- Introduction to 802.11 Wireless -

802.11 Overview

In the mid 1990's, the IEEE LAN/MAN committee began developing a series of **Wireless Local Area Network (WLAN)** standards. Collectively, these wireless standards are identified as the **802.11 standard.**

Note: The 802.11 standard is occasionally referred to as **Wi-Fi**, though the term 'Wi-Fi' has been applied to other wireless standards as well.

Various **amendments** have been made to the 802.11 standard. These are identified by the letter appended to the standard, such as 802.11*a* or 802.11*g*. The 802.11 amendments will be covered in greater detail later in this guide.

Wireless devices communicate across a specific range of **RF frequencies** known as a **channel**, using an antenna off of a **radio card**. 802.11 antennas come in several forms:
- Omnidirectional
- Semi-directional
- Highly-directional

A group of communicating 802.11 wireless devices is known as a **service set.** A wireless client can connect point-to-point with another wireless client – this is referred to as an **ad-hoc** connection, or an **Independent Basic Service Set (IBSS).**

More commonly, wireless client are centrally connected via a **wireless access point (WAP).** This is referred to as an **infrastructure** connection, or a **Basic Service Set (BSS)**. Wireless clients must **associate** with a WAP before data can be forwarded. WAPs often serve as a *gateway* between the wired and wireless networks.

In environments where a single WAP does not provide sufficient coverage, multiple WAPs can be *linked* as part of an **Extended Service Set (ESS)**.

Radio Frequency Overview

Wireless communication is accomplished using **Radio Frequency (RF)** waves. **Frequency** is a measurement of the number of cycles completed per a given time period for an electromagnetic wave. The standard frequency measurement unit is the **hertz (Hz),** or one *cycle* per *second*.

Note: Ranges of frequencies are often identified by their specific use; these ranges are often referred to as **bands.**

Transmitting devices tune the signal to a specific frequency; receiving devices must tune to this frequency to receive the transmission. A signal at a specific frequency is referred to as the **carrier signal**. However, a carrier signal alone cannot contain data.

Modulation is the method of altering a signal to convey a message or data stream, usually by varying its **amplitude, frequency,** or **phase.**

Voltage   Amplitude Modulation

Time

Voltage   Frequency Modulation

Time
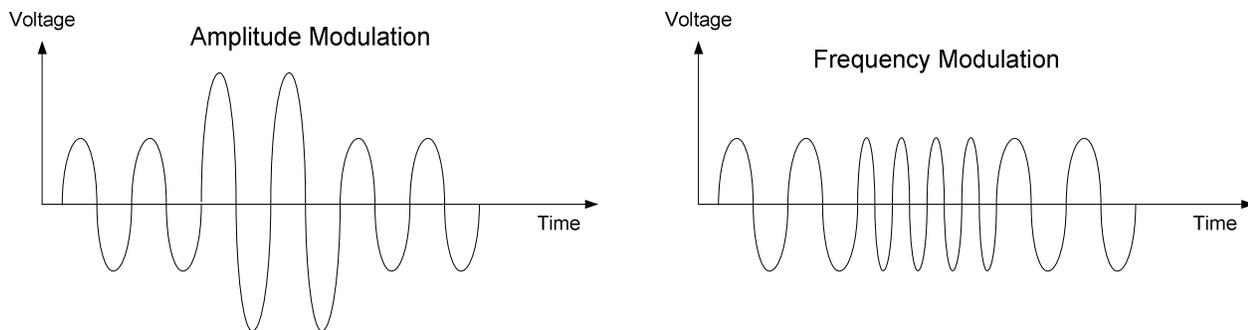
802.11 devices employ multiple advanced modulation techniques, depending on the 802.11 amendment. This modulation requires that 802.11 devices communicate on a small subset of frequencies (referred to as a **channel)** varying around the carrier signal.

Each 802.11 amendment operates in either the **2.4-GHz** or **5-GHz** band:
* The **2.4-GHz band** provides the greatest range, but is *unregulated* and shared with appliances like microwaves and cordless phones. This can result in interference and degraded performance. The 2.4-GHz band is a subset of the industrial, scientific, and medical (ISM) band.
* The **5-GHz band** is *regulated* and thus generally free of interference. However, signals at this frequency suffer from poor range and are easily obstructed by intermediary objects. The 5-GHz band is referred to as the Unlicensed National Information Infrastructure (UNII) band.

RF Signal Strength

RF signals will attenuate in the open air. The power output of the RF antenna dictates the signal strength, and the usable distance of the signal.

RF power output is not usually measured in absolute terms (such as Watts). Instead, it is measured in **decibels (dB),** as a *ratio* of power to a *reference point*. The reference point is usually one Watt (W) or one milliWatt (mW).

The resulting power measurements are **Decibel Watts (dBw)** and **Decibel milliWatts (dBm)**. One milliWatt of power output is represented as **0 dBm.**

Decibel measurements are *logarithmic* in nature. The formula for calculating power output in decibel form is as follows:

$$dB = 10\log_{10}(P_{signal} / P_{reference})$$

The abbreviation *P* is short for power. Using the above formula, a signal transmitting at an *absolute* power of 20 mW would be represented as having a *relative* output power of 13 dBm.

Because decibels measure a *ratio* of power, it is possible to have a *negative* value. A negative value indicates that the amount of power is *less* than the reference point. For example, .25 mW of absolute power would be represented as -6 dBm. Conversely, a positive value indicates that the amount of power is *more* than the reference point.

(Reference: CCNP BCMSN Official Exam Certification Guide 4th Edition. David Hucaby. Pages 452-457; http://en.wikipedia.org/wiki/Decibel)

* * *

RF Interference and Obstruction

In addition to open-air attenuation, RF signals are susceptible to **interference**, degrading the performance and integrity of the communication. As stated previously, other devices operating in the **same frequency range** can interfere with a signal.

Physical objects can also obstruct or alter the trajectory of a RF signal:

- **Reflection –** occurs when a signal *bounces off* of a reflective material, altering its intended trajectory (and sometimes back towards the sender). Metal objects and water often cause reflection. If the signal is reflected in multiple directions, it is referred to as **scattering.**

- **Refraction –** occurs when the trajectory of a signal is *bent* as it *passes through* an object, such as a wall.

- **Absorption –** occurs when the energy of a signal is *absorbed* as it passes through an object, such as a wall or a tree. This loss of energy degrades the strength of the signal.

- **Diffraction –** occurs when a signal *bends around* a signal-absorbing object. For example, a sufficiently-strong signal can bend around an obstructing building, or around a corner within a building. However, this weakens and negatively affects the trajectory of a point-to-point signal.

Because of diffraction, it is particularly important to maintain **line-of-sight** when employing a point-to-point wireless signal over a long distance. Buildings, trees, and even the curvature of the earth can obstruct the line-of-sight of the transmitting/receiving antennas.

RF Fresnel Zones

Specifically, line-of-sight must be maintained within a signal's elliptical-shaped **Fresnel zone**.
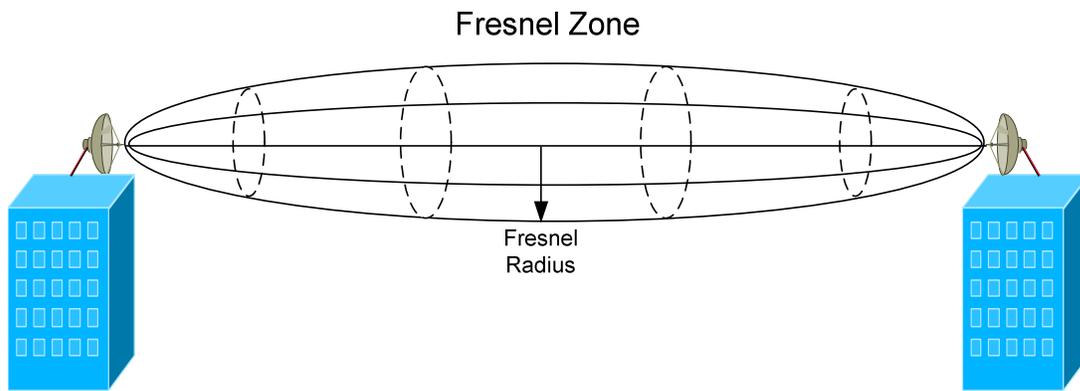
Fresnel Zone

If more than 40% of the lower radius of the Fresnel zone is obstructed, the signal will be negatively impacted from diffraction. Thus, it is imperative to maintain a *minimum* of 60% clearance in this radius.

Calculating the radius of a Fresnel zone requires a complex formula, which is beyond the scope of this guide. Various factors must be accounted for, such as atmospheric refraction, the curvature of the earth, frequency, and the relative heights of the two antennas.

A free Fresnel zone calculator is available online at:

http://www.afar.net/fresnel-zone-calculator/

(Reference: CCNP BCMSN Official Exam Certification Guide 4th Edition. David Hucaby. Pages 450-452)

802.11 Channels

Recall that all amendments to the 802.11 standard operate in one of two frequency bands:

- **2.4-GHz band** (specifically, 2.4000 to 2.4835GHz)
- **5.0-GHz band** (specifically, 5.150 to 5.825GHz)

The 2.4 GHz band supports a **total of 14 channels**, though the FCC limits this to **11 channels** in the United States. The *center* frequency of each channel is separated by only **5 MHz.**

Channel #	Center Frequency	Channel #	Center Frequency
1	2.412 MHz	8	2.447 MHz
2	2.417 MHz	9	2.452 MHz
3	2.422 MHz	10	2.457 MHz
4	2.427 MHz	11	2.462 MHz
5	2.432 MHz	12*	2.467 MHz
6	2.437 MHz	13*	2.472 MHz
7	2.442 MHz	14*	2.484 MHz

* Restricted in US

The 802.11 amendments that use the 2.4-GHz band (specifically, 802.11b and 802.11g) require a **22 MHz range** to modulate the signal. Thus, with each channel's center frequency separated by only 5 MHz, **channel overlap** will occur.

In fact, the 2.4-GHz band supports only **three non-overlapping channels**. Specifically, these are channels **1, 6,** and **11.** Devices competing on the same or adjacent channels will interfere with each other, degrading performance and reliability.

The less-often used 5-GHz band supports up to **12 non-overlapping channels** (in the U.S.), and is further separated into three sub-bands (with four channels each). The lower and middle bands are dedicated for *indoor* use, and the higher band is dedicated for *outdoor* use.

Remember that the 2.4-GHz band is *unregulated*, and the 5.0-GHz band is *regulated*.

(Reference: http://en.wikipedia.org/wiki/List_of_WLAN_channels;
http://www.cisco.com/en/US/docs/wireless/access_point/1200/vxworks/configuration/guide/bkscgaxa.html)

Preventing Channel Overlap

In large environments, a single WAP is often insufficient for full wireless coverage. Multiple WAPs can be linked together as part of an **Extended Service Set (ESS)**.

However, special considerations must be made when installing WAPs in close proximity to each other. Recall that only a limited number of non-overlapping channels are available in both the 2.4-GHz and 5.0-GHz bands.
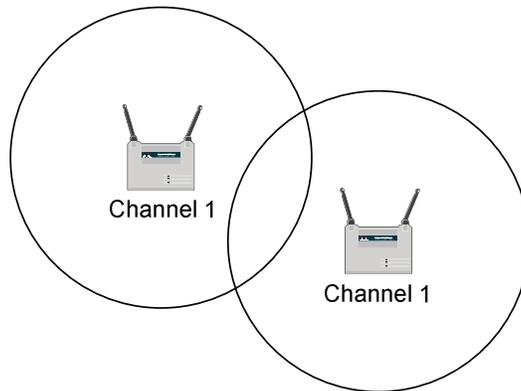
Adjacent WAPs should *never* be configured on the same channel; the overlapping wireless fields will interfere with each other and *severely* degrade performance.
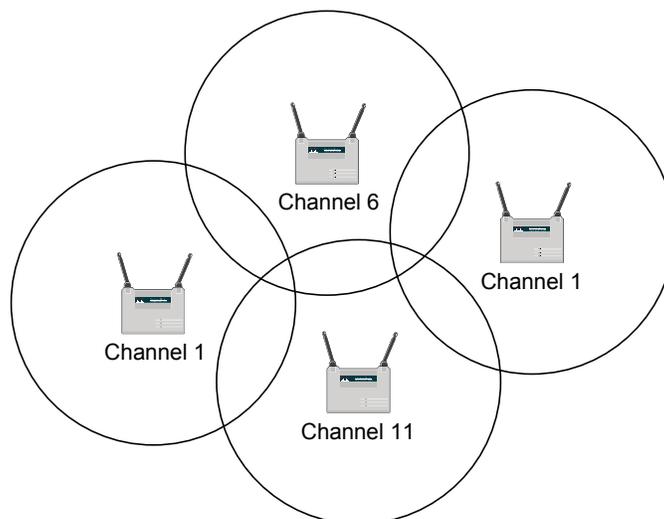
Providing full wireless coverage while preventing channel overlap can be challenging, especially if the environment has multiple floors. Performing a comprehensive **wireless site survey** is helpful in mapping out an accurate solution.

802.11 and Collisions

If two devices on a half-duplex Ethernet (802.3) network send a packet simultaneously, a **collision** will occur. Similarly, if two 802.11 wireless devices transmit simultaneously, their signals will mix resulting in unusable noise (essentially a *wireless collision*).

Half-duplex 802.3 Ethernet uses **Carrier Sense Multiple Access with Collision Detect (CSMA/CD)** to control media access. Devices monitor the physical link, and will only transmit a frame if the link is idle. When a collision is detected, both devices will wait a random amount of time before resending their respective packets.

All 802.11 connections are half-duplex. The only way to achieve full duplex is to send over one channel, and receive over another. The 802.11 standard currently has no such implementation.

802.11 devices have no method of *detecting* a collision, beyond the failure of the receiving device to send an **acknowledgement**. Instead, 802.11 devices attempt to avoid collisions using **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).** Devices will listen before attempting to transmit, and will *only* transmit if no other device is currently transmitting.

If another device *is* transmitting, other devices must wait until that transmission is finished, using a process called **Distributed Coordination Function (DCF)**. The currently transmitting device includes a **duration value** within the 802.11 header, informing other devices of the estimated time-length of its transmission.

Other 802.11 devices will not only wait out this duration value, but will wait an additional random amount of time (referred to as the **DCF interframe space (DIFS))**, before beginning their own transmissions. The random DIFS was implemented to prevent devices from transmitting simultaneously after waiting out another device's transmission duration. DIFS is often referred to as a **random back-off timer.**

The 802.11 Amendments

The **original 802.11** standard was release in 1997, and utilized **direct-sequence spread spectrum (DSSS)** to modulate data onto an RF signal. The standard operated in the **2.4 GHz** frequency range, and had a maximum throughput of **2 Mbps**.

The original 802.11 standard never saw widespread adoption, and was quickly supplanted by the 802.11a and 802.11b **amendments**, which were developed concurrently and released in 1999.

802.11 wireless amendments that are currently in deployment include:

- **802.11a**
- **802.11b**

- **802.11g**
- **802.11n**

(Reference: http://en.wikipedia.org/wiki/IEEE_802.11_(legacy_mode))

802.11a

The **802.11a** amendment was released in 1999, and utilizes **orthogonal frequency-division multiplexing (OFDM)** for modulation. 802.11a operates in the **5.0-GHz** frequency band, and has a *maximum* throughput of **54 Mbps.** Specifically, 802.11a supports data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps, though the higher throughput is only available in close proximity to the wireless access point (WAP)/transmitter.

Because 802.11a operates in the regulated 5.0-GHz band, it is generally free of interference from other RF devices. However, the higher frequency reduces the effective distance of the signal, and is more susceptible to being absorbed by obstructing objects or walls.

802.11a is generally *not compatible* with other 802.11 amendments, as most of the other amendments operate in the 2.4-GHz band.

In the U.S., 802.11a supports a total of **12** non-overlapping channels, **4** of which can be used outdoors. Despite offering a large number of channels and good throughput, 802.11a did not see the same level of widespread deployment as the less expensive 802.11b and 802.11g amendments.

(Reference: http://en.wikipedia.org/wiki/IEEE_802.11a-1999)

802.11b

The **802.11b** amendment was also released in 1999, and utilizes **complementary code keying (CCK)** for modulation. 802.11b operates in the **2.4-GHz** frequency band, and has a *maximum* throughput of **11 Mbps.** Specifically, 802.11b supports data rates of 1, 2, 5.5, and 11 Mbps.

Because 802.11b operates in the unregulated 2.4-GHz band, it is susceptible to interference from other household RF devices.

In the U.S., 802.11b supports a total of **3** non-overlapping channels, specifically channels **1, 6,** and **11**.

(Reference: http://en.wikipedia.org/wiki/IEEE_802.11b-1999)

802.11g

The **802.11g** amendment was released in 2003, and utilizes **orthogonal frequency-division multiplexing (OFDM)** for modulation. 802.11g operates in the **2.4-GHz** frequency band, and has a *maximum* throughput of **54 Mbps.** Specifically, 802.11g supports data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.

As with 802.11b, 802.11g operates in the unregulated 2.4-GHz band, and is susceptible to interference from other household RF devices.

In the U.S., 802.11g supports a total of **3** non-overlapping channels, specifically channels **1, 6,** and **11**.

802.11g is **backward-compatible** with 802.11b, as they both operate in the 2.4-GHz band. However, if an 802.11b device is present in an 802.11g environment, 802.11g will revert to CCK modulation, and will only support throughputs of 1, 2, 5.5, and 11 Mbps.

Neither 802.11b nor 802.11g are backward-compatible with 802.11a.

(Reference: http://en.wikipedia.org/wiki/IEEE_802.11g-2003)

802.11n

The **802.11n** amendment was officially released in 2009, though pre-release (or *draft*) equipment has been available since 2007. 802.11n supports significantly higher data rates than previous 802.11 amendments, through the use of wider channels (**40MHz** channels instead of 20MHz) and **Multiple-Input Multiple-Output (MIMO)**.

MIMO employs multiple antennas on both the transmitter and receiver. The resulting multiple data streams are then combined using **Spatial Division Multiplexing (SDM).** This, coupled with 40MHz channels, allows 802.11n to support throughput up to **600 Mbps.**

802.11n devices are identified by the number of *transmit* and *receive* antennas they support, with a format of Antenna$_{transmit}$ x Antenna$_{receive}$. For example, a WAP with four transmit and three receive antennas would be identified as a *4 x 3* MIMO WAP.

802.11n can operate in either the **2.4-GHz** *or* the **5.0-GHz** frequency bands, *or* both simultaneously. Thus, 802.11n is backwards compatible with 802.11a, 802.11b, and 802.11g. A pure 802.11n environment should operate in the 5.0-GHz band to maximize throughput and to limit interference.

Note also that the wider 40-MHz channel reduces the number of available non-overlapping channels in each band, which provides more incentive to use the 5.0-GHz band. 802.11n *does* support 20-MHz channels, though this will greatly reduce the maximum throughput.

(Reference: http://en.wikipedia.org/wiki/IEEE_802.11n-2009; http://www.airmagnet.com/assets/whitepaper/WP-802.11nPrimer.pdf; http://www.ciscosystems.sc/en/US/prod/collateral/wireless/ps5678/ps6973/ps8382/prod_white_paper0900aecd806b8ce7_ns767_Networking_Solutions_White_Paper.html)

The 802.11 Amendments – Quick Reference

	802.11	**802.11a**	**802.11b**	**802.11g**	**802.11n**
Max Throughput	2 Mbps	54Mbps	11Mbps	54Mbps	600Mbps
Modulation	DSSS	OFDM	CCK	OFDM	OFDM
Frequency Band	2.4GHz	5.0GHz	2.4GHz	2.4GHz	2.4/5.0GHz
Non-Overlapping Channels	-	12	3	3	Varies*
Released	1997	1999	1999	2003	2009

* Varies depending on the Frequency Band, and whether 20MHz or 40MHz channels are being utilized.

* * *

Associating with a Wireless Access Point (WAP)

Recall that a group of communicating 802.11 wireless devices is known as a **service set,** and that there are two *modes* of 802.11 communication:
- **Ad-hoc** or **Independent Basic Service Set (IBSS) –** where wireless clients communicate point-to-point with each other.
- **Infrastructure** or **Basic Service Set (BSS) –** where wireless clients communicate via a Wireless Access Point (WAP).

Wireless clients must **associate** with a WAP before data can be forwarded. Various parameters must match between the client and the WAP:
- **Service Set Identifier (SSID)**
- **Data Rate**
- **Authentication**
- **Encryption/Data Integrity**

The **SSID** is used to *identify* the wireless connection between a WAP (or WAPs) and clients. A wireless client must be configured with the WAP's SSID to associate with it. Otherwise, a client can also *request* (via a *probe*) the SSID if the WAP is configured to **broadcast** the SSID (via a *beacon*). As a best practice, broadcasting is usually disabled in secure environments.

The SSID is often mistaken as a security feature; however, the SSID does not authenticate users or encrypt data – it merely serves as an identifier for a wireless connection. The SSID also provides separation between multiple wireless LANs that might exist in an environment.

Wireless clients are often required to *authenticate* to a WAP. The original 802.11 standard provides for two methods of authentication:
- **Open Authentication -** authenticates *any* wireless client request.
- **Shared-Key Authentication –** requires a matching key to be configured on both the wireless client and WAP.

Open authentication (essentially, *no* authentication) is used for devices that cannot support a complex authentication process. Shared-key authentication employs **Wireless Equivalence Protocol (WEP)** keys for authenticating clients. WEP is covered in detail in the next section.

MAC-address filtering is an additional form of authentication, though not defined in the 802.11 standard. A list of allowed MAC addresses must be maintained on the WAP itself.

Wireless Equivalence Protocol (WEP)

The emergence of 802.11 technologies has introduced new security concerns, due to the open-air nature of wireless transmissions. Such transmissions are easily intercepted, which necessitates mechanisms to not only *authenticate* wireless clients, but also to *secure* data transfer (using **encryption)** and to ensure *data integrity* (using a **32-bit CRC**)**.**

Wireless Equivalence Protocol (WEP) was developed as part of the original 802.11 standard. WEP utilizes the **RC4 stream cipher** for encryption, which combines a **key** with a randomly-generated **initialization vector (IV)** to provide confidentiality.

WEP comes in two common forms:
* **64-bit WEP** – employs a *40-bit* key with a *24-bit* IV.
* **128**-**bit WEP –** employs a *104-bit* key with a *24-bit* IV.

The 128-bit WEP key is represented as a 26-digit hexadecimal string.

WEP can be used with both Open and Shared-Key authentication. With Open authentication, the WEP key is used only for encrypting data. With Shared-Key authentication, the WEP key used for *both* authenticating the wireless client and encrypting data. Regardless of the authentication method, the WEP key(s) **must be identical** on both the wireless client and the WAP.

WEP Shared-Key authentication employs a four-way handshake:
1. The client makes an authentication request to the WAP.
2. The WAP responds with a clear-text *challenge.*
3. The client *encrypts* the challenge with its WEP key.
4. The WAP *decrypts* the encrypted challenge and compares it to the original clear-text challenge.

The authentication process will only be successful if the WEP key is identical on both the WAP and the client. Surprisingly, Shared-Key authentication is less secure than Open authentication. A malicious attacker can intercept both the clear-text and encrypted challenges, and thus somewhat easily derive the encryption key.

WEP is no longer considered a viable security mechanism, as it is *easily* compromised. Additionally, WEP provides only one-way authentication; there is no mechanism within WEP for a client to authenticate the WAP.

(Reference: http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy; http://www.wi-fiplanet.com/tutorials/article.php/1368661)

Wi-Fi Protected Access (WPA)

Wi-Fi Protected Access (WPA) was developed by the Wi-Fi Alliance to address the shortcomings of WEP. WPA incorporates some of the techniques and protocols that were eventually standardized as part of the **802.11i amendment**.

Temporal Key Integrity Protocol (TKIP) is the core component of WPA. Though TKIP employs a **RC4 stream cipher** like WEP, it offers several improvements, including:

- Per-Packet Key Hashing
- 64-bit Message Integrity Check (MIC)
- Broadcast Key Rotation
- Sequence Counting

Note: Cisco developed a proprietary implementation of TKIP that is *not compatible* with WPA TKIP. However, Cisco devices will often support both the standardized and propriety forms of TKIP.

WPA2, also developed by the Wi-Fi Alliance, incorporates *all* portions of the 802.11i amendment. It added support for **Advanced Encryption Standard (AES)** encryption with **Cipher Block Chaining Message Authentication Code Protocol (CCMP)**. AES-CCMP is considered *significantly* more secure than the RC4 stream cipher used by WEP/TKIP. WPA2 also added native support for Intrusion Detection Systems (IDS).

Both WPA and WPA2 support two *modes,* **Personal** and **Enterprise.**

WPA Personal employs pre-shared key (or passphrase) for authentication, and is often referred to as WPA-PSK (Pre-Shared Key). The WPA key can be represented as a 64-digit hexadecimal string, or an 8 to 63 character ASCII string. As with WEP, this key-string must be identical on both the client and the WAP.

WPA Enterprise employs an 802.1X/EAP server (such as a RADIUS server) for centralized authentication. An authentication server eliminates the need for static encryption/authentication keys to be configured on both the client and the WAP.

802.1X/EAP authentication is covered in detail in the next sections.

(Reference: http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access;
http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps430/ps4076/prod_brochure09186a00801f7d0b.html)

802.1X and Extensible Authentication Protocol (EAP)

The **802.1X standard** was developed by the IEEE to authenticate devices on a Layer-2 port basis. It was originally developed for Ethernet (802.3) bridges and switches, but was expanded to support the authentication of 802.11 wireless devices as well.

802.1X defines three *roles* in the authentication process:
- **Supplicant** – the device being authenticated. In an 802.11 environment, the supplicant would be the wireless client software.
- **Authenticator** – the device that is *requiring* the authentication. In an 802.11 environment, this is often the WAP.
- **Authentication Server** – the device that stores the user database, for validating authentication credentials. This is often an external RADIUS server, though some WAPs support a local user database.

802.1X provides the *encapsulation* of **Extensible Authentication Protocol (EAP)** traffic, which serves as the *framework* for authenticating clients. EAP is not an authentication mechanism in itself. Instead, EAP transports the authentication data between supplicants, authenticators, and authentication servers (all three of which must support 802.1X/EAP).

As a general framework, EAP supports a large number of *methods* for authentication, including (but not limited to):
- **Lightweight EAP (LEAP)**
- **EAP - Flexible Authentication via Secure Tunneling (EAP-FAST)**
- **EAP - Transport Layer Security (EAP-TLS)**
- **Protected EAP (PEAP)**

With any form of EAP, wireless clients *must* authenticate with a RADIUS server before any data traffic will be forwarded. Only EAP traffic is allowed between the client and WAP before authentication occurs.

Authenticating clients using 802.1X/EAP offers several advantages over Static-WEP and WPA-PSK, including:
- Centralized management of credentials
- Support for multiple encryption types
- Dynamic encryption keys

(Reference: http://en.wikipedia.org/wiki/IEEE_802.1X; http://en.wikipedia.org/wiki/Extensible_Authentication_Protocol; http://www.ieee802.org/1/files/public/docs2000/P8021XOverview.PDF)

* * *

Lightweight Extensible Authentication Protocol (LEAP)

Lightweight Extensible Authentication Protocol (LEAP) was developed by Cisco, and is supported by WPA/WPA2 as an 802.1X authentication method. LEAP employs a username/password for authentication via a RADIUS server, and does *not* require the use of certificates.

LEAP is supported by most operating system, including Mac OS, Linux, DOS, and most versions of Windows. LEAP additionally supports single sign-on in Windows environments, allowing clients to perform Active Directory (or NT Domain) and 802.1X authentication simultaneously.

LEAP authentication is a multi-step process:
1. The supplicant initiates the connection with a *Start* message.
2. The authenticator responds with a *Request/Identity* message.
3. The supplicant responds with an *Identity* message containing a username.
4. The authenticator then forwards the username to the authentication server with an *Access Request* message.
5. The supplicant and authentication server then authenticate *each other* using a challenge/response method. The authentication server sends a randomly-generated *challenge* to the supplicant. The supplicant then generates a hash value from the challenge and its password, using MD5. This hash value serves as the *response* back to the authentication server, and eliminates the need for the actual password to be transmitted between the two devices.
6. A *Success* message is generated if the supplicant and authentication server have successfully authenticated each other, which informs the authenticator that the supplicant can now pass data traffic.

Once authentication is completed, the supplicant and authentication server then generate a **pairwise master key (PMK).** The PMK is used to create the actual encryption keys for data transfer, via a four-way handshake.

LEAP was built on a variation of MS-CHAP, and is thus vulnerable to dictionary attacks. A strong password policy is extremely important when employing LEAP in a business environment. If strong passwords are not possible, Cisco recommends utilizing EAP-FAST instead of LEAP.

(Reference: http://www.ciscosistemi.com/en/US/prod/collateral/wireless/ps5678/ps430/prod_qas0900aecd801764f1.html; http://www.cisco.com/application/pdf/en/us/guest/netsol/ns386/c649/ccmigration_09186a0080871da5.pdf; *CCNP ONT Exam Certification Guide, Amir Ranjbar. Pages 262-264*)

* * *

EAP with Flexible Authentication via Secure Tunneling (EAP-FAST)

EAP-FAST was also developed by Cisco as an alternative to LEAP, and was standardized by the IETF. Like LEAP, it utilizes a username/password for authentication via a RADIUS server, and does *not* require the use of certificates. Unlike LEAP, EAP-FAST is not vulnerable to dictionary attacks, as it establishes a secure tunnel between the supplicant and authentication server.

EAP-FAST is supported by most versions of Windows, and supports Windows single sign-on in Active Directory/Domain environments.

EAP-FAST authentication is a three-phase process:
- **Phase 0** (*optional)* – the supplicant is assigned a **Protected Access Credential (PAC),** on a per-user basis**.** This phase is optional because the PAC can be manually configured on the supplicant.
- **Phase 1 –** the supplicant and authentication server establish a secure tunnel using the PAC.
- **Phase 2 –** the supplicant sends its username/password credentials to the authentication server, via the secure tunnel.

(Reference: http://www.cisco.com/en/US/prod/collateral/wireless/ps5679/ps5861/prod_qas09186a00802030dc_ps430_Products_Q_and_A_Item.html; *CCNP ONT Exam Certification Guide, Amir Ranjbar. Pages 264-266*)

EAP with Transport Layer Security (EAP-TLS)

EAP with **Transport Layer Security (EAP-TLS)** is an IETF standard protocol, and was the *first* EAP authentication method used with 802.11 wireless networks.

EAP-TLS utilizes Public Key Infrastructure (PKI) to authenticate supplicants using certificates. Both the supplicant *and* the authentication server must be assigned a certificate from a Certificate Authority (CA) server. Because of this, EAP-TLS is considered *extremely* secure, though the complexity of client-side certificates makes it somewhat unpopular.

EAP-TLS is natively supported on most versions of Windows (2000 and newer).

(Reference: http://www.cisco.com/en/US/tech/tk722/tk809/technologies_white_paper09186a008009256b.shtml; *CCNP ONT Exam Certification Guide, Amir Ranjbar. Pages 266-267*)

Protected EAP (PEAP)

Protected EAP (PEAP) was developed jointly by Cisco, Microsoft, and RSA Security, and was submitted to the IETF for standardization.

PEAP utilizes TLS to create a secure tunnel between the supplicant and authentication server. The key difference between PEAP and EAP-TLS is that only the *authentication server* requires a PKI certificate – no certificate is required on the supplicant.

PEAP authentication is a two-phase process:
1. The supplicant authenticates the authentication server by verifying the server-side PKI certificate. If successful, the supplicant and authentication server form the TLS tunnel.
2. The supplicant sends its username/password credentials to the authentication server, via the secure tunnel. This is accomplished using either EAP-MSCHAPv2 (for Windows-based authentication servers) or EAP-GTC (Generic Token Card, for LDAP-based authentication servers).

As with the other EAP-methods, a *Success* message is generated if the supplicant and authentication server have successfully authenticated each other, which informs the authenticator that the supplicant can pass traffic.

* * *