CCNP BCMSN Notes

31 Mar 2008

Chapter 3: Switch Operation

Layer 2 Switching

Switching Decision

Factors in a switching decision:

Layer 2 forwarding table - *Content Addressable Memory (CAM)* table

Security ACLs - Access lists are stored in compiled form in the *Ternary CAM (TCAM)*

QoS ACLs - Used to police traffic flow, also stored in the TCAM

Multilayer Switching

Route Caching

Route caching is the first generation multilayer switching.

Requires a *route processor (RP)* and *switching engine (SE)*.

The RP routes the first packet in a flow, and creates a record for the destination in the MLS cache.

The SE forwards all subsequent packets for that destination based on the MLS cache entry.

Route caching is used by NetFlow to generate traffic statistics.

Topology-based

Second generation multilayer switching, known as *Cisco Express Forwarding (CEF)*.

Layer 3 routing information builds a database containing the entire network topology, contained in hardware *Forwarding Information Base (FIB)*.

The hardware database can be updated dynamically with no performance penalty.

Switching Decision

Layer 2 forwarding table - The destination MAC is checked against the CAM table to determine if the frame contains a layer 3 packet (if the MAC address belongs to a layer 3 interface on the switch)

Layer 3 forwarding table - The destination IP is checked against the FIB; the next-hop IP,

next-hop MAC, and egress port (and VLAN) are returned

Security ACLs - Same as in L2

QoS ACLs - Same as in L2

Multilayer Switching Exceptions

Packets which require processing cannot be forwarded by CEF:

ARP

IP packets requiring a response from the router

IP broadcasts relayed as unicasts (via IP helpers)

Routing protocol updates

CDP

IPX routing protocol and service advertisements

Packets needing encryption

Packets requiring *Network Address Translation (NAT)*

Other non-IP and non-IPX packets

Switching Tables

Content Addressable Memory (CAM)

The CAM table stores MAC-to-port/VLAN bindings on all Catalyst switches.

CAM is updated with each frame received.

The CAM table can be inspected with `show mac address-table`.

Ternary Content Addressable Memory (TCAM)

TCAMs facilitate the processing of inbound and outbound security and QoS ACLs in hardware.

Physically separate memory allows ACLs checks to be done in parallel with forwarding decisions.

The *Feature Manager (FM)* compiles ACLs into machine code and inserts them into the TCAM.

The *Switching Database Manager (SDM)* allows for configuration and repartitioning of the TCAM.

TCAMs operate with *values*, *masks*, and *results*:

Value - 134-bit value composed of source and destination addresses and other protocol information; format is dependent on ACL type

Mask - 134-bit mask in the same format as its complement value; used to mark bits which must be matched in the value

Result - A numerical value which represents which action should be taken next

Layer 4 port ranges are stored in *Logical Operation Unit (LOU)* registers.

Chapter 4: Switch Port Configuration

IEEE standards:

802.3 - Ethernet

802.3u - Fast Ethernet

802.3z - Gigabit Ethernet

802.3ab - Gigabit Ethernet over copper (1000Base-T)

802.3ae - 10 Gigabit Ethernet

Gigabit Ethernet was the result of merging IEEE 802.3 Ethernet and ANSI X3T11 FibreChannel.

GBIC/SFP types:

1000Base-SX

1000Base-LX/LH

1000Base-ZX

1000Base-T

GigaStack (Cisco Proprietary)

Fiber modules take the Rx fiber on the left and Tx on the right (when facing the module).

Interface error disabling can be tuned in global config:

```
Switch(config)# [no] errdisable detect cause [ all | <cause> ]
Switch(config)# errdisable recovery cause [ all | <cause> ]
Switch(config)# errdisable interval <seconds>
```

Chapter 5: VLANs and Trunks

The normal range allows for VLANs 1 - 1005; IEEE 802.1Q expands this to 1 - 4095.

VTP version 1 and 2 only support VLANs 1 - 1005. VTPv3 will support extended VLANs but isn't available yet.

Dynamic VLANs can be configured by a *VLAN Membership Policy Server (VMPS)*.

Trunk types:

Inter-Switch Link (ISL) - Cisco proprietary; 26-byte header

IEEE 802.1Q - Open standard; 4-byte header

IEEE 802.3ac extends the Ethernet MTU to 1522 to account for the 4-byte 802.1Q header.

Trunk configuration:

```
Switch(config-if)# switchport mode {trunk | dynamic {desirable | auto}}
Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}
Switch(config-if)# switchport trunk native vlan <vlan>
Switch(config-if)# switchport trunk allowed vlan <list>
```

Chapter 6: VLAN Trunking Protocol

VTP modes:

Client - Rely on VLAN information advertised by a server; no local configuration possible

Server - Have full control over VLAN creation and modification for the VTP domain

Transparent - Does not participate in VTP but will forward advertisements

VTP advertisements:

Summary - Sent every 300 seconds or whenever a change occurs

Subset - Sent after a summary advertisement when a change occurs; these provide more detail to reflect the change that was made

Advertisement Request - Issued by clients to request VLAN information

Pruning Request - Sent from clients to servers to announce active VLANs; inactive VLANs may be pruned from the trunk

VTP configuration:

```
Switch(config)# vtp mode {server | client | transparent}
```

```
Switch(config)# vtp version {1 | 2}
Switch(config)# vtp domain <domain>
Switch(config)# vtp password <password>
```

VTP pruning can be enabled with `vtp pruning`.

Verification:

```
show vtp status
show vtp counters
```

Chapter 7: Aggregating Switch Links

EtherChannel Load Balancing

EtherChannel distributes load across multiple physical links by examining between one and three low order bits of an arbitrary address. XOR is used when multiple addresses are examined.

Address types eligible for examination:

Source and destination MAC - `src-mac` (default for L2 channels), `dst-mac`, or `src-dst-mac`

Source and destination IP - `src-ip`, `dst-ip`, or `src-dst-ip` (default for L3 channels)

Source and destination L4 port - `src-port`, `dst-port`, or `src-dst-port` (Catalyst 6500/4500 only)

Port channel load balancing is configured globally:

```
Switch(config)# port-channel load-balance <method>
```

EtherChannel Negotiation

Port Aggregation Protocol (PAgP)

PAgP is Cisco proprietary.

Port channels are configured as `desirable` (active) or `auto` (passive).

Addition of the `non-silent` parameter will ensure the etherchannel will not be formed without receiving PAgP packets from the neighbor.

Configuring PAgP:

```
Switch(config)# interface range f0/1 - 4
Switch(config-if)# channel-protocol pagp
Switch(config-if)# channel-group <group number> mode {auto | desirable} [non-silent]
```

Link Aggregation Control Protocol (LACP)

LACP is defined in IEEE 802.3ad.

The switch with the lowest priority designates which interfaces participate in the etherchannel.

Interfaces are configured as `active` or `passive`.

`lacp port-priority <priority>` is used to assign an LACP priority to individual ports.

Lower-priority interfaces beyond the eight-port limit for a single channel will be designated as *standby* interfaces should one of the higher-priority links fail.

Configuring LACP:

```
Switch(config)# lacp system-priority <priority>
Switch(config)# interface range f0/1 - 4
Switch(config-if)# channel-protocol lacp
Switch(config-if)# channel-group <group number> mode {passive | active}
```

Static

EtherChannel interfaces can be set to on, forming a permanent etherchannel with no autonegotiation protocol (neither PAgP or LACP is used).

```
Switch(config)# interface range f0/1 - 4
Switch(config-if)# channel-group <group number> mode on
```

Troubleshooting

```
show etherchannel summary
show etherchannel port
show etherchannel load-balance
show {pagp | lacp} neighbor
```

Chapter 8: Traditional Spanning Tree Protocol

STP is defined in IEEE 802.1D.

BPDUs

STP messages are carried by *Bridge Protocol Data Unit (BPDU)* frames; BPDUs are multicast to `01:80:c2:00:00:00`.

BPDU types:

Configuration - Used for spanning-tree computation

Topology Change Notification (TCN) - Used to announce changes in the network

Configuration BPDUs are sent out every port every two seconds by default.

Root Bridge Election

A *root bridge* is elected to serve as a common reference point for the topology.

A switch's *bridge ID* is composed of two parts:

Bridge priority (2 bytes) - Administratively set; defaults to 32,768 (0x8000)

MAC address (6 bytes) - One of the switch's MAC addresses

All switches assume they are the root bridge at boot. The actual root bridge is the switch with the lowest bridge ID.

Configuration BPDUs are only generated by the root bridge; all other bridges insert their own sender ID and relay them.

Root Port Election

All non-root switches must designate a single interface as the *root port* (the port with the best path to the root bridge).

All interfaces are assigned an 8-bit *cost* derived from their speed.

Port costs:

Bandwidth	Cost
4 Mbps	250
10 Mbps	100

16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

The port with the lowest path cost to the root bridge is designated as the root port.

The root path cost noted in a BPDU is incremented by the cost assigned to the port on which it was received.

Designated Port Selection

If multiple switches reside on a segment, the one with the lowest root path cost has the *designated port*; the other ports will be set to blocking.

Designated port selection process:

1. Lowest root bridge ID
2. Lowest root path cost
3. Lowest sender bridge ID
4. Lowest sender port ID

STP States

Disabled - Shutdown

Blocking - The first state when an interface comes up; only receives BPDUs; indefinite duration

Listening - Can send and receive BPDUs; able to participate in STP; duration specified by forward delay timer

Learning - Can send and receive BPDUs and learn MAC addresses; duration specified by forward delay timer

Forwarding - Normal operation; indefinite duration

STP Timers

Hello Time - The rate at which configuration BPDUs are advertised by the root bridge (default is 2 seconds)

Forward Delay - Length of time a port spends in both the listening and learning states (default is 15 seconds)

Max Age - Life of the most recent BPDU advertised from the root bridge (default is 20 seconds)

Timers can be individually adjusted manually on the root bridge, or automatically adjusted by modifying the network diameter (number of switch hops which extend from the root).

STP Types

Common Spanning Tree (CST) - Defined in 802.1Q; one tree for all VLANs

Per-VLAN Spanning Tree (PVST) - Cisco proprietary; one tree per VLAN

Per-VLAN Spanning Tree Plus (PVST+) - PVST featuring compatibility with CST BPDUs

Chapter 9: Spanning Tree Configuration

Root Bridge Configuration

The root bridge should be positioned centrally in the network to ensure the spanning tree forms in a predictable manner.

Two bridge ID formats are available:

802.1D Standard - 16-bit priority + unique MAC address for the VLAN

802.1t Extended - 4-bit priority multiplier + 12-bit VLAN ID + non-unique MAC address

The extended ID format is enabled by default, or with `spanning-tree extend system-id`.

An extended system ID priority must be a multiple of 4096.

There are two ways to designate the root bridge.

Setting a static priority:

```
Switch(config)# spanning-tree vlan <vlans> priority <priority>
```

Allowing the switch to decide its own priority relative to other switches on the network:

```
Switch(config)# spanning-tree vlan <vlans> root {primary | secondary}
 [diameter <diameter>]
```

Port Cost and ID Tuning

To manually configure the cost for a port (and override the default cost associated with its speed):

```
Switch(config-if)# spanning-tree [vlan <vlans>] cost <cost>
```

The port ID consists of an 4-bit port priority (default value is 128) and a 12-bit port number.

To manually configure the priority of a port:

```
Switch(config-if)# spanning-tree [vlan <vlans>] port-priority <priority>
```

Tuning Spanning-Tree Convergence

Manually configuring STP timers:

```
Switch(config)# spanning-tree [vlan <vlans>] {hello-time | forward-time | max-age}
 <seconds>
```

Timers can also be automatically adjusted when designating an automatically determined bridge priority (via the `diameter` parameter):

```
Switch(config)# spanning-tree vlan <vlans> root {primary | secondary} [diameter
 <diameter> [hello-timer <seconds>]]
```

Redundant Link Convergence

PortFast - Applied to access ports to allow fast establishment of connectivity

UplinkFast - Enables fast failover to an alternate uplink toward root

BackboneFast - Enables fast convergence in the core after a topology change

PortFast

Can be enabled globally:

```
Switch(config)# spanning-tree portfast default
```

Or, per-interface:

```
Switch(config-if)# spanning-tree portfast
```

UplinkFast

Can only be enabled on switches which do not act as transit to root (and are not root).

Enabled globally, for all ports and VLANs:

```
Switch(config)# spanning-tree uplinkfast [max-update-rate <packets per second>]
```

The optional `max-update-rate` (default 150) specifies how fast to flood out spoofed multicast frames from sources in the CAM so that upstream switches see them on the new link.

BackboneFast

BackboneFast allows a switch to respond to inferior BPDUs (BPDUs from a new switch claiming to be root) immediately, instead of waiting for the Max Age timer to expire.

Root Link Query (RLQ) protocol requests are sent out to see if upstream switches have a path to root.

BackboneFast is enabled globally:

```
Switch(config)# spanning-tree backbonefast
```

If enabled, BackboneFast should be enabled on all switches in the domain, due to its reliance on RLQ.

Troubleshooting Spanning-Tree

```
show spanning-tree [detail | summary | ...]
```

Chapter 10: Protecting the Spanning Tree Protocol Topology

Root Guard

If a switch with a lower bridge ID enters the network, it can preempt the current STP root.

Root guard can be enabled on an interface to prevent it from becoming a root port:

```
Switch(config-if)# spanning-tree guard root
```

Root guard will affect all VLANs on the port.

Ports disabled by root guard can be viewed with `show spanning-tree inconsistentports`.

BPDU Guard

BPDU guard automatically places an interface in the error-disabled state upon receipt of a BPDU.

BPDU guard can be enabled globally or per interface:

```
Switch(config)# spanning-tree portfast bpduguard default
Switch(config-if)# [no] spanning-tree bpduguard enable
```

Loop Guard

Loop guard prevents a blocked port from transitioning to the forwarding state if it stops receiving BPDUs. Instead, the port is placed in the `loop-inconsistent` state and continues to block traffic.

Loop guard operates per VLAN, and can be enabled globally or per interface:

```
Switch(config)# spanning-tree loopguard default
Switch(config-if)# [no] spanning-tree guard loop
```

Unidirectional Link Detection (UDLD)

UDLD can detect link failures which do no explicitly shutdown the interface (such as a unidirectional fiber link or failed intermediate media converter).

UDLD transmits frames across a link at regular intervals, expecting the distant end to transmit them back.

The default UDLD message timer is 7 or 15 seconds (depending on the platform), allowing it to detect a unidirectional link before STP has time to transition the interface to forwarding mode.

UDLD has two modes of operation:

Normal mode - UDLD will notice and log a unidirectional link condition, but the interface is allowed to continue operating.

Aggressive mode - UDLD will transmit 8 additional messages (1 per second); if none of these are echoed back the interface is placed in the error-disabled state.

UDLD can be enabled globally for all fiber interfaces, or per-interface:

```
Switch(config)# udld {enable | aggressive | message time <seconds>}
Switch(config-if)# udld {enable | aggressive | disable}
```

The UDLD message time can be from 7 to 90 seconds.

UDLD will not consider a link eligible for disabling until it has seen a neighbor on the interface already. This prevents it from disabling an interface when only one end of the link has been configured to support UDLD.

`udld reset` can be issued in user exec to re-enable interfaces which UDLD has disabled.

BPDU Filtering

BPDU filter can be enabled globally or per-interface to effectively disable STP:

```
Switch(config)# spanning-tree portfast bpdufilter default
Switch(config-if)# spanning-tree bpdufilter {enable | disable}
```

Chapter 11: Advanced Spanning Tree Protocol

Rapid STP (RSTP)

RSTP was developed to provide a faster converging alternative to STP, and is defined in IEEE 802.1w.

Like STP, RSTP can be applied as a single instance or per VLAN.

A root is elected by lowest bridge ID, as in 802.1D STP.

RSTP provides its own set of port roles:

Root port - Same as in 802.1D

Designated port - Same as in 802.1D

Alternate port - A port with an alternate, less desirable path to root

Backup port - A port which provides an alternate, less desirable path to a segment which already has a designated port

RSTP defines port states based on what action is taken on incoming frames:

Discarding - Frames are dropped, no addresses are learned (replaces 802.1D disabled, blocking and listening states)

Learning - Frames are dropped, but addresses are learned

Forwarding - Frames are forwarded

RSTP defines a new version of BPDU (v2) which is backward-compatible with 802.1D.

BPDUs are sent out from every switch at *hello time* intervals; a neighbor is assumed down if three intervals are missed.

If an RSTP switch detects a traditional (version 0) BPDU on a port, that port changes to operate in 802.1D mode.

Port types:

Edge port - A port to which a single host connects; identified by enabling PortFast; loses its edge status upon receipt of a BPDU

Root port - The port with the best path to root; alternates can be identified as well

Point-to-point port - A designated port connected directly to another switch; only full-duplex ports are eligible by default

RSTP Synchronization

All non-edge ports begin in the discarding state.

Proposal messages are used to determine the root port of a segment based on bridge priorities.

When a switch receives a proposal message on a port, it moves all other non-edge ports to the discarding state until it sends an agreement to the sender of the proposal.

When an agreement is reached, the ports on both ends of the link begin forwarding.

This method of proposal/agreement handshakes allows the synchronization process to complete much faster than traditional STP, as no timers are needed.

Topology change BPDUs are sent only when a non-edge port transitions to forwarding.

RSTP Configuration

RSTP is enabled by configuring Rapid PVST:

```
Switch(config)# spanning-tree mode rapid-pvst
```

Half-duplex links to other switches can be administratively designated as point-to-point links:

```
Switch(config-if)# spanning-tree link-type point-to-point
```

Multiple Spanning Tree (MST)

MST was developed to offer a middle ground between CST (one instance for all VLANs) and PVST (one instance for each VLAN).

An MST region is defined by several attributes:

- Configuration name (32 characters)
- Configuration revision number (16-bit)
- Instance-to-VLAN mapping table (up to 4096 entries)

All attributes must match for two switches to belong to the same region.

An MST region is seen as a single virtual bridge by an outside CST, and runs an *Internal Spanning Tree (IST)* inside.

Up to 16 *MST Instances (MSTIs)* numbered 0 through 15 run inside an MST region; MSTI 0 is the IST.

Additional MSTIs can be created and have VLANs assigned to them.

MST Configuration

Enabling MST:

```
Switch(config)# spanning-tree mode mst
```

Creating an MST region:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# name <name>
Switch(config-mst)# revision <revision>
Switch(config-mst)# instance <instance ID> vlan <VLAN list>
```

View pending changes before they are applied:

```
Switch(config-mst)# show pending
```

Chapter 12: Multilayer Switching

Interfaces on multilayer switch are designated as switch ports (layer 2) with `switchport` or routed ports (layer 3) with `no switchport`.

Switched Virtual Interfaces (SVIs) can be defined to provide a routed interface to a VLAN.

Cisco Express Forwarding (CEF)

Traditional multilayer switching ("route once, switch many", also known as *NetFlow switching* or *route cache switching*) was done through the combination of a route processor and a switching engine.

CEF is the second generation of multilayer switching, and is enabled by default in hardware which supports it.

CEF operation relies on two components working in tandem: the layer 3 engine (routing) and the layer 3 forwarding engine (switching).

The layer 3 forwarding engine contains the *Forwarding Information Base (FIB)* and its *Adjacency Table*.

Forwarding Information Base (FIB)

The FIB is an optimized copy of the routing table, with more-specific routes listed first.

Each entry in the FIB has layer 2 and 3 next-hop addressing information associated with it.

FIB entries can be examined with `show ip cef`.

Packets meeting certain conditions cannot be CEF-switched and will be punted to the layer 3 engine for traditional software routing:

- Expired TTL
- MTU exceeded
- ICMP redirect required
- Unsupported encapsulation type
- Compression and/or encryption is necessary
- An ACL log entry must be generated

Accelerated CEF (aCEF) can be implemented in some hardware to cache portions of the FIB on each line card.

Distributed CEF (dCEF) stores the entire FIB on all capable line cards.

Adjacency Table

The adjacency table is the portion of the FIB which contains layer 2 next-hop information (MAC addresses which correspond to the layer 3 next-hop addresses).

Similar to how the FIB is built from the routing table, the adjacency table is built from the ARP table.

Adjacency information can be examined with `show adjacency`.

Adjacency table entries with missing or expired layer 2 addresses are placed in the *CEF glean* state; packets must be punted to the L3 engine so an ARP request/reply can be generated.

When a route is placed in the glean state, incoming packets will be dropped for up to two seconds as the switch awaits an ARP reply.

Other adjacency states include:

- **Null** - Represents a null interface (black hole)
- **Drop** - Indicates packets cannot be forwarded to the destination and should be dropped
- **Discard** - An ACL or other policy mandates that packets be dropped
- **Punt** - Further processing is required by the layer 3 engine

Packet Rewrite

The packet rewrite engine reconstructs the incoming packet with the appropriate next hop address information.

Fields rewritten include:

- Layer 2 destination
- Layer 2 source
- IP TTL
- IP Checksum
- Layer 2 frame checksum

Fallback Bridging

Non-IP protocols are not supported by CEF.

Each SVI carrying nonroutable traffic can be assigned to a *bridge group* and bridged transparently, separate from normal L2 switching.

A special type of STP known as *VLAN-bridge* is run on these bridge groups.

Fallback bridging must be manually configured:

```
Switch(config)# bridge-group <group> protocol vlan-bridge
Switch(config)# interface vlan 10
Switch(config-if)# bridge-group <group>
Switch(config-if)# interface vlan 20
Switch(config-if)# bridge-group <group>
```

Verifying Multilayer Switching

`show interface switchport` ("Disabled" verifies layer 3 operation)

`show ip cef [detail]`

`show bridge group`

Chapter 13: Router, Supervisor, and Power Redundancy

Hot Standby Router Protocol (HSRP)

HSRP is Cisco proprietary, but defined in **RFC 2281**.

HSRP routers multicast to the *all-routers* address `224.0.0.2` on UDP port 1985.

HSRP group numbers (0 - 255) are only significant to an interface.

HSRP group configuration:

```
Switch(config-if)# standby <group> [priority <priority>]
Switch(config-if)# standby <group> ip <address> [secondary]
```

HSRP virtual interfaces are assigned a MAC in the range `0000.0c07.acXX` where the last 8 bits represent the standby group.

Router Election

HSRP priority ranges from 0 to 255; default is 100.

The highest priority wins; highest IP wins a tie.

HSRP interface states:

Disabled

Init

Listen

Speak

Standby

Active

The default hello timer is 3 seconds; holddown timer is 10 seconds.

Timers can be adjusted:

```
Switch(config-if)# standby <group> timers [msec] <hello time> [msec] <hold time>
```

By default a router with higher priority cannot preempt the current active router; this can be allowed:

```
Switch(config-if)# standby <group> preempt [delay [minimum <seconds>] [reload <seconds>]]
```

`minimum` defines the time the router must wait after it becomes HSRP-capable for the interface. `reload` defines the time it must wait after reloading.

Authentication

Cisco devices by default use the plaintext string "cisco" for authentication.

Plaintext or MD5 authentication can be used

```
Switch(config-if)# standby <group> authentication [md5 key-string [0 | 7]] <string>
```

Conceding the Election

A router can be configured to withdraw from active status if one or more of its other interfaces fail:

```
Switch(config-if)# standby <group> track <interface> [<decrement value>]
```

The router's priority will be decremented by the associated value (default 10) if the tracked interface fails.

If another router now has a higher priority and has been configured to preempt, it will take over as the

active router for the group.

Verification

```
show standby [brief] [interface]
```

Virtual Router Redundancy Protocol (VRRP)

Standards-based alternative to HSRP, defined in **RFC 2338**.

VRRP refers to the active router as the *master router*; all others are in the *backup* state.

VRRP virtual interfaces take their MAC from the range `0000.5e00.01XX` where the last eight bits represent the group number.

VRRP advertisements are multicast to `224.0.0.18`, using IP protocol 112.

VRRP advertisements are sent in 1-second intervals by default; backup routers can optionally learn the interval from the master router.

VRRP routers will preempt the master by default if they have a higher priority.

VRRP is unable to track interfaces and concede an election.

VRRP Configuration

VRRP configuration is very similar to HSRP configuration:

```
Switch(config-if)# vrrp <group> priority <priority>
Switch(config-if)# vrrp ip <address> [secondary]
Switch(config-if)# vrrp <group> timers {learn | advertise [msec] <interval>}
Switch(config-if)# [no] vrrp preempt [delay <seconds>]
Switch(config-if)# vrrp authentication <string>
```

Verification

```
show vrrp [brief]
```

Gateway Load Balancing Protocol (GLBP)

GLBP is Cisco proprietary, and acts like HSRP/VRRP with true load-balancing capability: all routers in a group forward traffic simultaneously.

GLBP group numbers range from 0 to 1023. Priorities range from 0 to 255 (default is 100).

IP address(es), router preemption, and hello/hold timers (default 3/10 seconds) can be configured like for HSRP:

```
Switch(config-if)# glbp <group> priority <priority>
Switch(config-if)# glbp ip <address> [secondary]
Switch(config-if)# glbp <group> preempt [delay minimum <seconds>]
```

Timers only need to be configured on the AVG; other routers will learn from it.

Active Virtual Gateway (AVG)

The AVG has the highest priority in the GLBP group (or the highest IP address in the event of a tie); it answers all ARP requests for the group's virtual IP address.

Active Virtual Forwarder (AVF)

All routers sharing load in GLBP are AVFs.

If an AVF fails, the AVG reassigns its virtual MAC to another router.

Two timers are used to age out the virtual MAC of a failed AVF:

Redirect timer (default 600 seconds) - Determines when the AVG will stop responding to ARP requests with the MAC of the failed AVF

Timeout timer (default 4 hours) - Determines when the failed AVF is no longer expected to return, and its virtual MAC will be flushed from the GLBP group

Configuring the timers:

```
Switch(config-if)# glbp <group> timers redirect <redirect> <timeout>
```

AVFs are assigned a maximum weight (1-254; default is 100).

Interfaces can be tracked and the AVF's weight adjusted when interfaces go down:

```
Switch(config)# track <object number> interface <interface> {line-protocol | ip routing}
Switch(config-if)# glbp <group> weighting <maximum> [lower <lower>] [upper <upper>]
Switch(config-if)# glbp <group> weighting track <object number> [decrement <value>]
```

When the `upper` or `lower` threshold is reached, the AVF enters or leaves the group, respectively.

Load Balancing

Up to four virtual MACs can be assigned by the AVG.

Traffic can be distributed among AVFs using one of the following methods:

> **Round robin** (default) - Each new ARP request is answered with the next MAC address available; traffic is distributed evenly among AVFs
>
> **Weighted** - AVFs are assigned load in proportion to their weight
>
> **Host-dependent** - Statically maps a requesting client to a single AVF MAC

Configuring load balancing:

```
Switch(config-if)# glbp <group> load-balancing {round-robin | weighted | host-dependent}
```

Verification

> `show glbp [brief]`

Switch Chassis Redundancy

Redundant supervisor modes:

> **Route Processor Redundancy (RPR)** (> 2 minutes) - The standby supervisor is only partially initialized; when the active sup fails, the standby must reload all modules and finish initializing itself.
>
> **Route Processor Redundancy Plus (RPR+)** (>30 seconds) - The standby supervisor boots but does not operate; when the active sup fails, the standby can take over without reloading the modules.
>
> **Stateful Switchover (SSO)** (>1 second) - Configuration and layer 2 information are stored on both supervisors; the standby sup takes over immediately.

Configuring supervisor redundancy:

```
Switch(config)# redundancy
Switch(config-red)# mode {rpr | rpr-plus | sso}
```

If configuring redundancy for the first time, it must be configured manually on both supervisors.

Redundant operation can be verified with `show redundancy states`.

Non-Stop Forwarding (NSF)

When a standby supervisor takes over, it must populate its RIB; this can be achieved quickly with Cisco's proprietary NSF. NSF-aware neighbors provide routing information to quickly populate the new RIB.

BGP, EIGRP, OSPF, and IS-IS support NSF, but it must be enabled through manual configuration under the relevant protocol:

```
Router(config)# router eigrp <AS>
Router(config-router)# nsf
```

Redundant Power Supplies

Switches with multiple power supplies can operate in one of two power modes:

Combined mode - The load for a single power supply may be exceeded; does not provide redundancy.

Redundant mode (default) - Load is shared but may not exceed the output of a single power supply.

Configuring power mode:

```
Switch(config)# power redundancy-mode {redundant | combined}
```

Power may be administratively removed from or applied to individual modules:

```
Switch(config)# [no] power enable module <slot>
```

Verification:

`show power [redundancy-mode | status | available | used | total]`

`show power inline` - Displays power drawn from PoE interfaces

Chapter 14: IP Telephony

Power Over Ethernet (PoE)

Two solutions exist to supply PoE:

Cisco Inline Power (ILP) - Cisco proprietary solution developed before IEEE 802.3af

IEEE 802.3af - Standard

IEEE 802.3af

An 802.3af PoE switch applies a small voltage across the wire and checks for 25K Ohm resistance to determine if a PoE device is connected.

Depending on the resistance presented at differing test voltages, the switch can determine which power class a device belongs to:

Class 0 - 15.4W (default)

Class 1 - 4.0W

Class 2 - 7.0W

Class 3 - 15.4W

Class 4 - Reserved for future use

The power class determines how much of the switch's power budget is allocated to the interface.

Power is supplied over pairs 1,2 and 3,6 or pairs 4,5 and 7,8.

Cisco ILP

A Cisco ILP switch transmits a 340kHz test tone on the Tx pair to detect a PoE device; if a Cisco ILP-capable device is present, the tone will be echoed back.

Power is supplied over pairs 1,2 and 3,6.

Cisco ILP detects a device's power requirement via CDP.

Configuring PoE

All capable switch ports will attempt PoE by default (`auto`).

```
Switch(config-if)# power inline {auto [max <mW>] | static [max <mW>] | never}
```

PoE can be verified with `show power inline`.

Voice VLANs

Trunks to IP phones are automatically negotiated by *Dynamic Trunking Protocol (DTP)* and CDP.

Configuring a voice VLAN:

```
Switch(config-if)# switchport voice vlan {<VLAN> | dot1p | untagged | none}
```

none (default) - No trunk is formed; voice and data traffic traverse the same access VLAN

vlan - Forms an 802.1Q trunk with designated voice VLAN and native (access) VLAN for data; 802.1p CoS bits in 802.1Q header provide independent QoS

dot1p - Forms an 802.1Q trunk with voice in VLAN 0, data in native VLAN; 802.1p CoS bits used

untagged - Forms an 802.1Q trunk with voice and data both untagged; 802.1p is not used

Voice QoS

QoS models:

Best-Effort Delivery - No QoS

Integrated Services Model - Bandwidth is reserved along a path via Resource Reservation Protocol (RSVP); defined in **RFC 1633**

Differentiated Services Model - QoS is handled dynamically per-hop, based on protocol headers and defined policies

Layer 2 DiffServ QoS

Layer 2 frames transported in a trunk receive a designated *Class of Service (CoS)* value in the trunk header (802.1p bits).

802.1Q native VLAN frames are not tagged and thus are treated with the default CoS.

ISL trunks duplicate the same priority scheme as 802.1p.

Layer 3 DiffServ QoS

The IP *Type of Service (ToS)* header field originally defined a 3-bit *IP precedence* value and a 4-bit ToS value.

The DiffServ QoS model reinterprets this field as a 6-bit *Differentiated Services Control Point (DSCP)*, composed of a 3-bit class selector and a 3-bit drop precedence.

Class 0 - Best effort forwarding

Classes 1-4 - *Assured Forwarding (AF)* with drop preferences

Class 5 - *Expedited Forwarding (EF)*

Classes 6-7 - Network control

Configuring a Trust Boundary

```
Switch(config)# mls qos
Switch(config-if)# mls qos trust {cos | ip-precedence | dscp}
Switch(config-if)# mls qos trust device cisco-phone
Switch(config-if)# switchport priority extend {cos <value> | trust}
```

`mls qos trust device cisco-phone` enables QoS trust only when a Cisco IP phone is detected via CDP.

`switchport priority extend` instructs the phone on how the trust boundary should be extended to a connected PC. The `cos` option overwrites all frames with the given CoS value.

Auto-QoS

Auto-QoS was developed to ease implementation of QoS.

Auto-QoS is a macro which automatically performs the following configurations:

- Enabling QoS
- CoS-to-DSCP mapping
- Ingress and egress queue tuning
- Strict priority queues for egress voice traffic
- Establishing an interface QoS trust boundary

Configuring Auto-QoS on an interface:

```
Switch(config-if)# auto qos voip {cisco-phone | cisco-softphone | trust}
```

Any existing QoS configuration must be completely removed from an interface before Auto-QoS can be applied.

`debug auto qos` can be enabled before applying Auto-QoS to monitor the explicit commands being issued by the macro.

Verifying QoS

- `show mls qos interface <interface>`
- `show interface <interface> switchport`

Chapter 15: Securing Switch Access

Port Security

Port security can be used to restrict which or how many hosts connect to a switch port:

```
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address <MAC>
Switch(config-if)# switchport port-security max <maximum>
Switch(config-if)# switchport port-security violation {protect | restrict | shutdown }
```

Violation actions:

protect - The port continues to function without logging a violation, but frames from violating MAC addresses are dropped.

restrict - As with **protect** mode, frames from violating MAC addresses are dropped, but the violation is logged.

shutdown - The port is transitioned to the error-disabled state, and no traffic is accepted.

IEEE 802.1x

Extensible Authentication Protocol Over LANs (EAPOL) is used to authenticate a connecting host to a switch via layer 2.

Enable AAA and specify a RADIUS server to be referenced for authentication:

```
Switch(config)# aaa new-model
Switch(config)# radius-server host {<hostname> | <IP>} [key <string>]
Switch(config)# aaa authentication dot1x default group radius
```

Enable 802.1x globally:

```
Switch(config)# dot1x system-auth-control
```

Configure authorization per interface:

```
Switch(config-if)# dot1x port-control {force-authorized | force-unauthorized | auto}
```

Port control states:

`force-authorized` (default) - Port is always authorized

`force-unauthorized` - Port will never become authorized

`auto` - Authorization depends on a successful 802.1x authentication

Multiple hosts can be allowed to share a single port with `dot1x host-mode multi-host`.

`show dot1x all` will verify 802.1x operation.

DHCP Snooping

DHCP Snooping prevents DHCP-influenced *Man-in-the-Middle (MITM)* attacks by blocking DHCP replies from untrusted ports.

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan <vlan> [<end vlan>]
```

To designate a port as trusted (DHCP replies are allowed inbound):

```
Switch(config-if)# ip dhcp snooping trust
```

DHCP snooping can also rate-limit DHCP requests on untrusted ports:

```
Switch(config-if)# ip dhcp snooping limit rate <packets per second>
```

A DHCP snooping switch can inject its own MAC and the port on which a request was received in option 82 of the request:

```
Switch(config)# ip dhcp snooping information option
```

`show ip dhcp snooping [binding]` displays DHCP snooping status.

IP Source Guard

IP source guard is used to mitigate IP spoofing, and relies on DHCP snooping bindings to determine the legitimacy of a source address.

To enable IP source guard:

```
Switch(config-if)# ip verify source [port-security]
```

The addition of the `port-security` parameter also enables validation of source MAC addresses for

ports configured with port security.

Static address mappings can be entered for hosts which do not use DHCP:

```
Switch(config)# ip source binding <MAC> vlan <VLAN> <IP> interface <interface>
```

Verification:

`show ip verify source` - Displays the IP source guard status

`show ip source binding` - Displays the IP source guard database

Dynamic ARP Inspection (DAI)

DAI mitigates ARP spoofing attacks (ARP cache poisoning); static ARP entries or the DHCP snooping database must be used for reference.

To enable DAI per VLAN:

```
Switch(config)# ip arp inspection vlan <VLAN range>
```

DAI is only performed on untrusted ports; all ports are untrusted by default.

To configure an interface as trusted:

```
Switch(config-if)# ip arp inspection trust
```

Static ARP entries can be defined in an ARP access list:

```
Switch(config)# arp access-list <name>
Switch(config-acl)# permit ip host <IP> mac host <MAC> [log]
```

To apply the ARP ACL to one or more VLANs:

```
Switch(config)# ip arp inspection filter <ACL name> vlan <VLAN range> [static]
```

The `static` keyword disables checking against the DHCP snooping database (ARP replies will only be allowed from hosts listed in the ACL).

By default DAI only checks the ARP source MAC and IP. DAI can be configured to also inspect the

Ethernet header source and destination MAC, and ARP source IP:

```
Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}
```

`show ip arp inspection` displays DAI status information.

Chapter 16: Securing with VLANs

VLAN Access Lists (VACLs)

VACLs can filter traffic within a VLAN and do not require a routed interface.

A VACL can match traffic from a MAC, IP, or IPX access list.

VACL configuration:

```
Switch(config)# vlan access-map <name> [<sequence number>]
Switch(config-access-map)# match {ip | ipx | mac} address <ACL>
Switch(config-access-map)# action {drop | forward | redirect <interface>}
```

To apply a VACL to a VLAN:

```
Switch(config)# vlan filter <name> vlan-list <VLANs>
```

Private VLANs

Private VLANs (PVLANs) can be implemented to prevent hosts within a VLAN from communicating directly.

Primary (regular) VLANs are associated with *secondary* (private) VLANs.

A secondary VLAN can be one of two types:

Isolated - Hosts associated with the VLAN can only reach the primary VLAN.

Community - Hosts can communicate with the primary VLAN and other hosts within the secondary VLAN, but not with other secondary VLANs.

PVLAN information is not communicated by VTP.

PVLAN ports are configured to operate in one of two modes:

Promiscuous - Port attaches to a router, firewall, etc; can communicate with all hosts

Host - Can only communicate with a promiscuous port, or ports within the same *community* PVLAN

Private VLAN Configuration

Defining a secondary PVLAN:

```
Switch(config)# vlan <number>
Switch(config-vlan)# private-vlan {isolated | community}
```

Defining a primary PVLAN:

```
Switch(config)#vlan <number>
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association <secondary VLANs>
```

Designating a host port:

```
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association <primary VLAN> <secondary VLAN>
```

Designating a promiscuous port:

```
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping <primary VLAN> <secondary VLANs>
```

Host ports are *associated* with one primary and one secondary VLAN, whereas promiscuous ports are *mapped* to one primary and multiple secondary VLANs.

Secondary VLANs can be mapped to an SVI like a promiscuous port, but without the need to specify the primary VLAN:

```
Switch(config)# interface Vlan100
Switch(config-if)# switchport private-vlan mapping <secondary VLANs>
```

Securing VLAN Trunks

Explicitly configure all access ports to protect against trunk spoofing:

```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan <VLAN>
```

VLAN hopping can be mitigated by ensuring an access VLAN is not used as the native VLAN of a trunk.

Chapter 17: Wireless LAN Overview

Frame Transmission

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is used in 802.11 WLANs to avoid collisions.

The *Distributed Coordination Function (DCF)* handles the transmission of frames.

If one station is currently transmitting, a station wishing to transmit must wait for the current station to finish plus the length of the *DCF Inter-frame Space (DIFS)* and a random back-off timer before it may transmit.

Service Sets

Service Set Identifiers (SSIDs) are used to logically group related wireless clients.

Service set types:

Independent Basic Service Set (IBSS) - An ad-hoc network where all clients communicate directly

Basic Service Set (BSS) - Access is centralized on an access point

Extended Service Set (ESS) - An access point bridged the wireless network to a wired network

An SSID can be mapped to a VLAN on an Ethernet network.

Radio Frequency

2.4 GHz band = 2.412 - 2.484 GHz

5 GHz band = 5.150 - 5.825 GHz

Types of interference:

Reflection - Signal is reflected off an object

Refraction - Bending of a signal as it passes through material of varying density

Absorption - Signal strength weakens as it passes through an object

Scattering - A signal is reflected in many different directions

Diffraction - The bending of a signal around an object which partially blocks its path

Fresnel zones - The elliptical sphere of space which must remain clear between two line-of-sight wireless transmitters to prevent diffraction

Measurements of signal strength:

dB - Logarithmic ratio to a reference signal

dBm - Reference to a 1.0 mW signal

dBw - Reference to a 1.0 W signal

Receivers are generally rated in negative dBm, noting their sensitivity.

Antenna gain is expressed in dBi, referenced to a theoretical *isotropic* antenna which propagates a signal evenly in all directions.

Effective Isotropic Radiated Power (EIRP) = Tx power (dBm) + antenna gain (dBi) - cable loss (dB).

WLAN Antenna Categories

Omnidirectional - Distributes signal in a donut-shaped area; low gain

Semi-Directional - Produces an elongated but broad coverage area in one direction, includes Yagi antennas; medium gain

Directional - Directs signal to a single point; high gain

WLAN Standards

Frame Types

Management Frames - Used for service advertisement and membership management

- Beacons
- Client association
- Client authentication

Control Frames - Control traffic flow

- Probe request/response

RTS/CTS messages

Data Frames - Contain data payload

WLAN frames have a 32-byte header and 4-byte trailing checksum.

802.11b

Operates on the 14 channels within the 2.4GHz *Industrial, Scientific, Medical (ISM)* band.

Only channels 1, 6, and 11 are non-overlapping.

Direct Sequence Spread Spectrum (DSSS) modulation allows for varying speeds: 1.0, 2.0, 5.5, and 11.0 Mbps. Higher data rates require stronger signal strength.

DSSS speeds can be mixed among clients within an AP cell, allowing each client to transmit at its fastest potential.

802.11g

Expands upon 802.11b with greater speeds and more complex modulation.

802.11g operates on the same frequencies and channels as 802.11b.

Orthogonal Frequency Division Multiplexing (OFDM) allows for additional speeds of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.

802.11g is backward compatible with 802.11b, but if an 802.11b client joins an 802.11g cell, *all* clients must fall back to 802.11b.

802.11a

Shares the same data rates and modulation techniques as 802.11g, but is not compatible with it or 802.11b.

Operates on the 5 GHz *Unlicensed National Information Infrastructure (U-NII)* band.

The U-NII was divided by the FCC into three smaller bands:

Lower band - 5.5 to 5.25 GHz; indoor use

Middle band - 5.25 - 5.35 GHz; indoor and outdoor use

Upper band - 5.725 - 5.825 GHz; outdoor use

Four non-overlapping channels are offered within each band (12 total).

Other Standards

802.11e - QoS for WLANs

802.11i - Security enhancements

802.11n - Improvements for higher throughput

Chapter 18: Wireless Architecture and Design

Legacy Authentication Types

Open Authentication

No authentication is used; any client can associate to an AP.

Pre-Shared Key (PSK)

A pre-shared static *Wired Equivalence Protocol (WEP)* key authenticates the client to the AP.

Extensible Authentication Protocol (EAP) Types

EAP is an authentication framework originally developed for PPP authentication (**RFC 3748**).

The wireless variants of EAP are defined in **RFC 4017**.

Lightweight EAP (LEAP)

LEAP (also known as Cisco EAP) is a Cisco-proprietary extension to EAP.

Client and AP authentication is performed through a RADIUS server.

Each authenticated client is assigned a unique WEP key.

EAP-TLS

EAP-TLS (defined in **RFC 2716**) uses *Transport Layer Security (TLS)* and relies on digital certificates for authentication.

Every client and AP must have a valid digital certificate to be authenticated.

Each authenticated client is assigned a unique WEP key.

Protected EAP (PEAP)

PEAP is similar to EAP-TLS (it also relied on TLS), but only the authentication server is required to have

a digital certificate; this certificate is used to authenticate the server to clients.

Clients are authenticated using MS-CHAPv2.

EAP Flexible Authentication via Secure Tunneling (EAP-FAST)

EAP-FAST establishes a secure tunnel between the client and the authentication server using a *Protected Access Credential (PAC)*.

The PAC can be assigned from a PAC server or generated dynamically.

Wi-Fi Protected Access (WPA) Types

WPA

First-generation WPA was based on draft 802.11i.

WPA utilizes *Temporal Key Integrity Protocol (TKIP)*; WEP keys are incremented per-packet, and regenerated on reauthentication.

Some form of EAP or a preshared key is used for the initial authentication exchange.

Message Integrity Check (MIC) is used to provide message integrity (hashing).

WPA2

WPA2 was developed from the finalized 802.11i standard.

WPA2 relies on *Advanced Encryption Standard (AES)* for encryption, which requires a hardware upgrade from WEP/WPA.

TKIP is supported for backward-compatibility with WPA.

Proactive Key Caching (PKC) can be used to allow a client to roam between APs without reauthenticating to each.

Cisco Compatible Extensions (CCX)

Cisco developed CCX as a certification process for ensuring compatibility between devices:

- **CCXv1** - Basic 802.11 compatibility, 802.1x for LEAP, multiple SSIDs
- **CCXv2** - WPA, 802.1x for PEAP, fast roaming with CCKM, RF scanning
- **CCXv3** - WPA2, 802.1x for EAP-FAST, Wi-Fi Multimedia QoS (802.11e)
- **CCXv4** - Cisco NAC, VOIP call admission control, VOIP metrics, enhanced roaming, RFID

functionality

Roaming

APs with overlapping coverage areas should be configured to operate on non-overlapping channels (1, 6, and 11 for 802.11b/g).

Vendor-specific roaming algorithms determine when a wireless client will decide to roam.

Clients can scan channels for other APs in two ways:

Passive scanning - A client only listens for beacon frames

Active scanning - A client actively transmits probe requests

A client must disassociate with the current AP before it can associate with another.

WLAN Design

AP Cell Size

A larger cell has the potential to support more clients than is desired.

Effective transmitter power determines the cell size.

Antenna type determines the cell shape.

Channel Layout

Cells should be laid out in a honeycomb fashion, preventing dead space and overlapping channels; this can be accomplished with only three non-overlapping channels.

Chapter 19: Cisco Unified Wireless Network

Autonomous APs can be burdensome to manage in large numbers; a lightweight solution is preferred.

Lightweight Access Points (LAPs) communicate with a centralized *Wireless LAN Controller (WLC)* through *Lightweight Wireless Access Point Protocol (LWAPP)* tunnels.

The division of layer two functions between a LAP and WLC is referred to as a *split-MAC architecture*.

LWAPP tunnels:

Control messages - Encrypted control traffic between the WLC and LAPs

Data - Cleartext data between wireless clients and the WLC

LWAPP traverses UDP ports 12222 and 12223.

WLC Functions

- Dynamic channel assignment
- Transmit power optimization
- Self-healing wireless coverage
- Flexible client roaming
- Dynamic client load balancing
- RF monitoring
- Security management

The Cisco *Wireless Control System (WCS)* is a server application which can be used to administer WLCs.

LAP Operation

Bootstrap process:

1. Obtains an IP address via DHCP
2. Learns IP addresses of available WLCs via DHCP option 43
3. Requests to join the first responsive WLC
4. WLC checks the LAP's code version and optionally upgrades and reboots it
5. LAP and WLC form one secured and one unsecured tunnel for management and client traffic, respectively

Traffic between any two wireless clients connected to an LAP must pass through the WLC.

Roaming

When a client roams between LAPs connected to two WLCs in different subnets, the WLCs perform a mobility exchange and build an Ether-IP tunnel to carry the client's layer 3 data; the client does not use a get a new IP address.

Ether-IP tunnels operate as IP protocol 97, defined in **RFC 3378**.

The original WLC is the *anchor point* and the new WLC is the *foreign agent*.

Mobility Groups

WLCs are arranged in mobility groups to facilitate roaming.

Up to 24 WLCs can belong to a single mobility group.

A client must reassociate and receive a new IP address when roaming to a new mobility group.

WLC Configuration

WLC interfaces:

Management - Static address used for in-band management

AP Manager - Static address on which LWAPP tunnels to the APs are terminated

Virtual - A logical interface used to relay DHCP requests from wireless clients; common to a mobility group

Service port - Out-of-band debugging interface on 4100 and 4400 series WLCs

Distribution system port - Interface facing the wired campus network

Dynamic - Automatically created virtual interface(s) for user VLANs

Initial WLC configuration is done through a CLI wizard.

Pending successful initial configuration, the WLC can be managed through its web interface.

LAP Configuration

LAPs will automatically obtain a code image and configuration at boot time provided they can communicate with a WLC.

LAPs connect to an access switchport (no trunking is required).

LAPs can receive power from an external AC adapter or inline via PoE.

A LAP can be manually configured with an IP address, or it can pull one automatically via DHCP.

WLC addresses can be passed to LAPs via DHCP option 43 (the option payload format varies between models).

The running IOS version determines whether an AP is running in autonomous or lightweight mode; a "JX" suffix denotes lightweight operation.