

CCNP ONT Notes

4 Apr 2008

Chapter 1: Cisco VOIP Implementations

Benefits of packet telephony:

More efficient use of bandwidth

Consolidated network expenses (converged infrastructure)

Improved employee productivity

Access to a variety of communication devices (soft phones, PDAs, etc.)

Packet telephony components:

Phones

Gateways - Interconnect packet- and circuit-switched voice networks

Multipoint Control Units (MCU) - Conference hardware; comprised of a multipoint controller and optional multipoint processor

Application/database servers - TFTP, XML services, etc.

Gatekeepers - Provide call routing (name-address resolution) and Call Admission Control (CAC, permission granting for call setup)

Call agents - Responsible for call routing, address translation, call setup, etc. in a centralized call control model

Video end points

Digital Signal Processor (DSP) - Implementation of voice and/or video codec(s)

Analog interfaces:

Foreign Exchange Office (FXO) - Faces upstream PSTN; acts like an analog phone

Foreign Exchange Station - Faces analog phones; acts like a CO switch

E&M - Used to connect gateways, PBX switches, or CO switches

Phone call stages:

1. **Call setup** - Call routing, CAC, parameter negotiation (IP addresses, UDP ports, codec)
2. **Call maintenance** - Statistics and error collection
3. **Call tear-down** - Notification of call end, frees resources on control devices

Call control:

Distributed - H.323 and *Session Initiation Protocol (SIP)*; all functionality is performed by the end nodes

Centralized - *Media Gateway Control Protocol (MGCP)*; end points rely on centralized call agent(s) for call routing, CAC, etc.

Analog to Digital Conversion

1. **Sampling** - Capturing voice as a *Pulse Amplitude Modulation (PAM)* stream
2. **Quantization** - Assigning numeric value to each sample in a PAM stream
3. **Encoding** - Representation of the quantized values in binary format
4. **Compression** (optional)

The *Nyquist theorem* states that an analog signal must be sampled at at least twice its highest frequency to be accurately reconstructed by the receiving end; a 4KHz voice signal is sampled at 8KHz.

Comparing codec quality:

Mean Opinion Score (MOS) - humans judge quality relative to an in-person conversation on a scale of 1 to 5.

Perceptual Speech Quality Measurement (PSQM) - Automated; 0 = best, 6.5 = worst

Perceptual Analysis Measurement System (PAMS) - Predictive

Perceptual Evaluation of Speech Quality (PESQ) - Predictive

Codecs:

G.711 - Normal PCM; 64Kbps

G.726 - *Adaptive Differential PCM (ADPCM)*; three possible implementations (r32, r24, r16) use 32Kbps, 24Kbps, and 16Kbps respectively by sending only 4, 3, or 2 bits per sample

G.722 - Wideband speech encoding; input signal is split into two sub-bands, each encoded with a modified version of ADPCM; 64Kbps, 56Kbps, or 48Kbps

G.728 - *Low Delay Code Exited Linear Prediction (LDCELP)*; expresses wave shapes of five samples with 10-bit values; 16Kbps

G.729 - *Conjugative Structure Algebraic Code Exited Linear Prediction (CS-ACELP)*; like G.728 but with ten samples; 8Kbps

Digital Signal Processors (DSPs) are processors dedicated to processing voice, and are found in pluggable *Packet Voice DSP Modules (PVDMS)*.

DSP services:

Voice termination

Transcoding (between two different codecs)

Conferencing

Bandwidth Utilization

Overhead: IP (20 bytes) + UDP (8 bytes) + RTP (12 bytes) = 40 bytes

Overhead can be greatly reduced by using *Compressed RTP (cRTP)*, which requires only 2 bytes (4 bytes with checksum).

Because of the processor overhead involved, cRTP should only be used on slow links.

VOIP bandwidth calculation:

1. Determine the codec and packetization period (samples per packet)
2. Determine protocol overhead (cRTP, tunneling, etc)
3. Calculate the packetization size (amount of voice data per packet)
4. Add the lower layer protocol headers to calculate the total frame size (RTP/UDP/IP or cRTP + IPsec, etc)
5. Calculate the packet rate (inverse of packetization period) in packets per second
6. Calculate total bandwidth (#4 multiplied by #5)

Voice Activity Detection (VAD) detects silence on the line and momentarily stops generating data to conserve bandwidth.

Cisco Unified CallManager Functions

Call processing

Dial plan administration

Signaling and device control

Phone feature administration

Directory and XML services

Provides a programming interface to external applications

Survivable Remote Site Telephony (SRST) provides bare VOIP services to branch phones should the connection to a central CallManager be lost

Chapter 2: IP Quality of Service

QoS concerns:

- Available bandwidth
- End-to-end delay
- Jitter (delay variation)
- Packet loss

Common solutions to address bandwidth availability:

- Increase available bandwidth
- Classification and prioritization (QoS)
- Header and/or payload compression
- Increase interface buffers

Implementing QoS

Step 1: Identifying traffic types and requirements

- Perform audits during busy and slow periods
- Determine the business importance of each application
- Define service levels for each traffic class

Step 2: Classifying traffic

- VOIP
- Mission-critical
- VOIP signaling (call setup/tear-down)
- Interactive applications
- Best-effort
- "Scavenger" (unimportant)

Step 3: Defining policies

- Assign minimum and maximum bandwidth for each class
- Assign each class a relative priority
- Assign queuing type

QoS Models

Best-Effort

The best-effort model is simple the absence of QoS policy.

Integrated Services (IntServ)

Resource Reservation Protocol (RSVP) is used to reserve a minimum amount of bandwidth along an end-to-end path.

Provides explicit end-to-end admission control per request (flow).

Substantial overhead is involved; poor scalability.

Differentiated Services (DiffServ)

DiffServ is defined in RFCs [2474](#) and [2475](#).

QoS is configured and performed separately at each hop in the path.

Traffic is administratively grouped into classes with different qualities of service.

The DiffServ model sacrifices end-to-end service guarantee in favor of scalability.

QoS Implementation

Legacy CLI

Non-modular, tedious configuration at the interface level.

Modular QoS CLI (MQC)

MQC provides a structured framework for defining classes and policies.

1. Traffic classes are defined with the `class-map` command
2. QoS policies are linked to traffic classes with the `policy-map` command
3. Policies are applied to interfaces with the `service-policy` command

`show class-map` and `show policy-map` can be used to verify MQC configurations.

AutoQoS

AutoQoS facilitates the automatic generation and application of QoS policies.

AutoQoS Discovery can perform automatic classification using NBAR and CDP.

Perceived bandwidth must be configured accurately on interfaces with the bandwidth statement.

First generation AutoQoS is configured with `auto qos voip` on an interface, only automating QoS configuration for VOIP traffic.

Modern (Enterprise) AutoQoS is configured with `auto discovery qos` to enable NBAR traffic analysis and `auto qos` for policy construction.

SDM QoS Wizard

The SDM Wizard is a GUI frontend for QoS configuration using three built-in classes (VOIP, business-critical, and best-effort).

Allows for periodic monitoring of QoS performance.

Chapter 3: Classification, Marking, and NBAR

Marking should be performed as close to the source as possible.

Layer 2 Class of Service (CoS)

Ethernet 802.1Q/p

CoS is implemented in the 3-bit PRI field of the 802.1Q header.

Binary Value	Name	Application
000	Routine	Best-effort
001	Priority	Medium priority data
010	Immediate	High priority data
011	Flash	Call signaling
100	Flash override	Video conferencing
101	Critical	Voice bearer
110	Internet	Internet network control
111	Network	Network control

DE (Frame Relay) and CLP (ATM)

1-bit *Discard Eligibility (DE)* and *Cell Loss Priority (CLP)* flags determine whether the frame/cell is a candidate for being dropped in the event of congestion.

MPLS EXP

The MPLS EXP field is 3 bits wide, compatible with the IP Precedence/DSCP field.

EXP can be automatically copied from IP's Precedence/DSCP or administratively configured.

IP DSCP

The original IP specification (**RFC 791**) used only a 3-bit precedence value in the 8-bit *Type of Service (ToS)* field.

Modern IP QoS examines the ToS field as a 6-bit *Differentiated Services Code Point (DSCP)*; the remaining two bits are used for *Explicit Congestion Notification (ECN)*.

DSCP is backward-compatible with IP precedence, but more granular.

A *per-hop behavior (PHB)* is the QoS action taken at one node in a path.

PHB types:

Class selector - 3 least significant DSCP bits are set to 0; equivalent to IP precedence/ToS

Default - 3 most significant bits set to 0; best-effort (no QoS)

Assured Forwarding (AF) - 3 most significant bits set to 001, 010, 011, or 100; AF1 through AF4 used for guaranteed bandwidth

Expedited Forwarding (EF) - 5 most significant bits set to 10111 (decimal 46); best unreserved class of service, used to provide minimal delay

Each of the four AF classes are broken into three groups: low (010), medium (100), and high (110) drop preference.

Lower AF drop preference provides better quality of service within each AF class.

Trust Boundaries

Trust boundaries are formed to determine where QoS markings should be evaluated (trusted). This prevents a user from inadvertently or maliciously marking his own traffic as more favorable.

The trust boundary can be established at an end system (such as an IP phone), access switch, or distribution switch.

Network Based Application Recognition (NBAR)

NBAR tasks:

Protocol discovery

Traffic statistics collection

Traffic classification

NBAR limitations:

Requires CEF

Won't function on an etherchannel

Max of 24 simultaneous hosts/URLs/MIME types

Analyzes only the first 400 bytes of a packet

NBAR identifies upper-layer protocols using expandable *Packet Description Language Modules (PLDMs)*.

Configuring NBAR

Enable NBAR on an interface:

```
Router(config-if)# ip nbar protocol-discovery
```

Add a PDLM:

```
Router(config)# ip nbar pdlm <location>
```

Modify protocol:port assignment:

```
Router(config)# ip nbar port-map <protocol> [tcp | udp] <port>
```

Use in QoS:

```
Router(config)# class-map foo  
Router(config-cmap)# match protocol <protocol> <...>
```

Verification:

```
show ip nbar protocol-discovery
```

```
show ip nbar port-map
```

Chapter 4: Congestion Management and Queuing

The default queuing method on an interface faster than 2.048 Mbps is *First In, First Out (FIFO)*. Interfaces operating at 2.048 Mbps or slower perform *Weighted Fair Queuing (WFQ)*.

Each physical interface has hardware and software queuing mechanisms; software queues are only used when the hardware queue is congested.

Tail-drop occurs when all queues are full and a packet is dropped.

Hardware queue sizes can be configured with `tx-ring-limit` and verified with `show controllers <interface>`.

Simple Queuing

First-In-First-Out (FIFO)

Packets are transmitted in the order they are received with no preference (no QoS).

FIFO is the default mechanism for interfaces >2.048Mbps.

Priority Queuing (PQ)

PQ provides four queues: high, medium, normal, and low.

All packets in a higher priority queue will be processed before any packets in a lower priority queue.

Lower priority queues can be starved if higher priority queues consume all available bandwidth.

PQ is implemented by defining and applying priority lists:

```
Router(config)# priority-list 1 {interface | protocol} ...  
    {high | medium | normal | low}
```

Round Robin (RR)

All queues are equal priority; one packet is taken from each queue per cycle.

Round robin does not provide for traffic prioritization, and queues with larger packets will consume more bandwidth than queues with smaller packets.

Weighted Round Robin (WRR)

WRR is a modification to RR which allows for disproportionate allowance of bandwidth to queues.

Custom Queuing (CQ) is an example of WRR; it specifies a certain number of bytes to be processed from each queue.

Weighted Fair Queuing

WFQ is the default mechanism on and only supported on interfaces less than or equal to 2.048 Mbps.

WFQ queues are created per flow and are not configurable.

Each flow is assigned to a dynamic FIFO queue by source/destination IP address, protocol number, ToS value, or source/destination port number.

The maximum number of dynamic queues is configurable between 16 - 4096 (256 by default).

Packets are dropped from aggressive flows more frequently than from less aggressive flows.

The *hold queue* is the sum of all memory available to the WFQ system; all packets are *aggressively dropped* while the hold queue is full.

Each queue has a *Congestive Discard Threshold (CDT)* which allows for *early dropping* of packets before the queue is completely full.

WFQ can be disabled on an interface with `no fair-queue` (queuing is switched to FIFO).

Queue information can be viewed with `show interface` or `show queue <interface>`.

Class-Based Weighted Fair Queuing (CBWFQ)

CBWFQ is similar to WFQ but with user-defined queue classes instead of dynamically created flow-based queues.

CBWFQ supports a maximum of 64 queues.

Each queue is allotted a certain amount or percentage of the available bandwidth.

The default queue named `class-default` is always present and will match all traffic not matched by other queues.

Bandwidth can be allocated in Kbps, percentage, or remaining percentage. All classes within a policy map must use the same unit of measure (Kbps or percentage).

The default maximum reserved bandwidth is 75%; this can be modified with `max-reserved-bandwidth` (applied to the interface).

Fair queuing (instead of FIFO) can be enabled for the default class with `fair-queue` followed by the maximum number of dynamic queues.

The queue size for each class can be adjusted with `queue-limit`.

Configuration example:

```
policy-map Foo
  class Critical_Apps
    bandwidth percent 20
    queue-limit 50
  class Normal_Apps
    bandwidth remaining recent 50
  class class-default
    fair-queue 32
```

Low Latency Queuing (LLQ)

LLQ implements a strict-priority queue which is favored over all other queues.

LLQ is typically used for delay-sensitive traffic like VOIP.

The priority queue is policed to a certain bandwidth to prevent starvation of other queues.

Priority queues are created under a class with `priority <bandwidth>` or `priority percent <percentage>`.

Configuration example:

```
policy-map Foo
  class VOIP
    priority 128
  class Important_Stuff
    bandwidth 512
    queue-limit 100
  class class-default
    fair-queue 32
```

`show policy-map [interface]` can be used to inspect policy maps.

Chapter 5: Congestion Avoidance, Policing, Shaping, and Link Efficiency Mechanisms

Congestion avoidance is implemented to avoid *tail drop*, which occurs when there is no room left in a queue for incoming packets.

Tail drop is not selective; less aggressive flows are not preferred to aggressive flows, thus no QoS can be provided.

TCP global synchronization occurs when tail dropping of packets forces flows to cycle between small and large windows.

TCP starvation occurs when stateless protocols like UDP fill available queue space before the throttled TCP flows.

Random Early Detection (RED)

When RED is implemented, packets are randomly dropped before the queue becomes full.

The rate of drop increases as the queues nears its maximum size.

RED mitigates the problem of TCP synchronization.

Configuration parameters:

Minimum threshold - Below this no packets are dropped

Maximum threshold - Above this all packets are dropped

Mark Probability Denominator (MPD) - An integer specifying the base probability of drop

Weighted Random Early Detection (WRED)

WRED is RED with the added capability of favoring prioritized traffic, based on the IP precedence or DSCP.

Class-Based WRED (CBWRED)

CBWRED is WRED implemented inside a CBWFQ system.

CBWRED is applied to a CBWFQ class (under a policy map) with random-detect.

CBWRED operates on IP precedence by default but can be configured to evaluate DSCP.

Each precedence/DSCP value can be configured with a unique MPD and minimum and maximum thresholds.

Configuration example:

```
policy-map Foo
  class Precedence_Based_WRED
```

```
bandwidth 100
random-detect
class DSCP_Based_WRED
bandwidth 100
random-detect dscp-based
```

Traffic Shaping and Policing

Policing

Policing restricts the amount of bandwidth consumed by traffic.

Traffic which exceeds the policed threshold can be dropped or remarked to a lower QoS.

Purposes:

- Enforcing subrate access; limiting available bandwidth to less than that of the physical interface

- To limit the traffic rate per class

- To remark traffic not conforming to an SLA

Policing can be applied inbound or outbound on an interface.

Shaping

Shaping buffers excess traffic for transmission, introducing a delay.

Purposes:

- To slow the rate at which traffic is sent to a congested destination

- To comply with a subscribed rate (bandwidth cap)

- To transmit traffic from different classes at different rates

Shaping can only be applied outbound.

Shaping introduces variable delay when traffic is buffered.

Shaping can be configured to respond to network conditions and signals, such as frame relay *Backward Explicit Congestion Notifications (BECNs)*.

Link Efficiency Mechanisms

Most link efficiency mechanisms are only required or supported on slow links.

Layer 2 Payload Compression

Layer 2 payload compression is implemented on a link-by-link basis, and compresses the entire layer 2 payload.

Compression introduces a processing delay, but reduces serialization delay and increases available bandwidth.

Compression can be performed in hardware or software; compression performed in software is CPU-intensive and not recommended.

Header Compression

Header compression can be used with TCP or RTP. Only headers are compressed, not payload.

Like L2 payload compression, header compression is implemented on a link-by-link basis.

Link Fragmentation and Interleaving (LFI)

Large frames are fragmented and interleaved with smaller, high-priority frames to reduce jitter.

Chapter 6: Implementing QoS Pre-Classify and Deploying End-to-End QoS

QoS Pre-Classify

By default, when an IP packet is encapsulated into a tunnel, the IP ToS field is copied from the original header to the new one.

QoS preclassification is needed when other aspects (such as source and destination address or port) must be evaluated for the application of a QoS policy.

Preclassification creates a copy of the original (inner) packet header for the egress interface to reference when QoS is performed on the encapsulated (outer) packet header.

A service policy applied to a physical interface affects all tunnels originating from that interface.

qos pre-classify is applied to the virtual interface and/or crypto map:

```
interface Serial0
  ip address 10.0.0.1 255.255.255.252
  service-policy WAN
```

```
!  
interface Tunnel0  
  ip address 192.168.0.1 255.255.255.252  
  tunnel source serial0  
  tunnel destination 10.0.0.2  
  crypto map VPN  
  qos pre-classify  
!  
crypto map VPN 10 ipsec-isakmp  
  ...  
  qos pre-classify
```

Deploying End-to-End QoS

Guidelines for implementing QoS:

- Classify and mark traffic as close to the source as possible
- Police traffic as close to the source as possible
- Establish trust boundaries
- Classify real-time traffic as high-priority
- Use multiple queues on transmit interfaces
- Prefer hardware-based QoS to software-based

Control Plane Policing (CoPP)

CoPP protects the control plane of a router or switch from excessive traffic.

Configuring CoPP:

- Define packet classification criteria (`class-map`)
- Define a service policy (`policy-map`)
- Apply the service policy to the control plane (`service-policy`)

Configuration example limiting telnet traffic:

```
class-map Telnet  
  match access-group 100  
!  
policy-map Telnet_Access  
  class Telnet
```

```
    police 8000 conform transmit exceed drop
!
control-plane
service-policy input Telnet_Access
!
access-list 100 permit tcp any any eq telnet
```

Chapter 7: Implementing AutoQoS

AutoQoS VOIP:

- First generation of AutoQoS
- Available on routers and switches
- Relies on NBAR for classification and marking
- Configures QoS for VOIP traffic only

AutoQoS Enterprise:

- Second generation, introduced in IOS 12.3(7)T
- Available only on routers
- Two deployment stages: traffic discovery via NBAR, and policy implementation

AutoQoS interface requirements:

- CEF must be enabled for the interface
- No service policy can already be applied
- Bandwidth must be accurately configured

Deploying AutoQoS Enterprise on Routers

The default AutoQoS discovery period is three days, but this can be modified.

AutoQoS discovery is enabled with `auto discovery qos [trust]` on an interface.

Discovery results (even unfinished) can be viewed with `show auto discovery qos`.

After the discovery phase has completed, AutoQoS is enabled per interface:

```
Router(config-if)# auto qos [voip [trust] [fr-atm]]
```

The `voip` keyword forces legacy AutoQoS (VOIP only).

Verification:

```
show auto qos - Displays the auto-generated AutoQoS class and policy maps
```

```
show policy-map interface - Displays applied policy map and QoS parameters for each interface
```

Deploying AutoQoS VOIP on Switches

To configure a port as trusted only when a trusted device is detected, such as a Cisco IP phone (requires CDPv2):

```
Switch(config-if)# auto qos voip cisco-phone
```

To enable a permanently trusted interface (for example, a trunk or uplink):

```
Switch(config-if)# auto qos voip trust
```

The default CoS-to-DSCP mappings can be modified with `mls qos map`.

Verification:

```
show auto qos - Displays the auto-generated AutoQoS configuration
```

```
show mls qos interface <interface> - Displays QoS parameters for an interface
```

```
show mls qos maps - Displays the CoS-to-DSCP mappings used by AutoQoS
```

Common AutoQoS Issues

Too many classes are created

The configuration generated by AutoQoS doesn't automatically adjust to changing network conditions

Even with auto discovery, AutoQoS may not fit some scenarios

Chapter 8: Wireless LAN QoS Implementation

Wireless LANs use *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* as the MAC mechanism.

Collision avoidance is performed by *Distributed Coordinated Function (DCF)*, which employs *Inter-Frame Spacing (IFS)* and random back-off windows to minimize collisions.

Wireless LAN QoS

WLAN QoS is defined in IEEE 802.11e.

Wireless Multimedia (WMM) was released prior to 802.11e as an interim standard.

WMM provides four access categories, or queues:

Platinum - Voice

Gold - Video

Silver - Best effort (default)

Bronze - Background

802.11e provides eight priority levels, 0 through 7.

802.11e priorities can be mapped to WMM access categories for backward compatibility:

WMM 802.11e

Platinum 6 and 7

Gold 4 and 5

Silver 0 and 3

Bronze 1 and 2

802.11e and WMM use *Enhanced DCF (EDCF)* to provide proportional back-off window sizes for each class.

Split-MAC Architecture

The split-MAC architecture separates MAC services to real-time and non-real-time functions.

Real-time functions are performed by *Lightweight Access Points (LAPs)*:

Beacon generation

Probe transmission/response

Power management

802.11e/WMM QoS

Encryption/decryption

Control frame processing

Packet buffering

Non-real-time functions are handled by a centralized *Wireless LAN Controller (WLC)*:

- Client association/disassociation
- 802.11e/WMM resource reservation
- 802.1x EAP
- Key management

Lightweight Access Point Protocol (LWAPP) provides tunneling between LAPs and a WLC.

802.11e/WMM QoS values are translated to DSCP values on the LWAPP packet header to ensure end-to-end QoS.

Chapter 9: 802.1x and Configuring Encryption and Authentication

Wireless Security

Wired Equivalent Privacy (WEP) was the first implementation of wireless encryption, and has several drawbacks:

- Weak encryption (proven to be easily broken)
- Vulnerable to dictionary attacks
- Does not offer protection against rogue access points
- Keys must be manually distributed

Cisco developed *Lightweight Extensible Authentication Protocol (LEAP)* to extend WEP.

LEAP provides several benefits:

- Server-based authentication using 802.1x
- Dynamic keys
- Mutual client and server authentication
- Replay attack protection

Wi-Fi Protected Access (WPA) was developed by the Wi-Fi Alliance Group as an interim non-proprietary solution to replace WEP.

IEEE 802.11i (also known as WPA2) was released after WPA, but required a hardware upgrade to implement the stronger AES encryption.

IEEE 802.1x

802.1x provides port-based network access control.

802.1x is used in conjunction with *Extensible Authentication Protocol (EAP)* to secure wireless LANs.

EAP Authentication Protocols

Cisco LEAP

Provides fast and secure roaming and single sign-on.

EAP-FAST

EAP Flexible Authentication via Secure Tunneling (EAP-FAST) is nonproprietary.

EAP-FAST does not require certificates.

EAP-FAST consists of three phases:

Phase 0 (optional) - Client is dynamically provisioned with a *Protected Access Credential (PAC)*

Phase 1 - Client establishes a secure tunnel with the AAA server using PAC

Phase 2 - Client authentication

EAP-TLS

EAP Transport Layer Security (EAP-TLS) uses TLS and PKI.

Clients and servers must have certificates to be authenticated.

PEAP

Protected EAP (PEAP) only requires the authentication server to have a certificate.

PEAP has two phases:

Phase 1 - The server is authenticated and an encrypted tunnel is formed

Phase 2 - Client authentication

Client authentication can be performed using *Generic Token Card (GTC)* (called PEAP-GTC) or *Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) version 2 (PEAP-MSCHAPv2)*.

	Cisco LEAP	EAP-FAST	EAP-TLS	PEAP-GTC	PEAP-MSCHAPv2
Active Directory auth	Yes	Yes	Yes	Yes	Yes
LDAP auth	No	Yes	Yes	Yes	No
OTP auth	No	No	Yes	Yes	No
Novell NDS auth	No	No	Yes	Yes	No
Requires server cert	No	No	Yes	Yes	Yes
Requires client cert	No	No	Yes	No	No
Windows single sign-on?	Yes	Yes	Yes	No	Yes
Fast secure roaming?	Yes	Yes	No	No	No
WPA/WPA2	Yes	Yes	Yes	Yes	Yes

WPA

WPA performs authentication using either 802.1x/EAP or with preshared keys.

First-generation WPA uses *Temporal Key Integrity Protocol (TKIP)*, which is based on the same RC4 encryption used by WEP, and *Message Integrity Code (MIC)*.

IEEE 802.11i (also known as WPA2) was released shortly after WPA.

WPA2 uses CCMP to implement AES encryption; old WPA hardware typically cannot support the stronger AES encryption, requiring a hardware upgrade.

WPA/WPA2 provide two modes of operation:

Personal mode - Authentication is performed using preshared keys

Enterprise mode - 802.1x/EAP is used for authentication

Chapter 10: WLAN Management

Cisco Unified Wireless Networks

Five core elements:

Client devices

Mobility platform - Lightweight access points (LWAPs)

Network unification - Wireless LAN Controllers (WLCs)

Word-class network management - Wireless Control System (WCS)

Unified advanced services - Yet another ambiguous buzzword conjured by marketing people for the sake of confusing honest network engineers

LWAPs include the 1500, 1300, 1240AG, 1230AG, 1130AG, and 1000 models.

WLCs include the 4400 and 2000 models, as well as the Catalyst 6500 Wireless Services Module (WSM) and ISR and Catalyst 3750 integration.

WLAN Implementation

Wireless LANs can be implemented with either autonomous or lightweight access points:

Autonomous APs - Each AP is independently configured and monitored

Lightweight APs - Configuration and monitoring is centralized on a WLC

A *Wireless LAN Solution Engine (WLSE)* and *Wireless Domain Services (WDS)* server can be used to provide centralized management of autonomous APs.

WLAN components comparison:

	Autonomous solution	Lightweight solution
Access points	Autonomous	LWAPs
Control	WDS	WLC
Management	WLSE	WCS

Management Solutions

WLSE

Two versions:

CiscoWorks WLSE - Supports up to 2500 WLAN devices

WLSE Express - Supports up to 100 WLAN devices

WCS

The WCS supports up to 50 WLCs and 1500 APs.

Three versions:

WCS Base

WCS Location - Adds RF fingerprinting technology

WCS Location + 2700 Series Wireless Location Appliance - Tracks devices in real-time

