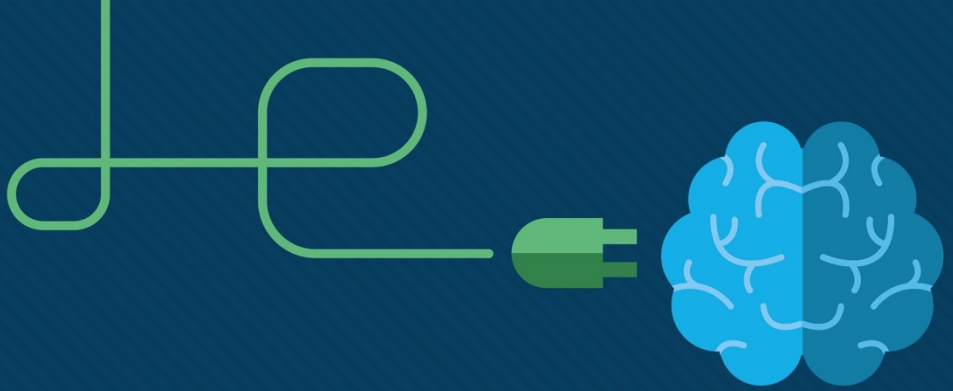


Chapter 15: IP Services
Chapter 16: Overlay Tunnels

Instructor Materials

CCNP Enterprise: Core Networking

Chapter 15 Content

This chapter covers the following content:

Time Synchronization - This section describes the need for synchronizing time in an environment and covers Network Time Protocol and its operations to keep time consistent across devices.

First-Hop Redundancy Protocol - This section gives details on how multiple routers can provide resilient gateway functionality to hosts at the Layer 2/Layer 3 boundaries.

Network Address Translation (NAT) - This section explains how a router can translate IP addresses from one network realm to another.

Time Synchronization

- A device's system time is used to measure periods of idle state or computation. It is important that time is consistent on a system because applications often use the system time to tune internal processes.
- The rate a device can maintain its time can deviate from device to device. Time intervals can vary from one device to another and the times would eventually begin to drift away from each other.

Time Synchronization

It is important that a device's system time is consistent, and from the perspective of managing a network, that the time be synchronized between network devices for the several reasons:

- Managing passwords that change at specific time intervals
- Encryption key exchanges
- Checking validity of certificates based on expiration date and time
- Correlation of security-based events across multiple devices (routers, switches, firewalls, network access control systems, and so on)
- Troubleshooting network devices and correlating events to identify the root cause of an event

- Before NTP, other protocols such as Daytime protocol, Time protocol, and ICMP timestamp provided this service.

Network Time Protocol and Stratums

- Network Time Protocol (NTP) is used to synchronize a set of network clocks in a distributed client/server architecture.
- NTP is a **UDP-based** protocol that connects with servers on port 123. The client source port is dynamic.
- NTP is based on a **hierarchical concept** of communication. At the top of the hierarchy are authoritative devices that operate as an NTP server with an atomic clock. The NTP client queries the NTP server for its time and then updates its time based on the response.
- The NTP synchronization process **is not fast**, gaining an accuracy of tens of milliseconds requires hours or days of comparisons.
- **Stratums** are used to identify the accuracy of the time clock source. NTP servers directly attached to an authoritative time source are stratum 1 servers.
- An NTP client that queries a stratum 1 server is considered a stratum 2 client.
- The higher the stratum, the greater the chance of deviation in time from the authoritative time source due to the number of time drifts between the NTP stratums.

Network Time Protocol Modes

- NTP broadcast client mode — the router can be configured to passively listen for NTP broadcasts, avoiding a static entry to one specific time server.
- NTP static client mode — the router can be configured to listen and exchange messages between statically configured NTP servers.
- NTP master mode — the router can be configured as an NTP server forwarding NTP broadcasts.
- NTP peer associations — the router can be configured to form an NTP peer association with another router. The router can either synchronize to the other system or allow the other system to synchronize to it.

- NTP options and time-related configurations — NTP options include perform authentication and setting the calendar. Time-related options include setting daylight saving time and the current time zone

NTP Configuration

- To configure an NTP client use the following global command (the keywork **prefer** indicates which NTP server to use for time synchronization).
  - **ntp** *ip-address* [**prefer**] [**source** *interface-id*].
- To statically set the stratum for a device when it act as an NTP server use
  - **ntp master** *stratum-number*

Example 15-1 *Simple Multi-Stratum NTP Configuration*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ntp master 1
```

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ntp server 192.168.1.1
```

```
R3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# ntp server 192.168.2.2 source loopback 0
```

```
R4# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)# ntp server 192.168.1.1
```

NTP Status and Associations

The command **`show ntp status`** displays the status of the NTP service. It shows the following:

- Whether the hardware clock is synchronized to the software clock, the stratum reference of the local device, and the reference clock identifier (local or IP address)
- The frequency and precision of the clock
- The NTP uptime and granularity
- The reference time
- The clock offset and delay between the client and the lower-level stratum server
- Root dispersion and peer dispersion
- NTP loopfilter
- Polling interval and time since last update

Example 15-2 *Viewing NTP Status*

```
R1# show ntp status
Clock is synchronized, stratum 1, reference is .LOCL.
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**10
ntp uptime is 2893800 (1/100 of seconds), resolution is 4000
reference time is E0E2D211.E353FA40 (07:48:17.888 EST Wed Jul 24 2019)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 2.24 msec, peer dispersion is 1.20 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 16, last update was 4 sec ago.
```

```
R2# show ntp status
Clock is synchronized, stratum 2, reference is 192.168.1.1
nominal freq is 250.0000 Hz, actual freq is 249.8750 Hz, precision is 2**10
ntp uptime is 2890200 (1/100 of seconds), resolution is 4016
reference time is E0E2CD87.28B45C3E (07:28:55.159 EST Wed Jul 24 2019)
clock offset is 1192351.4980 msec, root delay is 1.00 msec
root dispersion is 1200293.33 msec, peer dispersion is 7938.47 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 1 sec ago.
```

```
R3# show ntp status
Clock is synchronized, stratum 3, reference is 192.168.2.2
nominal freq is 250.0000 Hz, actual freq is 250.0030 Hz, precision is 2**10
ntp uptime is 28974300 (1/100 of seconds), resolution is 4000
reference time is E0E2CED8.E147B080 (07:34:32.880 EST Wed Jul 24 2019)
clock offset is 0.5000 msec, root delay is 2.90 msec
root dispersion is 4384.26 msec, peer dispersion is 3939.33 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000012120 s/s
system poll interval is 64, last update was 36 sec ago.
```

A streamlined version of the NTP server status and delay can be viewed using the command **`show ntp associations`**.

Stratum Preference

An NTP client configured with multiple NTP servers will only use the NTP server with the lowest stratum.

If R2 crashes, preventing R4 from reaching R1, R4 will synchronize with R3 and become a stratum 4 time device. When R2 recovers, R4 will synchronize with R1 and become a stratum 2 device again.

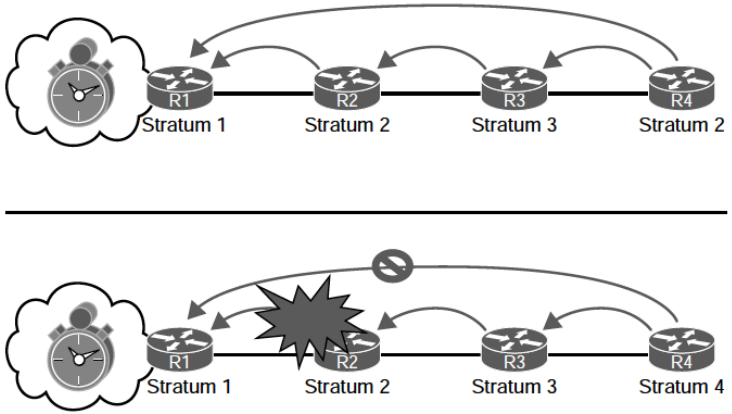


Figure 15-2 *NTP Stratum Preferences*

NTP Peers

An NTP client will change it's time to that of the NTP server. However, an NTP server does not change its time to reflect an NTP client. NTP peers act as clients and servers to each other. They can query and synchronize their time to each other. NTP peers are configured with the command **ntp peer** *ip-address*.

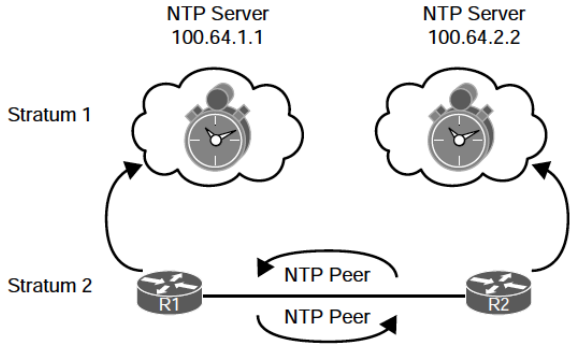


Figure 15-3 *NTP Stratums*

Example 15-4 *NTP Peer Configuration*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ntp peer 192.168.2.2
```

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ntp peer 192.168.1.1
```

First-Hop Redundancy Protocol

- Network resiliency is a key component of network design.
- Network resiliency can be accomplished by adding redundant devices such as Layer 2 switches or Layer 3 routers into a topology.

Network Resiliency/First Hop Redundancy Protocols

The figure shows the concept of adding resiliency to the network. In both scenarios:

- Two devices (172.16.1.2 and 172.16.1.3) can be the PC's gateway.
- There are two resilient Layer 2 links that connect SW6 to a switch that can connect the PC to either gateway.

First-hop redundancy protocols (FHRPs) solve the problem of end devices configuring multiple gateways. They do this by creating a virtual IP (VIP) gateway that is shared between the Layer 3 devices. The following are FHRPs:

- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP)
- Gateway Load Balancing Protocol (GLBP)

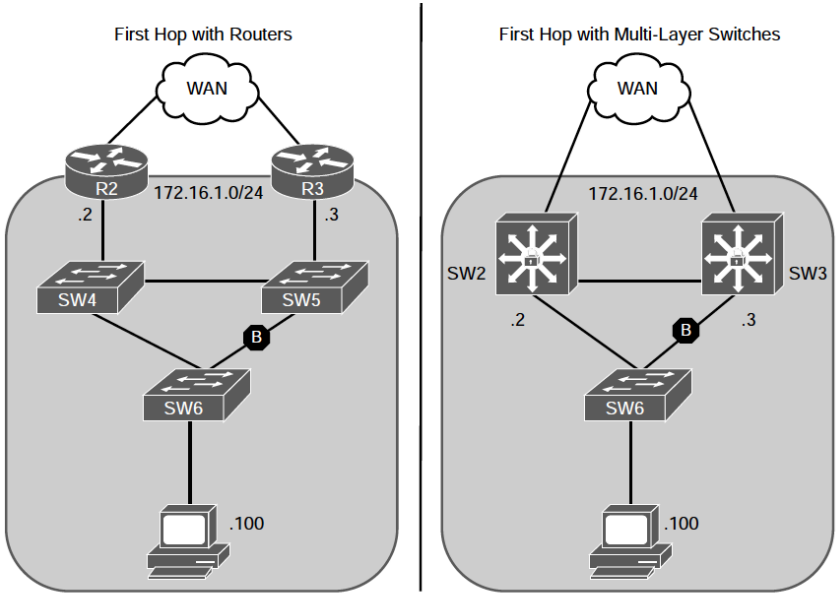


Figure 15-4 *Resiliency with Redundancy with Layer 2 and Layer 3 Devices*

Object Tracking (here routes in the routing table)

Object tracking offers a flexible and customizable mechanism for linking with FHRPs and other routing components.

Users **can track specific objects** in the network and take necessary action when any object's state change affects the network traffic.

To track **routes in the routing table** use the command

track *object-number* **ip route** *route/prefix-length* **reachability**.

The status of object tracking can be viewed with the command

show track [*object-number*].

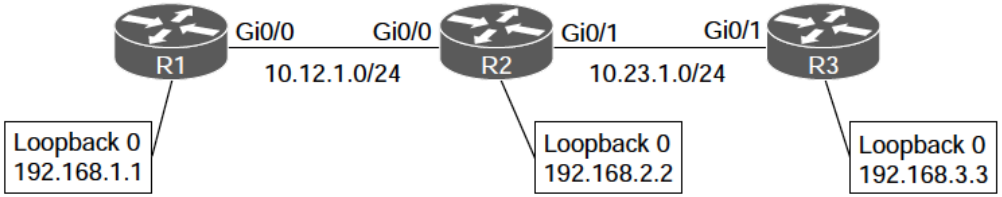


Figure 15-5 *Object Tracking*

Example 15-5 *Tracking R3's Loopback Interface*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# track 1 ip route 192.168.3.3/32 reachability

R1# show track
Track 1
  IP route 192.168.3.3 255.255.255.255 reachability
  Reachability is Up (EIGRP)
    1 change, last change 00:00:32
  First-hop interface is GigabitEthernGi0/0
```

Tracking an Interface

To track an interface's line protocol state use the command

track *object-number* **interface** *interface-id* **line-protocol**.

The example shows R2 being configured for tracking the Gi0/1 interface toward R3.

Shutting down R2's Gi0/1 interface changed the tracked object state on R1 and R2 to a down state.

Object tracking works with protocols such as

- Hot Standby Router Protocol (HSRP),
- Virtual Router Redundancy Protocol (VRRP),
- Gateway Load Balancing Protocol (GLBP).

They take action when the state of an object changes.

Example 15-6 *Tracking R2's Gi0/1 Interface Line Protocol State*

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# track 2 interface GigabitEthernGi0/1 line-protocol
```

```
R2# show track
Track 2
  Interface GigabitEthernGi0/1 line-protocol
  Line protocol is Up
    1 change, last change 00:00:37
```

Example 15-7 *Demonstrating a Change of Tracked State*

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# interface GigabitEthernGi0/1
R2(config-if)# shutdown
*03:04:18.975: %TRACK-6-STATE: 2 interface Gi0/1 line-protocol Up -> Down
*03:04:18.980: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.23.1.3
(GigabitEthernGi0/1) is * 03:04:20.976: %LINK-5-CHANGED: Interface
GigabitEthernGi0/1, changed state to administratively down
* 03:04:21.980: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernGi0/1,
changed state to down
```

```
R1#
03:04:24.007: %TRACK-6-STATE: 1 ip route 192.168.3.3/32 reachability Up -> Down
```

Hot Standby Router Protocol

Hot Standby Routing Protocol (HSRP) is a Cisco **proprietary** protocol. It provides routing redundancy for hosts configured with a default gateway IP address.

- A minimum of two devices are required to enable HSRP:
  - **One** device acts as the **active** device and takes care of forwarding the packets.
  - The **other** acts as a **standby** that is ready to take over the role of active device in the event of a failure.
- A **virtual IP address** is configured on each HSRP-enabled **interface** that belongs to the **same HSRP group**. A **virtual MAC address** is **also** assigned for the **group**.
- The **active router receives and routes the packets destined for the virtual MAC address** of the group.
- HSRP-enabled interfaces send and receive multicast UDP-based **hello messages** to **detect any failure** and designate active and standby routers.
- When the HSRP active router fails, the HSRP standby router assumes control of the virtual IP address and virtual MAC address of the group.

HSRP Elections & Versions

- A **HSRP election** selects the router with the **highest priority** (default is 100).
- In the event of a tie in priority, the router with the **highest IP address** for the network segment is preferred.
- HSRP does **not** support **preemption by default**. If a router with a lower priority becomes active, it stays active regardless if the superior router comes back online.
- The transition of the HSRP active to the standby is transparent to all hosts on the segment because the MAC address moves with the virtual IP address.
- HSRP has two versions, HSRPv1 and HSRPv2.

Table 15-2 HSRP Versions

	HSRPv1	HSRPv2
Timers	Does not support millisecond timer values	Supports millisecond timer values
Group range	0 to 255	0 to 4095
Multicast address	224.0.0.2	224.0.0.102
MAC address range	0000.0C07.AC*xy*, where *xy* is a hex value representing the HSRP group number	0000.0C9F.F000 to 0000.0C9F.FFFF

Configuring HSRP Virtual IP Address

The following steps show how to configure an HSRP virtual IP (VIP) gateway instance:

Step 1. Define the HSRP instance by using the command **standby** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Configure HSRP router preemption with the command **standby** *instance-id* **preempt**.

Step 3. (Optional) Configure the HSRP priority by using the command **standby** *instance-id* **priority** *priority*. The priority is a value between 0 and 255.

Step 4. (Optional) Configure the HSRP MAC address with the command **standby** *instance-id* **mac-address** *mac-address*.

Step 5. (Optional) Define the HSRP timers by using the command **standby** *instance-id* **timers** {*seconds* | **msec** *milliseconds*}. HSRP can poll in intervals of 1 to 254 seconds or 15 to 999 milliseconds

Step 6. (Optional) Establish HSRP authentication by using the command **standby** *instance-id* **authentication** {*text-password* | **text** *text-password* | **md5** {**key-chain** *key-chain* | **key-string** *key-string*}}.

HSRP Configuration and State

Example 15-9 *Simple HSRP Configuration*

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface vlan 10
03:55:35.148: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state
to down
SW2(config-if)# ip address 172.16.10.2 255.255.255.0
SW2(config-if)# standby 10 ip 172.16.10.1
03:56:00.097: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
SW2(config-if)# standby 10 preempt
```

```
SW3(config)# interface vlan 10
03:56:04.478: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state
to down
SW3(config-if)# ip address 172.16.10.3 255.255.255.0
SW3(config-if)# standby 10 ip 172.16.10.1
SW1(config-if)# standby 10 preempt
03:58:22.113: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
```

Example 15-10 *Viewing the Summarized HSRP State*

```
SW2# show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State   Active          Standby         Virtual IP
V110        10   100 P Standby 172.16.10.3     local           172.16.10.1
```

```
SW3# show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State   Active          Standby         Virtual IP
V110        10   100 P Active  local           172.16.10.2     172.16.10.1
```

HSRP Tracked Objects

HSRP provides the capability to link object tracking to priority.

Example 15-12 shows the configuration of SW2 where a tracked object is created against VLAN 1's interface line protocol, increasing the HSRP priority to 110, and linking HSRP to the tracked object so that the **priority decrements by 20 if interface VLAN 1 goes down**.

Example 15-13 shows that the HSRP group on VLAN 10 on SW2 correlates the status of the tracked object for the VLAN 1 interface.

Example 15-12 *Correlating HSRP to Tracked Objects*

```
SW2(config)# track 1 interface vlan 1 line-protocol
SW2(config-track)# interface vlan 10
SW2(config-if)# standby 10 priority 110
04:44:16.973: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
SW2(config-if)# standby 10 track 1 decrement 20
```

Example 15-13 *Verifying the Linkage of HSRP to Tracked Objects*

```
SW2# show standby
! Output omitted for brevity
Vlan10 - Group 10
  State is Active
    10 state changes, last state change 00:06:12
  Virtual IP address is 172.16.10.1
..
  Preemption enabled
  Active router is local
  Standby router is 172.16.10.3, priority 100 (expires in 9.856 sec)
  Priority 110 (configured 110)
    Track object 1 state Up decrement 20
```

Verifying HSRP State With Tracked Objects

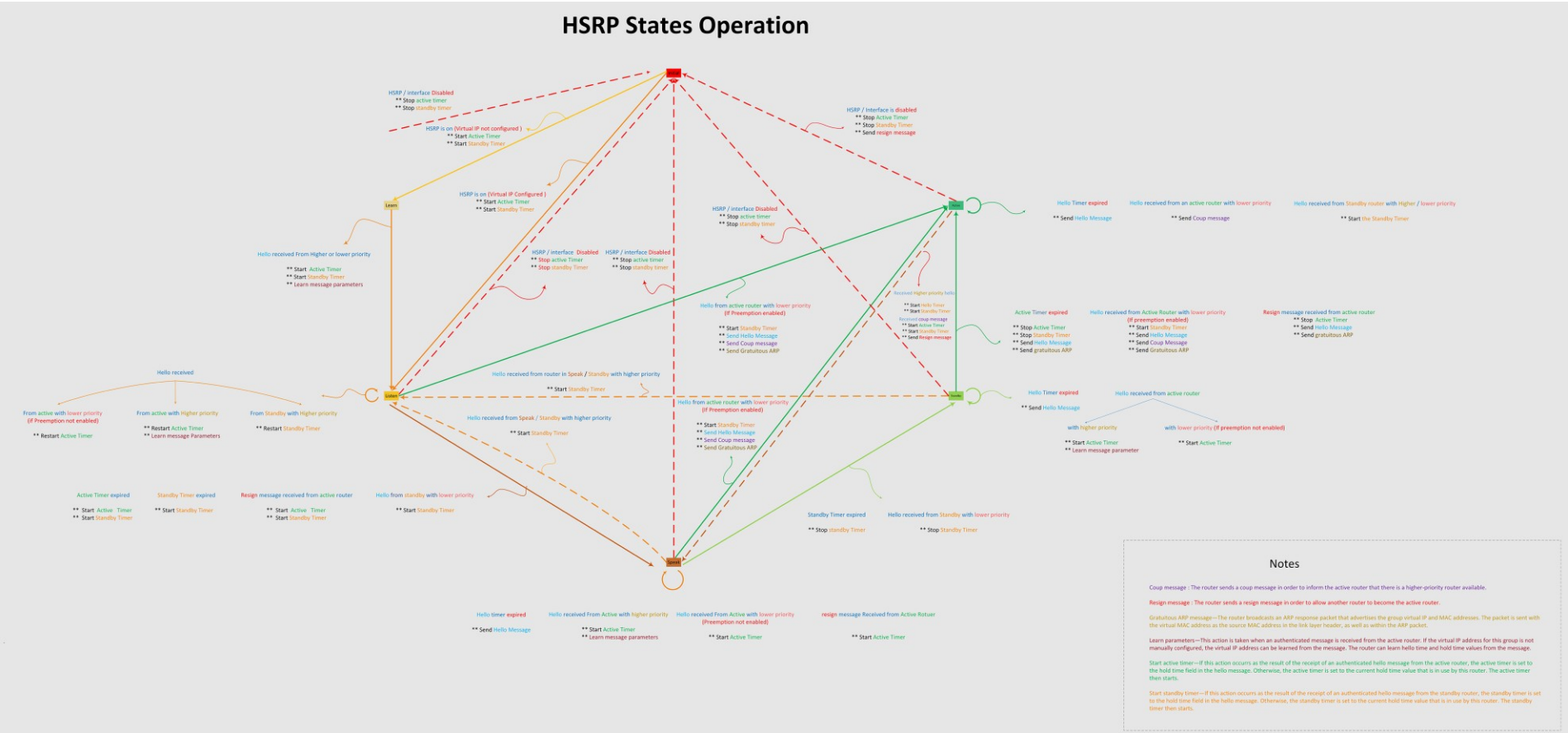
Example 15-14 verifies the anticipated behavior by shutting down the VLAN 1 interface on SW2. The syslog messages indicate that the object track state changed immediately after the interface was shut down, and shortly thereafter, the HSRP role changed to a standby state.

Example 15-14 *Verifying the Change of HSRP State with Object Tracking*

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface vlan 1
SW2(config-if)# shut
 04:53:16.490: %TRACK-6-STATE: 1 interface Vl1 line-protocol Up -> Down
 04:53:17.077: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Active -> Speak
 04:53:18.486: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively
down
 04:53:19.488: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state
to down
 04:53:28.267: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
```

```
SW2# show standby
! Output omitted for brevity
Vlan10 - Group 10
  State is Standby
    12 state changes, last state change 00:00:39
..
  Active router is 172.16.10.3, priority 100 (expires in 9.488 sec)
  Standby router is local
  Priority 90 (configured 110)
    Track object 1 state Down decrement 20
  Group name is "hsrp-Vl10-10" (default)
```

Pouze pro názornost ☺



Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) is an industry standard protocol that operates similarly to HSRP. However, the differences are as follows:

- The preferred active router controlling the VIP gateway is called the **master** router. All other VRRP routers are known as **backup** routers.
- VRRP enables **preemption by default**.
- The **MAC** address of the VIP gateway uses the structure **0000.5e00.01xx**, where xx reflects the group ID in hex.
- **VRRP** uses the multicast address **224.0.0.18** for communication.

There are currently two versions of VRRP:

- VRRPv2: Supports IPv4
- VRRPv3: Supports IPv4 and IPv6

Legacy VRRP Configuration

Early VRRP configurations supported only VRRPv2 and was non-hierarchical in its configuration. The following are steps used to configure older software versions with VRRP:

Step 1. Define the VRRP instance by using the comman **vrrp** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Define the VRRP priority by using the command **vrrp** *instance-id* **priority** *priority* The priority is a value between 0 and 255.

Step 3. (Optional) Enable object tracking so that the priority is decremented when the object is false by using the command **vrrp** *instance-id* **track** *object-id* **decrement** *decrement-value*.

Step 4. (Optional) Establish VRRP authentication by using the command `vrrp instance-id authentication {text-password | text text-password | md5 {key-chain key-chain | key-string key-string}}`

Example 15-15 *Legacy VRRP Configuration*

```
R2# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# interface GigabitEthernet 0/0
R2(config-if)# ip address 172.16.20.2 255.255.2
R2(config-if)# vrrp 20 ip 172.16.20.1
 04:32:14.109: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Backup
 04:32:14.113: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Backup
 04:32:17.728: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Backup -> Master
 04:32:47.170: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Master -> Backup
```

```
R3# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# interface GigabitEthernGi0/0
R3(config-if)# ip add 172.16.20.3 255.255.255.0
 04:32:43.550: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Backup
 04:32:43.554: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Backup
 04:32:47.170: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Backup -> Master
```

VRRP State

The command **`show vrrp [brief]`** provides an update on the VRRP group, along with other relevant information for troubleshooting. Example 15-16 shows the brief iteration of the command and 15-17 shows the detailed state of VRRP.

Example 15-16 *Viewing the Summarized VRRP State*

```
R2# show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr     Group addr
Gi0/0              20  100 3609       Y  Backup  172.16.20.3     172.16.20.1
```

```
R3# show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr     Group addr
Gi0/0              20  100 3609       Y  Master  172.16.20.3     172.16.20.1
```

Example 15-17 *Viewing the Detailed VRRP State*

```
R2# show vrrp
EthernGi0/0 - Group 20
  State is Backup
  Virtual IP address is 172.16.20.1
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 172.16.20.3, priority is 100
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec (expires in 2.904 sec)
```

Hierarchical VRRP Configuration

The newer version of IOS XE software provides configuration of VRRP in a multi-address format that is hierarchical. The following are steps to configure hierarchical VRRP:

Step 1. Enable VRRPv3 on the router by using the command **fhrp version vrrp v3**.

Step 2. Define the VRRP instance by using the command **vrrp** *instance-id* **address-family** {**ipv4** | **ipv6**}.

Step 3. (Optional) Change VRRP to Version 2 by using the command **vrrpv2**. VRRPv2 and VRRPv3 are not compatible.

Step 4. Define the gateway VIP by using the command **address** *ip-address*.

Step 5. (Optional) Define the VRRP priority by using the command **priority** *priority*.

Step 6. (Optional) Enable object tracking so that the priority is decremented when the object is false using the command **track** *object-id* **decrement** *decrement-value*.

Example 15-18 *Configuring Hierarchical VRRP Configuration*

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# fhrp version vrrp v3
SW2(config)# interface vlan 22
 19:45:37.385: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan22, changed
 state to up
SW2(config-if)# ip address 172.16.22.2 255.255.255.0
SW2(config-if)# vrrp 22 address-family ipv4
SW2(config-if-vrrp)# address 172.16.22.1
SW2(config-if-vrrp)# track 1 decrement 20
SW2(config-if-vrrp)# priority 110
SW2(config-if-vrrp)# track 1 decrement 20
 19:48:00.338: %VRRP-6-STATE: Vlan22 IPv4 group 22 state INIT -> BACKUP
 19:48:03.948: %VRRP-6-STATE: Vlan22 IPv4 group 22 state BACKUP -> MASTER
```

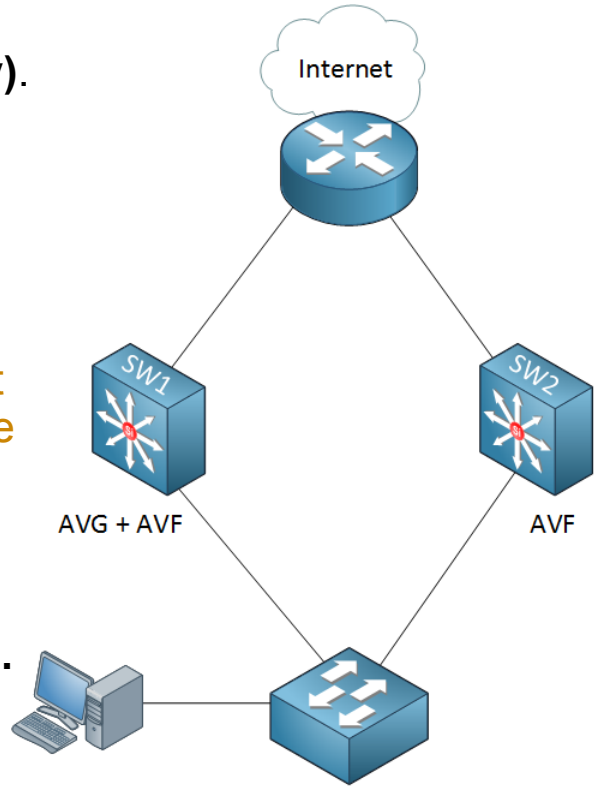
```
SW3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW3(config)# fhrp version vrrp v3
SW3(config)# interface vlan 22
 19:46:13.798: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan22, changed state
to up
SW3(config-if)# ip address 172.16.22.3 255.255.255.0
SW3(config-if)# vrrp 22 address-family ipv4
SW3(config-if-vrrp)# address 172.16.22.1
 19:48:08.415: %VRRP-6-STATE: Vlan22 IPv4 group 22 state INIT -> BACKUP
```

The status of the VRRP routers can be viewed with the command **show vrrp** [**brief**]. The output is identical to that of the legacy VRRP configuration.

Global Load Balancing Protocol

All devices running GLBP elect an **AVG (Active Virtual Gateway)**. There will be only one AVG for a single group running GLBP but other devices can take over this rule if the AVG fails. The role of the AVG is to assign a virtual MAC address to all other devices running GLBP.

All devices will become an **AVF (Active Virtual Forwarder)** including the AVG. Whenever a computer sends an ARP Request the AVG will respond with one of the virtual MAC addresses of the available AVFs. Because of this mechanism all devices running GLBP will be used to forward IP packets.



GLBP supports **four active AVFs** and **one AVG per GLBP group.**

- If an AVG fails, its role is transferred to a standby AVG device.
- If an AVF fails, another router takes over the forwarding responsibilities for that AVF, which includes the virtual MAC address for that instance.

GLBP Load Balancing

GLBP supports three methods of load balancing traffic:

- **Round-robin** – the AVG will hand out the virtual MAC address of AVF1, then AVF2, AVF3 and gets back to AVF1 etc.
- **Host-dependent** – A host will be able to use the same virtual MAC address of an AVF as long as it is reachable.
- **Weighted** – If you want some AVFs to forward more traffic than others you can assign them a different weight.

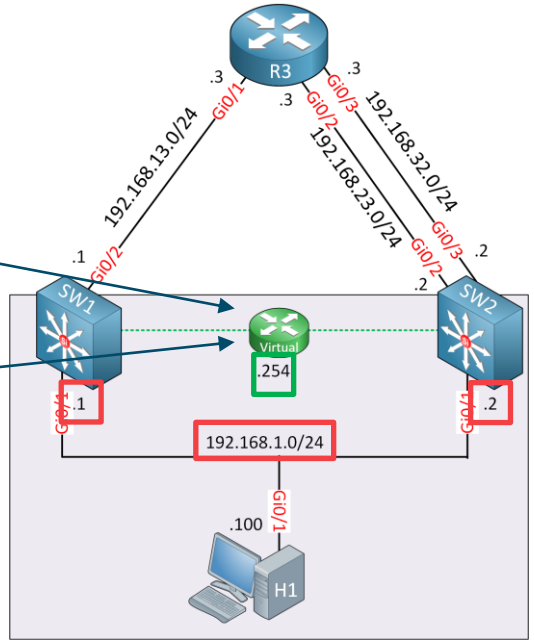
The load-balancing method can be changed with the command **glbp** *instance-id* **load-balancing** {**host-dependent** | **round-robin** | **weighted**}. The weighted load-balancing method has the AVG direct traffic to the AVFs based on the percentage of weight a router has over the total weight of all GLBP routers. The weight can be set for a router with the command **glbp** *instance-id* **weighting** *weight*.

First-Hop Redundancy Protocol

GLBP Configuration

```
SW1(config)#interface Vlan1
SW1(config-if)#glbp 1 ip 192.168.1.254
SW1(config-if)#glbp 1 priority 150

SW2(config)#interface Vlan1
SW2(config-if)#glbp 1 ip 192.168.1.254
```



SW1#show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Vl1	1	-	150	Active	192.168.1.254	local	192.168.1.2
Vl1	1	1	-	Active	0007.b400.0101	local	-
Vl1	1	2	-	Listen	0007.b400.0102	192.168.1.2	-

SW2#show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Vl1	1	-	100	Standby	192.168.1.254	192.168.1.1	local
Vl1	1	1	-	Listen	0007.b400.0101	192.168.1.1	-
Vl1	1	2	-	Active	0007.b400.0102	local	

GLBP Configuration

The following steps detail how to configure a GLBP:

Step 1. Define the GLBP instance by using the command **glbp** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Configure GLBP preemption with the command **glbp** *instance-id* **preempt**.

Step 3. (Optional) Define the GLBP priority by using the command **glbp** *instance-id* **priority** *priority*. The priority is a value between 0 and 255.

Step 4. (Optional) Define the GLBP timers by using the command **glbp** *instance-id* **timers** {*hello-seconds* | **msec** *hello-milliseconds*} {*hold-seconds* | **msec** *hold-milliseconds*}.

Step 5. (Optional) Establish GLBP authentication by using the command **glbp** instance-id **authentication** {**text** *text-password* | **md5** {**key-chain** *key-chain* | **key-string** *key-string*}}.

Network Address Translation

- In the early stages of the internet, large network blocks were assigned to organizations.
- Network engineers started to realize that as more people connected to the internet, the IP address space would become exhausted.

Private Network Addressing

RFC 1918 established common network blocks that are non-globally routed networks. These address blocks provide large private network blocks for companies to connect their devices together, but private IP addressing doesn't exist on the internet. The private address blocks are as follows:

10.0.0.0/8 accommodates 16,777,216 hosts.

172.16.0.0/24 accommodates 1,048,576 hosts.

192.168.0.0/16 accommodates 65,536 hosts.

NAT enables the internal IP network to appear as a publicly routed external network. A NAT device (typically a router or firewall) modifies the source or destination IP addresses in a packet's header as the packet is received on the outside or inside interface. NAT can be used in use cases other than just providing internet connectivity to private networks such as providing connectivity when a company buys another company, and the two companies have overlapping networks.

Network Address Translation

NAT enables the internal IP network to appear as a publicly routed external network.

A NAT device (typically a router or firewall) modifies the source or destination IP addresses in a packet's header as the packet is received on the outside or inside interface.

NAT can be used in use cases other than just providing internet connectivity to private networks, such as providing connectivity when a company buys another company, and the two companies have overlapping networks.

Most routers and switches perform NAT translation only with the IP header addressing and do not translate IP addresses within the payload (for example, DNS requests). Some firewalls can perform NAT within the payload for certain types of traffic.

Inside/Outside Local and Global

This is about source

- **Inside local** - The actual private IP address assigned to a device on the inside network(s).
- **Inside global** - The public IP address that represents one or more inside local IP addresses to the outside. **Or, inside local address as it is seen from outside.**

- **Outside local** - The IP address of an outside host as it appears to the inside network. The IP address does not have to be reachable by the outside but is considered private and must be reachable by the inside network. **Or, outside global address as it is seen from inside.**
- **Outside global** - The **public IP address** assigned to a host on the outside network. This IP address must be reachable by the outside network. „Normal“ address of host in the Internet.

This is about destination

Types of NAT

Three (according Cisco ☺) types of NAT commonly used today are as follows:

- Static NAT - Provides a static one-to-one mapping of a local IP address to a global IP address.
  - Inside static
  - Outside static
- Pooled NAT - Provides a dynamic one-to-one mapping of a local IP address to a global IP address. The global IP address is temporarily assigned to a local IP address. After a certain amount of idle NAT time, the global IP address is returned to the pool.
- Port Address Translation (PAT) - Provides a dynamic many-to-one mapping of many local IP addresses to one global IP address. The NAT device translates the private IP address and port to a different global IP address and port. The port is unique from any other ports, which enables the NAT device to track the global IP address to local IP addresses based on the unique port mapping.

NAT Example

Figure 15-7 is used throughout this section to illustrate NAT.

R5 performs the translation; its Gi0/0 interface (10.45.1.5) is the outside interface, and its Gi0/1 (10.56.1.5) interface is the inside interface. The other devices act as either clients or servers to demonstrate how NAT functions.

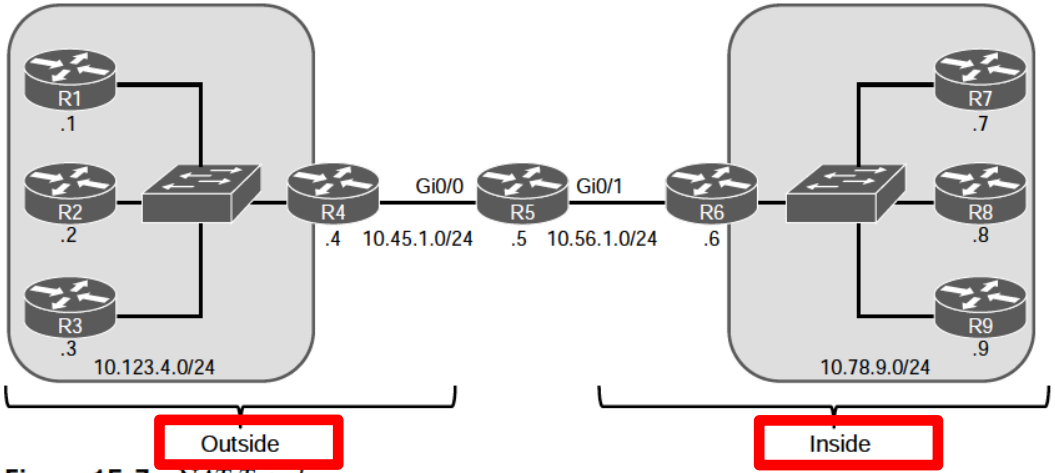


Figure 15-7 *NAT Topology*

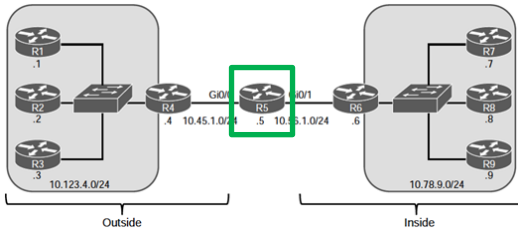
Static NAT

Static NAT involves the translation of a global IP address to a local IP address, based on a static mapping of the global IP address to the local IP address.

There are two types of static NAT:

- **Inside static NAT** - involves the mapping of
  an inside local (private) IP address to an inside global (public) IP address.
- **Outside static NAT** - involves the mapping of
  an outside global (public) IP address to an outside local (private) IP address.

Inside Static NAT



The steps for configuring inside static NAT are as follows:

Step 1. Configure the outside interfaces by using the command **`ip nat outside`**.

Step 2. Configure the inside interface with the command **`ip nat inside`**.

Step 3. Configure the inside static NAT by using the command **`ip nat inside source static`** *`inside-local-ip inside-global-ip`*.

Example 15-28 *Configuring Inside Static NAT*

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat inside source static 10.78.9.7 10.45.1.7
```

Inside global

Inside local

Identifying the Source with Inside Static NAT/NAT Translation Table

With NAT configured, a telnet session with R1 is initiated. Viewing the TCP session on R1, the local address remains 10.123.4.1 but the remote address now reflects 10.45.1.7.

Example 15-29 *Identification of the Source with Inside Static NAT*

```
R7# telnet 10.123.4.1
Trying 10.123.4.1 ... Open
************************************************************
* You have remotely connected to R1 on line 3
************************************************************
User Access Verification
Password:

R1# show tcp brief
TCB        Local Address               Foreign Address             (state)
F6D25D08   10.123.4.1.23               10.45.1.7.56708             ESTAB
```

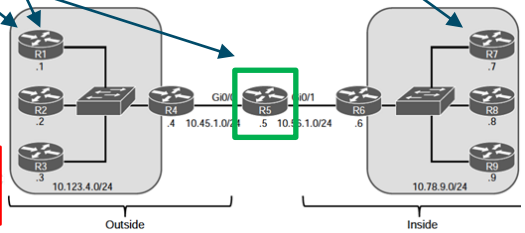
This „public“ address

The NAT translation table consists of static and dynamic entries. The NAT translation table is displayed with the command **show ip nat translations**.

- The first entry is the dynamic entry correlating to the Telnet session.
- The second entry is the inside static NAT entry that was configured.

Example 15-30 *NAT Translation Table for Inside Static NAT*

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 10.45.1.7:56708    10.78.9.7:56708    10.123.4.1:23      10.123.4.1:23
--- 10.45.1.7          10.78.9.7          ---                ---
```



10.123.4.1 ← 10.45.1.7 ← 10.78.9.7

NAT Translation Steps

The NAT translation follows these steps:

Step 1. As traffic enters the Gi0/1 interface on R5, R5 performs a **route lookup** for the destination IP address, which points out of its Gi0/0 interface. R1 is aware that the **Gi0/0** interface is an **outside NAT** interface and that the **Gi0/1** interface is an **inside NAT** interface and therefore checks the NAT table for an entry.

Step 2. Only the inside static NAT entry exists, so R5 **creates** a **dynamic inside NAT entry** with the packet's destination 10.123.4.1 for the **outside local and outside global address.**

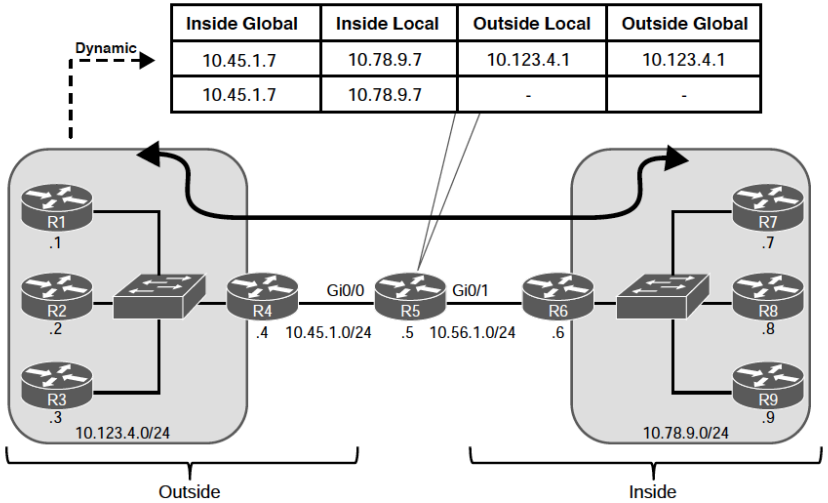


Figure 15-8 *Inside Static NAT Topology for R7 as 10.45.1.7*

10.123.4.1 ← 10.45.1.7 ← 10.78.9.7

NAT Translation Steps (Cont.)

Step 3. R5 translates (that is, changes) the packet's **source IP address** from 10.78.9.7 to 10.45.1.7.

Step 4. R1 **registers** the session as coming from 10.45.1.7 and then transmits **a return packet**. The packet is forwarded to R4 using the **static default route**, and R4 forwards the packet using the static default route.

Step 5. As the packet enters on the Gi0/0 interface of R5, R5 is aware that the Gi0/0 interface is an outside NAT interface and checks the NAT table for an entry.

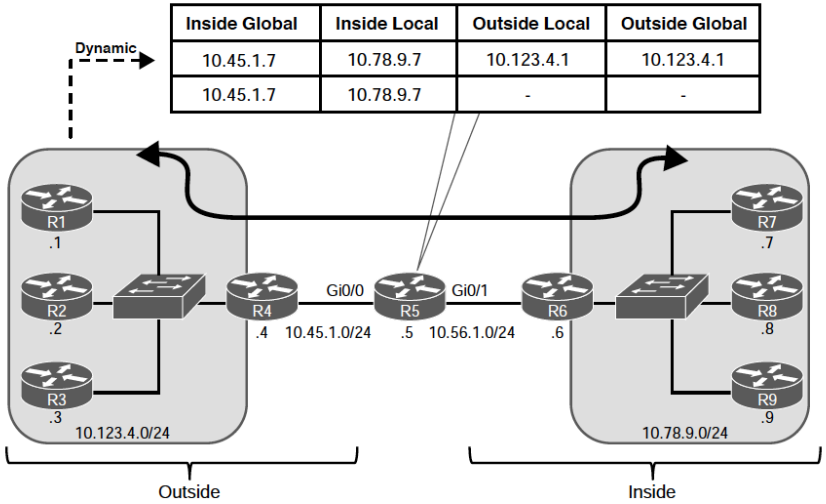


Figure 15-8 *Inside Static NAT Topology for R7 as 10.45.1.7*

10.123.4.1 ← 10.45.1.7 ← 10.78.9.7

NAT Translation Steps (Cont.)

Step 6. R5 correlates the packet's source and destination ports with the first NAT entry, as shown in Example 15-30, and knows to **modify** the packet's destination IP address from 10.45.1.7 to 10.78.9.7.

Step 7. R5 routes the packet out the Gi0/1 interface toward R6.

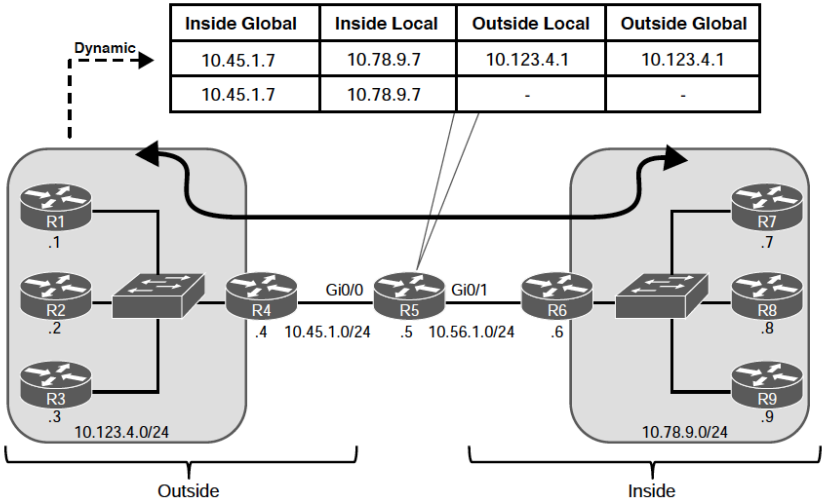


Figure 15-8 *Inside Static NAT Topology for R7 as 10.45.1.7*

10.123.4.1 ← 10.45.1.7 ← 10.78.9.7

Connectivity from External Devices to the Inside Global IP Address

In Example 15-31:

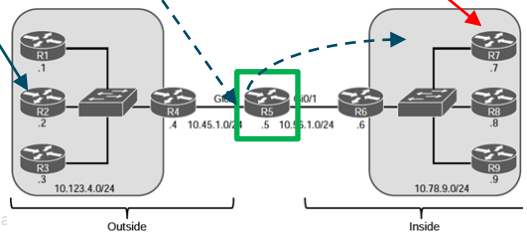
- R2 establishes a Telnet session with R7, using the inside global IP address 10.45.1.7.
- R5 simply creates a second dynamic entry for this new session.
- From R7's perspective, it has connected with R2 (10.123.4.2).

Example 15-31 *Connectivity from External Devices to the Inside Global IP Address*

```
R2# telnet 10.45.1.7
Trying 10.45.1.7 ... Open
************************************************************
* You have remotely connected to R7 on line 2
************************************************************
User Access Verification
Password:

R7# show tcp brief
TCB        Local Address               Foreign Address             (state)
F6561AE0   10.78.9.7.23                10.123.4.2.63149            ESTAB
F65613E0   10.78.9.7.23579             10.123.4.1.23               ESTAB

R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 10.45.1.7:56708    10.78.9.7:56708    10.123.4.1:23      10.123.4.1:23
tcp 10.45.1.7:23       10.78.9.7:23       10.123.4.2:63149   10.123.4.2:63149
--- 10.45.1.7          10.78.9.7          ---                ---
```



10.123.4.2 → 10.45.1.7 → 10.78.9.7

Outside Static NAT

Outside static NAT involves the mapping of an outside global (public) IP address to an outside local (private) IP address. The steps for configuring outside static NAT are as follows:

Step 1. Configure the outside interfaces by using the command **`ip nat outside`**.

Step 2. Configure the inside interface with the command **`ip nat inside`**.

Step 3. Configure the outside static NAT by using the command **`ip nat outside source static`** *`outside-global-ip outside-local-ip [add-route]`*.

Example 15-32 *Configuring Outside Static NAT*

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat outside source static 10.123.4.2 10.123.4.222
```

outside local

outside global

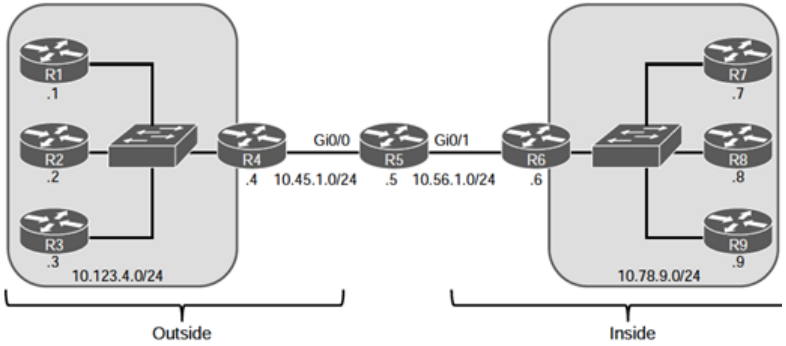
Outside Static NAT Demonstration

R6, R7, R8, or R9 could initiate a **Telnet** session **with R2's IP address 10.123.4.2** and **no NAT translation** would occur.

The same routers could initiate a Telnet session **with R2's outside local IP address 10.123.4.222**; or R2 could initiate a session with any of the inside hosts (R6, R7, R8, or R9) to demonstrate the **outside static NAT** entry.

Example 15-33 shows R2 establishing a Telnet session with R9 10.78.9.9.

- From R9's perspective, the connection came from 10.123.4.222.
- At the same time, R8 initiated a Telnet session with the outside static NAT outside local IP address 10.123.4.222.
- From R2's perspective, the source address is R8's 10.78.9.8 IP address.



Example 15-33 *Generating Network Traffic with Outside Static NAT*

```
R2# telnet 10.78.9.9
Trying 10.78.9.9 ... Open
************************************************************
* You have remotely connected to R9 on line 2
************************************************************
User Access Verification
Password:

R9#show tcp brief
TCB        Local Address               Foreign Address             (state)
F6A23AF0   10.78.9.9.23                10.123.4.222.57126          ESTAB
```

```
R8# telnet 10.123.4.222
Trying 10.123.4.222 ... Open
************************************************************
* You have remotely connected to R2 on line 2
************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB        Local Address               Foreign Address             (state)
F64C9460   10.123.4.2.57126            10.78.9.9.23                ESTAB
F64C9B60   10.123.4.2.23               10.78.9.8.11339             ESTAB
```

NAT Translation Table for Outside Static NAT

Figure 15-9 shows the translation table of R5 for the outside static NAT entry of R2 for 10.123.4.222.

Example 15-34 shows the NAT translation table of R5.

There are three entries:

- The **first** entry is the **outside static** NAT entry that was configured.
- The **second** entry is the Telnet session launched **from R8 to the 10.123.4.222** IP address.
- The **third** entry is the Telnet session launched **from R2 to R9's IP address 10.78.9.9**.

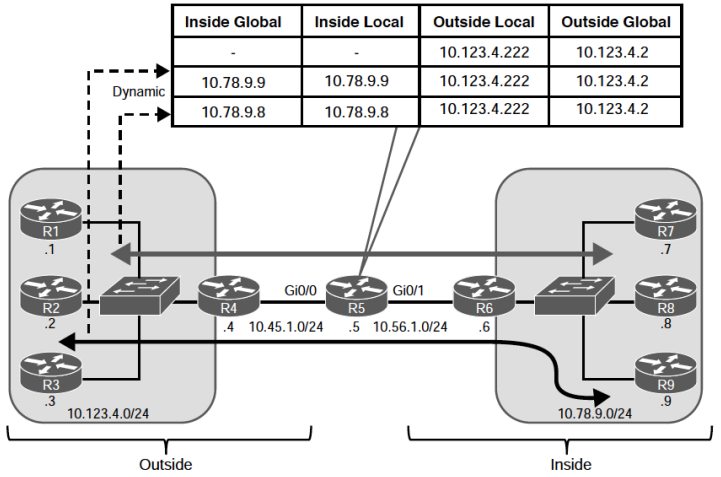


Figure 15-9 *Outside Static NAT Topology for R2 as 10.123.4.222*

Example 15-34 *NAT Translation Table for Outside Static NAT*

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- ---                ---                10.123.4.222       10.123.4.2
tcp 10.78.9.8:11339    10.78.9.8:11339    10.123.4.222:23    10.123.4.2:23
tcp 10.78.9.9:23       10.78.9.9:23       10.123.4.222:57126 10.123.4.2:57126
```

Pooled NAT

Remark to static NAT: there is the number of configurations entries that must be created on the NAT device. In addition, the number of global IP addresses must match the number of local IP addresses.

- Pooled NAT provides a more dynamic method of providing a one-to-one IP address mapping—but on a dynamic, as-needed basis.

- The **dynamic NAT translation stays** in the translation table until the **the timeout period** expired
  - Since traffic flow from the local address to the global address has stopped.
  - 24 hours by default.
  - Unused global IP address is then returned to the pool to be used again.

- Pooled NAT can operate as
  - inside NAT
  - outside NAT.

Pooled NAT Configuration Steps

The steps for configuring inside pooled NAT are as follows:

Step 1. Configure the outside interfaces by using the command **`ip nat outside`**.

Step 2. Configure the inside interface with the command **`ip nat inside`**.

Step 3. Specify which traffic to translate by using a standard or extended ACL referenced by number or name. Using a user-friendly name may be simplest from an operational support perspective

Step 4. Define the global **pool** of IP addresses by using the command **`ip nat pool`** *`nat-pool-name starting-ip ending-ip`* **`prefix-length`** *`prefix-length`*.

Step 5. Configure the inside pooled NAT by using the command **`ip nat inside source list`** *`acl`* **`pool`** *`nat-pool-name`***.**

Configuring Inside Pooled NAT

Example 15-35 uses a NAT pool with the IP addresses 10.45.1.10 and 10.45.1.11. A named ACL, ACL-NAT-CAPABLE, allows only packets sourced from the 10.78.9.0/24 network to be eligible for pooled NAT.

In Example 15-35, R7 and R8 ping R1 in order to generate traffic and build the dynamic inside NAT translations.

Example 15-35 *Configuring Inside Pooled NAT*

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# ip access-list standard ACL-NAT-CAPABLE
R5(config-std-nacl)# permit 10.78.9.0 0.0.0.255
R5(config-std-nacl)# exit
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat pool R5-OUTSIDE-POOL 10.45.1.10 10.45.1.11 prefix-length 24
R5(config)# ip nat inside source list ACL-NAT-CAPABLE pool R5-OUTSIDE-POOL
```

Example 15-36 *Initial Traffic for Pooled NAT*

```
R7# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R8# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Pooled NAT Table

In Example 15-37, there are a total of four translations in the translation table of R5. Two of them are for the full flow and specify the protocol, inside global, inside local, outside local, and outside global IP addresses.

In Example 15-38, R8 establishes a Telnet session with R2. R2 detects that the remote IP address of the session is 10.45.1.11. A second method of confirmation is to examine the NAT translation on R5, where there is a second dynamic translation entry for the full Telnet session.

Example 15-37 *Viewing the Pooled NAT Table for R5*

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 10.45.1.10:0      10.78.9.7:0        10.123.4.1:0       10.123.4.1:0
--- 10.45.1.10         10.78.9.7          ---                ---
icmp 10.45.1.11:0      10.78.9.8:0        10.123.4.1:0       10.123.4.1:0
--- 10.45.1.11         10.78.9.8          ---                ---
```

Example 15-38 *Using the Dynamic One-to-One Mappings for Address Consistency*

```
R8# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
************************************************************
* You have remotely connected to R2 on line 2
************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB        Local Address               Foreign Address             (state)
F3B64440   10.123.4.2.23               10.45.1.11.34115            ESTAB
```

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 10.45.1.10:1      10.78.9.7:1        10.123.4.1:1       10.123.4.1:1
--- 10.45.1.10         10.78.9.7          ---                ---
icmp 10.45.1.11:1      10.78.9.8:1        10.123.4.1:1       10.123.4.1:1
tcp 10.45.1.11:34115   10.78.9.8:34115    10.123.4.2:23      10.123.4.2:23
--- 10.45.1.11         10.78.9.8          ---                ---
```

Failed NAT Pool Allocation/Reset NAT Pool

When the pool is exhausted, no additional translation can occur until the global IP address is returned to the pool.

Example 15-39 demonstrates this concept with NAT failing on R5 and packets being dropped.

- The default timeout for NAT translations is 24 hours, but this can be changed with the command **`ip nat translation timeout`** *`seconds`*.
- The dynamic NAT translations can be cleared out with the command **`clear ip nat translation`** {*`ip-address`* | *}
  - This removes all existing translations
    - Could interrupt traffic flow on active sessions as they might be assigned new global IP addresses.

Example 15-39 *Failed NAT Pool Allocation*

```
R9# telnet 10.123.4.1
Trying 10.123.4.1 ...
% Destination unreachable; gateway or host down
```

```
R5# debug ip nat detailed
IP NAT detailed debugging is on
R5#
 02:22:58.685: NAT: failed to allocate address for 10.78.9.9, list/map
ACL-NAT-CAPABLE
 02:22:58.685:  mapping pointer available mapping:0
 02:22:58.685: NAT*: Can't create new inside entry - forced_punt_flags: 0
 02:22:58.685: NAT: failed to allocate address for 10.78.9.9, list/map ACL-NAT-CAPABLE
 02:22:58.685:  mapping pointer available mapping:0
 02:22:58.685: NAT: translation failed (A), dropping packet s=10.78.9.9 d=10.123.4.1
```

Example 15-40 *Clearing NAT Translation to Reset the NAT Pool*

```
R5# clear ip nat translation *
```

```
R9# telnet 10.123.4.1
Trying 10.123.4.1 ... Open
************************************************************
* You have remotely connected to R1 on line 2
************************************************************
User Access Verification
Password:

R1#
```

Port Address Translation

Pooled NAT translation has the limitation of ensuring that the number of global IP addresses is adequate to meet the needs of the local IP addresses.

Port Address Translation (PAT) is an iteration of NAT that allows for a mapping of **many local** IP addresses **to one global** IP address.

The NAT device maintains the state of translations by dynamically changing the source ports as a packet leaves the outside interface.

Another term for PAT is NAT overload.

Configuring PAT

The steps for configuring PAT are as follows:

Step 1. Configure the outside interface by using the command **`ip nat outside`**.

Step 2. Configure the inside interface with the command **`ip nat inside`**.

Step 3. Specify which traffic can be translated by using a standard or extended ACL

referenced by number or name.

Step 4. Configure Port Address Translation by using the command **`ip nat inside source list`** *`acl`* {`interface` *`interface-id`* | `pool` *`nat-pool-name`*} **`overload`**.

Example 15-41 *Configuring PAT on R5*

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# ip access-list standard ACL-NAT-CAPABLE
R5(config-std-nacl)# permit 10.78.9.0 0.0.0.255
R5(config-std-nacl)# exit
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config)# ip nat source list ACL-NAT-CAPABLE interface GigabitEthernGi0/0 overload
```

Generating Traffic for PAT

Example 15-42 *Generating Network Traffic for PAT*

```
R7# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R8# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R9# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R7# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
************************************************************
* You have remotely connected to R2 on line 2
************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB        Local Address               Foreign Address             (state)
F3B64440   10.123.4.2.23               10.45.1.5.51576             ESTAB
```

Now that PAT has been configured on R5, traffic can be generated for testing.

```
R8# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
************************************************************
* You have remotely connected to R2 on line 3
************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB        Local Address               Foreign Address             (state)
F3B64440   10.123.4.2.23               10.45.1.5.51576             ESTAB
F3B65560   10.123.4.2.23               10.45.1.5.31515             ESTAB
```

NAT Translation Table With PAT

Figure 15-10 shows R5's translation table after all the various flows have established.

Example 15-43 shows R5's NAT translation table. By taking the ports from the TCP brief sessions on R2 and correlating them to R5's NAT translation table, you can identify which TCP session belongs to R7 or R8.

Inside Global	Inside Local	Outside Local	Outside Global
10.45.1.5:4	10.78.9.7:3	10.123.4.1:3	10.123.4.1:4
10.45.1.5:3	10.78.9.8:3	10.123.4.1:3	10.123.4.1:3
10.45.1.5:1	10.78.9.9:1	10.123.4.1:1	10.123.4.1:1
10.45.1.5:51576	10.78.9.7:51576	10.123.4.2:23	10.123.4.2:23
10.45.1.5:31515	10.78.9.8:31515	10.123.4.2:23	10.123.4.2:23

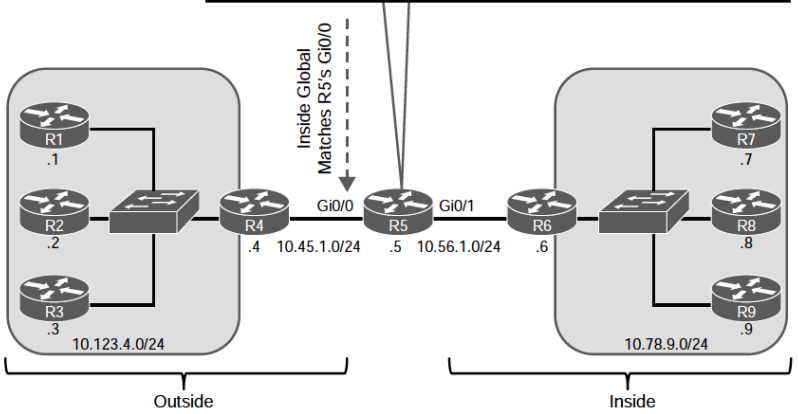
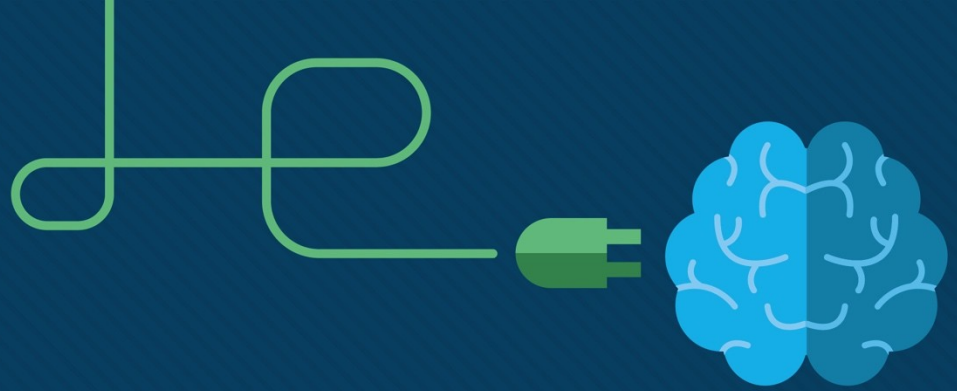


Figure 15-10 *R5's Translation Table for PAT*

Example 15-43 *R5's NAT Translation Table with PAT*

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 10.45.1.5:4       10.78.9.7:3        10.123.4.1:3       10.123.4.1:4
icmp 10.45.1.5:3       10.78.9.8:3        10.123.4.1:3       10.123.4.1:3
icmp 10.45.1.5:1       10.78.9.9:1        10.123.4.1:1       10.123.4.1:1
tcp 10.45.1.5:51576    10.78.9.7:51576    10.123.4.2:23      10.123.4.2:23
tcp 10.45.1.5:31515    10.78.9.8:31515    10.123.4.2:23      10.123.4.2:23
```



Chapter 16: Overlay Tunnels

Instructor Materials

CCNP Enterprise: Core Networking

Generic Routing Encapsulation (GRE) Tunnels

- GRE is a tunneling protocol that provides connectivity to a wide variety of network-layer protocols by encapsulating and forwarding packets over an IP-based network.
- GRE can be used to tunnel traffic through a firewall or an ACL or to connect discontiguous networks.
- The most important application of GRE tunnels is that they can be used to create VPNs.

GRE Packet Headers

- When a router encapsulates a packet for a GRE tunnel, it adds new header information (known as encapsulation) to the packet. This new header contains the remote endpoint IP address as the destination.
- The new IP header information enables the packet to be routed between the two tunnel endpoints without inspection of the packet's payload.
- When the packet reaches the remote tunnel endpoint, the GRE headers are removed (known as de-encapsulation) and the original packet is forwarded out of the router.

Figure 16-1 illustrates an IP packet before and after GRE encapsulation. GRE tunnels support IPv4 or IPv6 addresses as an underlay or overlay network.

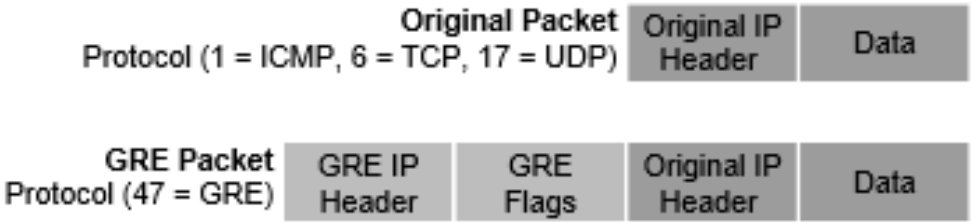


Figure 16-1 *IP Packet Before and After GRE Headers*

GRE Tunnel Configuration

Figure 16-2 illustrates a topology where R1 and R2 are using their respective ISP routers as their default gateways to reach the internet. Example 16-1 shows the routing table on R1.

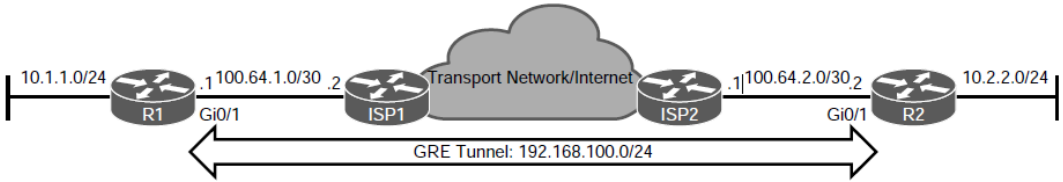


Figure 16-2 *GRE Tunnel Topology*

Example 16-1 *R1's Routing Table Without GRE Tunnel*

```
R1# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
..
       ia - IS-IS inter area, * - candidate default, U - per-user static route

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 100.64.1.2
..
```

GRE Tunnel Configuration (Cont.)

The steps for configuring GRE tunnels are as follows:

Step 1. Create the tunnel interface by using the global configuration command **`interface tunnel`** *`tunnel-number`*.

Step 2. Identify the local source of the tunnel by using the interface parameter command **`tunnel source`** {*`ip-address | interface-id`*}.

- The tunnel source can be a physical interface or a loopback interface.

Step 3. Identify the remote destination IP address by command **`tunnel destination`** *`ip-address`*.

Step 4. Allocate an IP address to the tunnel interface by using the command **`ip address`** *`ip-address subnet-mask`*.

GRE Tunnel Configuration (Cont.)

Optional GRE configuration steps:

Step 5. (Optional) Define the tunnel **bandwidth** for use by QoS or for routing protocol metrics. Bandwidth is defined with the interface parameter command **`bandwidth`** *`[1-10000000]`*, which is measured **in kilobits per second**.

Step 6. (Optional) Specify a GRE tunnel keepalive with the interface parameter command **`keepalive`** *`[seconds [retries]]`*. The default timer is 10 seconds, with 3 retries. Tunnel keepalives ensure that bidirectional communication exists between tunnel endpoints to keep the line protocol up.

Step 7. (Optional) Define the IP **maximum transmission unit** (MTU) for the tunnel interface. Specifying the IP MTU on the tunnel interface has the router perform the fragmentation in advance of the host having to detect and specify the packet MTU. IP MTU is configured with the interface parameter command **`ip mtu`** *`mtu`*.

GRE Tunnel Configuration (C

Example 16-2 provides a GRE tunnel configuration for R1 and R2, following the steps for GRE configuration listed earlier.

With this configuration, R1 and R2 become direct OSPF neighbors over the GRE tunnel and learn each other's routes.

R1

```
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.1 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.2.2
!
router ospf 1
 router-id 1.1.1.1
 network 10.1.1.1 0.0.0.0 area 1
 network 192.168.100.1 0.0.0.0 area 0
!
ip route 0.0.0.0 0.0.0.0 100.64.1.2
```

R2

```
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.2 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.1.1
!
router ospf 1
 router-id 2.2.2.2
 network 10.2.2.0 0.0.0.255 area 2
 network 192.168.100.2 0.0.0.0 area 0

ip route 0.0.0.0 0.0.0.0 100.64.2.1
```

Example 16-2 *Configuring GRE*

GRE Tunnel Verification

The state of the GRE tunnel can be verified with the command **show interface tunnel** *number*. Example 16-3 shows output from this command.

Example 16-3 *Displaying GRE Tunnel Parameters*

```
R1# show interfaces tunnel 100 | include Tunnel.*is|Keepalive|Tunnel s|Tunnel p
Tunnel100 is up, line protocol is up
  Keepalive set (5 sec), retries 3
  Tunnel source 100.64.1.1 (GigabitEthernet0/1), destination 100.64.2.2
  Tunnel protocol/transport GRE/IP
```

GRE Tunnel Verification (Cont.)

Additional commands to verify the status of a GRE tunnel include **`show ip route`** and **`traceroute`**. Examples 16-4 and 16-5 show the output of these commands when the GRE tunnel is active.

Example 16-4 *R1 Routing Table with GRE*

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

! Output omitted for brevity

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 100.64.1.2
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
        10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C          10.1.1.0/24 is directly connected, GigabitEthernet0/3
L          10.1.1.1/32 is directly connected, GigabitEthernet0/3
O IA       10.2.2.0/24 [110/26] via 192.168.100.2, 00:17:37, Tunnel100
        100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          100.64.1.0/30 is directly connected, GigabitEthernet0/1
L          100.64.1.1/32 is directly connected, GigabitEthernet0/1
        192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.100.0/24 is directly connected, Tunnel100
L          192.168.100.1/32 is directly connected, Tunnel100
```

Example 16-5 *Verifying the Tunnel*

```
R1# traceroute 10.2.2.2 source 10.1.1.1
Tracing the route to 10.2.2.2
  1 192.168.100.2 3 msec 5 msec *
```

Problems with Overlay Networks

Recursive routing and outbound interface selection are two common problems with tunnel or overlay networks.

- Recursive routing can occur when the transport network is advertised into the same routing protocol that runs on the overlay network.
- Routers detect recursive route and generate syslog messages.
- Recursive routing problems are remediated by preventing the tunnel endpoint address from being advertised across the tunnel network.

IPsec Fundamentals

- IPsec is a framework of open standards for creating highly secure virtual private networks (VPNs).
- IPsec provides security services such as peer authentication, data confidentiality, data integrity and replay detection.

IPSec Security Services

Table 16-3 *IPsec Security Services*

Security Service	Description	Methods Used
Peer authentication	Verifies the identity of the VPN peer through authentication.	• Pre-Shared Key (PSK) • Digital certificates
Data confidentiality	Protects data from eavesdropping attacks through encryption algorithms. Changes plaintext into encrypted ciphertext.	• Data Encryption Standard (DES) • Triple DES (3DES) • Advanced Encryption Standard (AES) The use of DES and 3DES is not recommended.
Data integrity	Prevents man-in-the-middle (MitM) attacks by ensuring that data has not been tampered with during its transit across an unsecure network.	Hash Message Authentication Code (HMAC): • Message Digest 5 (MD5) algorithm • Secure Hash Algorithm (SHA-1) The use of MD5 is not recommended.
Replay detection	Prevents MitM attacks where an attacker captures VPN traffic and replays it back to a VPN peer with the intention of building an illegitimate VPN tunnel.	Every packet is marked with a unique sequence number. A VPN device keeps track of the sequence number and does not accept a packet with a sequence number it has already processed.

IPSec Packet Headers

IPsec uses two different packet headers to deliver security:

- **Authentication Header** - The authentication header ensures that the original data packet (before encapsulation) has not been modified during transport on the public network. The authentication header does not support encryption, and is not recommended unless authentication is all that is desired.
- **Encapsulating Security Payload (ESP)** - ESP ensures that the original payload (before encapsulation) maintains data confidentiality by encrypting the payload and adding a new set of headers during transport across a public network.

IPSec Packet Transport

Traditional IPsec provides two modes of packet transport:

- **Tunnel mode** - Encrypts the entire original packet and adds a new set of IPsec headers. These new headers are used to route the packet and also provide overlay functions.
- **Transport mode** - Encrypts and authenticates only the packet payload. This mode does not provide overlay functions and routes based on the original IP headers.

Figure 16-3 shows an original packet, an IPsec packet in transport mode, and an IPsec packet in tunnel mode.

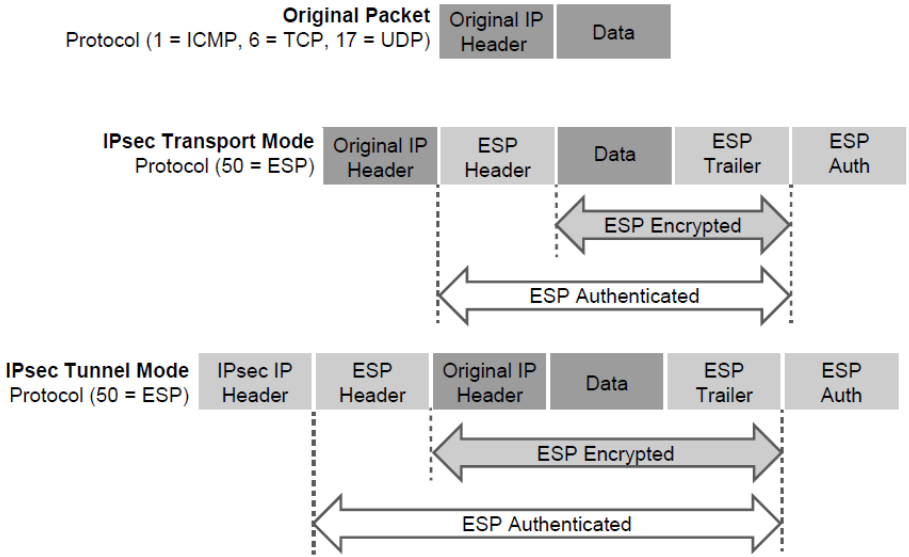


Figure 16-3 *IPsec Transport and Tunnel Encapsulation*

IPSec Encryption, Hashing and Keying

IPsec supports encryption, hashing, and keying methods to provide security services:

- **Data Encryption Standard (DES) -** A 56-bit symmetric data encryption algorithm that can encrypt the data sent over a VPN. This algorithm is very weak and should be avoided.
- **Triple DES (3DES) -** A data encryption algorithm that runs the DES algorithm three times with three different 56-bit keys. Using this algorithm is no longer recommended. The more advanced and more efficient AES should be used instead.
- **Advanced Encryption Standard (AES) -** A symmetric encryption algorithm used for data encryption that was developed to replace DES and 3DES. AES supports key lengths of 128 bits, 192 bits, or 256 bits and is based on the Rijndael algorithm.

IPSec Encryption, Hashing and Keying (Cont.)

- **Message Digest 5 (MD5) -** A one-way, 128-bit hash algorithm used for data authentication. Cisco devices use MD5 HMAC, which provides an additional level of protection against MitM attacks. Using this algorithm is no longer recommended, and SHA should be used instead.
- **Secure Hash Algorithm (SHA) -** A one-way, 160-bit hash algorithm used for data authentication. Cisco devices use the SHA-1 HMAC, which provides additional protection against MitM attacks.
- **Diffie-Hellman (DH) -** An asymmetric key exchange protocol that enables two peers to establish a shared secret key used by encryption algorithms such as AES over an unsecure communications channel.
- **RSA signatures -** A public-key (digital certificates) cryptographic system used to mutually authenticate the peers.
- **Pre-Shared Key -** A security mechanism in which a locally configured key is used as a credential to mutually authenticate the peers

Transform Sets

IPSec proposal

A transform set is a combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

Transform Type	Transform	Description
Authentication header transform (only one allowed)	ah-md5-hmac	Authentication header with the MD5 authentication algorithm (not recommended)
	ah-sha-hmac	Authentication header with the SHA authentication algorithm
	ah-sha256-hmac	Authentication header with the 256-bit AES authentication algorithm
	ah-sha384-hmac	Authentication header with the 384-bit AES authentication algorithm
	ah-sha512-hmac	Authentication header with the 512-bit AES authentication algorithm

Table 16-4 Allowed Transform Set Combinations

Transform Sets (Cont.)

Transform Type	Transform	Description
ES ESP encryption transform (only one allowed)	esp-aes	ESP with the 128-bit AES encryption algorithm
	esp-gcm esp-gmac	ESP with either a 128-bit (default) or a 256-bit encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des esp-3des	ESPs with 56-bit and 168-bit DES encryption (no longer recommended)
	esp-null	Null encryption algorithm
	esp-seal	ESP with the 160-bit SEAL encryption algorithm

Table 16-4 Allowed Transform Set Combinations

Transform Sets (Cont.)

Transform Type	Transform	Description
ESP authentication transform (only one allowed)	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm (no longer recommended)
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP compression transform	comp-lzs	IP compression with the Lempel-Ziv-Stac (LZS) algorithm

Table 16-4 Allowed Transform Set Combinations

Internet Key Exchange

- Internet Key Exchange (IKE) is a protocol that performs authentication between two end- points to establish security associations (SAs), also known as IKE tunnels.
- There are two versions of IKE: IKEv1 (specified in RFC 2409) and IKEv2 (specified in RFC 7296).
- Internet Security Association Key Management Protocol (ISAKMP) is a framework for authentication and key exchange between two peers to establish, modify, and tear down SAs.
- For Cisco platforms, IKE is analogous to ISAKMP, and the two terms are used interchangeably.

Internet Key Exchange (Cont.)

IKEv1 defines two phases of key negotiation for IKE and IPsec SA establishment:

- **Phase 1 -** Establishes a bidirectional SA between two IKE peers, known as an **ISAKMP SA**. Because the SA is bidirectional, once it is established, either peer may initiate negotiations for phase 2.
- **Phase 2 -** Establishes unidirectional IPsec SAs, leveraging the ISAKMP SA established in phase 1 for the negotiation.

Phase 1 negotiation can occur using **main mode (MM)** or **aggressive mode (AM).** The peer that initiates the SA negotiation process is known as the initiator, and the other peer is known as the responder.

IKE Phase 1 Negotiation Modes

Main mode (MM) consists of six message exchanges and protects information during the negotiation so as not to expose it to eavesdropping.

The six MM message exchanges:

- **MM1 -** First message containing the SA proposals.
- **MM2 -** Sent from the responder with the matching SA proposal.
- **MM3 -** Initiator starts the DH key exchange.
- **MM4 -** Responder sends its own key to the initiator.
- **MM5 -** Initiator starts authentication by sending peer its IP address.
- **MM6 -** Responder sends back a similar packet and authenticates the session. At this point, the ISAKMP SA is established.

IKE Phase 1 Negotiation Modes (Cont.)

Aggressive mode (AM) consists of a three-message exchange and takes less time to negotiate keys between peers. However, it doesn't offer the same level of encryption security provided by MM negotiation, and the identities of the two peers trying to establish a security association are exposed to eavesdropping. These are the three aggressive mode messages:

- **AM1 -** In this message, the initiator sends all the information contained in MM1 through MM3 and MM5.
- **AM2 -** This message sends all the same information contained in MM2, MM4, and MM6.
- **AM3 -** This message sends the authentication that is contained in MM5.

IKE Phase 2 Session Establishment

Phase 2 uses the existing bidirectional IKE SA to securely exchange messages to establish one or more IPsec SAs between the two peers. The method used to establish the IPsec SA is known as **quick mode (QM)**. Quick mode uses a three-message exchange:

- **QM1 -** The initiator (which could be either peer) can start multiple IPsec SAs in a single exchange message. This message includes agreed-upon algorithms for encryption and integrity decided as part of phase 1, as well as what traffic is to be encrypted or secured.
- **QM2 -** This message from the responder has matching IPsec parameters.
- **QM3 -** After this message, there should be two unidirectional IPsec SAs between the two peers.

Perfect Forward Secrecy (PFS) https://en.wikipedia.org/wiki/Forward_secrecy

- feature that gives assurances that session keys will not be compromised even if long-term secrets used in the session key exchange are compromised.
- optional additional function for phase 2
  - it requires additional DH exchanges that consume additional CPU cycles.

IKEv2

IKEv2 is an evolution of IKEv1 that includes many changes and improvements. In IKEv2, communications consist of request and response pairs called exchanges and are sometimes just called request/response pairs.

1. **IKE_SA_INIT** negotiates cryptographic algorithms, exchanges nonces, and performs a DH exchange. This single exchange is equivalent to IKEv1's first two pairs of messages MM1 to MM4.
2. **IKE_AUTH** authenticates the previous messages and exchanges identities and certificates. Then it establishes an IKE SA and a child SA (the IPsec SA). This is equivalent to IKEv1's MM5 to MM6 as well as QM1 and QM2.

It takes a total of four messages to bring up the bidirectional IKE SA and the unidirectional IPsec SAs, as opposed to six with IKEv1 aggressive mode or nine with main mode.

Differences Between IKEv1 and IKEv2

IKEv1	IKEv2
Exchange Modes	
Main Mode Aggressive Mode Quick Mode	IKE Security Association Initialization (SA_INIT) IKE_Auth CREATE_CHILD_SA
Minimum Number of Messages Needed to Establish IPsec SAs	
Nine with main mode Six with aggressive mode	Four
Supported Authentication Methods	
Pre-Shared Key (PSK) Digital RSA Cert (RSA-SIG) Public Key *Both peers must use the same authentication method*	Pre-Shared Key (RSA-SIG) Elliptic Curve Digital Signature Cert (ECDSA-SIG) *Asymetric authentication is supported. Authentication method can be specified during the IKE_AUTH exchange.*

Differences Between IKEv1 and IKEv2 (Cont.)

IKEv1	IKEv2
Next Generation Encryption (NGE)	
Not Supported.	AES-GCM (Galois/Counter Mode) mode SHA-256 SHA-384 SHA-512 HMAC-SHA-256 Elliptic Curve Diffie-Hellman (ECDH) ECDH-384 ECDSA-384
Attack Protection	
MitM protection Eavesdropping protection	MitM protection Eavesdropping protection Anti-DoS protection

Table 16-5 Major Differences Between IKEv1 and IKEv2

IPsec VPN Solutions

Cisco IPsec VPN Solutions:

- **Site-to-Site (LAN-to-LAN) IPsec VPNs -** Site-to-site IPsec VPNs are the most versatile solution for site-to-site encryption because they are the only solution to allow for multivendor interoperability. Difficult to manage in large networks.
- **Cisco Dynamic Multipoint VPN (DMVPN) -** Simplifies configuration for hub-and-spoke and spoke-to-spoke VPNs in Cisco networks. It accomplishes this by combining multipoint GRE (mGRE) tunnels, IPsec, and Next Hop Resolution Protocol (NHRP).
- **Cisco Group Encrypted Transport VPN (GET VPN) -** Developed specifically for enterprises to build any-to-any tunnel-less VPNs (where the original IP header is used) across service provider MPLS networks or private WANs. Provides encryption over private networks which addresses regulatory-compliance guidelines.
- **Cisco FlexVPN -** FlexVPN is Cisco's implementation of the IKEv2 standard, featuring a unified VPN solution that combines site-to-site, remote access, hub-and-spoke topologies and partial meshes (spoke-to-spoke direct). Remains compatible with legacy VPN implementations using crypto maps.
- **Remote VPN Access -** Remote VPN access allows remote users to securely VPN into a corporate network. It is supported on IOS with FlexVPN (IKEv2 only) and on ASA 5500-X and FirePOWER firewalls.

Configuring IPsec Cheatsheet ☺

https://packetlife.net/library/cheat-sheets/

IPsec

packetlife.net

Protocols

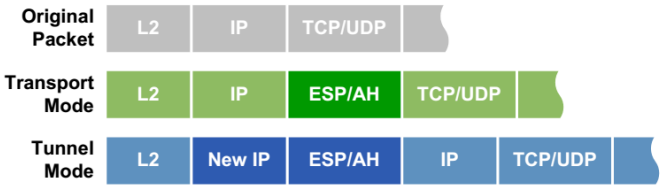
Internet Security Association and Key Management Protocol (ISAKMP)
A framework for the negotiation and management of security associations between peers (traverses UDP/500)

Internet Key Exchange (IKE)
Responsible for key agreement using asymmetric cryptography

Encapsulating Security Payload (ESP)
Provides data encryption, data integrity, and peer authentication; IP protocol 50

Authentication Header (AH)
Provides data integrity and peer authentication, but not data encryption; IP protocol 51

IPsec Modes



Transport Mode
The ESP or AH header is inserted behind the IP header; the IP header can be authenticated but not encrypted

Tunnel Mode
A new IP header is created in place of the original; this allows for encryption of the entire original packet

Configuration

Encryption Algorithms

	Type	Key Length (Bits)	Strength
DES	Symmetric	56	Weak
3DES	Symmetric	168	Medium
AES	Symmetric	128/192/256	Strong
RSA	Asymmetric	1024+	Strong

Hashing Algorithms

	Length (Bits)	Strength
MD5	128	Medium
SHA-1	160	Strong

IKE Phases

Phase 1
A bidirectional ISAKMP SA is established between peers to provide a secure management channel (IKE in main or aggressive mode)

Phase 1.5 (optional)
Xauth can optionally be implemented to enforce user authentication

Phase 2
Two unidirectional IPsec SAs are established for data transfer using separate keys (IKE quick mode)

Terminology

Data Integrity
Secure hashing (HMAC) is used to ensure data has not been altered in transit

Configuring IPsec VPNs

Even though crypto maps are no longer recommended for tunnels, they are still widely deployed and should be understood. The steps to enable IPsec over GRE using crypto maps are as follows:

- **Step 1.** Configure a crypto ACL to classify VPN traffic by using these commands:

```
ip access-list extended acl _name
 permit gre host {tunnel-source IP}  host {tunnel-destination IP}
```

- **Step 2.** Configure an ISAKMP policy for IKE SA by using the command **crypto isakmp policy** *priority*. Within the ISAKMP policy configuration mode, encryption, hash, authentication, and the DH group can be specified with the following commands:

```
encryption {des|3des|aes|aes192|aes256}
hash {sha|sha256|sha384|md5}
authentication {rsa-sig|sa-encr|pre-share}
group {1|2|5|14|15|16|19|20|24}
```

see next slide

The keyword **priority** uniquely identifies the IKE policy and assigns a priority to the policy, where 1 is the highest priority.

Configuring IPsec VPNs

Diffie – Hellman groups

Group 1	768-bit modulo	nepoužívat (prolomitelné šifrování)
Group 2	1024-bit modulo	nepoužívat (prolomitelné šifrování)
Group 5	1536-bit modulo	nepoužívat (prolomitelné šifrování)
Group 14	2048-bit modulo	doporučené minimum
Group 15	3072-bit modulo	nedoporučeno (nedostatečná podpora)
Group 16	4096-bit modulo	nedoporučeno (nedostatečná podpora)
Group 17	6144-bit modulo	nedoporučeno (nedostatečná podpora)
Group 18	8192-bit modulo	nedoporučeno (nedostatečná podpora)
Group 19	192-bit eliptická křivka	doporučeno
Group 21	224-bit eliptická křivka	doporučeno
Group 23	256-bit eliptická křivka	doporučeno
Group 24	384-bit eliptická křivka	doporučeno
Group 25	521-bit eliptická křivka	doporučeno

Configuring IPsec VPNs (Cont.)

- **Step 3.** Configure PSK by using the command **crypto isakmp key** *keystring* **address** *peer-address* [*mask*]. The *keystring* should match on both peers. For *peeraddress* [*mask*], the value **0.0.0.0 0.0.0.0** allows a match against **any peer**.
- **Step 4.** Create a transform set and enter transform set configuration mode by using the command **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]. In transform set configuration mode, enter the command **mode** [**tunnel | transport**] to specify tunnel or transport modes.
- **Step 5**. Configure a crypto map and enter crypto map configuration mode by using the command **crypto map** *map-name seq-num* [ipsec-isakmp]. In **crypto map configuration mode**, use the following commands to specify the crypto ACL to be matched, the IPsec peer, and the transform sets to be negotiated:

match address *acl-name*

set peer {*hostname*|*ip-address*}

set transform-set *transform-set-name1* [*transform-setname2...transform-set-name6*]

- **Step 6.** Apply a crypto map to the outside interface by using the command **crypto map** *map-name*

Example 16-7 *Configuring GRE over IPsec Site-to-Site Tunnel with Pre-Shared Key*

R1

```
crypto isakmp policy 10
authentication pre-share
hash sha256
encryption aes
group 14
!
crypto isakmp key CISCO123 address 100.64.2.2
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport
!
ip access-list extended GRE_IPSEC_VPN
permit gre host 100.64.1.1 host 100.64.2.2
!
crypto map VPN 10 ipsec-isakmp
match address GRE_IPSEC_VPN
set transform AES_SHA
set peer 100.64.2.2
!
interface GigabitEthernet0/1
 ip address 100.64.1.1 255.255.255.252
crypto map VPN
!
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.1 255.255.255.0
 ip mtu 1400
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.2.2

router ospf 1
 router-id 1.1.1.1
 network 10.1.1.1 0.0.0.0 area 1
 network 192.168.100.1 0.0.0.0 area 0
```

One site, the second one is a mirrored copy

Configuring IPsec Site-to-Site VPN

Verifying Site-to-Site VPN

Commands that can provide information to verify the operation of a site-to-site VPN include:

- **`show interface tunnel100 | include Tunnel protocol`**
- **`show ip ospf neighbor`**
- **`show ip route ospf`**
- **`show crypto isakmp sa`**
- **`show crypto ipsec sa`**

Virtual Tunnel Interface (VTI) VPN

IPSec virtual tunnel interface (VTI)

- Provides a routable interface type for terminating IPSec tunnels.
- Gives easy way to define protection between sites to form an overlay network.
- Simplify configuration of IPSec for protection of remote links .
- Supports multicast.
- Simplifies network management and load balancing.

Virtual Tunnel Interface (VTI) VPN

Features – advantages against policy based VPN

1. Simplier configuratiopn – *configuring IPSec tunnels can be an administrative nightmare if you have a lot of remote peers*.
2. IP Addressing - the tunnel interface will typically have an IP address, but an unnumbered interface can be configured (then set interface IP to 0.0.0.0).
3. Security – tunnel can be referenced by the zone firewall. The tunnel interface can belong to a separate security zone and policies can be defined to control traffic flows across the tunnel interface
4. Routing – static routes can be defined to use the tunnel interface. Dynamic routing protocols can use the tunnel interface. E.g. OSPF neighborships can be formed across the tunnel.
5. Diagnostics – packet captures can be performed on the tunnel interface. This can be valuable when troubleshooting traffic flows across the tunnel.

Cisco Location/ID Separation Protocol (LISP)

- The rapid growth of the default-free zone (DFZ), also known as the internet routing table, led to the development of the *Cisco Location/ID Separation Protocol (LISP).*
- LISP is a routing architecture and a data and control plane protocol that was created to address routing scalability problems on the internet.

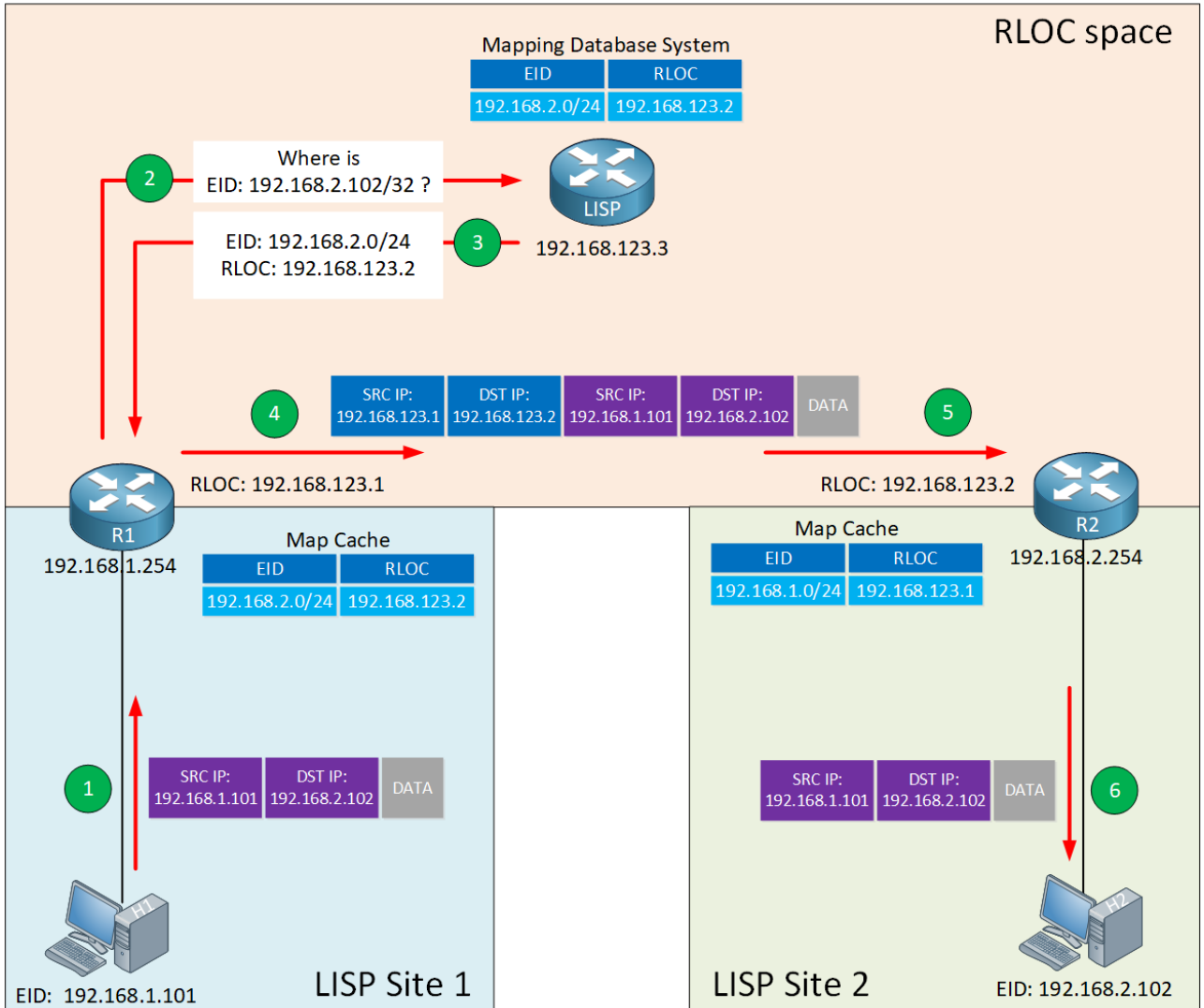
LISP

- A very simplified one-sentence explanation is that LISP is a tunneling protocol that uses a DNS-like system to figure out to which router they should send IP packets.
- Cisco Locator ID Separation Protocol (LISP) is a mapping and encapsulation protocol.
- It is an open standard, defined in RFC 6830. Originally it was designed for the Internet, but nowadays, you can see LISP in data centers, IoT, WAN, and the campus.
- Internet routing tables have grown exponentially, putting a burden on BGP routers. Routing on the Internet is meant to be hierarchical, but because of disaggregation, a full Internet routing table nowadays contains over 800.000 prefixes (https://www.cidr-report.org/as2.0/).
- With traditional IP routing, an IP address has two functions:
  - Identity: To identify the device.
  - Location: The location of the device in the network; we use this for routing.
- **LISP separates these two functions of an IP address into two separate functions:**
  - **Endpoint Identifier (EID)**: Assigned to hosts like computers, laptops, printers, etc.
  - **Routing Locators (RLOC)**: Assigned to routers. We use the RLOC address to reach EIDs.

Cisco Location/ID Separation Protocol (LISP)

LISP

https://networklessons.com/cisco/ccnp-encor-350-401/cisco-locator-id-separation-protocol-lisp



LISP Architecture Components

Key LISP architecture components:

- **Endpoint identifier (EID) -** An EID is the IP address of an endpoint within a LISP site. EIDs are the same IP addresses in use today on endpoints (IPv4 or IPv6), and they operate in the same way.
- **LISP site -** This is the name of a site where LISP routers and EIDs reside.
- **Ingress tunnel router (ITR) -** ITRs are LISP routers that LISP-encapsulate IP packets coming from EIDs that are destined outside the LISP site.
- **Egress tunnel router (ETR) -** ETRs are LISP routers that de-encapsulate LISP-encapsulated IP packets coming from sites outside the LISP site and destined to EIDs within the LISP site.
- **Tunnel router (xTR) -** xTR refers to routers that perform ITR and ETR functions (which is most routers).
- **Proxy ITR (PITR) -** PITRs are just like ITRs but for non-LISP sites that send traffic to EID destinations.

LISP Architecture Components (Cont.)

- **Proxy ETR (PETR) -** PETRs act just like ETRs but for EIDs that send traffic to destinations at non-LISP sites.
- **Proxy xTR (PxTR) -** PxTR refers to a router that performs PITR and PETR functions.
- **LISP router -** A LISP router is a router that performs the functions of any or all of the following: ITR, ETR, PITR, and/or PETR.
- **Routing locator (RLOC) -** An RLOC is an IPv4 or IPv6 address of an ETR that is internet facing or network core facing.
- **Map server (MS) -** This is a network device (typically a router) that learns EID-to-prefix mapping entries from an ETR and stores them in a local EID-to-RLOC mapping database.
- **Map resolver (MR) -** This is a network device (typically a router) that receives LISP-encapsulated map requests from an ITR and finds the appropriate ETR to answer those requests by consulting the map server.
- **Map server/map resolver (MS/MR) -** When MS and the MR functions are implemented on the same device, the device is referred to as an MS/MR.

LISP Architecture and Protocols

LISP Routing Architecture

LISP separates IP addresses into **endpoint identifiers (EIDs)** and **routing locators (RLOCs).** Unlike in traditional IP routing, endpoints can roam from site to site, and the only thing that changes is their RLOC; the EID remains the same.

LISP Control Plane

The control plane operates in a very similar manner to the Domain Name System (DNS). Just as DNS can resolve a domain name into an IP address, LISP can resolve an EID into an RLOC by sending map requests to the **Map Resolver (MR).**

LISP Architecture and Protocols (Cont.)

LISP Data Plane

Ingress Tunnel Routers (ITRs) LISP-encapsulate IP packets received from EIDs in an outer IP UDP header with source and destination addresses in the RLOC space; in other words, they perform IP-in-IP/UDP encapsulation.

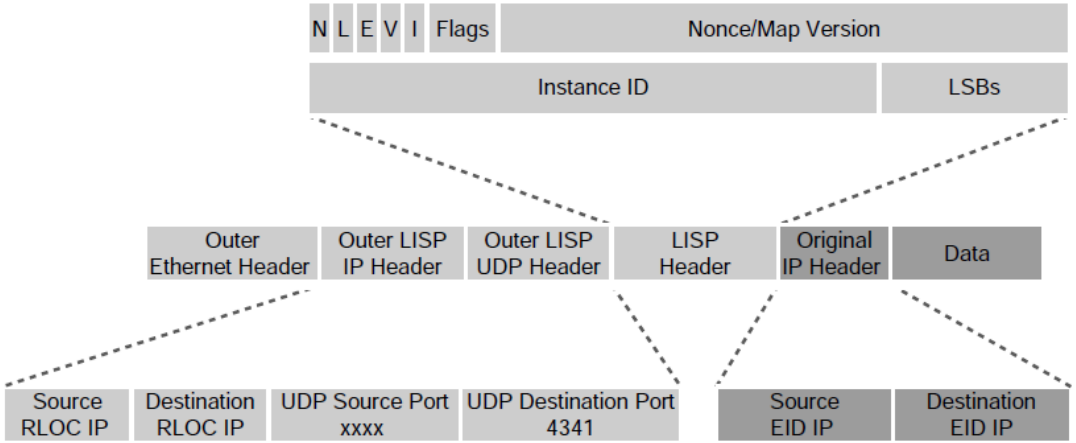


Figure 16-7 *LISP Packet Format*

Virtual Extensible Local Area Network (VXLAN)

- Server Virtualization has placed an increased demand on legacy network infrastructure.
- Layer 2 networks were not designed to support hundreds of thousands of MAC addresses and tens of thousands of VLANs.
- VXLAN is designed to address the issues being seen in traditional Layer 2 networks.

Issues with Legacy Layer 2 Networks

Virtualization has led to a number of problems with traditional Layer 2 Networks:

- The 12-bit VLAN ID yields 4000 VLANs, which are insufficient for server virtualization.
- Large MAC address tables are needed due to the hundreds of thousands of VMs and containers attached to the network.
- STP blocks links to avoid loops, and this results in a large number of disabled links, which is unacceptable.
- ECMP (Equal-cost multi-path routing) is not supported.
- Host mobility is difficult to implement.

VXLAN in a nutshell

Virtual Extensible LAN (VXLAN) is a network virtualization technology, that

- Attempts to address the scalability problems associated with large cloud computing deployments.
- Uses a VLAN-like encapsulation technique to encapsulate OSI layer 2 Ethernet frames within layer 4 UDP datagrams
- Uses port 4789/UDP as the default IANA-assigned destination UDP port number.
- VXLAN endpoints, which terminate VXLAN tunnels, may be either virtual or physical switch ports, are known as VXLAN tunnel endpoints (VTEPs).
- RFC 7348

VXLAN Network Identifier

VXLAN has a 24-bit **VXLAN network identifier (VNI)**, which allows for up to 16 million VXLAN segments (more commonly known as overlay networks) to coexist within the same infrastructure.

- VNI is located in the VXLAN shim header that encapsulates the original inner MAC frame originated by an endpoint. The VNI is used to provide segmentation for Layer 2 and Layer 3 traffic.
- To facilitate the discovery of VNIs over the underlay Layer 3 network, virtual tunnel endpoints (VTEPs) are used.
- Each VTEP has two interfaces:
  Local LAN interfaces - These interfaces on the local LAN segment provide bridging between local hosts.
  IP interface - This is a core-facing network interface for VXLAN. The IP interface's IP address helps identify the VTEP in the network.

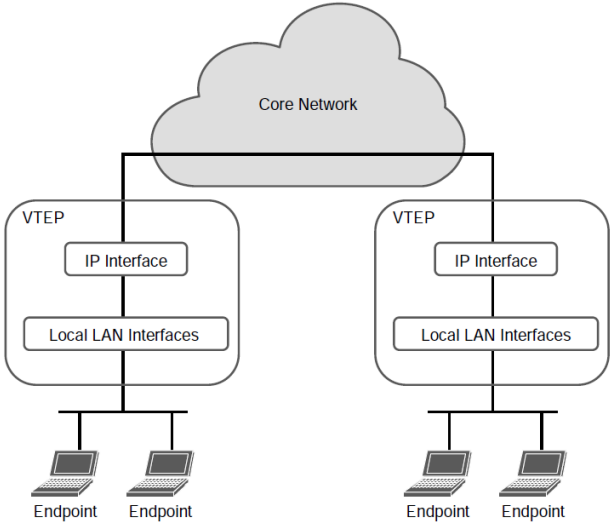


Figure 16-14 *VXLAN VTEP*

VXLAN Headers

There are minor differences between the **Layer 2 LISP** specification and the **VXLAN** specification headers. LISP fields not ported over to VXLAN are reserved for future use.

Cisco Software Defined Access (SD-Access) is an example of an implementation of VXLAN with the LISP control plane.

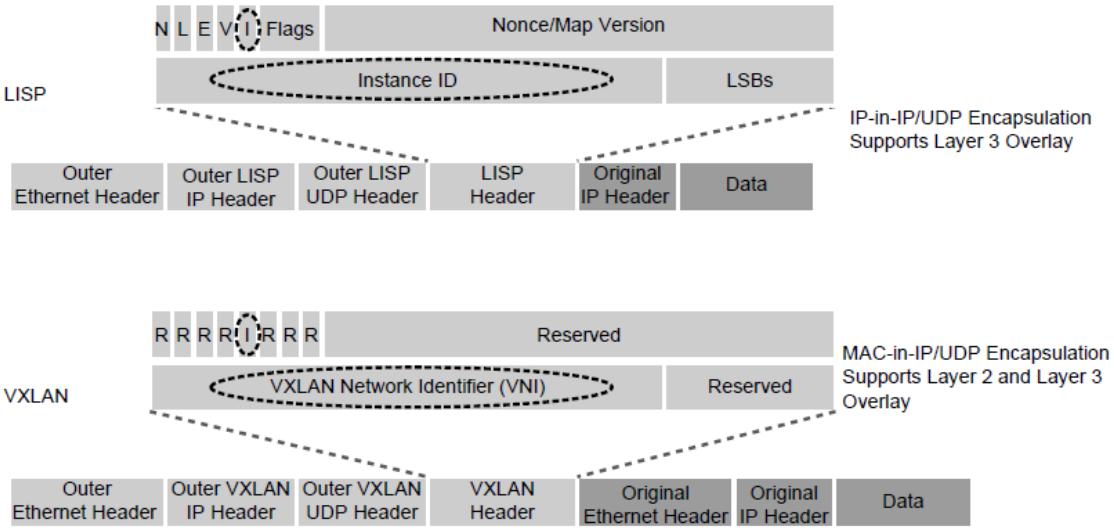


Figure 16-15 *LISP and VXLAN Packet Format Comparison*


CISCO