802.11s
Mesh Networking

Whitepaper

Author: Jerome Henry

Editor: Marcus Burton

November 2011
Version 1.00

Table of Contents

Introduction

After more than 7 years of efforts, in the fall of 2011, the IEEE published the 802.11 amendment for mesh networking, 802.11s. Although this amendment if primarily focused on mesh networks, it contains innovative mechanisms that, once integrated into the 802.11 standard, may apply to all Wi-Fi networks, and may solve issues such as WPA/WPA2 pre-shared key authentication attacks or frame collisions between neighboring access points in dense environments.

This whitepaper describes mesh networks, the initial target for the 802.11s amendment, and lists issues that the 802.11s amendment addresses. This document then examines the main 802.11s mechanisms. You will learn how mesh stations discover each other, build peering relationships, secure their communications, and dynamically discover the best path to any given destination. You will see how the path discovery dynamically adapts to changes in the RF environment and how it integrates collision avoidance mechanisms. You will also see that 802.11s takes into account power consumption, allowing new variations on power save: a light sleep mode and a deep sleep mode.

What are mesh networks, and why an amendment was needed

With the rapid adoption of wireless networks came the need to provide wireless access in places where connecting an AP to a switch was not possible. The length of an Ethernet cable is limited to 100 m (328 feet), making it difficult to position some access points in the center of large indoor environments, such as warehouses. The issue becomes even worse with the need to provide wireless coverage outdoors. The use case may be simple extension of the indoor wireless network to a parking lot, a campus, or an outdoor industrial area, or it may encompass entire cities to provide wireless access to the general public, municipality services, or emergency responders. WLAN services can range from vehicle tracking and monitoring or wireless security cameras to utility reporting (for example, gas meter monitoring and reporting). The use cases are many and growing every day.

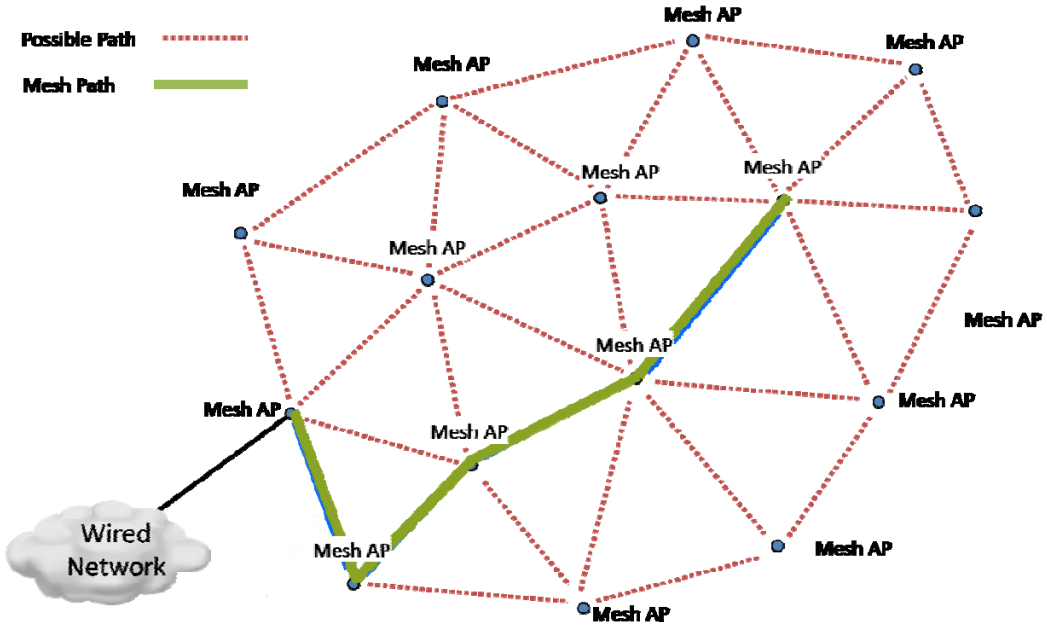
The idea to use a wireless link to replace some Ethernet cables is as old as 802.11. In 2003, the 802.11 working group defined the concept of a Wireless Distribution System (WDS) as *a mechanism for wireless communication using a four address frame format* (802.11 def 3.170) between access points. But the working group could not go any further than this simple definition, and indicated that *the standard describe[d] such a frame format, but [did] not describe how such a mechanism or frame format would be used* (802.11 def 3.170).

Replacing an Ethernet cable with a wireless link indeed brings many benefits:

- The first benefit is obviously the increased **flexibility** of a wireless link over a wired link. When all the access points connect to a switch, you need as many switch ports as you have access points, and all the access points must be within a 100 meter range of the switch. With wireless links, you may need a first access point to connect to a switch and the wired network, but then many other APs can connect through this first access point, even if they are miles away from the switch, and even if they are on moving objects (trains, cranes, etc.). The benefit of flexibility also lies in the path(s) taken by the wireless link. With an Ethernet cable, there is only one possible path from the AP to the

switch. With a wireless link, any AP may be in range of one or several APs, and it can choose the best radio path. This possibility for any AP to connect to one or several other APs, and the possibility for a redundant connection, is the very definition of a wireless mesh network. The multiple inter-mesh AP links form what is called the **backhaul**, as multiple users' data is backhauled through the mesh cloud to the main distribution points to the wired network.

Figure 1: Basic mesh topology



- The second benefit is that such a wireless network is **self forming**. If an algorithm is embedded into a mesh AP to detect the best path to the wired network, building or expanding a wireless mesh network may be as simple as adding new access points and making sure that they are in range of other access points.
- The third benefit is that such a network is **self-healing**. If an access point has several possible paths to the wired network, and if the AP is able to automatically choose the best path, removing one access point in the mesh cloud simply forces the other access points to find the new best path to the wired network, without the need for a wireless engineer to be deployed to replace the missing access point.

These benefits prompted several vendors to start designing and implementing mesh solutions as early as 2003. However, along with the benefits come several delicate questions:

- How does a mesh AP discover the other access points?
- How does a mesh AP determine the "best path to the wired network"?
- How should mesh APs form the mesh cloud? In other words, by what mechanism should an access point allow another access point to borrow its wireless link to transmit data elsewhere?

- How to prevent unwanted access points from joining the mesh cloud?
- How to secure the communication between mesh access points, to prevent eavesdropping, data theft or even mesh network hijacking?

All these questions were left unanswered by the 802.11 standard, even in its 2007 version. The concept itself of a wireless mesh cloud is nowhere to be found in the 802.11 standard. Such a cloud is not an IBSS. An IBSS implies stations forming a network without connectivity to a distribution system. Such a cloud is also not an infrastructure network, as no access point is the central point of communication for the stations in the cloud.

This lack of definition has been accompanied by proprietary and conflicting features among vendors. Each vendor implementing a proprietary solution found different answers to some of the questions above. When implementing a wireless mesh network today, you know that there will be no interoperability with access points from other vendors (even indoor access points serving the same SSID as the mesh access points). The vendor will use specific and proprietary terms, frame formats, and exchanges for each feature, making inter-vendor comparison very challenging. You cannot even know if all the questions above were answered by the vendor solution, or if the focus was put only on some aspects of the mesh cloud issues.

There was a strong need for a framework providing a clear terminology, list of features, and behaviors that could be integrated to the 802.11 standard. This effort started in 2003 by the creation of an IEEE 802.11 study group for mesh networks, then a task group in July 2004. It took more than 7 years for the task group to come to a final amendment for mesh networking. This amendment was finally approved in July 2011 and published in fall 2011 as 802.11s.

This amendment is key to a common understanding of wireless mesh networks because it defines mesh network protocols and functionalities, while still leaving wide space for proprietary implementations of each of the possible features. By comparing any proprietary mesh solution to the 802.11s amendment, you will be able to determine the list of features the vendor implemented, compare the proprietary solution to the standard mechanisms, and decide if the vendor implemented all key elements or if critical questions were left unanswered.

802.11s architecture

Definitions

station: Any device that contains an IEEE 802.11-conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

AP: Any entity that has station (STA) functionality and provides access to the distribution services, via the wireless medium (WM) for associated STAs.

mesh facility: The set of enhanced functions, channel access rules, frame formats, mutual authentication methods, and managed objects used to provide data transfer among autonomously operating stations (STAs) that may not be in direct communication with each other over a single instance of the wireless medium.

mesh station: A quality-of-service (QoS) STA that implements the mesh facility.

mesh gate: Any entity that has mesh station (STA) functionality and provides access to one or more distribution systems, via the wireless medium (WM) for the mesh basic service set (MBSS).

mesh BSS (MBSS): A basic service set (BSS) that forms a self-contained network of mesh stations (STAs). An MBSS contains zero or more mesh gates.

portal: The logical point at which the integration service is provided.

mesh coordination function (MCF): A coordination function that combines aspects of the contention-based and scheduled access methods. The MCF includes the functionality provided by both enhanced distributed channel access (EDCA) and MCF controlled channel access (MCCA).

mesh coordination function (MCF) controlled channel access (MCCA): A coordination function for the mesh basic service set (MBSS).

precursor: A neighbor peer mesh STA on the mesh path to the destination mesh STA, that identifies the mesh STA as the next-hop mesh STA.

source: A mesh STA from which a MAC service data unit (MSDU) enters the mesh basic service set (MBSS). A source mesh STA may be a mesh STA that is the source of an MSDU or a proxy mesh gate that receives an MSDU from a STA outside of the MBSS and forwards the MSDU on a mesh path.

Architectural overview

Because mesh networks are different from traditional wireless networks, the 802.11s amendment changes the name and functionality of several wireless infrastructure components. Nothing needs to change on the wireless client station (non-AP client device, such as a laptop or VoWLAN phone) side. A wireless client can still associate to an access point's BSS normally, without needing to know if the access point connects to the wired network directly (standard AP) or through a mesh cloud (mesh AP). In Figure 2, client stations H, I, K, L N, O, Q, and R do not need any specific 802.11s functionality to connect to their respective access points.

Yet the 802.11s amendment defines the concept of a **mesh station** (5.2.14.3). A Mesh station is simply a station that supports the **mesh facility** and is capable of participating in a mesh cloud, or **Mesh Basic Service Set (MBSS)**. The mesh facility is simply the set of features, functions, and frame formats that enable mesh operation. The term mesh station typically designates an access point, but nothing prevents a non-AP STA (client device) from being a mesh station, at least theoretically. In Figure 2, A, B, C, D, E, F, G, J, and M are mesh stations. Most of them are access points, but C and D are non-AP stations that integrate mesh (and 802.11s) functionalities—this is not likely in actual deployments.

Mesh Basic Service Set (MBSS) is the official name of the mesh could. An MBSS is *an 802.11 LAN consisting of autonomous stations* (5.2.14.2). These stations establish peer-to-peer wireless links and transfer messages mutually. In this sense, the MBSS is closer to an IBSS than to a standard BSS. An important difference with an IBSS, however, is that

messages can be transferred between stations that are not in direct communication with each other through other stations of the MBSS, using the mesh cloud backhaul. All stations appear as one Layer 2 group of devices. Stations in a mesh BSS might be sources, sinks or propagators of traffic. In the illustration below, station B may be a source mesh station (sending traffic but not relaying any other mesh station traffic) or a sink (receiving traffic but not sending or forwarding anything), depending on the direction of traffic. A, C, D, E, and G may just be forwarding traffic coming from other stations (propagators), without sending or receiving traffic of their own.

The 802.11s amendment states that a mesh station is not a member of an IBSS or an infrastructure BSS and that, consequently, *mesh stations do not communicate with non-mesh stations* (5.2.14.4). This can seem confusing: how can a mesh network possibly connect to the wired network and other 802.11 devices? The reasoning behind these definitions is that, in the 802.11s amendment, the "mesh station" is a function. In our everyday vernacular use, a "station" is typically a single physical device, but in 802.11 terms, a "station" is a subset of functions for a device. An access point can integrate the mesh station function to communicate with other mesh stations, and also integrate an AP function to allow for 802.11 client device associations. The MBSS can interconnect with infrastructure BSSs through the Distribution System (DS). Then, mesh stations can communicate with non-mesh stations through the DS. Therefore, a logical architectural component is introduced in order to integrate the MBSS with the distribution system—the **mesh gate**. Data moves between an MBSS and the DS via one or more mesh gates. Confusion often arises at this point between the terms AP, Portal, and Mesh gate.

The term distribution system (DS) is fundamental to a thorough understanding of the roles of a portal and a gate, so a quick explanation may be helpful. The DS is a logical component that handles address to destination mapping. That is, it "distributes" frames from one interface to another. This could distribution of a frame from a client on the 2.4 GHz radio to a client on the 5 GHz radio; it could be from a client on the 2.4 GHz radio to the wired network; or it could be from a client on the 2.4 GHz radio to an MBSS on the 5 GHz radio. In any case, it is important to understand that the DS is a logical construct and not necessarily a separate medium, like the Ethernet.

The 802.11-2007 standard defines the **AP** (def 3.3) as *a station that provides access to the distribution services, via the wireless medium (WM), for associated stations*. In other words, the AP is the central point of transit for client stations connected to the AP cell, and delivers the 802.11 frames sent by these stations. In the illustration below, J, M, P, and S are APs. The AP can deliver frames to other stations in the cell or to the wired network. J, M, and P also deliver their associated stations' traffic to the distribution system, while S only delivers the wireless traffic to other stations in the cell. The function of an AP that performs this translation between the wireless network and the wired network (as with M and P) is called the **portal**. The portal is a function. It is just *the logical point where wireless MSDUs are translated to and from a non-802.11 network* (802.11-2007, def 3.39 and 3.110). An AP commonly performs the portal function. M, and P are APs and portals.

When the translation occurs between a mesh BSS and a non-mesh 802.11 DS, this function is called **mesh gate**. The mesh gate is the logical point where mesh MSDUs are translated to and from a non-mesh 802.11 DS and format. Other functions are usually required to complete the translation. For example, if a frame coming from the MBSS is sent to an 802.11 cell (BSS), the AP function takes care of forwarding the frame to the wireless station(s) in the cell. If a frame coming from the MBSS is sent to a non-802.11 network, the portal function

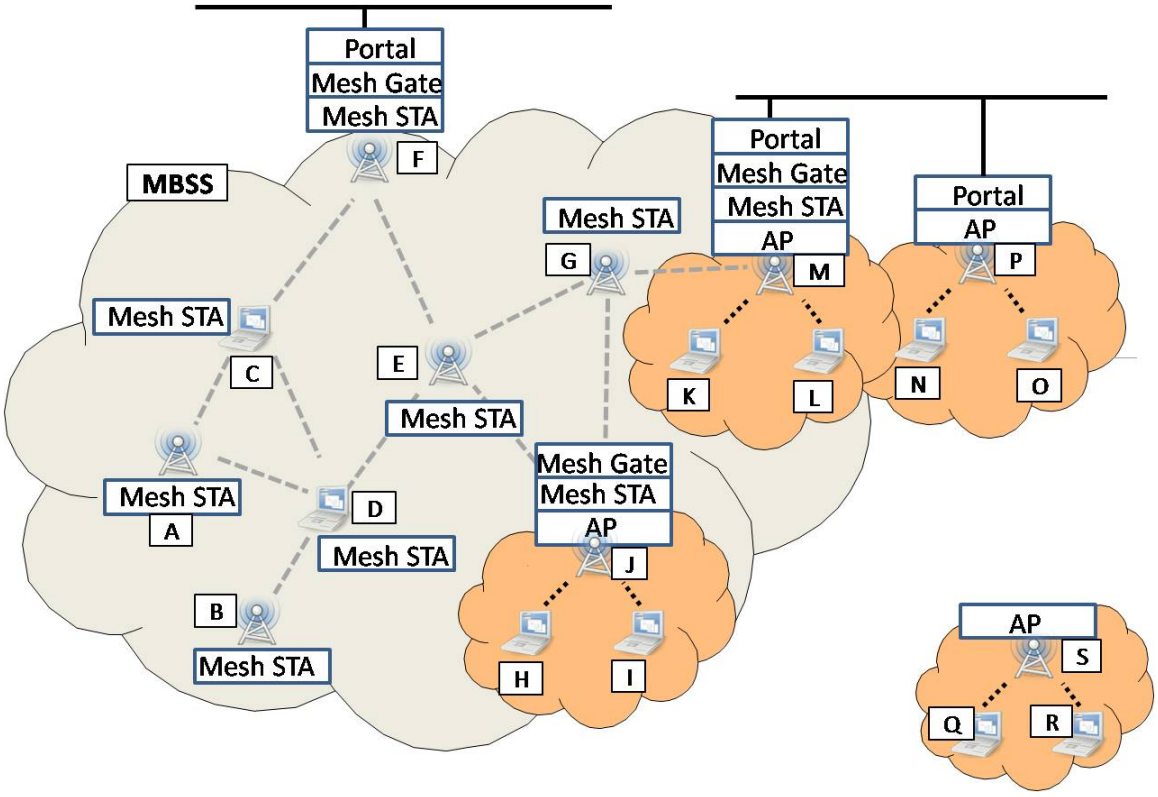
takes care of forwarding the frame to the wired network. A single device can be the central point of a cell (thus being an AP), connect its wireless clients to the wired network (thus being a portal), while also participating in a mesh BSS and connect other mesh stations to the DS (thus being a mesh gate):

- J is an AP because it connects its associated wireless client stations, a mesh station because it participates in the MBSS, and a mesh gate because it translates between the MBSS and DS.
- M is an AP because it connects its associated wireless stations, a portal because it connects its wireless clients to a non-802.11 DS, a mesh station because it participates in the MBSS, and also a mesh gate because it connects the MBSS to the DS.
- F is a mesh station because it participates in the MBSS, and also a mesh gate and a portal because it connects the MBSS to the non-802.11 DS. F is not an AP as it does not have any associated stations.

A mesh gate is an important element of the mesh cloud, and identifies itself as such to the other mesh stations in the MBSS.
From the perspective of any mesh station in the mesh cloud, a next hop mesh station on the path to the destination mesh station is called a **precursor** mesh station. E is a precursor mesh station for J or D, their next hop on the path to mesh STA F.

For stations participating in an MBSS to be able to detect each other, they must be on the same channel. Access to the channel is a key component of the mesh infrastructure, and a new coordination function is created for mesh networks; the **mesh coordination function (MCF)** is a coordination function that combines aspects of the contention-based and scheduled access methods. The MCF integrates elements of 802.11e EDCA and HCCA to form the mesh coordination function **mesh controlled channel access (MCCA).**

Figure 2: Mesh architecture components



Forming the mesh: discovery, peering, security

Discovering other mesh stations

When a mesh station boots up, it needs to discover and join a mesh network (that is, establish peer relationships with other mesh stations). The discovery process uses the standard active and passive scanning mechanisms (5.2.14.5.1 and 11C.2). Mesh stations participating in an MBSS send beacons and answer to probe requests with probe responses. The major difference with standard 802.11 frames is that mesh stations' broadcasts and probes (requests and responses) contain several new elements. These elements form what is called the **mesh profile**. This mesh profile is a set of parameters that specifies the attributes of a mesh BSS; these attributes consist of a **Mesh ID** and multiple parameters advertised in the **Mesh configuration Element**. In a mesh BSS all mesh STAs use the same mesh profile. Mesh profiles are considered the same if all parameters in the mesh profiles match. A mesh station cannot establish a peering with another mesh station if their mesh profiles are different. A mesh profile consists of the following:

- The **Mesh ID element** (7.3.2.99 and 11C.2.2). The Mesh ID is a 0 to 32 byte field, close to the SSID in concept. It can be an ASCII string, and uniquely identifies the MBSS.
- The **Mesh Configuration element** (7.3.2.98). This element contains several subfields that describe the mesh capabilities of the local mesh station:

- A **path selection protocol identifier** (7.3.2.98.2), identifying which protocol is used to determine the best path to the wired network or any destination in the mesh cloud
- A **path selection metric identifier** (7.3.2.98.3), identifying the metric used to calculate the best path to the wired network or any other destination in the cloud
- A **congestion control mode identifier** (7.3.2.98.4), identifying which protocol is used to manage congestion in the MBSS
- A **synchronization method identifier** (7.3.2.98.5), identifying the synchronization method among mesh stations
- An **authentication protocol identifier** (7.3.2.98.6), identifying which authentication method and protocol are in used between mesh stations.
- A **Mesh Formation Info element** (7.3.2.98.7), that specifies how many peers the local station has, and if the station is connected to the wired network or to a mesh gate
- A **Mesh capability element** (7.3.2.98.8), specifying among other parameters if the station accepts new peerings.

Note that mesh beacons are sent independently from standard 802.11 beacons. A mesh station also performing the role of an AP would send 2 beacons, one for its AP role (for each BSS), and one for its mesh station role.

Peering with other mesh stations

After mesh discovery with passive or active scanning, two neighbor mesh STAs (STAs within direct wireless communication with one another) need to agree to establish a mesh peering to each other. After successfully establishing the mesh peering, they become **peer mesh stations** and can communicate directly with one another. A mesh station can establish a mesh peering with multiple neighbor mesh stations (5.2.14.5.2), and can also establish multiple peering sessions with a given neighbor, if necessary.

A key characteristic of the peering mechanism is to be a distributed, non-hierarchical, and non-exclusive agreement to communicate. Each mesh station manages its peerings with other mesh stations. When peering occurs, each side offers and agrees (with a Confirm message) to parameters that define the conditions of the peering and the subsequent communications. Two peering modes are defined: a secured peering mode, through the **Authenticated Mesh Peering Exchange** (**AMPE**, 11C.5), and an unsecured peering mode through standard **Mesh Peering Management** (**MPM**, 11C.3). When security is enabled on mesh stations, AMPE is always used. MPM is used only when security is not enabled.

Peering uses **Mesh Peering Open frames, Mesh Peering Confirm frames, and Mesh Peering Close frames** to establish, manage, and tear down a mesh peering. All these frames are action frames, but are also considered mesh management frames.

After having discovered a neighbor sharing the same mesh profile, a mesh station can send a **mesh peering open frame** to offer a peering connection to the neighbor. The sending station is the **peering initiator**, and the responding station the **peering responder**. The mesh peering open frame is an action frame (but also considered a mesh management frame). This frame is close in its structure to the association request frame, but was modified to match the needs of an MBSS. The mesh peering open frame contains the station

capability, supported rates, power management capability (when 802.11h is in use), supported channels, RSN, HT information, mesh security information (when security is in use), and optional vendor specific information. The frame also contains the mesh ID, the mesh configuration IE and a mesh peering management field. This field specifies if security is in place for this peering, and provides a unique number to the link to this neighbor. A space is also present in this field to confirm the link number assigned by the neighbor when peering is confirmed.

If the neighbor agrees to the peering (because the neighbor has the same mesh profile and is set to accept peerings), a **mesh peering confirm** action frame format is returned. This frame contains the same elements as the mesh peering open frame, with 2 differences:

- The mesh peering management field now mentions both the local link ID, a unique identifier assigned to this peering by the local mesh station, and the peer link ID, the unique identifier assigned to this peering by the neighbor mesh station
- The frame also contains an AID. This AID is a number assigned by the local mesh station to the neighbor mesh station, and is similar in concept to the AID assigned to associated stations by an AP: this number uniquely identifies the neighbor.

Notice that the peering process has to occur both ways: each side has to offer attributes, and each side has to confirm the peering:

- Side A offers, side B confirms, then side B offers and side A confirms
- Or side A offers, side B offers, side A confirms and side B confirms (or side B confirms then side A confirms)

Any sequence of operation is allowed, but the process must be bidirectional to be complete.

Peering is then maintained as long as mesh stations are in range of each other and share the same mesh profile. Peering can be terminated (11C.3.8 and 11C.4.3) if the local station fails to hear the neighbor for a long (configurable) time, if the neighbor fails to answer after a frame was sent a certain (configurable) amount of times, if the local station exceeds its maximum (configurable) number of peers, if the neighbor mesh profile does not match the local station mesh profile anymore, or if there is a security parameter mismatch between stations. If the neighbor station is a path toward the wired network, the local station may also choose to cancel its peering if the neighbor stops providing access to the wired network.

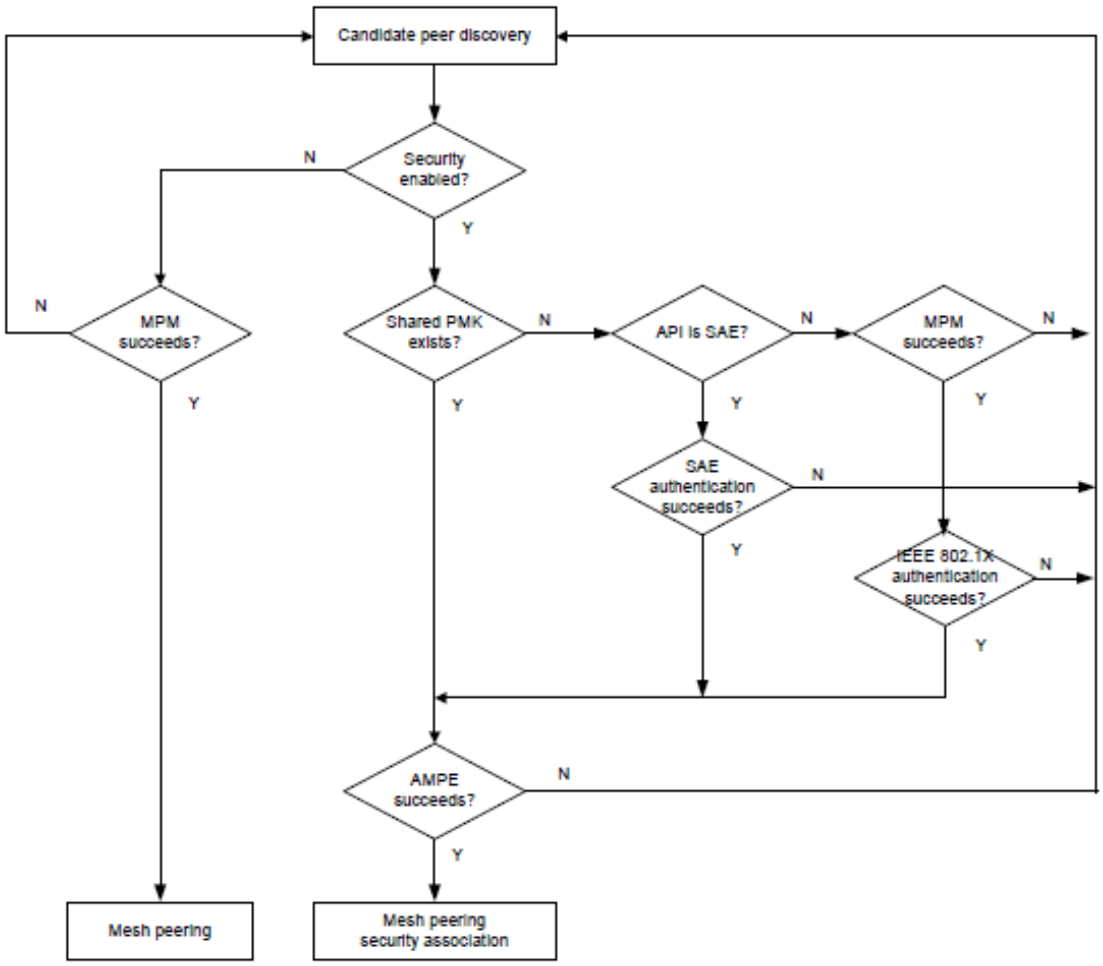
The **mesh peering close** action frame (7.4.14.4.2) is a simple frame that contains the Mesh ID, security elements if security is in use, optional vendor specific information, and the mesh peering management field. This field still contains the local and neighbor unique link identifiers, but also contains a reason code for the termination.

Securing Mesh Peers

A key concern for peering is security. As peering is a very flexible process, the risk exists that a rogue mesh station would peer with a valid mesh station, thus hijacking a legitimate MBSS's bandwidth or offering rogue connections to fake resources or the wired network. Although open peering with standard MPM is allowed, Authenticated Mesh Peering

Exchange (AMPE) is the expected common implementation. Figure 3 is a flowchart that summarizes the different mesh peering authentication mechanisms.

Figure 3: Mesh peering and security flowchart



AMPE is MPM with security provided by a PMK exchange. This PMK can be derived with two possible methods: 802.1X and Simultaneous Authentication of Equals (SAE). 802.1X is very secure and relies on an authentication server to uniquely identify each mesh station and provide a PMK to secure subsequent exchanges. With 802.1X, the initiator still sends a mesh peering open frame. The RSN IE of the frame indicates that 802.1X is used. The responder agrees on this scheme and acts as an 802.1X authenticator, relaying the initiator identification to the authentication server. As peering is bidirectional, the responder will become the initiator in the next part of the exchange, so that both sides can be authenticated. Of course, the authentication becomes unidirectional if only one station is connected to an authentication server, which is indicated in the *Connected to AS* field. A great flexibility of this scheme is that authentication can occur at any phase of the peering process. As 802.1X authentication can be time consuming, this authentication does not have to be a condition of the peering. In other words, mesh stations can use MPM while negotiating a more robust authentication with 802.1X, then switch to the secure mode by setting up a new peering based on 802.1X and terminating the MPM peering.

A limitation of 802.1X is AAA server reachability. If the Authentication server is on the wired network, using 802.1X implies that both mesh stations have access to the wired network and the server. Positioning the AAA server in the wireless network (in a mesh gate/portal or mesh station for example) simply moves the reachability issue to the wireless side of the network without providing an easy solution. For this reason, the 802.1X authentication is not expected to be the main authentication scheme, at least in the first implementations.

This limitation is the reason why another authentication mechanism was built, **Simultaneous Authentication of Equals (SAE)**, which is required for mesh STAs to promote interoperability. SAE is a peer to peer, mutual authentication process. It relies on the fact that a password was defined on both neighbors (and that they have the same password): no central server is required. An interesting aspect of SAE is that the process was built to protect the password in the exchanges. The password is never sent between peers (in clear or encrypted in any way) during the SAE exchange. *SAE is a variant of Dragonfly, a password-authenticated key exchange based on a zero-knowledge proof (a method to prove that you know a password without revealing anything about this password). SAE is used by STAs to authenticate with a password; it has the following security properties:*

- *The successful termination of the protocol results in a PMK shared between the two STAs.*
- *An attacker is unable to determine either the password or the resulting PMK by passively observing an exchange or by interposing itself into the exchange by faithfully relaying messages between the two STAs.*
- *An attacker is unable to determine either the password or the resulting shared key by modifying, forging, or replaying frames to an honest, uncorrupted STA.*
- *An attacker is unable to make more than one guess at the password per attack. This implies that the attacker cannot make one attack and then go offline and make repeated guesses at the password until successful. In other words, SAE is resistant to dictionary attack.*
- *Compromise of a PMK from a previous run of the protocol does not provide any advantage to an adversary attempting to determine the password or the shared key from any other instance.*
- *Compromise of the password does not provide any advantage to an adversary in attempting to determine the PMK from the previous instance (8.2.1a).*

When authenticating each other, each side derives a number from the shared secret, and sends this **derived number** with an identifier (the **scalar**). The other side verifies that it would derive the same number when using the same identifier, so that the other side "guessed" the password correctly. Deriving the original password from the identifier and derived number is so complex that this derivation is deemed impossible. An offline brute force or dictionary attack cannot be used to deduce the password from the derived number and the identifier.

Unlike other authentication protocols, SAE does not have an absolute notion of an "initiator" and "responder" or of a "supplicant" and "authenticator." The parties to the exchange are equals, with each side being able to initiate the protocol. Each side may initiate the protocol simultaneously such that each side views itself as the "initiator" for a particular run of the protocol. The parties involved are identified by their MAC addresses. Stations begin the protocol when they discover a peer through Beacons and Probe Responses, or *when they*

receive an IEEE 802.11 authentication frame indicating SAE authentication from a peer (8.2a.1). Just like for 802.1X, authentication can occur before peering, during the MPM peering phase or after, using first MPM then switching to secured peering with AMPE. This flexibility in the process is implemented to save time. But a peering with SAE or 802.1X is complete only if the authentication succeeded and if peering follows this authentication. In other words, 802.1X or SAE occur after discovery but before secured peering. If peering using MPM is done first, a new peering must be conducted using SAE or 802.1X once the authentication phase has completed.

The SAE process consists of two message exchanges, a **commitment exchange** and a **confirmation exchange**. The commitment exchange is used to force each party to the exchange to commit to a single guess of the password. The confirmation exchange is used to prove that the password guess was correct. Authentication frames are used to perform these exchanges (8.2a.5.1).

Once a station receives a commit message, it processes the message to verify if the other station guessed the password correctly. It also sends its own commit message to prove that it can guess the password.
Once both parties have committed, and if the password is correct, either side can reply with a confirmation message. In other words, a station cannot send a confirm message before both sides have committed.

The side that receives the confirm message then silently accepts the authentication. Once both sides have accepted the authentication (that is, have first sent a commit message then received a confirm message), SAE terminates. In other words:

- A party may commit at any time
- A party confirms after it has committed and its peer has committed
- A party accepts authentication after a peer has confirmed
- The protocol successfully terminates after each peer has accepted

SAE is the recommended method whenever 802.1X is difficult to implement. It is considered as more secure than a PSK exchange. From the point when SAE completes, both sides have a PMK and can start encrypting their communication. This state is comparable to the result of a WPA2 authentication state where both parties have the PMK, except that SAE is not an authentication process based directly on keying dialog. All subsequent neighbor communications are encrypted and protected using AES-CCMP (8.4.1.1). The PMK is used to derive a Mesh Temporal Key (MTK), used for encrypting unicast traffic between peer mesh stations, and a group key (GTK) is derived for broadcast traffic. In an MBSS, each mesh station defines its own “transmit mesh GTKSA”, which is used to encrypt its group addressed transmissions. Also each mesh STA stores a separate “receive mesh GTKSA” for each peer mesh station so that encrypted group addressed traffic received from the peer mesh stations may be decrypted. (8.4.1.1.3b)

Once authentication is complete and both stations have peered, a mesh station can start building its determination of the “best path to destinations”.

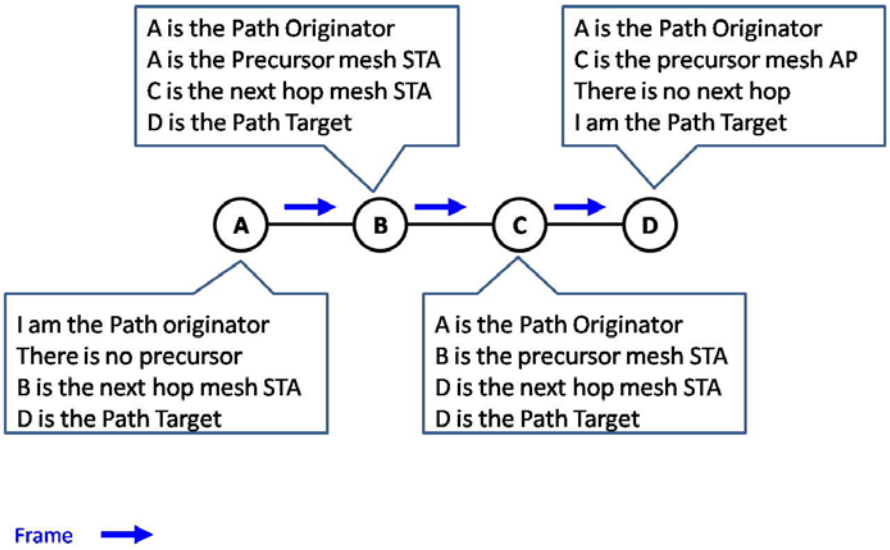
Selecting and maintaining the best path while allowing nodes to doze

Finding the best path

Finding a way to reach a given destination is called **Path Selection** in 802.11s. This destination is commonly the wired network, but can also be any other MAC address reachable through the MBSS.
A vendor proprietary method can be used for path discovery and selection, but mesh stations implementing 802.11s must also support the default path selection protocol, **Hybrid Wireless Mesh Protocol (HWMP)**. HWMP provides both proactive path selection and reactive path selection. A mesh station that needs to transmit a frame to an unknown destination can dynamically discover the best path to this destination. Mesh stations can also proactively discover the MBSS and determine best paths to any point of the mesh cloud before needing to send any data frame.

A specific terminology is used to describe each mesh station role in the path determination process, which is illustrated in Figure 4. Suppose that a mesh station A needs to send a frame to a mesh station D. D may not be the final destination MAC address of the frame (the DA), but may be the last mesh station in the MBSS to have access to the DA. D may be a mesh gate/portal and DA may be on the wired network. D may be an AP and DA may be a wireless station attached to D's cell. D may also be the intended final destination of the frame. In any case, A needs to find a path to D. As such, A is called the **Path Originator**. A is not in direct wireless range of D, and will have to send a path discovery frame to its neighbor B. From A's standpoint, B is the **Next Hop**. From B's standpoint, A is the **Precursor**, as B received the frame from A. A is also the path originator, as A is the station initiating the path discovery process. A will be seen as the path originator by all mesh stations on the path to D. D being the mesh station to be discovered is called the **Path Target**.

Figure 4: Terminology used with path determination



Specific frames are used for path management: **HWMP Mesh Path Selection frames** (7.4.15.3). These are action frames; the Action category is 13 (mesh), and the subtype is 1 (HWMP Mesh Path Selection). The HWMP Mesh Path Selection frame contains several Information Elements. All these elements are optional, and their presence depends on the type of action the frame is intended to accomplish: these elements are the **Path Request element** (optional element 3), the **Path Reply element** (optional element 4), the **Path Error element** (optional element 5) and the **Root Announcement element** (optional element 6).

Path discovery relies on **Path Requests (PREQ)** and **Path Replies (PREP).** Suppose that mesh station A needs to discover the path to mesh station D. Station A sends PREQ frames to all mesh stations in range. The PREQ element is contained in a HWMP Mesh Path Selection frame, with only element 3 set. The PREQ element is 37 to 352 bytes long, and contains several subfields of importance (7.3.2.113):

- The **MAC address** of the **originator** of the PREQ (in our example, mesh station A). This MAC address will be transmitted along with the action frame, so that all stations on the path know which station originated the request.
- An **Originator HWMP Sequence Number**: this number uniquely identifies the request sent by the originator, and is kept unchanged as the frame is transmitted from one station to the next. Each station can check this number and verify if the request was already received (sign of a loop) or if it is new.
- A **path discovery ID**: this number is set by the originator, and kept unchanged. It is used to uniquely identify the path that the originator is trying to build.
- A **Time To Live** (TTL) field and a Life Time field: these fields are present to avoid loops. The PREQ is allowed a defined number of hops (TTL). Each station on the path decrements this field by 1. The Life Time plays the same role, but is expressed in TUs.
- A **Hop count to the originator**. This field is incremented by each station on the path, and allows each station to determine its hop count to the originator. This field is also used for loop prevention.
- A **metric** field. This component shows the total metric to the originator, and is modified by each station on the path. Any receiving station determines its metric back to the originator by taking the value of the metric field, and by adding its own metric value to the emitting station.
- A **target count**: the originator may need to discover a specific station MAC address, or can proactively discover several of its neighbors. A mesh station can use the PREQ process to discover up to 20 other mesh stations.
- For each target station, subfields to the PREQ element mention the **target address** (the broadcast MAC address, or a specific MAC address), and a **target** specific **HWMP Sequence Number**. This number is the last known HWMP sequence number to that target, and is unused if no previous HWMP sequence number was used to discover this target.

A station receiving an HWMP Mesh Path Selection frame containing a PREQ may reject it in some cases (the main case being when the receiving station has no information about the destination MAC address that is to be discovered). The station that has no path to the intended destination replies with a **Path Error** (**PERR**) message, that identifies the target address, the HWMP sequence number, and provides a reason for a rejection.

When a station has a path to the target destination, the station accepts the frame and replies with a frame containing a Path Reply (PREP) element. The PREP element is contained in a HWMP Mesh Path Selection frame, with only element 4 set. The PREP element is 31 to 37 byte long, and contains several subfields of importance (7.3.2.114):

- The **target MAC address**, and the associated target HWMP sequence number. This is useful, as a PREQ may be used to discover several target MAC addresses. These 2 subfields identify the target and the message used to discover it.
- The **hop count to the target**: this critical element will allow the originator to know how far the target is, from the responding station standpoint.
- The **metric** to the target: this information will be combined with the hop count by the originator to determine a best path to the target.
- A **Time To Live** (TTL) field and a Life Time field, used for loop prevention, just like for the PREQ process. These 2 fields are set by the responding station, and changed by the stations on the path back to the originator.

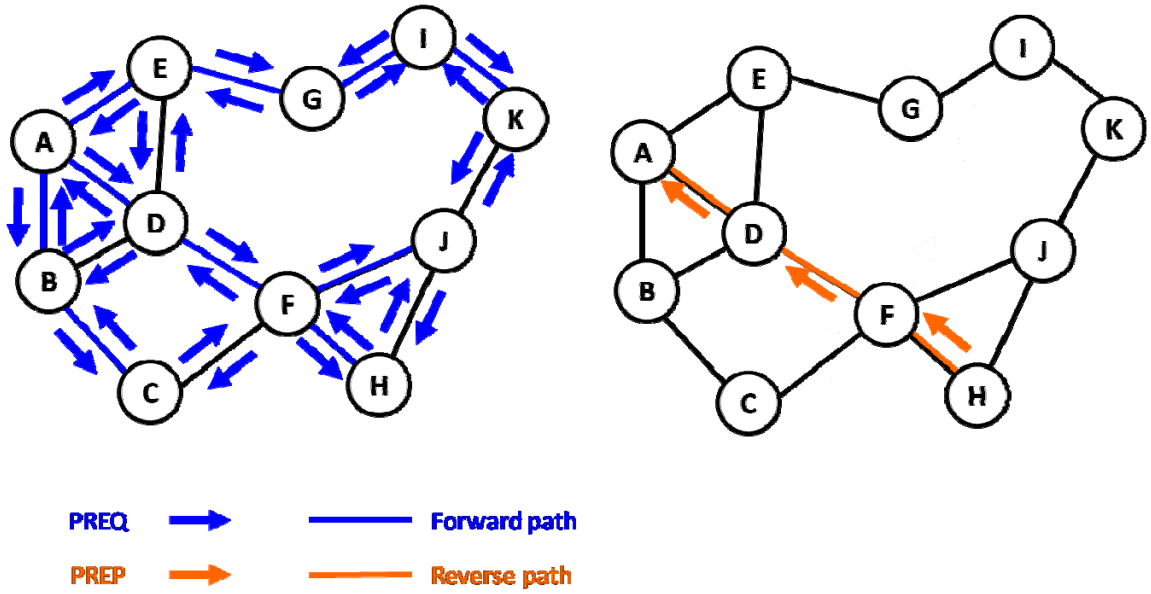
With this process, the originator obtains the information it needs to determine the best path to its destination. Using an example, shown in Figure 5, will make the concept clearer. Suppose station A in the figure below is trying to discover a path to station H. Station A sends a PREQ message that is received by the mesh stations in range of station A, namely stations B, D, and E. All stations examine the PREQ and decide to accept it. As A is specifically trying to discover H (H is the Target address), B, D, and E cannot reply with a PREP. Only the target can reply. B, D, and E would reply if the target was a broadcast address, used when a station tries to discover its neighbors. B, D, and E have to forward the PREQ, after decrementing the TTL and life time, and changing Hop Count and Metric to the Originator fields.

B, D, and E forward the PREQ through their radio; C, F, and G receive the PREQ. In this example, A is also going to receive the PREQ forwarded by B, D, and E. D is also in range of E and B, and will receive the PREQ forwarded by B and E. From the HWMP sequence number, the TTL and Life Time values, A and D will know that this was the previous PREQ that bounced through another station, and will discard the message. Stations receiving such duplicates will discard the PREQ message the same way, and will forward only if the HWMP sequence number is new, or if the received PREQ has a lower TTL and Life Time than the previously received PREQ for the same HWMP sequence number.

Following the same process, E will send the message, which will be received and forwarded by G, I, K, J, and finally H. At the same time, the PREQ will also be received and forwarded by D, C, J, and H. In the chain, F received the PREQ three times, once from D, once from J and once from C. Suppose F received the PREQ from C first. F forwarded that PREQ. When F received the PREQ from D, the new frame contains the same sequence number but a better hop count to the originator. (F-D-A instead of F-C-B-A). F learns that there is a better path to A and forwards the second frame. If the PREQ from D had been received before the PREQ from C, F would have discarded the second PREQ with a higher hop count to A.

In the end, H receives the PREQ. H is the Target and answers the PREQ with a PREP. The PREP is sent back along the best path back to the originator. If H received the PREQ from J first, H may answer to J, but eventually, as H receives a “better PREQ” (with lower metrics and hop counts back to A), the PREP information will be forwarded through F, D, and A, and A will learn the best path to H.

Figure 5: Using PREQ and PREP for mesh path selection



A station on the path (for example D) will remember that it is on the best path from A to H (identified by the components of the PREQ), and that the next hop to H is F, and the precursor (next hop back to A) is A itself.

802.11s was built for the MBSS to survive the inherent instability of 802.11 links. If a path gets disrupted (a station on the path between an originator and a target loses its path to the target), a PERR is generated at link break and propagated towards the originator, allowing the originator to generate a new path to the target.

This discovery can be done dynamically. In most cases though, the discovery is proactive (both modes are allowed). When a mesh station starts, it peers with its neighbors and immediately starts broadcasting PREQs using the broadcast address as the target MAC address. All stations will answer the PREQ (as the PREQ target is broadcast), allowing the mesh station to discover its MBSS. This process can be intensive if all stations try to discover all possible best paths to all other stations. It is a strength of 802.11s: you do not need to create any hierarchy for the MBSS and each mesh station can discover all the others dynamically. But if this mode is too noisy for your implementation, you can also organize your MBSS by designating some stations as **Root Mesh APs**. These are commonly the mesh gates/portals, but you can configure any mesh station as a root. This configuration is common because in most cases, mesh stations will try to find a way to forward frames to the wired network, or to a station of specific importance. Configuring Root Mesh APs allows you to organize the mesh cloud.

A station configured as a Root Mesh AP starts sending **Root Announcement messages (RANN).** The RANN element is transmitted in an HWMP Mesh Path Selection frame, is 21 byte long, and contains several subfields of importance (7.3.2.112):

- The **root mesh AP MAC address**
- A bit indicating if the root mesh AP is a **gate**
- A **HWMP sequence number**, to uniquely identify this RANN.

- An **Interval field**, indicating (in TUs) how often the mesh root AP is sending the RANN message.
- A **Time To Live** (TTL) value, that will be decremented by each station transmitting the RANN. The TTL is used for loop prevention.
- A **Hop Count** field and a Metric field, modified by the stations forwarding the RANN so that each station knows the distance to the root mesh AP.

The root mesh STA periodically propagates a RANN element into the network. The information contained in the RANN is used to disseminate path metrics for other stations to reach the root mesh STA, but reception of a RANN in itself does not "establish a path". *Upon reception of a RANN, each mesh STA that has to create or refresh a path to the root mesh STA sends an individually addressed PREQ to the root mesh STA via the mesh STA from which it received the RANN. The root mesh STA sends a PREP in response to each PREQ. The individually addressed PREQ creates the reverse path from the root mesh STA to the originator mesh STA, while the PREP creates the forward path from the mesh STA to the root mesh STA* (11C.9.4.3).

Notice that the RANN mentions if the root mesh AP is a gate. A gate that is not a root mesh AP would also inform the MBSS, using a **Gate Announcement frame (GANN).** This process is similar to the RANN in its logic and behavior, announcing the gate instead of the root.

The Mesh Metric

2 keys elements are present in all of the aforementioned messages and are used for best path determination: hop count and metric.

Hop Count simply counts the number of stations between the local station and the target destination. This element in itself is not enough to determine a best path. Just like for wired network a 2-hop fast link might be a better choice than a one-hop slow link. The metric is used to complement the hop count. With 802.11s, the metric is a combination of data rate and bit error rate. Suppose 2 mesh stations, A and B, communicating at 1 Mbps (Annex Y.5). Suppose that the frame that is used to sample the metric is 1024-bytes long (8192 bits). The metric determination will first determine the time taken to transmit that sample frame: 144 µs to send the PLCP preamble and 48 µs for the PLCP header. The payload is 8192 bits, taking8192 µs to be transmitted at 1 Mbps. The data transmission time is therefore 8416 µs. This value is then converted into units of 0.01 TU (10.24 µs). This determines the link metric, in this case 822 (8416 / 10.24 = 821.75, rounded to 822). This calculation implies that the link quality is perfect, and that no loss occurs. In reality, losses will occur, and the bit error rate (the percentage of frames that do not reach their destination) has to be factored into the metric. If the frame error rate to the neighbor is 0%, the metric is 822. If the frame error rate is 80%, the metric is 4110 (i.e., 821.75/(1–0.8), rounded).

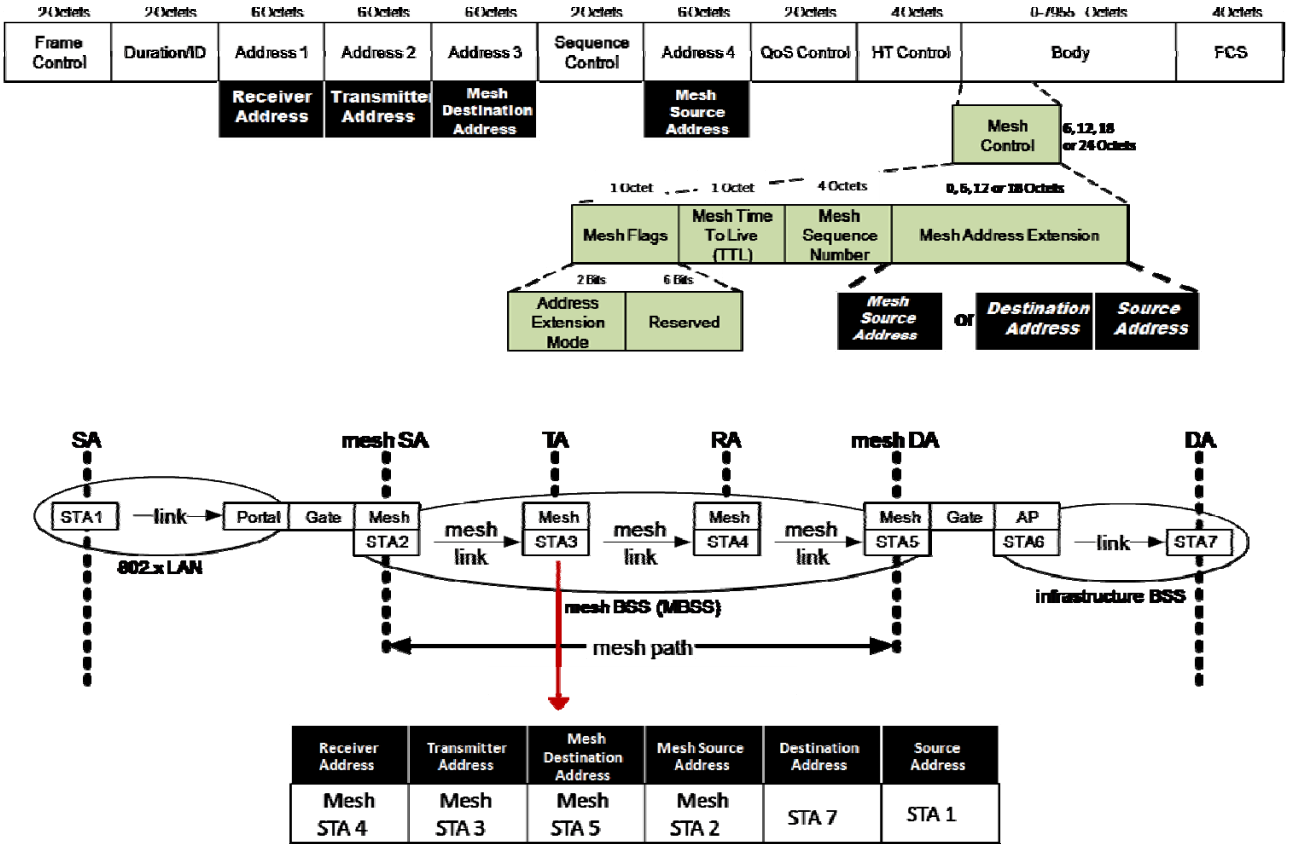
When determining the path to the target H above, each station adds its own metric before forwarding the PREP. F adds F-H metrics when forwarding the PREP to D. D adds D-F metric when forwarding the PREP to A. This way, the metric seen by A is the total of all links metrics to H. The metric described above is the default mechanism for best path determination in 802.11s, called **Airtime link metric**. The presence of hop count allows vendors to combine both hop count and metrics in a proprietary best path determination.

A new frame format: up to 6 MAC addresses

Once a mesh path has been established, mesh stations can start forwarding frames. The mesh frame format is slightly different from the standard 802.11 frame format, because a station needs to indicate more parameters. The main difficulty is that the original source of the frame and the final destination of the frame may be non-mesh stations. In order to be able to mention all addresses, the mesh frame contains 6 addresses: up to 4 addresses in the standard 802.11 header, and 1 or 2 addresses in a specific Mesh control field positioned after the HT control field. Non-mesh stations see that Mesh control field as part of the frame body:

- Address 1 is the Receiver Address (RA)
- Address 2 is the Transmitter Address (TA)
- Address 3 is the *mesh* destination address, the target mesh station (which may or may not be the frame's ultimate destination)
- Address 4 is the *mesh* source address, the mesh originator (which may or may not be the frame's ultimate source). When address 4 is not used (e.g. management frames) the mesh source address can also be inside the Mesh Address Extension field.
- Inside the Mesh Address Extension subfield of the Mesh Control field, address 5 is the final destination and address 6 is the original source. One or both of these addresses may be identical to one or 2 of the first 4 addresses if the original source, the final destination, or both are mesh stations. When address 4 is not used, the Mesh Address Extension field can also host the mesh source address.

Figure 6: Addressing and the mesh frame format



Taking care of Power Management concerns, and using power management to influence best path selection

An important element that 802.11s took into account is power save. Some mesh stations may be mobile stations operating on battery. Some non-mobile stations may be located in areas where power is not available all day long, and may also have to rely on battery. The ability to use power save for battery conservation is a key element for the longevity of the MBSS. At the same time, a mechanism must be put in place to avoid lost frames or MBSS disruption when a mesh station is dozing.

The 802.11s amendment describes three states related to power consumption for mesh stations:

- **Active Mode**: in this mode, the mesh station is available at any time, to participate in data forwarding, path discovery, and MBSS management functions. *The mesh station operates in the 802.11 standard Awake mode* (11C.13.8.3).
- **Light Sleep Mode**: in this mode, the mesh station tries to conserve battery while still performing some MBSS functions. The station alternates between

Awake and Doze states. This mode is close to the APSD mode in the 802.11 standard. The station in light sleep mode can doze, then awaken to receive the Beacon frame from the peer mesh station. *The mesh station can then return to the Doze state after the beacon reception, if the peer mesh station did not indicate buffered individually addressed or group addressed frames. If an indication of buffered individually addressed frames is received, the light sleep mode mesh station sends a peer trigger frame to receive the buffered traffic* (11C.13.8.4).

- **Deep Sleep Mode**: in this mode, *the station does not monitor its peer mesh stations* (11C.13.8.5). The sleeping station still has to awaken at a regular interval to send its own messages (beacons for example). It then has to stay awake a little bit longer, to give an opportunity to other stations to send a message to the local station, before going back to light or deep sleep.

The mesh station power state is communicated to its neighbors. The mesh station uses a combination of the Power Management field in the Frame Control field and the Mesh Power Save Level subfield in the QoS Control field contained in Mesh Data frames to indicate the station mesh power mode (11C.31):

Table 1: Mesh power modes and fields

Mesh Power Mode	Power Management field	Mesh Power Save Level subfield
Active	0	0
Light Sleep	1	0
Deep Sleep	1	1

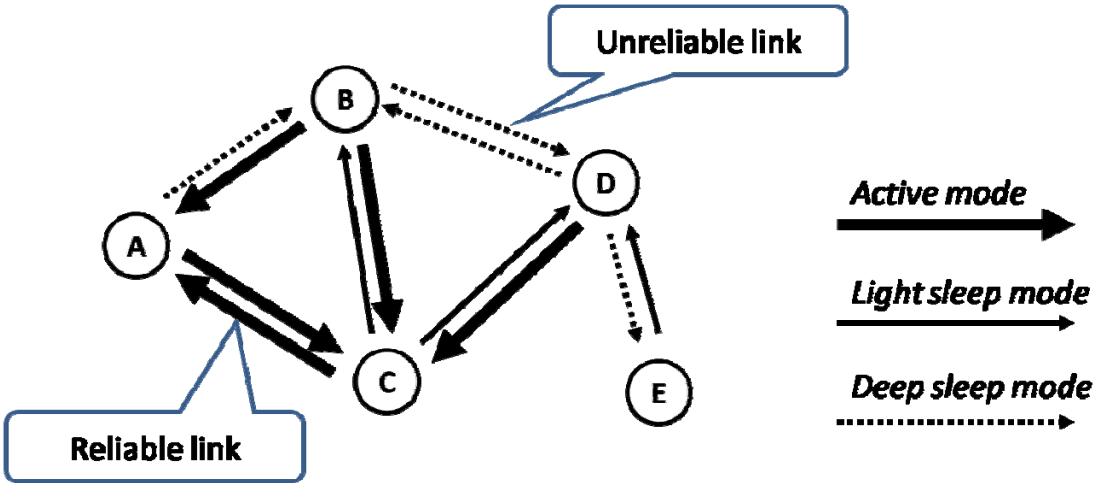
A mesh station beacon and probe response also contains a specific information element called the **Mesh Awake Window**, which specifies the duration of the mesh awake window in TUs. A mesh station can send a frame to a peer mesh station that is in light sleep mode or deep sleep mode for the corresponding mesh peering only during the mesh awake window of this peer mesh station.

This is the general principle. A very interesting aspect of this power management mechanism is that it can be used globally (non-peer mesh power mode) or on a per link (per mesh neighbor) basis (peer specific mesh power mode) for best path management. The mesh station uses the Power Management field in the Frame Control field to signal its non-peer mesh power mode, and the Mesh Power Save Level subfield to indicate additional state information for individual peers. Light Sleep and Deep Sleep are both seen as “Sleep” for non-peer stations. But as communications are peer-to-peer, and as each station is responsible for its own state announcement, a mesh station can decide to change its power status for a given neighbor. Suppose a station A has two paths to a station D, one path through B and one path through C. Suppose that the path through B is seen as less reliable than the path through C (for example because the loss rate, and therefore the metric, is changing all the time). Station A can decide to stay in Active mode with C, but announce to B a deep sleep mode, forcing B to buffer its frames for A. B may still be in active mode toward

A, which means that B is in awake state and available for any management or data frame that A would need to send.

The same logic can apply to any other link. In Figure 7, D and B are in deep sleep mode, which means that they decided not to use their link. D is in active mode with C, which means that D is available for any frame coming from C. At the same time, C announced a Light Sleep mode to D and B, which means that C will wake up for B and C beacons to check if they have any buffered traffic.

Figure 7: Example mesh power save scenario



In extreme scenarios, a mesh station can declare itself in deep sleep mode for all its neighbors, giving the station the control to decide when to send and when to receive frames. This would allow a station to never forward other stations traffic, and be active in the MBSS only for its own traffic. 802.11s is very flexible on the conditions upon which a mesh station can decide to change its power mode for one or several neighbors, leaving a great liberty in vendor implementations to adapt the power mode to local architectural conditions.

Managing Collisions and Traffic Priorities

In indoor standard networks, collisions are limited to a few cells, and are already a critical issue in dense deployments. This issue is worsened for mesh networks. Mesh access points are built for redundant paths. This means that a single mesh station is typically in range of several other mesh stations. Mesh access points typically employ powerful amplifiers and antennas. Access points in communication with one another must be on the same channel. These characteristics mean that QoS and collisions are two major concerns in mesh networks. Therefore, a first concern is to avoid collisions. This avoidance is difficult to achieve because frames will be transmitted over the wireless medium for several hops, and each station has little to no visibility of the medium state beyond its own neighbors. However, neighbors can communicate, and 802.11s describes several mechanisms by which neighbors can communicate medium-related information and work together to reduce collisions and congestion.

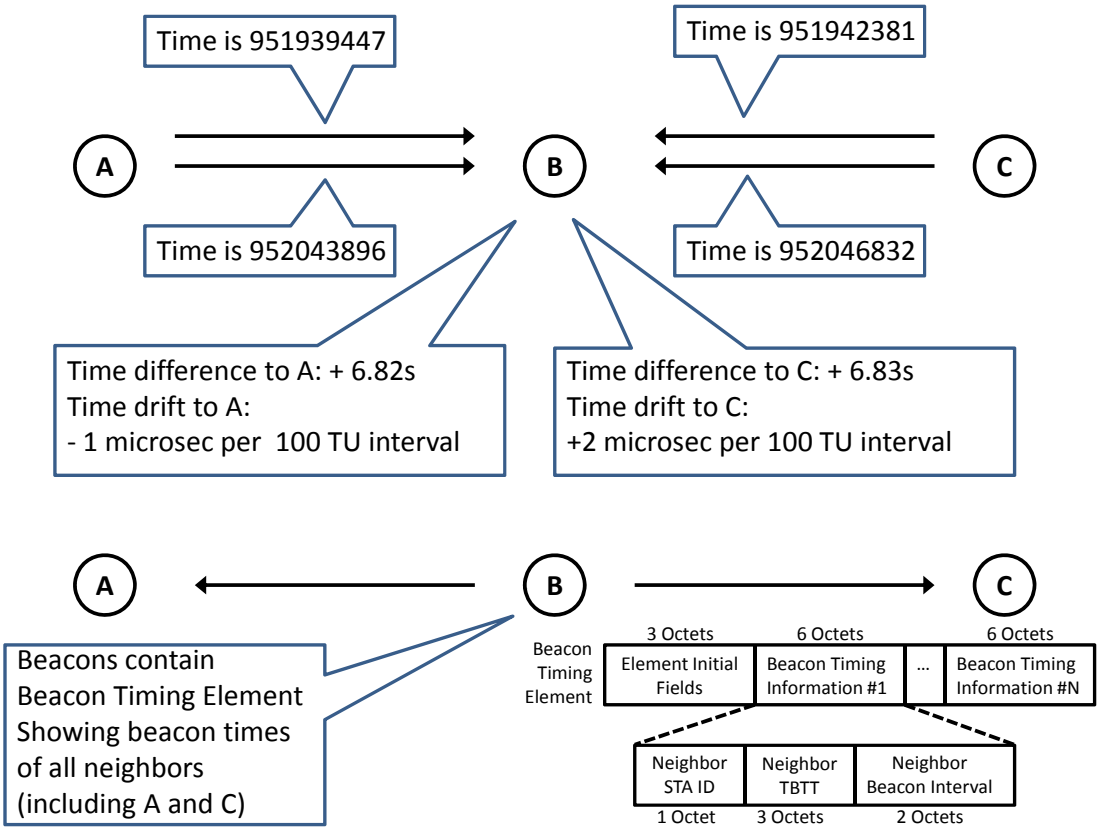
Synchronization mesh stations time

A prerequisite for proper communication is synchronization. If a mesh station cannot determine when a neighbor's frame was sent, then communicating efficiently about the present state of the medium becomes challenging. To solve this issue, 802.11s allows for multiple synchronization methods between neighbors. Vendors can implement their own method, but 802.11s describes and recommends a simple mechanism called **Neighbor Offset Synchronization**. This mechanism relies on the fact that beacons and probe response frames contain a timestamp value. Mesh stations embed a series of functions called Neighbor offset, by which each station reads and analyzes the timestamp value when receiving beacons and probe responses from a neighbor. The timestamp expresses the local time at the emitting station. This information is already useful in itself to determine the time difference between the local station and the emitting neighbor. But in 802.11s, the logic is brought further and the timestamp is also used to calculate the **neighbor clock drift**, by comparing the received timestamp to the expected timestamp.

This is easily understood with an example, which is illustrated in Figure 8. Suppose a mesh station neighbor A's first beacon mentions the timestamp 951939447. The local mesh station B can compare this timestamp to its own timestamp and deduce the time difference between neighbors. For example, the local timestamp could be 945112870 at the time of reception, and station B can then calculate that the time difference between the local time and the neighbor A time is (951939447 – 945112870) 6826577. This number may be hard to read for a human eye, but it simply shows that there is approximately a 6.83 second difference between both mesh stations' clocks.

Things get more interesting when station A sends its next beacon. Suppose A's beacon timestamp then shows 952043896, and station B timestamp at the time of reception was 945217320. The difference is now 6826576, 1 microsecond less than for the previous beacon difference. This means that both stations' clocks do not run at the same speed. With this information, station B can translate in its own local clock the time information sent by station A, not only in terms of absolute time, but also in terms of duration. This allows station B to adjust the time values received from station A and translate what they mean for its own time.

Figure 8: Synchronization with timestamps



Anti-collision mechanism for beacon frames, MBCA (11C.12.4)

This clock synchronization can be used for many purposes. One of them is to avoid beacon collisions, using a process called **Mesh Beacon Collision Avoidance (MBCA).** These collisions are especially likely to occur in mesh networks because of hidden node issues. In Figure 8, A and B as well as B and C are in range of each other, but A and C may not be in range. A and C may be transmitting frames to B at the same time. These frames would collide when reaching B. Beacon frames are very likely to collide because they are unsolicited frames, sent very often. To mitigate this risk of collision, 802.11s mesh stations add a new information element to their own beacon, called the **Beacon Timing Element**. For each detected neighbor, this element contains the neighbor ID, the target beacon transmission time (TBTT), and the beacon interval for this neighbor. The TBTT shows when the next beacon is due for this neighbor and the beacon interval—the amount of time (in TUs) between beacons for this neighbor.

Each mesh station must take this information in account for its own beacons. For example, station C will receive beacons coming from station B and will therefore learn about station A's

TBTT and beacon interval. As those values are expressed in station B's time, and as C knows its time difference and drift with B, C can determine exactly, in its own time reference, when station A's beacon is due. If station C's beacon timing is such that they would collide with station A's beacons, station C should change its TBTT.

To avoid disruption to the MBSS, station C will start by setting its frames' Mesh Information Element **TBTT Adjusting bit** to 1, to inform its neighbor that it is aware that its beacons are colliding, and that it is about to change its TBTT. The neighbors will then know that station C's TBTT and beacon intervals are temporarily unreliable. Station C would then choose another TBTT that would not create collisions and resume its beacon operation, setting the TBTT Adjusting bit in its frames' Mesh Information Element back to 0 (11C.12.4.4.3).
This adjustment process can be done independently by a station detecting this collision risk (11C.12.4.4.1 **Self-determined TBTT adjustment**), but can also be requested by a neighboring station (11C.12.4.4.2 **Requested TBTT Adjustment**). In our example, station B may ask station C to adjust its beacons to avoid collisions with station A's beacons. In this process, station B would first give station C a chance to take into account the Beacon Timing Element Information and adjust by itself. But if collisions continue (802.11s is open on how long B should wait before taking action), B can send a **TBTT Adjustment Request** action frame (7.4.15.11) to the neighbor mesh station of which the TBTT comes last at a particular collision timing in order to request this neighbor mesh station to adjust its TBTT (11C.12.4.4.2). In our example, if B starts receiving beacons from A just before starting to receive beacons from C, B would ask C to adjust its TBTT. C would answer with a **TBTT Adjustment Response** action frame (7.4.15.12) to acknowledge the request (or justify why the request is rejected), and would then adjust its TBTT.

Prioritizing frames at the scale of the entire mesh cloud

MBCA only applies to beacons. 802.11s also implements anti-collision and anti-congestion mechanisms for other frames. The 802.11 standard itself already implements a mechanism, called Enhanced Distributed Channel Access (EDCA), that implements prioritization mechanisms for stations using Distributed Coordination Function (DCF), the standard contention-based medium access function. In a MBSS, the medium access function is called Mesh Coordination Function (MCF). It is different from DCF because it implements the additional functions detailed in the previous pages. MCF also implements EDCA for contention-based channel access (9.9.1), and another mechanism for contention-free channel access, called MCF controlled channel access (MCCA, 9.9a.3). Contention-free? If you have been working with 802.11 for a long time, "contention-free" may remind you of PCF or HCCA, and you may think that this has very little chances of being actually implemented. However, a major difference between mesh networks and other 802.11 networks is that the mesh backhaul on a given mesh deployment will most likely be controlled by one single vendor. This limits the risk of incompatibilities, and increases the possibilities for that vendor to fine tune MCCA to offer prioritization with the best performance.

MCCA does not have to be deployed at the scale of the entire MBSS. Each station can enable its support for MCCA and show this support by setting to 1 the **MCCA Enabled bit**, which is in the Mesh Capability Subfield of the Mesh Configuration Element present in beacons and probe responses. Another station may support MCCA but not implement it (the Mesh Capability subfield also includes a **MCCA Supported bit**). In that case, the station can participate in the MCCA mechanism, but cannot initiate any MCCA reservation.

MCCA is close to HCCA in its principles. MCCA-enabled mesh STAs use management frames to make reservations for transmissions. To initiate medium reservation, a mesh station transmits an **MCCA Setup Request** frame, and becomes what is called the **mesh coordination function controlled channel access opportunity (MCCAOP) owner** of the MCCAOP reservation.

The MCCA Setup Request is an action frame that can use a unicast or group destination address. The frame contains a **MCCAOP Setup Request element** (7.3.2.106) that specifies 3 items needed to identify the reservation. To understand these items, you have to be aware that MCCA specifies times in terms of DTIM intervals. The interval between 2 DTIMs represents one time period. Inside this time period, a station may want to reserve several units of time. The first unit would start at a given time after the DTIM, and each subsequently reserved units would be separated by a specific time interval:

- The **MCCAOP Offset field** (3 octets) specifies how long after the DTIM should the first reservation start (in units of 32 microseconds).
- The **MCCAOP duration field** (1 octet) specifies how long each reservation should last, in units of 32 microseconds.
- The **MCCAOP Periodicity** specifies how many MCCAOPs are scheduled for each DTIM interval.

The receiver(s) of the MCCA Setup Request frame are called the **MCCAOP responders**. Each MCCAOP responder should return a **MCCA Setup Reply** frame that accepts or reject the MCCAOP. The reasons for the rejection might be that the MCCAOP conflicts with another reservation, that the MCCAOP responder already has too many MCCAOPs to be able to accept and track any more reservations, or that the responder does not have any time left available for the MCCAOP. This last reason may commonly occur when the vendor implements a maximum percentage of the mesh station bandwidth that can be allocated to reservations.

If the MCCAOP is accepted, the MCCAOP owner and the MCCAOP responders advertise this MCCAOP reservation to their neighbors via an **MCCAOP advertisement information element** present in beacons and probe responses and specific **MCCA Advertisement** frames. These frames are action frames sent to the broadcast address as soon as a change occurs in a MCCAOP. The MCCAOP advertisement information element contains details about the various MCCAOP on the emitting station. Any MCCA enabled neighbor mesh station in range, that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, will not initiate a transmission during these reserved time periods.

An important aspect of MCCA is its distributed aspect. Each station can reserve channel time and inform its neighbors of this desire. Neighbors can in turn reserve channel time for the same traffic, resulting in an end to end reservation across the MBSS if all mesh stations are MCCA-enabled. However, if one station does not implement MCCA, that station may become a bottleneck and cancel the benefits of MCCAOP reservations. Consistency in MCCA support and configuration across the MBSS is a key factor to evaluate the efficiency of this mechanism.

Summary

802.11s was originally designed to address concerns specific to mesh networks. The complexity of those networks generated many additional considerations that delayed the final amendment for several years. The final amendment reuses standard 802.11 mechanisms for mesh stations to discover one another. Once this discovery has occurred, mesh stations can establish peer relationships with their neighbors. This process is non-hierarchical, distributed, and yet very secure. Even if mesh peering must rely on shared keys, the authentication process is more secure than the PSK authentication mechanisms in the current 802.11 standard, and could be used as a replacement to WPA or WPA2 PSK for client access in future implementations. Once peering has occurred, mesh stations use an algorithm based on link data rate and packet error rate to determine best paths to any destinations, inside or beyond the mesh cloud. Mesh stations are informed of an exit point to the DS, and mesh network designers can organize the cloud by configuring root mesh stations. Power management is taken into account, and can be used for path optimization.

QoS is also addressed, by first allowing stations to synchronize their clock with the clock of their neighbors then use time information to avoid beacon collisions using MBCA, and reserve backhaul bandwidth for critical traffic with MCCA. Once 802.11s has been integrated into the 802.11 standard, MBCA could very easily be adapted to allow access points in dense indoor environments to avoid co-channel interference due to beacon collisions. Similarly, APs could coordinate staggered beacons to avoid interference resulting from delivery of buffered traffic when power save is in use.

During all the years while 802.11s was discussed, vendors have implemented specific mesh solutions without taking into account the features, terms, and considerations described in the 802.11s successive drafts. Today, most vendors' mesh solutions are strongly established on a given set of features supported by vendor-specific functionality and terminology, and one may wonder if any vendor has any interest in implementing 802.11s. Yet a powerful advantage of 802.11s is to set a standard, from which features can be compared. Ratifying 802.11s does not force any vendor to implement the standard protocol, but allows a better and more thorough comparison between vendors' features by using a common set of names and functionalities. This common point of reference typically offers the possibility to contrast solutions, and often influences vendors to implement the aspects of the standard that their proprietary solution was missing, improving the efficiency and security of all mesh solutions.

About the Author

Jerome Henry is a wireless expert at Fast Lane. Before joining Fast Lane in 2006, he was consulting and teaching Heterogeneous Networks and Wireless Integration with the European Airespace team, which was later acquired by Cisco to become their main wireless solution. He is a certified wireless networking expert (CWNE #45), CCIE Wireless (#24750), CCNP Wireless, and has developed several Cisco courses focusing on wireless topics (IUWNE, IUWMS, IUWVN, LBS, CWMN lab guide, etc.) and authored several Wireless books (IUWMS, CUWSS Quick Reference, etc.). Jerome is also an IEEE 802.11 member and follows very closely the work of the various 802.11 task groups. With more than 5000 hours in the classroom, Jerome was awarded the IT Training Award best Instructor silver medal in 2009. He is based in Cary, NC.

About the Editor

Marcus Burton is the Director of Product Development at CWNP. Marcus has authored or co-authored numerous WLAN exams, course guides, whitepapers, and articles. In addition, he has edited several WLAN books including CWTS, CWSP, and CWAP, and co-authored the CWDP study guide. At CWNP, Marcus actively participates with many Wi-Fi vendors in product testing and review, and has in-depth knowledge of vendor products as well as 802.11 protocols. Marcus is CWNE #78.

About CWNP

CWNP is the recognized industry standard for enterprise Wi-Fi training and certification. CWNP is the only vendor neutral wireless LAN certification program in the industry, covering the full range of technologies underlying all enterprise WLAN products. CWNP offers four levels of enterprise WLAN certification, from novice to expert and prepares IT professionals to design, install, manage, and troubleshoot wireless LAN infrastructure and applications regardless of the vendor solution utilized. Professionals in more than 130 countries have achieved CWNP certifications, enabling them to make wireless LANs more cost-effective, reliable, and secure. CWNP is a privately-held corporation based in Atlanta, GA. For more information about CWNP, visit www.cwnp.com.