAT: s=155.1.8.8->155.1.188.2, d=155.1.45.4 [41]
IP: tableid=0, s=155.1.188.2 (FastEthernet0/0), d=155.1.45.4
(Serial0/1), routed via RIB
IP: s=155.1.188.2 (FastEthernet0/0), d=155.1.45.4 (Serial0/1),
g=155.1.45.4, len 100, forward
    ICMP type=8, code=0
```

Note the NAT direction is always “inside” for NVI based NAT. This means a routing lookup is always performed before the translation. After the routing decision is made the packet source is translated, then the packet is forwarded down the Serial interface.

```
IP: tableid=0, s=155.1.45.4 (Serial0/1), d=155.1.188.2 (NVI0), routed
via RIB
NAT: i: icmp (155.1.45.4, 10) -> (155.1.188.2, 10) [41]
NAT: s=155.1.45.4, d=155.1.188.2->155.1.8.8 [41]
IP: tableid=0, s=155.1.45.4 (Serial0/1), d=155.1.8.8 (FastEthernet0/0),
routed via RIB
```

When the return packet comes back it is routed using the NVI-based static route that is automatically. Note that NAT direction is “inside” again - this is why the NVI route is needed in this situation. With normal NAT outside to inside translation (the return traffic flow in this case) the packet destination would first get untranslated, and then routed using the true destination.

```
Rack1R5#show ip route static
     155.1.0.0/24 is subnetted, 17 subnets
S       155.1.188.0 [0/0] via 0.0.0.0, NVI0
```

Note that the NVI based static route eliminates the need for a statically configured route to Null0, but it does not remove the need of advertising the static route into the dynamic routing domain with the redistribute static statement.

To verify translation issue the show ip nat nvi translations command, not the normal show ip nat translations command.

```
Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global

Rack1R5#show ip nat nvi translations
Pro Source global      Source local       Destin  local      Destin  global
--- 155.1.188.2        155.1.8.8          ---                ---
```


13.30 NAT Default Interface

- Remove all previous NAT configurations on R5.
- Disable R5's connection to the Frame Relay network.
- Configure R5's point-to-point link to R4 as a passive-interface under RIP.
- Configure R5 so that all connections destined to its point-to-point Serial link are redirected to SW2's Loopback0 interface.
- All other inside to outside traffic on R5 should be dynamically overloaded to the point-to-point Serial link's IP address.

Configuration

```
R5:
interface FastEthernet 0/0
 ip nat inside
!
interface Serial 0/1
 ip nat outside
!
interface Serial 0/0
 shutdown
!
router rip
 passive-interface Serial0/1
!
ip access-list standard ALL
 permit any
!
ip nat inside source list ALL interface Serial 0/1 overload
ip nat inside source static 150.1.8.8 interface Serial 0/1
```


Verification

Note

The NAT Default Interface feature allows all traffic received on the outside interface that does not already match an existing dynamic translation to be statically forwarded to an inside host. This feature is useful for applications that use different outbound versus inbound traffic flows.

To verify this configuration telnet to any outside destination from the inside devices, and view the translation table on R5.

```
Rack1SW2#telnet 150.1.1.1
Trying 150.1.1.1 ... Open

User Access Verification

Password:
Rack1R1>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 01:51:29
* 66 vty 0                 idle                 00:00:00 155.1.45.5

  Interface    User               Mode         Idle     Peer Address

Rack1SW4#telnet 155.1.146.6
Trying 155.1.146.6 ... Open

User Access Verification

Password:
Rack1R6>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 01:51:25
* 66 vty 0                 idle                 00:00:00 155.1.45.5

  Interface    User               Mode         Idle     Peer Address
```


All inside traffic sent outbound is translated to R5's interface address 155.1.45.5.

```
Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 155.1.45.5         150.1.8.8          ---                ---
tcp 155.1.45.5:15418   155.1.58.8:15418   150.1.1.1:23       150.1.1.1:23
tcp 155.1.45.5:19938   155.1.108.10:19938 155.1.146.6:23     155.1.146.6:23
```

All outside traffic send inbound is redirected to the default host of SW2.

```
Rack1R4#telnet 155.1.45.5
Trying 155.1.45.5 ... Open

User Access Verification

Password:
Rack1SW2>

Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.45.5:23      150.1.8.8:23       155.1.45.4:32493   155.1.45.4:32493
--- 155.1.45.5         150.1.8.8          ---                ---
```


13.31 Reversible NAT

- Remove all previous NAT configurations on R5.
- Configure R5 so that all traffic received on the inside interface connecting to VLAN 58 is translated to the pool 155.X.45.100 – 155.X.45.200 as it exits out the point-to-point link to R4.
- Use route-map based NAT for this configuration to allow outside hosts to initiate connections back to hosts on the inside network once an inside to outside dynamic entry is created in the NAT table.

Configuration

```
R5:
interface FastEthernet 0/0
 ip nat inside
!
interface Serial 0/1
 ip nat outside
!
ip nat pool POOL 155.1.45.100 155.1.45.200 prefix-length 24
!
ip access-list standard INSIDE_HOSTS
 permit 155.1.0.0 0.0.255.255
 permit 150.1.0.0 0.0.255.255
!
route-map CREATE_EXTENDABLE_ENTRIES
 match ip address INSIDE_HOSTS
 match interface Serial0/1
!
ip nat inside source route-map CREATE_EXTENDABLE_ENTRIES pool POOL
reversible
```


Verification

Note

By default when you use route-maps with NAT rules, extendable entries are created. This disallows an external user to open a reverse connection back to an inside host since no one-to-one mapping exists in translation table. Reversible NAT allows creating of extendable entries along with reversible one-to-one mappings.

To verify this configuration, attempt to open a connection from the outside network to hosts on the inside network using the inside global NAT pool.

```
Rack1R6#telnet 155.1.45.101
Trying 155.1.45.101 ...
% Connection timed out; remote host not responding

Rack1R6#telnet 155.1.45.102
Trying 155.1.45.102 ...
% Connection timed out; remote host not responding

Rack1R6#telnet 155.1.45.103
Trying 155.1.45.103 ...
% Connection timed out; remote host not responding

Rack1R6#telnet 155.1.45.104
Trying 155.1.45.104 ...
% Connection timed out; remote host not responding
```

These connections are denied because R5 has no entries installed in the NAT table.

```
Rack1R5#show ip nat translations
```


Next, initiate a connection from the inside network to the outside network and verify R5's translation table.

```
Rack1SW4#telnet 155.1.146.6
Trying 155.1.146.6 ... Open

User Access Verification

Password:
Rack1R6>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:05:26
* 66 vty 0                 idle                 00:00:00 155.1.45.102

  Interface    User               Mode         Idle     Peer Address

Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.45.103:33927 155.1.108.10:33927 155.1.146.6:23     155.1.146.6:23
--- 155.1.45.103       155.1.108.10       ---                ---
```

Note that R5 creates both an extendable entry and a one-to-one entry. Once this one-to-one entry is created, the connection is reversible, and hosts on the outside network can initiate connections to hosts on the inside.

```
Rack1R1#telnet 155.1.45.103
Trying 155.1.45.103 ... Open

User Access Verification

Password:
Rack1SW4>
```


13.32 Static Extendable NAT

- Configure R5 so that SW4's Loopback0 address is reachable from the outside of the network using the IP addresses 155.1.45.201 and 155.1.45.202 at the same time.

Configuration

```
R5:
ip nat inside source static 150.1.10.10 155.1.45.201 extendable
ip nat inside source static 150.1.10.10 155.1.45.202 extendable
```

Verification

> **Note**
>
> Extendable static NAT allows you to configure multiple static mappings for the same local or global IP address. In this example the same local IP address is mapped to multiple global IP addresses. The extendable keyword allows every new translation to be fully extended, without binding a local IP address to a fixed global IP.
>
> To verify this telnet to both of the global IP addresses mapped to the local IP addresses of SW4.

```
Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 155.1.45.201       150.1.10.10        ---                ---
--- 155.1.45.202       150.1.10.10        ---                ---
```

> Before traffic is initiated R5 does not install extendable entries for the static mappings.

```
Rack1R6#telnet 155.1.45.201
Trying 155.1.45.201 ... Open

User Access Verification

Password:
Rack1SW4>exit

[Connection to 155.1.45.201 closed by foreign host]

Rack1R6#telnet 155.1.45.202
Trying 155.1.45.202 ... Open

User Access Verification

Password:
Rack1SW4>exit
```


Once traffic is sent the translations are fully extended.

```
Rack1R5#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 155.1.45.201:23    150.1.10.10:23     155.1.146.6:57575  155.1.146.6:57575
tcp 155.1.45.202:23    150.1.10.10:23     155.1.146.6:58011  155.1.146.6:58011
--- 155.1.45.201       150.1.10.10        ---                ---
--- 155.1.45.202       150.1.10.10        ---                ---
```

The rule that appears first in the configuration is used to resolve the ambiguity when translating from one to many addresses. This can be verified from initiating connections from the inside out.

```
Rack1SW4#telnet 150.1.1.1 /source-interface Loopback0
Trying 150.1.1.1 ... Open

User Access Verification

Password:
Rack1R1>show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:04:39
* 66 vty 0                 idle                 00:00:00 155.1.45.201

  Interface    User               Mode         Idle     Peer Address

Rack1SW4#telnet 150.1.6.6 /source-interface Loopback0
Trying 150.1.6.6 ... Open

User Access Verification

Password:
Rack1R6>show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:04:24
* 66 vty 0                 idle                 00:00:00 155.1.45.201

  Interface    User               Mode         Idle     Peer Address
```

Static policy NAT with route-maps can be used to explicitly specify which NAT rules to use and when they are used (i.e. per interface).

13.33 IP Precedence Accounting

- Configure R6 to account for the number of input and output packets based on their IP Precedence values on it's connection to BB1.

Configuration

```
R6:
interface Serial 0/0
 ip accounting precedence input
 ip accounting precedence output
```

Verification

> **Note**
>
> Routing protocols by default use an IP precedence value of 6. ICMP ping uses a default IP precedence of 0. These packets can be used to test the precedence accounting.

```
Rack1R6#ping 112.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 112.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/33 ms
Rack1R6#
```


Extended ping can be used to change the IP precedence of the packets generated.

```
Rack1R6#ping
Protocol [ip]:
Target IP address: 112.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
```

The first three bits of of the Type of Service (TOS) field are IP precedence. Therefore a precedence of 3 in binary would be 01100000, or 96 in decimal. A shortcut for this is to take the precedence value and multiple it by 32 to get its decimal representation (i.e. 3 X 32 = 96).

```
Type of service [0]: 96
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 112.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

To verify these statistics accumulated by the Serial interface issue the **show interface [*interface*] precedence** command. Use the **clear counters** command to clear these statistics.

```
Rack1R6#show interfaces serial 0/0 precedence
Serial0/0
  Input
    Precedence 0:  5 packets, 520 bytes
    Precedence 3:  5 packets, 520 bytes
    Precedence 6:  712 packets, 58455 bytes
  Output
    Precedence 0:  5 packets, 520 bytes
    Precedence 3:  5 packets, 520 bytes
    Precedence 6:  62 packets, 3634 bytes
```


13.34 IP Output Packet Accounting

- Configure R1 to account for the number of output packets on all physical interfaces.
- Only account for packets going to subnets of 155.X.0.0/16.
- Limit the accounting database size to 4096 entries.
- The number of records for packets not matching the account list should be set to 1.

Configuration

```
R1:
ip accounting-threshold 4096
ip accounting-transits 1
ip accounting-list 155.1.0.0 0.0.255.255
!
interface FastEthernet 0/0
 ip accounting output-packets
!
interface Serial 0/0
 ip accounting output-packets
!
interface Serial 0/1
 ip accounting output-packets
```

Verification

Note

IP accounting for output packets keeps local statistics of source & destination pairs, packet counts, and byte counts for packets leaving a particular link. The accounting-list matches either sources or destinations of outgoing packets that should be accounted for. Transit-records are accounting records created for packets not matching any of the accounting lists.

Those records do not contain detailed information, but instead just a notification of the first packet in the flow. By default if there is no accounting-list configured the value of the transit records is ignored. The default number of transit records is 0, so when you enable an accounting-list all non-matching packets are not accounted for.

To test this configuration, generate some traffic to populate the accounting database.

```
Rack1R3#ping 155.1.146.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms

Rack1R3#ping 150.1.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms

Rack1R3#ping 150.1.6.6 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.6.6, timeout is 2 seconds:
Packet sent with a source address of 150.1.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/45/108 ms

Rack1R3#ping 150.1.4.4 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
Packet sent with a source address of 150.1.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
```


Issue the **show ip accounting** command to view the accounting database. In this case we should see entries for flows that match the accounting-list (from or to the 155.X.0.0/16 subnet). Addtionally there should be just one entry for the packet exchange between R3's Loopback0 and R6's Loopback0. This is because the transit-record limit is set to 1. The second traffic flow from R3's Loopback0 to R4's Loopback0 will exceed the transit-record limit and therefore won't be recorded.

```
Rack1R1#show ip accounting
   Source           Destination              Packets              Bytes
 150.1.3.3        150.1.6.6                      1                100
 150.1.6.6        155.1.13.3                     5                500
 155.1.13.3       150.1.6.6                      5                500
 155.1.146.4      155.1.13.3                     5                500
 155.1.13.3       155.1.146.4                    5                500
```

When you clear the active accounting database it is copied to the "checkpoint" database, and accounting starts from scratch. You can retrieve the checkpoint database separately, as seen below. Every time you clear the active database it is copied to the checkpoint.

```
Rack1R1#clear ip accounting
Rack1R1#show ip accounting
   Source           Destination              Packets              Bytes

Accounting data age is 0

Rack1R1#show ip accounting checkpoint
   Source           Destination              Packets              Bytes
 150.1.3.3        150.1.6.6                      1                100
 150.1.6.6        155.1.13.3                     5                500
 155.1.13.3       150.1.6.6                      5                500
 155.1.146.4      155.1.13.3                     5                500
 155.1.13.3       155.1.146.4                    5                500

Accounting data age is 8
```


13.35 IP Access Violation Accounting

- Configure R6 to deny packets with an IP precedence of 4 coming from BB1.
- Account for the packets denied by this policy.

Configuration

```
R6:
access-list 101 deny ip any any precedence 4
access-list 101 permit ip any any
!
interface Serial 0/0
 ip access-group 101 in
 ip accounting access-violations
```

Verification

> **Note**
>
> To test this configuration generate packets from R6 to BB1 using a TOS of 128 (IP Precedence value of 4).

```
Rack1R6#sh ip accounting access-violations
   Source          Destination             Packets               Bytes   ACL

Accounting data age is 01:34

Rack1R6#ping
Protocol [ip]:
Target IP address: 112.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]: 128
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 112.0.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```



```
Rack1R6#show ip accounting access-violations
   Source           Destination              Packets                Bytes   ACL
 112.0.0.1        54.1.1.6                       5                  500   101

Accounting data age is 01:35
```

Pitfall

Access-violation accounting does not work with named access-lists.

13.36 MAC Address Accounting

- Configure R1 to account for the MAC addresses in IP packets entering or leaving its connection to VLAN 146.

Configuration

```
R1:
interface FastEthernet0/0
 ip accounting mac-address input
 ip accounting mac-address output
```

Verification

> **Note**
>
> MAC address accounting has fixed size of 512 entries for input and output frames. The below output shows source MAC addresses for input packets, and destination MAC addresses for output packets, as well as packet counts and their cumulative size.

```
Rack1R1#show interfaces fastEthernet 0/0 mac-accounting
FastEthernet0/0
  Input  (510 free)
    0006.d7bc.ed82(2  ):  21 packets, 8526 bytes, last: 6551ms ago
    0011.9221.da80(248):  21 packets, 9786 bytes, last: 4275ms ago
                  Total:  42 packets, 18312 bytes
  Output  (511 free)
    0100.5e00.0009(86 ):  21 packets, 6006 bytes, last: 1831ms ago
                  Total:  21 packets, 6006 bytes
```


13.37 TCP Optimization

- Modify R1's TCP configuration to meet the following requirements:
  - Avoid the TCP "silly window syndrome".
  - Enable high-performance TCP options.
  - Set the TCP window size to twice the standard 16 bit maximum value.
  - Limit the wait time for a TCP SYN response to the minimum.
  - Enable the feature to avoiding fragmentation with TCP sessions.
  - Hold no more that 16 packets in outgoing TCP queue.

Configuration

```
R1:
service nagle
ip tcp ecn
ip tcp selective-ack
ip tcp timestamp
ip tcp window-size 131070
ip tcp queuemax 16
ip tcp synwait-time 5
ip tcp path-mtu-discovery
```

Verification

> **Note**
>
> For more information on **`service nagle`** refer to RFC 896, *Congestion control in IP/TCP internetworks*. Refer to RFC 1323 for *TCP Extensions for High Performance*.
>
> To verify this configuration enable the same set of options in R6 and telnet from R6 to R1. Once the session is open examine the TCP connection status on R6 and R1.

```
Rack1R6#telnet 150.1.1.1
Trying 150.1.1.1 ... Open


User Access Verification

Password:
Rack1R1>
```



```
Rack1R6#show tcp brief all
TCB       Local Address           Foreign Address         (state)
856D54A0  54.1.1.6.60615          54.1.1.254.179          ESTAB
856BE248  155.1.146.6.33209       150.1.1.1.23            ESTAB
849057EC  150.1.6.6.23            155.1.146.4.60166       ESTAB
856D4FEC  *.179                   54.1.1.254.*            LISTEN
854E4AFC  *.80                    *.*                     LISTEN

Rack1R6#show tcp tcb 856BE248
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Enabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 155.1.146.6, Local port: 33209
Foreign host: 150.1.1.1, Foreign port: 23

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x1080595):
Timer          Starts    Wakeups            Next
Retrans             6          0             0x0
TimeWait            0          0             0x0
AckHold             3          1             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 1568200946  snduna: 1568200984  sndnxt: 1568200984     sndwnd:
131032
irs: 1153885002  rcvnxt: 1153885082  rcvwnd:     130991  delrcvwnd:
79

SRTT: 165 ms, RTTO: 1172 ms, RTV: 1007 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: active open, retransmission timeout, retransmitting, nagle
  path mtu capable, SACK option permitted, Timestamp option used
  win-scale
IP Precedence value : 6

Datagrams (max data segment is 1448 bytes):
Rcvd: 10 (out of order: 0), with data: 7, total data bytes: 79
Sent: 11 (retransmit: 0, fastretransmit: 0, partialack: 0, Second
Congestion: 0), with data: 7, total data bytes: 37
```



```
Rack1R1#show tcp brief all
TCB       Local Address           Foreign Address        (state)
848F416C  150.1.1.1.23            155.1.146.6.33209      ESTAB
854BE9AC  *.80                    *.*                    LISTEN

Rack1R1#show tcp tcb 848F416C
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Enabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 150.1.1.1, Local port: 23
Foreign host: 155.1.146.6, Foreign port: 33209

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x10CB15B):
Timer          Starts    Wakeups            Next
Retrans             4          0             0x0
TimeWait            0          0             0x0
AckHold             5          2             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 1153885002  snduna: 1153885082  sndnxt: 1153885082     sndwnd:
130990
irs: 1568200946  rcvnxt: 1568200984  rcvwnd:     131033  delrcvwnd:
37

SRTT: 124 ms, RTTO: 1405 ms, RTV: 1281 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, active open, retransmission timeout,
retransmitting
  nagle, SACK option permitted, Timestamp option used, win-scale
IP Precedence value : 0

Datagrams (max data segment is 1448 bytes):
Rcvd: 11 (out of order: 0), with data: 7, total data bytes: 37
Sent: 10 (retransmit: 0, fastretransmit: 0, partialack: 0, Second
Congestion: 0), with data: 7, total data bytes: 79
```


13.38 IOS Small Services & Finger

- Configure R1 to meet the following requirements:
  - Users who telnet or send UDP packets to R1 at port 7 should have any typed characters echoed back to them.
  - Users who telnet or send UDP packets to R1 at port 9 should have any typed characters thrown away.
  - Users who telnet to R1 at port 13 should be returned the date and time.
  - Users who telnet or send UDP packets to R1 at port 19 should be sent a stream of ASCII characters.
  - Users who telnet to R1 at port 79 should be sent a list of currently logged in users.

Configuration

```
R1:
service udp-small-servers
service tcp-small-servers
ip finger
```

Verification

> **Note**
>
> Connect to the well-know TCP ports on R1 to verify the small services. Use the escape sequence “Ctrl-Shift-6-6 X” to escape the session and disconnect it.

```
Rack1R1#telnet 150.1.1.1 echo
Trying 150.1.1.1, 7 ... Open
tteesstt

Rack1R1#telnet 150.1.1.1 discard
Trying 150.1.1.1, 9 ... Open
test

Rack1R1#telnet 150.1.1.1 daytime
Trying 150.1.1.1, 13 ... Open
Thursday, October 9, 2008 11:03:56-UTC
```



```
Rack1R1#telnet 150.1.1.1 chargen
Trying 150.1.1.1, 19 ... Open
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh
"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi
#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghij
$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijk
%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm
<snip>

Rack1R1#telnet 150.1.1.1 finger
Trying 150.1.1.1, 79 ... Open

    Line       User       Host(s)              Idle       Location
   0 con 0                 150.1.1.1            00:00:00
* 66 vty 0                idle                 00:00:00 150.1.1.1

  Interface    User               Mode         Idle     Peer Address
```

13.39 Directed Broadcasts & UDP Forwarding

- Configure R5 so that if it receives packets for the destination address 155.X.58.255 they are forwarded out the link to SW2, but the destination should not be changed to 255.255.255.255.
- Configure R5 to listen to UDP broadcasts on the point-to-point link to R4, and forward them to the address 155.X.58.255.
- This forwarding should only occur for DNS packets.
- Verify this configuration by sending ICMP pings to the address 155.X.58.255 and DNS broadcast requests from R4.

Configuration

```
R5:
no ip forward-protocol udp bootps
no ip forward-protocol udp tftp
no ip forward-protocol udp time
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs
!
interface Serial 0/1
 ip helper-address 155.1.58.255
!
interface FastEthernet 0/0
 ip broadcast-address 155.1.58.255
 ip directed-broadcast
```

Verification

Note

A directed broadcast packet is a packet whose destination address matches the last IP address in a subnet assigned to the local device. For example the network 10.0.0.0/8 has a directed broadcast address, or subnet broadcast address, of 10.255.255.255. The directed broadcast feature, which is disabled by default, allows remote devices to send a broadcast packet to a particular link by sending the packet to the directed broadcast address. Normally this feature is not allowed due to security risks inherent to broadcasts, such as smurf and fraggle DoS attacks.

When the router receives a directed broadcast, and the **`ip directed-broadcast`** command is enabled on the outgoing interface where the address match occurs, the router changes the destination address in the packet from the directed broadcast address to the all subnet broadcast address of 255.255.255.255. This behavior can be overridden with the **`ip broadcast-address`** command, which it is in the above case.

The UDP forwarding feature, controlled by the **`ip helper-address`**, is used to take broadast UDP packets and change the destination address to a unicast or directed broadcast address. By default the router will forward UDP packets only for the following protocols: TACACS (not TACACS+), TFTP, BOOTP, Time, NetBIOS Name Server and NetBIOS Datagram Services and DNS. In the above case all of these protocols are disabled with the exception of DNS.

To verify this configuration send packets from R4 to broadcast IP address of VLAN 58. Enable IP packet debugging for ICMP packets on R5 and SW2 to view the results.

```
R5, SW2:
access-list 100 permit icmp any any

Rack1R5#debug ip packet detail 100
Rack1SW2#debug ip packet detail 100

Rack1R4#ping 155.1.58.255

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.58.255, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms

Rack1R5#
IP: =155.1.45.4 (Serial0/1), d=155.1.58.255 (Ethernet0/0), routed via
RIB
IP: s=155.1.45.4 (Serial0/1), d=155.1.58.255 (Ethernet0/0),
g=255.255.255.255, len 100, forward directed broadcast
    ICMP type=8, code=0
IP: s=155.1.45.4 (Serial0/1), d=155.1.58.255 (Ethernet0/0), len 100,
rcvd 5
    ICMP type=8, code=0
IP: tableid=0, s=155.1.45.5 (local), d=155.1.45.4 (Serial0/1), routed
via FIB
IP: s=155.1.45.5 (local), d=155.1.45.4 (Serial0/1), len 100, sending
    ICMP type=0, code=0
```



```
Rack1SW2#
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255 (Vlan58), len 100, rcvd 3
    ICMP type=8, code=0
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255, len 100, stop process pak
for forus packet
    ICMP type=8, code=0
IP: s=155.1.58.8 (local), d=155.1.45.4 (Vlan58), len 100, sending
    ICMP type=0, code=0
```

R5 receives the directed broadcast and forwards it onto VLAN 58. SW2 receives the packet, processes the packet locally, and sends an ICMP echo reply back to R4. This behavior is considered a security risk and is exploited in the smurf and fraggle attacks. These attacks send ICMP or UDP echo packets to a directed broadcast address with a spoofed source. When the destination segment receives the broadcast, all hosts on the segment reply back to the spoofed source (the victim). If the link is a /24 subnet that is fully populated with 253 hosts, it means that for every one request sent from the attacker 253 replies are sent back to the victim. To prevent this the **`ip directed-broadcast`** command is simply disabled at the interface level.

To test the UDP forwarding enable DNS name lookup on R4 without configuring a DNS server. This causes the router to send DNS requests to the all subnet broadcast address of 255.255.255.255 out all links. When R5 hears this packet it changes the destination address to the helper, 155.1.58.255, and forwards it as a directed broadcast to VLAN 58.

In a real network design this behavior can be useful for DNS and DHCP forwarding. If there are multiple DNS or DHCP servers on a segment, traffic can be load shared to the fastest server by sending the name or address request to both at the same time. Whichever server replies first will be the one that is used for that particular application.

```
Rack1R4#conf t
Rack1R4#ip domain-lookup
Rack1R4(config)#access-list 100 permit udp any any eq 53
Rack1R4#debug ip packet detail 100

Rack1R4#testname
Translating "testname"...domain server (255.255.255.255)
IP: s=204.12.1.4 (local), d=255.255.255.255 (Ethernet0/0), len 54,
sending broad/multicast
UDP src=54655, dst=53
IP: s=155.1.0.4 (local), d=255.255.255.255 (Serial0/0), len 54, sending
broad/multicast
UDP src=54655, dst=53
IP: s=155.1.146.4 (local), d=255.255.255.255 (Ethernet0/1), len 54,
sending broad/multicast
UDP src=54655, dst=53
```


SW2 receives the DNS request sent from R4 as a directed broadcast through R5.

```
Rack1SW2#
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255 (Vlan58), len 54, rcvd 3
    UDP src=54655, dst=53
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255, len 54, stop process pak for
forus packet
    UDP src=54655, dst=53
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255 (Vlan58), len 54, rcvd 3
    UDP src=54655, dst=53
IP: s=155.1.45.4 (Vlan58), d=155.1.58.255, len 54, stop process pak for
forus packet
```


13.40 DRP Server Agent

- Configure R1 to support reporting of information to a Distributed Director.
- Only allow connections from Directors from VLAN 146, authenticate the communication using the key value CISCO.

Configuration

```
R1:
ip drp access-group 99
ip drp authentication key-chain DRP
ip drp server
!
key chain DRP
 key 1
  key-string CISCO
!
access-list 99 permit 155.1.146.0 0.0.0.255
```

Verification

```
Rack1R1#show ip drp
Director Responder Protocol Agent is enabled
0 director requests:
0 successful route table lookups
0 successful measured lookups
0 no route in table
0 nortt
0 DRP packet failures returned
0 successful echos
0 Boomerang requests
0 Boomerang-raced DNS responses
Authentication is enabled, using "DRP" key-chain
Director requests filtered by access-list 99
rttprobe source port is     : 53
rttprobe destination port is: 53
```


13.41 WCCPv1 Web-Cache

- Configure R4 and R6 for WCCPv1 web-cache redirection.
- R4 should redirect outgoing packets on its VLAN 43 interface, and R6 should redirect incoming packets on its VLAN 146 connection.
- Exclude packets coming in R4's Serial connection to R5 from redirection.
- Exclude connections sourced from the VLAN 146 subnet from redirection.
- Do not redirect flows excluded by outbound ACLs.

Configuration

```
R4:
access-list 199 deny ip 155.1.146.0 0.0.0.255 any
access-list 199 permit ip any any
!
ip wccp web-cache redirect-list 199
ip wccp outbound-acl-check
!
interface FastEthernet 0/0
 ip wccp web-cache redirect out
!
interface Serial 0/1
 ip wccp redirect exclude in

R6:
access-list 199 deny ip 155.1.146.0 0.0.0.255 any
access-list 199 permit ip any any
!
ip wccp version 1
ip wccp web-cache redirect-list 199
ip wccp outbound-acl-check
!
interface FastEthernet 0/0.146
 ip wccp web-cache redirect in
```


Verification

Note

Once a WCCP cache server is configured to register with the router, the router starts redirecting HTTP connections to the cache server. Note that the direction of the redirection indicates which traffic flows are redirected, either inbound or outbound flows, not where the cache engine is located. Verification of this feature is not available without an actual server that supports the WCCP protocol, however the basic WCCP configuration status can be verified as follows.

```
Rack1R4#show ip wccp
Global WCCP information:
    Router information:
      Router Identifier:                   150.1.4.4
      Protocol Version:                    1.0

    Service Identifier: web-cache
      Number of Service Group Clients:     0
      Number of Service Group Routers:     1
      Total Packets s/w Redirected:        0
        Process:                           0
        Fast:                              0
        CEF:                               0
      Redirect access-list:                199
      Total Packets Denied Redirect:       0
      Total Packets Unassigned:            0
      Group access-list:                   -none-
      Total Messages Denied to Group:      0
      Total Authentication failures:       0

Rack1R4#show ip wccp interface
WCCP interface configuration:
    Ethernet0/0
      Output services: 1
      Input services:  0
      Mcast services:  0
      Exclude In:      FALSE

    Serial0/1
      Output services: 0
      Input services:  0
      Mcast services:  0
      Exclude In:      TRUE
```



```
Rack1R6#show ip wccp
Global WCCP information:
    Router information:
      Router Identifier:                   150.1.6.6
      Protocol Version:                    1.0

    Service Identifier: web-cache
      Number of Service Group Clients:     0
      Number of Service Group Routers:     1
      Total Packets s/w Redirected:        0
        Process:                           0
        Fast:                              0
        CEF:                               0
      Redirect access-list:                199
      Total Packets Denied Redirect:       0
      Total Packets Unassigned:            0
      Group access-list:                   -none-
      Total Messages Denied to Group:      0
      Total Authentication failures:       0

Rack1R6#show ip wccp  interface
WCCP interface configuration:
    FastEthernet0/0.146
      Output services: 0
      Input services:  1
      Mcast services:  0
      Exclude In:      FALSE
```


13.42 WCCPv2 Services

- Configure R5 to listen for WCCPv2 cache engines on its FastEthernet0/0 link using a dynamic service-group numbered 50.
- The cache engines will be sending packets to the multicast address 224.0.1.100.
- Authenticate them with the password CISCO.
- Only accept packets from the cache engines 155.X.58.100 and 155.X.58.200.
- Redirect traffic from hosts in the VLAN 58 subnet to these cache engines.

Configuration

```
R5:
access-list 10 permit 155.1.58.200
access-list 10 permit 155.1.58.100
!
access-list 20 permit 155.1.58.0 0.0.0.255
!
ip wccp version 2
ip wccp 50 group-address 224.0.1.100 redirect-list 20 group-list 10
password CISCO
!
interface FastEthernet 0/0
 ip wccp 50 redirect in
 ip wccp 50 group-listen
```

Verification

Note

WCCPv2 allows the cache engine to support “dynamic service-groups” with the router. During the registration process the router is told which protocols and ports it should redirect to the cache engines. The router must be configured for the service-group ID before the cache engines register the service-group.

Additionally, WCCPv2 enhancements allow many routers to communicate with a cache cluster, and use multicast packets for more effective signaling. WCCPv2 also allows authenticating communications using an MD5 hashing scheme.

To verify the configuration check the WCCPv2 service group 50 status.


```
Rack1R5#show ip wccp 50
Global WCCP information:
    Router information:
      Router Identifier:                   -not yet determined-
      Protocol Version:                    2.0

    Service Identifier: 50
      Number of Service Group Clients:     0
      Number of Service Group Routers:     0
      Total Packets s/w Redirected:        0
        Process:                           0
        Fast:                              0
        CEF:                               0
      Redirect access-list:                20
      Total Packets Denied Redirect:       0
      Total Packets Unassigned:            0
      Group access-list:                   10
      Total Messages Denied to Group:      0
      Total Authentication failures:       0
      Total Bypassed Packets Received:     0

Rack1R5#show ip wccp interfaces
WCCP interface configuration:
    FastEthernet0/0
      Output services: 0
      Input services:  1
      Mcast services:  1
      Exclude In:      FALSE
```


13.43 SLB Directed Mode

- Configure Server Load Balancing on R5 to distribute connections going to virtual the HTTP server IP address 155.X.58.55.
- The real hosts should consist of the Loopback0 IP addresses of R1, R2 and R3.
- Configure R5 so that traffic to the virtual address is NAT translated, and load balanced in a round robin fashion with a ratio of 1:2:3 between R1, R2, and R3 respectively.
- Reassign a client to another server after two failed initial SYN packets.
- Detect a failed server by counting to 3 failed connections, but retry a failed server after 120 seconds.

Configuration

```
R5:
ip slb serverfarm SERVERFARM
  nat server
  predictor roundrobin
  real 150.1.1.1
    reassign 2
    faildetect numconns 3
    retry 120
    weight 1
    inservice
    exit
  real 150.1.2.2
    reassign 2
    faildetect numconns 3
    retry 120
    weight 2
    inservice
    exit
  real 150.1.3.3
    reassign 2
    faildetect numconns 3
    retry 120
    weight 3
    inservice
    end
!
ip slb vserver VSERVER
  virtual 155.1.58.55 tcp telnet
  serverfarm SERVERFARM
  inservice
  exit
```


Verification

Note

In directed mode, Server Load Balancing (SLB) uses NAT to redirect connections to a real server. This design impacts performance, but removes the restriction that the physical servers need to be attached to the load-balancing router. The SLB process does not use the IOS NAT and IOS ALGs (Application Level Gateways), but instead utilizes its own NAT engine. This limits SLB directed mode to protocols that are not affected by NAT, such as HTTP or Telnet. Directed SLB mode will not work with FTP, since this application needs the NAT ALG to work properly.

The basic SLB configuration can be verified as follows.

```
Rack1R5#show ip slb vservers

slb vserver      prot  virtual               state         conns
-------------------------------------------------------------------
VSERVER          TCP   155.1.58.55:23        INSERVICE     0

Rack1R5#show ip slb serverfarms

server farm      predictor    nat   reals   bind id
-----------------------------------------------
SERVERFARM       ROUNDROBIN   S     3       0

Rack1R5#show ip slb reals

real                  server farm      weight  state          conns
-------------------------------------------------------------------
150.1.1.1             SERVERFARM       1       OPERATIONAL    0
150.1.2.2             SERVERFARM       2       OPERATIONAL    0
150.1.3.3             SERVERFARM       3       OPERATIONAL    0
```


To verify the operation of SLB, open multiple connections to the virtual server. Souce these connections from different IP addresses, such as on SW2 and SW4, to simulate multiple clients. Do not disconnect the previous connections while testing, rather escape them and open new ones in parallel with the old.

```
Rack1SW4#telnet 155.1.58.55
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R2>

Rack1SW4#telnet 155.1.58.55 /source Loopback0
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R2>

Rack1SW4#telnet 155.1.58.55 /source vlan 10
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R3>

Rack1SW2#telnet 155.1.58.55
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R1>

Rack1SW2#telnet 155.1.58.55 /source-interface loopback 0
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R2>

Rack1SW2#telnet 155.1.58.55 /source-interface port-channel 1
Trying 155.1.58.55 ... Open

User Access Verification

Password:
Rack1R2>
```



```
Rack1R5#show ip slb conns

vserver         prot client               real                 state     nat
-------------------------------------------------------------------------------
VSERVER         TCP  155.1.108.8:11019     150.1.2.2            ESTAB     S
VSERVER         TCP  150.1.8.8:11018       150.1.2.2            ESTAB     S
VSERVER         TCP  155.1.58.8:11017      150.1.1.1            ESTAB     S
VSERVER         TCP  150.1.10.10:11004     150.1.2.2            ESTAB     S
VSERVER         TCP  155.1.10.10:11005     150.1.3.3            ESTAB     S
VSERVER         TCP  155.1.108.10:11003    150.1.2.2            ESTAB     S

Rack1R5#show ip slb stats
Pkts via normal switching:  10252
Pkts via special switching: 0
Connections Created:        65
Connections Established:    18
Connections Destroyed:      65
Connections Reassigned:     282
Zombie Count:               0
```


13.44 SLB Dispatched Mode

- Create additional Loopback1 interfaces on R1 and R6 with the same IP address 100.X.100.100/32 but do not advertise them into IGP.
- Configure an SLB server farm on R4 so that packets sent to the address 100.X.100.100 are redirected to the VLAN 146 IP addresses of R1 and R6.
- Load balance the connections equally, and limit the number of maximum connections to two per physical server.

Configuration

```
R1:
interface Loopback 1
 ip address 100.1.100.100 255.255.255.255

R6:
interface Loopback 1
 ip address 100.1.100.100 255.255.255.255

R4:
ip slb serverfarm SERVERFARM
  predictor roundrobin
  real 155.1.146.1
    weight 1
    maxconns 2
    inservice
    exit
  real 155.1.146.6
    weight 1
    maxconns 2
    inservice
    exit
!
ip slb vserver VSERVER
  virtual 100.1.100.100 tcp telnet
  advertise
  serverfarm SERVERFARM
  inservice
  exit
```


Verification

> **Note**
>
> SLB dispatched mode uses layer 2 modifications only to forward packets to the real servers, and does not do any layer 3 IP packet modifications. This means that the real servers each are assigned a unique IP address, and share an overlapping anycast address to accept packets directed to the virtual server.
>
> In dispatched mode SLB tracks the connection status and server failures, and implements a load-balancing algorithm. Note that the weight setting actually specifies the number of connections a server can accept before the load-balancer moves to another server in the physical farm.

```
BB3>telnet 100.1.100.100
Trying 100.1.100.100 ... Open

User Access Verification

Password:
Rack1R6>

Rack1R6>exit

[Connection to 100.1.100.100 closed by foreign host]

BB3>telnet 100.1.100.100
Trying 100.1.100.100 ... Open

User Access Verification

Password:
Rack1R1>

Rack1R4#show ip slb serverfarm detail
SERVERFARM, predictor = ROUNDROBIN, nat = none
            virtuals inservice: 1, reals = 2, bind id = 0
  Real servers:
    155.1.146.1, weight = 1, MAXCONNS, conns = 1
    155.1.146.6, weight = 1, OPERATIONAL, conns = 0
  Total connections = 1

Rack1R4#show ip slb vserver

slb vserver      prot  virtual               state         conns
-----------------------------------------------------------------
VSERVER          TCP   100.1.100.100:23      INSERVICE     1
```

13.45 NBAR Protocol Discovery

- Configure R4 and R6 to collect protocol statistics on their connections to backbone routers using NBAR.
- Both routers should be able to classify connections to HTTP proxy ports 3128 and 8080.
- Change the SOCKS protocol mapping to port 2080.
- Create a new custom protocol mapping called TEST that matches the ASCII character “A” in the beginning of a TCP segment flowing to the destination port 3001.

Configuration

```
R4:
interface FastEthernet 0/0
 ip nbar protocol-discovery
!
ip nbar custom HTTP_PROXY destination tcp 3128 8080
ip nbar port-map socks tcp 2080
ip nbar custom TEST 0 ascii A destination tcp 3001

R6:
interface Serial 0/0
 ip nbar protocol-discovery
!
ip nbar custom HTTP_PROXY destination tcp 3128 8080
ip nbar port-map socks tcp 2080
ip nbar custom TEST 0 ascii A destination tcp 3001
```

Verification

Note

NBAR protocol discovery is used for network traffic monitoring. It displays the protocols flowing across a particular interface along with their statistics. As shown in this example you can also define new protocols using port-mapping or a low-level byte string match.

Rack1R4#show ip nbar protocol-discovery

```
 FastEthernet0/0
                            Input                     Output
                            -----                     ------
   Protocol                 Packet Count              Packet Count
                            Byte Count                Byte Count
                            5min Bit Rate (bps)       5min Bit Rate (bps)
                            5min Max Bit Rate (bps)   5min Max Bit Rate (bps)
   ------------------------ ------------------------- -------------------------
   rip                      52                        154
                            10712                     60984
                            0                         0
                            0                         0
   bgp                      10                        5
                            600                       300
                            0                         0
                            0                         0
   HTTP_PROXY               0                         0
<snip>
```

Rack1R1#ping 30.0.0.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
Rack1R1#
```

Rack1R4#show ip nbar protocol-discovery

```
 FastEthernet0/0
                            Input                     Output
                            -----                     ------
   Protocol                 Packet Count              Packet Count
                            Byte Count                Byte Count
                            5min Bit Rate (bps)       5min Bit Rate (bps)
                            5min Max Bit Rate (bps)   5min Max Bit Rate (bps)
   ------------------------ ------------------------- -------------------------
   rip                      98                        292
                            20188                     115632
                            0                         0
                            0                         0
   bgp                      22                        11
                            1320                      660
                            0                         0
                            0                         0
   icmp                     5                         5
                            570                       570
                            0                         0
                            0                         0
   HTTP_PROXY               0                         0
                            0                         0
                            0                         0
                            0                         0
```


To verify a byte-string match on R4, configure R1 to listen on port 3001 and initiate a TCP connection to it.

```
Rack1R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R1(config)#line vty 0
Rack1R1(config-line)#rotary 1

BB3>telnet 150.1.1.1 3001
Trying 150.1.1.1, 3001 ... Open

User Access Verification

Password: cisco
Rack1R1>AAAAA
Translating "AAAAA"

Translating "AAAAA"
% Unknown command or computer name, or unable to find computer address
```

```
Rack1R4#show ip nbar protocol-discovery protocol TEST

 FastEthernet0/0
                          Input                    Output
                          -----                    ------
   Protocol               Packet Count             Packet Count
                          Byte Count               Byte Count
                          5min Bit Rate (bps)      5min Bit Rate (bps)
                          5min Max Bit Rate (bps)  5min Max Bit Rate (bps)
   ---------------------- ------------------------ ------------------------
   TEST                   9                        7
                          540                      509
                          0                        0
                          0                        0
```

The port values that NBAR uses for matching can be listed as follows.

```
Rack1R1#show ip nbar port-map
port-map HTTP_PROXY               tcp 3128 8080
port-map TEST                     tcp 3001
port-map bgp                      udp 179
port-map bgp                      tcp 179
port-map citrix                   udp 1604
port-map citrix                   tcp 1494
port-map cuseeme                  udp 7648 7649 24032
port-map cuseeme                  tcp 7648 7649
port-map dhcp                     udp 67 68
port-map dns                      udp 53
port-map dns                      tcp 53
port-map edonkey                  tcp 4662
port-map exchange                 tcp 135
port-map fasttrack                tcp 1214
port-map finger                   tcp 79
<snip>
```


13.46 Netflow Ingress & Egress

- Configure R4 and R6 to export Netflow version 5 information to a host 155.X.146.100 in VLAN 146 using port 9999.
- R4 should collect statistics on incoming flows on all interfaces, while R6 should collect statistics on outgoing flows on all interfaces.
- Limit the size of the flow cache to 4096 entries and export the BGP AS of Origin if available.
- Collect additional information about ICMP message information as well as layer 2 VLAN numbers where applicable.

Configuration

```
R4:
ip flow-capture vlan-id
ip flow-capture icmp
ip flow-export version 5
ip flow-export destination 155.1.146.100 9999
ip flow-cache entries 4096
!
interface Serial 0/0
 ip flow ingress
!
interface Serial 0/1
 ip flow ingress
!
interface FastEthernet 0/0
 ip flow ingress
!
interface FastEthernet 0/1
 ip flow ingress

R6:
ip flow-capture vlan-id
ip flow-capture icmp
ip flow-export version 5
ip flow-export destination 155.1.146.100 9999
ip flow-cache entries 4096
!
interface Serial 0/0
 ip flow egress
!
interface FastEthernet 0/0.67
 ip flow egress
!
interface FastEthernet 0/0.146
 ip flow egress
```


Verification

Note

Netfow evolved from a route-caching (switching path) technology to a powerful statistics-gathering tool. Flow collection on interfaces can be enabled for either inbound (ingress) or outbound (egress) flow. The difference between them is that ingress flow monitoring will capture all flows entering the router, including the packets destined to the router itself. Egress flow collects packets switched across the router that are leaving via an egress interface, but does not include packets originated from the router itself.

Egress flow collection has some performance penalties, but is useful when you need to collect Netflow information on MPLS traffic, which leaves the router untagged. This is needed since ingress Netflow can't look inside the MPLS tagged packets. You may distinguish egress flows in the show command output by recognizing a "*" sign near the destination interface. Another difference between ingress and egress Netflow is their interaction with other IOS features, such as NAT, rate-limiting, and encryption. Ingress Netflow is applied before ACLs and rate-limiting, while egress Netflow is applied after rate-limiting and encryption features.

Different versions of Netflow are available, with the most basic, widely available, and default version being 5. Version 8 supports aggregation caches, and version 9, which is incompatible with previous versions, allows for the most flexibility with choosing the export template format. Aggregation caches are also supported by version 9, so version 8 is becoming deprecated.

To verify this configuration, generate some flows and check verbose flow-cache output as follows.

```
Rack1R1#telnet 212.18.1.1
Trying 212.18.1.1 ... Open

<snip>

Rack1R6#show ip cache verbose flow
IP packet size distribution (30 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .566 .000 .333 .033 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .066 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  2 active, 4094 inactive, 4 added
  50 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  4 active, 1020 inactive, 8 added, 4 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
ICMP                 2      0.0         5   100      0.0       0.1      15.9
Total:               2      0.0         5   100      0.0       0.1      15.9

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active
Fa0/0.146      155.1.146.1     Se0/0*         212.18.1.1      06 C0  1A      10
2AF8 /24 0                     0017 /24 0     54.1.1.254             43     0.4
FFlags: 01
VLAN id:       146                           000
Min plen:       40                           Max plen:        52

Se0/0          212.18.1.1      Fa0/0.146*     155.1.146.1     06 C0  1A      10
0017 /24 0                     2AF8 /24 0     155.1.146.1           160     0.6
FFlags: 01
VLAN id:       000                           146
Min plen:       40                           Max plen:       576
```



```
Rack1R1#ping 30.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

Rack1R4#show ip cache verbose flow
IP packet size distribution (190 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .015 .000 .105 .000 .052 .000 .000 .168 .000 .000 .000 .494 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .163 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  6 active, 4090 inactive, 21 added
  1389 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  12 active, 1012 inactive, 42 added, 21 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-BGP              3      0.0         1    44      0.0       0.0      15.7
UDP-other           10      0.0         1   192      0.0       0.0      15.5
ICMP                 2      0.0         5   100      0.0       0.0      15.5
Total:              15      0.0         1   132      0.0       0.0      15.5

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active
Et0/1          155.1.146.1     Et0/0          30.0.0.1        01 00  10       5
0000 /24 0                     0800 /16 0     204.12.1.254           100     0.0
VLAN id:       000                            000
Min plen:      100                            Max plen:       100
ICMP type:       8                            ICMP code:        0

Se0/1          155.1.45.5      Null           224.0.0.9       11 C0  10      32
0208 /24 0                     0208 /24 0     0.0.0.0                412   284.7
VLAN id:       000                            000
Min plen:      412                            Max plen:       412

<snip>

Et0/0          30.0.0.1        Et0/1          155.1.146.1     01 00  10       5
0000 /16 0                     0000 /24 0     155.1.146.1            100     0.0
VLAN id:       000                            000
Min plen:      100                            Max plen:       100
ICMP type:       0                            ICMP code:        0
```


13.47 Netflow Top Talkers

- Configure R4 and R6 to collect information about 10 most active ICMP flows.
- R4 should count packets, while R6 counts bytes.
- Match only TCP and ICMP flows coming from the subnet 155.X.0.0/16.

Configuration

```
R4:
ip flow-top-talkers
 top 10
 sort-by packets
 match source address 155.1.0.0 255.255.0.0
 match protocol 1

R6:
ip flow-top-talkers
 top 10
 sort-by bytes
 match source address 155.1.0.0 255.255.0.0
 match protocol 1
```


Verification

> **Note**
>
> The top talkers feature collects information about the most packet or traffic-intense flows. The information is displayed only about the active, non-expired flows, so you may miss some short and bursty transmission. A flexible filtering system allows customizing the view per your requirements. The feature can be verified as follows.

```
Rack1R1#ping 212.18.1.1 repeat 50

Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 212.18.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 32/35/44 ms

Rack1R1#ping 30.0.0.1 repeat 50

Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 4/5/8 ms

Rack1R4#show ip flow top-talkers

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Et0/1         155.1.146.1     Et0/0         30.0.0.1        01 0000 0800    50
1 of 10 top talkers shown. 1 of 7 flows matched.

Rack1R6#show ip flow top-talkers

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP Bytes
Fa0/0.146     155.1.146.1     Se0/0*        212.18.1.1      01 0000 0800  5000
1 of 10 top talkers shown. 1 of 2 flows matched.
```


13.48 Netflow Aggregation Cache

- Create an aggregation cache that is destination-prefix based on R4 and source-based on R6.
- Export the new information to 155.X.146.100 at port 9998 with Netflow version 8.
- Do not summarize addresses to more than a /8 prefix length.
- Store no more than 1024 aggregated entries.

Configuration

```
R4:
ip flow-aggregation cache destination-prefix
 cache entries 1024
 export destination 155.1.146.100 9998
 mask destination minimum 8
 enabled

R6:
ip flow-aggregation cache source-prefix
 cache entries 1024
 export destination 155.1.146.100 9998
 mask source minimum 8
 enabled
```

Verification

Note

The Netflow aggregation cache is an additional flow cache used to store aggregated prefix information. A special Netflow packet format that was first supported in version 8, and more recently in version 9, is used to export the aggregated information to a collector.

The aggregation cache is based on the main (normal) flow cache, and flow aggregation is based on the routing table contents. For example when a flow is exported from the main cache to the aggregation cache, the flow's source and/or destination IP addresses (based on the configured scheme) are logically ANDed bitwise with the network masks of its corresponding route in the routing table. The resulting "masked" (less-specific) flow entry is merged with the aggregation cache. This will result in multiple flow entries with source or destination IP addresses belonging to the same prefix in IP routing table looking the same in aggregation cache, with the bytes and/or packet count being the sum of all specific flows. Additionally the number of merged flows for each aggregated entry is also stored in aggregation cache.

An additional feature, called the minimum aggregation mask, allows preserving a specified level of granularity even when widely scoped summary routes are installed in the routing table. For example if the router has the prefix 10.0.0.0/8 in its routing table, you may still want to know detailed flow information information about the subnets 10.X.X.0. By setting the minimum aggregation mask to /24 the aggregation cache will choose the longer of the two masks when creating an entry. If a flow source with the IP address 10.1.2.3 is about to be aggregated, the routing table mask of /8 is compared against the minimum aggregation mask of /24, and the result is that the router installs the aggregate entry as 10.1.2.0.

```
Rack1R1#ping 30.2.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.2.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

Rack1R1#ping 30.2.0.254 time 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.2.0.254, timeout is 0 seconds:
.....
Success rate is 0 percent (0/5)

Rack1R1#ping 30.1.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

Rack1R1#ping 30.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
```

In the above case is to be summarized based on destination addresses. On R1 all prefixes in routing table for the subnets of 30.0.0.0 have prefix length of 16. Since max(minimum-mask, routing-prefix) = max(8,16) = 16, the prefix length in the aggregation table is /16.

Note that prefix 30.2.0.0/16 has twice as many packets as the other due to the two pings to destinations in that range.


```
Rack1R4#show ip route 30.0.0.0
Routing entry for 30.0.0.0/16, 4 known subnets
  Redistributing via rip

R       30.2.0.0 [120/1] via 204.12.1.254, 00:00:23, Ethernet0/0
R       30.3.0.0 [120/1] via 204.12.1.254, 00:00:23, Ethernet0/0
R       30.0.0.0 [120/1] via 204.12.1.254, 00:00:23, Ethernet0/0
R       30.1.0.0 [120/1] via 204.12.1.254, 00:00:23, Ethernet0/0

Rack1R4#show ip cache flow aggregation destination-prefix

IP Flow Switching Cache, 69636 bytes
  5 active, 1019 inactive, 150 added
  2336 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 5896 bytes
  5 active, 251 inactive, 150 added, 150 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added

Minimum destination mask is configured to /8

Dst If         Dst Prefix      Msk  AS    Flows  Pkts B/Pk  Active
Null           224.0.0.0       /24  0         1     1   192     0.0
Et0/1          155.1.146.0     /24  0         3    15   100    14.7
Et0/0          30.2.0.0        /16  0         2    10   100     8.2
Et0/0          30.0.0.0        /16  0         1     5   100     0.0
Et0/0          30.1.0.0        /16  0         1     5   100     0.0
```


13.49 Netflow Random Sampling

- Modify R6's Netflow configuration so that it only samples every 10th packet in each flow.

Configuration

```
R6:
flow-sampler-map SAMPLER
 mode random one-out-of 10
!
policy-map NETFLOW_MAP
 class class-default
   netflow-sampler SAMPLER
!
interface Serial 0/0
 no ip flow egress
 service-policy output NETFLOW_MAP
!
interface FastEthernet 0/0.146
 no ip flow egress
 service-policy output NETFLOW_MAP
!
interface FastEthernet 0/0.67
 no ip flow egress
 service-policy output NETFLOW_MAP
```

Verification

Note

Flow sampling allows you to lower the amount of Netflow information collected, which lowers the performance impact on the router collecting statistics. Commonly random sampling is used for the purpose of capacity planning, where precise numbers are not required, but statistical information is required. The random flow sampler randomly picks every 1 of Nth packets for each flow and uses it to build flow information. When a flow sampler is set to sample 1 of 1, it essentially behaves in normal flow-collecting mode.

The application of the flow samplers is through an MQC policy-map. An outgoing service policy will implement egress Netflow, and an incoming service policy implements ingress Netflow. The interface-level netflow configuration will override the MQC settings, so always remove the **`ip flow ingress`** or **`ip flow egress`** commands when applying Netflow samplers.

When a sampler is applied in the class-default of a policy-map it effectively applies to all traffic entering or leaving the respective interface, provided that traffic does not match any other classes in the policy-map.


```
Rack1R1#ping 212.18.2.1 repeat 500

Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 212.18.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!
<snip>

Rack1R6#show ip cache flow
IP packet size distribution (1631 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .068 .000 .925 .001 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .003 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  2 active, 4094 inactive, 28 added
  452 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  4 active, 1020 inactive, 56 added, 28 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          14      0.0         8    72      0.0       1.4       9.6
ICMP                12      0.0       117   100      0.0       4.5      15.4
Total:              26      0.0        58    97      0.0       2.8      12.3

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0.146     155.1.146.1     Se0/0*        212.18.2.1      01 0000 0800    57
Se0/0         212.18.2.1      Fa0/0.146*    155.1.146.1     01 0000 0000    43
```

Note that the number of packets accounted is approximately 10 times less than the number of packets actually sent and received. As the number of packets grows, the relation will tend to be more precise, so for the purpose of statistical analysis it is possible to simply scale the numeric values collected by 10 to get realistic results.


```
Rack1R6#show policy-map interface serial 0/0

      netflow-sampler: SAMPLER
 Serial0/0

  Service-policy output: NETFLOW_MAP

    Class-map: class-default (match-any)
      1503 packets, 287842 bytes
      5 minute offered rate 2000 bps, drop rate 0 bps
      Match: any

Rack1R6#show policy-map int fastEthernet 0/0.146
      netflow-sampler: SAMPLER
 FastEthernet0/0.146

  Service-policy output: NETFLOW_MAP

    Class-map: class-default (match-any)
      1066 packets, 136236 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

13.50 Netflow Input Filters

- Modify R6’s Netflow sampling so that every packet is sampled for sources on VLAN 146.
- Other packets should still be randomly sampled.

Configuration

```
R6:
ip access-list extended VLAN146
 permit ip 155.1.146.0 0.0.0.255 any
 permit ip any 155.1.146.0 0.0.0.255
!
class-map VLAN146
 match access-group name VLAN146
!
flow-sampler-map NORMAL
 mode random one-out-of 1
!
policy-map NETFLOW_MAP
 class VLAN146
   netflow-sampler NORMAL
```

Verification

Note

Netflow input filters allow you to attach a flow sampler to any class in a policy-map. In this example class VLAN146 samples 1/1 packets, while class-default (everything else) still samples 1/10 packets.

```
Rack1R1#ping 212.18.2.1 repeat 500

Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 212.18.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
Success rate is 100 percent (500/500), round-trip min/avg/max =
32/37/408 ms
```

```
Rack1R6#show ip cache flow
IP packet size distribution (2631 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .042 .000 .954 .001 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .002 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  2 active, 4094 inactive, 30 added
  528 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  4 active, 1020 inactive, 60 added, 30 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          14      0.0         8    72      0.0       1.4       9.6
ICMP                14      0.0       107   100      0.0       6.5      15.4
Total:              28      0.0        58    97      0.0       3.9      12.5

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Se0/0         212.18.2.1      Fa0/0.146*    155.1.146.1     01 0000 0000   500
Fa0/0.146     155.1.146.1     Se0/0*        212.18.2.1      01 0000 0800   500
```

Generate traffic from a subnet that is classified by class-default in the policy-map. Note that this traffic is statistically sampled.

```
Rack1R1#ping 212.18.2.1 repeat 500 source loopback 0

Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 212.18.2.1, timeout is 2 seconds:
Packet sent with a source address of 150.1.1.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
Success rate is 100 percent (500/500), round-trip min/avg/max =
32/37/400 ms
Rack1R1#
```

```
Rack1R6#show ip cache flow
IP packet size distribution (2731 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .041 .000 .955 .001 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .002 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  2 active, 4094 inactive, 32 added
  594 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  4 active, 1020 inactive, 64 added, 32 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          14      0.0         8    72      0.0       1.4       9.6
ICMP                16      0.0       156   100      0.0       8.1      15.4
Total:              30      0.0        87    98      0.0       4.9      12.7

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0.146     150.1.1.1       Se0/0*        212.18.2.1      01 0000 0800    47
Se0/0         212.18.2.1      Fa0/0.146*    150.1.1.1       01 0000 0000    53
```

13.51 IOS Authoritative DNS Server

- Configure R4 and R6 as primary DNS servers for the domain cisco.com.
- The primary name server should be called ns.cisco.com with a contact mailbox of ccie.cisco.com.
- Both R4 and R6 should resolve the names R4.cisco.com and R6.cisco.com to all connected IP addresses of the respective router.
- R1 should use both R4 and R6 as its DNS servers in a round-robin fashion.
- R1's DNS client should complete all unqualified domain-names with the name "cisco.com".

Configuration

```
R1:
ip name-server 155.1.146.4 155.1.146.6
ip domain-lookup
ip domain name cisco.com
ip domain round-robin

R4:
ip dns server
ip dns primary cisco.com soa ns.cisco.com ccie.cisco.com
!
ip host cisco.com ns 155.1.146.4
ip host cisco.com ns 155.1.146.6
!
ip host R4.cisco.com 150.1.4.4 155.1.146.4 155.1.45.4 204.12.1.4
155.1.0.4
ip host R6.cisco.com 150.1.6.6 155.1.146.6 155.1.67.6 54.1.1.6

R6:
ip dns server
!
ip dns server
ip dns primary cisco.com soa ns.cisco.com ccie.cisco.com
!
ip host cisco.com ns 155.1.146.4
ip host cisco.com ns 155.1.146.6
!
ip host R4.cisco.com 150.1.4.4 155.1.146.4 155.1.45.4 204.12.1.4
155.1.0.4
ip host R6.cisco.com 150.1.6.6 155.1.146.6 155.1.67.6 54.1.1.6
```


Verification

> **Note**
>
> IOS supports basic DNS server resolution, in addition to SOA, MX, NS, SRV resource records to act as an authoritative DNS server.
>
> The IOS DNS client ican load-balance between multiple A entries returned in a DNS response in a round-robin fashion. You can specify the default domain name to complete “unqualified” names (such as “R4”) to fully-qualified DNS names (“R4.cisco.com”).

```
Rack1R4#show ip dns primary
Primary for zone cisco.com:
  SOA information:
  Zone primary (MNAME): ns.cisco.com
  Zone contact (RNAME): ccie.cisco.com
  Refresh (seconds):    21600
  Retry (seconds):      900
  Expire (seconds):     7776000
  Minimum (seconds):    86400

Rack1R6#show ip dns primary
Primary for zone cisco.com:
  SOA information:
  Zone primary (MNAME): ns.cisco.com
  Zone contact (RNAME): ccie.cisco.com
  Refresh (seconds):    21600
  Retry (seconds):      900
  Expire (seconds):     7776000
  Minimum (seconds):    86400

Rack1R1#ping R4
Translating "R4"...domain server (155.1.146.4) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.45.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/100 ms

Rack1R1#ping R4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 204.12.1.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/100 ms
```



```
Rack1R1#ping R4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/116/124 ms

Rack1R1#show hosts
Default domain is cisco.com
Name/address lookup uses domain service
Name servers are 155.1.146.4, 155.1.146.6

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
R4.cisco.com              None  (temp, OK)  0   IP    155.1.0.4
                                                      150.1.4.4
                                                      155.1.146.4
                                                      155.1.45.4
                                                      204.12.1.4
```

```
Rack1R1#ping R6
Translating "R6"...domain server (155.1.146.4) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.67.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/104 ms

Rack1R1#ping R6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 54.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/104 ms

Rack1R1#ping R6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/104 ms

Rack1R1#ping R6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.146.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Rack1R1#show hosts
Default domain is cisco.com
Name/address lookup uses domain service
Name servers are 155.1.146.4, 155.1.146.6

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
R6.cisco.com              None  (temp, OK)  0   IP    155.1.146.6
                                                      155.1.67.6
                                                      54.1.1.6
                                                      150.1.6.6
```

13.52 IOS Caching DNS Server

- Configure R3 as a DNS client of R1.
- When R1 receives a DNS query from R3 it should check its local cache.
- If the entry is not cached a request should be sent from R1 to R4 or R6 on behalf of R3.

Configuration

```
R1:
ip name-server 155.1.146.4 155.1.146.6
ip domain-lookup
ip dns sever

R3:
ip name-server 150.1.1.1
ip domain-lookup
```

Verification

> **Note**
>
> To verify this configuration issue ping commands from R3 using R4 and R6's fully-qualified domain-names.

```
Rack1R3#ping R4.cisco.com.

Translating "R4.cisco.com."...domain server (150.1.1.1) [OK]

Translating "R4.cisco.com."...domain server (150.1.1.1) [OK]

Translating "R4.cisco.com."...domain server (150.1.1.1) [OK]

Translating "R4.cisco.com."...domain server (150.1.1.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/128/132 ms

Rack1R3#ping R6.cisco.com

Translating "R6.cisco.com"...domain server (150.1.1.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 144/144/144 ms
```


Debug DNS server activities on R1 and send a DNS query from R3.

```
Rack1R1#debug domain
Domain Name System debugging is on

Rack1R3#ping R6.cisco.com

Translating "R6.cisco.com"...domain server (150.1.1.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 144/152/188 ms

Rack1R1#
DNS: Incoming UDP query (id#43)
DNS: Type 1 DNS query (id#43) for host 'R6.cisco.com' from
155.1.13.3(55196)
DNS: Re-sending DNS query (type 1, id#43) to 155.1.146.4
DNS: Incoming UDP query (id#43)
DNS: Type 1 response (id#43) for host <R6.cisco.com> from
155.1.146.4(53)
DOM: dom2cache: hostname is R6.cisco.com, RR type=1, class=1, ttl=10,
n=4
DOM: dom2cache: hostname is R6.cisco.com, RR type=1, class=1, ttl=10,
n=4
DOM: dom2cache: hostname is R6.cisco.com, RR type=1, class=1, ttl=10,
n=4
DOM: dom2cache: hostname is R6.cisco.com, RR type=1, class=1, ttl=10,
n=4
DNS: Forwarding back A response - no director required
DNS: Finished processing query (id#43) in 0.012 secs
DNS: Forwarding back reply to 155.1.13.3/55196
```


13.53 IOS DNS Spoofing

- Configure R5 as a DNS client of R4.
- Configure SW2 as a DNS client of R5.
- If R5 loses connectivity to R4 it should respond to all DNS queries with the IP address of its Loopback0 interface.

Configuration

```
R5:
ip dns spoofing 150.1.5.5
ip name-server 155.1.4.4
ip domain lookup
ip dns server

SW2:
ip domain lookup
ip name-server 155.1.58.5
```

Verification

> **Note**
>
> A router acting as a DNS caching server with just one upstream interface will forward DNS requests to its upstream server as long as the connection is healthy. When the router loses reachability to the DNS servers, it will respond to all DNS queries with a pre-configured IP address, i.e. with the IP address of an internal HTTP server to display an informational message.

```
Rack1SW2#ping R4.cisco.com.

Translating "R4.cisco.com."...domain server (155.1.58.5) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 42/45/51 ms

Rack1SW2#ping TEST

Translating "TEST"...domain server (155.1.58.5)
% Unrecognized host or address, or protocol not running.
```


Shutdown R5's connections to make sure it can't reach the Loopback0 subnet of R4 and enable DNS debugging on R5.

```
Rack1R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R5(config)#interface Serial 0/0
Rack1R5(config-if)#shutdown
Rack1R5(config-if)#interface Serial 0/1
Rack1R5(config-if)#shutdown

Rack1R5#debug domain
```

Ping any name from SW2 to see that it actually resolves to the IP address of R5's Loopback0 interface.

```
Rack1SW2#ping TEST

Translating "TEST"...domain server (155.1.58.5) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/8 ms

Rack1SW2#ping R4.cisco.com.

Translating "R4.cisco.com."...domain server (155.1.58.5) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Rack1SW2#ping ANYNAME

Translating "ANYNAME"...domain server (155.1.58.5) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
```

Note that queries for the hostname of R5 return the IP address of R5 interface that received the query.


```
Rack1SW2#ping Rack1R5

Translating "Rack1R5"...domain server (155.1.58.5) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.58.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9 ms
```

Observe the debugging output at R5. The router responds with the configured spoofing address to any query with the exception of queries for its own hostname. For its own name the router returns the IP address of the interface that received the query.

```
Rack1R5#
DNS: Incoming UDP query (id#7)
DNS: Type 1 DNS query (id#7) for host 'ANYNAME' from 155.1.58.8(52461)
DNS: No name-servers are accessible
DNS: Spoofing reply to query (id#7)
DNS: Finished processing query (id#7) in 0.000 secs
DNS: Incoming UDP query (id#8)
DNS: Type 1 DNS query (id#8) for host 'Rack1R5' from 155.1.58.8(53293)
DNS: Query for my own hostname: Rack1R5
DNS: Spoofing reply to query (id#8)
DNS: Finished processing query (id#8) in 0.004 secs
```


13.54 IP Event Dampening

- Configure dampening on R6 so that after a reload its connection to BB1 is not advertised into IGP for 30 seconds.
- Additionally configure R6 so that when this connection flaps it does not disappear for more that 60 seconds from the routing table no matter how much penalty it accumulates.

Configuration

```
R6:
interface Serial 0/0
 dampening 30 1000 2000 60 restart 2000
```

Verification

> **Note**
>
> To find the default IP event dampening values configure the **dampening** command with no arguments at the interface level, and then issue the **show interface dampening** command.
>
> Next, to suppress the interface after the router reloads, apply a restart penalty value. With a half-life time of 30 seconds and restart penalty of 2000, it means that 30 seconds after bootup the penalty will have decayed to 1000. If the reuse value is set to 1000 the end result is that the interface is installed in the routing table 30 seconds after bootup is complete.

```
Rack1R6#show dampening interface
1 interface is configured with dampening.
No interface is being suppressed.
Features that are using interface dampening:
  IP Routing

Rack1R6#show interfaces dampening
Serial0/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      5       0   FALSE       0      30    1000    2000      60    4000    2000
```

Verify interface suppression by simulating a flapping link.

```
Rack1R6#debug dampening interface
interface debugging is on
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#int s0/0
Rack1R6(config-if)#shut
Rack1R6(config-if)#
IF-EvD(Serial0/0): IP Routing reports state transition from UP to DOWN
EvD(Serial0/0): charge penalty 1000, new accum. penalty 1000, flap count 6
EvD(Serial0/0): accum. penalty 1000, not suppressed
IF-EvD(Serial0/0): update IP Routing state to DOWN, interface is not
suppressed
%LINK-5-CHANGED: Interface Serial0/0, changed state to administratively
down
Rack1R6(config-if)#no shut
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to DOWN
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to
down
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to DOWN
Rack1R6(config-if)#shut
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to UP
IF-EvD(Serial0/0): update IP Routing state to UP, interface is not suppressed
%LINK-3-UPDOWN: Interface Serial0/0, changed state to up
IF-EvD(Serial0/0): IP Routing reports state transition from UP to UP
IF-EvD(Serial0/0): IP Routing reports state transition from UP to UP
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
IF-EvD(Serial0/0): IP Routing reports state transition from UP to UP
IF-EvD(Serial0/0): IP Routing reports state transition from UP to DOWN
EvD(Serial0/0): accum. penalty decayed to 870 after 6 second(s)
EvD(Serial0/0): charge penalty 1000, new accum. penalty 1870, flap count 7
EvD(Serial0/0): accum. penalty 1870, not suppressed
IF-EvD(Serial0/0): update IP Routing state to DOWN, interface is not suppressed
%LINK-5-CHANGED: Interface Serial0/0, changed state to administratively down
Rack1R6(config-if)#no shut
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to DOWN
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to
down
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to DOWN
IF-EvD(Serial0/0): IP Routing reports state transition from DOWN to UP
IF-EvD(Serial0/0): update IP Routing state to UP, interface is not suppressed
%LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Rack1R6(config-if)#shut
IF-EvD(Serial0/0): IP Routing reports state transition from UP to UP
IF-EvD(Serial0/0): IP Routing reports state transition from UP to UP
IF-EvD(Serial0/0): IP Routing reports state transition from UP to DOWN
EvD(Serial0/0): accum. penalty decayed to 1627 after 6 second(s)
EvD(Serial0/0): charge penalty 1000, new accum. penalty 2627, flap count 8
EvD(Serial0/0): accum. penalty 2627, now suppressed with a reuse intervals of
42
Rack1R6(config-if)#do show dampening interface
1 interface is configured with dampening.
1 interface is being suppressed.
Features that are using interface dampening:
  IP Routing
```

