# Chapter 1

As more devices are added to a cable, the less efficient the network becomes as devices wait until there is not any communication. All of the devices are in the same collision domain. Network hubs proliferate the problem because they add port density while repeating traffic, thereby increasing the size of the collision domain. Network hubs do not have any intelligence in them to direct network traffic; they simply repeat traffic out of every port.

*Virtual LANs (VLANs)* provide logical segmentation by creating multiple broadcast domains on the same network switch. VLANs provide higher utilization of switch ports because a port can be associated to the necessary broadcast domain, and multiple broadcast domains can reside on the same switch. Network devices in one VLAN cannot communicate with devices in a different VLAN via traditional Layer 2 or broadcast traffic.

## Access Ports

*Access ports* are the fundamental building blocks of a managed switch. An access port is assigned to only one VLAN. It carries traffic from the specified VLAN to the device connected to it or from the device to other devices on the same VLAN on that switch. The 802.1Q tags are not included on packets transmitted or received on access ports.

Catalyst switches place switch ports as Layer 2 access ports for VLAN 1 by default. The port can be manually configured as an access port with the command **switchport mode access**. A specific VLAN is associated to the port with the command **switchport access {vlan** *vlan-id* **| name** *vlanname***}**. The ability to set VLANs to an access port by name was recently added with newer code but is stored in numeric form in the configuration.

### Trunk Ports

*Trunk ports* can carry multiple VLANs. Trunk ports are typically used when multiple VLANs need connectivity between a switch and another switch, router, or firewall and use only one port. Upon receipt of the packet on the remote trunk link, the headers are examined, traffic is associated to the proper VLAN, then the 802.1Q headers are removed, and traffic is forwarded to the next port, based on MAC address for that VLAN.

The MAC address table resides in *content addressable memory (CAM)*. The CAM uses high-speed memory that is faster than typical computer RAM due to its search techniques. The CAM table provides a binary result for any query of 0 for true or 1 for false. The CAM is used with other functions to analyze and forward packets very quickly. Switches are built with large CAM to accommodate all the Layer 2 hosts for which they must maintain forwarding tables.

The *Address Resolution Protocol (ARP)* table provides a method of mapping Layer 3 IP addresses to Layer 2 MAC addresses by storing the IP address of a host and its corresponding MAC address. The device then uses the ARP table to add the appropriate Layer 2 headers to the data packet before sending it down to Layer 2 for processing and forwarding.

The source device must add the appropriate Layer 2 headers (source and destination MAC addresses), but the destination MAC address is needed for the next-hop IP address. The device looks for the next-hop IP addresses entry in the ARP table and uses the MAC address from the next-hop IP address's entry as the destination MAC address. The next step is to send the data packet down to Layer 2 for processing and forwarding.

The next router receives the packet based on the destination MAC address, analyzes the destination IP address, locates the appropriate network entry in its routing table, identifies the outbound interface, and then finds the MAC address for the destination device (or the MAC address for the next-hop address if it needs to be routed further). The router then modifies the source MAC address to the MAC address of the router's outbound interface and modifies the destination MAC address to the MAC address for the destination device (or next-hop router).

IPv4 addresses are assigned with the interface configuration command **ip address** *ip-address subnet-mask*. An interface with a configured IP address and that is in an *up* state injects the associated network into the router's routing table (*Routing Information Base [RIB]*). Connected networks or routes have an *administrative distance (AD)* of zero. It is not possible for any other routing protocol to preempt a connected route in the RIB.

### Process Switching

*Process switching*, also referred to as *software switching* or *slow path*, is a switching mechanism in which the general-purpose CPU on a router is in charge of packet switching. In IOS, the ip_input process runs on the general-purpose CPU for processing incoming IP packets. Process switching is the fallback for CEF because it is dedicated to processing punted IP packets when they cannot be switched by CEF.

### Cisco Express Forwarding

*Cisco Express Forwarding (CEF)* is a Cisco proprietary switching mechanism developed to keep up with the demands of evolving network infrastructures. It has been the default switching mechanism on most Cisco platforms that do all their packet switching using the general-purpose CPU (software-based routers) since the 1990s, and it is the default switching mechanism used by all Cisco platforms that use specialized application-specific integrated circuits (ASICs) and network processing units (NPUs) for high packet throughput (hardware-based routers).

The general-purpose CPUs on software-based and hardware-based routers are similar and perform all the same functions; the difference is that on software-based routers, the general-purpose CPU is in charge of all operations, including CEF switching (software CEF), and the hardware-based routers do CEF switching using forwarding engines that are implemented in specialized ASICs, ternary content addressable memory (TCAM), and NPUs (hardware CEF). Forwarding engines provide the packet switching, forwarding, and route lookup capability to routers.

### Ternary Content Addressable Memory

A switch's *ternary content addressable memory (TCAM)* allows for the matching and evaluation of a packet on more than one field. TCAM is an extension of the CAM architecture but enhanced to allow for upper-layer processing such as identifying the Layer 2/3 source/destination addresses, protocol, QoS markings, and so on. TCAM provides more flexibility in searching than does CAM, which is binary. A TCAM search provides three results: 0 for true, 1 false, and X for do not care, which is a ternary combination.

The TCAM entries are stored in Value, Mask, and Result (VMR) format. The value indicates the fields that should be searched, such as the IP address and protocol fields. The mask indicates the field that is of interest and that should be queried. The result indicates the action that should be taken with a match on the value and mask. Multiple actions can be selected besides allowing or dropping traffic, but tasks like redirecting a flow to a QoS policer or specifying a pointer to a different entry in the routing table are possible.

Most switches contain multiple TCAM entries so that inbound/outbound security, QoS, and Layer 2 and Layer 3 forwarding decisions occur all at once. TCAM operates in hardware, providing faster processing and scalability than process switching. This allows for some features like ACLs to process at the same speed regardless of whether there are 10 entries or 500.

## Software CEF

Software CEF, also known as the *software Forwarding Information Base*, consists of the following components:

- **Forwarding Information Base:** The FIB is built directly from the routing table and contains the next-hop IP address for each destination in the network. It keeps a mirror image of the forwarding information contained in the IP routing table. When a routing or topology change occurs in the network, the IP routing table is updated, and these changes are reflected in the FIB. CEF uses the FIB to make IP destination prefix-based switching decisions.
- **Adjacency table:** The adjacency table, also known as the Adjacency Information Base (AIB), contains the directly connected next-hop IP addresses and their corresponding next-hop MAC addresses, as well as the egress interface's MAC address. The adjacency table is populated with data from the ARP table or other Layer 2 protocol tables.

## SDM Templates

The capacity of MAC addresses that a switch needs compared to the number of routes that it holds depends on where it is deployed in the network. The memory used for TCAM tables is limited and statically allocated during the bootup sequence of the switch. When a section of a hardware resource is full, all processing overflow is sent to the CPU, which seriously impacts the performance of the switch.

The allocation ratios between the various TCAM tables are stored and can be modified with Switching Database Manager (SDM) templates. Multiple Cisco switches exist, and the SDM template can be configured on Catalyst 9000 switches with the global configuration command **sdm prefer** {**vlan** | **advanced**}. The switch must then be restarted with the **reload** command.

# Chapter 2

The 802.1D STP standard defines the following three port types:

- **Root port (RP):** A network port that connects to the root bridge or an upstream switch in the spanning-tree topology. There should be only one root port per VLAN on a switch.
- **Designated port (DP):** A network port that receives and forwards BPDU frames to other switches. Designated ports provide connectivity to downstream devices and switches. There should be only one active designated port on a link.
- **Blocking port:** A network that is not forwarding traffic because of STP calculations.

## STP Key Terminology

Several key terms are related to STP:

- **Root bridge:** The root bridge is the most important switch in the Layer 2 topology. All ports are in a forwarding state. This switch is considered the top of the spanning tree for all path calculations by other switches. All ports on the root bridge are categorized as designated ports.
- **Bridge protocol data unit (BPDU):** This network packet is used for network switches to identify a hierarchy and notify of changes in the topology. A BPDU uses the destination MAC address 01:80:c2:00:00:00. There are two types of BPDUs:
  - **Configuration BPDU:** This type of BPDU is used to identify the root bridge, root ports, designated ports, and blocking ports. The configuration BPDU consists of the following fields: STP type, root path cost, root bridge identifier, local bridge identifier, max age, hello time, and forward delay.
  - **Topology change notification (TCN) BPDU:** This type of BPDU is used to communicate changes in the Layer 2 topology to other switches. This is explained in greater detail later in the chapter.
- **Root path cost:** This is the combined cost for a specific path toward the root switch.
- **System priority:** This 4-bit value indicates the preference for a switch to be root bridge. The default value is 32,768.
- **System ID extension:** This 12-bit value indicates the VLAN that the BPDU correlates to. The system priority and system ID extension are combined as part of the switch's identification of the root bridge.
- **Root bridge identifier:** This is a combination of the root bridge system MAC address, system ID extension, and system priority of the root bridge.
- **Local bridge identifier:** This is a combination of the local switch's bridge system MAC address, system ID extension, and system priority of the root bridge.
- **Max age:** This is the maximum length of time that passes before a bridge port saves its BPDU information. The default value is 20 seconds, but the value can be configured with the command **spanning-tree vlan** *vlan-id* **max-age** *maxage*. If a switch loses contact with the BPDU's source, it assumes that the BPDU information is still valid for the duration of the Max Age timer.
- **Hello time:** This is the time that a BPDU is advertised out of a port. The default value is 2 seconds, but the value can be configured to 1 to 10 seconds with the command **spanning-tree vlan** *vlan-id* **hello-time** *hello-time*.
- **Forward delay:** This is the amount of time that a port stays in a listening and learning state. The default value is 15 seconds, but the value can be changed to a value of 15 to 30 seconds with the command **spanning-tree vlan** *vlan-id* **forward-time** *forward-time*.

### Root Bridge Election

The first step with STP is to identify the root bridge. As a switch initializes, it assumes that it is the root bridge and uses the local bridge identifier as the root bridge identifier. It then listens to its neighbor's configuration BPDU and does the following:

- If the neighbor's configuration BPDU is inferior to its own BPDU, the switch ignores that BPDU.
- If the neighbor's configuration BPDU is preferred to its own BPDU, the switch updates its BPDUs to include the new root bridge identifier along with a new root path cost that correlates to the total path cost to reach the new root bridge. This process continues until all switches in a topology have identified the root bridge switch.

STP deems a switch more preferable if the priority in the bridge identifier is lower than the priority of the other switch's configuration BPDUs. If the priority is the same, then the switch prefers the BPDU with the lower system MAC.

### Locating Root Ports

After the switches have identified the root bridge, they must determine their root port (RP). The root bridge continues to advertise configuration BPDUs out all of its ports. The switch compares the BPDU information to identify the RP. The RP is selected using the following logic (where the next criterion is used in the event of a tie):

1. The interface associated to lowest path cost is more preferred.
2. The interface associated to the lowest system priority of the advertising switch is preferred next.
3. The interface associated to the lowest system MAC address of the advertising switch is preferred next.
4. When multiple links are associated to the same switch, the lowest port priority from the advertising switch is preferred.
5. When multiple links are associated to the same switch, the lower port number from the advertising switch is preferred.

## STP Topology Changes

In a stable Layer 2 topology, configuration BPDUs always flow from the root bridge toward the edge switches. However, changes in the topology (for example, switch failure, link failure, or links becoming active) have an impact on all the switches in the Layer 2 topology.

The switch that detects a link status change sends a topology change notification (TCN) BPDU toward the root bridge, out its RP. If an upstream switch receives the TCN, it sends out an acknowledgment and forwards the TCN out its RP to the root bridge.

Upon receipt of the TCN, the root bridge creates a new configuration BPDU with the Topology Change flag set, and it is then flooded to all the switches. When a switch receives a configuration BPDU with the Topology Change flag set, all switches change their MAC address timer to the forwarding delay timer (with a default of 15 seconds). This flushes out MAC addresses for devices that have not communicated in that 15-second window but maintains MAC addresses for devices that are actively communicating.

Flushing the MAC address table prevents a switch from sending traffic to a host that is no longer reachable by that port. However, a side effect of flushing the MAC address table is that it temporarily increases the unknown unicast flooding while it is rebuilt. Remember that this can impact hosts because of their CSMA/CD behavior. The MAC address timer is then reset to normal (300 seconds by default) after the second configuration BPDU is received.

TCNs are generated on a VLAN basis, so the impact of TCNs directly correlates to the number of hosts in a VLAN. As the number of hosts increase, the more likely TCN generation is to occur and the more hosts that are impacted by the broadcasts. Topology changes should be checked as part of the troubleshooting process. Chapter 3 describes mechanisms such as portfast that modify this behavior and reduce the generation of TCNs.

Topology changes are seen with the command **show spanning-tree** [**vlan** *vlan-id*] **detail** on a switch bridge. The output of this command shows the topology change count and time since the last change has occurred. A sudden or continuous increase in TCNs indicates a potential problem and should be investigated further for flapping ports or events on a connected switch.

## Rapid Spanning Tree Protocol

802.1D did a decent job of preventing Layer 2 forwarding loops, but it used only one topology tree, which introduced scalability issues. Some larger environments with multiple VLANs need different STP topologies for traffic engineering purposes (for example, load-balancing, traffic steering). Cisco created Per-VLAN Spanning Tree (PVST) and Per-VLAN Spanning Tree Plus (PVST+) to allow more flexibility.

PVST and PVST+ were proprietary spanning protocols. The concepts in these protocols were incorporated with other enhancements to provide faster convergence into the IEEE 802.1W specification, known as Rapid Spanning Tree Protocol (RSTP).

## RSTP (802.1W) Port States

RSTP reduces the number of port states to three:

- **Discarding:** The switch port is enabled, but the port is not forwarding any traffic to ensure that a loop is not created. This state combines the traditional STP states disabled, blocking, and listening.
- **Learning:** The switch port modifies the MAC address table with any network traffic it receives. The switch still does not forward any other network traffic besides BPDUs.
- **Forwarding:** The switch port forwards all network traffic and updates the MAC address table as expected. This is the final state for a switch port to forward network traffic.

## Building the RSTP Topology

With RSTP, switches exchange handshakes with other RSTP switches to transition through the following STP states faster. When two switches first connect, they establish a bidirectional handshake across the shared link to identify the root bridge. This is straightforward for an environment with only two switches; however, large environments require greater care to avoid creating a forwarding loop. RSTP uses a synchronization process to add a switch to the RSTP topology without introducing a forwarding loop. The synchronization process starts when two switches (such as SW1 and SW2) are first connected. The process proceeds as follows:

1. As the first two switches connect to each other, they verify that they are connected with a point-to-point link by checking the full-duplex status.
2. They establish a handshake with each other to advertise a proposal (in configuration BPDUs) that their interface should be the DP for that port.
3. There can be only one DP per segment, so each switch identifies whether it is the superior or inferior switch, using the same logic as in 802.1D for the system identifier (that is, the lowest priority and then the lowest MAC address). Using the MAC addresses from Figure 2-1, SW1 (0062.ec9d.c500) is the superior switch to SW2 (0081.c4ff.8b00).
4. The inferior switch (SW2) recognizes that it is inferior and marks its local port (Gi1/0/1) as the RP. At that same time, it moves all non-edge ports to a discarding state. At this point in time, the switch has stopped all local switching for non-edge ports.
5. The inferior switch (SW2) sends an agreement (configuration BPDU) to the root bridge (SW1), which signifies to the root bridge that synchronization is occurring on that switch.
6. The inferior switch (SW2) moves its RP (Gi1/0/1) to a forwarding state. The superior switch moves its DP (Gi1/0/2) to a forwarding state, too.
7. The inferior switch (SW2) repeats the process for any downstream switches connected to it.

The RSTP convergence process can occur quickly, but if a downstream switch fails to acknowledge the proposal, the RSTP switch must default to 802.1D behaviors to prevent a forwarding loop.

# Chapter 3

## Root Bridge Placement

Ideally the root bridge is placed on a core switch, and a secondary root bridge is designated to minimize changes to the overall spanning tree. Root bridge placement is accomplished by lowering the system priority on the root bridge to the lowest value possible, raising the secondary root bridge to a value slightly higher than that of the root bridge, and (ideally) increasing the system priority on all other switches. This ensures consistent placement of the root bridge. The priority is set with either of the following commands:

- **spanning-tree vlan** *vlan-id* **priority** *priority*: The priority is a value between 0 and 61,440, in increments of 4,096.
- **spanning-tree vlan** *vlan-id* **root {primary | secondary} [diameter** *diameter***]:** This command executes a script that modifies certain values. The **primary** keyword sets the priority to 24,576, and the **secondary** keyword sets the priority to 28,672.

The best way to prevent erroneous devices from taking over the STP root role is to set the priority to 0 for the primary root switch and to 4096 for the secondary root switch. In addition, root guard should be used.

By changing the STP port costs with the command **spanning tree** [**vlan** *vlan-id*] **cost** *cost*, you can modify the STP forwarding path. You can lower a path that is currently an alternate port while making it designated, or you can raise the cost on a port that is designated to turn it into a blocking port. The **spanning tree** command modifies the cost for all VLANs unless the optional **vlan** keyword is used to specify a VLAN.

## Root Guard

Root guard is an STP feature that is enabled on a port-by-port basis; it prevents a configured port from becoming a root port. Root guard prevents a downstream switch (often misconfigured or rogue) from becoming a root bridge in a topology. Root guard functions by placing a port in an ErrDisabled state if a superior BPDU is received on a configured port. This prevents the configured DP with root guard from becoming an RP.

Root guard is enabled with the interface command **spanning-tree guard root**. Root guard is placed on designated ports toward other switches that should never become root bridges.

In the sample topology shown in Figure 3-1, root guard should be placed on SW2's Gi1/0/4 port toward SW4 and on SW3's Gi1/0/5 port toward SW5. This prevents SW4 and SW5 from ever becoming root bridges but still allows for SW2 to maintain connectivity to SW1 via SW3 if the link connecting SW1 to SW2 fails.

## STP Portfast

The generation of TCN for hosts does not make sense as a host generally has only one connection to the network. Restricting TCN creation to only ports that connect with other switches and network devices increases the L2 network's stability and efficiency. The STP portfast feature disables TCN generation for access ports.

Another major benefit of the STP portfast feature is that the access ports bypass the earlier 802.1D STP states (learning and listening) and forward traffic immediately. This is beneficial in environments where computers use Dynamic Host Configuration Protocol (DHCP) or Preboot Execution Environment (PXE). If a BPDU is received on a portfast-enabled port, the portfast functionality is removed from that port.



The portfast feature is enabled on a specific access port with the command **spanning-tree portfast** or globally on all access ports with the command **spanning-tree portfast default**. If portfast needs to be disabled on a specific port when using the global configuration, you can use the interface configuration command **spanning-tree portfast disable** to remove portfast on that port.

Portfast can be enabled on trunk links with the command **spanning-tree portfast trunk**. However, this command should be used only with ports that are connecting to a single host (such as a server with only one NIC that is running a hypervisor with VMs on different VLANs). Running this command on interfaces connected to other switches, bridges, and so on can result in a bridging loop.

### BPDU Guard

BPDU guard is a safety mechanism that shuts down ports configured with STP portfast upon receipt of a BPDU. Assuming that all access ports have portfast enabled, this ensures that a loop cannot accidentally be created if an unauthorized switch is added to a topology.

BPDU guard is enabled globally on all STP portfast ports with the command **spanning-tree portfast bpduguard default**. BPDU guard can be enabled or disabled on a specific interface with the command **spanning-tree bpduguard** {**enable** | **disable**}.

## BPDU Filter

BPDU filter simply blocks BPDUs from being transmitted out a port. BPDU filter can be enabled globally or on a specific interface. The behavior changes depending on the configuration:

- The global BPDU filter configuration uses the command **spanning-tree portfast bpdufilter default**, and the port sends a series of 10 to 12 BPDUs. If the switch receives any BPDUs, it checks to identify which switch is more preferred.
  - The preferred switch does not process any BPDUs that it receives, but it still transmits BPDUs to inferior downstream switches.
  - A switch that is not the preferred switch processes BPDUs that are received, but it does not transmit BPDUs to the superior upstream switch.
- The interface-specific BPDU filter is enabled with the interface configuration command **spanning-tree bpdufilter enable**. The port does not send any BPDUs on an ongoing basis. If the remote port has BPDU guard on it, that generally shuts down the port as a loop prevention mechanism.

# Chapter 4

## Multiple Spanning Tree Protocol

The original 802.1D standard, much like the 802.1Q standard, supported only one STP instance for an entire switch network. In this situation, referred to as *Common Spanning Tree (CST)*, all VLANs used the same topology, which meant it was not possible to load share traffic across links by blocking for specific VLANs on one link and then blocking for other VLANs on alternate links.

Now, in environments with thousands of VLANs, maintaining an STP state for all the VLANs can become a burden to the switch's processors. The switches must process BPDUs for every VLAN, and when a major trunk link fails, they must compute multiple STP operations to converge the network. MST provides a blended approach by mapping one or multiple VLANs onto a single STP tree, called an *MST instance (MSTI)*.

A grouping of MST switches with the same high-level configuration is known as an *MST region*. MST incorporates mechanisms that make an MST region appear as a single virtual switch to external switches as part of a compatibility mechanism.

MST uses a special STP instance called the *internal spanning tree (IST)*, which is always the first instance, instance 0. The IST runs on all switch port interfaces for switches in the MST region, regardless of the VLANs associated with the ports. Additional information about other MSTIs is included (nested) in the IST BPDU that is transmitted throughout the MST region. This enables the MST to advertise only one set of BPDUs, minimizing STP traffic regardless of the number of instances while providing the necessary information to calculate the STP for other MSTIs.

## MST Region Boundary

The topology for all the MST instances is contained within the IST, which operates internally to the MST region. An *MST region boundary* is any port that connects to a switch that is in a different MST region or that connects to 802.1D or 802.1W BPDUs.

MSTIs never interact outside the region. MST switches can detect PVST+ neighbors at MST region boundaries. Propagating the CST (derived from the IST) at the MST region boundary involves a feature called the *PVST simulation mechanism*.

The PVST simulation mechanism sends out PVST+ (and includes RSTP, too) BPDUs (one for each VLAN), using the information from the IST. To be very explicit, this requires a mapping of one topology (IST) to multiple VLANs (VLANs toward the PVST link). The PVST simulation mechanism is required because PVST+/RSTP topologies do not understand the IST BPDU structure.

When the MST boundary receives PVST+ BPDUs, it does not map the VLANs to the appropriate MSTIs. Instead, the MST boundary maps the PVST+ BPDU from VLAN 1 to the IST instance. The MST boundary engages the PVST simulation mechanism only when it receives a PVST BPDU on a port.

There are two design considerations when integrating an MST region with a PVST+/RSTP environment: The MST region is the root bridge or the MST region is not a root bridge for any VLAN.

# Chapter 5

## VLAN Trunking Protocol

Before APIs were available on Cisco platforms, configuring a switch was a manual process. Cisco created the proprietary protocol, VLAN Trunking Protocol (VTP), to reduce the burden of provisioning VLANs on switches. Adding a VLAN might seem like a simple task, but in an environment with 100 switches, adding a VLAN required logging in to 100 switches to provision one VLAN. Thanks to VTP, switches that participate in the same VTP domain can have a VLAN created once on a VTP server and propagated to other VTP client switches in the same VTP domain.

There are four roles in the VTP architecture:

- **Server:** The server switch is responsible for the creation, modification, and deletion of VLANs within the VTP domain.
- **Client:** The client switch receives VTP advertisements and modifies the VLANs on that switch. VLANs cannot be configured locally on a VTP client.
- **Transparent:** VTP transparent switches receive and forward VTP advertisements but do not modify the local VLAN database. VLANs are configured only locally.
- **Off:** A switch does not participate in VTP advertisements and does not forward them out of any ports either. VLANs are configured only locally.

It is very important that every switch that connects to a VTP domain has the VTP revision number reset to 0. Failing to reset the revision number on a switch could result in the switch providing an update to the VTP server. This is not an issue if VLANs are added but is catastrophic if VLANs are removed because those VLANs will be removed throughout the domain.

Dynamic trunk ports are established by the switch port sending Dynamic Trunking Protocol (DTP) packets to negotiate whether the other end can be a trunk port. If both ports can successfully negotiate an agreement, the port will become a trunk switch port. DTP advertises itself every 30 seconds to neighbors so that they are kept aware of its status. DTP requires that the VTP domain match between the two switches.

A static trunk port attempts to establish and negotiate a trunk port with a neighbor by default. However, the interface configuration command **switchport nonegotiate** prevents that port from forming a trunk port with a dynamic desirable or dynamic auto switch port. Example 5-7 demonstrates the use of this command on SW1's Gi1/0/2 interface. The setting is then verified by looking at the switch port status. Notice that Negotiation of Trunk now displays as Off.

### PAgP Port Modes

PAgP advertises messages with the multicast MAC address 0100:0CCC:CCCC and the protocol code 0x0104. PAgP can operate in two modes:

- **Auto:** In this PAgP mode, the interface does not initiate an EtherChannel to be established and does not transmit PAgP packets out of it. If an PAgP packet is received from the remote switch, this interface responds and then can establish a PAgP adjacency. If both devices are PAgP auto, a PAgP adjacency does not form.
- **Desirable:** In this PAgP mode, an interface tries to establish an EtherChannel and transmit PAgP packets out of it. Active PAgP interfaces can establish a PAgP adjacency only if the remote interface is configured to auto or desirable.

### LACP Port Modes

LACP advertises messages with the multicast MAC address 0180:C200:0002. LACP can operate in two modes:

- **Passive:** In this LACP mode, an interface does not initiate an EtherChannel to be established and does not transmit LACP packets out of it. If an LACP packet is received from the remote switch, this interface responds and then can establish an LACP adjacency. If both devices are LACP passive, an LACP adjacency does not form.
- **Active:** In this LACP mode, an interface tries to establish an EtherChannel and transmit LACP packets out of it. Active LACP interfaces can establish an LACP adjacency only if the remote interface is configured to active or passive.

### EtherChannel Configuration

It is possible to configure EtherChannels by going into the interface configuration mode for the member interfaces and assigning them to an EtherChannel ID and configuring the appropriate mode:

- **Static EtherChannel:** A static EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode on.**
- **LACP EtherChannel:** An LACP EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode** {**active** | **passive**}.
- **PAgP EtherChannel:** A PAgP EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode** {**auto** | **desirable**} [**non-silent**].

### Minimum Number of Port-Channel Member Interfaces

An EtherChannel interface becomes active and up when only one member interface successfully forms an adjacency with a remote device. In some design scenarios using LACP, a minimum number of adjacencies is required before a port-channel interface becomes active. This option can be configured with the port-channel interface command **port-channel min-links** *min-links*.

### Maximum Number of Port-Channel Member Interfaces

An EtherChannel can be configured to have a specific maximum number of member interfaces in a port channel. This may be done to ensure that the active member interface count proceeds with powers of two (for example, 2, 4, 8) to accommodate load-balancing hashes. The maximum number of member interfaces in a port channel can be configured with the port-channel interface command **lacp max-bundle** *max-links*.

### LACP System Priority

The *LACP system priority* identifies which switch is the master switch for a port channel. The master switch on a port channel is responsible for choosing which member interfaces are active in a port channel when there are more member interfaces than the maximum number of member interfaces associated with a port-channel interface. The switch with the lower system priority is preferred. The LACP system priority can be changed with the command **lacp system-priority** *priority*.

### LACP Interface Priority

*LACP interface priority* enables the master switch to choose which member interfaces are active in a port channel when there are more member interfaces than the maximum number of member interfaces for a port channel. A port with a lower port priority is preferred. The interface configuration command **lacp port-priority** *priority* sets the interface priority.

### Troubleshooting EtherChannel Bundles

It is important to remember that a port channel is a logical interface, so all the member interfaces must have the same characteristics. If they do not, problems will occur.

As a general rule, when configuring port channels on a switch, place each member interface in the appropriate switch port type (Layer 2 or Layer 3) and then associate the interfaces to a port channel. All other port-channel configuration is done via the port-channel interface.

## Load Balancing Traffic with EtherChannel Bundles

Traffic that flows across a port-channel interface is not forwarded out member links on a round-robin basis per packet. Instead, a hash is calculated, and packets are consistently forwarded across a link based on that hash, which runs on the various packet header fields. The load-balancing hash is a systemwide configuration that uses the global command **port-channel load-balance** *hash*. The *hash* option has the following keyword choices:

- **dst-ip:** Destination IP address
- **dst-mac:** Destination MAC address
- **dst-mixed-ip-port:** Destination IP address and destination TCP/UDP port
- **dst-port:** Destination TCP/UDP port
- **src-dst-ip:** Source and destination IP addresses
- **src-dest-ip-only:** Source and destination IP addresses only
- **src-dst-mac:** Source and destination MAC addresses
- **src-dst-mixed-ip-port:** Source and destination IP addresses and source and destination TCP/UDP ports
- **src-dst-port:** Source and destination TCP/UDP ports only
- **src-ip:** Source IP address
- **src-mac:** Source MAC address
- **src-mixed-ip-port:** Source IP address and source TCP/UDP port
- **src-port:** Source TCP/UDP port

# Chapter 6

## Distance Vector Algorithms

Distance vector routing protocols, such as RIP, advertise routes as vectors, where distance is a metric (or cost) such as hop count, and vector is the next-hop router's IP used to reach the destination:

- **Distance:** The distance is the route metric to reach the network.
- **Vector:** The vector is the interface or direction to reach the network.

Routers running distance vector protocols advertise the routing information to their neighbors from their own perspective, modified from the original route received. Therefore, a distance vector protocol does not have a complete map of the whole network; instead, its database reflects that a neighbor router knows how to reach the destination network and how far the neighbor router is from the destination network. The advantage of distance vector protocols is that they require less CPU and memory and can run on low-end routers.

## Enhanced Distance Vector Algorithms

The diffusing update algorithm (DUAL) is an enhanced distance vector algorithm that EIGRP uses to calculate the shortest path to a destination within a network. EIGRP advertises network information to its neighbors as other distance vector protocols do, but it has some enhancements, as its name suggests. The following are some of the enhancements introduced into this algorithm compared to other distance vector algorithms:

- It offers rapid convergence time for changes in the network topology.
- It sends updates only when there is a change in the network. It does not send full routing table updates in a periodic fashion, as distance vector protocols do.
- It uses hellos and forms neighbor relationships just as link-state protocols do.
- It uses bandwidth, delay, reliability, load, and maximum transmission unit (MTU) size instead of hop count for path calculations.
- It has the option to load balance traffic across equal- or unequal-cost paths.

EIGRP is sometimes referred to as a *hybrid routing protocol* because it has characteristics of both distance vector and link-state protocols, as shown in the preceding list. EIGRP relies on more advanced metrics other than hop count (for example, bandwidth) for its best-path calculations. By default, EIGRP advertises the total path delay and minimum bandwidth for a route. This information is advertised out every direction, as happens with a distance vector routing protocol; however, each router can calculate the best path based on the information provided by its direct neighbors.

### Link-State Algorithms

A link-state dynamic IP routing protocol advertises the link state and link metric for each of its connected links and directly connected routers to every router in the network. OSPF and IS-IS are two link-state routing protocols commonly used in enterprise and service provider networks. OSPF advertisements are called *link-state advertisements (LSAs)*, and IS-IS uses *link-state packets (LSPs)* for its advertisements.

### Path Vector Algorithm

A path vector protocol such as BGP is similar to a distance vector protocol; the difference is that instead of looking at the distance to determine the best loop-free path, it looks at various BGP path attributes. BGP path attributes include autonomous system path (AS_Path), multi-exit discriminator (MED), origin, next hop, local preference, atomic aggregate, and aggregator. BGP path attributes are covered in Chapter 11, "BGP," and Chapter 12, "Advanced BGP."

A path vector protocol guarantees loop-free paths by keeping a record of each autonomous system that the routing advertisement traverses. Any time a router receives an advertisement in which it is already part of the AS_Path, the advertisement is rejected because accepting the AS_Path would effectively result in a routing loop.

## Path Selection

A router identifies the path a packet should take by evaluating the prefix length that is programmed in the *Forwarding Information Base (FIB)*. The FIB is programmed through the routing table, which is also known as the *Routing Information Base (RIB)*. The RIB is composed of routes presented from the routing protocol processes. Path selection has three main components:

- **Prefix length:** The prefix length represents the number of leading binary bits in the subnet mask that are in the on position.
- **Administrative distance:** Administrative distance (AD) is a rating of the trustworthiness of a routing information source. If a router learns about a route to a destination from more than one routing protocol, and all the routes have the same prefix length, then the AD is compared.
- **Metrics:** A metric is a unit of measure used by a routing protocol in the best-path calculation. The metrics vary from one routing protocol to another.

If a packet needs to be forwarded, the route chosen depends on the prefix length, where the longest prefix length is always preferred. For example, /28 is preferred over /26, and /26 is preferred over /24. The following is an example, using Table 6-2 as a reference:

- If a packet needs to be forwarded to 10.0.3.14, the router matches all three routes as it fits into all three IP address ranges. But the packet is forwarded to next hop 10.1.1.1 with the outgoing interface Gigabit Ethernet 1/1 because 10.0.3.0/28 has the longest prefix match.
- If a packet needs to be forwarded to 10.0.3.42, the router matches the 10.0.3.0/24 and 10.0.3.0/26 prefixes. But the packet is forwarded to 10.2.2.2 with the outgoing interface Gigabit Ethernet 2/2 because 10.0.3.0/26 has the longest prefix match.
- If a packet needs to be forwarded to 10.0.3.100, the router matches only the 10.0.3.0/24 prefix. The packet is forwarded to 10.3.3.3 with the outgoing interface Gigabit Ethernet 3/3.

The forwarding decision is a function of the FIB and results from the calculations performed in the RIB. The RIB is calculated through the combination of routing protocol metrics and administrative distance.

The RIB is programmed from the various routing protocol processes. Every routing protocol presents the same information to the RIB for insertion: the destination network, the next-hop IP address, the AD, and metric values. The RIB accepts or rejects a route based on the following logic:

- If the route does not exist in the RIB, the route is accepted.
- If the route exists in the RIB, the AD must be compared. If the AD of the route already in the RIB is lower than the process submitting the second route, the route is rejected. Then that routing process is notified.
- If the route exists in the RIB, the AD must be compared. If the AD of the route already in the RIB is higher than the routing process submitting the alternate entry, the route is accepted, and the current source protocol is notified of the removal of the entry from the RIB.

Understanding the order of processing from a router is critical because in some scenarios the path with the lowest AD may not always be installed in the RIB. For example, BGP's path selection process could choose an iBGP path over an eBGP path. So BGP would present the path with an AD of 200, not 20, to the RIB, which might not preempt a route learned via OSPF that has an AD of 110. These situations are almost never seen; but remember that it is the best route from the routing protocol presented to the RIB when AD is then compared.

### Equal-Cost Multipathing

If a routing protocol identifies multiple paths as a best path and supports multiple path entries, the router installs the maximum number of paths allowed per destination. This is known as *equal-cost multipathing (ECMP)* and provides load sharing across all links. RIP, EIGRP, OSPF, and IS-IS all support ECMP. ECMP provides a mechanism to increase bandwidth across multiple paths by splitting traffic equally across the links.

### Unequal-Cost Load Balancing

By default, routing protocols install only routes with the lowest path metric. However, EIGRP can be configured (not enabled by default) to install multiple routes with different path metrics. This allows for unequal-cost load balancing across multiple paths. Traffic is transmitted out the router's interfaces based on that path's metrics in ratio to other the interface's metrics.

### Directly Attached Static Routes

Point-to-point (P2P) serial interfaces do not have to worry about maintaining an adjacency table and do not use Address Resolution Protocol (ARP), so static routes can directly reference the outbound interface of a router. A static route that uses only the outbound next-hop interface is known as a *directly attached static route*, and it requires that the outbound interface be in an up state for the route to be installed into the RIB.

Directly attached static routes are configured with the command **ip route** *network subnet-mask next-hop-interface-id*.

### Recursive Static Routes

The forwarding engine on Cisco devices needs to know which interface an outbound packet should use. A *recursive static route* specifies the IP address of the next-hop address. The recursive lookup occurs when the router queries the RIB to locate the route toward the next-hop IP address (connected, static, or dynamic) and then cross-references the adjacency table.

Recursive static routes are configured with the command **ip route** *network subnet-mask next-hop-ip*. Recursive static routes require the route's next-hop address to exist in the routing table to install the static route into the RIB. A recursive static route may not resolve the next-hop forwarding address using the default route (0.0.0.0/0) entry. The static route will fail next-hop reachability requirements and will not be inserted into the RIB.

### Fully Specified Static Routes

Static route recursion can simplify topologies if a link fails because it may allow the static route to stay installed while it changes to a different outbound interface in the same direction as the destination. However, problems arise if the recursive lookup resolves to a different interface pointed in the opposite direction.

To correct this issue, the static route configuration should use the outbound interface and the next-hop IP address. A static route with both an interface and a next-hop IP address is known as a *fully specified static route*. If the interface listed is not in an up state, the router removes the static route from the RIB. Specifying the next-hop address along with the physical interface removes the recursive lookup and does not involve the ARP processing problems that occur when using only the outbound interface.

Fully specified static routes are configured with the command **ip route** *network subnet-mask interface-id next-hop-ip*.

### Floating Static Routing

The default AD on a static route is 1, but a static route can be configured with an AD value of 1 to 255 for a specific route. The AD is set on a static route by appending the AD as part of the command structure.

Using a floating static route is a common technique for providing backup connectivity for prefixes learned via dynamic routing protocols. A floating static route is configured with an AD higher than that of the primary route. Because the AD is higher than that of the primary route, it is installed in the RIB only when the primary route is withdrawn.

## Static Null Routes

The null interface is a virtual interface that is always in an up state. Null interfaces do not forward or receive network traffic and drop all traffic destined toward them without adding overhead to a router's CPU.

Configuring a static route to a null interface provides a method of dropping network traffic without requiring the configuration of an access list. Creating a static route to the Null0 interface is a common technique to prevent routing loops. The static route to the Null0 interface uses a summarized network range, and routes that are more specific point toward the actual destination.

### IPv6 Static Routes

The static routing principles for IPv4 routes are exactly the same for IPv6. It is important to ensure that IPv6 routing is enabled by using the configuration command **ipv6 unicast routing.** IPv6 static routes are configured with the command **ipv6 route** *network/prefix-length* { *next-hop-interface-id* | [*next-hop-interface-id*] *next-ip-address*}.

# Chapter 7

**Table 7-2** EIGRP Terminology

| Term | Definition |
|---|---|
| Successor route | The route with the lowest path metric to reach a destination.<br><br>The successor route for R1 to reach 10.4.4.0/24 on R4 is R1→R3→R4. |
| Successor | The first next-hop router for the successor route.<br><br>The successor for 10.4.4.0/24 is R3. |
| Feasible distance (FD) | The metric value for the lowest-metric path to reach a destination. The feasible distance is calculated locally using the formula shown in the "Path Metric Calculation" section, later in this chapter.<br><br>The FD calculated by R1 for the 10.4.4.0/24 network is 3328 (that is, 256+256+2816). |
| Reported distance (RD) | The distance reported by a router to reach a prefix. The reported distance value is the feasible distance for the advertising router.<br><br>R3 advertises the 10.4.4.0/24 prefix with an RD of 3072.<br><br>R4 advertises the 10.4.4.0/24 to R1 and R2 with an RD of 2816. |
| Feasibility condition | A condition under which, for a route to be considered a backup route, the reported distance received for that route must be less than the feasible distance calculated locally. This logic guarantees a loop-free path. |
| Feasible successor | A route that satisfies the feasibility condition and is maintained as a backup route. The feasibility condition ensures that the backup route is loop free.<br><br>The route R1→R4 is the feasible successor because the RD 2816 is lower than the FD 3328 for the R1→R3→R4 path. |

### Topology Table

EIGRP contains a *topology table* that makes it different from a "true" distance vector routing protocol. EIGRP's topology table is a vital component to DUAL and contains information to identify loop-free backup routes. The topology table contains all the network prefixes advertised within an EIGRP autonomous system. Each entry in the table contains the following:

- Network prefix
- EIGRP neighbors that have advertised that prefix
- Metrics from each neighbor (for example, reported distance, hop count)
- Values used for calculating the metric (for example, load, reliability, total delay, minimum bandwidth)

**Table 7-3** EIGRP Packet Types

| Type | Packet Name | Function |
|---|---|---|
| 1 | Hello | Used for discovery of EIGRP neighbors and for detecting when a neighbor is no longer available |
| 2 | Request | Used to get specific information from one or more neighbors |
| 3 | Update | Used to transmit routing and reachability information with other EIGRP neighbors |
| 4 | Query | Sent out to search for another path during convergence |
| 5 | Reply | Sent in response to a query packet |

Figure 7-7 displays the information in the EIGRP update packets for the 10.1.1.0/24 prefix propagating through the autonomous system. Notice that the hop count increments, minimum bandwidth decreases, total delay increases, and RD changes with each router in the AS.

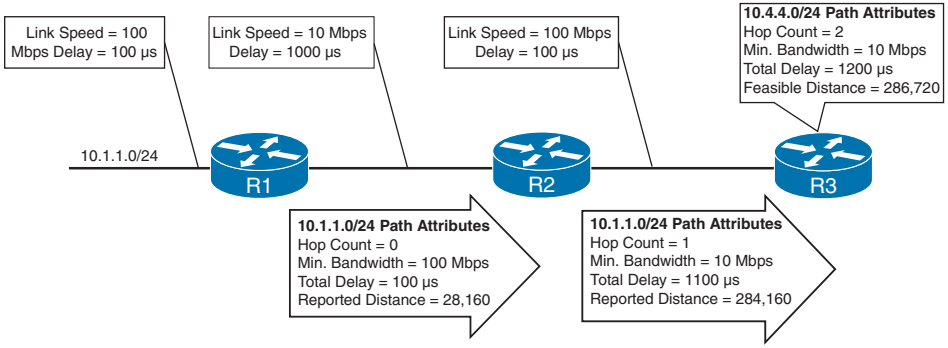



**Figure 7-7** *EIGRP Attribute Propagation*

Figure 7-9 shows the explicit EIGRP wide metrics formula. Notice that an additional K value ($K_6$) is included that adds an extended attribute to measure jitter, energy, or other future attributes.

$$\text{Wide Metric} = \left[\left(K_1 * BW + \frac{K_2 * BW}{256 - \text{Load}} + K_3 * \text{Latency} + K_6 * \text{Extended}\right) * \frac{K_5}{K_4 + \text{Reliability}}\right]$$

**Figure 7-9** *EIGRP Wide Metrics Formula*

EIGRP supports unequal-cost load balancing, which allows installation of both successor routes and feasible successors into the EIGRP RIB. EIGRP supports unequal-cost load balancing by changing EIGRP's *variance multiplier*. The EIGRP *variance value* is the feasible distance (FD) for a route multiplied by the EIGRP variance multiplier. Any feasible successor's FD with a metric below the EIGRP variance value is installed into the RIB. EIGRP installs multiple routes where the FD for the routes is less than the EIGRP multiplier value up to the maximum number of ECMP routes, as discussed earlier.

## Convergence

When a link fails, and the interface protocol moves to a down state, any neighbor attached to that interface moves to a down state, too. When an EIGRP neighbor moves to a down state, path recomputation must occur for any prefix where that EIGRP neighbor was a successor (upstream router).

When EIGRP detects that it has lost its successor for a path, the feasible successor instantly becomes the successor route, providing a backup route. The router sends out an update packet for that path because of the new EIGRP path metrics. Downstream routers run their own DUAL for any impacted prefixes to account for the new EIGRP metrics. It is possible that a change of the successor route or feasible successor may occur upon receipt of new EIGRP metrics from a successor router for a prefix. Figure 7-13 demonstrates such a scenario when the link between R1 and R3 fails.

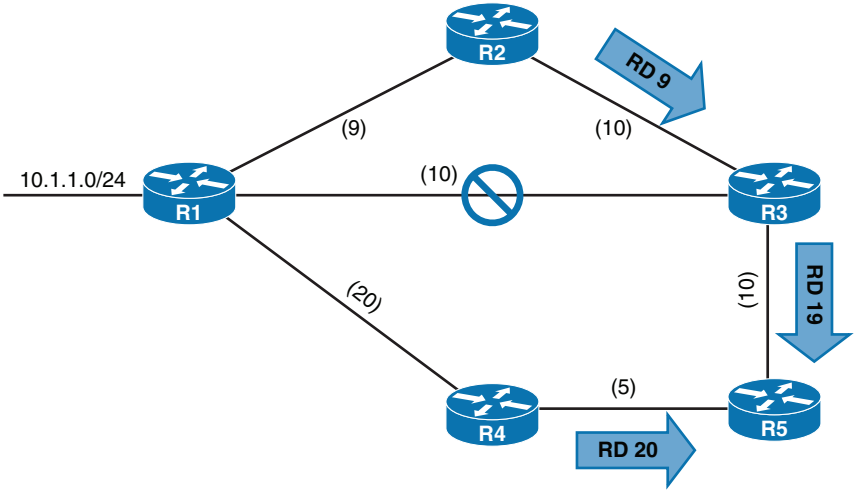



**Figure 7-13** *EIGRP Topology with Link Failure*

R3 installs the feasible successor path advertised from R2 as the successor route. R3 sends an update packet with a new RD of 19 for the 10.1.1.0/24 prefix. R5 receives the update packet from R3 and calculates an FD of 29 for the R1–R2–R3 path to 10.1.1.0/24. R5 compares that path to the one received from R4, which has a path metric of 25. R5 chooses the path via R4 as the successor route.

If a feasible successor is not available for a prefix, DUAL must perform a new route calculation. The route state changes from passive (P) to active (A) in the EIGRP topology table.

# Chapter 8

OSPF provides scalability for the routing table by using multiple OSPF areas within the routing domain. Each OSPF area provides a collection of connected networks and hosts that are grouped together. OSPF uses a two-tier hierarchical architecture, where Area 0 is a special area known as the backbone, to which all other areas must connect. In other words, Area 0 provides transit connectivity between nonbackbone areas. Nonbackbone areas advertise routes into the backbone, and the backbone then advertises routes into other nonbackbone areas.

## Inter-Router Communication

OSPF runs directly over IPv4, using its own protocol 89, which is reserved for OSPF by the Internet Assigned Numbers Authority (IANA). OSPF uses multicast where possible to reduce unnecessary traffic. The two OSPF multicast addresses are as follows:

- **AllSPFRouters:** IPv4 address 224.0.0.5 or MAC address 01:00:5E:00:00:05. All routers running OSPF should be able to receive these packets.
- **AllDRouters:** IPv4 address 224.0.0.6 or MAC address 01:00:5E:00:00:06. Communication with designated routers (DRs) uses this address.

**Table 8-2** OSPF Packet Types

| Type | Packet Name | Functional Overview |
|---|---|---|
| 1 | Hello | These packets are for discovering and maintaining neighbors. Packets are sent out periodically on all OSPF interfaces to discover new neighbors while ensuring that other adjacent neighbors are still online. |
| 2 | Database description (DBD) or (DDP) | These packets are for summarizing database contents. Packets are exchanged when an OSPF adjacency is first being formed. These packets are used to describe the contents of the LSDB. |
| 3 | Link-state request (LSR) | These packets are for database downloads. When a router thinks that part of its LSDB is stale, it may request a portion of a neighbor's database by using this packet type. |
| 4 | Link-state update (LSU) | These packets are for database updates. This is an explicit LSA for a specific network link and normally is sent in direct response to an LSR. |
| 5 | Link-state ack | These packets are for flooding acknowledgments. These packets are sent in response to the flooding of LSAs, thus making flooding a reliable transport feature. |

**Table 8-4** OSPF Neighbor States

| State | Description |
|---|---|
| Down | This is the initial state of a neighbor relationship. It indicates that the router has not received any OSPF hello packets. |
| Attempt | This state is relevant to NBMA networks that do not support broadcast and require explicit neighbor configuration. This state indicates that no information has been received recently, but the router is still attempting communication. |
| Init | This state indicates that a hello packet has been received from another router, but bidirectional communication has not been established. |
| 2-Way | Bidirectional communication has been established. If a DR or BDR is needed, the election occurs during this state. |
| ExStart | This is the first state in forming an adjacency. Routers identify which router will be the master or slave for the LSDB synchronization. |
| Exchange | During this state, routers are exchanging link states by using DBD packets. |
| Loading | LSR packets are sent to the neighbor, asking for the more recent LSAs that have been discovered (but not received) in the Exchange state. |
| Full | Neighboring routers are fully adjacent. |

OSPF overcomes this inefficiency by creating a pseudonode (virtual router) to manage the adjacency state with all the other routers on that broadcast network segment. A router on the broadcast segment, known as the *designated router (DR)*, assumes the role of the pseudonode. The DR reduces the number of OSPF adjacencies on a multi-access network segment because routers only form a full OSPF adjacency with the DR and not each other. The DR is responsible for flooding updates to all OSPF routers on that segment as the updates occur.

### OSPF Network Statement

The OSPF network statement identifies the interfaces that the OSPF process will use and the area that those interfaces participate in. The network statements match against the primary IPv4 address and netmask associated with an interface.

### Interface-Specific Configuration

The second method for enabling OSPF on an interface for IOS is to configure it specifically on an interface with the command **ip ospf** *process-id* **area** *area-id* [**secondaries none**]. This method also adds secondary connected networks to the LSDB unless the **secondaries none** option is used.

This method provides explicit control for enabling OSPF; however, the configuration is not centralized and increases in complexity as the number of interfaces on the routers increases. If a hybrid configuration exists on a router, interface-specific settings take precedence over the network statement with the assignment of the areas.

### Passive Interfaces

Enabling an interface with OSPF is the quickest way to advertise a network segment to other OSPF routers. However, it might be easy for someone to plug in an unauthorized OSPF router on an OSPF-enabled network segment and introduce false routes, thus causing havoc in the network. Making the network interface passive still adds the network segment into the LSDB but prohibits the interface from forming OSPF adjacencies. A *passive interface* does not send out OSPF hellos and does not process any received OSPF packets.

The command **passive** *interface-id* under the OSPF process makes the interface passive, and the command **passive interface default** makes all interfaces passive. To allow for an interface to process OSPF packets, the command **no passive** *interface-id* is used.

### Requirements for Neighbor Adjacency

The following list of requirements must be met for an OSPF neighborship to be formed:

- RIDs must be unique between the two devices. They should be unique for the entire OSPF routing domain to prevent errors.
- The interfaces must share a common subnet. OSPF uses the interface's primary IP address when sending out OSPF hellos. The network mask (netmask) in the hello packet is used to extract the network ID of the hello packet.
- The MTUs (maximum transmission units) on the interfaces must match. The OSPF protocol does not support fragmentation, so the MTUs on the interfaces should match.
- The area ID must match for the segment.
- The DR enablement must match for the segment.
- OSPF hello and dead timers must match for the segment.
- Authentication type and credentials (if any) must match for the segment.
- Area type flags must match for the segment (for example, Stub, NSSA). (These are not discussed in this book.)

**Table 8-6** OSPF Interface Columns

| Field | Description |
|---|---|
| Interface | Interfaces with OSPF enabled |
| PID | The OSPF process ID associated with this interface |
| Area | The area that this interface is associated with |
| IP Address/Mask | The IP address and subnet mask for the interface |
| Cost | The cost metric assigned to an interface that is used to calculate a path metric |
| State | The current interface state, which could be DR, BDR, DROTHER, LOOP, or Down |
| Nbrs F | The number of neighbor OSPF routers for a segment that are fully adjacent |
| Nbrs C | The number of neighbor OSPF routers for a segment that have been detected and are in a 2-Way state |

**Table 8-7** OSPF Neighbor State Fields

| Field | Description |
|---|---|
| Neighbor ID | The router ID (RID) of the neighboring router. |
| PRI | The priority for the neighbor's interface, which is used for DR/BDR elections. |
| State | The first field is the neighbor state, as described in Table 8-3.<br><br>The second field is the DR, BDR, or DROTHER role if the interface requires a DR. For non-DR network links, the second field shows just a hyphen (-). |
| Dead Time | The time left until the router is declared unreachable. |
| Address | The primary IP address for the OSPF neighbor. |
| Interface | The local interface to which the OSPF neighbor is attached. |

## Default Route Advertisement

OSPF supports advertising the default route into the OSPF domain. The default route is advertised by using the command **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] underneath the OSPF process.

If a default route does not exist in a routing table, the **always** optional keyword advertises a default route even if a default route does not exist in the RIB. In addition, the route metric can be changed with the **metric** *metric-value* option, and the metric type can be changed with the **metric-type** *type-value* option.

## Link Costs

Interface cost is an essential component of Dijkstra's SPF calculation because the shortest path metric is based on the cumulative interface cost (that is, metric) from the router to the destination. OSPF assigns the OSPF link cost (that is, metric) for an interface by using the formula in Figure 8-9.

$$\text{Cost} = \frac{\text{Reference Bandwidth}}{\text{Interface Bandwidth}}$$

**Figure 8-9** *OSPF Interface Cost Formula*

### Failure Detection

A secondary function of the OSPF hello packets is to ensure that adjacent OSPF neighbors are still healthy and available. OSPF sends hello packets at set intervals, based on the hello timer. OSPF uses a second timer called the *OSPF dead interval timer*, which defaults to four times the hello timer. Upon receipt of a hello packet from a neighboring router, the OSPF dead timer resets to the initial value and then starts to decrement again.

If a router does not receive a hello before the OSPF dead interval timer reaches 0, the neighbor state is changed to down. The OSPF router immediately sends out the appropriate LSA, reflecting the topology change, and the SPF algorithm processes on all routers within the area.

### Designated Router Elections

The DR/BDR election occurs during OSPF neighborship—specifically during the last phase of 2-Way neighbor state and just before the ExStart state. When a router enters the 2-Way state, it has already received a hello from the neighbor. If the hello packet includes a RID other than 0.0.0.0 for the DR or BDR, the new router assumes that the current routers are the actual DR and BDR.

Any router with OSPF priority of 1 to 255 on its OSPF interface attempts to become the DR. By default, all OSPF interfaces use a priority of 1. The routers place their RID and OSPF priorities in their OSPF hellos for that segment.

Routers then receive and examine OSPF hellos from neighboring routers. If a router identifies itself as being a more favorable router than the OSPF hellos it receives, it continues to send out hellos with its RID and priority listed. If the hello received is more favorable, the router updates its OSPF hello packet to use the more preferable RID in the DR field. OSPF deems a router more preferable if the priority for the interface is the highest for that segment. If the OSPF priority is the same, the higher RID is more favorable.

Once all the routers have agreed on the same DR, all routers for that segment become adjacent with the DR. Then the election for the BDR takes place. The election follows the same logic for the DR election, except that the DR does not add its RID to the BDR field of the hello packet.

The OSPF DR and BDR roles cannot be preempted after the DR/BDR election. Only upon the failure (or process restart of the DR or BDR) does the election start to replace the role that is missing.

**Table 8-9** OSPF Network Types

| Type | Description | DR/BDR Field in OSPF Hellos | Timers |
|---|---|---|---|
| Broadcast | Default setting on OSPF-enabled Ethernet links | Yes | Hello: 10<br>Wait: 40<br>Dead: 40 |
| Non-broadcast | Default setting on OSPF-enabled Frame Relay main interface or Frame Relay multipoint subinterfaces | Yes | Hello: 30<br>Wait: 120<br>Dead: 120 |
| Point-to-point | Default setting on OSPF-enabled Frame Relay point-to-point subinterfaces. | No | Hello: 10<br>Wait: 40<br>Dead: 40 |
| Point-to-multipoint | Not enabled by default on any interface type. Interface is advertised as a host route (/32) and sets the next-hop address to the outbound interface. Primarily used for hub-and-spoke topologies. | No | Hello: 30<br>Wait: 120<br>Dead: 120 |
| Loopback | Default setting on OSPF-enabled loopback interfaces. Interface is advertised as a host route (/32). | N/A | N/A |

# Chapter 9

Area 0 is a special area called *the backbone*. By design, all areas must connect to Area 0 because OSPF expects all areas to inject routing information into the backbone, and Area 0 advertises the routes into other areas. The backbone design is crucial to preventing routing loops.

*Area border routers (ABRs)* are OSPF routers connected to Area 0 and another OSPF area, per Cisco definition and according to RFC 3509. ABRs are responsible for advertising routes from one area and injecting them into a different OSPF area. Every ABR needs to participate in Area 0; otherwise, routes will not advertise into another area. ABRs compute an SPF tree for every area that they participate in.

### Area ID

The area ID is a 32-bit field and can be formatted in simple decimal (0 through 4,294,967,295) or dotted decimal (0.0.0.0 through 255.255.255.255). During router configuration, the area can use decimal format on one router and dotted-decimal format on a different router, and the routers can still form an adjacency. OSPF advertises the area ID in dotted-decimal format in the OSPF hello packet.

## Link-State Announcements

When OSPF neighbors become adjacent, the LSDBs synchronize between the OSPF routers. As an OSPF router adds or removes a directly connected network link to or from its database, the router floods the link-state advertisement (LSA) out all active OSPF interfaces. The OSPF LSA contains a complete list of networks advertised from that router.

OSPF uses six LSA types for IPv4 routing:

- **Type 1, router LSA:** Advertises the LSAs that originate within an area
- **Type 2, network LSA:** Advertises a multi-access network segment attached to a DR
- **Type 3, summary LSA:** Advertises network prefixes that originated from a different area
- **Type 4, ASBR summary LSA:** Advertises a summary LSA for a specific ASBR
- **Type 5, AS external LSA:** Advertises LSAs for routes that have been redistributed
- **Type 7, NSSA external LSA:** Advertises redistributed routes in NSSAs

LSA types 1, 2, and 3, which are used for building the SPF tree for intra-area and interarea routes, are explained in this section.

Every OSPF router advertises a type 1 LSA. Type 1 LSAs are the essential building blocks within the LSDB. A type 1 LSA entry exists for each OSPF-enabled link (that is, every interface and its attached networks). Figure 9-5 shows that in this example, the type 1 LSAs are not advertised outside Area 1234, which means the underlying topology in an area is invisible to other areas.

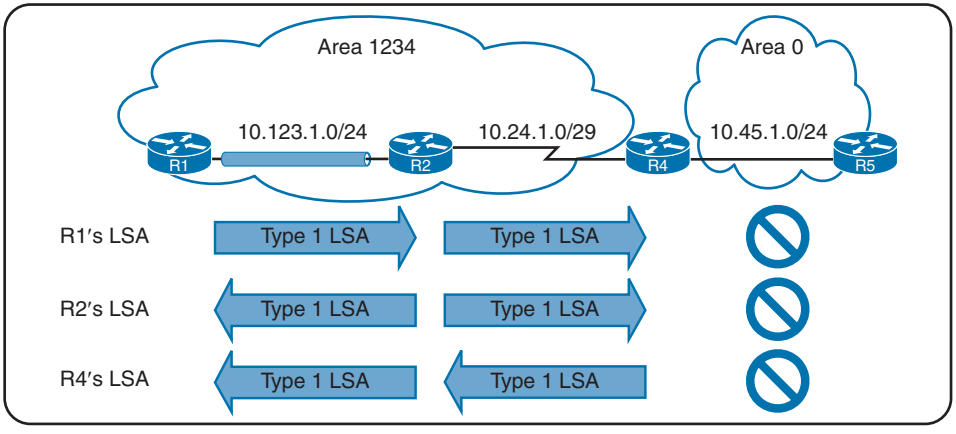



**Figure 9-5** *Type 1 LSA Flooding in an Area*

If we correlate just type 1 LSAs from the sample topology of Figure 9-6, then Figure 9-7 demonstrates the topology built by all routers in Area 1234 using the LSA attributes for Area 1234 from all four routers. Using only type 1 LSAs, a connection is made between R2 and R4 because they point to each other's RID in the point-to-point LSA. Notice that the three networks on R1, R2, and R3 (10.123.1.0) have not been directly connected yet.

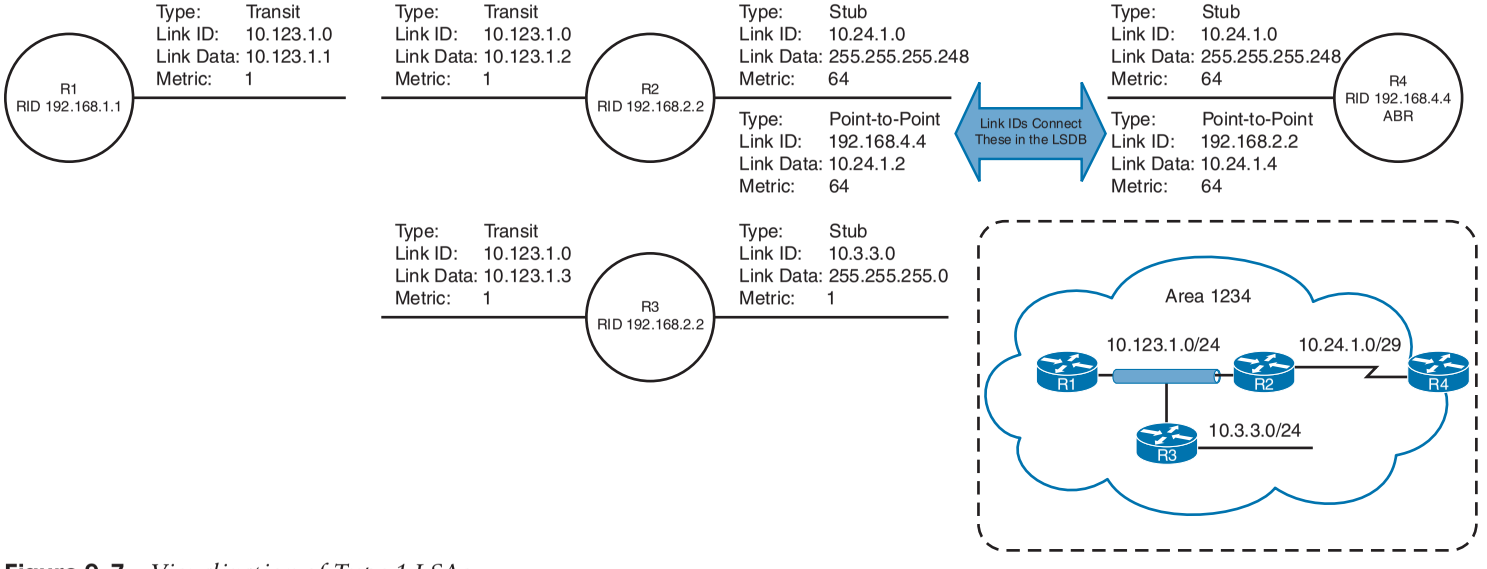



**Figure 9-7** *Visualization of Type 1 LSAs*

### LSA Type 2: Network Link

A type 2 LSA represents a multi-access network segment that uses a DR. The DR always advertises the type 2 LSA and identifies all the routers attached to that network segment. If a DR has not been elected, a type 2 LSA is not present in the LSDB because the corresponding type 1 transit link type LSA is a stub. Like type 1 LSAs, Type 2 LSAs are not flooded outside the originating OSPF area.

Area 1234 has only one DR segment that connects R1, R2, and R3 because R3 has not formed an OSPF adjacency on the 10.3.3.0/24 network segment. On the 10.123.1.0/24 network segment, R3 is elected as the DR, and R2 is elected as the BDR because of the order of the RIDs.

Now that we have the type 2 LSA for Area 1234, all the network links are connected. Figure 9-8 provides a visualization of the type 1 and type 2 LSAs, which correspond with Area 1234 perfectly.

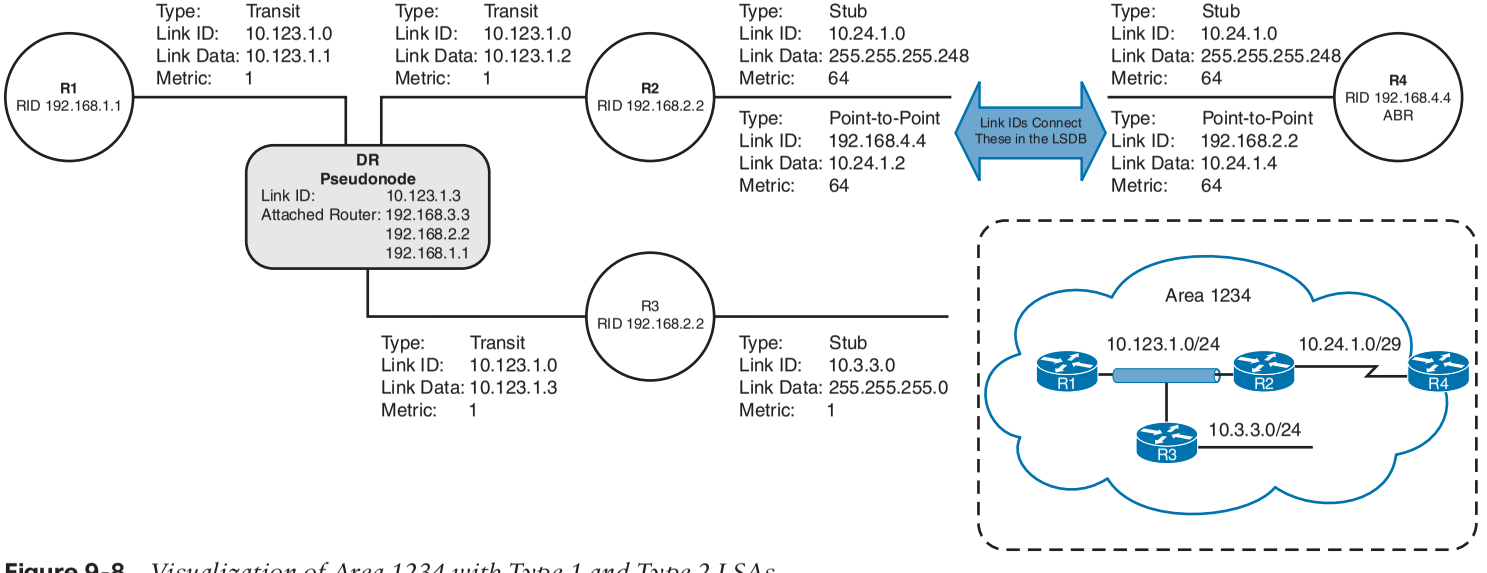



**Figure 9-8** *Visualization of Area 1234 with Type 1 and Type 2 LSAs*

### LSA Type 3: Summary Link

Type 3 LSAs represent networks from other areas. The role of the ABRs is to participate in multiple OSPF areas and ensure that the networks associated with type 1 LSAs are reachable in the non-originating OSPF areas.

As explained earlier, ABRs do not forward type 1 or type 2 LSAs into other areas. When an ABR receives a type 1 LSA, it creates a type 3 LSA referencing the network in the original type 1 LSA; the type 2 LSA is used to determine the network mask of the multi-access network. The ABR then advertises the type 3 LSA into other areas. If an ABR receives a type 3 LSA from Area 0 (the backbone), it regenerates a new type 3 LSA for the nonbackbone area and lists itself as the advertising router, with the additional cost metric.

Figure 9-9 demonstrates the concept of a type 3 LSA interaction with type 1 LSAs.

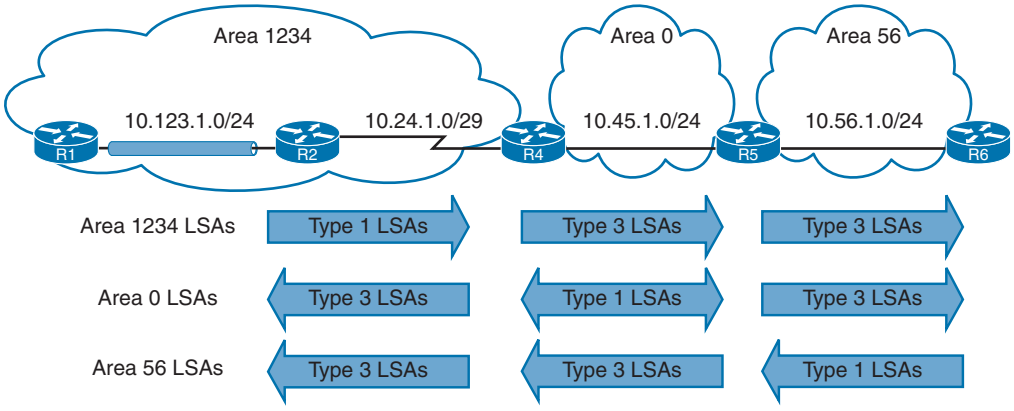



**Figure 9-9** *Type 3 LSA Conceptual Overview*

Most people would assume that the 10.34.1.0/24 route learned by Area 23 would then advertise into R2's Area 0 and then propagate to Area 12. However, they would be wrong. There are three fundamental rules ABRs use the for creating type 3 LSAs:

- Type 1 LSAs received from an area create type 3 LSAs into the backbone area and nonbackbone areas.
- Type 3 LSAs received from Area 0 are created for the nonbackbone area.
- Type 3 LSAs received from a nonbackbone area only insert into the LSDB for the source area. ABRs do not create a type 3 LSA for the other areas (including a segmented Area 0).

## OSPF Path Selection

OSPF executes Dijkstra's shortest path first (SPF) algorithm to create a loop-free topology of shortest paths. All routers use the same logic to calculate the shortest path for each network. Path selection prioritizes paths by using the following logic:

1. Intra-area
2. Interarea
3. External routes (which involves additional logic not covered in this book)

## Summarization of Routes

Route scalability is a large factor for the IGP routing protocols used by service providers because there can be thousands of routers running in a network. Splitting up an OSPF routing domain into multiple areas reduces the size of the LSDB for each area. While the number of routers and networks remains the same within the OSPF routing domain, the detailed type 1 and type 2 LSAs are exchanged for simpler type 3 LSAs.

### Interarea Summarization

Interarea summarization reduces the number of type 3 LSAs that an ABR advertises into an area when it receives type 1 LSAs. The network summarization range is associated with a specific source area for type 1 LSAs.

When a type 1 LSA within the summarization range reaches the ABR from the source area, the ABR creates a type 3 LSA for the summarized network range. The ABR suppresses the more specific type 3 LSAs, thereby preventing the generation of the subordinate route's type 3 LSAs. Interarea summarization does not impact the type 1 LSAs in the source area.

### Configuration of Interarea Summarization

To define the summarization range and associated area, use the command **area** *area-id* **range** *network subnet-mask* [**advertise** | **not-advertise**] [**cost** *metric*] under the OSPF process on the ABR. The default behavior is to advertise the summary prefix, so the keyword **advertise** is not necessary. Appending the **cost** *metric* keyword to the command statically sets the metric on the summary route.

Other network designs require filtering of OSPF routes based on other criteria. OSPF supports filtering when type 3 LSA generation occurs. This allows for the original route to be installed in the LSDB for the source area so that the route can be installed in the RIB of the ABR. Filtering can occur in either direction on the ABR. Figure 9-23 demonstrates the concept.

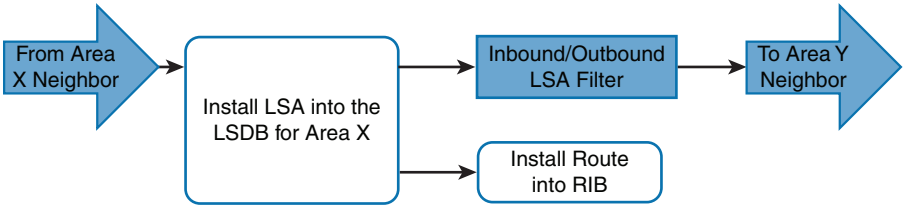



**Figure 9-23** *OSPF Area Filtering*

In some scenarios, routes need to be removed only on specific routers in an area. OSPF is a link-state protocol that requires all routers in the same area to maintain an identical copy of the LSDB for that area. A route can exist in the OSPF LSDB, but it could be prevented from being installed in the local RIB. This is accomplished by using a Distribute List. Figure 9-25 illustrates this concept.

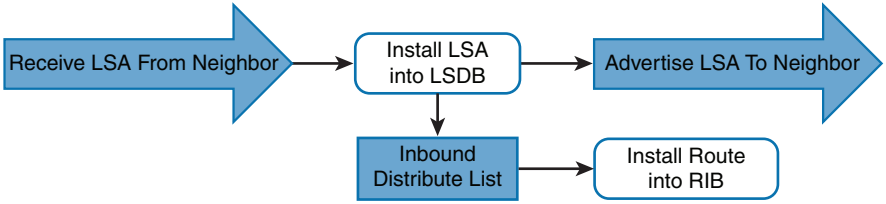



**Figure 9-25** *OSPF Distribute List Filtering Logic*

# Chapter 10

## OSPFv3 Fundamentals

OSPFv3 is different from OSPFv2 in the following ways:

- **Support for multiple address families:** OSPFv3 supports IPv4 and IPv6 address families.
- **New LSA types:** New LSA types have been created to carry IPv6 prefixes.
- **Removal of addressing semantics:** The IP prefix information is no longer present in the OSPF packet headers. Instead, it is carried as LSA payload information, making the protocol essentially address family independent, much like IS-IS. OSPFv3 uses the term *link* instead of *network* because the SPT calculations are per link instead of per subnet.
- **LSA flooding:** OSPFv3 includes a new link-state type field that is used to determine the flooding scope of LSA, as well as the handling of unknown LSA types.
- **Packet format:** OSPFv3 runs directly over IPv6, and the number of fields in the packet header has been reduced.
- **Router ID:** The router ID is used to identify neighbors, regardless of the network type in OSPFv3. When configuring OSPFv3 on IOS routers, the ID must always be manually assigned in the routing process.
- **Authentication:** Neighbor authentication has been removed from the OSPF protocol and is now performed through IPsec extension headers in the IPv6 packet.
- **Neighbor adjacencies:** OSPFv3 inter-router communication is handled by IPv6 link-local addressing. Neighbors are not automatically detected over non-broadcast multiple access (NBMA) interfaces. A neighbor must be manually specified using the link-local address. IPv6 allows for multiple subnets to be assigned to a single interface, and OSPFv3 allows for neighbor adjacency to form even if the two routers do not share a common subnet.
- **Multiple instances:** OSPFv3 packets include an instance ID field that may be used to manipulate which routers on a network segment are allowed to form adjacencies.

**Table 10-3** OSPFv3 Packet Types

| Type | Packet Name | Source | Destination | Purpose |
|---|---|---|---|---|
| 1 | Hello | Link-local address | FF02::5 (all routers) | Discover and maintain neighbors |
| | | Link-local address | Link-local address | Initial adjacency forming, immediate hello |
| 2 | Database description | Link-local address | Link-local address | Summarize database contents |
| 3 | Link-state request | Link-local address | Link-local address | Database information request |
| 4 | Link-state update | Link-local address | Link-local address | Initial adjacency forming, in response to a link-state request |
| | | Link-local address (from DR) | FF02::5 (all routers) | Database update |
| | | Link-local address (from non-DR) | FF02::6 (DR/BDR) | Database update |
| 5 | Link-state acknowledgment | Link-local address | Link-local address | Initial adjacency forming, in response to a link-state update |
| | | Link-local address (from DR) | FF02::5 (all routers) | Flooding acknowledgment |
| | | Link-local address (from non-DR) | FF02::6 (DR/BDR) | Flooding acknowledgment |

## OSPFv3 Verification

The commands for viewing OSPFv3 settings and statuses are very similar to those used in OSPFv2; they essentially replace **ip ospf** with **ospfv3 ipv6**. Supporting OSPFv3 requires verifying the OSPFv3 interfaces, neighborship, and the routing table.

Summarization of internal OSPFv3 routes follows the same rules as in OSPFv2 and must occur on ABRs. In our topology, R3 summarizes the three loopback addresses into the 2001:db8:0:0::/65 network. Summarization involves the command **area** *area-id* **range** *prefix/prefix-length*, which resides under the address family in the OSPFv3 process.

Enabling IPv4 support for OSPFv3 is straightforward:

**Step 1.** Ensure that the IPv4 interface has an IPv6 address (global or link local) configured. Remember that configuring a global address also places a global address; alternatively, a link-local address can statically be configured.

**Step 2.** Enable the OSPFv3 process for IPv4 on the interface with the command **ospfv3** *process-id* **ipv4 area** *area-id*.

# Chapter 11

## Autonomous System Numbers

An organization requiring connectivity to the Internet must obtain an autonomous system number (ASN). ASNs were originally 2 bytes (16-bit range), which made 65,535 ASNs possible. Due to exhaustion, RFC 4893 expanded the ASN field to accommodate 4 bytes (32-bit range). This allows for 4,294,967,295 unique ASNs, providing quite an increase from the original 65,535 ASNs.

Two blocks of private ASNs are available for any organization to use, as long as they are never exchanged publicly on the Internet. ASNs 64,512–65,535 are private ASNs in the 16-bit ASN range, and 4,200,000,000–4,294,967,294 are private ASNs within the extended 32-bit range.

The *Internet Assigned Numbers Authority (IANA)* is responsible for assigning all public ASNs to ensure that they are globally unique. IANA requires the following items when requesting a public ASN:

- Proof of a publicly allocated network range
- Proof that Internet connectivity is provided through multiple connections
- Need for a unique routing policy from providers

In the event that an organization cannot provide this information, it should use the ASN provided by its service provider.

## Path Attributes

BGP uses path attributes (PAs) associated with each network path. The PAs provide BGP with granularity and control of routing policies within BGP. The BGP prefix PAs are classified as follows:

- Well-known mandatory
- Well-known discretionary
- Optional transitive
- Optional non-transitive

Per RFC 4271, well-known attributes must be recognized by all BGP implementations. Well-known mandatory attributes must be included with every prefix advertisement; well-known discretionary attributes may or may not be included with a prefix advertisement.

Optional attributes do not have to be recognized by all BGP implementations. Optional attributes can be set so that they are transitive and stay with the route advertisement from AS to AS. Other PAs are *non-transitive* and cannot be shared from AS to AS. In BGP, the *Network Layer Reachability Information (NLRI)* is a routing update that consists of the network prefix, prefix length, and any BGP PAs for the specific route.

The BGP attribute AS_Path is a well-known mandatory attribute and includes a complete list of all the ASNs that the prefix advertisement has traversed from its source AS. AS_Path is used as a loop-prevention mechanism in BGP. If a BGP router receives a prefix advertisement with its AS listed in the AS_Path attribute, it discards the prefix because the router thinks the advertisement forms a loop.

Every address family maintains a separate database and configuration for each protocol (address family + sub-address family) in BGP. This allows for a routing policy in one address family to be different from a routing policy in a different address family, even though the router uses the same BGP session with the other router. BGP includes an AFI and SAFI with every route advertisement to differentiate between the AFI and SAFI databases.

## Inter-Router Communication

BGP does not use hello packets to discover neighbors, as do IGP protocols, and it cannot discover neighbors dynamically. BGP was designed as an inter-autonomous routing protocol, implying that neighbor adjacencies should not change frequently and are coordinated. BGP neighbors are defined by IP address.

BGP uses TCP port 179 to communicate with other routers. TCP allows for handling of fragmentation, sequencing, and reliability (acknowledgment and retransmission) of communication packets. Most recent implementations of BGP set the do-not-fragment (DF) bit to prevent fragmentation and rely on path MTU discovery.

IGPs follow the physical topology because the sessions are formed with hellos that cannot cross network boundaries (that is, single hop only). BGP uses TCP, which is capable of crossing network boundaries (that is, multi-hop capable). While BGP can form neighbor adjacencies that are directly connected, it can also form adjacencies that are multiple hops away.

A BGP session refers to the established adjacency between two BGP routers. Multi-hop sessions require that the router use an underlying route installed in the RIB (static or from any routing protocol) to establish the TCP session with the remote endpoint.

In Figure 11-2, R1 is able to establish a direct BGP session with R2. In addition, R2 is able to establish a BGP session with R4, even though it passes through R3. R1 and R2 use a directly connected route to locate each other. R2 uses a static route to reach the 10.34.1.0/24 network, and R4 has a static route to reach the 10.23.1.0/24 network. R3 is unaware that R2 and R4 have established a BGP session even though the packets flow through R3.

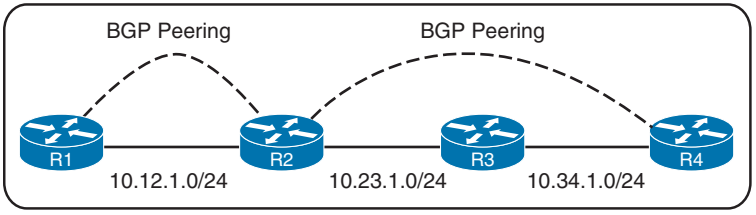



**Figure 11-2** *BGP Single- and Multi-Hop Sessions*

### BGP Session Types

BGP sessions are categorized into two types:

- **Internal BGP (iBGP):** Sessions established with an iBGP router that are in the same AS or that participate in the same BGP confederation. iBGP prefixes are assigned an administrative distance (AD) of 200 upon installation in the router's RIB.
- **External BGP (eBGP):** Sessions established with a BGP router that are in a different AS. eBGP prefixes are assigned an AD of 20 upon installation in the router's RIB.

### eBGP

eBGP peerings are the core component of BGP on the Internet. eBGP involves the exchange of network prefixes between autonomous systems. The following behaviors are different on eBGP sessions than on iBGP sessions:

- Time-to-live (TTL) on eBGP packets is set to 1 by default. eBGP packets drop in transit if a multi-hop BGP session is attempted. (TTL on iBGP packets is set to 255, which allows for multi-hop sessions.)
- The advertising router modifies the BGP next-hop address to the IP address sourcing the BGP connection.
- The advertising router prepends its ASN to the existing AS_Path variable.
- The receiving router verifies that the AS_Path variable does not contain an ASN that matches the local routers. BGP discards the NLRI if it fails the AS_Path loop prevention check.

The configurations for eBGP and iBGP sessions are fundamentally the same except that the ASN in the **remote-as** statement is different from the ASN defined in the BGP process.

## Basic BGP Configuration

When configuring BGP, it is best to think of the configuration from a modular perspective. BGP router configuration requires the following components:

- **BGP session parameters:** BGP session parameters provide settings that involve establishing communication to the remote BGP neighbor. Session settings include the ASN of the BGP peer, authentication, and keepalive timers.
- **Address family initialization:** The address family is initialized under the BGP router configuration mode. Network advertisement and summarization occur within the address family.
- **Activate the address family on the BGP peer:** In order for a session to initiate, one address family for a neighbor must be activated. The router's IP address is added to the neighbor table, and BGP attempts to establish a BGP session or accepts a BGP session initiated from the peer router

The following steps show how to configure BGP:

**Step 1.** Initialize the BGP routing process with the global command **router bgp** *as-number*.

**Step 2.** (Optional) Statically define the BGP router ID (RID). The dynamic RID allocation logic uses the highest IP address of the any *up* loopback interfaces. If there is not an *up* loopback interface, then the highest IP address of any active *up* interfaces becomes the RID when the BGP process initializes.

To ensure that the RID does not change, a static RID is assigned (typically representing an IPv4 address that resides on the router, such as a loopback address). Any IPv4 address can be used, including IP addresses not configured on the router. Statically configuring the BGP RID is a best practice and involves using the command **bgp router-id** *router-id*.

When the router ID changes, all BGP sessions reset and need to be reestablished.

**Step 3.** Identify the BGP neighbor's IP address and autonomous system number with the BGP router configuration command **neighbor** *ip-address* **remote-as** *as-number*. It is important to understand the traffic flow of BGP packets between peers. The source IP address of the BGP packets still reflects the IP address of the outbound interface. When a BGP packet is received, the router correlates the source IP address of the packet to the IP address configured for that neighbor. If the BGP packet source does not match an entry in the neighbor table, the packet cannot be associated to a neighbor and is discarded.

**NOTE** IOS activates the IPv4 address family by default. This can simplify the configuration in an IPv4 environment because steps 4 and 5 are optional but may cause confusion when working with other address families. The BGP router configuration command **no bgp default ip4-unicast** disables the automatic activation of the IPv4 AFI so that steps 4 and 5 are required.

**Step 4.** Initialize the address family with the BGP router configuration command **address-family** *afi safi*. Examples of *afi* values are IPv4 and IPv6, and examples of *safi* values are unicast and multicast.

**Step 5.** Activate the address family for the BGP neighbor with the BGP address family configuration command **neighbor** *ip-address* **activate**.

## Verification of BGP Sessions

The BGP session is verified with the command **show bgp** *afi safi* **summary**. Example 11-3 shows the IPv4 BGP unicast summary. Notice that the BGP RID and table version are the first components shown. The Up/Down column indicates that the BGP session is up for over 5 minutes.

## Prefix Advertisement

BGP **network** statements do not enable BGP for a specific interface; instead, they identify specific network prefixes to be installed into the BGP table, known as the *Loc-RIB table*.

After configuring a BGP **network** statement, the BGP process searches the global RIB for an exact network prefix match. The network prefix can be for a connected network, a secondary connected network, or any route from a routing protocol. After verifying that the **network** statement matches a prefix in the global RIB, the prefix is installed into the BGP Loc-RIB table. As the BGP prefix is installed into the Loc-RIB table, the following BGP PAs are set, depending on the RIB prefix type:

- **Connected network:** The next-hop BGP attribute is set to 0.0.0.0, the BGP origin attribute is set to i (IGP), and the BGP weight is set to 32,768.
- **Static route or routing protocol:** The next-hop BGP attribute is set to the next-hop IP address in the RIB, the BGP origin attribute is set to i (IGP), the BGP weight is set to 32,768, and the MED is set to the IGP metric.

Not every route in the Loc-RIB table is advertised to a BGP peer. All routes in the Loc-RIB table use the following process for advertisement to BGP peers.

**Step 1.** Pass a validity check. Verify that the NRLI is valid and that the next-hop address is resolvable in the global RIB. If the NRLI fails, the NLRI remains but does not process further.

**Step 2.** Process outbound neighbor route policies. After processing, if a route was not denied by the outbound policies, the route is maintained in the Adj-RIB-Out table for later reference.

**Step 3.** Advertise the NLRI to BGP peers. If the NLRI's next-hop BGP PA is 0.0.0.0, then the next-hop address is changed to the IP address of the BGP session.

Figure 11-9 shows the complete BGP route processing logic. It includes the receipt of a route from a BGP peers and the BGP best-path algorithm.

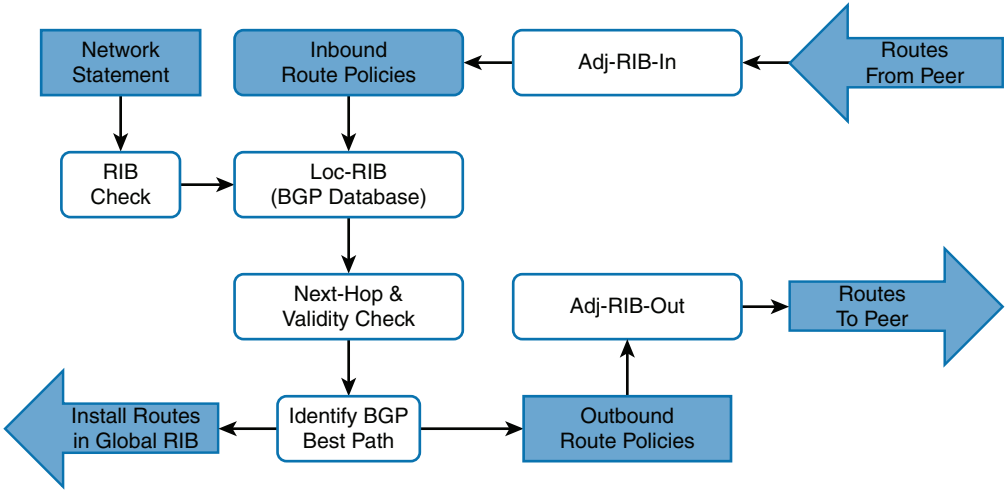



**Figure 11-9** *BGP Database Processing*

**Table 11-4** BGP Table Fields

| Field | Description |
|---|---|
| Network | A list of the network prefixes installed in BGP. If multiple NLRIs exist for the same prefix, only the first prefix is identified, and others are blank.<br>Valid NLRIs are indicated by the *.<br>The NLRI selected as the best path is indicated by an angle bracket (>). |
| Next Hop | A well-known mandatory BGP path attribute that defines the IP address for the next hop for that specific NLRI. |
| Metric | *Multiple-exit discrimator (MED)*: An optional non-transitive BGP path attribute used in BGP for the specific NLRI. |
| LocPrf | *Local Preference*: A well-known discretionary BGP path attribute used in the BGP best-path algorithm for the specific NLRI. |
| Weight | A locally significant Cisco-defined attribute used in the BGP best-path algorithm for the specific NLRI. |
| Path and Origin | *AS_Path*: A well-known mandatory BGP path attribute used for loop prevention and in the BGP best-path algorithm for the specific NLRI.<br>*Origin*: A well-known mandatory BGP path attribute used in the BGP best-path algorithm. A value of *i* represents an IGP, *e* indicates EGP, and *?* indicates a route that was redistributed into BGP. |

There are two techniques for BGP summarization:

- **Static:** Create a static route to Null0 for the summary network prefix and then advertise the prefix with a **network** statement. The downfall of this technique is that the summary route is always advertised, even if the networks are not available.
- **Dynamic:** Configure an aggregation network prefix. When viable component routes that match the aggregate network prefix enter the BGP table, then the aggregate prefix is created. The originating router sets the next hop to Null0 as a discard route for the aggregated prefix for loop prevention.

### Aggregate Address

Dynamic route summarization is accomplished with the BGP address family configuration command **aggregate-address** *network subnet-mask* [**summary-only**] [**as-set**].

Notice that the 172.16.0.0/20 and 192.168.0.0/16 network prefixes are visible, but the smaller component network prefixes still exist on all the routers. The **aggregate-address** command advertises the aggregated route in addition to the original component network prefixes. Using the optional **summary-only** keyword suppresses the component network prefixes in the summarized network range. Example 11-17 shows the configuration with the **summary-only** keyword.

### Atomic Aggregate

Aggregated routes act like new BGP routes with a shorter prefix length. When a BGP router summarizes a route, it does not advertise the AS_Path information from before the aggregation. BGP path attributes like AS_Path, MED, and BGP communities are not included in the new BGP advertisement.

The atomic aggregate attribute indicates that a loss of path information has occurred. To demonstrate this best, the previous BGP route aggregation on R1 has been removed and added to R2 so that R2 is now aggregating the 172.16.0.0/20 and 192.168.0.0/16 networks with suppression.

### Route Aggregation with AS_SET

To keep the BGP path information history, the optional **as-set** keyword may be used with the **aggregate-address** command. As the router generates the aggregate route, BGP attributes from the component aggregate routes are copied over to it. The AS_Path settings from the original prefixes are stored in the AS_SET portion of the AS_Path. The AS_SET, which is displayed within brackets, only counts as one hop, even if multiple ASs are listed.

## Multiprotocol BGP for IPv6

Multiprotocol BGP (MP-BGP) enables BGP to carry NLRI for multiple protocols, such as IPv4, IPv6, and Multiprotocol Label Switching (MPLS) Layer 3 virtual private networks (L3VPNs).

RFC 4760 defines the following new features:

- A new address family identifier (AFI) model
- New BGPv4 optional and nontransitive attributes:
  - Multiprotocol reachable NLRI
  - Multiprotocol unreachable NLRI

## IPv6 Configuration

All the BGP configuration rules demonstrated earlier apply with IPv6, except that the IPv6 address family must be initialized, and the neighbor is activated. Routers with only IPv6 addressing must statically define the BGP RID to allow sessions to form.

The protocol used to establish the BGP session is independent of the AFI/SAFI route advertisements. The TCP session used by BGP is a Layer 4 protocol, and it can use either an IPv4 or IPv6 address to form a session adjacency and exchange routes. Advertising IPv6 prefixes over an IPv4 BGP session is feasible but beyond the scope of this book as additional configuration is required.

### IPv6 Summarization

The same process for summarizing or aggregating IPv4 routes occurs with IPv6 routes, and the format is identical except that the configuration is placed under the IPv6 address family using the command **aggregate-address** *prefix/prefix-length* [**summary-only**] [**as-set**].

# Chapter 12

## Resiliency in Service Providers

Routing failures can occur within a service provider network, and some organizations chose to use a different SP for each circuit. A second service provider could be selected for a variety of reasons, but the choice typically comes down to cost, circuit availability for remote locations, or separation of the control plane.

By using a different SP, if one SP has problems in its network, network traffic can still flow across the other SP. In addition, adding more SPs means traffic can select an optimal path between devices due to the BGP best-path algorithm, discussed later in this chapter.

## Internet Transit Routing

If an enterprise uses BGP to connect with more than one service provider, it runs the risk of its autonomous system (AS) becoming a transit AS. In Figure 12-2, AS 500 is connecting to two different service providers (SP3 and SP4) for resiliency.

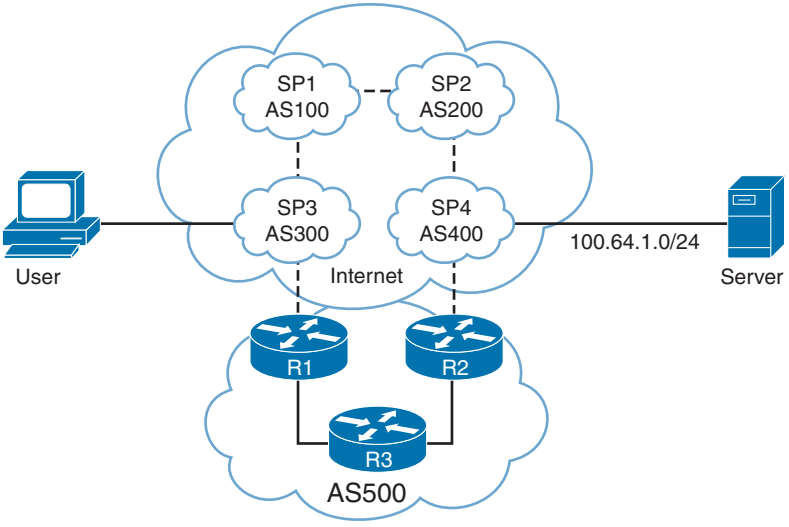



**Figure 12-2** *Enterprise Transit Routing*

Problems can arise if R1 and R2 use the default BGP routing policy. A user that connects to SP3 (AS 300) routes through the enterprise network (AS 500) to reach a server that attaches to SP4 (AS 400). SP3 receives the 100.64.1.0/24 prefix from AS 100 and AS 500. SP3 selects the path through AS 500 because the AS_Path is much shorter than going through SP1 and SP2's networks.

The AS 500 network is providing transit routing to everyone on the Internet, which can saturate AS 500's peering links. In addition to causing problems for the users in AS 500, this situation has an impact on traffic from the users that are trying to transverse AS 500.

Transit routing can be avoided by applying outbound BGP route policies that only allow for local BGP routes to be advertised to other autonomous systems. This is discussed later in this chapter, in the section "BGP Route Filtering and Manipulation."

### IGP Network Selection

When ACLS are used for IGP network selection, the source fields of the ACL are used to identify the network, and the destination fields identify the smallest prefix length allowed in the network range. Table 12-3 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with the extended ACL. Notice that the subtle difference in the destination wildcard for the 172.16.0.0 network affects the network ranges that are permitted in the second and third rows of the table.

**Table 12-3** Extended ACL for IGP Route Selection

| ACE Entry | Networks |
|---|---|
| **permit ip any any** | Permits all networks |
| **permit ip host 172.16.0.0 host 255.240.0.0** | Permits all networks in the 172.16.0.0/12 range |
| **permit ip host 172.16.0.0 host 255.255.0.0** | Permits all networks in the 172.16.0.0/16 range |
| **permit host 192.168.1.1** | Permits only the 192.168.1.1/32 network |

### BGP Network Selection

Extended ACLs react differently when matching BGP routes than when matching IGP routes. The source fields match against the network portion of the route, and the destination fields match against the network mask, as shown in Figure 12-5. Until the introduction of prefix lists, extended ACLs were the only match criteria used with BGP.

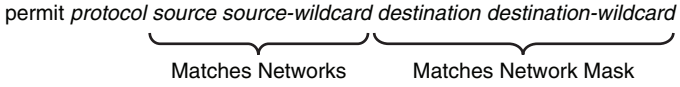



**Figure 12-5** *BGP Extended ACL Matches*

Table 12-4 demonstrates the concept of the wildcard for the network and subnet mask.

**Table 12-4** Extended ACL for BGP Route Selection

| Extended ACL | Matches These Networks |
|---|---|
| **permit ip 10.0.0.0 0.0.0.0 255.255.0.0 0.0.0.0** | Permits only the 10.0.0.0/16 network |
| **permit ip 10.0.0.0 0.0.255.0 255.255.255.0 0.0.0.0** | Permits any 10.0.x.0 network with a /24 prefix length |
| **permit ip 172.16.0.0 0.0.255.255 255.255.255.0 0.0.0.255** | Permits any 172.16.x.x network with a /24 to /32 prefix length |
| **permit ip 172.16.0.0 0.0.255.255 255.255.255.128 0.0.0.127** | Permits any 172.16.x.x network with a /25 to /32 prefix length |

A prefix match specification contains two parts: a high-order bit pattern and a high-order bit count, which determines the high-order bits in the bit pattern that are to be matched. Some documentation refers to the high-order bit pattern as the address or network and the high-order bit count as the length or mask length.

At this point, the prefix match specification logic looks identical to the functionality of an access list. The true power and flexibility comes in using matching length parameters to identify multiple networks with specific prefix lengths with one statement. The matching length parameter options are

- **le:** Less than or equal to, <=
- **ge:** Greater than or equal to, >=

### Prefix Lists

Prefix lists can contain multiple prefix matching specification entries that contain a permit or deny action. Prefix lists process in sequential order in a top-down fashion, and the first prefix match processes with the appropriate permit or deny action.

Prefix lists are configured with the global configuration command **ip prefix-list** *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern/high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*].

If a sequence is not provided, the sequence number auto-increments by 5, based on the highest sequence number. The first entry is 5. Sequencing enables the deletion of a specific entry. Because prefix lists cannot be resequenced, it is advisable to leave enough space for insertion of sequence numbers at a later time.

IOS and IOS XE require that the *ge-value* be greater than the high-order bit count and that the *le-value* be greater than or equal to the *ge-value*:

*high-order bit count* < *ge-value* <= *le-value*

### Regular Expressions (regex)

There may be times when conditionally matching on network prefixes may be too complicated, and identifying all routes from a specific organization is preferred. In such a case, path selection can be made by using a BGP AS_Path.

*Regular expressions (regex)* are used to parse through the large number of available ASNs (4,294,967,295). Regular expressions are based on query modifiers used to select the appropriate content. The BGP table can be parsed with regex by using the command **show bgp** *afi safi* **regexp** *regex-pattern*.

A route map has four components:

- **Sequence number:** Dictates the processing order of the route map.
- **Conditional matching criteria:** Identifies prefix characteristics (network, BGP path attribute, next hop, and so on) for a specific sequence.
- **Processing action:** Permits or denies the prefix.
- **Optional action:** Allows for manipulations, depending on how the route map is referenced on the router. Actions can include modification, addition, or removal of route characteristics.

A route map uses the command syntax **route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]. The following rules apply to route map statements:

- If a processing action is not provided, the default value **permit** is used.
- If a sequence number is not provided, the sequence number is incremented by 10 automatically.
- If a matching statement is not included, an implied *all prefixes* is associated with the statement.
- Processing within a route map stops after all optional actions have processed (if configured) after matching a conditional matching criterion.

## Conditional Matching

Now that the components and processing order of a route map have been explained, this section expands on how a route can be matched. Table 12-7 provides the command syntax for the most common methods for conditionally matching prefixes and describes their usage. As you can see, there are a number of options available.

**Table 12-7** Conditional Match Options

| Match Command | Description |
|---|---|
| **match as-path** *acl-number* | Selects prefixes based on a regex query to isolate the ASN in the BGP path attribute (PA) AS path. The AS path ACLs are numbered 1 to 500. This command allows for multiple match variables. |
| **match ip address** {*acl-number* \| *acl-name*} | Selects prefixes based on network selection criteria defined in the ACL. This command allows for multiple match variables. |
| **match ip address prefix-list** *prefix-list-name* | Selects prefixes based on prefix selection criteria. This command allows for multiple match variables. |
| **match local-preference** *local-preference* | Selects prefixes based on the BGP attribute local preference. This command allows for multiple match variables. |
| **match metric** {*1-4294967295* \| **external** *1-4294967295*}[**+-** *deviation*] | Selects prefixes based on a metric that can be exact, a range, or within acceptable deviation. |
| **match tag** *tag-value* | Selects prefixes based on a numeric tag (0 to 4294967295) that was set by another router. This command allows for multiple match variables. |

### Multiple Conditional Match Conditions

If there are multiple variables (ACLs, prefix lists, tags, and so on) configured for a specific route map sequence, only one variable must match for the prefix to qualify. The Boolean logic uses an OR operator for this configuration.

## Optional Actions

In addition to permitting the prefix to pass, route maps can modify route attributes. Table 12-8 provides a brief overview of the most popular attribute modifications.

**Table 12-8** Route Map Set Actions

| Set Action | Description |
|---|---|
| **set as-path prepend** {*as-number-pattern* \| **last-as** *1-10*} | Prepends the AS path for the network prefix with the pattern specified or from multiple iterations from a neighboring AS. |
| **set ip next-hop** { *ip-address* \| **peer-address** \| **self** } | Sets the next-hop IP address for any matching prefix. BGP dynamic manipulation uses the **peer-address** or **self** keywords. |
| **set local-preference** *0-4294967295* | Sets the BGP PA local preference. |
| **set metric** {**+***value* \| **-***value* \| *value*}<br>(where value parameters are 0–4294967295) | Modifies the existing metric or sets the metric for a route. |
| **set origin** {**igp** \| **incomplete**} | Sets the BGP PA origin. |
| **set tag** *tag-value* | Sets a numeric tag (0–4294967295) for identification of networks by other routers |
| **set weight** *0–65535* | Sets the BGP PA weight. |

### Distribute List Filtering

Distribute lists allow the filtering of network prefixes on a neighbor-by-neighbor basis, using standard or extended ACLs. Configuring a distribute list requires using the BGP address-family configuration command **neighbor** *ip-address* **distribute-list** {*acl-number* | *acl-name*} {**in**|**out**}. Remember that extended ACLs for BGP use the source fields to match the network portion and the destination fields to match against the network mask.

### Prefix List Filtering

Prefix lists allow the filtering of network prefixes on a neighbor-by-neighbor basis, using a prefix list. Configuring a prefix list involves using the BGP address family configuration command **neighbor** *ip-address* **prefix-list** *prefix-list-name* {**in** | **out**}.

Processing is peformed in a sequential top-down order, and the first qualifying match processes against the appropriate **permit** or **deny** action. An implicit deny exists at the end of the AS path ACL. IOS supports up to 500 AS path ACLs and uses the command **ip as-path access-list** *acl-number* {**deny** | **permit**} *regex-query* for creating an AS path ACL. The ACL is then applied with the command **neighbor** *ip-address* **filter-list** *acl-number* {**in**|**out**}.

## Route Maps

As explained earlier, route maps provide additional functionality over pure filtering. Route maps provide a method to manipulate BGP path attributes as well. Route maps are applied on a BGP neighbor basis for routes that are advertised or received. A different route map can be used for each direction. The route map is associated with the BGP neighbor with the command **neighbor** *ip-address* **route-map** *route-map-name* {**in**|**out**} under the specific address family.

## BGP Communities

BGP communities provide additional capability for tagging routes and for modifying BGP routing policy on upstream and downstream routers. BGP communities can be appended, removed, or modified selectively on each attribute as a route travels from router to router.

*BGP communities* are an optional transitive BGP attribute that can traverse from AS to AS. A BGP community is a 32-bit number that can be included with a route. A BGP community can be displayed as a full 16-bit number (0–4,294,967,295) or as two 16-bit numbers (0–65535):(0–65535), commonly referred to as *new format*.

*Private BGP communities* follow a particular convention where the first 16 bits represent the AS of the community origination, and the second 16 bits represent a pattern defined by the originating AS. A private BGP community pattern can vary from organization to organization, does not need to be registered, and can signify geographic locations for one AS while signifying a method of route advertisement in another AS. Some organizations publish their private BGP community patterns on websites such as http://www.onesc.net/communities/.

In 2006, RFC 4360 expanded BGP communities' capabilities by providing an extended format. *Extended BGP communities* provide structure for various classes of information and are commonly used for VPN services. RFC 8092 provides support for communities larger than 32 bits (which are beyond the scope of this book).

## Enabling BGP Community Support

IOS and IOS XE routers do not advertise BGP communities to peers by default. Communities are enabled on a neighbor-by-neighbor basis with the BGP address family configuration command **neighbor** *ip-address* **send-community** [**standard** | **extended** | **both**] under the neighbor's address family configuration. If a keyword is not specified, standard communities are sent by default.

IOS XE nodes can display communities in new format, which is easier to read, with the global configuration command **ip bgp-community new-format**.

Conditionally matching requires the creation of a community list that shares a similar structure to an ACL, can be standard or expanded, and can be referenced by number or name. Standard community lists are numbered 1 to 99 and match either well-known communities or a private community number (*as-number:16-bit-number*). Expanded community lists are numbered 100 to 500 and use regex patterns.

### Setting Private BGP Communities

A private BGP community is set in a route map with the command **set community** *bgp-community* [**additive**]. By default, when setting a community, any existing communities are over-written but can be preserved by using the optional **additive** keyword.

### Routing Path Selection Using Longest Match

Routers always select the path a packet should take by examining the prefix length of a network entry. The path selected for a packet is chosen based on the prefix length, where the longest prefix length is always preferred. For example, /28 is preferred over /26, and /26 is preferred over /24.

This logic can be used to influence path selection in BGP. Assume that an organization owns the 100.64.0.0/16 network range but only needs to advertise two subnets (100.64.1.0/24 and 100.64.2.0/24). It could advertise both prefixes (100.64.1.0/24 and 100.64.2.0/24) from all its routers, but how can it distribute the load for each subnet if all traffic comes in on one router (such as R1)?

The organization could modify various BGP path attributes (PAs) that are advertised externally, but an SP could have a BGP routing policy that ignores those path attributes, resulting in random receipt of network traffic.

A more elegant way that guarantees that paths are selected deterministically outside the organization is to advertise a summary prefix (100.64.0.0/16) out both routers. Then the organization can advertise a longer matching prefix out the router that should receive network traffic for that prefix.

The BGP best-path algorithm uses the following attributes, in the order shown, for the best-path selection:

1. Weight
2. Local preference
3. Local originated (network statement, redistribution, or aggregation)
4. AIGP
5. Shortest AS_Path
6. Origin type
7. Lowest MED
8. eBGP over iBGP
9. Lowest IGP next hop
10. If both paths are external (eBGP), prefer the first (oldest)
11. Prefer the route that comes from the BGP peer with the lower RID
12. Prefer the route with the minimum cluster list length
13. Prefer the path that comes from the lowest neighbor address

# Chapter 13

Multicast traffic provides one-to-many communication, where only one data packet is sent on a link as needed and then is replicated between links as the data forks (splits) on a network device along the multicast distribution tree (MDT). The data packets are known as a *stream* that uses a special destination IP address, known as a *group address*. A server for a stream still manages only one session, and network devices selectively request to receive the stream. Recipient devices of a multicast stream are known as *receivers*. Common applications that take advantage of multicast traffic include Cisco TelePresence, real-time video, IPTV, stock tickers, distance learning, video/audio conferencing, music on hold, and gaming.

**Table 13-2** IP Multicast Addresses Assigned by IANA

| Designation | Multicast Address Range |
|---|---|
| Local network control block | 224.0.0.0 to 224.0.0.255 |
| Internetwork control block | 224.0.1.0 to 224.0.1.255 |
| Ad hoc block I | 224.0.2.0 to 224.0.255.255 |
| Reserved | 224.1.0.0 to 224.1.255.255 |
| SDP/SAP block | 224.2.0.0 to 224.2.255.255 |
| Ad hoc block II | 224.3.0.0 to 224.4.255.255 |
| Reserved | 224.5.0.0 to 224.255.255.255 |
| Reserved | 225.0.0.0 to 231.255.255.255 |
| Source Specific Multicast (SSM) block | 232.0.0.0 to 232.255.255.255 |
| GLOP block | 233.0.0.0 to 233.251.255.255 |
| Ad hoc block III | 233.252.0.0 to 233.255.255.255 |
| Reserved | 234.0.0.0 to 238.255.255.255 |
| Administratively scoped block | 239.0.0.0 to 239.255.255.255 |

**Table 13-3** Well-Known Reserved Multicast Addresses

| IP Multicast Address | Description |
|---|---|
| 224.0.0.0 | Base address (reserved) |
| 224.0.0.1 | All hosts in this subnet (all-hosts group) |
| 224.0.0.2 | All routers in this subnet |
| 224.0.0.5 | All OSPF routers (AllSPFRouters) |
| 224.0.0.6 | All OSPF DRs (AllDRouters) |
| 224.0.0.9 | All RIPv2 routers |
| 224.0.0.10 | All EIGRP routers |
| 224.0.0.13 | All PIM routers |
| 224.0.0.18 | VRRP |
| 224.0.0.22 | IGMPv3 |
| 224.0.0.102 | HSRPv2 and GLBP |
| 224.0.1.1 | NTP |
| 224.0.1.39 | Cisco-RP-Announce (Auto-RP) |
| 224.0.1.40 | Cisco-RP-Discovery (Auto-RP) |

## Layer 2 Multicast Addresses

Historically, NICs on a LAN segment could receive only packets destined for their burned-in MAC address or the broadcast MAC address. Using this logic can cause a burden on routing resources during packet replication for LAN segments. Another method for multicast traffic was created so that replication of multicast traffic did not require packet manipulation, and a method of using a common destination MAC address was created.

A MAC address is a unique value associated with a NIC that is used to uniquely identify the NIC on a LAN segment. MAC addresses are 12-digit hexadecimal numbers (48 bits in length), and they are typically stored in 8-bit segments separated by hyphens (-) or colons (:) (for example, 00-12-34-56-78-00 or 00:12:34:56:78:00).

Every multicast group address (IP address) is mapped to a special MAC address that allows Ethernet interfaces to identify multicast packets to a specific group. A LAN segment can have multiple streams, and a receiver knows which traffic to send to the CPU for processing based on the MAC address assigned to the multicast traffic.

The first 24 bits of a multicast MAC address always start with 01:00:5E. The low-order bit of the first byte is the *individual/group bit (I/G)* bit, also known as the unicast/multicast bit, and when it is set to 1, it indicates that the frame is a multicast frame, and the 25th bit is always 0. The lower 23 bits of the multicast MAC address are copied from the lower 23 bits of the multicast group IP address.

*Internet Group Management Protocol (IGMP)* is the protocol that receivers use to join multicast groups and start receiving traffic from those groups. IGMP must be supported by receivers and the router interfaces facing the receivers. When a receiver wants to receive multicast traffic from a source, it sends an IGMP join to its router. If the router does not have IGMP enabled on the interface, the request is ignored.

## IGMPv2

IGMPv2 uses the message format shown in Figure 13-7. This message is encapsulated in an IP packet with a protocol number of 2. Messages are sent with the IP router alert option set, which indicates that the packets should be examined more closely, and a time-to-live (TTL) of 1. TTL is an 8-bit field in an IP packet header that is set by the sender of the IP packet and decremented by every router on the route to its destination. If the TTL reaches 0 before reaching the destination, the packet is discarded. IGMP packets are sent with a TTL of 1 so that packets are processed by the local router and not forwarded by any router.

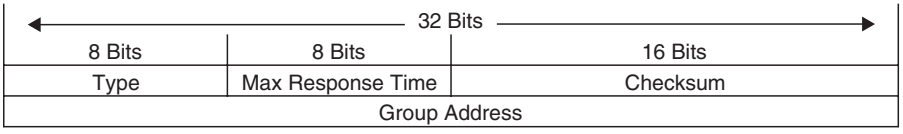



**Figure 13-7** *IGMP Message Format*

The IGMP message format fields are defined as follows:

- **Type:** This field describes five different types of IGMP messages used by routers and receivers:
  - **Version 2 membership report** (type value 0x16) is a message type also commonly referred to as an IGMP join; it is used by receivers to join a multicast group or to respond to a local router's membership query message.
  - **Version 1 membership report** (type value 0x12) is used by receivers for backward compatibility with IGMPv1.
  - **Version 2 leave group** (type value 0x17) is used by receivers to indicate they want to stop receiving multicast traffic for a group they joined.
  - **General membership query** (type value 0x11) is sent periodically sent to the all-hosts group address 224.0.0.1 to see whether there are any receivers in the attached subnet. It sets the group address field to 0.0.0.0.
  - **Group specific query** (type value 0x11) is sent in response to a leave group message to the group address the receiver requested to leave. The group address is the destination IP address of the IP packet and the group address field.
- **Max response time:** This field is set only in general and group-specific membership query messages (type value 0x11); it specifies the maximum allowed time before sending a responding report in units of one-tenth of a second. In all other messages, it is set to 0x00 by the sender and ignored by receivers.
- **Checksum:** This field is the 16-bit 1s complement of the 1s complement sum of the IGMP message. This is the standard checksum algorithm used by TCP/IP.
- **Group address:** This field is set to 0.0.0.0 in general query messages and is set to the group address in group-specific messages. Membership report messages carry the address of the group being reported in this field; group leave messages carry the address of the group being left in this field.

When a receiver wants to receive a multicast stream, it sends an unsolicited membership report, commonly referred to as an IGMP join, to the local router for the group it wants to join (for example, 239.1.1.1). The local router then sends this request upstream toward the source using a PIM join message. When the local router starts receiving the multicast stream, it forwards it downstream to the subnet where the receiver that requested it resides.

In IGMPv2, when a receiver sends a membership report to join a multicast group, it does not specify which source it would like to receive multicast traffic from. IGMPv3 is an extension of IGMPv2 that adds support for multicast source filtering, which gives the receivers the capability to pick the source they wish to accept multicast traffic from.

IGMP snooping, defined in RFC 4541, is the most widely used method and works by examining IGMP joins sent by receivers and maintaining a table of interfaces to IGMP joins. When the switch receives a multicast frame destined for a multicast group, it forwards the packet only out the ports where IGMP joins were received for that specific multicast group.

Receivers use IGMP to join a multicast group, which is sufficient if the group's source connects to the same router to which the receiver is attached. A multicast routing protocol is necessary to route the multicast traffic throughout the network so that routers can locate and request multicast streams from other routers. Multiple multicast routing protocols exist, but Cisco fully supports only Protocol Independent Multicast (PIM).

A *source tree* is a multicast distribution tree where the source is the root of the tree, and branches form a distribution tree through the network all the way down to the receivers. When this tree is built, it uses the shortest path through the network from the source to the leaves of the tree; for this reason, it is also referred to as a shortest path tree (SPT).

A shared tree is a multicast distribution tree where the root of the shared tree is not the source but a router designated as the rendezvous point (RP). For this reason, shared trees are also referred to as *RP trees (RPTs)*. Multicast traffic is forwarded down the shared tree according to the group address G that the packets are addressed to, regardless of the source address. For this reason, the forwarding state on the shared tree is referred to by the notation (*,G), pronounced "star comma G." Figure 13-13 illustrates a shared tree where R2 is the RP, and the (*,G) is (*,239.1.1.1).

The following list defines the common PIM terminology illustrated in Figure 13-14:

- **Reverse Path Forwarding (RPF) interface:** The interface with the lowest-cost path (based on administrative distance [AD] and metric) to the IP address of the source (SPT) or the RP, in the case of shared trees. If multiple interfaces have the same cost, the interface with the highest IP address is chosen as the tiebreaker. An example of this type of interface is Te0/1/2 on R5 because it is the shortest path to the source. Another example is Te1/1/1 on R7 because the shortest path to the source was determined to be through R4.
- **RPF neighbor:** The PIM neighbor on the RPF interface. For example, if R7 is using the RPT shared tree, the RPF neighbor would be R3, which is the lowest-cost path to the RP. If it is using the SPT, R4 would be its RPF neighbor because it offers the lowest cost to the source.
- **Upstream:** Toward the source of the tree, which could be the actual source in source-based trees or the RP in shared trees. A PIM join travels upstream toward the source.
- **Upstream interface:** The interface toward the source of the tree. It is also known as the RPF interface or the incoming interface (IIF). An example of an upstream interface is R5's Te0/1/2 interface, which can send PIM joins upstream to its RPF neighbor.
- **Downstream:** Away from the source of the tree and toward the receivers.
- **Downstream interface:** Any interface that is used to forward multicast traffic down the tree, also known as an outgoing interface (OIF). An example of a downstream interface is R1's Te0/0/0 interface, which forwards multicast traffic to R3's Te0/0/1 interface.
- **Incoming interface (IIF):** The only type of interface that can accept multicast traffic coming from the source, which is the same as the RPF interface. An example of this type of interface is Te0/0/1 on R3 because the shortest path to the source is known through this interface.
- **Outgoing interface (OIF):** Any interface that is used to forward multicast traffic down the tree, also known as the downstream interface.
- **Outgoing interface list (OIL):** A group of OIFs that are forwarding multicast traffic to the same group. An example of this is R1's Te0/0/0 and Te0/0/1 interfaces sending multicast traffic downstream to R3 and R4 for the same multicast group.
- **Last-hop router (LHR):** A router that is directly attached to the receivers, also known as a leaf router. It is responsible for sending PIM joins upstream toward the RP or to the source.
- **First-hop router (FHR):** A router that is directly attached to the source, also known as a root router. It is responsible for sending register messages to the RP.
- **Multicast Routing Information Base (MRIB):** A topology table that is also known as the multicast route table (mroute), which derives from the unicast routing table and PIM. MRIB contains the source S, group G, incoming interfaces (IIF), outgoing interfaces (OIFs), and RPF neighbor information for each multicast route as well as other multicast-related information.

- **Multicast Forwarding Information Base (MFIB):** A forwarding table that uses the MRIB to program multicast forwarding information in hardware for faster forwarding.
- **Multicast state:** The multicast traffic forwarding state that is used by a router to forward multicast traffic. The multicast state is composed of the entries found in the mroute table (S, G, IIF, OIF, and so on).

There are currently five PIM operating modes:

- PIM Dense Mode (PIM-DM)
- PIM Sparse Mode (PIM-SM)
- PIM Sparse Dense Mode
- PIM Source Specific Multicast (PIM-SSM)
- PIM Bidirectional Mode (Bidir-PIM)

**Table 13-4** PIM Control Message Types

| Type | Message Type | Destination | PIM Protocol |
|---|---|---|---|
| 0 | Hello | 224.0.0.13 (all PIM routers) | PIM-SM, PIM-DM, Bidir-PIM and SSM |
| 1 | Register | RP address (unicast) | PIM-SM |
| 2 | Register stop | First-hop router (unicast) | PIM SM |
| 3 | Join/prune | 224.0.0.13 (all PIM routers) | PIM-SM, Bidir-PIM and SSM |
| 4 | Bootstrap | 224.0.0.13 (all PIM routers) | PIM-SM and Bidir-PIM |
| 5 | Assert | 224.0.0.13 (all PIM routers) | PIM-SM, PIM-DM, and Bidir-PIM |
| 8 | Candidate RP advertisement | Bootstrap router (BSR) address (unicast to BSR) | PIM-SM and Bidir-PIM |
| 9 | State refresh | 224.0.0.13 (all PIM routers) | PIM-DM |
| 10 | DF election | 224.0.0.13 (all PIM routers) | Bidir-PIM |

PIM routers can be configured for PIM Dense Mode (PIM-DM) when it is safe to assume that the receivers of a multicast group are located on every subnet within the network—in other words, when the multicast group is densely populated across the network.

PIM Sparse Mode (PIM-SM) was designed for networks with multicast application receivers scattered throughout the network—in other words, when the multicast group is sparsely populated across the network. However, PIM-SM also works well in densely populated networks. It also assumes that no receivers are interested in multicast traffic unless they explicitly request it.

Figure 13-17 shows Receiver A attached to the LHR joining multicast group G. The LHR knows the IP address of the RP for group G, and it then sends a (*,G) PIM join for this group to the RP. If the RP were not directly connected, this (*,G) PIM join would travel hop-by-hop to the RP, building a branch of the shared tree that would extend from the RP to the LHR. At this point, group G multicast traffic arriving at the RP can flow down the shared tree to the receiver.

### Source Registration

In Figure 13-17, as soon as the source for a group G sends a packet, the FHR that is attached to this source is responsible for registering this source with the RP and requesting the RP to build a tree back to that router.

The FHR encapsulates the multicast data from the source in a special PIM-SM message called the *register message* and unicasts that data to the RP using a unidirectional PIM tunnel.

When the RP receives the register message, it decapsulates the multicast data packet inside the register message, and if there is no active shared tree because there are no interested receivers, the RP sends a register stop message directly to the registering FHR, without traversing the PIM tunnel, instructing it to stop sending the register messages.

If there is an active shared tree for the group, it forwards the multicast packet down the shared tree, and it sends an (S,G) join back toward the source network S to create an (S,G) SPT. If there are multiple hops (routers) between the RP and the source, this results in an (S,G) state being created in all the routers along the SPT, including the RP. There will also be a (*,G) in R1 and all of the routers between the FHR and the RP.

As soon as the SPT is built from the source router to the RP, multicast traffic begins to flow natively from the source S to the RP.

Once the RP begins receiving data natively (that is, down the SPT) from source S, it sends a register stop message to the source's FHR to inform it that it can stop sending the unicast register messages. At this point, multicast traffic from the source is flowing down the SPT to the RP and, from there, down the shared tree (RPT) to the receiver.

The PIM register tunnel from the FHR to the RP remains in an active up/up state even when there are no active multicast streams, and it remains active as long as there is a valid RPF path for the RP.

PIM-SM allows the LHR to switch from the shared tree to an SPT for a specific source. In Cisco routers, this is the default behavior, and it happens immediately after the first multicast packet is received from the RP via the shared tree, even if the shortest path to the source is through the RP. Figure 13-18 illustrates the SPT switchover concept. When the LHR receives the first multicast packet from the RP, it becomes aware of the IP address of the multicast source. At this point, the LHR checks its unicast routing table to see which is the shortest path to the source, and it sends an (S,G) PIM join hop-by-hop to the FHR to form an SPT. Once it receives a multicast packet from the FHR through the SPT, if necessary, it switches the RPF interface to be the one in the direction of the SPT to the FHR, and it then sends a PIM prune message to the RP to shut off the duplicate multicast traffic coming from it through the shared tree. In Figure 13-18, the shortest path to the source is between R1 and R3; if that link were shut down or not present, the shortest path would be through the RP, in which case an SPT switchover would still take place.

When multiple PIM-SM routers exist on a LAN segment, PIM hello messages are used to elect a designated router (DR) to avoid sending duplicate multicast traffic into the LAN or the RP. By default, the DR priority value of all PIM routers is 1, and it can be changed to force a particular router to become the DR during the DR election process, where a higher DR priority is preferred. If a router in the subnet does not support the DR priority option or if all routers have the same DR priority, the highest IP address in the subnet is used as a tiebreaker.

*Reverse Path Forwarding (RPF)* is an algorithm used to prevent loops and ensure that multicast traffic is arriving on the correct interface. RPF functions as follows:

- If a router receives a multicast packet on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a router forwards the packet out the interfaces present in the outgoing interface list (OIL) of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is discarded to prevent loops.

## PIM Forwarder

There are certain scenarios in which duplicate multicast packets could flow onto a multi-access network. The PIM assert mechanism stops these duplicate flows.

Figure 13-20 illustrates R2 and R3 both receiving the same (S,G) traffic via their RPF interfaces and forwarding the packets on to the LAN segment. R2 and R3 therefore receive an (S,G) packet via their downstream OIF that is in the OIF of their (S,G) entry. In other words, they detect a multicast packet for a specific (S,G) coming into their OIF that is also going out the same OIF for the same (S,G). This triggers the assert mechanism.

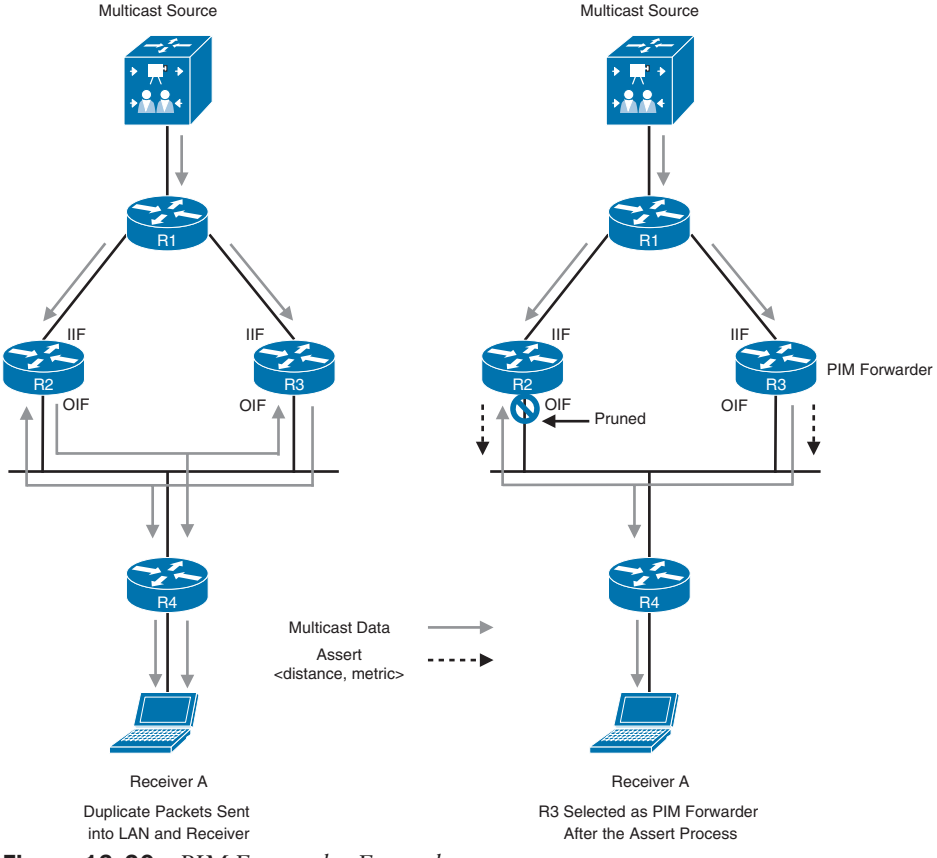



**Figure 13-20** *PIM Forwarder Example*

R2 and R3 both send PIM assert messages into the LAN. These assert messages send their administrative distance (AD) and route metric back to the source to determine which router should forward the multicast traffic to that network segment.

Each router compares its own values with the received values. Preference is given to the PIM message with the lowest AD to the source. If a tie exists, the lowest route metric for the protocol wins; and as a final tiebreaker, the highest IP address is used.

The losing router prunes its interface just as if it had received a prune on this interface, and the winning router is the PIM forwarder for the LAN.

In PIM-SM, it is mandatory to choose one or more routers to operate as *rendezvous points (RPs)*. An RP is a single common root placed at a chosen point of a shared distribution tree, as described earlier in this chapter. An RP can be either configured statically in each router or learned through a dynamic mechanism. A PIM router can be configured to function as an RP either statically in each router in the multicast domain or dynamically by configuring Auto-RP or a PIM bootstrap router (BSR), as described in the following sections.

It is possible to statically configure RP for a multicast group range by configuring the address of the RP on every router in the multicast domain. Configuring static RPs is relatively simple and can be achieved with one or two lines of configuration on each router. If the network does not have many different RPs defined or if the RPs do not change very often, this could be the simplest method for defining RPs. It can also be an attractive option if the network is small.

Auto-RP is a Cisco proprietary mechanism that automates the distribution of group-to-RP mappings in a PIM network.

A C-RP advertises its willingness to be an RP via RP announcement messages. These messages are sent by default every RP announce interval, which is 60 seconds by default, to the reserved well-known multicast group 224.0.1.39 (Cisco-RP-Announce). The RP announcements contain the default group range 224.0.0.0/4, the C-RP's address, and the hold time, which is three times the RP announce interval. If there are multiple C-RPs, the C-RP with the highest IP address is preferred.

RP MAs join group 224.0.1.39 to receive the RP announcements. They store the information contained in the announcements in a group-to-RP mapping cache, along with hold times. If multiple RPs advertise the same group range, the C-RP with the highest IP address is elected.

The *bootstrap router (BSR)* mechanism, described in RFC 5059, is a nonproprietary mechanism that provides a fault-tolerant, automated RP discovery and distribution mechanism.

PIM uses the BSR to discover and announce RP set information for each group prefix to all the routers in a PIM domain. This is the same function accomplished by Auto-RP, but the BSR is part of the PIM Version 2 specification. The RP set is a group-to-RP mapping that contains the following components:

- Multicast group range
- RP priority
- RP address
- Hash mask length
- SM/Bidir flag

A router that is configured as a candidate RP (C-RP) receives the BSR messages, which contain the IP address of the currently active BSR. Because it knows the IP address of the BSR, the C-RP can unicast candidate RP advertisement (C-RP-Adv) messages directly to it. A C-RP-Adv message carries a list of group address and group mask field pairs. This enables a C-RP to specify the group ranges for which it is willing to be the RP.

# Chapter 14

There are three different QoS implementation models:

- **Best effort:** QoS is not enabled for this model. It is used for traffic that does not require any special treatment.
- **Integrated Services (IntServ):** Applications signal the network to make a bandwidth reservation and to indicate that they require special QoS treatment.
- **Differentiated Services (DiffServ):** The network identifies classes that require special QoS treatment.

The IntServ model was created for real-time applications such as voice and video that require bandwidth, delay, and packet-loss guarantees to ensure both predictable and guaranteed service levels. In this model, applications signal their requirements to the network to reserve the end-to-end resources (such as bandwidth) they require to provide an acceptable user experience. IntServ uses Resource Reservation Protocol (RSVP) to reserve resources throughout a network for a specific application and to provide call admission control (CAC) to guarantee that no other IP traffic can use the reserved bandwidth. The bandwidth reserved by an application that is not being used is wasted.

DiffServ was designed to address the limitations of the best-effort and IntServ models. With this model, there is no need for a signaling protocol, and there is no RSVP flow state to maintain on every single node, which makes it highly scalable; QoS characteristics (such as bandwidth and delay) are managed on a hop-by-hop basis with QoS policies that are defined independently at each device in the network. DiffServ is not considered an end-to-end QoS solution because end-to-end QoS guarantees cannot be enforced.

DiffServ divides IP traffic into classes and marks it based on business requirements so that each of the classes can be assigned a different level of service. As IP traffic traverses a network, each of the network devices identifies the packet class by its marking and services the packets according to this class. Many levels of service can be chosen with DiffServ. For example, IP phone voice traffic is very sensitive to latency and jitter, so it should always be given preferential treatment over all other application traffic. Email, on the other hand, can withstand a great deal of delay and could be given best-effort service, and non-business, non-critical scavenger traffic (such as from YouTube) can either be heavily rate limited or blocked entirely. The DiffServ model is the most popular and most widely deployed QoS model and is covered in detail in this chapter.

### Classification

*Packet classification* is a QoS mechanism responsible for distinguishing between different traffic streams. It uses traffic descriptors to categorize an IP packet within a specific class. Packet classification should take place at the network edge, as close to the source of the traffic as possible. Once an IP packet is classified, packets can then be marked/re-marked, queued, policed, shaped, or any combination of these and other actions.

The following traffic descriptors are typically used for classification:

- **Internal:** QoS groups (locally significant to a router)
- **Layer 1:** Physical interface, subinterface, or port
- **Layer 2:** MAC address and 802.1Q/p Class of Service (CoS) bits
- **Layer 2.5:** MPLS Experimental (EXP) bits
- **Layer 3:** Differentiated Services Code Points (DSCP), IP Precedence (IPP), and source/destination IP address
- **Layer 4:** TCP or UDP ports
- **Layer 7:** Next Generation Network-Based Application Recognition (NBAR2)

NBAR2 is a deep packet inspection engine that can classify and identify a wide variety of protocols and applications using Layer 3 to Layer 7 data, including difficult-to-classify applications that dynamically assign Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers.

NBAR2 can recognize more than 1000 applications, and monthly protocol packs are provided for recognition of new and emerging applications, without requiring an IOS upgrade or router reload.

NBAR2 has two modes of operation:

- **Protocol Discovery:** Protocol Discovery enables NBAR2 to discover and get real-time statistics on applications currently running in the network. These statistics from the Protocol Discovery mode can be used to define QoS classes and policies using MQC configuration.
- **Modular QoS CLI (MQC):** Using MQC, network traffic matching a specific network protocol such as Cisco Webex can be placed into one traffic class, while traffic that matches a different network protocol such as YouTube can be placed into another traffic class. After traffic has been classified in this way, different QoS policies can be applied to the different classes of traffic.

## Marking

Packet *marking* is a QoS mechanism that colors a packet by changing a field within a packet or a frame header with a traffic descriptor so it is distinguished from other packets during the application of other QoS mechanisms (such as re-marking, policing, queuing, or congestion avoidance).

The following traffic descriptors are used for marking traffic:

- **Internal:** QoS groups
- **Layer 2:** 802.1Q/p Class of Service (CoS) bits
- **Layer 2.5:** MPLS Experimental (EXP) bits
- **Layer 3:** Differentiated Services Code Points (DSCP) and IP Precedence (IPP)

The 802.1Q standard is an IEEE specification for implementing VLANs in Layer 2 switched networks. The 802.1Q specification defines two 2-byte fields: Tag Protocol Identifier (TPID) and Tag Control Information (TCI), which are inserted within an Ethernet frame following the Source Address field, as illustrated in Figure 14-2.

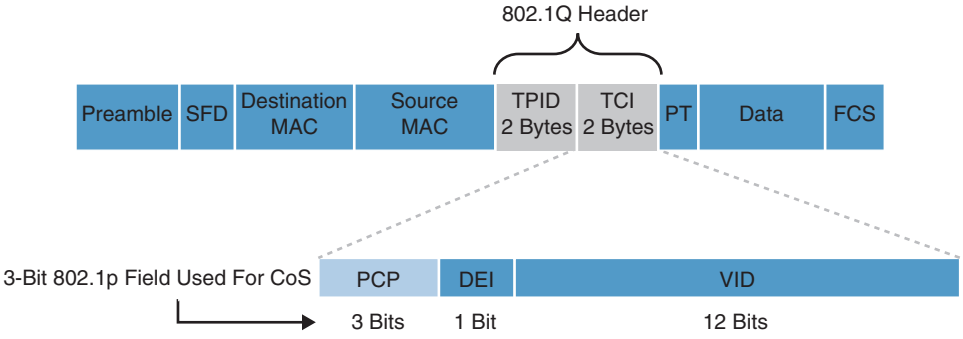



**Figure 14-2** *802.1Q Layer 2 QoS Using 802.1p CoS*

The TCI field is a 16-bit field composed of the following three fields:

- Priority Code Point (PCP) field (3 bits)
- Drop Eligible Indicator (DEI) field (1 bit)
- VLAN Identifier (VLAN ID) field (12 bits)

### Priority Code Point (PCP)

The specifications of the 3-bit PCP field are defined by the IEEE 802.1p specification. This field is used to mark packets as belonging to a specific CoS. The CoS marking allows a Layer 2 Ethernet frame to be marked with eight different levels of priority values, 0 to 7, where 0 is the lowest priority and 7 is the highest. Table 14-2 includes the IEEE 802.1p specification standard definition for each CoS.

**Table 14-2** IEEE 802.1p CoS Definitions

| PCP Value/Priority | Acronym | Traffic Type |
|---|---|---|
| 0 (lowest) | BK | Background |
| 1 (default) | BE | Best effort |
| 2 | EE | Excellent effort |
| 3 | CA | Critical applications |
| 4 | VI | Video with < 100 ms latency and jitter |
| 5 | VO | Voice with < 10 ms latency and jitter |
| 6 | IC | Internetwork control |
| 7 (highest) | NC | Network control |

The ToS field is an 8-bit field where only the first 3 bits of the ToS field, referred to as *IP Precedence (IPP)*, are used for marking, and the rest of the bits are unused. IPP values, which range from 0 to 7, allow the traffic to be partitioned in up to six usable classes of service; IPP 6 and 7 are reserved for internal network use.

Newer standards have redefined the IPv4 ToS and the IPv6 Traffic Class fields as an 8-bit Differentiated Services (DiffServ) field. The DiffServ field uses the same 8 bits that were previously used for the IPv4 ToS and the IPv6 Traffic Class fields, and this allows it to be backward compatible with IP Precedence. The DiffServ field is composed of a 6-bit Differentiated Services Code Point (DSCP) field that allows for classification of up to 64 values (0 to 63) and a 2-bit Explicit Congestion Notification (ECN) field.

Packets are classified and marked to receive a particular per-hop forwarding behavior (that is, expedited, delayed, or dropped) on network nodes along their path to the destination. The DiffServ field is used to mark packets according to their classification into DiffServ Behavior Aggregates (BAs). A DiffServ BA is a collection of packets with the same DiffServ value crossing a link in a particular direction. *Per-hop behavior (PHB)* is the externally observable forwarding behavior (forwarding treatment) applied at a DiffServ-compliant node to a collection of packets with the same DiffServ value crossing a link in a particular direction (DiffServ BA).

Four PHBs have been defined and characterized for general use:

- **Class Selector (CS) PHB:** The first 3 bits of the DSCP field are used as CS bits. The CS bits make DSCP backward compatible with IP Precedence because IP Precedence uses the same 3 bits to determine class.
- **Default Forwarding (DF) PHB:** Used for best-effort service.
- **Assured Forwarding (AF) PHB:** Used for guaranteed bandwidth service.
- **Expedited Forwarding (EF) PHB:** Used for low-delay service.

### Trust Boundary

To provide an end-to-end and scalable QoS experience, packets should be marked by the endpoint or as close to the endpoint as possible. When an endpoint marks a frame or a packet with a CoS or DSCP value, the switch port it is attached to can be configured to accept or reject the CoS or DSCP values. If the switch accepts the values, it means it trusts the endpoint and does not need to do any packet reclassification and re-marking for the received endpoint's packets. If the switch does not trust the endpoint, it rejects the markings and reclassifies and re-marks the received packets with the appropriate CoS or DSCP value.

For example, consider a campus network with IP telephony and host endpoints; the IP phones by default mark voice traffic with a CoS value of 5 and a DSCP value of 46 (EF), while incoming traffic from an endpoint (such as a PC) attached to the IP phone's switch port is re-marked to a CoS value of 0 and a DSCP value of 0. Even if the endpoint is sending tagged frames with a specific CoS or DSCP value, the default behavior for Cisco IP phones is to not trust the endpoint and zero out the CoS and DSCP values before sending the frames to the switch. When the IP phone sends voice and data traffic to the switch, the switch can classify voice traffic as higher priority than the data traffic, thanks to the high-priority CoS and DSCP markings for voice traffic.

*Traffic policers and shapers* are traffic-conditioning QoS mechanisms used to classify traffic and enforce other QoS mechanisms such as rate limiting. They classify traffic in an identical manner but differ in their implementation:

- **Policers:** Drop or re-mark incoming or outgoing traffic that goes beyond a desired traffic rate.
- **Shapers:** Buffer and delay egress traffic rates that momentarily peak above the desired rate until the egress traffic rate drops below the defined traffic rate. If the egress traffic rate is below the desired rate, the traffic is sent immediately.

### Markdown

When a desired traffic rate is exceeded, a policer can take one of the following actions:

- Drop the traffic.
- Mark down the excess traffic with a lower priority.

Marking down excess traffic involves re-marking the packets with a lower-priority class value; for example, excess traffic marked with AFx1 should be marked down to AFx2 (or AFx3 if using two-rate policing). After marking down the traffic, congestion-avoidance mechanisms, such as DSCP-based weighted random early detection (WRED), should be configured throughout the network to drop AFx3 more aggressively than AFx2 and drop AFx2 more aggressively than AFx1.

Cisco IOS policers and shapers are based on token bucket algorithms. The following list includes definitions that are used to explain how token bucket algorithms operate:

- **Committed Information Rate (CIR):** The policed traffic rate, in bits per second (bps), defined in the traffic contract.
- **Committed Time Interval (Tc):** The time interval, in milliseconds (ms), over which the committed burst (Bc) is sent. Tc can be calculated with the formula Tc = (Bc [bits] / CIR [bps]) × 1000.
- **Committed Burst Size (Bc):** The maximum size of the CIR token bucket, measured in bytes, and the maximum amount of traffic that can be sent within a Tc. Bc can be calculated with the formula Bc = CIR × (Tc / 1000).
- **Token:** A single token represents 1 byte or 8 bits.
- **Token bucket:** A bucket that accumulates tokens until a maximum predefined number of tokens is reached (such as the Bc when using a single token bucket); these tokens are added into the bucket at a fixed rate (the CIR). Each packet is checked for conformance to the defined rate and takes tokens from the bucket equal to its packet size; for example, if the packet size is 1500 bytes, it takes 12,000 bits (1500 × 8) from the bucket. If there are not enough tokens in the token bucket to send the packet, the traffic conditioning mechanism can take one of the following actions:
  - Buffer the packets while waiting for enough tokens to accumulate in the token bucket (traffic shaping)
  - Drop the packets (traffic policing)
  - Mark down the packets (traffic policing)

There are different policing algorithms, including the following:

- Single-rate two-color marker/policer
- Single-rate three-color marker/policer (srTCM)
- Two-rate three-color marker/policer (trTCM)

There are many queuing algorithms available, but most of them are not adequate for modern rich-media networks carrying voice and high-definition video traffic because they were designed before these traffic types came to be. The legacy queuing algorithms that predate the MQC architecture include the following:

- **First-in, first-out queuing (FIFO):** FIFO involves a single queue where the first packet to be placed on the output interface queue is the first packet to leave the interface (first come, first served). In FIFO queuing, all traffic belongs to the same class.
- **Round robin:** With round robin, queues are serviced in sequence one after the other, and each queue processes one packet only. No queues starve with round robin because every queue gets an opportunity to send one packet every round. No queue has priority over others, and if the packet sizes from all queues are about the same, the interface bandwidth is shared equally across the round robin queues. A limitation of round robin is it does not include a mechanism to prioritize traffic.
- **Weighted round robin (WRR):** WRR was developed to provide prioritization capabilities for round robin. It allows a weight to be assigned to each queue, and based on that weight, each queue effectively receives a portion of the interface bandwidth that is not necessarily equal to the other queues' portions.
- **Custom queuing (CQ):** CQ is a Cisco implementation of WRR that involves a set of 16 queues with a round-robin scheduler and FIFO queueing within each queue. Each queue can be customized with a portion of the link bandwidth for each selected traffic type. If a particular type of traffic is not using the bandwidth reserved for it, other traffic types may use the unused bandwidth. CQ causes long delays and also suffers from all the same problems as FIFO within each of the 16 queues that it uses for traffic classification.
- **Priority queuing (PQ):** With PQ, a set of four queues (high, medium, normal, and low) are served in strict-priority order, with FIFO queueing within each queue. The high-priority queue is always serviced first, and lower-priority queues are serviced only when all higher-priority queues are empty. For example, the medium queue is serviced only when the high-priority queue is empty. The normal queue is serviced only when the high and medium queues are empty; finally, the low queue is serviced only when all the other queues are empty. At any point in time, if a packet arrives for a higher queue, the packet from the higher queue is processed before any packets in lower-level queues. For this reason, if the higher-priority queues are continuously being serviced, the lower-priority queues are starved.
- **Weighted fair queuing (WFQ):** The WFQ algorithm automatically divides the interface bandwidth by the number of flows (weighted by IP Precedence) to allocate bandwidth fairly among all flows. This method provides better service for high-priority real-time flows but can't provide a fixed-bandwidth guarantee for any particular flow.

The current queuing algorithms recommended for rich-media networks (and supported by MQC) combine the best features of the legacy algorithms. These algorithms provide real-time, delay-sensitive traffic bandwidth and delay guarantees while not starving other types of traffic. The recommended queuing algorithms include the following:

- **Class-based weighted fair queuing (CBWFQ):** CBWFQ enables the creation of up to 256 queues, serving up to 256 traffic classes. Each queue is serviced based on the bandwidth assigned to that class. It extends WFQ functionality to provide support for user-defined traffic classes. With CBWFQ, packet classification is done based on traffic descriptors such as QoS markings, protocols, ACLs, and input interfaces. After a packet is classified as belonging to a specific class, it is possible to assign bandwidth, weight, queue limit, and maximum packet limit to it. The bandwidth assigned to a class is the minimum bandwidth delivered to the class during congestion. The queue limit for that class is the maximum number of packets allowed to be buffered in the class queue. After a queue has reached the configured queue limit, excess packets are dropped. CBWFQ by itself does not provide a latency guarantee and is only suitable for non-real-time data traffic.
- **Low-latency queuing (LLQ):** LLQ is CBWFQ combined with priority queueing (PQ) and it was developed to meet the requirements of real-time traffic, such as voice. Traffic assigned to the strict-priority queue is serviced up to its assigned bandwidth before other CBWFQ queues are serviced. All real-time traffic should be configured to be serviced by the priority queue. Multiple classes of real-time traffic can be defined, and separate bandwidth guarantees can be given to each, but a single priority queue schedules all the combined traffic. If a traffic class is not using the bandwidth assigned to it, it is shared among the other classes. This algorithm is suitable for combinations of real-time and non-real-time traffic. It provides both latency and bandwidth guarantees to high-priority real-time traffic. In the event of congestion, real-time traffic that goes beyond the assigned bandwidth guarantee is policed by a congestion-aware policer to ensure that the non-priority traffic is not starved.

The Cisco implementation of RED is known as weighted RED (WRED). The difference between RED and WRED is that the randomness of packet drops can be manipulated by traffic weights denoted by either IP Precedence (IPP) or DSCP. Packets with a lower IPP value are dropped more aggressively than are higher IPP values; for example, IPP 3 would be dropped more aggressively than IPP 5 or DSCP, AFx3 would be dropped more aggressively than AFx2, and AFx2 would be dropped more aggressively than AFx1.

# Chapter 15

## Network Time Protocol

RFC 958 introduced Network Time Protocol (NTP), which 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 then queries the NTP server for its time and updates its time based on the response. Because NTP is considered an application, the query can occur over multiple hops, requiring NTP clients to identify the time accuracy based on messages with other routers.

The NTP synchronization process is not fast. In general, an NTP client can synchronize a large time discrepancy to within a couple seconds of accuracy with a few cycles of polling an NTP server. However, gaining accuracy of tens of milliseconds requires hours or days of comparisons. In some ways, the time of the NTP clients drifts toward the time of the NTP server.

NTP uses the concept of stratums to identify the accuracy of the time clock source. NTP servers that are 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.

### Stratum Preference

An NTP client can be configured with multiple NTP servers. The device will use only the NTP server with the lowest stratum. The top portion of Figure 15-2 shows R4 with two NTP sessions: one session with R1 and another with R3.

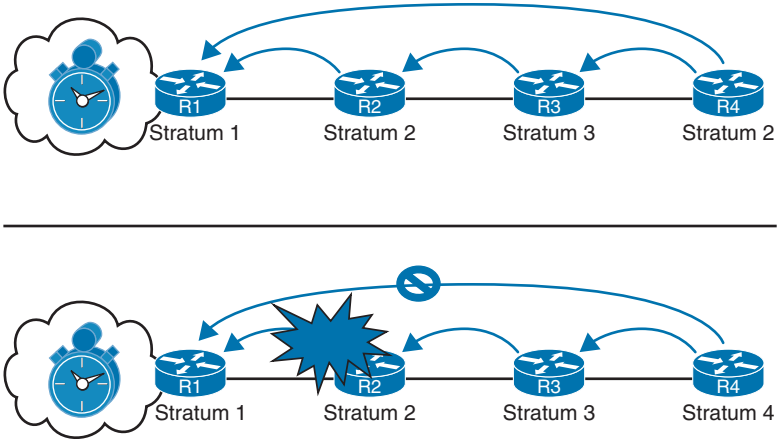



**Figure 15-2** *NTP Stratum Preferences*

In the topology shown in Figure 15-2, R4 will always use R1 for synchronizing its time because it is a stratum 1 server. If R2 crashes, as shown at the bottom of Figure 15-2, preventing R4 from reaching R1, it synchronizes with R3's time (which may or may not be different due to time drift) and turns into a stratum 4 time device. When R2 recovers, R4 synchronizes with R1 and becomes a stratum 2 device again.

### NTP Peers

Within the NTP client architecture, the NTP client changes its time to the time of the NTP server. The NTP server does not change its time to reflect the clients. Most enterprise organizations (such as universities, governments, and pool.ntp.org) use an external NTP server. A common scenario is to designate two devices to query a different external NTP source and then to peer their local stratum 2 NTP devices.

NTP peers act as clients and servers to each other, in the sense that they try to blend their time to each other. The NTP peer model is intended for designs where other devices can act as backup devices for each other and use different primary reference sources.

The deployment of first-hop redundancy protocols (FHRPs) solves the problem of hosts configuring multiple gateways. FHRPs work by creating a virtual IP (VIP) gateway instance that is shared between the Layer 3 devices. This book covers the following FHRPs:

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

### Hot Standby Router Protocol

Hot Standby Routing Protocol (HSRP) is a Cisco proprietary protocol that provides transparent failover of the first-hop device, which typically acts as a gateway to the hosts.

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 to allow a more preferred router to take the active router status from an inferior active HSRP router. Enable preemption with the command **standby** *instance-id* **preempt**.

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

**Step 4.** Define the HSRP MAC Address (Optional).

The MAC address can be set with the command **standby** *instance-id* **mac-address** *mac-address*. Most organizations accept the automatically generated MAC address, but in some migration scenarios, the MAC address needs to be statically set to ease transitions when the hosts may have a different MAC address in their ARP table.

**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 provides the capability to link object tracking to priority. For example, assume that traffic should flow through SW2's WAN connection whenever feasible. Traffic can be routed by SW3 to SW2 and then on to SW2's WAN connection; however, making SW2 the VIP gateway streamlines the process. But when SW2 loses its link to the WAN, it should move the HSRP active speaker role to SW3.

## Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) is an industry standard and operates similarly to HSRP. The behavior of VRRP is so close to that of HSRP that the following differences should be noted:

- 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 0000.5e00.01*xx*, 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

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

**Step 1.** Define the VRRP instance by using the command **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. Do so by using the command **vrrp** *instance-id* **track** *object-id* **decrement** *decrement-value*. The decrement value should be high enough so that when it is removed from the priority, the value is lower than that of the other VRRP router.

**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*}}.

The newer version of IOS XE software provides configuration of VRRP in a multi-address format that is hierarchical. The steps for configuring hierarchical VRRP are as follows:

**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**}. This places the configuration prompt into the VRRP group for additional configuration.

**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*. The priority is a value between 0 and 255.

**Step 6.** (Optional) Enable object tracking so that the priority is decremented when the object is false. Do so by using the command **track** *object-id* **decrement** *decrement-value*. The decrement value should be high enough so that when it is removed from the priority, the value is lower than that of the other VRRP router.

## Global Load Balancing Protocol

As the name suggests, Gateway Load Balancing Protocol (GLBP) provides gateway redundancy and load-balancing capability to a network segment. It provides redundancy with an active/standby gateway, and it provides load-balancing capability by ensuring that each member of the GLBP group takes care of forwarding the traffic to the appropriate gateway.

The GLBP contains two roles:

- **Active virtual gateway (AVG):** The participating routers elect one AVG per GLBP group to respond to initial ARP requests for the VIP. For example, when a local PC sends an ARP request for the VIP, the AVG is responsible for replying to the ARP request with the virtual MAC address of the AVF.
- **Active virtual forwarder (AVF):** The AVF routes traffic received from assigned hosts. A unique virtual MAC address is created and assigned by the AVG to the AVFs. The AVF is assigned to a host when the AVG replies to the ARP request with the assigned AVF's virtual MAC address. ARP replies are unicast and are not heard by other hosts on that broadcast segment. When a host sends traffic to the virtual AVF MAC, the current router is responsible for routing it to the appropriate network. The AVFs are also recognized as *Fwd* instances on the routers.

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 to allow for a more preferred router to take the active virtual gateway status from an inferior active GLBP router. Preemption is enabled 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*}}.

By default, GLBP balances the load of traffic in a round-robin fashion, as highlighted in Example 15-22. However, GLBP supports three methods of load balancing traffic:

- **Round robin:** Uses each virtual forwarder MAC address to sequentially reply for the virtual IP address.
- **Weighted:** Defines weights to each device in the GLBP group to define the ratio of load balancing between the devices. This allows for a larger weight to be assigned to bigger routers that can handle more traffic.
- **Host dependent:** Uses the host MAC address to decide to which virtual forwarder MAC to redirect the packet. This method ensures that the host uses the same virtual MAC address as long as the number of virtual forwarders does not change within the group.

Connectivity is established with Network Address Translation (NAT). Basically, 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.

Four important terms are related to NAT:

- **Inside local:** The actual private IP address assigned to a device on the inside network.
- **Inside global:** The public IP address that represents one or more inside local IP addresses to the 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.
- **Outside global:** The public IP address assigned to a host on the outside network. This IP address must be reachable by the outside network.

Three types of NAT are commonly used today:

- **Static NAT:** Provides a static one-to-one mapping of a local IP address to a global IP address.
- **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 needs a mechanism to identify the specific private IP address for the return network traffic. 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.

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*.

The NAT translation table consists of static and dynamic entries. The NAT translation table is displayed with the command **show ip nat translations**. Example 15-30 shows R5's NAT translation table after R7 initiated a Telnet session to R1. There are two entries:

- The first entry is the dynamic entry correlating to the Telnet session. The inside global, inside local, outside local, and outside global fields all contain values. Notice that the ports in this entry correlate with the ports in Example 15-29.
- The second entry is the inside static NAT entry that was configured.

The NAT translation follows these steps:

1. As traffic enters on R5's Gi0/1 interface, 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.
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.
3. R5 translates (that is, changes) the packet's source IP address from 10.78.9.7 to 10.45.1.7.
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.
5. As the packet enters on R5's Gi0/0 interface, R5 is aware that the Gi0/0 interface is an outside NAT interface and checks the NAT table for an entry.
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.
7. R5 routes the packet out the Gi0/1 interface toward R6.

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 by using the command **ip nat inside**.

**Step 3.** Configure the outside static NAT entry by using the command **ip nat outside source static** *outside-global-ip outside-local-ip* [**add-route**]. The router performs a route lookup first for the *outside-local-ip* address, and a route must exist for that network to forward packets out of the outside interface before NAT occurs. The optional **add-route** keyword adds the appropriate static route entry automatically.

Pooled NAT can operate as inside NAT or outside NAT. In this section, we focus on inside pooled NAT. 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 by using the command **ip nat inside**.

**Step 3.** Specify which 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*.

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* | *}, which removes all existing translations and could interrupt traffic flow on active sessions as they might be assigned new global 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 involves the following steps:

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

**Step 2.** Configure the inside interface by using 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. Using a user-friendly name may be simplest from an operational support perspective.

**Step 4.** Configure Port Address Translation by using the command the command **ip nat inside source list** *acl* {**interface** *interface-id* | **pool** *nat-pool-name*} **overload**. Specifying an interface involves using the primary IP address assigned to that interface. Specifying a NAT pool requires the creation of the NAT pool, as demonstrated earlier, and involves using those IP addresses as the global address.

# Chapter 16

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 was originally created to provide transport for non-routable legacy protocols such as *Internetwork Packet Exchange (IPX)* across an IP network and is now more commonly used as an overlay for IPv4 and IPv6. GRE tunnels have many uses. For example, they can be used to tunnel traffic through a firewall or an ACL or to connect discontiguous networks, and they can even be used as networking duct tape for bad routing designs. Their most important application is that they can be used to create VPNs.

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 interface indicates the interface that will be used for encapsulation and de-encapsulation of the GRE tunnel. The tunnel source can be a physical interface or a loopback interface. A loopback interface can provide reachability if one of the transport interfaces fails.

**Step 3.** Identify the remote destination IP address by using the interface parameter command **tunnel destination** *ip-address*. The tunnel destination is the remote router's underlay IP address toward which the local router sends GRE packets.

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

**Step 5.** (Optional) Define the tunnel bandwidth. Virtual interfaces do not have the concept of latency and need to have a reference bandwidth configured so that routing protocols that use bandwidth for best-path calculation can make an intelligent decision. Bandwidth is also used for *quality of service (QoS)* configuration on the interface. 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. Tunnel interfaces are GRE *point-to-point (P2P)* by default, and the line protocol enters an up state when the router detects that a route to the tunnel destination exists in the routing table. If the tunnel destination is not in the routing table, the tunnel interface (line protocol) enters a down state.

Tunnel keepalives ensure that bidirectional communication exists between tunnel endpoints to keep the line protocol up. Otherwise, the router must rely on routing protocol timers to detect a dead remote endpoint.

Keepalives are configured with the interface parameter command **keepalive** [*seconds* [*retries*]]. The default timer is 10 seconds, with three retries.

**Step 7.** (Optional) Define the IP *maximum transmission unit (MTU)* for the tunnel interface. The GRE tunnel adds a minimum of 24 bytes to the packet size to accommodate the headers that are added to the packet. 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*.

*IPsec* is a framework of open standards for creating highly secure virtual private networks (VPNs) using various protocols and technologies for secure communication across unsecure networks, such as the Internet. IPsec tunnels provide the security services listed in Table 16-3.

**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)<br>■ Digital certificates |
| Data confidentiality | Protects data from eavesdropping attacks through encryption algorithms. Changes plaintext into encrypted ciphertext. | ■ Data Encryption Standard (DES)<br>■ Triple DES (3DES)<br>■ Advanced Encryption Standard (AES)<br>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) functions:<br>■ Message Digest 5 (MD5) algorithm<br>■ Secure Hash Algorithm (SHA-1)<br>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. |

### Authentication Header

The IP authentication header provides data integrity, authentication, and protection from hackers replaying packets. The authentication header ensures that the original data packet (before encapsulation) has not been modified during transport on the public network. It creates a digital signature similar to a checksum to ensure that the packet has not been modified, using protocol number 51 located in the IP header. The authentication header does not support encryption (data confidentiality) and *NAT traversal (NAT-T)*, and for this reason, its use is not recommended, unless authentication is all that is desired.

### Encapsulating Security Payload

Encapsulating Security Payload (ESP) provides data confidentiality, authentication, and protection from hackers replaying packets. Typically, *payload* refers to the actual data minus any headers, but in the context of ESP, the payload is the portion of the original packet that is encapsulated within the IPsec headers. 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. ESP uses the protocol number 50, located in the IP header. Unlike the authentication header, ESP does provide data confidentiality and supports NAT-T.

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

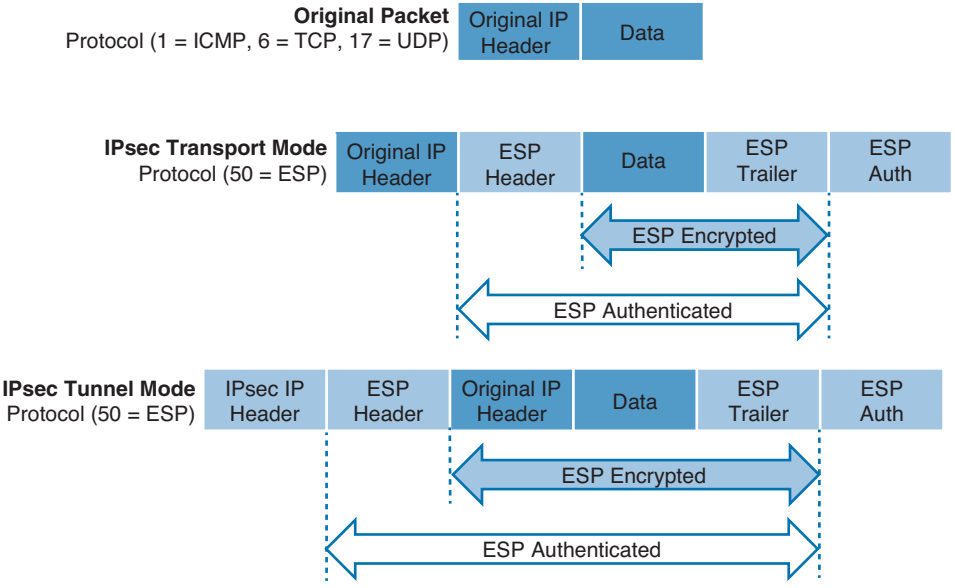



**Figure 16-3** *IPsec Transport and Tunnel Encapsulation*

IPsec supports the following 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.
- **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. A DH group refers to the length of the key (modulus size) to use for a DH key exchange. For example, group 1 uses 768 bits, group 2 uses 1024, and group 5 uses 1536, where the larger the modulus, the more secure it is. The purpose of DH is to generate shared secret symmetric keys that are used by the two VPN peers for symmetrical algorithms, such as AES. The DH exchange itself is asymmetrical and CPU intensive, and the resulting shared secret keys that are generated are symmetrical. Cisco recommends avoiding DH groups 1, 2, and 5 and instead use DH groups 14 and higher.
- **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

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. When such a transform set is found, it is selected and applied to the IPsec SAs on both peers. Table 16-4 shows the allowed transform set combinations.

**Table 16-4** Allowed Transform Set Combinations

| 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 |
| ESP encryption transform (only one allowed) | esp-aes | ESP with the 128-bit AES encryption algorithm |
| | esp-gcm<br>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<br>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 |
| 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 |

### Internet Key Exchange

*Internet Key Exchange (IKE)* is a protocol that performs authentication between two endpoints to establish security associations (SAs), also known as IKE tunnels. These security associations, or tunnels, are used to carry control plane and data plane traffic for IPsec. There are two versions of IKE: IKEv1 (specified in RFC 2409) and IKEv2 (specified in RFC 7296). IKEv2 was developed to overcome the limitations of IKEv1 and provides many improvements over IKEv1's implementation. For example, it supports EAP (certificate-based authentication), has anti-DoS capabilities, and needs fewer messages to establish an IPsec SA. Understanding IKEv1 is still important because some legacy infrastructures have not yet migrated to IKEv2 or have devices or features that don't support IKEv2.

### IKEv1

*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. It is designed to support many different kinds of key exchanges. ISAKMP uses UDP port 500 for communication between peers.

IKE is the implementation of ISAKMP using the Oakley and Skeme key exchange techniques. Oakley provides perfect forward secrecy (PFS) for keys, identity protection, and authentication; Skeme provides anonymity, repudiability, and quick key refreshment. For Cisco platforms, IKE is analogous to ISAKMP, and the two terms are used interchangeably.

### IKEv2

IKEv2 is an evolution of IKEv1 that includes many changes and improvements that simplify it and make it more efficient. One of the major changes has to do with the way the SAs are established. In IKEv2, communications consist of request and response pairs called *exchanges* and sometimes just called *request/response pairs*.

The first exchange, IKE_SA_INIT, negotiates cryptographic algorithms, exchanges nonces, and performs a Diffie-Hellman exchange. This is the equivalent to IKEv1's first two pairs of messages MM1 to MM4 but done as a single request/response pair.

The second exchange, 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 but done as a single request/response pair.

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.

If additional IPsec SAs are required in IKEv2, it uses just two messages (a request/response pair) with a CREATE_CHILD_SA exchange, whereas IKEv1 would require three messages with quick mode.

Since the IKEv2 SA exchanges are completely different from those of IKEv1, they are incompatible with each other.

**Table 16-5** Major Differences Between IKEv1 and IKEv2

| IKEv1 | IKEv2 |
|---|---|
| **Exchange Modes** | |
| Main mode<br>Aggressive mode<br>Quick mode | IKE Security Association Initialization (SA_INIT)<br>IKE_Auth<br>CREATE_CHILD_SA |
| **Minimum Number of Messages Needed to Establish IPsec SAs** | |
| Nine with main mode<br>Six with aggressive mode | Four |
| **Supported Authentication Methods** | |
| Pre-Shared Key (PSK)<br>Digital RSA Certificate (RSA-SIG)<br>Public key<br>Both peers must use the same authentication method. | Pre-Shared Key<br>Digital RSA Certificate (RSA-SIG)<br>Elliptic Curve Digital Signature Certificate (ECDSA-SIG)<br>Extensible Authentication Protocol (EAP)<br>Asymmetric authentication is supported. Authentication method can be specified during the IKE_AUTH exchange. |
| **Next Generation Encryption (NGE)** | |
| Not supported | AES-GCM (Galois/Counter Mode) mode<br>SHA-256<br>SHA-384<br>SHA-512<br>HMAC-SHA-256<br>Elliptic Curve Diffie-Hellman (ECDH) ECDH-384<br>ECDSA-384 |
| **Attack Protection** | |
| MitM protection<br>Eavesdropping protection | MitM protection<br>Eavesdropping protection<br>Anti-DoS protection |

**Table 16-6** Cisco IPsec VPN Solutions

| Features and Benefits | Site-to-Site IPsec VPN | Cisco DMVPN | Cisco GET-VPN | FlexVPN | Remote Access VPN |
|---|---|---|---|---|---|
| Product interoperability | Multivendor | Cisco only | Cisco only | Cisco only | Cisco only |
| Key exchange | IKEv1 and IKEv2 | IKEv1 and IKEv2 (both optional) | IKEv1 and IKEv2 | IKEv2 only | TLS/DTLS and IKEv2 |
| Scale | Low | Thousands for hub-and-spoke; hundreds for partially meshed spoke- to-spoke connections | Thousands | Thousands | Thousands |
| Topology | Hub-and-spoke; small-scale meshing as manageability allows | Hub-and-spoke; on-demand spoke- to-spoke partial mesh; spoke-to-spoke connections automatically terminated when no traffic present | Hub-and-spoke; any-to-any | Hub-and-spoke; any-to-any, remote access | Remote access |
| Routing | Not supported | Supported | Supported | Supported | Not supported |
| QoS | Supported | Supported | Supported | Native support | Supported |
| Multicast | Not supported | Tunneled | Natively supported across MPLS and private IP networks | Tunneled | Not supported |
| Non-IP protocols | Not supported | Not supported | Not supported | Not supported | Not supported |
| Private IP addressing | Supported | Supported | Requires use of GRE or DMVPN with Cisco GET-VPN to support private addresses across the Internet | Supported | Supported |

| Features and Benefits | Site-to-Site IPsec VPN | Cisco DMVPN | Cisco GET-VPN | FlexVPN | Remote Access VPN |
|---|---|---|---|---|---|
| High availability | Stateless failover | Routing | Routing | Routing IKEv2-based dynamic route distribution and server clustering | Not supported |
| Encapsulation | Tunneled IPsec | Tunneled IPsec | Tunnel-less IPsec | Tunneled IPsec | Tunneled IPsec/TLS |
| Transport network | Any | Any | Private WAN/MPLS | Any | Any |

VTI over IPsec encapsulates IPv4 or IPv6 traffic without the need for an additional GRE header, while GRE over IPsec first encapsulates traffic within GRE and a new IP header before encapsulating the resulting GRE/IP packet in IPsec transport mode. Figure 16-4 illustrates a comparison of GRE packet encapsulation and IPsec tunnel mode with a VTI.

There are two different ways to encrypt traffic over a GRE tunnel:

- Using crypto maps
- Using tunnel IPsec profiles

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*}

This access list identifies traffic that needs to be protected by IPsec. It is used to match all traffic that passes through the GRE tunnel.

**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** | **aes 192** | **aes 256**}

**hash** {**sha** | **sha256** | **sha384** | **md5**}

**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}

**group** {**1** | **2** | **5** | **14** | **15** | **16** | **19** | **20** | **24**}

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

The DES and 3DES encryption algorithms are no longer recommended. DES is the default encryption used, so it is recommended to choose one of the AES encryption algorithms

The MD5 hash is no longer recommended. The default is SHA.

Authentication allows for public keys (**rsa-encr**), digital certificates (**rsa-sig**), or PSK (**pre-share**) to be used.

The **group** command indicates the DH group, where 1 is the default. It is recommended to choose one of the DH groups higher than 14. The following DH groups are available:

- **1:** 768-bit DH (no longer recommended)
- **2:** 1024-bit DH (no longer recommended)
- **5:** 1536-bit DH (no longer recommended)
- **14:** The 2048-bit DH group
- **15:** The 3072-bit DH group
- **16:** The 4096-bit DH group
- **19:** The 256-bit ECDH group
- **20:** The 384-bit ECDH group
- **24:** The 2048-bit DH/DSA group

**Step 3.** Configure PSK by using the command **crypto isakmp key** *keystring* **address** *peer-address* [*mask*]. The *keystring* should match on both peers. For

*peer-address* [*mask*], the value 0.0.0.0 0.0.0.0 can be used to allow 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. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow. **mode** indicates the IPsec tunnel mode to be either tunnel or transport.

**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-set-
name2...transform-set-name6]
```

*acl-name* is the crypto ACL defined in step 1, which determines the traffic that should be protected by IPsec. The command **set peer** can be repeated for multiple remote peers. The command **set transform-set** specifies the transform sets to be negotiated. List multiple transform sets in priority order (highest priority first).

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

The steps to enable IPsec over GRE using IPsec profiles are as follows:

**Step 1.** Configure an ISAKMP policy for IKE SA by entering 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 | aes 192 | aes 256}
hash {sha | sha256 | sha384 | md5}
authentication {rsa-sig | rsa-encr | pre-share}
group {1 | 2 | 5 | 14 | 15 | 16 | 19 | 20 | 24}
```

**Step 2.** Configure PSK by using the command **crypto isakmp key** *keystring* **address** *peer-address* [*mask*]. *keystring* should match on both peers.

**Step 3.** 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 the transform set configuration mode, enter the command **mode** [**tunnel** | **transport**] to specify tunnel or transport modes. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow. **mode** indicates the IPsec tunnel mode to be either tunnel or transport. To avoid double encapsulation (from GRE and IPsec), transport mode should be chosen.

**Step 4.** Create an IPsec profile and enter IPsec profile configuration mode by entering the command **crypto ipsec profile** *ipsec-profile-name*. In IPsec profile configuration mode, specify the transform sets to be negotiated by using the command **set transform-set** *transform-set-name* [*transform-set-name2*... *transform-set-name6*]. List multiple transform sets in priority order (highest priority first).

**Step 5.** Apply the IPsec profile to a tunnel interface by using the command **tunnel protection** *ipsec profile profile-name*.

### Site-to-Site VTI over IPsec

The steps to enable a VTI over IPsec are very similar to those for GRE over IPsec configuration using IPsec profiles. The only difference is the addition of the command **tunnel mode ipsec** {**ipv4** | **ipv6**} under the GRE tunnel interface to enable VTI on it and to change the packet transport mode to tunnel mode. To revert to GRE over IPsec, the command **tunnel mode gre** {**ip** | **ipv6**} is used.

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:

- **Aggregation issues:** Many routes on the Internet routing table are provider-independent routes that are non-aggregable, and this is part of the reason the Internet routing table is so large and still growing.
- **Traffic engineering:** A common practice for ingress traffic engineering into a site is to inject more specific routes into the Internet, which exacerbates the Internet routing table aggregation/scalability problems.
- **Multihoming:** Proper multihoming to the Internet requires a full Internet routing table (785,000 IPv4 routes at the time of writing). If a small site requires multihoming, a powerful router is needed to be able to handle the full routing table (with large memory, powerful CPUs, more TCAM, more power, cooling, and so on), which can be cost-prohibitive for deployment across small sites.
- **Routing instability:** Internet route instability (also known as *route churn*) causes intensive router CPU and memory consumption, which also requires powerful routers.

Even though LISP was created to address the routing scalability problems of the Internet, it is also being implemented in other types of environments, such as data centers, campus networks, branches, next-gen WANs, and service provider cores. In addition, it can also serve for applications or use cases such as mobility, network virtualization, Internet of Things (IoT), IPv4-to-IPv6 transition, and traffic engineering.

Following are the definitions for the LISP architecture components illustrated in Figure 16-5.

- **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.
- **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 Routing Architecture

In traditional routing architectures, an endpoint IP address represents the endpoint's identity and location. If the location of the endpoint changes, its IP address also changes. LISP separates IP addresses into endpoint identifiers (EIDs) and routing locators (RLOCs). This way, 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 MR, as illustrated in Figure 16-6. This makes it a very efficient and scalable on-demand routing protocol because it is based on a pull model, where only the routing information that is necessary is requested (as opposed to the push model of traditional routing protocols, such as BGP and OSPF, that push all the routes to the routers—including unnecessary ones).

### LISP Data Plane

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. The original IP header and data are preserved; this is referred to as the *inner header*. Between the outer UDP header and the inner header, a LISP shim header is included to encode information necessary to enable forwarding plane functionality, such as network virtualization. Figure 16-7 illustrates the LISP packet frame format.

The following steps describe the map registration process illustrated in Figure 16-8:

**Step 1.** The ETR sends a map register message to the MS to register its associated EID prefix 10.1.2.0/24. In addition to the EID prefix, the message includes the RLOC IP address 100.64.2.2 to be used by the MS when forwarding map requests (re-formatted as encapsulated map requests) received through the mapping database system.

An ETR by default responds to map request messages, but in a map register message it may request that the MS answer map requests on its behalf by setting the proxy map reply flag (P-bit) in the message.

**Step 2.** The MS sends a map notify message to the ETR to confirm that the map register has been received and processed. A map notify message uses UDP port 4342 for both source and destination.

The following steps outline the map request and reply process illustrated in Figure 16-9:

**Step 1.** The endpoint in LISP Site 1 (host1) sends a DNS request to resolve the IP address of the endpoint in LISP Site 2 (host2.cisco.com). The DNS server replies with the IP address 10.1.2.2, which is the destination EID. host1 sends IP packets with destination IP 10.1.2.2 to its default gateway, which for this example is the ITR router. If host1 was not directly connected to the ITR, the IP packets would be forwarded through the LISP site as normal IP packets, using traditional routing, until they reached the ITR.

**Step 2.** The ITR receives the packets from host1 destined to 10.1.2.2. It performs a FIB lookup and evaluates the following forwarding rules:

- Did the packet match a default route because there was no route found for 10.1.2.2 in the routing table?
  - If yes, continue to next step.
  - If no, forward the packet natively using the matched route.
- Is the source IP a registered EID prefix in the local map cache?
  - If yes, continue to next step.
  - If no, forward the packet natively.

**Step 3.** The ITR sends an encapsulated map request to the MR for 10.1.2.2. A map request message uses the UDP destination port 4342, and the source port is chosen by the ITR.

**Step 4.** Because the MR and MS functionality is configured on the same device, the MS mapping database system forwards the map request to the authoritative (source of truth) ETR. If the MR and MS functions were on different devices, the MR would forward the encapsulated map request packet to the MS as received from the ITR, and the MS would then forward the map request packet to the ETR.

**Step 5.** The ETR sends to the ITR a map reply message that includes an EID-to-RLOC mapping 10.1.2.2 Ð 100.64.2.2. The map reply message uses the UDP source port 4342, and the destination port is the one chosen by the ITR in the map request message. An ETR may also request that the MS answer map requests on its behalf by setting the proxy map reply flag (P-bit) in the map register message.

**Step 6.** The ITR installs the EID-to-RLOC mapping in its local map cache and programs the FIB; it is now ready to forward LISP traffic.

The following steps describe the encapsulation and de-encapsulation process illustrated in Figure 16-10:

**Step 1.** The ITR receives a packet from EID host1 (10.1.1.1) destined to host2 (10.2.2.2).

**Step 2.** The ITR performs a FIB lookup and finds a match. It encapsulates the EID packet and adds an outer header with the RLOC IP address from the ITR as the source IP address and the RLOC IP address of the ETR as the destination IP address. The packet is then forwarded using UDP destination port 4341 with a tactically selected source port in case ECMP load balancing is necessary.

**Step 3.** ETR receives the encapsulated packet and de-encapsulates it to forward it to host2.

The following steps describe the proxy ETR process illustrated in Figure 16-11:

**Step 1.** host1 perform a DNS lookup for www.cisco.com. It gets a response form the DNS server with IP address 100.64.254.254 and starts forwarding packets to the ITR with the destination IP address 100.64.254.254.

**Step 2.** The ITR sends a map request to the MR for 100.64.254.254

**Step 3.** The mapping database system responds with a negative map reply that includes a calculated non-LISP prefix for the ITR to add it to its mapping cache and FIB.

**Step 4.** The ITR can now start sending LISP-encapsulated packets to the PETR.

**Step 5.** The PETR de-encapsulates the traffic and sends it to www.cisco.com.

The following steps describe the proxy ITR process illustrated in Figure 16-12:

**Step 1.** Traffic from www.cisco.com is received by the PITR with the destination IP address 10.1.1.1 from host1.cisco.com.

**Step 2.** The PITR sends a map request to the MR for 10.1.1.1.

**Step 3.** The mapping database system forwards the map request to the ETR.

**Step 4.** The ETR sends a map reply to the PITR with the EID-to-RLOC mapping 10.1.1.1 Ð 100.64.1.1.

**Step 5.** The PITR LISP-encapsulates the packets and starts forwarding them to the ETR.

**Step 6.** The ETR receives the LISP-encapsulated packets, de-encapsulates them, and sends them to host1.

VXLAN is an overlay data plane encapsulation scheme that was developed to address the various issues seen in traditional Layer 2 networks. It extends Layer 2 and Layer 3 overlay networks over a Layer 3 underlay network, using MAC-in-IP/UDP tunneling. Each overlay is termed a VXLAN *segment*.

Unlike the VLAN ID, which has only 12 bits and allows for 4000 VLANs, 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.

The 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. VTEPs are entities that originate or terminate VXLAN tunnels. They map Layer 2 and Layer 3 packets to the VNI to be used in the overlay network. 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. It is also used for VXLAN traffic encapsulation and de-encapsulation.

The VXLAN standard defines VXLAN as a data plane protocol, but it does not define a VXLAN control plane; it was left open to be used with any control plane. Currently four different VXLAN control and data planes are supported by Cisco devices:

- VXLAN with Multicast underlay
- VXLAN with static unicast VXLAN tunnels
- VXLAN with MP-BGP EVPN control plane
- VXLAN with LISP control plane

MP-BGP EVPN and Multicast are the most popular control planes used for data center and private cloud environments. For campus environments, VXLAN with a LISP control plane is the preferred choice.

Cisco Software Defined Access (SD-Access) is an example of an implementation of VXLAN with the LISP control plane. An interesting fact is that the VXLAN specification originated from a Layer 2 LISP specification (draft-smith-lisp-layer2-00) that aimed to introduce Layer 2 segmentation support to LISP. The VXLAN specification introduced the term *VXLAN* in lieu of *Layer 2 LISP* and didn't port over some of the fields from the Layer 2 LISP specification into the VXLAN specification. The minor differences between the Layer 2 LISP specification and the VXLAN specification headers are illustrated in Figure 16-15. Fields that were not ported over from Layer 2 LISP into VXLAN were reserved for future use.

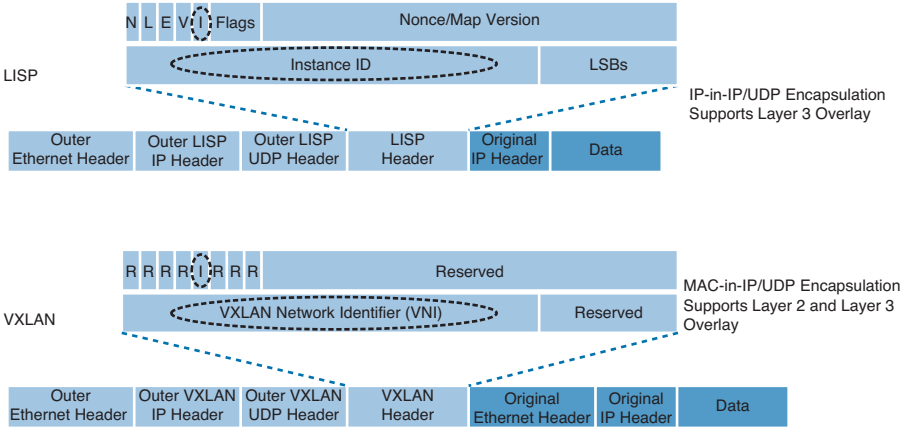



**Figure 16-15** *LISP and VXLAN Packet Format Comparison*

# Chapter 17

The *decibel (dB)* is a handy function that uses logarithms to compare one absolute measurement to another. It was originally developed to compare sound intensity levels, but it applies directly to power levels, too. After each power value has been converted to the same logarithmic scale, the two values can be subtracted to find the difference. The following equation is used to calculate a dB value, where P1 and P2 are the absolute power levels of two sources:

$$dB = 10(log_{10}P2 - log_{10}P1)$$

There are three cases where you can use mental math to make power-level comparisons using dB. By adding or subtracting fixed dB amounts, you can compare two power levels through multiplication or division. You should memorize the following three laws, which are based on dB changes of 0, 3, and 10, respectively:

- **Law of Zero:** A value of 0 dB means that the two absolute power values are equal.

  If the two power values are equal, the ratio inside the logarithm is 1, and the $\log_{10}(1)$ is 0. This law is intuitive; if two power levels are the same, one is 0 dB greater than the other.

- **Law of 3s:** A value of 3 dB means that the power value of interest is double the reference value; a value of –3 dB means the power value of interest is half the reference.

  When P2 is twice P1, the ratio is always 2. Therefore, $10\log_{10}(2) = 3$ dB.

  When the ratio is 1/2, $10\log_{10}(1/2) = -3$ dB.

  The Law of 3s is not very intuitive, but is still easy to learn. Whenever a power level doubles, it increases by 3 dB. Whenever it is cut in half, it decreases by 3 dB.

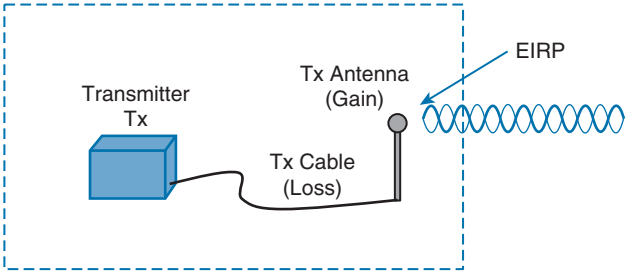
- **Law of 10s:** A value of 10 dB means that the power value of interest is 10 times the reference value; a value of –10 dB means the power value of interest is 1/10 of the reference.

  When P2 is 10 times P1, the ratio is always 10. Therefore, $10\log_{10}(10) = 10$ dB.

  When P2 is one tenth of P1, then the ratio is 1/10 and $10\log_{10}(1/10) = -10$ dB.

  The Law of 10s is intuitive because multiplying or dividing by 10 adds or subtracts 10 dB, respectively.

EIRP is a very important parameter because it is regulated by government agencies in most countries. In those cases, a system cannot radiate signals higher than a maximum allowable EIRP. To find the EIRP of a system, simply add the transmitter power level to the antenna gain and subtract the cable loss, as illustrated in Figure 17-20.




EIRP = Tx Power – Tx Cable + Tx Antenna

**Figure 17-20** *Calculating EIRP*

For reference, the free space path loss (FSPL) in dB can be calculated according to the following equation:

$$\text{FSPL (dB)} = 20\log_{10}(d) + 20\log_{10}(f) + 32.44$$

where $d$ is the distance from the transmitter in kilometers and $f$ is the frequency in megahertz. Do not worry, though: You will not have to know this equation for the ENCOR 350-401 exam. It is presented here to show two interesting facts:

- Free space path loss is an exponential function; the signal strength falls off quickly near the transmitter but more slowly farther away.
- The loss is a function of distance and frequency only.

You should also be aware that the free space path loss is greater in the 5 GHz band than it is in the 2.4 GHz band. In the equation, as the frequency increases, so does the loss in dB. This means that 2.4 GHz devices have a greater effective range than 5 GHz devices, assuming an equal transmitted signal strength. Figure 17-24 shows the range difference, where both transmitters have an equal EIRP. The dashed circles show where the effective range ends, at the point where the signal strength of each transmitter is equal.

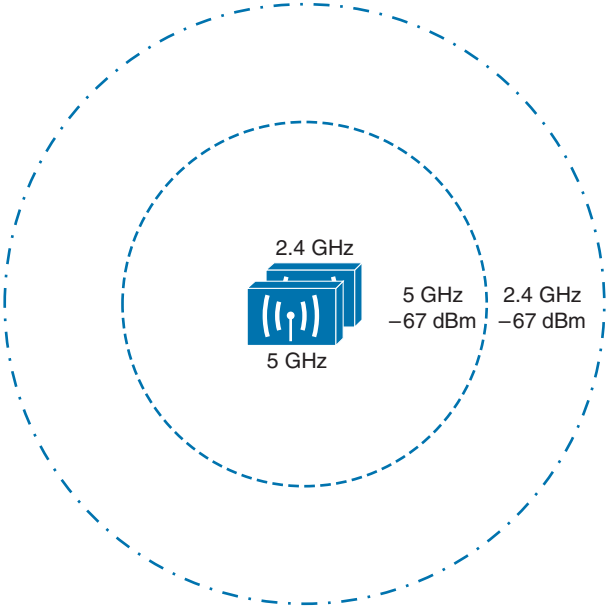



**Figure 17-24** *Effective Range of 2.4 GHz and 5 GHz Transmitters*

Assuming that a transmitter is sending an RF signal with enough power to reach a receiver, what received signal strength value is good enough? Every receiver has a *sensitivity level*, or a threshold that divides intelligible, useful signals from unintelligible ones. As long as a signal is received with a power level that is greater than the sensitivity level, chances are that the data from the signal can be understood correctly. Figure 17-25 shows an example of how the signal strength at a receiver might change over time. The receiver's sensitivity level is –82 dBm.

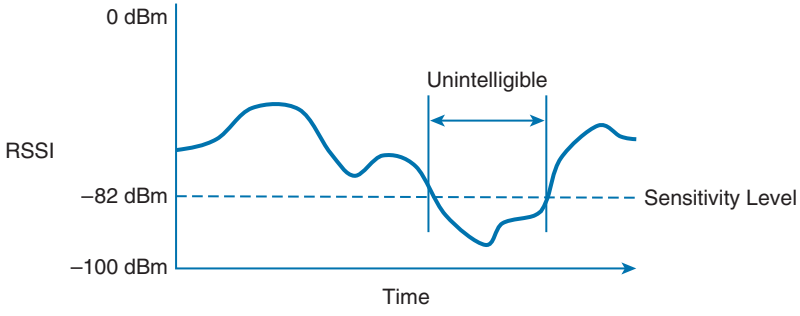



**Figure 17-25** *Example of Receiver Sensitivity Level*

Due to the physical properties of an RF signal, a modulation scheme can alter only the following attributes:

- Frequency, but only by varying slightly above or below the carrier frequency
- Phase
- Amplitude

**Table 17-4** A Summary of Common 802.11 Standard Amendments

| Standard | 2.4 GHz? | 5 GHz? | Data Rates Supported | Channel Widths Supported |
|---|---|---|---|---|
| 802.11b | Yes | No | 1, 2, 5.5, and 11 Mbps | 22 MHz |
| 802.11g | Yes | No | 6, 9, 12, 18, 24, 36, 48, and 54 Mbps | 22 MHz |
| 802.11a | No | Yes | 6, 9, 12, 18, 24, 36, 48, and 54 Mbps | 20 MHz |
| 802.11n | Yes | Yes | Up to 150 Mbps* per spatial stream, up to 4 spatial streams | 20 or 40 MHz |
| 802.11ac | No | Yes | Up to 866 Mbps per spatial stream, up to 4 spatial streams | 20, 40, 80, or 160 MHz |
| 802.11ax | Yes* | Yes* | Up to 1.2 Gbps per spatial stream, up to 8 spatial streams | 20, 40, 80, or 160 MHz |

* 802.11ax is designed to work on any band from 1 to 7 GHz, provided that the band is approved for use.

As a simple example, Figure 17-31 illustrates DRS operation on the 2.4 GHz band. Each concentric circle represents the range supported by a particular modulation and coding scheme. (You can ignore the cryptic names because they are beyond the scope of the ENCOR 350-401 exam.) The figure is somewhat simplistic because it assumes a consistent power level across all modulation types. Notice that the white circles denote OFDM modulation (802.11g), and the shaded circles contain DSSS modulation (802.11b). None of the 802.11n/ac/ax modulation types are shown, for simplicity. The data rates are arranged in order of increasing circle size or range from the transmitter.

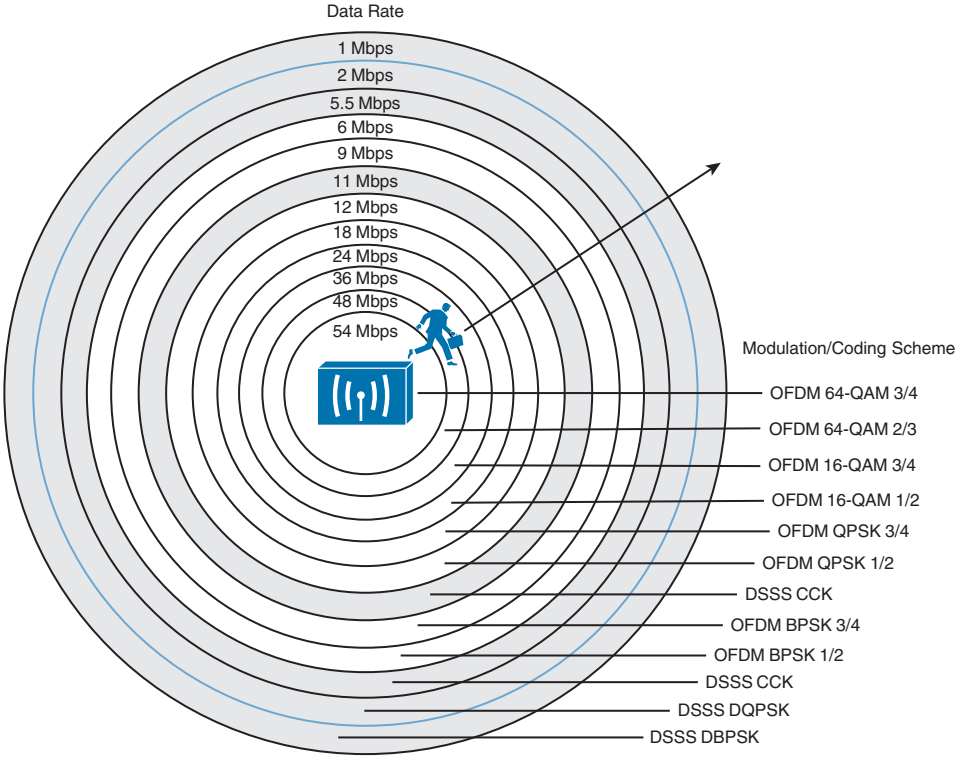



**Figure 17-31** *Dynamic Rate Shifting as a Function of Range*

# Chapter 18

Autonomous APs are self-contained, each offering one or more fully functional, standalone basic service sets (BSSs). They are also a natural extension of a switched network, connecting wireless service set identifiers (SSIDs) to wired virtual LANs (VLANs) at the access layer. Figure 18-1 shows the basic architecture; even though only four APs are shown across the bottom, a typical enterprise network could consist of hundreds or thousands of APs.

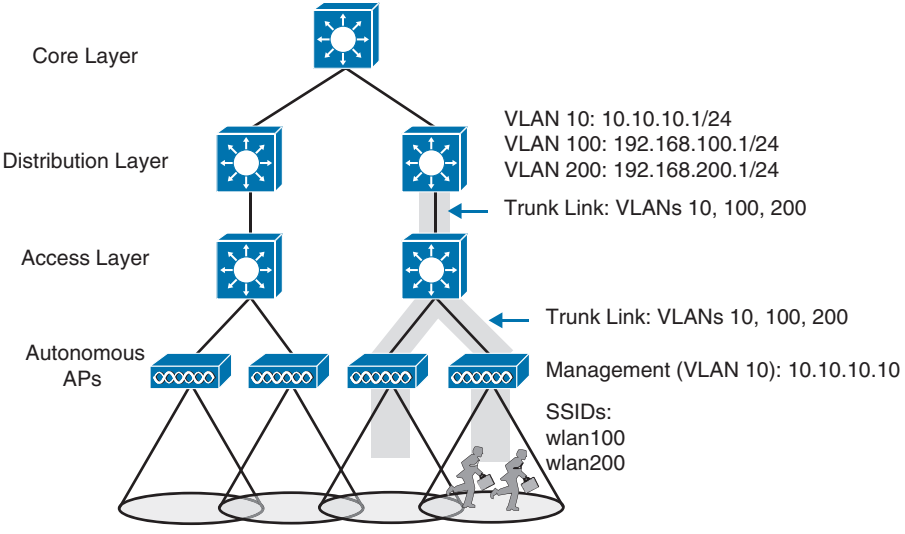



**Figure 18-1** *Wireless Network Topology Using Autonomous APs*

Several topologies can be built from a WLC and a collection of APs. These differ according to where the WLC is located within the network. For example, a WLC can be placed in a central location, usually in a data center or near the network core, so that you can maximize the number of APs joined to it. This is known as a *centralized* or *unified* wireless LAN topology, as shown in Figure 18-3. This tends to follow the concept that most of the resources users need to reach are located in a central location, such as a data center or the Internet. Traffic to and from wireless users travels from the APs over CAPWAP tunnels that reach into the center of the network. A centralized WLC also provides a convenient place to enforce security policies that affect all wireless users.

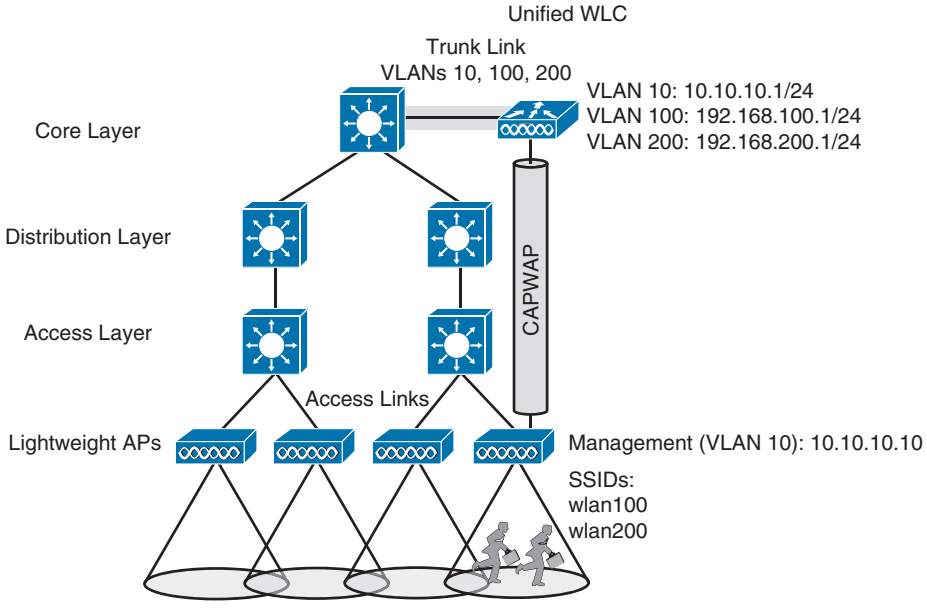



**Figure 18-3** *WLC Location in a Centralized Wireless Network Topology*

Now imagine that a WLC can be located further down in the network hierarchy. In Figure 18-5, the WLC is co-located with an access layer switch. This can be desirable when the switch platform can also support the WLC function. This is known as an *embedded* wireless network topology because the WLC is embedded in the switch hardware.

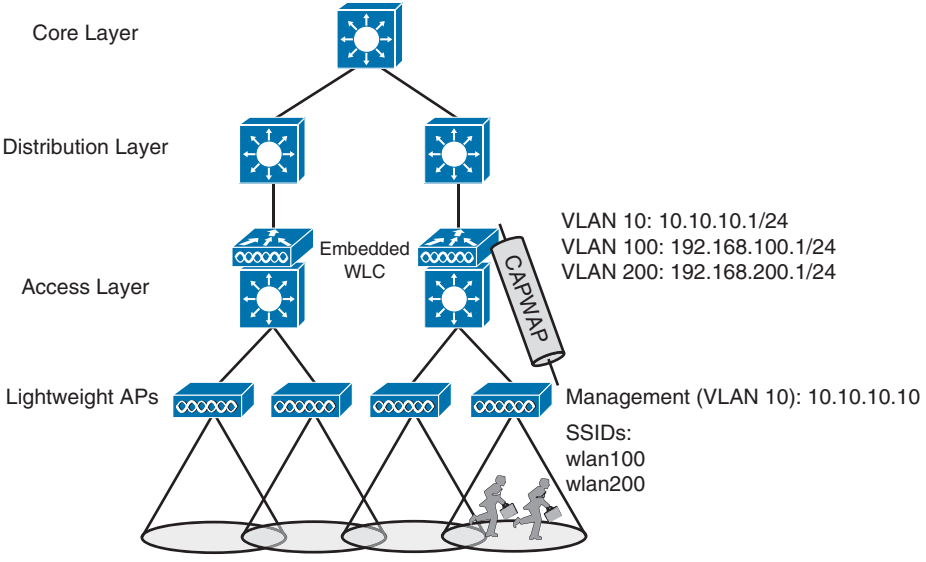



**Figure 18-5** *WLC Location in an Embedded Wireless Network Topology*

As you might have guessed, it is also possible to move the WLC even below the access layer and into an AP. Figure 18-7 illustrates the *Mobility Express* topology, where a fully functional Cisco AP also runs software that acts as a WLC. This can be useful in small scale environments, such as small, midsize, or multi-site branch locations, where you might not want to invest in dedicated WLCs at all. The AP that hosts the WLC forms a CAPWAP tunnel with the WLC, as do any other APs at the same location. A Mobility Express WLC can support up to 100 APs.

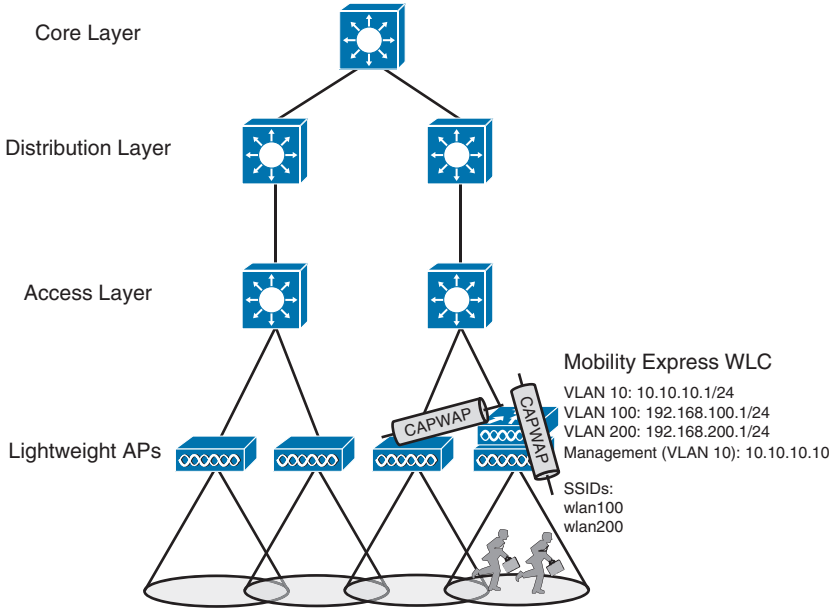



**Figure 18-7** *WLC Location in a Mobility Express Wireless Network Topology*

The sequence of the most common states is as follows:

1. **AP boots:** Once an AP receives power, it boots on a small IOS image so that it can work through the remaining states and communicate over its network connection. The AP must also receive an IP address from either a Dynamic Host Configuration Protocol (DHCP) server or a static configuration so that it can communicate over the network.
2. **WLC discovery:** The AP goes through a series of steps to find one or more controllers that it might join. The steps are explained further in the next section.
3. **CAPWAP tunnel:** The AP attempts to build a CAPWAP tunnel with one or more controllers. The tunnel will provide a secure Datagram Transport Layer Security (DTLS) channel for subsequent AP-WLC control messages. The AP and WLC authenticate each other through an exchange of digital certificates.
4. **WLC join:** The AP selects a WLC from a list of candidates and then sends a CAPWAP Join Request message to it. The WLC replies with a CAPWAP Join Response message. The next section explains how an AP selects a WLC to join.
5. **Download image:** The WLC informs the AP of its software release. If the AP's own software is a different release, the AP downloads a matching image from the controller, reboots to apply the new image, and then returns to step 1. If the two are running identical releases, no download is needed.
6. **Download config:** The AP pulls configuration parameters down from the WLC and can update existing values with those sent from the controller. Settings include RF, service set identifier (SSID), security, and quality of service (QoS) parameters.
7. **Run state:** Once the AP is fully initialized, the WLC places it in the "run" state. The AP and WLC then begin providing a BSS and begin accepting wireless clients.
8. **Reset:** If an AP is reset by the WLC, it tears down existing client associations and any CAPWAP tunnels to WLCs. The AP then reboots and starts through the entire state machine again.

To discover a WLC, an AP sends a unicast CAPWAP Discovery Request to a controller's IP address over UDP port 5246 or a broadcast to the local subnet. If the controller exists and is working, it returns a CAPWAP Discovery Response to the AP. The sequence of discovery steps used is as follows:

**Step 1.** The AP broadcasts a CAPWAP Discovery Request on its local wired subnet. Any WLCs that also exist on the subnet answer with a CAPWAP Discovery Response.

**NOTE** If the AP and controllers lie on different subnets, you can configure the local router to relay any broadcast requests on UDP port 5246 to specific controller addresses. Use the following configuration commands:

```
router(config)# ip forward-protocol udp 5246
router(config)# interface vlan n
router(config-int)# ip helper-address WLC1-MGMT-ADDR
router(config-int)# ip helper-address WLC2-MGMT-ADDR
```

**Step 2.** An AP can be "primed" with up to three controllers—a primary, a secondary, and a tertiary. These are stored in nonvolatile memory so that the AP can remember them after a reboot or power failure. Otherwise, if an AP has previously joined with a controller, it should have stored up to 8 out of a list of 32 WLC addresses that it received from the last controller it joined. The AP attempts to contact as many controllers as possible to build a list of candidates.

**Step 3.** The DHCP server that supplies the AP with an IP address can also send DHCP option 43 to suggest a list of WLC addresses.

**Step 4.** The AP attempts to resolve the name CISCO-CAPWAP-CONTROLLER.*local-domain* with a DNS request (where *localdomain* is the domain name learned from DHCP). If the name resolves to an IP address, the controller attempts to contact a WLC at that address.

**Step 5.** If none of the steps has been successful, the AP resets itself and starts the discovery process all over again.

From the WLC, you can configure a lightweight AP to operate in one of the following special-purpose modes:

- **Local:** The default lightweight mode that offers one or more functioning BSSs on a specific channel. During times when it is not transmitting, the AP scans the other channels to measure the level of noise, measure interference, discover rogue devices, and match against intrusion detection system (IDS) events.
- **Monitor:** The AP does not transmit at all, but its receiver is enabled to act as a dedicated sensor. The AP checks for IDS events, detects rogue access points, and determines the position of stations through location-based services.
- **FlexConnect:** An AP at a remote site can locally switch traffic between an SSID and a VLAN if its CAPWAP tunnel to the WLC is down and if it is configured to do so.
- **Sniffer:** An AP dedicates its radios to receiving 802.11 traffic from other sources, much like a sniffer or packet capture device. The captured traffic is then forwarded to a PC running network analyzer software such as LiveAction Omnipeek or Wireshark, where it can be analyzed further.
- **Rogue detector:** An AP dedicates itself to detecting rogue devices by correlating MAC addresses heard on the wired network with those heard over the air. Rogue devices are those that appear on both networks.
- **Bridge:** An AP becomes a dedicated bridge (point-to-point or point-to-multipoint) between two networks. Two APs in bridge mode can be used to link two locations separated by a distance. Multiple APs in bridge mode can form an indoor or outdoor mesh network.
- **Flex+Bridge:** FlexConnect operation is enabled on a mesh AP.
- **SE-Connect:** The AP dedicates its radios to spectrum analysis on all wireless channels. You can remotely connect a PC running software such as MetaGeek Chanalyzer or Cisco Spectrum Expert to the AP to collect and analyze the spectrum analysis data to discover sources of interference.

To describe the antenna's performance, you might draw a sphere with a diameter that is proportional to the signal strength, as shown in Figure 18-9. Most likely, you would draw the sphere on a logarithmic scale so that very large and very small numbers could be shown on the same linear plot. A plot that shows the relative signal strength around an antenna is known as the *radiation pattern*.

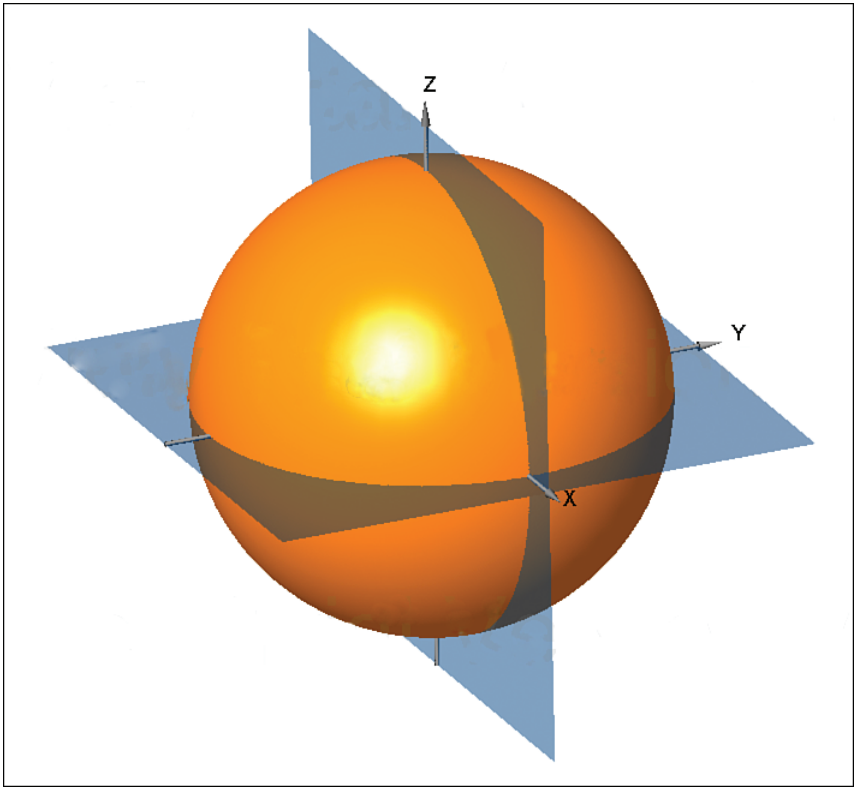



**Figure 18-9** *Plotting the Radiation Pattern of an Isotropic Antenna*

# Chapter 19

Suppose that the client then begins to move into AP 2's cell. Somewhere near the cell boundary, the client decides that the signal from AP 1 has degraded and it should look elsewhere for a stronger signal. The client decides to roam and reassociate with AP 2. Figure 19-2 shows the new scenario after the roam occurs. Notice that both APs have updated their list of associated clients to reflect Client 1's move from AP 1 to AP 2. If AP 1 still has any leftover wireless frames destined for the client after the roam, it forwards them to AP 2 over the wired infrastructure—simply because that is where the client's MAC address now resides.

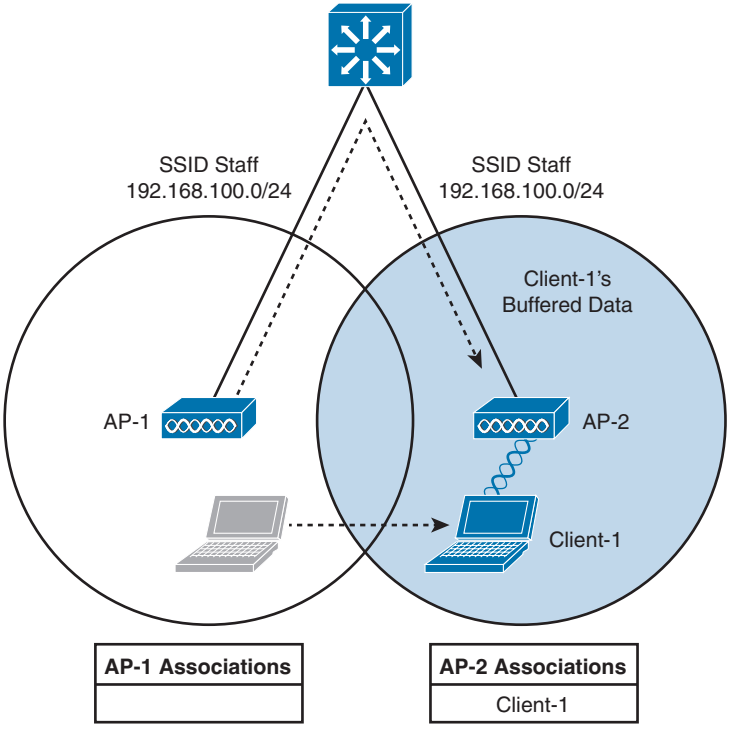



**Figure 19-2** *After Roaming Between Autonomous APs*

When Client 1 starts moving, it eventually roams to AP 2, as shown in Figure 19-5. Not much has changed except that the controller has updated the client association from AP 1 to AP 2. Because both APs are bound to the same controller, the roam occurs entirely within the controller. This is known as *intracontroller roaming*.

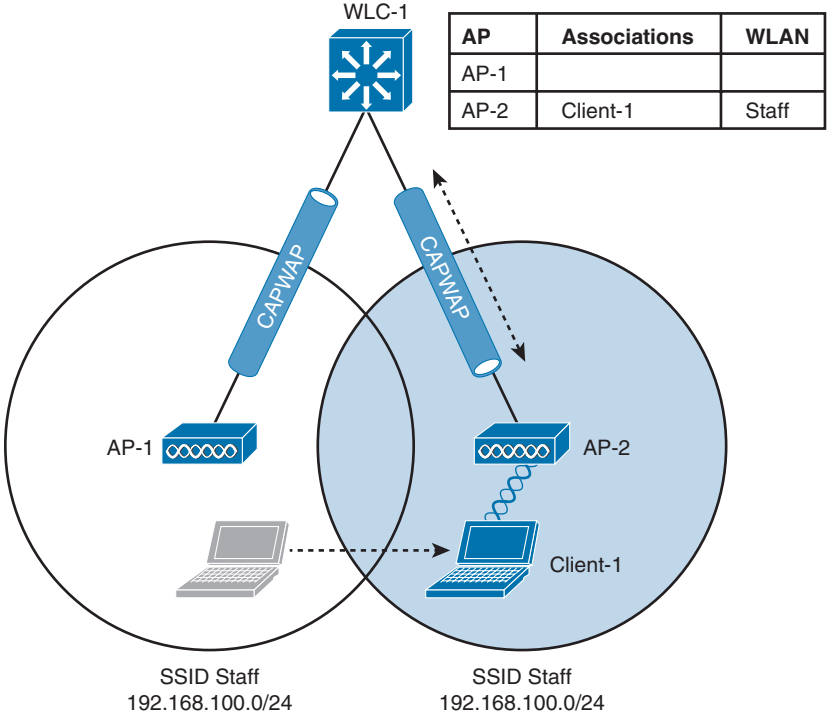



**Figure 19-5** *Cisco Wireless Network After an Intracontroller Roam*

When a client roams from one AP to another and those APs lie on two different controllers, the client makes an intercontroller roam. Figure 19-6 shows a simple scenario prior to a roam. Controller WLC 1 has one association in its database—that of Client 1 on AP 1. Figure 19-7 shows the result of the client roaming to AP 2.

| AP | Associations | WLAN | VLAN |
|---|---|---|---|
| | | | |

| AP | Associations | WLAN | VLAN |
|---|---|---|---|
| AP-2 | Client-1 | Staff | 100 |

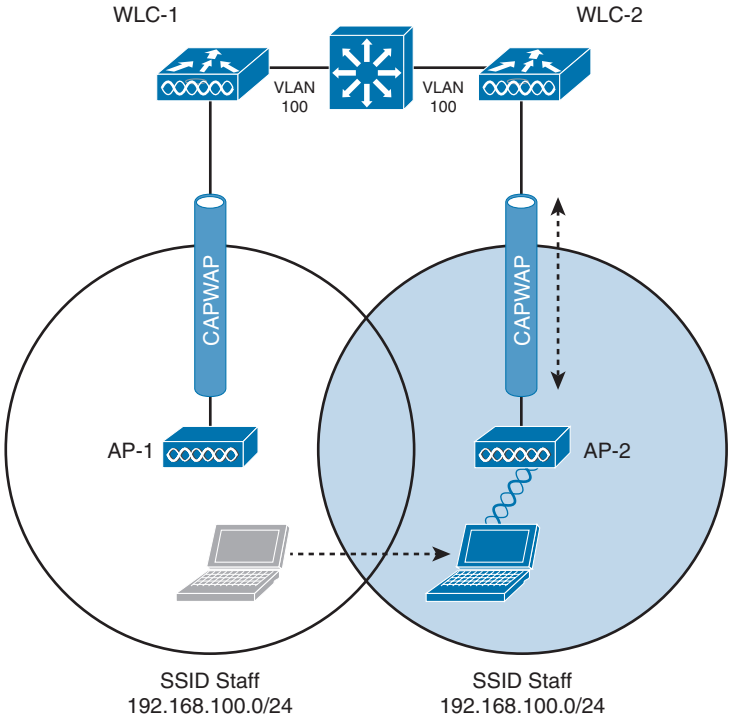



**Figure 19-7** *After an Intercontroller Roam*

A Layer 3 intercontroller roam consists of an extra tunnel that is built between the client's original controller and the controller it has roamed to. The tunnel carries data to and from the client as if it is still associated with the original controller and IP subnet. Figure 19-9 shows the results of a Layer 3 roam. The original controller (WLC 1) is called the *anchor controller*, and the controller with the roamed client is called the *foreign controller*. Think of the client being anchored to the original controller no matter where it roams later. When the client roams away from its anchor, it moves into foreign territory.

| AP | Associations | WLAN | VLAN |
|---|---|---|---|
| WLC-2 | Client-1 (Mobile) | Staff | 100 |

| AP | Associations | WLAN | VLAN |
|---|---|---|---|
| AP-2 | Client-1 | Staff | |

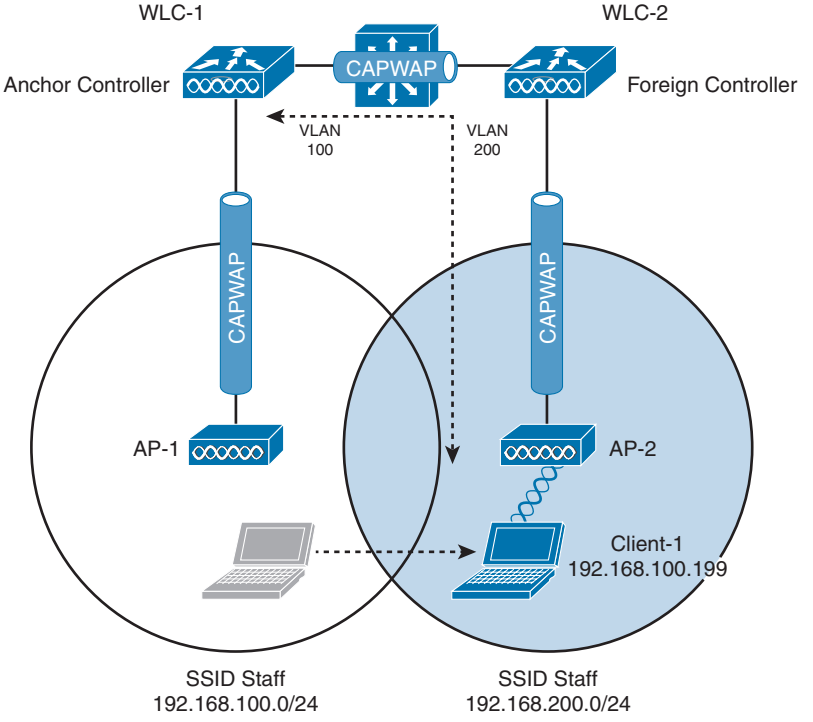



**Figure 19-9** *After a Layer 3 Intercontroller Roam*

Mobility groups have an implied hierarchy, as shown in Figure 19-10. Each controller maintains a mobility list that contains its own MAC address and the MAC addresses of other controllers. Each controller in the list is also assigned a mobility group name. In effect, the mobility list gives a controller its view of the outside world; it knows of and trusts only the other controllers configured in the list. If two controllers are not listed in each other's mobility list, they are unknown to each other and clients will not be able to roam between them. Clients will have to associate and authenticate from scratch.

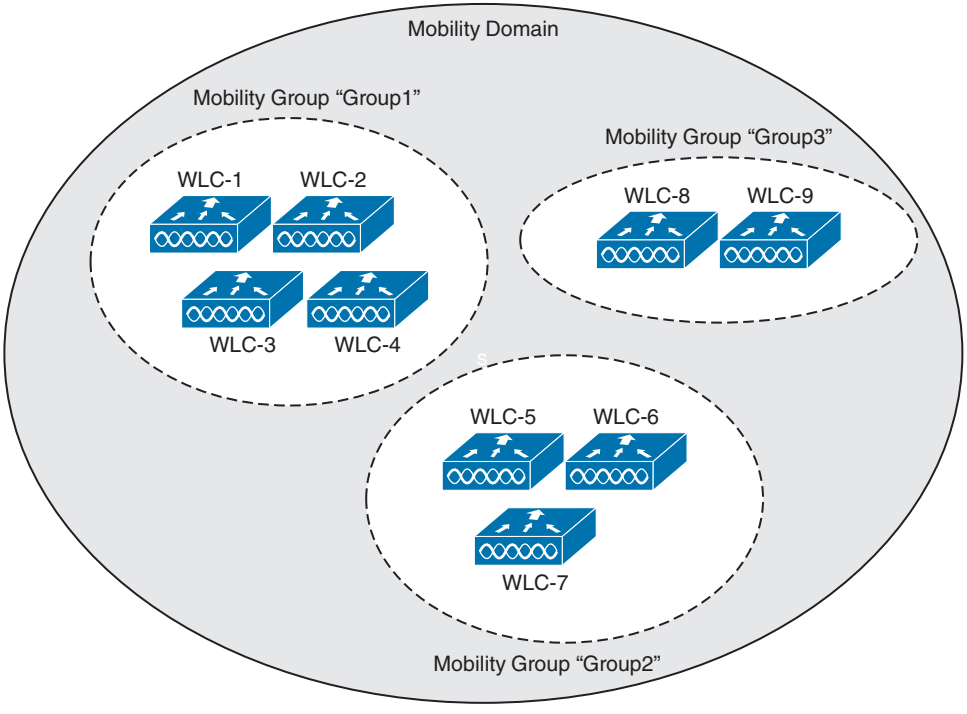



**Figure 19-10** *Mobility Group Hierarchy*

To locate a device more accurately, an AP can use the received signal strength (RSS) of a client device as a measure of the distance between the two. Free space path loss causes an RF signal to be attenuated or diminished exponentially as a function of its frequency and the distance it travels. That means a client's distance from an AP can be computed from its received signal strength. If the distance is measured from a single AP only, it is difficult to determine where the client is situated in relation to the AP. In the case of an indoor AP with an omnidirectional antenna, the client could be located anywhere along a circular path of fixed distance because the received signal strength would be fairly consistent at all points on the circle. A better solution is to obtain the same measurement from three or more APs, then correlate the results and determine where they intersect. Figure 19-11 illustrates the difference in determining a client's location with a single and multiple APs.

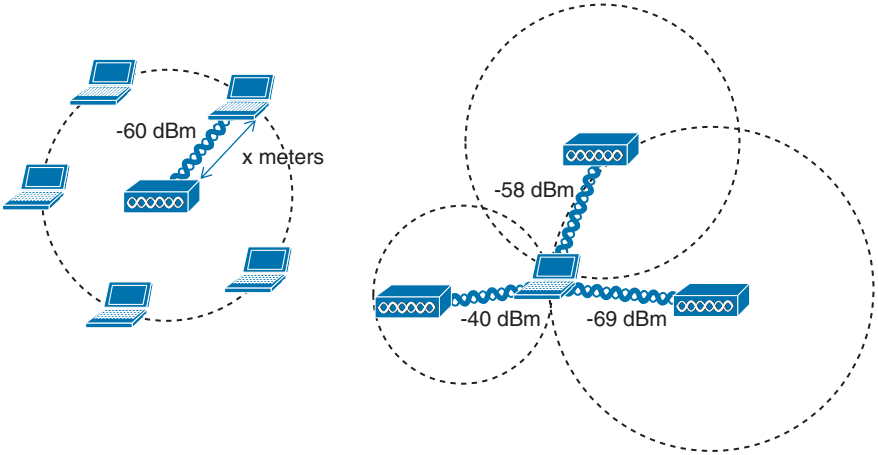



**Figure 19-11** *Locating a Wireless Device with One AP (left) and Three APs (right)*

The most intuitive way to interpret location data is to view devices on a map that represents the building and floor where they are located. Figure 19-12 shows an example map of one floor of a building from Cisco DNA Spaces. The square icons represent AP locations, which were manually entered on the map. Device locations are indicated by small colored dots that are dynamically placed on the map at regular time intervals. Green dots represent wireless devices that have successfully associated with APs, and red dots represent devices that are not associated but are actively sending probe requests to find nearby APs.

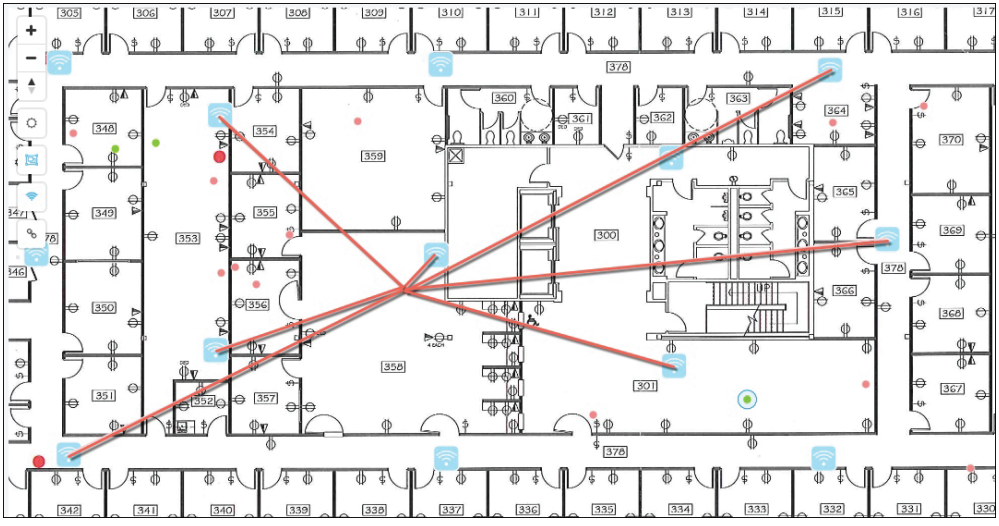



**Figure 19-12** *An Example Map Showing Real Time Location Data for Tracked Devices*

# Chapter 20

All three WPA versions support two client authentication modes, Pre-Shared Key (PSK) or 802.1x, depending on the scale of the deployment. These are also known as *personal mode* and *enterprise mode*, respectively. With personal mode, a key string must be shared or configured on every client and AP before the clients can connect to the wireless network. The pre-shared key is normally kept confidential so that unauthorized users have no knowledge of it. The key string is never sent over the air. Instead, clients and APs work through a four-way handshake procedure that uses the pre-shared key string to construct and exchange encryption key material that can be openly exchanged. When that process is successful, the AP can authenticate the client, and the two can secure data frames that are sent over the air.

With Open Authentication and PSK authentication, wireless clients are authenticated locally at the AP without further intervention. The scenario changes with 802.1x; the client uses Open Authentication to associate with the AP, and then the actual client authentication process occurs at a dedicated authentication server. Figure 20-8 shows the three-party 802.1x arrangement, which consists of the following entities:

- **Supplicant:** The client device that is requesting access
- **Authenticator:** The network device that provides access to the network (usually a wireless LAN controller [WLC])
- **Authentication server (AS):** The device that takes user or client credentials and permits or denies network access based on a user database and policies (usually a RADIUS server)

To use EAP-based authentication and 802.1x, you should leverage the enterprise modes of WPA, WPA2, and WPA3. (As always, you should use the highest WPA version that is supported on your WLCs, APs, and wireless clients.) The enterprise mode supports many EAP methods, such as LEAP, EAP-FAST, PEAP, EAP-TLS, EAP-TTLS, and EAP-SIM, but you do not have to configure any specific method on a WLC. Instead, specific EAP methods must be configured on the authentication server and supported on the wireless client devices. Remember that the WLC acts as the EAP middleman between the clients and the AS.

Web Authentication can be handled locally on the WLC for smaller environments through Local Web Authentication (LWA). You can configure LWA in the following modes:

- LWA with an internal database on the WLC
- LWA with an external database on a RADIUS or LDAP server
- LWA with an external redirect after authentication
- LWA with an external splash page redirect, using an internal database on the WLC
- LWA with passthrough, requiring user acknowledgement

# Chapter 21

As you prepare to troubleshoot a single wireless client, think about all the things a client needs to join and use the network. Figure 21-1 illustrates the following conditions that must be met for a successful association:

- The client is within RF range of an AP and asks to associate.
- The client authenticates.
- The client requests and receives an IP address.

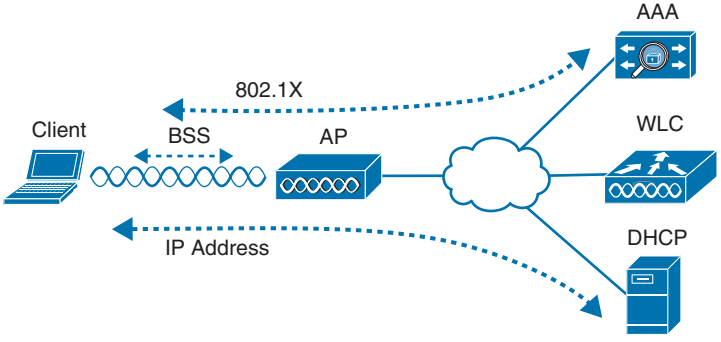



**Figure 21-1** *Conditions for a Successful Wireless Association*

Perhaps the most important information about the client is shown as the sequence of large dots under the Connectivity heading (refer to Figure 21-4). Before a controller will permit a client to fully associate with a basic service set (BSS), the client must progress through a sequence of states. Each state refers to a policy that the client must meet before moving on to the next state. The dots represent the client's status at each of the following crucial steps as it attempts to join the wireless network:

- **Start:** Client activity has just begun.
- **Association:** The client has requested 802.11 authentication and association with an AP.
- **Authentication:** The client must pass a Layer 2 Pre-Shared Key (PSK) or 802.1x authentication policy.
- **DHCP:** The WLC is waiting to learn the client's IP address from a Dynamic Host Configuration Protocol (DHCP) server.
- **Online:** The client has passed Layer 2 and Layer 3 policies, successfully associated, and can pass traffic.

Suppose that the same client moves to a different location and then complains of poor performance. By searching for the client's MAC address on the WLC, you see the new information shown in Figure 21-5. This time, AP is receiving the client's signal strength at –76 dBm and the SNR at 18 dB—both rather low values, causing the current data rate to fall to 29 Mbps. A quick look at the Connection Score value reveals a low 20%. It is safe to assume that the client has moved too far away from the AP where it is associated, causing the signal strength to become too low to support faster performance. This might indicate that you need to place a new AP in that area to boost the RF coverage. Or it could indicate a client device that is not roaming soon enough to a new AP with a stronger signal.

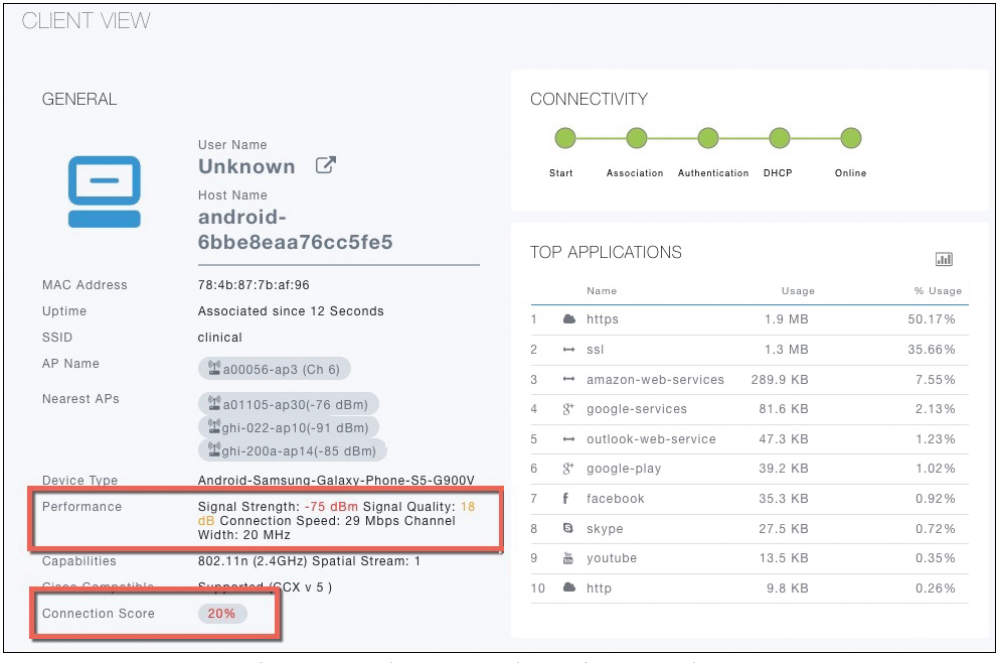



**Figure 21-5** *WLC Information About a Poorly Performing Client*

By scrolling to the bottom of the client search information, you can see the Client Test section, which offers links to four client testing tools:

- **Ping Test:** The WLC sends five ICMP echo packets to the client's IP address and measures the response time, as shown in Figure 21-9.

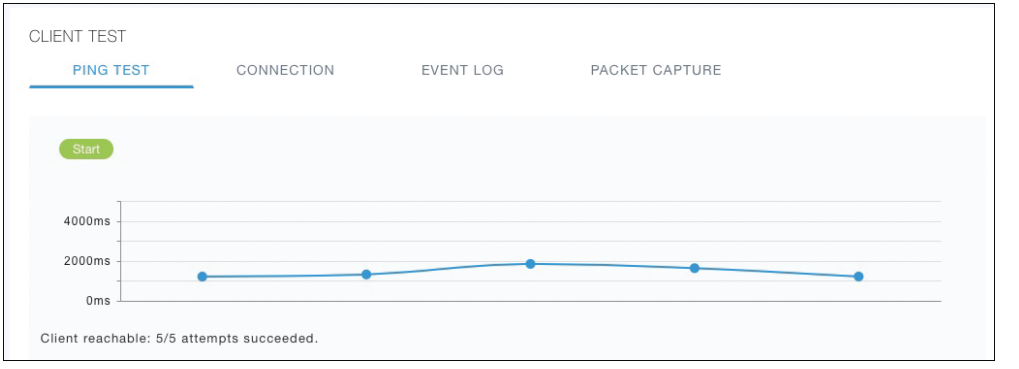



**Figure 21-9** *Testing Ping Response Times Between the WLC and a Client*

- **Connection:** The WLC debugs the client for up to three minutes and checks each policy step as the client attempts to join the wireless network. Figure 21-10 shows a client that has successfully joined, and Figure 21-11 shows a client that failed Layer 2 authentication with a pre-shared key because its key did not match the key configured on the WLC.

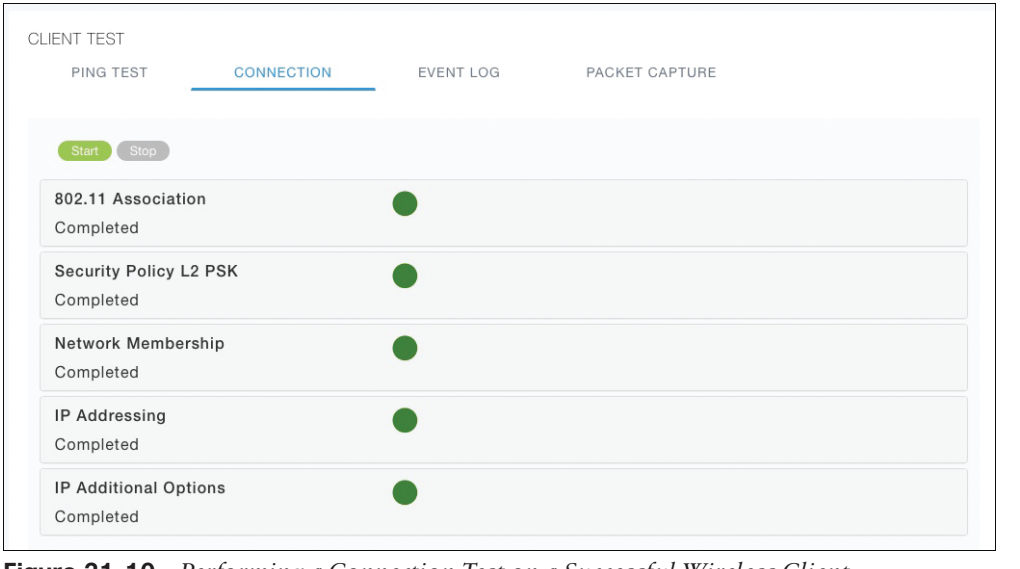



**Figure 21-10** *Performing a Connection Test on a Successful Wireless Client*

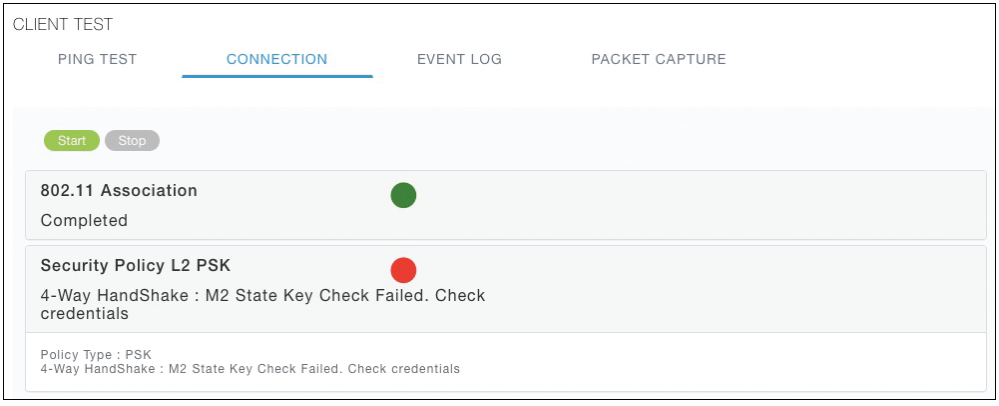



**Figure 21-11** *Performing a Connection Test on a Failed Wireless Client*

- **Event Log:** The WLC collects and displays a log of events as the client attempts to join the wireless network, as shown in Figure 21-12. This information is very complex and detailed and is usually more suited for Cisco TAC engineers.

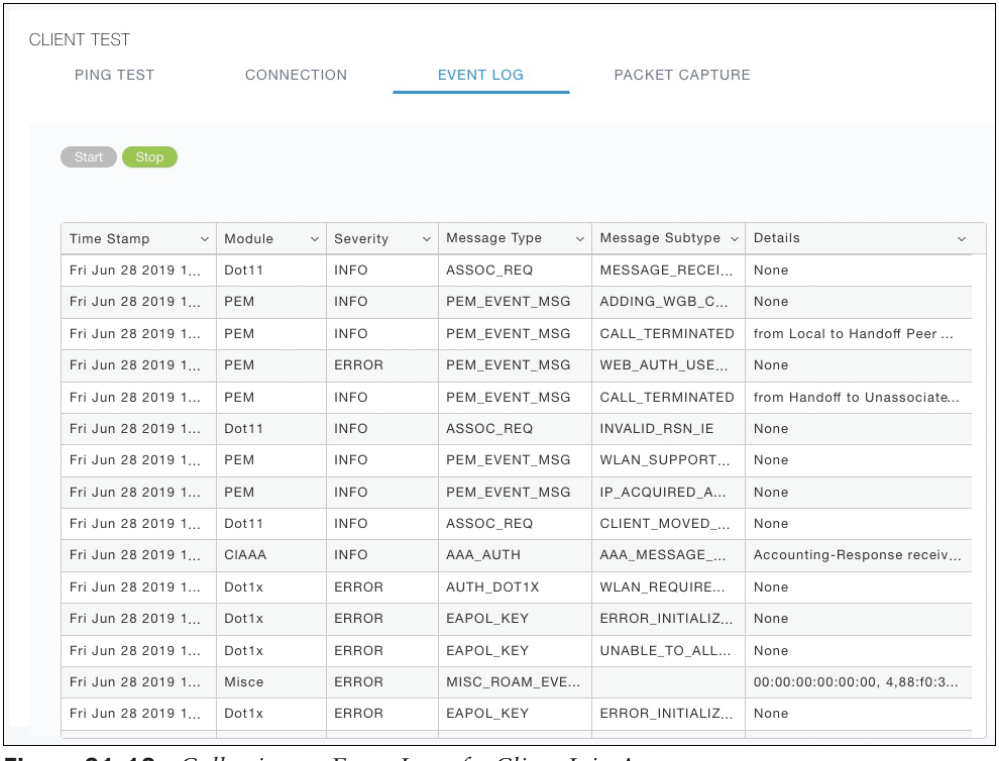



**Figure 21-12** *Collecting an Event Log of a Client Join Attempt*

- **Packet Capture:** The WLC enables a wireless packet capture at the AP where the client attempts to join, as shown in Figure 21-13. The captured data is saved to a specified FTP server, where it can be downloaded and analyzed using a packet analysis tool like Wireshark or LiveAction Omnipeek.

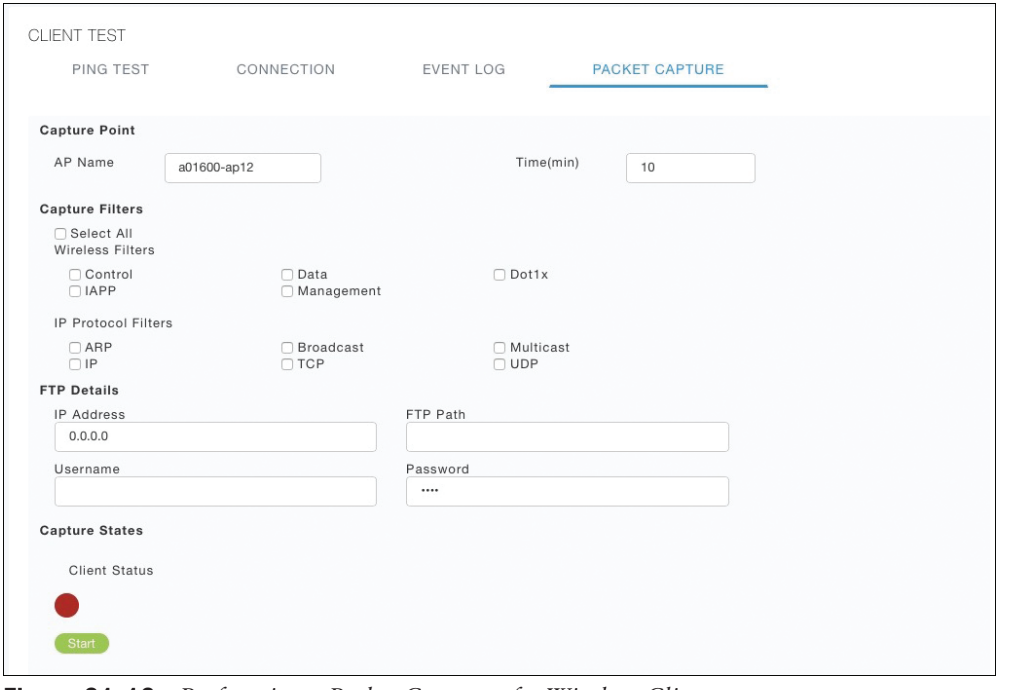



**Figure 21-13** *Performing a Packet Capture of a Wireless Client*

The easiest approach is to simply look for the AP in the list of live APs that have joined the controller. If you know which controller the AP should join, open a management session to it. Enter the AP's name in the search bar. If the search reveals a live AP that is joined to the controller, information is displayed in the Access Point View screen, as shown in Figure 21-14.

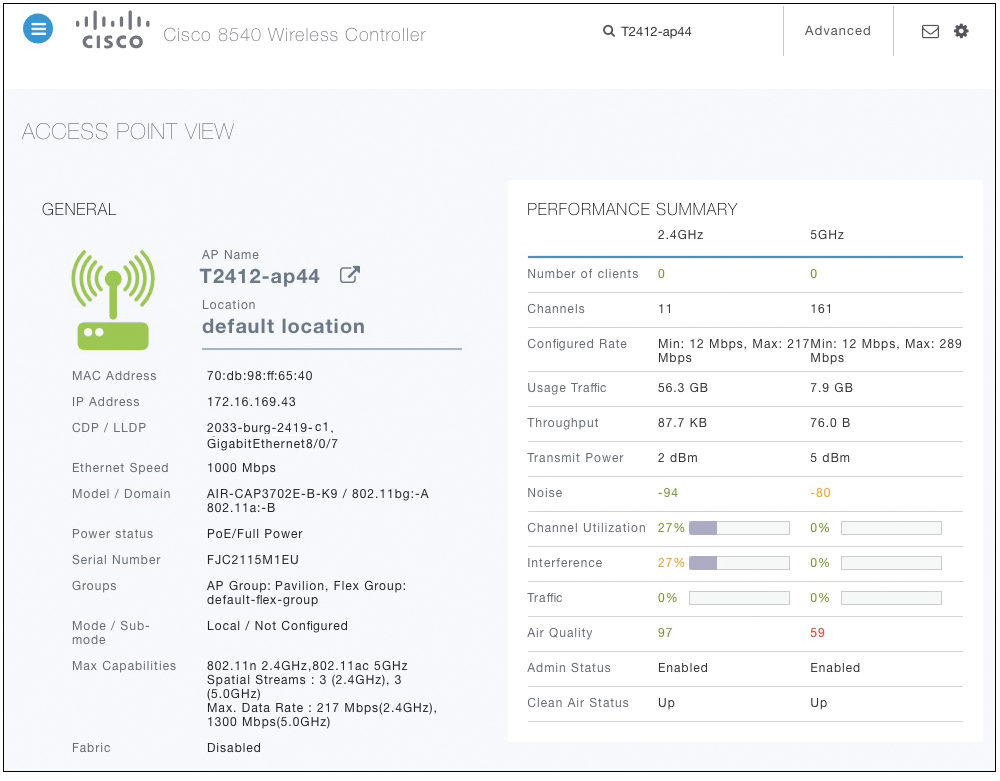



**Figure 21-14** *Displaying Information About an AP*

The channel information also shows an index of air quality. This is a measure of how competing and interfering devices affect the airtime quality or performance on a channel, presented as a number from 0 (worst) to 100 (best). For the best performance, a channel should have a high air quality value. A Cisco AP contains a built-in spectrum analyzer that can monitor wireless channels to detect and identify sources of interference.

# Chapter 22

## Hierarchical LAN Design Model

A hierarchical LAN design model divides the enterprise network architecture into modular layers. By breaking up the design into modular layers, you can have each layer to implement specific functions. These modular layers can be easily replicated throughout the network, which simplifies the network design and provides an easy way to scale the network as well as a consistent deployment method.

A hierarchical LAN design avoids the need for a flat and fully meshed network in which all nodes are interconnected. In fully meshed network architectures, network changes tend to affect a large number of systems. Hierarchical design provides fault containment by constraining the network changes to a subset of the network, which affects fewer systems and makes it easy to manage as well as improve resiliency. In a modular layer design, network components can be placed or taken out of service with little or no impact to the rest of the network; this facilitates troubleshooting, problem isolation, and network management.

The hierarchical LAN design divides networks or their modular blocks into the following three layers:

- **Access layer:** Gives endpoints and users direct access to the network.
- **Distribution layer:** Provides an aggregation point for the access layer and acts as a services and control boundary between the access layer and the core layer.
- **Core layer (also referred to as the backbone):** Provides connections between distribution layers for large environments.

## Access Layer

The *access layer*, also commonly referred as the *network edge*, is where end-user devices or endpoints connect to the network. It provides high-bandwidth device connectivity using wired and wireless access technologies such as Gigabit Ethernet and 802.11n and 802.11ac wireless. While endpoints in most cases will not use the full capacity of these connections for extended periods of time, the ability to burst up to these high bandwidths when required helps improve the quality of experience (QoE) and productivity of the end user.

### Distribution Layer

The primary function of the distribution layer is to aggregate access layer switches in a given building or campus. The distribution layer provides a boundary between the Layer 2 domain of the access layer and the core's Layer 3 domain. This boundary provides two key functions for the LAN: On the Layer 2 side, the distribution layer creates a boundary for Spanning Tree Protocol (STP), limiting propagation of Layer 2 faults, and on the Layer 3 side, the distribution layer provides a logical point to summarize IP routing information when it enters the core of the network. The summarization reduces IP routing tables for easier troubleshooting and reduces protocol overhead for faster recovery from failures.

## Core Layer

As networks grow beyond three distribution layers in a single location, organizations should consider using a core layer to optimize the design. The core layer is the backbone and aggregation point for multiple networks and provides scalability, high availability, and fast convergence to the network.

The core can provide high-speed connectivity for large enterprises with multiple campus networks distributed worldwide, and it can also provide interconnectivity between the end-user/endpoint campus access layer and other network blocks, such as the data center, the private cloud, the public cloud, the WAN, the Internet edge, and network services, as discussed later in this chapter.

The core layer reduces the network complexity, from $N \times (N - 1)$ to $N$ links for $N$ distributions.

### Two-Tier Design (Collapsed Core)

Smaller campus networks may have multiple departments spread across multiple floors within a building. In these environments, a core layer may not be needed, and collapsing the core function into the distribution layer can be a cost-effective solution (as no core layer means no core layer devices) that requires no sacrifice of most of the benefits of the three-tier hierarchical model. Prior to selecting a two-tier collapsed core and distribution layers, future scale, expansion, and manageability factors need to be considered.

## Three-Tier Design

Three-tier designs separate the core and distribution layers and are recommended when more than two pairs of distribution switches are required. Multiple pairs of distribution switches are typically required for the following reasons:

- When implementing a network for a large enterprise campus composed of multiple buildings, where each building requires a dedicated distribution layer
- When the density of WAN routers, Internet edge devices, data center servers, and network services are growing to the point where they can affect network performance and throughput
- When geographic dispersion of the LAN access switches across many buildings in a larger campus facility would require more fiber-optic interconnects back to a single collapsed core

When multiple distribution layers need to be interconnected, it becomes necessary to use a core layer.

### Layer 2 Access Layer (STP Based)

Traditional LAN designs use a Layer 2 access layer and a Layer 3 distribution layer. The distribution layer is the Layer 3 IP gateway for access layer hosts. Whenever possible, it is recommended to restrict a VLAN to a single access layer switch to eliminate topology loops, which are common points of failure in LANs, even when STP is enabled in the network. Restricting a VLAN to a single switch provides a loop-free design, but at the cost of network flexibility because all hosts within a VLAN are restricted to a single access switch. Some organizations require that the same Layer 2 VLAN be extended to multiple access layer switches to accommodate an application or a service. The looped design causes STP to block links, which reduces the bandwidth from the rest of the network and can cause slower network convergence.

### Layer 3 Access Layer (Routed Access)

Routed access is an alternative configuration in which Layer 3 is extended all the way to the access layer switches. In this design, access layer switches act as full Layer 3 routed nodes (providing both Layer 2 and Layer 3 switching), and the access-to-distribution Layer 2 uplink trunks are replaced with Layer 3 point-to-point routed links. Consequently, the Layer 2/Layer 3 demarcation point is moved from the distribution switch to the access switch.

### Simplified Campus Design

The simplified campus design relies on switch clustering such as a virtual switching system (VSS) and stacking technologies such as StackWise, in which multiple physical switches act as a single logical switch. Clustering and stacking technologies can be applied to any of the campus building blocks to simplify them even further. Using this design offers the following advantages:

- **Simplified design:** By using the single logical distribution layer design, there are fewer boxes to manage, which reduces the amount of time spent on ongoing provisioning and maintenance.
- **No first-hop redundancy protocol required:** It eliminates the need for first-hop redundancy protocols such as HSRP and VRRP because the default IP gateway is on a single logical interface.
- **Reduced STP dependence:** Because EtherChannel is used, it eliminates the need for STP for a Layer 2 access design; however, STP is still required as a failsafe in case multiple access switches are interconnected.
- **Increased uplink utilization:** With EtherChannel, all uplinks from access to distribution can be used, increasing the effective bandwidth available to the end users and endpoints connected to the access layer switches.
- **Easier troubleshooting:** The topology of the network from the distribution layer to the access layer is logically a hub-and-spoke topology, which reduces the complexity of the design and troubleshooting.
- **Faster convergence:** With EtherChannel, all links are in forwarding state, and this significantly optimizes the convergence time following a node or link failure event because EtherChannel provides fast sub-second failover between links in an uplink bundle.
- **Distributed VLANs:** With this design, VLANs can span multiple access switches without the need to block any links.

# Chapter 23

With SD-Access, an evolved campus network can be built that addresses the needs of existing campus networks by leveraging the following capabilities, features, and functionalities:

- **Network automation:** SD-Access replaces manual network device configurations with network device management through a single point of automation, orchestration, and management of network functions through the use of Cisco DNA Center. This simplifies network design and provisioning and allows for very fast, lower-risk deployment of network devices and services using best-practice configurations.
- **Network assurance and analytics:** SD-Access enables proactive prediction of network-related and security-related risks by using telemetry to improve the performance of the network, endpoints, and applications, including encrypted traffic.
- **Host mobility:** SD-Access provides host mobility for both wired and wireless clients.
- **Identity services:** *Cisco Identity Services Engine (ISE)* identifies users and devices connecting to the network and provides the contextual information required for users and devices to implement security policies for network access control and network segmentation.
- **Policy enforcement:** Traditional access control lists (ACLs) can be difficult to deploy, maintain, and scale because they rely on IP addresses and subnets. Creating access and application policies based on group-based policies using Security Group Access Control Lists (SGACLs) provides a much simpler and more scalable form of policy enforcement based on identity instead of an IP address.
- **Secure segmentation:** With SD-Access it is easier to segment the network to support guest, corporate, facilities, and IoT-enabled infrastructure.
- **Network virtualization:** SD-Access makes it possible to leverage a single physical infrastructure to support multiple virtual routing and forwarding (VRF) instances, referred to as *virtual networks (VNs)*, each with a distinct set of access policies.

Cisco SD-Access is based on existing hardware and software technologies. What makes Cisco SD-Access special is how these technologies are integrated and managed together. The Cisco SD-Access fabric architecture can be divided into four basic layers, as illustrated in Figure 23-2. The following sections focus on the relationships between these four layers.

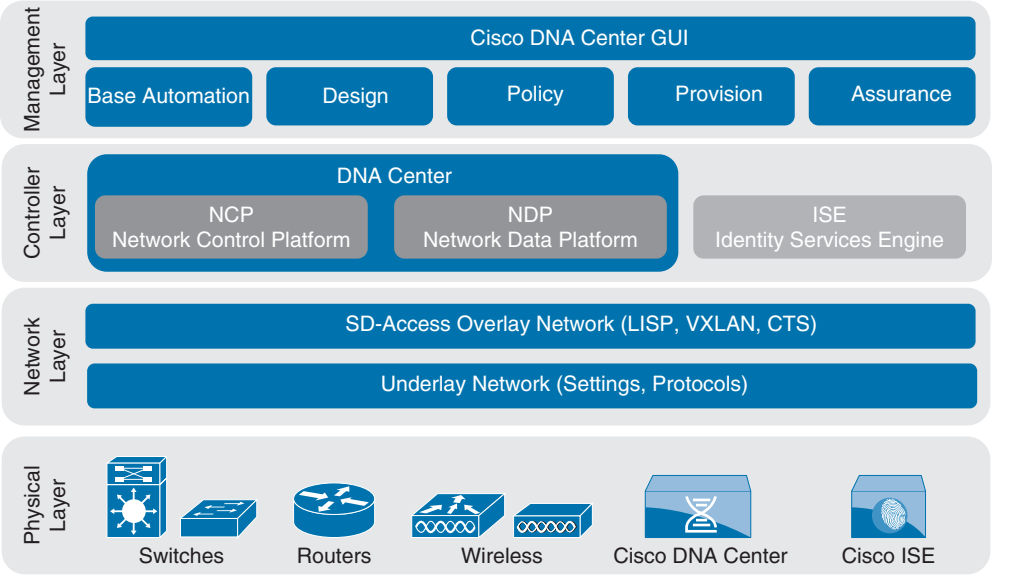



**Figure 23-2** *Cisco SD-Access Architecture*

### Underlay Network

The underlay network for SD-Access should be configured to ensure performance, scalability, and high availability because any problems with the underlay can affect the operation of the fabric overlay. While it is possible to use a Layer 2 network underlay design running Spanning Tree Protocol (STP), it is not recommended. The recommended design for the network underlay is to use a Layer 3 routed access campus design using IS-IS as the IGP. IS-IS offers operational advantages such as neighbor establishment without IP dependencies, peering capability using loopback addresses, and agnostic treatment of IPv4, IPv6, and non-IP traffic.

Two models of underlay are supported:

- **Manual underlay:** This type of underlay network is configured and managed manually (such as with a CLI or an API) rather than through Cisco DNA Center. An advantage of the manual underlay is that it allows customization of the network to fit any special design requirements (such as changing the IGP to OSPF); in addition, it allows SD-Access to run on the top of a legacy (or third-party) IP-based network.
- **Automated underlay:** In a fully automated network underlay, all aspects of the underlay network are configured and managed by the Cisco DNA Center LAN Automation feature. The LAN Automation feature creates an IS-IS routed access campus design and uses the Cisco Network Plug and Play features to deploy both unicast and multicast routing configuration in the underlay to improve traffic delivery efficiency for SD-Access. An automated underlay eliminates misconfigurations and reduces the complexity of the network underlay. It also greatly simplifies and speeds the building of the network underlay. A downside to an automated underlay is that it does not allow manual customization for special design requirements.

### Overlay Network (SD-Access Fabric)

The SD-Access fabric is the overlay network, and it provides policy-based network segmentation, host mobility for wired and wireless hosts, and enhanced security beyond the normal switching and routing capabilities of a traditional network.

In SD-Access, the fabric overlay is fully automated, regardless of the underlay network model used (manual or automated). It includes all necessary overlay control plane protocols and addressing, as well as all global configurations associated with operation of the SD-Access fabric.

There are three basic planes of operation in the SD-Access fabric:

- Control plane, based on Locator/ID Separation Protocol (LISP)
- Data plane, based on Virtual Extensible LAN (VXLAN)
- Policy plane, based on Cisco TrustSec

### SD-Access Control Plane

The SD-Access fabric control plane is based on *Locator/ID Separation Protocol (LISP)*. LISP is an IETF standard protocol defined in RFC 6830 that is based on a simple endpoint ID (EID) to routing locator (RLOC) mapping system to separate the identity (endpoint IP address) from its current location (network edge/border router IP address).

LISP dramatically simplifies traditional routing environments by eliminating the need for each router to process every possible IP destination address and route. It does this by moving remote destination information to a centralized mapping database called the LISP map server (MS) (a control plane node in SD-Access), which allows each router to manage only its local routes and query the map system to locate destination EIDs.

This technology provides many advantages for Cisco SD-Access, such as smaller routing tables, dynamic host mobility for wired and wireless endpoints, address-agnostic mapping (IPv4, IPv6, and/ or MAC), and built-in network segmentation through VRF instances.

In Cisco SD-Access, several enhancements to the original LISP specifications have been added, including distributed Anycast Gateway, VN Extranet, and Fabric Wireless, and more features are planned for the future.

### SD-Access Fabric Data Plane

The tunneling technology used for the fabric data plane is based on Virtual Extensible LAN (VXLAN). VXLAN encapsulation is IP/UDP based, meaning that it can be forwarded by any IP-based network (legacy or third party) and creates the overlay network for the SD-Access fabric. Although LISP is the control plane for the SD-Access fabric, it does not use LISP data encapsulation for the data plane; instead, it uses VXLAN encapsulation because it is capable of encapsulating the original Ethernet header to perform MAC-in-IP encapsulation, while LISP does not. Using VXLAN allows the SD-Access fabric to support Layer 2 and Layer 3 virtual topologies (overlays) and the ability to operate over any IP-based network with built-in network segmentation (VRF instance/VN) and built-in group-based policy. The differences between the LISP and VXLAN packet formats are illustrated in Figure 23-4.

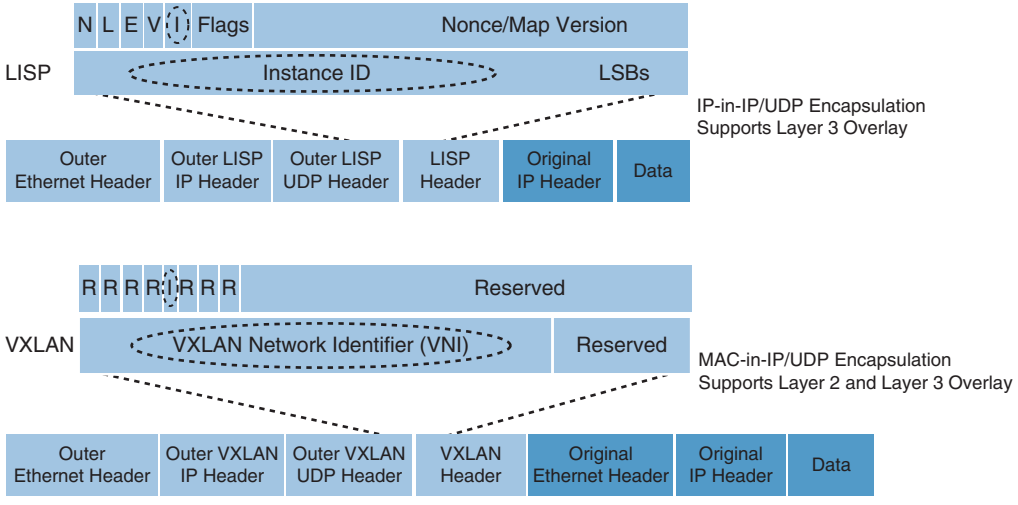



**Figure 23-4** *LISP and VXLAN Packet Format Comparison*

The original VXLAN specification was enhanced for SD-Access to support Cisco TrustSec Scalable Group Tags (SGTs). This was accomplished by adding new fields to the first 4 bytes of the VXLAN header in order to transport up to 64,000 SGT tags. The new VXLAN format is called VXLAN Group Policy Option (VXLAN-GPO), and it is defined in the IETF draft draft-smith-vxlan-group-policy-05.

### SD-Access Fabric Policy Plane

The fabric policy plane is based on Cisco TrustSec. Cisco TrustSec SGT tags are assigned to authenticated groups of users or end devices. Network policy (for example, ACLs, QoS) is then applied throughout the SD-Access fabric, based on the SGT tag instead of a network address (MAC, IPv4, or IPv6). This allows for the creation of network policies such as security, quality of service (QoS), policy-based routing (PBR), and network segmentation, based only on the SGT tag and not the network address (MAC, IPv4, or IPv6) of the user or endpoint.

There are five basic device roles in the fabric overlay:

- **Control plane node:** This node contains the settings, protocols, and mapping tables to provide the endpoint-to-location (EID-to-RLOC) mapping system for the fabric overlay.
- **Fabric border node:** This fabric device (for example, core layer device) connects external Layer 3 networks to the SDA fabric.
- **Fabric edge node:** This fabric device (for example, access or distribution layer device) connects wired endpoints to the SDA fabric.
- **Fabric WLAN controller (WLC):** This fabric device connects APs and wireless endpoints to the SDA fabric.
- **Intermediate nodes:** These are intermediate routers or extended switches that do not provide any sort of SD-Access fabric role other than underlay services.

### Fabric Edge Nodes

A fabric edge node provides onboarding and mobility services for wired users and devices (including fabric-enabled WLCs and APs) connected to the fabric. It is a LISP tunnel router (xTR) that also provides the anycast gateway, endpoint authentication, and assignment to overlay host pools (static or DHCP), as well as group-based policy enforcement (for traffic to fabric endpoints).

A fabric edge first identifies and authenticates wired endpoints (through 802.1x), in order to place them in a host pool (SVI and VRF instance) and scalable group (SGT assignment). It then registers the specific EID host address (that is, MAC, /32 IPv4, or /128 IPv6) with the control plane node.

A fabric edge provides a single Layer 3 anycast gateway (that is, the same SVI with the same IP address on all fabric edge nodes) for its connected endpoints and also performs the encapsulation and de-encapsulation of host traffic to and from its connected endpoints.

### Fabric Control Plane Node

A fabric control plane node is a LISP map server/resolver (MS/MR) with enhanced functions for SD-Access, such as fabric wireless and SGT mapping. It maintains a simple host tracking database to map EIDs to RLOCs.

The control plane (host database) maps all EID locations to the current fabric edge or border node, and it is capable of multiple EID lookup types (IPv4, IPv6, or MAC).

The control plane receives registrations from fabric edge or border nodes for known EID prefixes from wired endpoints and from fabric mode WLCs for wireless clients. It also resolves lookup requests from fabric edge or border nodes to locate destination EIDs and updates fabric edge nodes and border nodes with wired and wireless client mobility and RLOC information.

### Fabric Border Nodes

Fabric border nodes are LISP proxy tunnel routers (PxTRs) that connect external Layer 3 networks to the SD-Access fabric and translate reachability and policy information, such as VRF and SGT information, from one domain to another.

There are three types of border nodes:

- **Internal border (rest of company):** Connects only to the known areas of the organization (for example, WLC, firewall, data center).
- **Default border (outside):** Connects only to unknown areas outside the organization. This border node is configured with a default route to reach external unknown networks such as the Internet or the public cloud that are not known to the control plane nodes.
- **Internal + default border (anywhere):** Connects transit areas as well as known areas of the company. This is basically a border that combines internal and default border functionality into a single node.

### Fabric Wireless Controller (WLC)

A fabric-enabled WLC connects APs and wireless endpoints to the SD-Access fabric. The WLC is external to the fabric and connects to the SD-Access fabric through an internal border node. A fabric WLC node provides onboarding and mobility services for wireless users and endpoints connected to the SD-Access fabric. A fabric WLC also performs PxTR registrations to the fabric control plane (on behalf of the fabric edges) and can be thought of as a fabric edge for wireless clients. The control plane node maps the host EID to the current fabric access point and fabric edge node location the access point is attached to.

Better understanding the benefits and operation of Cisco SD-Access requires reviewing the following concepts related to how the multiple technologies that are used by the SD-WAN solution operate and interact in SD-Access:

- **Virtual network (VN):** The VN provides virtualization at the device level, using VRF instances to create multiple Layer 3 routing tables. VRF instances provide segmentation across IP addresses, allowing for overlapped address space and traffic segmentation. In the control plane, LISP instance IDs are used to maintain separate VRF instances. In the data plane, edge nodes add a VXLAN VNID to the fabric encapsulation.
- **Host pool:** A host pool is a group of endpoints assigned to an IP pool subnet in the SDA-Access fabric. Fabric edge nodes have a Switched Virtual Interface (SVI) for each host pool to be used by endpoints and users as their default gateway. The SD-Access fabric uses EID mappings to advertise each host pool (per instance ID), which allows host-specific (/32, /128, or MAC) advertisement and mobility. Host pools can be assigned dynamically (using host authentication, such as 802.1x) and/or statically (per port).
- **Scalable group:** A scalable group is a group of endpoints with similar policies. The SD-Access policy plane assigns every endpoint (host) to a scalable group using TrustSec SGT tags. Assignment to a scalable group can be either static per fabric edge port or using dynamic authentication through AAA or RADIUS using Cisco ISE. The same scalable group is configured on all fabric edge and border nodes. Scalable groups can be defined in Cisco DNA Center and/or Cisco ISE and are advertised through Cisco TrustSec. There is a direct one-to-one relationship between host pools and scalable groups. Therefore, the scalable groups operate within a VN by default. The fabric edge and border nodes include the SGT tag ID in each VXLAN header, which is carried across the fabric data plane. This keeps each scalable group separate and allows SGACL policy and enforcement.
- **Anycast gateway:** The anycast gateway provides a pervasive Layer 3 default gateway where the same SVI is provisioned on every edge node with the same SVI IP and MAC address. This allows an IP subnet to be stretched across the SD-Access fabric. For example, if the subnet 10.1.0.0/24 is provisioned on an SD-Access fabric, this subnet will be deployed across all of the edge nodes in the fabric, and an endpoint located in that subnet can be moved to any edge node within the fabric without a change to its IP address or default gateway. This essentially stretches these subnets across all of the edge nodes throughout the fabric, thereby simplifying the IP address assignment and allowing fewer but larger IP subnets to be deployed. In essence, the fabric behaves like a logical switch that spans multiple buildings, where an endpoint can be unplugged from one port and plugged into another port on a different building, and it will seem as if the endpoint is connecting to the same logical switch, where it can still reach the same SVI and other endpoints in the same VLAN.

## Controller Layer

The controller layer provides all of the management subsystems for the management layer, and this is all provided by Cisco DNA Center and Cisco ISE. Figure 23-8 illustrates the different components that comprise the controller layer and how they interact with each other as well as with the campus fabric.

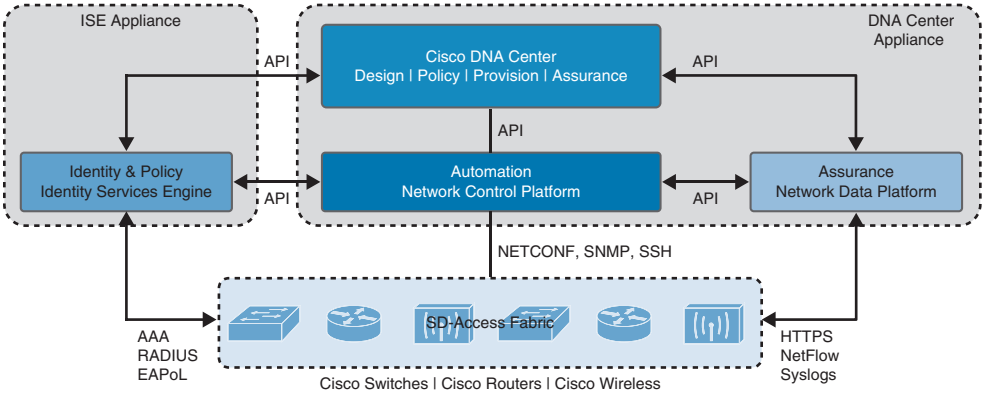



**Figure 23-8** *SD-Access Main Components*

There are three main controller subsystems:

- **Cisco Network Control Platform (NCP):** This is a subsystem integrated directly into Cisco DNA Center that provides all the underlay and fabric automation and orchestration services for the physical and network layers. NCP configures and manages Cisco network devices using NETCONF/YANG, Simple Network Management Protocol (SNMP), SSH/Telnet, and so on and then provides network automation status and other information to the management layer.
- **Cisco Network Data Platform (NDP):** NDP is a data collection and analytics and assurance subsystem that is integrated directly into Cisco DNA Center. NDP analyzes and correlates various network events through multiple sources (such as NetFlow and Switched Port Analyzer [SPAN]) and identifies historical trends. It uses this information to provide contextual information to NCP and ISE, and it provides network operational status and other information to the management layer.
- **Cisco Identity Services Engine (ISE):** The basic role of ISE is to provide all the identity and policy services for the physical layer and network layer. ISE provides network access control (NAC) and identity services for dynamic endpoint-to-group mapping and policy definition in a variety of ways, including using 802.1x, MAC Authentication Bypass (MAB), and Web Authentication (WebAuth). ISE also collects and uses the contextual information shared from NDP and NCP (and other systems, such as Active Directory and AWS). ISE then places the profiled endpoints into the correct scalable group and host pool. It uses this information to provide information to NCP and NDP, so the user (management layer) can create and manage group-based policies. ISE is also responsible for programming group-based policies on the network devices.

### Management Layer

The Cisco DNA Center management layer is the user interface/user experience (UI/UX) layer, where all the information from the other layers is presented to the user in the form of a centralized management dashboard. It is the intent-based networking aspect of Cisco DNA.

A full understanding of the network layer (LISP, VXLAN, and Cisco TrustSec) or controller layer (Cisco NCP, NDP, and ISE) is not required to deploy the fabric in SD-Access. Nor is there a requirement to know how to configure each individual network device and feature to create the consistent end-to-end behavior offered by SD-Access.

The management layer abstracts all the complexities and dependencies of the other layers and provides the user with a simple set of GUI tools and workflows to easily manage and operate the entire Cisco DNA network (hence the name Cisco DNA Center).

Cisco DNA Center applications are designed for simplicity and are based on the primary workflows defined by Cisco DNA Center: design, policy, provision, and assurance.

The Cisco SD-WAN solution has four main components and an optional analytics service:

- **vManage Network Management System (NMS):** This is a single pane of glass (GUI) for managing the SD-WAN solution.
- **vSmart controller:** This is the brains of the solution.
- **SD-WAN routers:** SD-WAN involves both vEdge and cEdge routers.
- **vBond orchestrator:** This authenticates and orchestrates connectivity between SD-WAN routers and vSmart controllers.
- **vAnalytics:** This is an optional analytics and assurance service.

### vManage NMS

The vManage NMS is a single pane of glass network management system (NMS) GUI that is used to configure and manage the full SD-WAN solution. It enables centralized provisioning and simplifies network changes.

## vSmart Controller

vSmart controllers (which are the brains of the SD-WAN solution) have pre-installed credentials that allow them to authenticate every SD-WAN router that comes online. These credentials ensure that only authenticated devices are allowed access to the SD-WAN fabric. After successful authentication, each vSmart controller establishes a permanent DTLS tunnel to each SD-WAN router in the SD-WAN fabric and uses these tunnels to establish Overlay Management Protocol (OMP) neighborships with each SD-WAN router. OMP is a proprietary routing protocol similar to BGP that can advertise routes, next hops, keys, and policy information needed to establish and maintain the SD-WAN fabric.

The vSmart controller processes the OMP routes learned from the SD-WAN routers (or other vSmart controllers) to determine the network topology and calculate the best routes to network destinations. Then it advertises reachability information learned from these routes to all the SD-WAN routers in the SD-WAN fabric.

vSmart controllers also implement all the control plane policies created on vManage, such as service chaining, traffic engineering, and segmentation per VPN topology. For example, when a policy is created on vManage for an application (such as YouTube) that requires no more than 1% loss and 150 ms latency, that policy is downloaded to the vSmart controller. vSmart converts the policy into a format that all the SD-WAN routers in the fabric can understand, and it automatically implements the policy on all SD-WAN routers without the need to rely on a CLI. The vSmart controller also works in conjunction with the vBond orchestrator to authenticate the devices as they join the network and to orchestrate connectivity between the SD-WAN routers.

### Cisco SD-WAN Routers (vEdge and cEdge)

Cisco SD-WAN routers deliver the essential WAN, security, and multicloud capabilities of the Cisco SD-WAN solution, and they are available as hardware, software, cloud, or virtualized routers that sit at the perimeter of a site, such as a remote office, branch office, campus, or data center.

SD-WAN routers support standard router features, such as OSPF, BGP, ACLs, QoS, and routing policies, in addition to the SD-WAN overlay control and data plane functions. Each SD-WAN router automatically establishes a secure Datagram Transport Layer Security (DTLS) connection with the vSmart controller and forms an OMP neighborship over the tunnel to exchange routing information. It also establishes standard IPsec sessions with other SD-WAN routers in the fabric. SD-WAN routers have local intelligence to make site-local decisions regarding routing, high availability (HA), interfaces, ARP management, and ACLs. The vSmart controller provides remote site routes and the reachability information necessary to build the SD-WAN fabric.

A main differentiator between SD-WAN cEdge routers and vEdge routers is that they support advanced security features, as demonstrated in Table 23-2.

**Table 23-2** SD-WAN Router Advanced Security Feature Comparison

| Feature | cEdge | vEdge |
|---|---|---|
| Cisco AMP and AMP Threat Grid | Yes | No |
| Enterprise Firewall | Yes | Yes |
| Cisco Umbrella DNS Security | Yes | Yes |
| URL filtering | Yes | No |
| The Snort intrusion prevention system (IPS) | Yes | No |
| Embedded platform security (including the Cisco Trust Anchor module) | Yes | No |

### vBond Orchestrator

The vBond orchestrator authenticates the vSmart controllers and the SD-WAN routers and orchestrates connectivity between them. It is the only device that must have a public IP address so that all SD-WAN devices in the network can connect to it. A vBond orchestrator is an SD-WAN router that *only* performs vBond orchestrator functions.

## Cisco SD-WAN Cloud OnRamp

Traditional enterprise WAN architectures are not designed for the cloud. As organizations adopt more SaaS applications such as Office 365 and public cloud infrastructures such as AWS and Microsoft Azure, the current network infrastructure poses major problems related to the level of complexity and end-user experience.

The Cisco SD-WAN solution includes a set of functionalities addressing optimal cloud SaaS application access and IaaS connectivity, called Cloud OnRamp. Cloud OnRamp delivers the best application quality of experience (QoE) for SaaS applications by continuously monitoring SaaS performance across diverse paths and selecting the best-performing path based on performance metrics (jitter, loss, and delay). In addition, it simplifies hybrid cloud and multicloud IaaS connectivity by extending the SD-WAN fabric to the public cloud while at the same time increasing high availability and scale.

# Chapter 24

## ping

**ping** is one of the most useful and underrated troubleshooting tools in any network. When following a troubleshooting flow or logic, it is critical to cover the basics first. For example, if a BGP peering adjacency is not coming up, it would make sense to check basic reachability between the two peers prior to doing any deep-dive BGP troubleshooting or debugging. Issues often lies in a lower level of the OSI model; physical layer issues, such as a cable being unplugged, can be found with a quick ping.

### traceroute

**traceroute** is another common troubleshooting tool. **traceroute** is often used to troubleshoot when trying to determine where traffic is failing as well as what path traffic takes throughout the network. **traceroute** shows the IP addresses or DNS names of the hops between the source and destination. It also shows how long it takes to reach the destination at each hop, measured in milliseconds. This tool is frequently used when more than one path is available to the destination or when there is more than one hop to the destination.

## Debugging

Debugging can be a very powerful part of troubleshooting complex issues in a network. Debugging is also informational. This section provides some basic OSPF debugging examples and illustrates how to use debugging when trying to narrow down issues in a network.

One of the most common use cases for debugging is when there is a need to see things at a deeper level (such as when routing protocols are having adjacency issues). There is a normal flow that is taken from a troubleshooting perspective, depending on the routing protocol. However, there are times when these steps have been taken, and the issue is not evident. With OSPF, for example, when troubleshooting adjacency issues, it is very helpful to have debugging experience. Using the simple topology shown in Figure 24-2, in this section, debugging is used to fix a couple issues in the OSPF area 0.

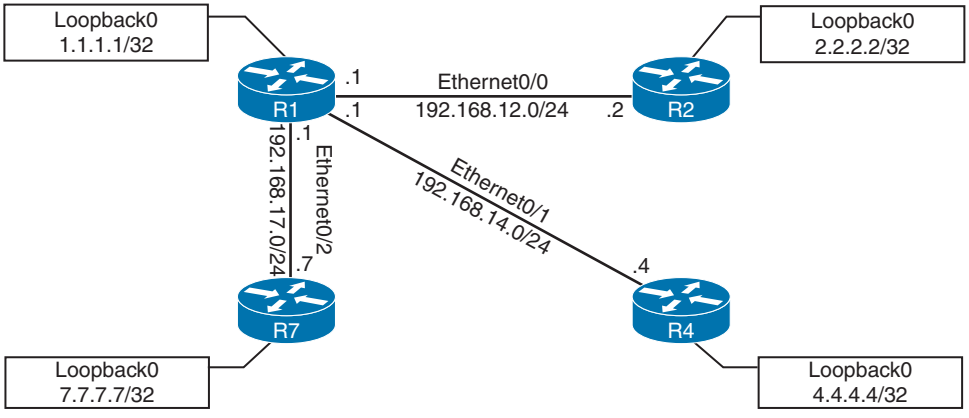



**Figure 24-2** *Debugging Topology*

Different network types have different hello intervals and dead intervals. Table 24-4 highlights the different hello and dead interval times based on the different OSPF network types.

**Table 24-4** OSPF Network Types and Hello/Dead Intervals

| Network Type | Hello Interval (in seconds) | Dead Interval (in seconds) |
|---|---|---|
| Broadcast | 10 | 40 |
| Non-broadcast | 30 | 120 |
| Point-to-point | 10 | 40 |
| Point-to-Multipoint | 30 | 120 |

### Simple Network Management Protocol (SNMP)

Network operations teams often have to rely on reactive alerting from network devices to be notified when something is happening—such as something failing or certain events happening on a device. The typical tool for this is Simple Network Management Protocol (SNMP). SNMP can also be used to configure devices, although this use is less common. More often when network engineering teams need to configure devices, configuration management tools such as Cisco Prime Infrastructure are used.

## NetFlow and Flexible NetFlow

Gathering statistics about a network during its operations is not only useful but important. Gathering statistical information on traffic flows is necessary for a number of reasons. Some businesses, such as service providers, use it for customer billing. Other businesses use it to determine whether traffic is optimally flowing through the network. Some use it for troubleshooting if the network is not performing correctly. NetFlow is very versatile and provides a wealth of information without much configuration burden. That being said, NetFlow has two components that must be configured: *NetFlow Data Capture* and *NetFlow Data Export*. NetFlow Data Capture captures the traffic statistics. NetFlow Data Export exports the statistical data to a NetFlow collector, such as Cisco DNA Center or Cisco Prime Infrastructure. Examples of each of these are provided in this section.

## Specifying the Source Ports

The source ports are defined with the global configuration command **monitor session** *session-id* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**rx** | **tx** | **both**]. The SPAN *session-id* allows for the switch to correlate the source ports to specific destination ports. One or more interfaces or VLANs can be entered by using either a comma (for delimiting multiple interfaces) or a hyphen (for setting a range). Another option is to repeat the command with a different value and let the system update the source range accordingly.

The direction of the traffic can be specified as part of the configuration. With the optional **rx** keyword you capture only traffic received on that source, with the optional **tx** keyword you capture traffic sent by that source, and with the **both** keyword you capture all traffic. By default, traffic is captured for both.

You can specify a trunk port as a source port to capture traffic for all VLANs that traverse that port. This might provide too much data and add noise to the traffic analysis tool. The VLANs can be filtered on the capture with the command **monitor session** *session-id* **filter vlan** *vlan-range*.

### Encapsulated Remote SPAN (ERSPAN)

In large environments, it might not be possible to move a network analyzer to other parts of the network. ERSPAN provides the ability to monitor traffic in one area of the network and route the SPAN traffic to a traffic analyzer in another area of the network through Layer 3 routing. Think of a large-scale WAN with multiple remote sites and being able to do packet captures from anywhere that has IP connectivity. That is a powerful use case for ERSPAN. The configuration commands are similar in nature to those for SPAN and RSPAN. However, because the traffic is routed to another portion of the network, some additional configuration settings must take place to enable this capability.

## IP SLA

*IP SLA* is a tool built into Cisco IOS software that allows for the continuous monitoring of various aspects of the network. The different types of probes that can be configured to monitor traffic within a network environment include the following:

- Delay (both round-trip and one-way)
- Jitter (directional)
- Packet loss (directional)
- Packet sequencing (packet ordering)
- Path (per hop)
- Connectivity (directional)
- Server or website download time
- Voice quality scores

## Cisco DNA Center Assurance

Networks have grown very complex. The influx of mobile devices strains network resources and the network operations staff. Security has become one the most important pieces of the network, and users expect a better experience. Customers demand a simple way to manage Day 0–2 operations and require a scalable and simple approach to running the network. Cisco DNA Center Assurance provides a tool for handling the most relevant customer requirements. Traditionally, multiple management tools were required to meet the needs of the business in terms of managing, operating, and troubleshooting the network. This all changes with Cisco DNA Center Assurance. From a high level, Cisco DNA Center Assurance offers some of the following capabilities (as well as many more):

- Cisco SD-Access fabric configuration
- Software image management (SWIM)
- Simplified provisioning for devices
- Wireless network management
- Simplified security policies
- Configuration templates
- Third-party integration
- Network assurance
- Plug and Play

# Chapter 25

Evolving cybersecurity threats such as phishing, malware, ransomware, and web-based exploits are very common. There is no single product in the industry that can successfully secure organizations from all these threats. To address this, Cisco created *Cisco SAFE*, a security architectural framework that helps design secure solutions for the following places in the network (PINs):

- **Branch:** Branches are typically less secure than the campus and data center PINs because the potentially large number of branches makes it cost-prohibitive to try to apply on them all the security controls found in campus and data center PINs. Branch locations are therefore prime targets for security breaches. It is important to ensure that vital security capabilities are included in the design while keeping it cost-effective. Top threats on branch PINs include endpoint malware (point-of-sale [POS] malware), wireless infrastructure exploits such as rogue APs and man-in-the-middle (MitM) attacks, unauthorized/malicious client activity, and exploitation of trust.
- **Campus:** Campuses contain large numbers of users, including employees, contractors, guests, and partners. Campuses are easy targets for phishing, web-based exploits, unauthorized network access, malware propagation, and botnet infestations.
- **Data center:** Data centers contain an organization's most critical information assets and intellectual capital, and they are therefore the primary goal of all targeted threats. Data centers typically contain hundreds or thousands of servers, which makes it very difficult to create and manage proper security rules to control network access. Typical threats seen in data centers are data extraction, malware propagation, unauthorized network access (application compromise), botnet infestation (scrumping), data loss, privilege escalation, and reconnaissance.
- **Edge:** The edge is the primary ingress and egress point for traffic to and from the Internet, and for this reason, it is the highest-risk PIN and the most important for e-commerce. Typical threats seen on the edge include web server vulnerabilities, distributed denial-of-service (DDoS) attacks, data loss, and MitM attacks.
- **Cloud:** Security in the cloud is dictated by service-level agreements (SLAs) with the cloud service provider and requires independent certification audits and risk assessments. The primary threats are web server vulnerabilities, loss of access, data loss, malware, and MitM attacks.
- **Wide area network (WAN):** The WAN connects the PINs together. In a large organization with hundreds of branches, managing security on the WAN is very challenging. Typical threats seen in WANs are malware propagation, unauthorized network access, WAN sniffing, and MitM attacks.

Implementing the Cisco SAFE framework in an organization provides advanced threat defense protection that spans the full attack continuum before, during, and after an attack for all the PINs:

- **Before:** In this phase, full knowledge of all the assets that need to be protected is required, and the types of threats that could target these assets need to be identified. This phase involves establishing policies and implementing prevention to reduce risk. Cisco solutions for this phase include next-generation firewalls, network access control, network security analysis, and identity services.
- **During:** This phase defines the abilities and actions that are required when an attack gets through. Threat analysis and incident response are some of the typical activities associated with this phase. For this phase, organizations can leverage next-generation intrusion prevention systems, next-generation firewalls, malware protection, and email and web security solutions that make it possible to detect, block, and defend against attacks that have penetrated the network and are in progress.
- **After:** This phase defines the ability to detect, contain, and remediate an attack. After a successful attack, any lessons learned need to be incorporated into the existing security solution. Organizations can leverage Cisco Advanced Malware Protection, next-generation firewalls, and malicious network behavior analysis using Stealthwatch to quickly and effectively scope, contain, and remediate an attack to minimize damage.

## Cisco Talos

*Talos* is the Cisco threat intelligence organization, an elite team of security experts who are supported by sophisticated security systems to create threat intelligence that detects, analyzes, and protects against both known and emerging threats for Cisco products.

Cisco Talos was created from the combination of three security research teams:

- IronPort Security Applications (SecApps)
- The Sourcefire Vulnerability Research Team (VRT)
- The Cisco Threat Research, Analysis, and Communications (TRAC) team

## Cisco Threat Grid

*Cisco Threat Grid* (acquired by Cisco in 2014) is a solution that can perform static file analysis (for example, checking filenames, MD5 checksums, file types, and so on) as well as dynamic file analysis (also known as behavioral analysis) by running the files in a controlled and monitored sandbox environment to observe and analyze the behavior against millions of samples and billions of malware artifacts to determine whether it is malware or not. Behavioral analysis is combined with threat intelligence feeds from Talos as well as with existing security technologies to protect against known and unknown attacks. If Threat Grid identifies a file as malware, it begins to understand what it is doing or attempting to do, the scope of the threat it poses, and how to defend against it. Malware typically includes code to detect whether it is being analyzed in a virtual sandbox environment, and if the malware detects that it is being executed in a sandbox, it won't run, rendering the analysis useless. However, Threat Grid evades being detected by malware by not having the typical instrumentation.

It is also possible to upload suspicious files into a sandbox environment called Glovebox to safely interact with them and observe malware behavior directly.

Threat Grid is available as an appliance and in the cloud, and it is also integrated into existing Cisco security products and third-party solutions.

## Cisco Advanced Malware Protection (AMP)

*Cisco Advanced Malware Protection (AMP)* (formerly FireAMP) is a malware analysis and protection solution that goes beyond point-in-time detection. Using targeted, context-aware malware, attackers have the resources, persistence, time, and expertise to compromise any network relying solely on point-in-time detection mechanisms. Point-in-time detection is completely blind to the scope and depth of a breach after it happens.

The architecture of AMP can be broken down into the following components:

- AMP Cloud (private or public)
- AMP connectors
  - AMP for Endpoints (Microsoft Windows, macOS X, Google Android, Apple iOS, and Linux)
  - AMP for Networks (NGFW, NGIPS, ISRs)
  - AMP for Email (ESA)
  - AMP for Web (WSA)
  - AMP for Meraki MX
- Threat intelligence from Cisco Talos and Cisco Threat Grid

## Cisco AnyConnect

The *Cisco AnyConnect Secure Mobility Client* is a modular endpoint software product that is not only a VPN client that provides VPN access through Transport Layer Security (TLS)/Secure Sockets Layer (SSL) and IPsec IKEv2 but also offers enhanced security through various built-in modules, such as a VPN Posture (HostScan) module and an ISE Posture module. These modules enable Cisco AnyConnect to assess an endpoint's compliance for things like antivirus, antispyware, and firewall software installed on the host. If an endpoint is found to be noncompliant, network access can be restricted until the endpoint is in compliance.

Cisco AnyConnect also includes web security through Cisco Cloud Web Security, network visibility into endpoint flows within Stealthwatch, and roaming protection with Cisco Umbrella—even while the AnyConnect client is not connected to the corporate network through a VPN. AnyConnect is supported across a broad set of platforms, including Windows, macOS, iOS, Linux, Android, Windows Phone/Mobile, BlackBerry, and ChromeOS.

### Cisco Umbrella

Cisco Umbrella (formerly known as OpenDNS) provides the first line of defense against threats on the Internet by blocking requests to malicious Internet destinations (domains, IPs, URLs) using the Domain Name System (DNS) before an IP connection is established or a file is downloaded. It is 100% cloud delivered, with no hardware to install or software to maintain.

## Cisco Web Security Appliance (WSA)

The Cisco *Web Security Appliance (WSA)* is an all-in-one web gateway that includes a wide variety of protections that can block hidden malware from both suspicious and legitimate websites. It leverages real-time threat intelligence from Cisco Talos and Cisco AMP Threat Grid that allows it to stay one step ahead of the evolving threat landscape to prevent the latest exploits from infiltrating the network. It also provides multiple layers of malware defense and vital data loss prevention (DLP) capabilities across the full attack continuum, as illustrated in Figure 25-5.

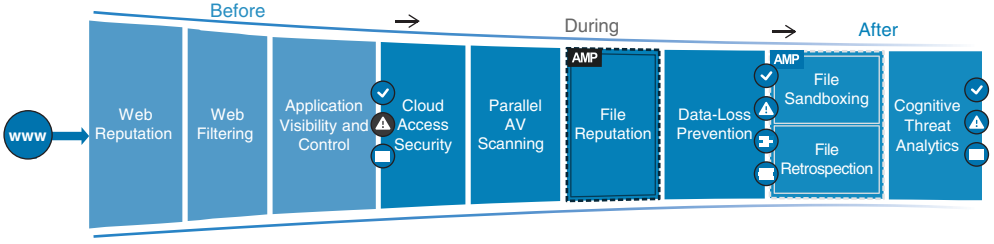



**Figure 25-5** *WSA Capabilities Across the Attack Continuum*

### Cisco Email Security Appliance (ESA)

For business organizations, email is the most important business communication tool, and at the same time, it is one of the top attack vectors for security breaches. The Cisco *Email Security Appliance (ESA)* enables users to communicate securely via email and helps organizations combat email security threats with a multilayered approach across the attack continuum.

A system that passively monitors and analyzes network traffic for potential network intrusion attacks and logs the intrusion attack data for security analysis is known as an *intrusion detection system (IDS)*. A system that provides IDS functions and also automatically blocks intrusion attacks is known as an *intrusion prevention system (IPS)*.

A next-generation IPS (NGIPS), according to Gartner, Inc., should include IPS functionality as well as the following capabilities:

- Real-time contextual awareness
- Advanced threat protection
- Intelligent security automation
- Unparalleled performance and scalability
- Application visibility and control (AVC) and URL filtering

A *firewall* is a network security device that monitors incoming and outgoing network traffic and allows or blocks traffic by performing simple packet filtering and stateful inspection based on ports and protocols. A firewall essentially establishes a barrier between trusted internal networks and untrusted outside networks such as the Internet.

In addition to providing standard firewall functionality, a *next-generation firewall (NGFW)* can block threats such as advanced malware and application-layer attacks. According to Gartner, Inc.'s definition, a NGFW firewall must include:

- Standard firewall capabilities such as stateful inspection
- An integrated IPS
- Application-level inspection (to block malicious or risky apps)
- The ability to leverage external security intelligence to address evolving security threats

## Cisco Stealthwatch

*Cisco Stealthwatch* is a collector and aggregator of network telemetry data that performs network security analysis and monitoring to automatically detect threats that manage to infiltrate a network as well as the ones that originate from within a network. Using advanced security analytics, Stealthwatch can quickly and with high confidence detect threats such as command-and-control (C&C) attacks, ransomware, DDoS attacks, illicit cryptomining, unknown malware, and inside threats. It is an agentless solution that brings threat visibility into every part of the network, including the cloud, and the only product that can detect malware in encrypted traffic and ensure policy compliance without decryption.

There are currently two offerings available for Stealthwatch:

- Stealthwatch Enterprise
- Stealthwatch Cloud

Stealthwatch Enterprise requires the following components:

- **Flow Rate License:** The Flow Rate License is required for the collection, management, and analysis of flow telemetry data and aggregates flows at the Stealthwatch Management Console as well as to define the volume of flows that can be collected.
- **Flow Collector:** The Flow Collector collects and analyzes enterprise telemetry data such as NetFlow, IP Flow Information Export (IPFIX), and other types of flow data from routers, switches, firewalls, endpoints, and other network devices. The Flow Collector can also collect telemetry from proxy data sources, which can be analyzed by Global Threat Analytics (formerly Cognitive Threat Analytics). It can also pinpoint malicious patterns in encrypted traffic using Encrypted Traffic Analytics (ETA) without having to decrypt it to identify threats and accelerate response. Flow Collector is available as a hardware appliance and as a virtual machine.
- **Stealthwatch Management Console (SMC):** The SMC is the control center for Stealthwatch. It aggregates, organizes, and presents analysis from up to 25 Flow Collectors, Cisco ISE, and other sources. It offers a powerful yet simple-to-use web console that provides graphical representations of network traffic, identity information, customized summary reports, and integrated security and network intelligence for comprehensive analysis. The SMC is available as a hardware appliance or a virtual machine.

Cisco Stealthwatch Cloud consists of two primary offerings:

- Public Cloud Monitoring
- Private Network Monitoring

### Cisco Identity Services Engine (ISE)

Cisco *Identity Services Engine (ISE)* is a security policy management platform that provides highly secure network access control (NAC) to users and devices across wired, wireless, and VPN connections. It allows for visibility into what is happening in the network, such as who is connected (endpoints, users, and devices), which applications are installed and running on endpoints (for posture assessment), and much more.

### 802.1x

IEEE 802.1x (referred to as Dot1x) is a standard for port-based network access control (PNAC) that provides an authentication mechanism for local area networks (LANs) and wireless local area networks (WLANs).

802.1x comprises the following components:

- **Extensible Authentication Protocol (EAP):** This message format and framework defined by RFC 4187 provides an encapsulated transport for authentication parameters.
- **EAP method (also referred to as EAP type):** Different authentication methods can be used with EAP.
- **EAP over LAN (EAPoL):** This Layer 2 encapsulation protocol is defined by 802.1x for the transport of EAP messages over IEEE 802 wired and wireless networks.
- **RADIUS protocol:** This is the AAA protocol used by EAP.

802.1x network devices have the following roles:

- **Supplicant:** Software on the endpoint communicates and provides identity credentials through EAPoL with the authenticator. Common 802.1x supplicants include Windows and macOS native supplicants as well as Cisco AnyConnect. All these supplicants support 802.1x machine and user authentication.
- **Authenticator:** A network access device (NAD) such as a switch or wireless LAN controller (WLC) controls access to the network based on the authentication status of the user or endpoint. The authenticator acts as the liaison, taking Layer 2 EAP-encapsulated packets from the supplicant and encapsulating them into RADIUS packets for delivery to the authentication server.
- **Authentication server:** A RADIUS server performs authentication of the client. The authentication server validates the identity of the endpoint and provides the authenticator with an authorization result, such as accept or deny.

The following are the most commonly used EAP methods, which are described in this section:

- EAP challenge-based authentication method
  - Extensible Authentication Protocol-Message Digest 5 (EAP-MD5)
- EAP TLS authentication method
  - Extensible Authentication Protocol-Transport Layer Security (EAP-TLS)
- EAP tunneled TLS authentication methods
  - Extensible Authentication Protocol Flexible Authentication via Secure Tunneling (EAP-FAST)
  - Extensible Authentication Protocol Tunneled Transport Layer Security (EAP-TTLS)
  - Protected Extensible Authentication Protocol (PEAP)
- EAP inner authentication methods
  - EAP Generic Token Card (EAP-GTC)
  - EAP Microsoft Challenge Handshake Authentication Protocol Version 2 (EAP-MSCHAPv2)
  - EAP TLS

### EAP Chaining

EAP-FAST includes the option of EAP chaining, which supports machine and user authentication inside a single outer TLS tunnel. It enables machine and user authentication to be combined into a single overall authentication result. This allows the assignment of greater privileges or posture assessments to users who connect to the network using corporate-managed devices.

## MAC Authentication Bypass (MAB)

*MAC Authentication Bypass (MAB)* is an access control technique that enables port-based access control using the MAC address of an endpoint, and it is typically used as a fallback mechanism to 802.1x. A MAB-enabled port can be dynamically enabled or disabled based on the MAC address of the endpoint that connects to it.

### Web Authentication (WebAuth)

In an organization, endpoints that try to connect to the network might not have 802.1x supplicants and might not know the MAC address to perform MAB. These endpoints can be employees and contractors with misconfigured 802.1x settings that require access to the corporate network or visitors and guests that need access to the Internet. For these cases, Web Authentication (WebAuth) can be used. WebAuth, like MAB, can be used as a fallback authentication mechanism for 802.1x. If both MAB and WebAuth are configured as fallbacks for 802.1x, when 802.1x times out, a switch first attempts to authenticate through MAB, and if it fails, the switch attempts to authenticate with WebAuth.

There are two types of WebAuth:

- Local Web Authentication
- Centralized Web Authentication with Cisco ISE

## Cisco TrustSec

TrustSec is a next-generation access control enforcement solution developed by Cisco to address the growing operational challenges related to maintaining firewall rules and ACLs by using Security Group Tag (SGT) tags.

TrustSec uses SGT tags to perform ingress tagging and egress filtering to enforce access control policy. Cisco ISE assigns the SGT tags to users or devices that are successfully authenticated and authorized through 802.1x, MAB, or WebAuth. The SGT tag assignment is delivered to the authenticator as an authorization option (in the same way as a dACL). After the SGT tag is assigned, an access enforcement policy (allow or drop) based on the SGT tag can be applied at any egress point of the TrustSec network.

TrustSec configuration occurs in three phases:

- Ingress classification
- Propagation
- Egress enforcement

There are two methods available for propagating an SGT tag—inline tagging (also referred to as *native tagging*) and the Cisco-created protocol SGT Exchange Protocol (SXP):

- **Inline tagging:** With inline tagging, a switch inserts the SGT tag inside a frame to allow upstream devices to read and apply policy. Native tagging is completely independent of any Layer 3 protocol (IPv4 or IPv6), so the frame or packet can preserve the SGT tag throughout the network infrastructure (routers, switches, firewalls, and so on) until it reaches the egress point. The downside to native tagging is that it is supported only by Cisco network devices with ASIC support for TrustSec. If a tagged frame is received by a device that does not support native tagging in hardware, the frame is dropped. Figure 25-10 illustrates a Layer 2 frame with a 16-bit SGT value.

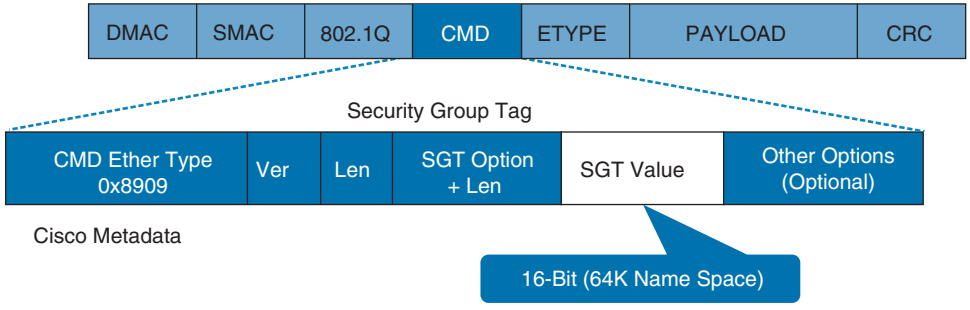



**Figure 25-10** *Layer 2 Ethernet Frame with an SGT Tag*

- **SXP propagation:** SXP is a TCP-based peer-to-peer protocol used for network devices that do not support SGT inline tagging in hardware. Using SXP, IP-to-SGT mappings can be communicated between non-inline tagging switches and other network devices. Non-inline tagging switches also have an SGT mapping database to check packets against and enforce policy. The SXP peer that sends IP-to-SGT bindings is called a *speaker*. The IP-to-SGT binding receiver is called a *listener*. SXP connections can be single-hop or multi-hop, as shown in Figure 25-11.

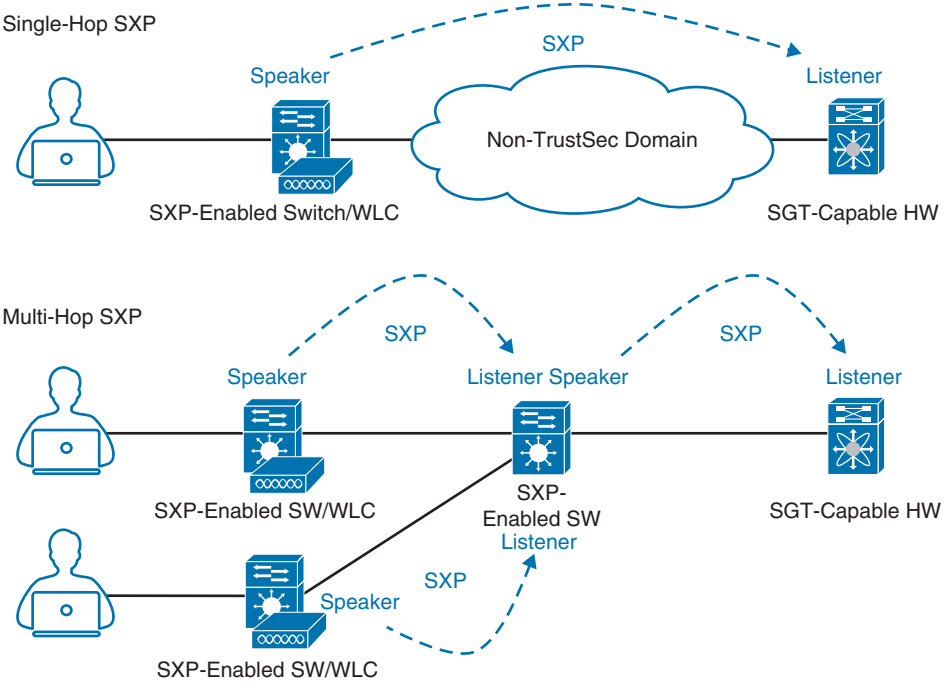



**Figure 25-11** *Single-Hop and Multi-Hop SXP Connections*

There are multiple ways to enforce traffic based on the SGT tag, and they can be divided into two major types:

- **Security Group ACL (SGACL):** Provides enforcement on routers and switches. Access lists provide filtering based on source and destination SGT tags.
- **Security Group Firewall (SGFW):** Provides enforcement on firewalls (such as Cisco ASA and NGFW). Requires tag-based rules to be defined locally on the firewall.

## MACsec

*MACsec* is an IEEE 802.1AE standards-based Layer 2 hop-by-hop encryption method; this means the traffic is encrypted only on the wire between two MACsec peers and is unencrypted as it is processed internally within the switch. This allows the switch to look into the inner packets for things like SGT tags to perform packet enforcement or QoS prioritization. MACsec also leverages onboard ASICs to perform the encryption and decryption rather than having to offload to a crypto engine, as with IPsec.

MACsec is based on the Ethernet frame format; however, an additional 16-byte MACsec Security Tag field (802.1AE header) and a 16-byte Integrity Check Value (ICV) field are added. This means that all devices in the flow of the MACsec communications must support MACsec for these fields to be used and to secure the traffic. MACsec provides authentication using Galois Method Authentication Code (GMAC) or authenticated encryption using Galois/Counter Mode Advanced Encryption Standard (AES-GCM).

Two MACsec keying mechanisms are available:

- **Security Association Protocol (SAP):** This is a proprietary Cisco keying protocol used between Cisco switches.
- **MACsec Key Agreement (MKA) protocol:** MKA provides the required session keys and manages the required encryption keys. The 802.1AE encryption with MKA is supported between endpoints and the switch as well as between switches.

### Downlink MACsec

*Downlink MACsec* is the term used to describe the encrypted link between an endpoint and a switch. The encryption between the endpoint and the switch is handled by the MKA keying protocol. This requires a MACsec-capable switch and a MACsec-capable supplicant on the endpoint (such as Cisco AnyConnect). The encryption on the endpoint may be handled in hardware (if the endpoint possesses the correct hardware) or in software, using the main CPU for encryption and decryption.

### Uplink MACsec

*Uplink MACsec* is the term for encrypting a link between switches with 802.1AE. By default, uplink MACsec uses Cisco proprietary SAP encryption. The encryption is the same AES-GCM-128 encryption used with both uplink and downlink MACsec.

Uplink MACsec may be achieved manually or dynamically. Dynamic MACsec requires 802.1x authentication between the switches.

# Chapter 26

## Access Control Lists (ACLs)

*Access control lists* (also known as *ACLs* or *access lists*) are sequential lists of access control entries (ACEs) that perform permit or deny packet classification, based on predefined conditional matching statements. Packet classification starts at the top (lowest sequence) and proceeds down (higher sequence) until a matching pattern is identified. When a match is found, the appropriate action (permit or deny) is taken, and processing stops. At the end of every ACL is an implicit deny ACE, which denies all packets that did not match earlier in the ACL.

While many different kinds of ACLs can be used for packet filtering, only the following types are covered in this chapter:

- **Numbered standard ACLs:** These ACLs define packets based solely on the source network, and they use the numbered entries 1–99 and 1300–1999.
- **Numbered extended ACLs:** These ACLs define packets based on source, destination, protocol, port, or a combination of other packet attributes, and they use the numbered entries 100–199 and 2000–2699.
- **Named ACLs:** These ACLs allow standard and extended ACLs to be given names instead of numbers and are generally preferred because they can provide more relevance to the functionality of the ACL.
- **Port ACLs (PACLs):** These ACLs can use standard, extended, named, and named extended MAC ACLs to filter traffic on Layer 2 switchports.
- **VLAN ACLs (VACLs):** These ACLs can use standard, extended, named, and named extended MAC ACLs to filter traffic on VLANs.

ACLs have no effect until they are applied to an interface. Therefore, the next step after creating an ACL is to apply it to an interface. In addition to the interface, the direction (in or out) in which the ACL needs to be applied has to be specified. Cisco routers allow only one inbound ACL and one outbound ACL per interface.

There are three basic methods to gain access to the CLI of an IOS device:

- **Console port (cty) line:** On any IOS device, this appears in configuration as **line con 0** and in the output of the command **show line** as **cty**. The console port is mainly used for local system access using a console terminal.
- **Auxiliary port (aux) line:** This appears in the configuration as **line aux 0**. The aux port is mainly used for remote access into the device through a modem.
- **Virtual terminal (vty) lines:** These lines are displayed by default in the configuration as **line vty 0 4**. They are used solely for remote Telnet and SSH connections. They are virtual because they are logical lines with no physical interface associated to them.

Each of these types of terminal lines should be password protected. There are three ways to add password protection to the lines:

- **Using a password configured directly on the line:** Not recommended
- **Using username-based authentication:** Recommended as a fallback
- **Using an AAA server:** Highly recommended and covered later in this chapter, in the section "Authentication, Authorization, and Accounting (AAA)"

## Password Types

There are five available password types in Cisco IOS:

- **Type 0 passwords:** These passwords are the most insecure because they are not encrypted and are visible in the device configuration in plaintext. The command **enable password** is an example of a command that uses a type 0 password. Type 0 passwords should be avoided whenever possible.
- **Type 5 passwords:** These passwords use an improved Cisco proprietary encryption algorithm that makes use of the MD5 hashing algorithm. This makes them much stronger because they are considered not reversible (uncrackable). The only way to crack type 5 passwords is by performing brute-force attacks. It is strongly recommended that you use type 5 encryption instead of type 0 or type 7 whenever possible. Type 5 encryption is applied by using the command **enable secret** to specify an additional layer of security over the command **enable password**. The command **enable password** should be used only on platforms with legacy IOS that do not support the command **enable secret**. If the command **enable secret** and the command **enable password** are configured concurrently, the command **enable secret** is preferred. The **username secret** command also uses type 5 encryption.
- **Type 7 passwords:** These passwords use a Cisco proprietary Vigenere cypher encryption algorithm and are known to be weak. There are multiple online password utilities available that can decipher type 7 encrypted passwords in less than a second. Type 7 encryption is enabled by the command **service password-encryption** for commands that use type 0 passwords, such as the **enable password**, **username password**, and **line password** commands.
- **Type 8 passwords:** Type 8 passwords specify a Password-Based Key Derivation Function 2 (PBKDF2) with a SHA-256 hashed secret and are considered to be uncrackable.
- **Type 9 passwords:** These use the SCRYPT hashing algorithm. Just like type 8 passwords, they are considered to be uncrackable.

There are three different ways to configure a username on IOS:

- Using the command **username** {*username*} **password** {*password*} configures a plaintext password (type 0).
- Using the command **username** {*username*} **secret** {*password*} provides type 5 encryption.
- Using the command **username** {*username*} **algorithm-type** {**md5** | **sha256** | **scrypt**} **secret** {*password*} provides type 5, type 8, or type 9 encryption, respectively.

The Cisco IOS CLI by default includes three privilege levels, each of which defines what commands are available to a user:

- **Privilege level 0:** Includes the **disable**, **enable**, **exit**, **help**, and **logout** commands.
- **Privilege level 1:** Also known as User EXEC mode. The command prompt in this mode includes a greater-than sign (R1>). From this mode it is not possible to make configuration changes; in other words, the command **configure terminal** is not available.
- **Privilege level 15:** Also known as Privileged EXEC mode. This is the highest privilege level, where all CLI commands are available. The command prompt in this mode includes a hash sign (R1#).

SSH, which provides secure encryption and strong authentication, is available in two versions:

- **SSH Version 1 (SSHv1):** This is an improvement over using plaintext Telnet, but some fundamental flaws exist in its implementation, so it should be avoided in favor of SSHv2.
- **SSH Version 2 (SSHv2):** This is a complete rework and stronger version of SSH that is not compatible with SSHv1. SSHv2 has many benefits and closes a security hole that is found in SSH version 1. SSH version 2 is certified under the National Institute of Standards and Technology (NIST) Federal Information Processing Standards (FIPS) 140-1 and 140-2 U.S. cryptographic standards and should be used where feasible.

AAA is an architectural framework for enabling a set of three independent security functions:

- **Authentication:** Enables a user to be identified and verified prior to being granted access to a network device and/or network services.
- **Authorization:** Defines the access privileges and restrictions to be enforced for an authenticated user.
- **Accounting:** Provides the ability to track and log user access, including user identities, start and stop times, executed commands (that is, CLI commands), and so on. In other words, it maintains a security log of events.

AAA is commonly used in the networking industry for the following two use cases:

- **Network device access control:** As described earlier in this chapter, Cisco IOS provides local features for simple device access control, such as local username-based authentication and line password authentication. However, these features do not provide the same degree of access control and scalability that is possible with AAA. For this reason, AAA is the recommended method for access control. TACACS+ is the protocol of choice for network device access control.
- **Secure network access control:** AAA can be used to obtain the identity of a device or user before that device or user is allowed to access to the network. RADIUS is the preferred protocol for secure network access. Secure network access control is covered in Chapter 25, "Secure Network Access Control."

One of the key differentiators of TACACS+ is its capability to separate authentication, authorization, and accounting into independent functions. This is why TACACS+ is so commonly used for device administration instead of RADIUS, even though RADIUS is capable of providing network device access control.

RADIUS is an IETF standard AAA protocol. As with TACACS+, it follows a client/server model, where the client initiates the requests to the server. RADIUS is the AAA protocol of choice for secure network access. The reason for this is that RADIUS is the AAA transport protocol for Extensible Authentication Protocol (EAP), while TACACS+ does not support this functionality. EAP is used for secure network access and is covered in Chapter 23.

Cisco *Zone-Based Firewall (ZBFW)* is the latest integrated stateful firewall technology included in IOS. ZBFW reduces the need for a firewall at a branch site to provide stateful network security.

ZBFW uses a flexible and straightforward approach to providing security by establishing security zones. Router interfaces are assigned to a specific zone, which can maintain a one-to-one or many-to-one relationship. A zone establishes a security border on the network and defines acceptable traffic that is allowed to pass between zones. By default, interfaces in the same security zone can communicate freely with each other, but interfaces in different zones cannot communicate with each other.

Within the ZBFW architecture, there are two system-built zones: self and default.

## Control Plane Policing (CoPP)

A *control plane policing (CoPP)* policy is a QoS policy that is applied to traffic to or sourced by the router's control plane CPU. CoPP policies are used to limit known traffic to a given rate while protecting the CPU from unexpected extreme rates of traffic that could impact the stability of the router.

Typical CoPP implementations use only an input policy that allows traffic to the control plane to be policed to a desired rate. In a properly planned CoPP policy, network traffic is placed into various classes, based on the type of traffic (management, routing protocols, or known IP addresses). The CoPP policy is then implemented to limit traffic to the control plane CPU to a specific rate for each class.

# Chapter 27

## Server Virtualization

One of the main drivers behind server virtualization was that server hardware resources were being underutilized; physical servers were typically each running a single operating system with a single application and using only about 10% to 25% of the CPU resources. VMs and containers increase the overall efficiency and cost-effectiveness of a server by maximizing the use of the available resources.

A *virtual machine (VM)* is a software emulation of a physical server with an operating system. From an application's point of view, the VM provides the look and feel of a real physical server, including all its components, such as CPU, memory, and network interface cards (NICs). The virtualization software that creates VMs and performs the hardware abstraction that allows multiple VMs to run concurrently is known as a *hypervisor*. VMware vSphere, Microsoft Hyper-V, Citrix XenServer, and Red Hat Kernel-based Virtual Machine (KVM) are the most popular hypervisors in the server virtualization market. Figure 27-1 provides a side-by-side comparison of a bare-metal server and a server running virtualization software.

There are two types of hypervisors, as illustrated in Figure 27-2:

- **Type 1:** This type of hypervisor runs directly on the system hardware. It is commonly referred to as "bare metal" or "native."
- **Type 2:** This type of hypervisor (for example, VMware Fusion) requires a host OS to run. This is the type of hypervisor that is typically used by client devices.

A *container* is an isolated environment where containerized applications run. It contains the application, along with the dependencies that the application needs to run. Even though they have these and many other similarities to VMs, containers are not the same as VMs, and they should not be referred to as "lightweight VMs."

A *virtual switch (vSwitch)* is a software-based Layer 2 switch that operates like a physical Ethernet switch. A vSwitch enables VMs to communicate with each other within a virtualized server and with external physical networks through the physical network interface cards (pNICs). Multiple vSwitches can be created under a virtualized server, but network traffic cannot flow directly from one vSwitch to another vSwitch within the same host, and the vSwitches cannot share the same pNIC.

*Network functions virtualization (NFV)* is an architectural framework created by the European Telecommunications Standards Institute (ETSI) that defines standards to decouple network functions from proprietary hardware-based appliances and have them run in software on standard x86 servers. It also defines how to manage and orchestrate the network functions. *Network function (NF)* refers to the function performed by a physical appliance, such as a firewall or a router function.

To overcome the performance impact on throughput due to interrupts, OVS was enhanced with the Data Plane Development Kit (DPDK) libraries. OVS with DPDK operates entirely in user space. The DPDK Poll Mode Driver (PMD) in OVS polls for data that comes into the pNIC and processes it, bypassing the network stack and the need to send an interrupt to the CPU when a packet is received—in other words, bypassing the kernel entirely. To be able to do this, DPDK PMD requires one or more CPU cores dedicated to polling and handling the incoming data. Once the packet is in OVS, it's already in user space, and it can then be switched directly to the appropriate VNF, resulting in huge performance benefits.

PCI passthrough allows VNFs to have direct access to physical PCI devices, which appear and behave as if they were physically attached to the VNF. This technology can be used to map a pNIC to a single VNF, and from the VNF's perspective, it appears as if it is directly connected to the pNIC.

SR-IOV is an enhancement to PCI passthrough that allows multiple VNFs to share the same pNIC. SR-IOV emulates multiple PCIe devices on a single PCIe device (such as a pNIC). In SR-IOV, the emulated PCIe devices are called *virtual functions (VFs)*, and the physical PCIe devices are called *physical functions (PFs)*. The VNFs have direct access to the VFs, using PCI passthrough technology.

The Cisco ENFV solution is a Cisco solution based on the ETSI NFV architectural framework. It reduces the operational complexity of enterprise branch environments by running the required networking functions as virtual networking functions (VNFs) on standard x86-based hosts. In other words, it replaces physical firewalls, routers, WLC, load balancers, and so on with virtual devices running in a single x86 platform. The Cisco ENFV solution provides the following benefits:

- Reduces the number of physical devices to be managed at the branch, resulting in efficiencies in space, power, maintenance, and cooling
- Reduces the need for truck rolls and technician site visits to perform hardware installations or upgrades
- Offers operational simplicity that allows it to roll out new services, critical updates, VNFs, and branch locations in minutes
- Centralizes management through Cisco DNA Center, which greatly simplifies designing, provisioning, updating, managing, and troubleshooting network services and VNFs
- Enhances network operations flexibility by taking full advantage of virtualization techniques such as virtual machine moves, snapshots, and upgrades
- Supports Cisco SD-WAN cEdge and vEdge virtual router onboarding
- Supports third-party VNFs

Cisco ENFV delivers a virtualized solution for network and application services for branch offices. It consists of four main components that are based on the ETSI NFV architectural framework:

- **Management and Orchestration (MANO):** Cisco DNA Center provides the VNF management and NFV orchestration capabilities. It allows for easy automation of the deployment of virtualized network services, consisting of multiple VNFs.
- **VNFs:** VNFs provide the desired virtual networking functions.
- **Network Functions Virtualization Infrastructure Software (NFVIS):** An operating system that provides virtualization capabilities and facilitates the deployment and operation of VNFs and hardware components.
- **Hardware resources:** x86-based compute resources that provide the CPU, memory, and storage required to deploy and operate VNFs and run applications.

### Management and Orchestration

Cisco DNA Center provides the MANO functionality to the Cisco Enterprise NFV solution. It includes a centralized dashboard and tools to design, provision, manage, and monitor all branch sites across the enterprise. Two of the main functions of DNA Center are to roll out new branch locations or deploy new VNFs and virtualized services.

Cisco DNA Center provides centralized policies, which enables consistent network policies across the enterprise branch offices. Centralized policies are created by building network profiles. Multiple network profiles can be created, each with specific design requirements and virtual services. Once they are created, branch sites are then assigned to network profiles that match the branch requirements.

### Virtual Network Functions and Applications

The Cisco Enterprise NFV solution provides an environment for the virtualization of both network functions and applications in the enterprise branch. Both Cisco and third-party VNFs can be onboarded onto the solution. Applications running in a Linux server or Windows server environment can also be instantiated on top of NFVIS (discussed later in this chapter) and can be supported by DNA Center.

### Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS is based on standard Linux packaged with additional functions for virtualization, VNF lifecycle management, monitoring, device programmability, and hardware acceleration. The components and functionality delivered by NFVIS are illustrated in Figure 27-16:

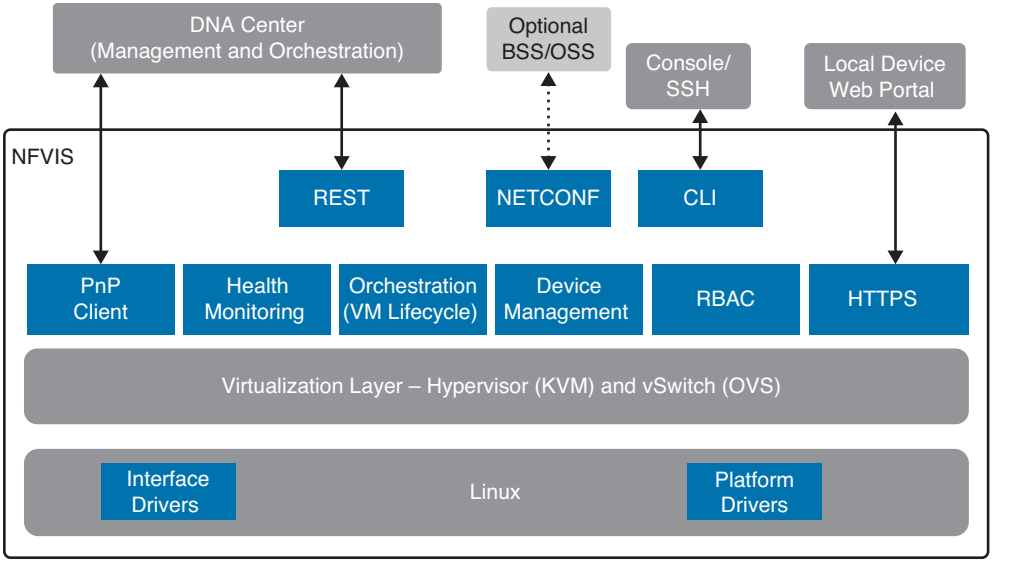



**Figure 27-16** *NFVIS Components*

# Chapter 28

Table 28-3 lists some of the most common HTTP functions and their associated use cases.

**Table 28-3** HTTP Functions and Use Cases

| HTTP Function | Action | Use Case |
|---|---|---|
| GET | Requests data from a destination | Viewing a website |
| POST | Submits data to a specific destination | Submitting login credentials |
| PUT | Replaces data in a specific destination | Updating an NTP server |
| PATCH | Appends data to a specific destination | Adding an NTP server |
| DELETE | Removes data from a specific destination | Removing an NTP server |

Table 28-4 lists the CRUD functions and their associated actions and use cases.

**Table 28-4** CRUD Functions and Use Cases

| CRUD Function | Action | Use Case |
|---|---|---|
| CREATE | Inserts data in a database or application | Updating a customer's home address in a database |
| READ | Retrieves data from a database or application | Pulling up a customer's home address from a database |
| UPDATE | Modifies or replaces data in a database or application | Changing a street address stored in a database |
| DELETE | Removes data from a database or application | Removing a customer from a database |

Table 28-5 lists the most common HTTP status codes as well as the reasons users may receive each one.

**Table 28-5** HTTP Status Codes

| HTTP Status Code | Result | Common Reason for Response Code |
|---|---|---|
| 200 | OK | Using GET or POST to exchange data with an API |
| 201 | Created | Creating resources by using a REST API call |
| 400 | Bad Request | Request failed due to client-side issue |
| 401 | Unauthorized | Client not authenticated to access site or API call |
| 403 | Forbidden | Access not granted based on supplied credentials |
| 404 | Not Found | Page at HTTP URL location does not exist or is hidden |

The key steps necessary to successfully set up the API call in Postman are as follows:

**Step 1.** In the URL bar, enter **https://sandboxdnac.cisco.com/api/system/v1/auth/token** to target the Token API.

**Step 2.** Select the HTTP POST operation from the dropdown box.

**Step 3.** Under the Authorization tab, ensure that the type is set to Basic Auth.

**Step 4.** Enter **devnetuser** as the username and **Cisco123!** as the password.

**Step 5.** Select the Headers tab and enter **Content-Type** as the key.

**Step 6.** Select application/json as the value.

**Step 7.** Click the Send button to pass the credentials to the Cisco DNA Center controller via the Token API.

You need to prepare Postman to use the token that was generated when you successfully authenticated to the controller by following these steps:

**Step 1.** Copy the token you received earlier and click a new tab in Postman.

**Step 2.** In the URL bar enter **https://sandboxdnac.cisco.com/api/v1/network-device** to target the Network Device API.

**Step 3.** Select the HTTP GET operation from the dropdown box.

**Step 4.** Select the Headers tab and enter **Content-Type** as the key.

**Step 5.** Select application/json as the value.

**Step 6.** Add another key and enter **X-Auth-Token**.

**Step 7.** Paste the token in as the value.

**Step 8.** Click Send to pass the token to the Cisco DNA Center controller and perform an HTTP GET to retrieve a device inventory list using the Network Device API.

In Postman, it is possible to modify the Network Device API URL and add **?limit=1** to the end of the URL to show only a single device in the inventory. It is also possible to add the **&offset=2** command to the end of the URL to state that only the second device in the inventory should be shown. These query parameters are part of the API and can be invoked using a client like Postman as well. Although it may sound confusing, the **limit** keyword simply states that a user only wants to retrieve one record from the inventory; the **offset** command states that the user wants that one record to be the second record in the inventory. Figure 28-10 shows how to adjust the Network Device API URL in Postman to show information on only the second device in the inventory.

# Chapter 29

*Embedded Event Manager (EEM)* is a very flexible and powerful Cisco IOS tool. EEM allows engineers to build software applets that can automate many tasks. EEM also derives some of its power from the fact that it enables you to build custom scripts using Tcl. Scripts can automatically execute, based on the output of an action or an event on a device. One of the main benefits of EEM is that it is all contained within the local device. There is no need to rely on an external scripting engine or monitoring device in most cases. Figure 29-1 illustrates some of the EEM event detectors and how they interact with the IOS subsystem.

## Puppet

Puppet is a robust configuration management and automation tool. Cisco supports the use of Puppet on a variety of devices, such as Catalyst switches, Nexus switches, and the Cisco Unified Computing System (UCS) server platform. Puppet works with many different vendors and is one of the more commonly used tools used for automation. Puppet can be used during the entire lifecycle of a device, including initial deployment, configuration management, and repurposing and removing devices in a network.

Puppet uses the concept of a *puppet master* (server) to communicate with devices that have the *puppet agent* (client) installed locally on the device. Changes and automation tasks are executed within the *puppet console* and then shared between the puppet master and puppet agents. These changes or automation tasks are stored in the *puppet database* (PuppetDB), which can be located on the same puppet master server or on a separate box. This allows the tasks to be saved so they can be pushed out to the puppet agents at a later time.

## Chef

Chef is an open source configuration management tool that is designed to automate configurations and operations of a network and server environment. Chef is written in Ruby and Erlang, but when it comes to actually writing code within Chef, Ruby is the language used.

### SaltStack (Agent and Server Mode)

SaltStack is another configuration management tool, in the same category as Chef and Puppet. Of course, SaltStack has its own unique terminology and architecture. SaltStack is built on Python, and it has a Python interface so a user can program directly to SaltStack by using Python code. However, most of the instructions or states that get sent out to the nodes are written in YAML or a DSL. These are called *Salt formulas*. Formulas can be modified but are designed to work out of the box. Another key difference from Puppet and Chef is SaltStack's overall architecture. SaltStack uses the concept of *systems*, which are divided into various categories. For example, whereas the Puppet architecture has a puppet master and puppet agents, SaltStack has *masters* and *minions*.

## Ansible

Ansible is an automation tool that is capable of automating cloud provisioning, deployment of applications, and configuration management. Ansible has been around for quite some time and was catapulted further into the mainstream when RedHat purchased the company in 2015. Ansible has grown very popular due to its simplicity and the fact that it is open source. Ansible was created with the following concepts in mind:

- Consistent
- Secure
- Highly reliable
- Minimal learning curve

### Puppet Bolt

Puppet Bolt allows you to leverage the power of Puppet without having to install a puppet master or puppet agents on devices or nodes. Much like Ansible, Puppet Bolt connects to devices by using SSH or WinRM connections. Puppet Bolt is an open source tool that is based on the Ruby language and can be installed as a single package.

In Puppet Bolt, tasks can be used for pushing configuration and for managing services, such as starting and stopping services and deploying applications. Tasks are sharable. For example, users can visit Puppet Forge to find and share tasks with others in the community. Tasks are really good for solving problems that don't fit in the traditional model of client/server or puppet master and puppet agent. As mentioned earlier in this chapter, Puppet is used to ensure configuration on devices and can periodically validate that the change or specific value is indeed configured. Puppet Bolt allows you to execute a change or configuration immediately and then validate it.

### SaltStack SSH (Server-Only Mode)

SaltStack offers an agentless option called Salt SSH that allows users to run Salt commands without having to install a minion on the remote device or node. This is similar in concept to Puppet Bolt. The main requirements to use Salt SSH are that the remote system must have SSH enabled and Python installed.

**Table 29-7** High-Level Configuration Management and Automation Tool Comparison

| Factor | Puppet | Chef | Ansible | SaltStack |
|---|---|---|---|---|
| Architecture | Puppet masters and puppet agents | Chef server and Chef clients | Control station and remote hosts | Salt master and minions |
| Language | Puppet DSL | Ruby DSL | YAML | YAML |
| Terminology | Modules and manifests | Cookbooks and recipes | Playbooks and plays | Pillars and grains |
| Support for large-scale deployments | Yes | Yes | Yes | Yes |
| Agentless version | Puppet Bolt | N/A | Yes | Salt SSH |