Network Working Group
Internet-Draft
Expires: December 13, 2007

P. Calhoun, Editor
Cisco Systems, Inc.
M. Montemurro, Editor
Research In Motion
D. Stanley, Editor
Aruba Networks
June 11, 2007

CAPWAP Protocol Specification
draft-ietf-capwap-protocol-specification-07

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on December 13, 2007.

Copyright Notice

Abstract

This specification defines the Control And Provisioning of Wireless Access Points (CAPWAP) Protocol. The CAPWAP protocol meets the IETF CAPWAP working group protocol requirements. The CAPWAP protocol is designed to be flexible, allowing it to be used for a variety of wireless technologies. This document describes the base CAPWAP protocol. The CAPWAP protocol binding which defines extensions for use with the IEEE 802.11 wireless LAN protocol is available in [12]. Extensions are expected to be defined to enable use of the CAPWAP protocol with additional wireless technologies.

1. Introduction

This document describes the CAPWAP Protocol, a standard, interoperable protocol which enables an Access Controller (AC) to manage a collection of Wireless Termination Points (WTPs). The CAPWAP protocol is defined to be independent of layer 2 technology.

The emergence of centralized IEEE 802.11 Wireless Local Area Network (WLAN) architectures, in which simple IEEE 802.11 WTPs are managed by an Access Controller (AC) suggested that a standards based, interoperable protocol could radically simplify the deployment and management of wireless networks. WTPs require a set of dynamic management and control functions related to their primary task of connecting the wireless and wired mediums. Traditional protocols for managing WTPs are either manual static configuration via HTTP, proprietary Layer 2 specific or non-existent (if the WTPs are self-contained). An IEEE 802.11 binding is defined in [12] to support use of the CAPWAP protocol with IEEE 802.11 WLAN networks.

CAPWAP assumes a network configuration consisting of multiple WTPs communicating via the Internet Protocol (IP) to an AC. WTPs are viewed as remote RF interfaces controlled by the AC. The CAPWAP protocol supports two modes of operation: Split and Local MAC. In Split MAC mode all L2 wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between the AC and the WTP. As shown in Figure 1, the wireless frames received from a mobile device, which is referred to in this specification as a Station (STA), are directly encapsulated by the WTP and forwarded to the AC.

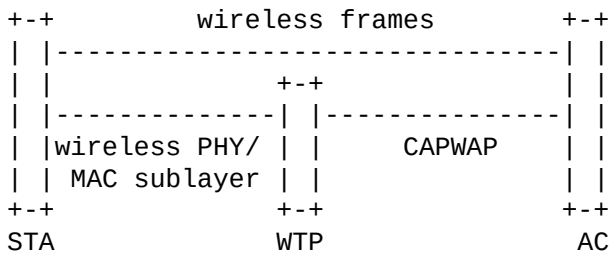


Figure 1: Representative CAPWAP Architecture for Split MAC

The Local MAC mode of operation allows for the data frames to be either locally bridged, or tunneled as 802.3 frames. The latter implies that the WTP performs the 802 bridging function. In either case the L2 wireless management frames are processed locally by the WTP, and then forwarded to the AC. Figure 2 shows the Local MAC mode, in which a station transmits a wireless frame which is encapsulated in an 802.3 frame and forwarded to the AC.

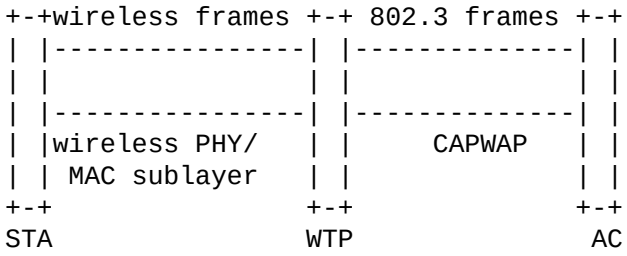


Figure 2: Representative CAPWAP Architecture for Local MAC

Provisioning WTPs with security credentials, and managing which WTPs are authorized to provide service are traditionally handled by proprietary solutions. Allowing these functions to be performed from a centralized AC in an interoperable fashion increases manageability and allows network operators to more tightly control their wireless network infrastructure.

1.1. Goals

The goals for the CAPWAP protocol are listed below:

1. To centralize the authentication and policy enforcement functions for a wireless network. The AC may also provide centralized bridging, forwarding, and encryption of user traffic. Centralization of these functions will enable reduced cost and higher efficiency by applying the capabilities of network processing silicon to the wireless network, as in wired LANs.

2. To enable shifting of the higher level protocol processing from the WTP. This leaves the time critical applications of wireless control and access in the WTP, making efficient use of the computing power available in WTPs which are the subject to severe cost pressure.

3. To provide a generic encapsulation and transport mechanism, enabling the CAPWAP protocol to be applied to many access point types in the future, via a specific wireless binding.

The CAPWAP protocol concerns itself solely with the interface between the WTP and the AC. Inter-AC and station-to AC-communication are strictly outside the scope of this document.

1.2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

1.3. Contributing Authors

This section lists and acknowledges the authors of significant text and concepts included in this specification.

The CAPWAP Working Group selected the Lightweight Access Point Protocol (LWAPP) [add reference, when available] to be used as the basis of the CAPWAP protocol specification. The following people are authors of the LWAPP document:

Bob O'Hara, Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134
Phone: +1 408-853-5513, Email: bob.ohara@cisco.com

Pat Calhoun, Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134
Phone: +1 408-853-5269, Email: pcalhoun@cisco.com

Rohit Suri, Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134
Phone: +1 408-853-5548, Email: rsuri@cisco.com

Nancy Cam Winget, Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134
Phone: +1 408-853-0532, Email: ncamwing@cisco.com

Scott Kelly, Aruba Networks
1322 Crossman Ave, Sunnyvale, CA 94089
Phone: +1 408-754-8408, Email: skelly@arubanetworks.com

Michael Glenn Williams, Nokia, Inc.
313 Fairchild Drive, Mountain View, CA 94043
Phone: +1 650-714-7758, Email: Michael.G.Williams@Nokia.com

Sue Hares, Nexthop Technologies, Inc.
825 Victors Way, Suite 100, Ann Arbor, MI 48108
Phone: +1 734 222 1610, Email: shares@nexthop.com

DTLS is used as the security solution for the CAPWAP protocol. The following people are authors of significant DTLS-related text included in this document:

Scott Kelly, Aruba Networks
1322 Crossman Ave, Sunnyvale, CA 94089
Phone: +1 408-754-8408, Email: skelly@arubanetworks.com

Eric Rescorla, Network Resonance
2483 El Camino Real, #212,Palo Alto CA, 94303

Email: ekr@networkresonance.com

The concept of using DTLS to secure the CAPWAP protocol was part of the Secure Light Access Point Protocol (SLAPP) proposal [add reference when available]. The following people are authors of the SLAPP proposal:

Partha Narasimhan, Aruba Networks
1322 Crossman Ave, Sunnyvale, CA 94089
Phone: +1 408-480-4716, Email: partha@arubanetworks.com

Dan Harkins, Tropos Networks
555 Del Rey Avenue, Sunnyvale, CA, 95085
Phone: +1 408 470 7372, Email: dharkins@tropos.com

Subbu Ponnuswamy, Aruba Networks
1322 Crossman Ave, Sunnyvale, CA 94089
Phone: +1 408-754-1213, Email: subbu@arubanetworks.com

The following individuals contributed significant security related text to the draft:

T. Charles Clancy, Laboratory for Telecommunications Sciences,
8080 Greenmead Drive, College Park, MD 20740
Phone: +1 240-373-5069, Email: clancy@ltsnet.net

Scott Kelly, Aruba Networks
1322 Crossman Ave, Sunnyvale, CA 94089
Phone: +1 408-754-8408, Email: skelly@arubanetworks.com

1.4. Terminology

Access Controller (AC): The network entity that provides WTPs access to the network infrastructure in the data plane, control plane, management plane, or a combination therein.

CAPWAP Control Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC control port, WTP control port and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP control packets are sent and received.

CAPWAP Data Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC data port, WTP data port, and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP data packets are sent and received.

Station (STA): A device that contains an IEEE 802.11 conformant

medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

Wireless Termination Point (WTP): The physical or network entity that contains an RF antenna and wireless PHY to transmit and receive station traffic for wireless access networks.

This document uses additional terminology defined in [15].

2. Protocol Overview

The CAPWAP protocol is a generic protocol defining AC and WTP control and data plane communication via a CAPWAP protocol transport mechanism. CAPWAP control messages, and optionally CAPWAP data messages, are secured using Datagram Transport Layer Security (DTLS) [7]. DTLS is a standards-track IETF protocol based upon TLS. The underlying security-related protocol mechanisms of TLS have been successfully deployed for many years.

The CAPWAP protocol Transport layer carries two types of payload, CAPWAP Data messages and CAPWAP Control messages. CAPWAP Data messages encapsulate forwarded wireless frames. CAPWAP protocol Control messages are management messages exchanged between a WTP and an AC. The CAPWAP Data and Control packets are sent over separate UDP ports. Since both data and control packets can exceed the Maximum Transmission Unit (MTU) length, the payload of a CAPWAP data or control message can be fragmented. The fragmentation behavior is defined in Section 3.

The CAPWAP Protocol begins with a discovery phase. The WTPs send a Discovery Request message, causing any Access Controller (AC) receiving the message to respond with a Discovery Response message. From the Discovery Response messages received, a WTP selects an AC with which to establish a secure DTLS session. CAPWAP protocol messages will be fragmented to the maximum length discovered to be supported by the network.

Once the WTP and the AC have completed DTLS session establishment, a configuration exchange occurs in which both devices agree on version information. During this exchange the WTP may receive provisioning settings. The WTP is then enabled for operation.

When the WTP and AC have completed the version and provision exchange and the WTP is enabled, the CAPWAP protocol is used to encapsulate the wireless data frames sent between the WTP and AC. The CAPWAP protocol will fragment the L2 frames if the size of the encapsulated wireless user data (Data) or protocol control (Management) frames causes the resulting CAPWAP protocol packet to exceed the MTU supported between the WTP and AC. Fragmented CAPWAP packets are reassembled to reconstitute the original encapsulated payload.

The CAPWAP protocol provides for the delivery of commands from the AC to the WTP for the management of stations that are communicating with the WTP. This may include the creation of local data structures in the WTP for the stations and the collection of statistical information about the communication between the WTP and the stations. The CAPWAP protocol provides a mechanism for the AC to obtain

statistical information collected by the WTP.

The CAPWAP protocol provides for a keep alive feature that preserves the communication channel between the WTP and AC. If the AC fails to appear alive, the WTP will try to discover a new AC.

2.1. Wireless Binding Definition

The CAPWAP protocol is independent of a specific WTP radio technology. Elements of the CAPWAP protocol are designed to accommodate the specific needs of each wireless technology in a standard way. Implementation of the CAPWAP protocol for a particular wireless technology MUST follow the binding requirements defined for that technology.

When defining a binding for wireless technologies, the authors MUST include any necessary definitions for technology-specific messages and all technology-specific message elements for those messages. At a minimum, a binding MUST provide:

1 - The definition for a binding-specific Statistics message element, carried in the WTP Event Request message

2 - A message element carried in the Station Configuration Request message to configure station information on the WTP

3 - A WTP Radio Information message element carried in the Discovery, Primary Discovery and Join Request and Response messages, indicating the binding specific radio types supported at the WTP and AC.

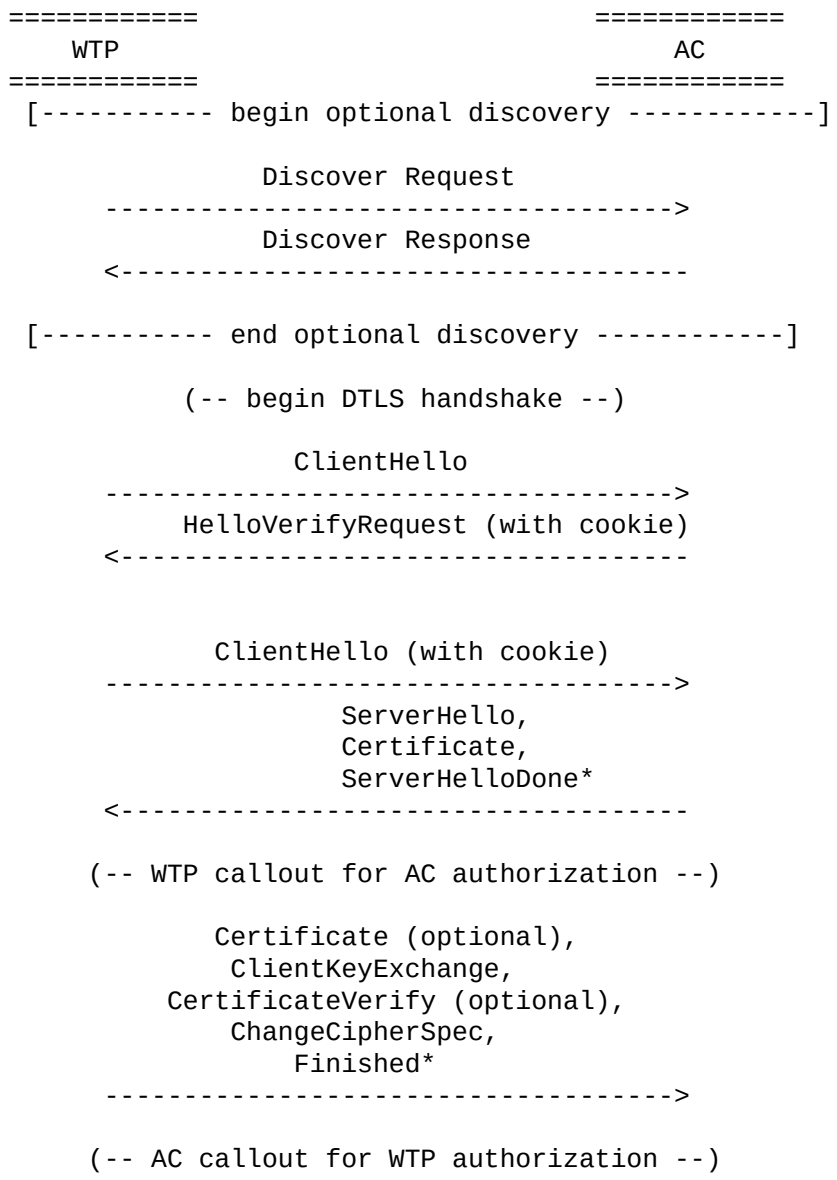
If technology specific message elements are required for any of the existing CAPWAP messages defined in this specification, they MUST also be defined in the technology binding document.

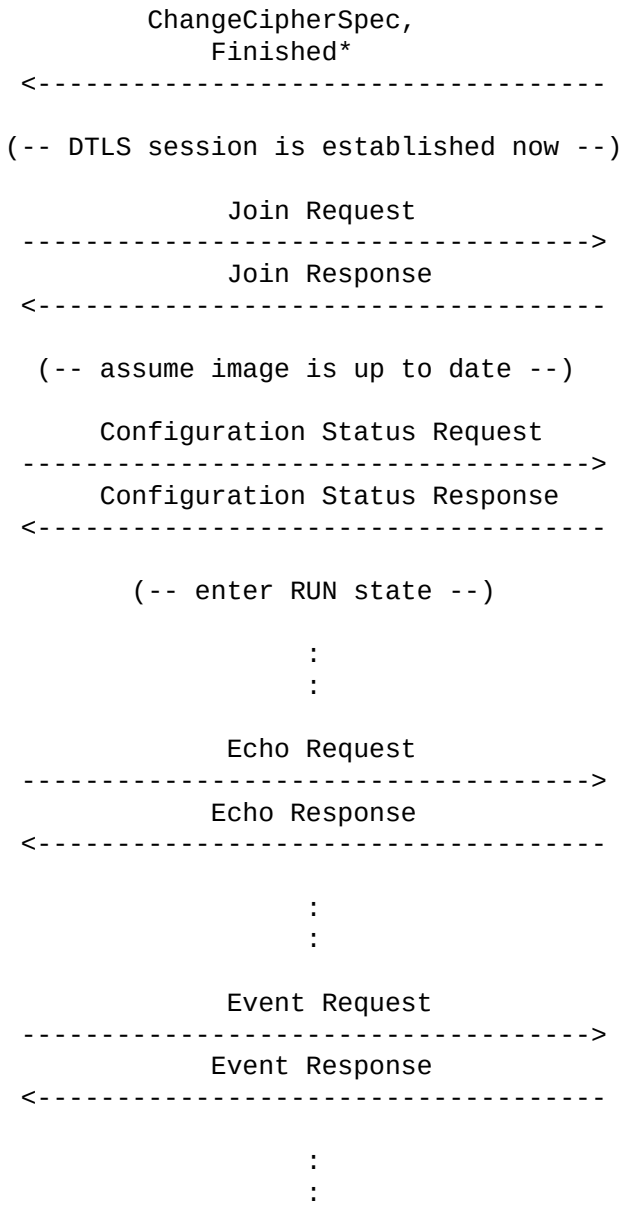
The naming of binding-specific message elements MUST begin with the name of the technology type, e.g., the binding for IEEE 802.11, provided in [12], begins with "IEEE 802.11".

The CAPWAP binding concept is also used in any future specifications that add functionality to either the base CAPWAP protocol specification, or any published CAPWAP binding specification. A separate WTP Radio Information message element MUST be created to properly advertise support for the specification. This mechanism allows for future protocol extensibility, while providing the necessary capabilities advertisement, through the WTP Radio Information message element, to ensure WTP/AC interoperability.

2.2. CAPWAP Session Establishment Overview

This section describes the session establishment process message exchanges in the ideal case. The annotated ladder diagram shows the AC on the right, the WTP on the left, and assumes the use of certificates for DTLS authentication. The CAPWAP Protocol State Machine is described in detail in Section 2.3. Note that DTLS allows certain messages to be aggregated into a single frame, which is denoted via an asterix in the following figure.





At the end of the illustrated CAPWAP message exchange, the AC and WTP are securely exchanging CAPWAP control messages. This is an idealized illustration, provided to clarify protocol operation. Section 2.3 provides a detailed description of the corresponding state machine.

2.3. CAPWAP State Machine Definition

The following state diagram represents the lifecycle of a WTP-AC session. Use of DTLS by the CAPWAP protocol results in the juxtaposition of two nominally separate yet tightly bound state machines. The DTLS and CAPWAP state machines are coupled through an API consisting of commands (see Section 2.3.2.1) and notifications (see Section 2.3.2.2). Certain transitions in the DTLS state machine are triggered by commands from the CAPWAP state machine, while certain transitions in the CAPWAP state machine are triggered by notifications from the DTLS state machine.

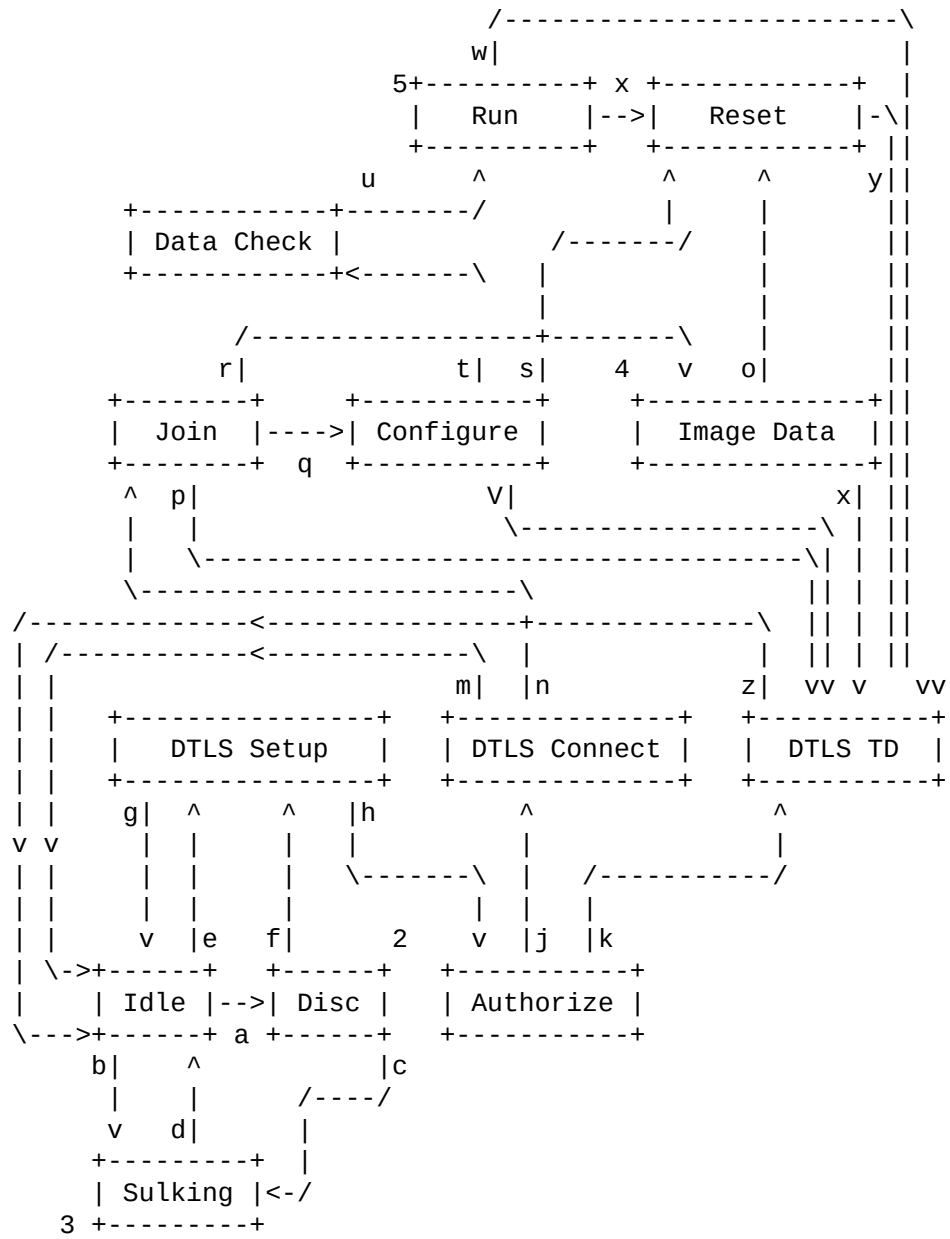


Figure 3: CAPWAP Integrated State Machine

The CAPWAP protocol state machine, depicted above, is used by both the AC and the WTP. In cases where states are not shared (i.e. not implemented in one or the other of the AC or WTP), this is explicitly called out in the transition descriptions below. For every state defined, only certain messages are permitted to be sent and received. The CAPWAP control messages definitions specify the state(s) in which

each message is valid.

Since the WTP only communicates with a single AC, it only has a single instance of the CAPWAP state machine. The AC has a separate instance of the CAPWAP state machine per WTP it is communicating with.

2.3.1. CAPWAP Protocol State Transitions

This section describes the various state transitions, and the events that cause them. This section does not discuss interactions between DTLS- and CAPWAP-specific states. Those interactions, and DTLS-specific states and transitions, are discussed in Section 2.3.2.

Idle to Discovery (a): This transition occurs once device initialization is complete.

WTP: The WTP enters the Discovery state prior to transmitting the first Discovery Request message (see Section 5.1). Upon entering this state, the WTP sets the DiscoveryInterval timer (see Section 4.7). The WTP resets the DiscoveryCount counter to zero (0) (see Section 4.8). The WTP also clears all information from ACs it may have received during a previous Discovery phase.

AC: The AC does not maintain state information for the WTP upon reception of the Discovery Request message, but it SHOULD respond with a Discovery Response message (see Section 5.2). This transition is a no-op for the AC.

Idle to Sulking (b): This transition occurs to force the WTP and AC to enter a quiet period to avoid repeatedly attempting to establish a connection.

WTP: The WTP enters this state when the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter reaches MaxFailedDTLSSessionRetry variable (see Section 4.8). Upon entering this state, the WTP MUST start the SilentInterval timer. While in the Sulking state, all received CAPWAP and DTLS protocol messages received MUST be ignored.

AC: The AC enters this state with the specific WTP when the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter reaches MaxFailedDTLSSessionRetry variable (see Section 4.8). Upon entering this state, the AC MUST start the SilentInterval timer. While in the Sulking state, all received CAPWAP and DTLS protocol messages received from the WTP MUST be ignored.

Discovery to Discovery (2): In the Discovery state, the WTP determines which AC to connect to.

WTP: This transition occurs when the DiscoveryInterval timer expires. If the WTP is configured with a list of ACs, it transmits a Discovery Request message to every AC from which it has not received a Discovery Response message. For every transition to this event, the WTP increments the DiscoveryCount counter. See Section 5.1 for more information on how the WTP knows the ACs to which it should transmit the Discovery Request messages. The WTP restarts the DiscoveryInterval timer whenever it transmits Discovery Request messages.

AC: This is a no-op.

Discovery to Sulking (c): This transition occurs on a WTP when Discovery or connectivity to the AC fails.

WTP: The WTP enters this state when the DiscoveryInterval timer expires or the DiscoveryCount variable is equal to the MaxDiscoveries variable (see Section 4.8). Upon entering this state, the WTP MUST start the SilentInterval timer. While in the Sulking state, all received CAPWAP protocol messages received MUST be ignored.

AC: This is a no-op.

Sulking to Idle (d): This transition occurs on a WTP when it must restart the discovery phase.

WTP: The WTP enters this state when the SilentInterval timer (see Section 4.7) expires. The FailedDTLSSessionCount, DiscoveryCount and FailedDTLSAuthFailCount counters are reset to zero.

AC: The AC enters this state when the SilentInterval timer (see Section 4.7) expires. The FailedDTLSSessionCount, DiscoveryCount and FailedDTLSAuthFailCount counters are reset to zero.

Sulking to Sulking (3): The Sulking state provides the silent period, minimizing the possibility for Denial of Service (DoS) attacks.

WTP: All packets received from the AC while in the sulking state are ignored.

AC: All packets receive from the WTP while in the sulking state are ignored.

Idle to DTLS Setup (e): This transition occurs to establish a secure DTLS session with the peer.

WTP: The WTP initiates this transition by invoking the DTLSStart command, which starts the DTLS session establishment with the chosen AC. When the discovery phase is bypassed, it is assumed the WTP has a locally configured AC.

AC: The AC initiates this transition by invoking the DTLSListen command, which informs the DTLS stack that it is willing to listen for an incoming session. The AC MAY provide optional qualifiers in the DTLSListen command to only accept session requests from specific WTPs.

Discovery to DTLS Setup (f): This transition occurs to establish a secure DTLS session with the peer.

WTP: The WTP initiates this transition by invoking the DTLSStart command (see Section 2.3.2.1), which starts the DTLS session establishment with the chosen AC. The decision of which AC to connect to is the result of the discovery phase, which is described in Section 3.3.

AC: The AC initiates this transition by invoking the DTLSListen command (see Section 2.3.2.1), which informs the DTLS stack that it is willing to listen for an incoming session. The AC MAY have maintained state information when it received the Discovery Request message to provide optional qualifiers in the DTLSListen command to only accept session requests from a specific WTP. Note that maintaining state information based on an unsecured Discovery Request message MAY lead to a Denial of Service attack. Therefore the AC SHOULD ensure that the state information is freed after a period, which is implementation specific.

DTLS Setup to Idle (g): This transition occurs when the DTLS Session failed to be established.

WTP: The WTP initiates this state transition when it receives a DTLSEstablishFail notification from DTLS (see Section 2.3.2.2). This error notification aborts the secure DTLS session establishment. When this notification is received, the FailedDTLSSessionCount counter is incremented.

AC: The WTP initiates this state transition when it receives a DTLSEstablishFail notification from DTLS (see Section 2.3.2.2). This error notification aborts the secure DTLS session establishment. When this notification is received, the FailedDTLSSessionCount counter is incremented.

DTLS Setup to Authorize (h): This transition occurs when an incoming DTLS session is being established, and the DTLS stack needs authorization to proceed with the session establishment.

WTP: This state transition occurs when the WTP receives the DTLSPeerAuthorize notification (see Section 2.3.2.2). Upon entering this state, the WTP performs an authorization check against the AC credentials. See Section 2.4.4 for more information on AC authorization.

AC: This state transition occurs when the AC receives the DTLSPeerAuthorize notification (see Section 2.3.2.2). Upon entering this state, the AC performs an authorization check against the WTP credentials. See Section 2.4.4 for more information on WTP authorization.

Authorize to DTLS Connect (j): This transition occurs to notify the DTLS stack that the session should be established.

WTP: This state transition occurs when the WTP has either opted to forgo the authorization check of the AC's credentials, or the credentials were successfully authorized. This is done by invoking the DTLSAccept DTLS command (see Section 2.3.2.1).

AC: This state transition occurs when the AC has either opted to forgo the authorization check of the WTP's credentials, or the credentials were successfully authorized. This is done by invoking the DTLSAccept DTLS command (see Section 2.3.2.1).

Authorize to DTLS Teardown (k): This transition occurs to notify the DTLS stack that the session should be aborted.

WTP: This state transition occurs when the WTP was unable to authorize the AC, using the AC credentials. The WTP then aborts the DTLS session by invoking the DTLSAbortSession command (see Section 2.3.2.1).

AC: This state transition occurs when the AC was unable to authorize the WTP, using the WTP credentials. The AC then aborts the DTLS session by invoking the DTLSAbortSession command (see Section 2.3.2.1).

DTLS Connect to Idle (m): This transition occurs when the DTLS Session failed to be established.

WTP: This state transition occurs when the WTP receives either a DTLSAborted or DTLSAuthenticateFail notification (see Section 2.3.2.2), indicating that the DTLS session was not successfully established. When this transition occurs due to the DTLSAuthenticateFail notification, the FailedDTLSAuthFailCount is incremented, otherwise the FailedDTLSSessionCount counter is incremented.

AC: This state transition occurs when the AC receives either a DTLSAborted or DTLSAuthenticateFail notification (see Section 2.3.2.2), indicating that the DTLS session was not successfully established. When this transition occurs due to the DTLSAuthenticateFail notification, the FailedDTLSAuthFailCount is incremented, otherwise the FailedDTLSSessionCount counter is incremented.

DTLS Connect to Join (n): This transition occurs when the DTLS Session is successfully established.

WTP: This state transition occurs when the WTP receives the DTLSEstablished notification (see Section 2.3.2.2), indicating that the DTLS session was successfully established. When this notification is received, the FailedDTLSSessionCount counter is set to zero.

AC: This state transition occurs when the AC receives the DTLSEstablished notification (see Section 2.3.2.2), indicating that the DTLS session was successfully established. When this notification is received, the FailedDTLSSessionCount counter is set to zero, and the WaitJoin timer is started (see Section 4.7).

Join to DTLS Teardown (p): This transition occurs when the join process failed.

WTP: This state transition occurs when the WTP receives a Join Response message with a Result Code message element containing an error, if the Image Identifier provided by the AC in the Join Response message differs from the WTP's currently running firmware version and the WTP has the requested image in its non-volatile memory, or if the WaitDTLS timer expires. This causes the WTP to initiate the DTLSShutdown command (see Section 2.3.2.1). This transition also occurs if the WTP receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect.

AC: This state transition occurs either if the WaitJoin timer expires or if the AC transmits a Join Response message with a Result Code message element containing an error. This causes the AC to initiate the DTLSShutdown command (see Section 2.3.2.1). This transition also occurs if the AC receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect.

Join to Image Data (r): This state transition is used by the WTP and the AC to download executable firmware.

WTP: The WTP enters the Image Data state when it receives a successful Join Response message and determines and the included Image Identifier message element is not the same as its currently running image. The WTP also detects that the requested image version is not currently available in the WTP's non-volatile storage (see Section 9.1 for a full description of the firmware download process). The WTP initializes the EchoInterval timer (see Section 4.7), and transmits the Image Data Request message (see Section 9.1.1) requesting the start of the firmware download.

AC: This state transition occurs when the AC receives the Image Data Request message from the WTP. The AC MUST transmit an Image Data Response message (see Section 9.1.2) to the WTP, which includes a portion of the firmware. The AC MUST start the NeighborDeadInterval timer (see Section 4.7).

Join to Configure (q): This state transition is used by the WTP and the AC to exchange configuration information.

WTP: The WTP enters the Configure state when it receives a successful Join Response, and determines that the included Image Identifier message element is the same as its currently running image. The WTP transmits the Configuration Status message (see Section 8.2) to the AC with message elements describing its current configuration. The WTP also starts the ResponseTimeout timer (see Section 4.7).

AC: This state transition occurs immediately after the AC transmits the Join Response message to the WTP. If the AC receives the Configuration Status message from the WTP, the AC MUST transmit a Configuration Status Response message (see Section 8.3) to the WTP, and MAY include specific message elements to override the WTP's configuration. The WTP also starts the ChangeStatePendingTimer timer (see Section 4.7).

Configure to Reset (s): This state transition is used to reset the connection either due to an error during the configuration phase, or when the WTP determines it needs to reset in order for the new configuration to take effect.

WTP: The WTP enters the Reset state when it receives a Configuration Status Response indicating an error or when it determines that a reset of the WTP is required, due to the characteristics of a new configuration.

AC: The AC transitions to the Reset state when it receives a Change State Event message from the WTP that contains an error for which AC policy does not permit the WTP to provide service. This state transition also occurs when the AC ChangeStatePendingTimer timer expires.

Configure to DTLS Teardown (V): This transition occurs when the configuration process aborts due to a DTLS error.

WTP: The WTP enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications.

AC: The AC enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications.

Image Data to Image Data (4): The Image Data state is used by the WTP and the AC during the firmware download phase.

WTP: The WTP enters the Image Data state when it receives an Image Data Response message indicating that the AC has more data to send.

AC: This state transition occurs when the AC receives the Image Data Request message from the WTP while already in the Image Data state, and it detects that the firmware download has not completed.

Image Data to Reset (o): This state transition is used to reset the DTLS connection prior to restarting the WTP after an image download.

WTP: When an image download completes, the WTP enters the Reset state. The WTP MAY also transition to this state upon receiving an Image Data Response message from the AC (see Section 9.1.2) indicating a failure.

AC: The AC enters the Reset state when the image download is complete, or if an error occurs during the image download process.

Image Data to DTLS Teardown (x): This transition occurs when the firmware download process aborts due to a DTLS error.

WTP: The WTP enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications.

AC: The AC enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications.

Configure to Data Check (t): This state transition occurs when the WTP and AC confirm the configuration.

WTP: The WTP enters this state when it receives a successful Configuration Status Response message from the AC. The WTP initializes the EchoInterval timer (see Section 4.7), and transmits the Change State Event Request message (see Section 8.6).

AC: This state transition occurs when the AC receives the Change State Event Request message (see Section 8.6) from the WTP. The AC responds with a Change State Event Response message (see Section 8.7). The AC MUST start the NeighborDeadInterval timer (see Section 4.7).

Data Check to Run (u): This state transition occurs when the linkage between the control and data channels has occured, causing the WTP and AC to enter their normal state of operation.

WTP: The WTP enters this state when it receives a successful Change State Event Response message from the AC. The WTP initiates the data channel, which MAY require the establishment of a DTLS session, starts the DataChannelKeepAlive timer (see Section 4.7) and transmits a Data Channel Keep Alive packet

(see Section 4.4.1). The WTP then starts the DataChannelDeadInterval timer (see Section 4.7).

AC: This state transition occurs when the AC receives the Data Channel Keep Alive packet (see Section 4.4.1), with a Session ID message element matching that included by the WTP in the Join Request message. Note that if AC policy is to require the data channel to be encrypted, this process would also require the establishment of a data channel DTLS session. Upon receiving the Data Channel Keep Alive packet, the AC transmits its own Data Channel Keep Alive packet.

Run to DTLS Teardown (u): This state transition occurs when an error has occured in the DTLS stack, causing the DTLS session to be torndown.

WTP: The WTP enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications. The WTP also transitions to this state if the underlying reliable transport's RetransmitCount counter has reached the MaxRetransmit variable (see Section 4.7).

AC: The AC enters this state when it receives one of the following DTLS notifications: DTLSAborted, DTLSReassemblyFailure or DTLSPeerDisconnect (see Section 2.3.2.2). The WTP MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications. The AC transitions to this state if the underlying reliable transport's RetransmitCount counter has reached the MaxRetransmit variable (see Section 4.7).

Run to Run (5): This is the normal state of operation.

WTP: This is the WTP's normal state of operation. There are many events that result this state transition:

Configuration Update: The WTP receives a Configuration Update Request message(see Section 8.4). The WTP MUST respond with a Configuration Update Response message (see Section 8.5).

Change State Event: The WTP receives a Change State Event Response message, or determines that it must initiate a Change State Event Request message, as a result of a failure or change in the state of a radio.

Echo Request: The WTP sends an Echo Request message (Section 7.1) or receives the corresponding Echo Response message, (see Section 7.2) from the AC.

Clear Config Request: The WTP receives a Clear Configuration Request message (see Section 8.8). The WTP MUST reset its configuration back to manufacturer defaults.

WTP Event: The WTP sends a WTP Event Request message, delivering information to the AC (see Section 9.4). The WTP receives a WTP Event Response message from the AC (see Section 9.5).

Data Transfer: The WTP sends a Data Transfer Request message to the AC (see Section 9.6). The WTP receives a Data Transfer Response message from the AC (see Section 9.7).

Station Configuration Request: The WTP receives a Station Configuration Request message (see Section 10.1), to which it MUST respond with a Station Configuration Response message (see Section 10.2).

AC: This is the AC's normal state of operation:

Configuration Update: The AC sends a Configuration Update Request message (see Section 8.4) to the WTP to update its configuration. The AC receives a Configuration Update Response message (see Section 8.5) from the WTP.

Change State Event: The AC receives a Change State Event Request message (see Section 8.6), to which it MUST respond with the Change State Event Response message (see Section 8.7).

Echo Request: The AC receives an Echo Request message (see Section 7.1), to which it MUST respond with an Echo Response message(see Section 7.2).

Clear Config Response: The AC receives a Clear Configuration Response message from the WTP (see Section 8.9).

WTP Event: The AC receives a WTP Event Request message from the WTP (see Section 9.4) and MUST generate a corresponding WTP Event Response message (see Section 9.5).

Data Transfer: The AC receives a Data Transfer Request message from the WTP (see Section 9.6) and MUST generate a corresponding Data Transfer Response message (see Section 9.7).

Station Configuration Request: The AC sends a Station Configuration Request message (see Section 10.1) or receives the corresponding Station Configuration Response message (see Section 10.2) from the WTP.

Run to Reset (x): This state transition is used when either the AC or WTP tear down the connection. This may occur as part of normal operation, or due to error conditions.

WTP: The WTP enters the Reset state when it receives a Reset Request message from the AC.

AC: The AC enters the Reset state when it transmits a Reset Request message to the WTP.

Reset to DTLS Teardown (y): This transition occurs when the CAPWAP reset is complete, to terminate the DTLS session.

WTP: This state transition occurs when the WTP receives a Reset Response message. This causes the WTP to initiate the DTLSShutdown command (see Section 2.3.2.1).

AC: This state transition occurs when the AC transmits a Reset Response message. The AC does not invoke the DTLSShutdown command (see Section 2.3.2.1).

DTLS Teardown to Idle (z): This transition occurs when the DTLS session has been shutdown.

WTP: This state transition occurs when the WTP has successfully cleaned up all resources associated with the control plane DTLS session. The data plane DTLS session is also shutdown, and all resources freed, if a DTLS session was established for the data plane. Any timers set for the current instance of the state machine are also cleared.

AC: This state transition occurs when the AC has successfully cleaned up all resources associated with the control plane DTLS session. The data plane DTLS session is also shutdown, and all resources freed, if a DTLS session was established for the data plane. Any timers set for the current instance of the state machine are also cleared.

2.3.2. CAPWAP/DTLS Interface

This section describes the DTLS Commands used by CAPWAP, and the notifications received from DTLS to the CAPWAP protocol stack.

2.3.2.1. CAPWAP to DTLS Commands

Six commands are defined for the CAPWAP to DTLS API. These "commands" are conceptual, and may be implemented as one or more function calls. This API definition is provided to clarify interactions between the DTLS and CAPWAP components of the integrated CAPWAP state machine.

Below is a list of the minimal command API:

- DTLSStart is sent to the DTLS component to cause a DTLS session to be established. Upon invoking the DTLSStart command, the WaitDTLS timer is started. The WTP initiates this DTLS command, as the AC does not initiate DTLS sessions.

- DTLSListen is sent to the DTLS component to allow the DTLS component to listen for incoming DTLS session requests.

- DTLSAccept is sent to the DTLS component to allow the DTLS session establishment to continue successfully.

- DTLSAbortSession is sent to the DTLS component to cause the session that is in the process of being established to be aborted. This command is also sent when the WaitDTLS timer expires. When this command is executed, the FailedDTLSSessionCount counter is incremented.

- DTLSShutdown is sent to the DTLS component to cause session teardown.

- DTLSMtuUpdate is sent by the CAPWAP component to modify the MTU size used by the DTLS component. The default size is 1468 bytes.

2.3.2.2. DTLS to CAPWAP Notifications

DTLS notifications are defined for the DTLS to CAPWAP API. These "notifications" are conceptual, and may be implemented in numerous ways (e.g. as function return values). This API definition is provided to clarify interactions between the DTLS and CAPWAP components of the integrated CAPWAP state machine. It is important to note that the notifications listed below MAY cause the CAPWAP state machine to jump from one state to another using a state transition not listed in Section 2.3.1. When a notification listed

below occurs, the target CAPWAP state shown in Figure 3 becomes the current state.

Below is a list of the API notifications:

- DTLSPeerAuthorize is sent to the CAPWAP component during DTLS session establishment once the peer's identity has been received. This notification MAY be used by the CAPWAP component to authorize the session, based on the peer's identity. The authorization process will lead to the CAPWAP component initiating either the DTLSAccept or DTLSAbortSession commands.

- DTLSEstablished is sent to the CAPWAP component to indicate that that a secure channel now exists, using the parameters provided during the DTLS initialization process. When this notification is received, the FailedDTLSSessionCount counter is reset to zero. When this notification is received, the WaitDTLS timer is stopped.

- DTLSEstablishFail is sent when the DTLS session establishment has failed, either due to a local error, or due to the peer rejecting the session establishment. When this notification is received, the FailedDTLSSessionCount counter is incremented.

- DTLSAuthenticateFail is sent when DTLS session establishment failed due to an authentication error. When this notification is received, the FailedDTLSAuthFailCount counter is incremented.

- DTLSAborted is sent to the CAPWAP component to indicate that session abort (as requested by CAPWAP) is complete; this occurs to confirm a DTLS session abort, or when the WaitDTLS timer expires. When this notification is received, the WaitDTLS timer is stopped.

- DTLSReassemblyFailure MAY be sent to the CAPWAP component to indicate DTLS fragment reassembly failure.

- DTLSDecapFailure MAY be sent to the CAPWAP module to indicate a decapsulation failure. DTLSDecapFailure MAY be sent to the CAPWAP module to indicate an encryption/authentication failure. This notification is intended for informative purposes only, and is not intended to cause a change in the CAPWAP state machine (see Section 12.4).

- DTLSPeerDisconnect is sent to the CAPWAP component to indicate the DTLS session has been torn down. Note that this notification is only received if the DTLS session has been established.

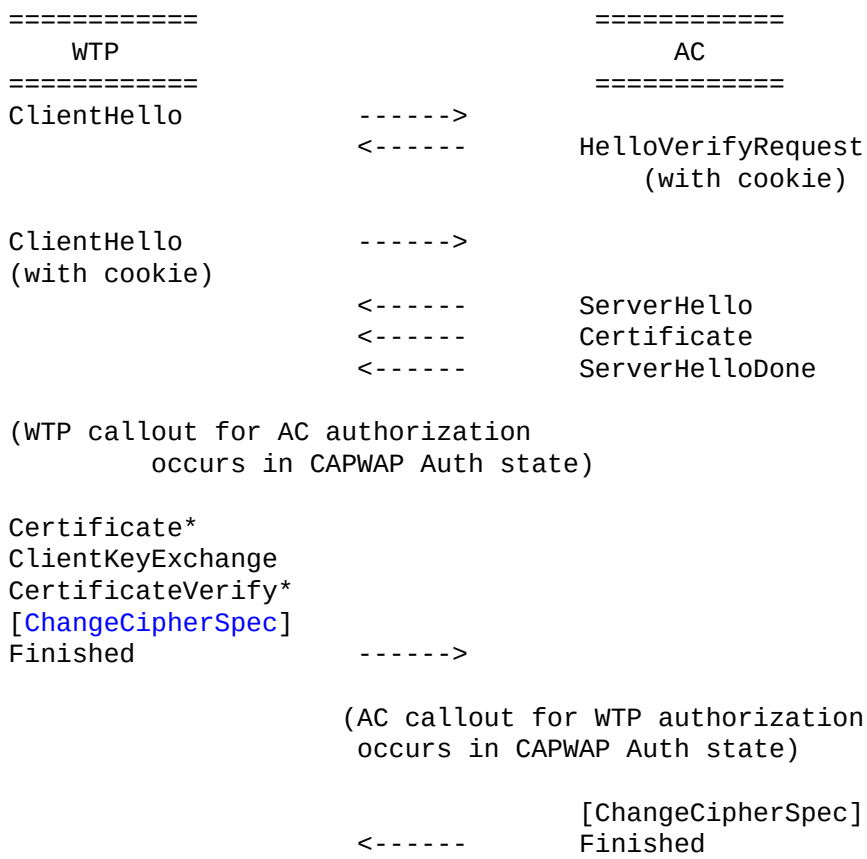
2.4. Use of DTLS in the CAPWAP Protocol

DTLS is used as a tightly-integrated, secure wrapper for the CAPWAP protocol. In this document DTLS and CAPWAP are discussed as nominally distinct entitites; however they are very closely coupled, and may even be implemented inseparably. Since there are DTLS library implementations currently available, and since security protocols (e.g. IPsec, TLS) are often implemented in widely available acceleration hardware, it is both convenient and forward-looking to maintain a modular distinction in this document.

This section describes a detailed walk-through of the interactions between the DTLS module and the CAPWAP module, via 'commands' (CAPWAP to DTLS) and 'notifications' (DTLS to CAPWAP) as they would be encountered during the normal course of operation.

2.4.1. DTLS Handshake Processing

Details of the DTLS handshake process are specified in [8]. This section describes the interactions between the DTLS session establishment process and the CAPWAP protocol. Note that the conceptual DTLS state is shown below to help understand the point at which the DTLS states transition. In the normal case, the DTLS handshake will proceed as follows (NOTE: this example uses certificates, but preshared keys are also supported):



DTLS, as specified, provides its own retransmit timers with an exponential back-off. However, DTLS will never terminate the handshake due to non-responsiveness; instead, DTLS will continue to increase its back-off timer period. Hence, timing out incomplete DTLS handshakes is entirely the responsiblity of the CAPWAP module.

The DTLS implementation used by CAPWAP MUST support TLS Session Resumption. Session resumption is used to establish the DTLS session used for the data channel. The DTLS implementation on the WTP MUST return some unique identifier to the CAPWAP module to enable subsequent establishment of a DTLS-encrypted data channel, if necessary.

2.4.2. DTLS Session Establishment

The WTP, either through the Discovery process, or through pre-configuration, determines the AC to connect to. The WTP uses the DTLSStart command to request that a secure connection be established to the selected AC. Prior to initiation of the DTLS handshake, the

WTP sets the WaitDTLS timer. Upon receiving the DTLSPeerAuthorize DTLS notification, the AC sets the WaitDTLS timer. If the DTLSEstablished notification is not received prior to timer expiration, the DTLS session is aborted by issuing the DTLSAbortSession DTLS command. This notification causes the CAPWAP module to transition to the Idle state. Upon receiving a DTLSEstablished notification, the WaitDTLS timer is deactivated.

2.4.3. DTLS Error Handling

If the AC does not respond to any DTLS messages sent by the WTP, the DTLS specification calls for the WTP to retransmit these messages. If the WaitDTLS timer expires, CAPWAP will issue the DTLSAbortSession command, causing DTLS to terminate the handshake and remove any allocated session context. Note that DTLS MAY send a single TLS Alert message to the AC to indicate session termination.

If the WTP does not respond to any DTLS messages sent by the AC, the CAPWAP protocol allows for three possiblities, listed below. Note that DTLS MAY send a single TLS Alert message to the AC to indicate session termination.

- The message was lost in transit; in this case, the WTP will re-transmit its last outstanding message, since it did not receive a reply.

- The WTP sent a DTLS Alert, which was lost in transit; in this case, the AC's WaitDTLS timer will expire, and the session will be terminated.

- Communication with the WTP has completely failed; in this case, the AC's WaitDTLS timer will expire, and the session will be terminated.

The DTLS specification provides for retransmission of unacknowledged requests. If retransmissions remain unacknowledged, the WaitDTLS timer will eventually expire, at which time the CAPWAP component will terminate the session.

If a cookie fails to validate, this could represent a WTP error, or it could represent a DoS attack. Hence, AC resource utilization SHOULD be minimized. The AC MAY log a message indicating the failure, but SHOULD NOT attempt to reply to the WTP.

Since DTLS handshake messages are potentially larger than the maximum record size, DTLS supports fragmenting of handshake messages across multiple records. There are several potential causes of re-assembly errors, including overlapping and/or lost fragments. The DTLS

component MUST send a DTLSReassemblyFailure notification to the CAPWAP component. Whether precise information is given along with notification is an implementation issue, and hence is beyond the scope of this document. Upon receipt of such an error, the CAPWAP component SHOULD log an appropriate error message. Whether processing continues or the DTLS session is terminated is implementation dependent.

DTLS decapsulation errors consist of three types: decryption errors, authentication errors, and malformed DTLS record headers. Since DTLS authenticates the data prior to encapsulation, if decryption fails, it is difficult to detect this without first attempting to authenticate the packet. If authentication fails, a decryption error is also likely, but not guaranteed. Rather than attempt to derive (and require the implementation of) algorithms for detecting decryption failures, decryption failures are reported as authentication failures. The DTLS component MUST provide a DTLSDecapFailure notification to the CAPWAP component when such errors occur. If a malformed DTLS record header is detected, the packets SHOULD be silently discarded, and the receiver MAY log an error message.

There is currently only one encapsulation error defined: MTU exceeded. As part of DTLS session establishment, the CAPWAP component informs the DTLS component of the MTU size. This may be dynamically modified at any time when the CAPWAP component sends the DTLSMtuUpdate command to the DTLS component (see Section 2.3.2.1). The DTLS component returns this notification to the CAPWAP component whenever a transmission request will result in a packet which exceeds the MTU.

2.4.4. DTLS EndPoint Authentication and Authorization

DTLS supports endpoint authentication with certificates or preshared keys. The TLS algorithm suites for each endpoint authentication method are described below.

2.4.4.1. Authenticating with Certificates

Note that only block ciphers are currently recommended for use with DTLS. To understand the reasoning behind this, see [17]. At present, the following algorithms MUST be supported when using certificates for CAPWAP authentication:

- TLS_RSA_WITH_AES_128_CBC_SHA

The following algorithms SHOULD be supported when using certificates:

- TLS_DH_RSA_WITH_AES_128_CBC_SHA

The following algorithms MAY be supported when using certificates:

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DH_RSA_WITH_AES_256_CBC_SHA

2.4.4.2. Authenticating with Preshared Keys

Pre-shared keys present significant challenges from a security perspective, and for that reason, their use is strongly discouraged. Several methods for authenticating with preshared keys are defined [6], and we focus on the following two:

- PSK key exchange algorithm - simplest method, ciphersuites use only symmetric key algorithms
- DHE_PSK key exchange algorithm - use a PSK to authenticate a Diffie-Hellman exchange. These ciphersuites give some additional protection against dictionary attacks and also provide Perfect Forward Secrecy (PFS).

The first approach (plain PSK) is susceptible to passive dictionary attacks; hence, while this alorithm MUST be supported, special care should be taken when choosing that method. In particular, user-readable passphrases SHOULD NOT be used, and use of short PSKs SHOULD be strongly discouraged.

The following cryptographic algorithms MUST be supported when using preshared keys:

- TLS_PSK_WITH_AES_128_CBC_SHA
- TLS_DHE_PSK_WITH_AES_128_CBC_SHA

The following algorithms MAY be supported when using preshared keys:

- TLS_PSK_WITH_AES_256_CBC_SHA
- TLS_DHE_PSK_WITH_AES_256_CBC_SHA

2.4.4.3. Certificate Usage

Certificate authorization by the AC and WTP is required so that only an AC may perform the functions of an AC and that only a WTP may perform the functions of a WTP. This restriction of functions to the AC or WTP requires that the certificates used by the AC MUST be

distinguishable from the certificate used by the WTP. To accomplish this differentiation, the x.509 certificates MUST include the Extended Key Usage (EKU) certificate extension [4].

The EKU field indicates one or more purposes for which a certificate may be used. It is an essential part in authorization. Its syntax is as follows:

```
ExtKeyUsageSyntax  ::=  SEQUENCE SIZE (1..MAX) OF KeyPurposeId

KeyPurposeId  ::=  OBJECT IDENTIFIER
```

Here we define two KeyPurposeId values, one for the WTP and one for the AC. Inclusion of one of these two values indicates a certificate is authorized for use by a WTP or AC, respectively. These values are formatted as id-kp fields.

```
id-kp  OBJECT IDENTIFIER  ::=
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) 3 }

 id-kp-capwapAC   OBJECT IDENTIFIER  ::=  { id-kp 18 }

 id-kp-capwapWTP  OBJECT IDENTIFIER  ::=  { id-kp 19 }
```

For an AC, the id-kp-capwapAC EKU MUST be present in the certificate. For a WTP, the id-kp-capwapWTP EKU MUST be present in the certificate.

Part of the CAPWAP certificate validation process includes ensuring that the proper EKU is included and allowing the CAPWAP session to be established only if the extension properly represents the device.

The certificate common name (CN) for both the WTP and AC MUST be the MAC address of that device. The MAC address MUST be formatted as ASCII HEX, e.g. 01:23:45:67:89:ab.

ACs and WTPs SHOULD authorize (e.g. through access control lists) certificates of devices to which they are connecting, based on the MAC address and organizational information specified in the O and OU fields. The identities specified in the certificates bind a particular DTLS session to a specific pair of mutually-authenticated and authorized MAC addresses.

2.4.4.4. PSK Usage

When DTLS uses PSK Ciphersuites, the ServerKeyExchange message MUST contain the "PSK identity hint" field and the ClientKeyExchange message MUST contain the "PSK identity" field. These fields are used to help the WTP select the appropriate PSK for use with the AC, and then indicate to the AC which key is being used. When PSKs are provisioned to WTPs and ACs, both the PSK Hint and PSK Identity for the key MUST be specified.

The PSK Hint SHOULD uniquely identify the AC and the PSK Identity SHOULD uniquely identify the WTP. It is RECOMMENDED that these hints and identities be the ASCII HEX-formatted MAC addresses of the respective devices, since each pairwise combination of WTP and AC SHOULD have a unique PSK. The PSK hint and identity SHOULD be sufficient to perform authorization, as simply having knowledge of a PSK does not necessarily imply authorization.

If a single PSK is being used for multiple devices on a CAPWAP network, which is NOT RECOMMENDED, the PSK Hint and Identity can no longer be a MAC address, so appropriate hints and identities SHOULD be selected to identify the group of devices to which the PSK is provisioned.

3. CAPWAP Transport

Communication between a WTP and an AC is established using the standard UDP client/server model. The CAPWAP protocol supports both UDP and UDP-Lite [11] transport protocols. The UDP protocol is used with IPv4. When CAPWAP is used over IPv6, the UDP-Lite protocol is used. This section describes how the CAPWAP protocol is carried over IP and UDP/UDP-Lite transport protocols.

3.1. UDP Transport

One of the CAPWAP protocol requirements is to allow a WTP to reside behind a firewall and/or Network Address Translation (NAT) device. Since a CAPWAP session is initiated by the WTP (client) to the well-known UDP port of the AC (server), the use of UDP is a logical choice. The UDP checksum field in CAPWAP packets MUST be set to zero.

CAPWAP protocol control packets sent from the WTP to the AC use the CAPWAP control channel, as defined in Section 1.4. The CAPWAP control port at the AC is the well known UDP port [to be IANA assigned]. The CAPWAP control port at the WTP can be any port selected by the WTP.

CAPWAP protocol data packets sent from the WTP to the AC use the CAPWAP data channel, as defined in Section 1.4. The CAPWAP data port at the AC is the well known UDP port [to be IANA assigned]. The CAPWAP data port at the WTP can be any port selected by the WTP.

3.2. UDP-Lite Transport

When CAPWAP is run over IPv6, UDP-Lite is used as the transport protocol, reducing the checksum processing required for each packet (compared to UDP and IPv6). When UDP-Lite is used, the checksum field MUST have a coverage of 8 [11].

UDP-Lite uses the same port assignments as UDP.

3.3. AC Discovery

The AC discovery phase allows the WTP to determine which ACs are available, and chose the best AC with which to establish a CAPWAP session. The discovery phase occurs when the WTP enters the optional Discovery state. A WTP does not need to complete the AC Discovery phase if it uses a pre-configured AC. This section details the mechanism used by a WTP to dynamically discover candidate ACs.

A WTP and an AC will frequently not reside in the same IP subnet

(broadcast domain). When this occurs, the WTP must be capable of discovering the AC, without requiring that multicast services are enabled in the network.

When the WTP attempts to establish communication with an AC, it sends the Discovery Request message and receives the Discovery Response message from the AC(s). The WTP MUST send the Discovery Request message to either the limited broadcast IP address (255.255.255.255), a well known multicast address or to the unicast IP address of the AC. For IPv6 networks, since broadcast does not exist, the use of "All ACs multicast address" is used instead. Upon receipt of the Discovery Request message, the AC sends a Discovery Response message to the unicast IP address of the WTP, regardless of whether the Discovery Request message was sent as a broadcast, multicast or unicast message.

WTP use of a limited IP broadcast, multicast or unicast IP address is implementation dependent.

When a WTP transmits a Discovery Request message to a unicast address, the WTP must first obtain the IP address of the AC. Any static configuration of an AC's IP address on the WTP non-volatile storage is implementation dependent. However, additional dynamic schemes are possible, for example:

DHCP: See [13] for more information on the use of DHCP to discover AC IP addresses.

DNS: The DNS name "CAPWAP-AC-Address" MAY be resolvable to one or more AC addresses.

An AC MAY also communicate alternative ACs to the WTP within the Discovery Response message through the AC IPv4 List (see Section 4.6.2) and AC IPv6 List (see Section 4.6.2). The addresses provided in these two message elements are intended to help the WTP discover additional ACs through means other than those listed above.

The AC Name with Index message element (see Section 4.6.5), is used to communicate a list of preferred ACs to the WTP. The WTP SHOULD attempt to utilize the ACs listed in the order provided by the AC. The Name to IP Address mapping is handled via the Discovery message exchange, in which the ACs provide their identity in the AC Name (see Section 4.6.4) message element in the Discovery Response message.

Once the WTP has received Discovery Response messages from the candidate ACs, it MAY use other factors to determine the preferred AC. For instance, each binding defines a WTP Radio Information message element (see Section 2.1), which the AC includes in Discovery

Response messages. The presence of one or more of these message elements is used to identify the CAPWAP bindings supported by the AC. A WTP MAY connect to an AC based on the supported bindings advertised.

3.4. Fragmentation/Reassembly

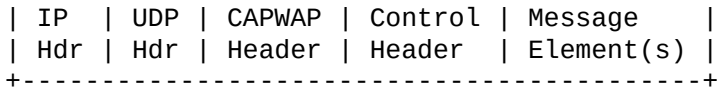
While fragmentation and reassembly services are provided by IP, the CAPWAP protocol also provides such services. Environments where the CAPWAP protocol is used involve firewall, NAT and "middle box" devices, which tend to drop IP fragments to minimize possible DoS attacks. By providing fragmentation and reassembly at the application layer, any fragmentation required due to the tunneling component of the CAPWAP protocol becomes transparent to these intermediate devices. Consequently, the CAPWAP protocol can be used in any network configuration.

4. CAPWAP Packet Formats

This section contains the CAPWAP protocol packet formats. A CAPWAP protocol packet consists of one or more CAPWAP Transport Layer packet headers followed by a CAPWAP message. The CAPWAP message can be either of type Control or Data, where Control packets carry signaling, and Data packets carry user payloads. The CAPWAP frame formats for CAPWAP Data packets, and for DTLS encapsulated CAPWAP Data and Control packets are defined below.

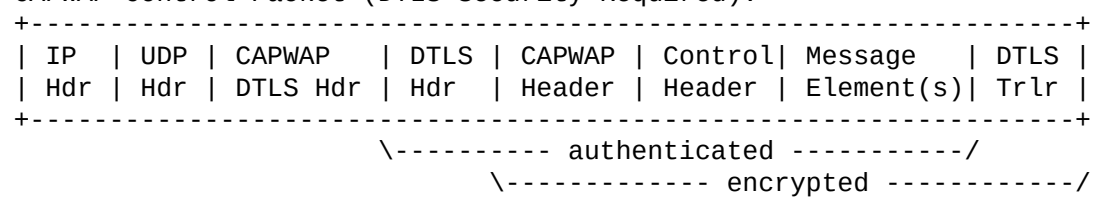
The CAPWAP Control protocol includes two messages that are never protected by DTLS: the Discovery Request message and the Discovery Response message. These messages need to be in the clear to allow the CAPWAP protocol to properly identify and process them. The format of these packets are as follows:

CAPWAP Control Packet (Discovery Request/Response):

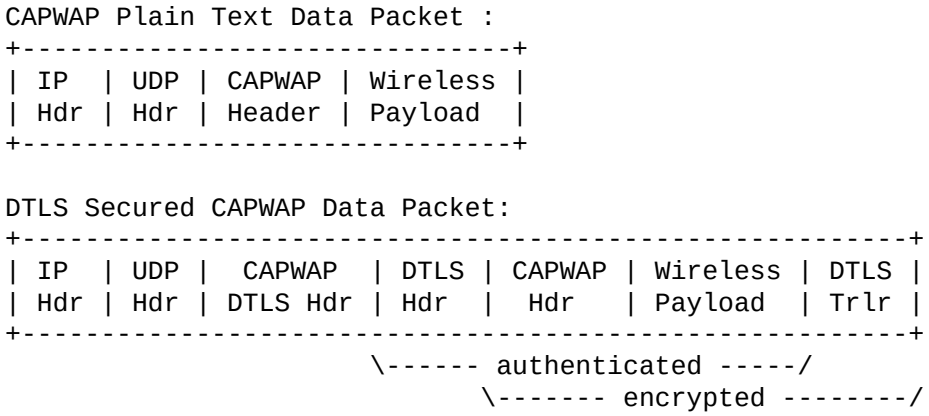


All other CAPWAP control protocol messages MUST be protected via the DTLS protocol, which ensures that the packets are both authenticated and encrypted. These packets include the CAPWAP DTLS Header, which is described in Section 4.2. The format of these packets is as follows:

CAPWAP Control Packet (DTLS Security Required):



The CAPWAP protocol allows optional protection of data packets, using DTLS. Use of data packet protection is determined by AC policy. When DTLS is utilized, the optional CAPWAP DTLS Header is present, which is described in Section 4.2. The format of CAPWAP data packets is shown below:



UDP Header: All CAPWAP packets are encapsulated within either UDP, or UDP-Lite when used over IPv6. Section 3 defines the specific UDP or UDP-Lite usage.

CAPWAP DTLS Header: All DTLS encrypted CAPWAP protocol packets are prefixed with the CAPWAP DTLS header (see Section 4.2).

DTLS Header: The DTLS header provides authentication and encryption services to the CAPWAP payload it encapsulates. This protocol is defined in RFC 4347 [8].

CAPWAP Header: All CAPWAP protocol packets use a common header that immediately follows the CAPWAP preamble or DTLS header. The CAPWAP Header is defined in Section 4.3.

Wireless Payload: A CAPWAP protocol packet that contains a wireless payload is a CAPWAP data packet. The CAPWAP protocol does not specify the format of the wireless payload, which is defined by the appropriate wireless standard. Additional information is in Section 4.4.

Control Header: The CAPWAP protocol includes a signalling component, known as the CAPWAP control protocol. All CAPWAP control packets include a Control Header, which is defined in Section 4.5.1. CAPWAP data packets do not contain a Control Header field.

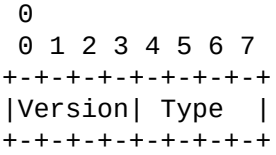
Message Elements: A CAPWAP Control packet includes one or more message elements, which are found immediately following the Control Header. These message elements are in a Type/Length/value style header, defined in Section 4.6.

A CAPWAP implementation MUST be capable of receiving a reassembled CAPWAP message of length 4096 bytes. A CAPWAP implementation MAY indicate that it supports a higher maximum message length, by

including the Maximum Message Length message element, see Section 4.6.29 in the Join Request message or the Join Response message.

4.1. CAPWAP Preamble

The CAPWAP preamble is common to all CAPWAP transport headers and is used to identify the header type that immediately follows. The reason for this header is to avoid needing to perform byte comparisons in order to guess whether the frame is DTLS encrypted or not. It also provides an extensibility framework that can be used to support additional transport types. The format of the preamble is as follows:



Version: A 4 bit field which contains the version of CAPWAP used in this packet. The value for this specification is zero (0).

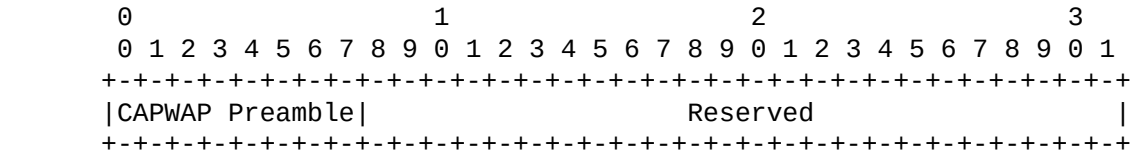
Payload Type: A 4 bit field which specifies the payload type that follows the UDP header. The following values are supported:

0 - CAPWAP Header. The CAPWAP Header (see Section 4.3) immediately follows the UDP header. If the packet is received on the CAPWAP data channel, the CAPWAP stack MUST treat the packet as a clear text CAPWAP data packet. If received on the CAPWAP control channel, the CAPWAP stack MUST treat the packet as a clear text CAPWAP control packet. If the control packet is not a Discovery Request or Discovery Response packet, the packet MUST be dropped.

1 - CAPWAP DTLS Header. The CAPWAP DTLS Header, and DTLS packet, immediately follows the UDP header (see Section 4.2).

4.2. CAPWAP DTLS Header

The CAPWAP DTLS Header is used to identify the packet as a DTLS encrypted packet. The first eight bits includes the common CAPWAP Preamble. The remaining 24 bits are padding to ensure 4 byte alignment, and MAY be used in a future version of the protocol. The DTLS packet [8] always immediately follows this header. The format of the CAPWAP DTLS Header is as follows:



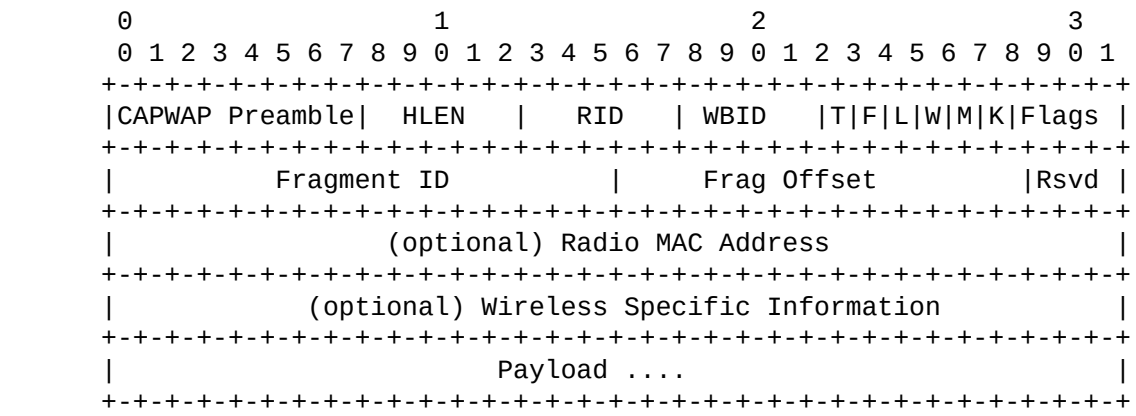
CAPWAP Preamble: The CAPWAP Preamble is defined in Section 4.1. The CAPWAP Preamble's Payload Type field MUST be set to one (1).

Reserved: The 24-bit field is reserved for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

4.3. CAPWAP Header

All CAPWAP protocol messages are encapsulated using a common header format, regardless of the CAPWAP Control or CAPWAP Data transport used to carry the messages. However, certain flags are not applicable for a given transport. Refer to the specific transport section in order to determine which flags are valid.

Note that the optional fields defined in this section MUST be present in the precise order shown below.



CAPWAP Preamble: The CAPWAP Preamble is defined in Section 4.1. The CAPWAP Preamble's Payload Type field MUST be set to zero (0). If the CAPWAP DTLS Header is present, the version number in both CAPWAP Preambles MUST match. The reason for this duplicate field is to avoid any possible tampering of the version field in the preamble which is not encrypted or authenticated.

HLEN: A 5 bit field containing the length of the CAPWAP transport header in 4 byte words (Similar to IP header length). This length includes the optional headers.

RID: A 5 bit field which contains the Radio ID number for this packet. Given that MAC Addresses are not necessarily unique across physical radios in a WTP, the Radio Identifier (RID) field is used to indiciate which physical radio the message is associated with.

WBID: A 5 bit field which is the wireless binding identifier. The identifier will indicate the type of wireless packet type associated with the radio. The following values are defined:

1 - IEEE 802.11

2 - IEEE 802.16

3 - EPCGlobal

T: The Type 'T' bit indicates the format of the frame being transported in the payload. When this bit is set to one (1), the payload has the native frame format indicated by the WBID field. When this bit is zero (0) the payload is an IEEE 802.3 frame.

F: The Fragment 'F' bit indicates whether this packet is a fragment. When this bit is one (1), the packet is a fragment and MUST be combined with the other corresponding fragments to reassemble the complete information exchanged between the WTP and AC.

L: The Last 'L' bit is valid only if the 'F' bit is set and indicates whether the packet contains the last fragment of a fragmented exchange between WTP and AC. When this bit is 1, the packet is the last fragment. When this bit is 0, the packet is not the last fragment.

W: The Wireless 'W' bit is used to specify whether the optional Wireless Specific Information field is present in the header. A value of one (1) is used to represent the fact that the optional header is present.

M: The M bit is used to indicate that the Radio MAC Address optional header is present. This is used to communicate the MAC address of the receiving radio.

K: The 'Keep-alive' K bit indicates the packet is a Data Channel Keep Alive packet. This packet is used to map the data channel to the control channel for the specified Session ID and to maintain freshness of the data channel. The K bit MUST NOT be set for data packets containing user data.

Flags: A set of reserved bits for future flags in the CAPWAP header. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

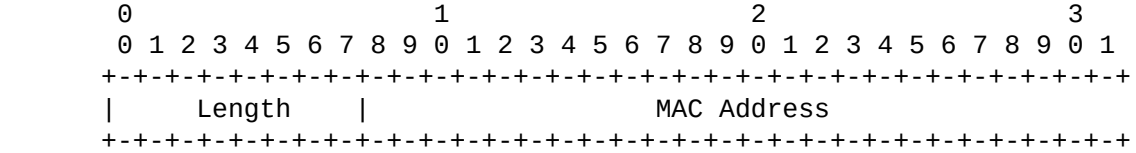
Fragment ID: A 16 bit field whose value is assigned to each group of fragments making up a complete set. The fragment ID space is managed individually for every WTP/AC pair. The value of Fragment ID is incremented with each new set of fragments. The Fragment ID wraps to zero after the maximum value has been used to identify a set of fragments.

Fragment Offset: A 13 bit field that indicates where in the payload this fragment belongs during re-assembly. This field is valid when the 'F' bit is set to 1. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero. Note the CAPWAP protocol does not allow for overlapping fragments.

Reserved: The 3-bit field is reserved for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

Radio MAC Address: This optional field contains the MAC address of the radio receiving the packet. This is useful in packets sent from the WTP to the AC, when the native wireless frame format is converted to 802.3 by the WTP. This field is only present if the 'M' bit is set. The HLEN field assumes 4 byte alignment, and this field MUST be padded with zeroes (0x00) if it is not 4 byte aligned.

The field contains the basic format:

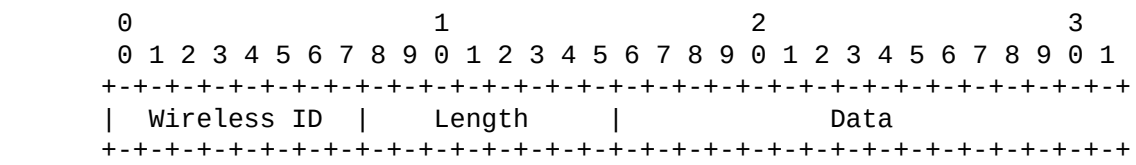


Length: The length of the MAC Address field [18] [19].

MAC Address: The MAC Address of the receiving radio.

Wireless Specific Information: This optional field contains technology specific information that may be used to carry per packet wireless information. This field is only present if the 'W' bit is set. The HLEN field assumes 4 byte alignment, and this field MUST be padded with zeroes (0x00) if it is not 4 byte aligned.

The Wireless Specific Information field uses the following format:



Wireless ID: The wireless binding identifier. The following values are defined:

1 - IEEE 802.11

2 - IEEE 802.16

3 - EPCGlobal

Length: The length of the data field

Data: Wireless specific information, defined by the wireless specific binding.

Payload: This field contains the header for a CAPWAP Data Message or CAPWAP Control Message, followed by the data contained in the message.

4.4. CAPWAP Data Messages

There are two different types of CAPWAP data packets, CAPWAP Data Channel Keep Alive packets and Data Payload packets. The first is used by the WTP to synchronize the control and data channels, and to maintain freshness of the data channel. The second is used to transmit user payloads between the AC and WTP. This section describes both types of CAPWAP data packet formats.

Both CAPWAP data messages are transmitted on the CAPWAP data channel.

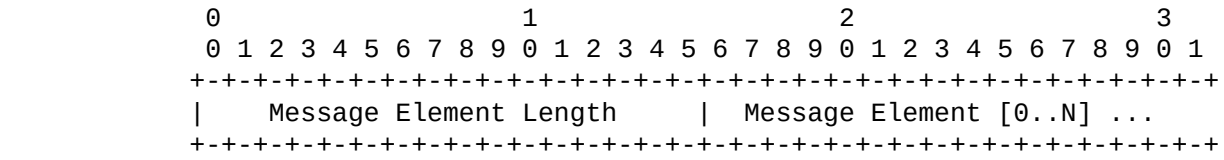
4.4.1. CAPWAP Data Keepalive

The CAPWAP Data Channel Keep Alive packet is used to bind the CAPWAP control channel with the data channel, and to maintain freshness of the data channel, ensuring that the channel is still functioning. The CAPWAP Data Channel Keep Alive packet is transmitted by the WTP when the DataChannelKeepAlive timer expires. When the CAPWAP Data Channel Keep Alive packet is transmitted, the WTP sets the DataChannelDeadInterval timer.

In the CAPWAP Data Channel Keep Alive packet, all of the fields in the CAPWAP header, except the HLEN field and the K bit, are set to zero upon transmission. Upon receiving a CAPWAP Data Channel Keep Alive packet, the AC transmits a CAPWAP Data Channel Keep Alive packet back to the WTP. The contents of the transmitted packet are identical to the contents of the received packet.

Upon receiving a CAPWAP Data Channel Keep Alive packet, the WTP cancels the DataChannelDeadInterval timer and resets the DataChannelKeepAlive timer. The CAPWAP Data Channel Keep Alive packet is retransmitted by the WTP in the same manner as the CAPWAP control messages. If the DataChannelDeadInterval timer expires, the WTP tears down the control DTLS session, and the data DTLS session if one existed.

The CAPWAP Data Channel Keep Alive packet contains the following payload immediately following the CAPWAP Header (see Section 4.3)



Message Element Length: The Length field indicates the number of bytes following the CAPWAP Header.

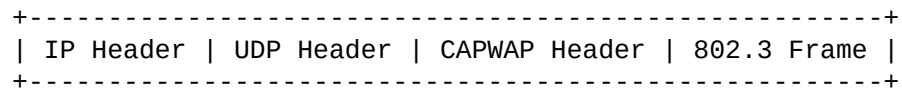
Message Element[0..N]: The message element(s) carry the information pertinent to each of the CAPWAP Data Keepalive message. The following message elements MUST be present in this CAPWAP message:

Session ID, see Section 4.6.35

4.4.2. Data Payload

A CAPWAP protocol Data Payload packet encapsulates a forwarded wireless frame. The CAPWAP protocol defines two different modes of encapsulation; IEEE 802.3 and native wireless. IEEE 802.3

encapsulation requires that the bridging function be performed in the WTP. An IEEE 802.3 encapsulated user payload frame has the following format:



The CAPWAP protocol also defines the native wireless encapsulation mode. The format of the encapsulated CAPWAP data frame is subject to the rules defined by the specific wireless technology binding. Each wireless technology binding MUST contain a section entitled "Payload Encapsulation", which defines the format of the wireless payload that is encapsulated within CAPWAP Data packets.

If the encapsulated frame would exceed the transport layer's MTU, the sender is responsible for fragmentation of the frame, as specified in Section 3.4.

4.4.3. Establishment of a DTLS Data Channel

If the AC and WTP are configured to tunnel the data channel over DTLS, the proper DTLS session must be initiated. To avoid having to reauthenticate and reauthorize an AC and WTP, the DTLS data channel MUST be initiated using the TLS session resumption feature [7].

When establishing the DTLS-encrypted data channel, the WTP MUST provide the identifier returned during the initialization of the control channel to the DTLS component so it can perform the resumption using the proper session information.

The AC DTLS implementation MUST NOT accept a session resumption request for a DTLS session in which the control channel for the session has been torn down.

4.5. CAPWAP Control Messages

The CAPWAP Control protocol provides a control channel between the WTP and the AC. Control messages are divided into the following message types:

Discovery: CAPWAP Discovery messages are used to identify potential ACs, their load and capabilities.

Join: CAPWAP Join messages are used by a WTP to request service from an AC, and for the AC to respond to the WTP.

Control Channel Management: CAPWAP control channel management messages are used to maintain the control channel.

WTP Configuration Management: The WTP Configuration messages are used by the AC to deliver a specific configuration to the WTP. Messages which retrieve statistics from a WTP are also included in WTP Configuration Management.

Station Session Management: Station Session Management messages are used by the AC to deliver specific station policies to the WTP.

Device Management Operations: Device management operations are used to request and deliver a firmware image to the WTP.

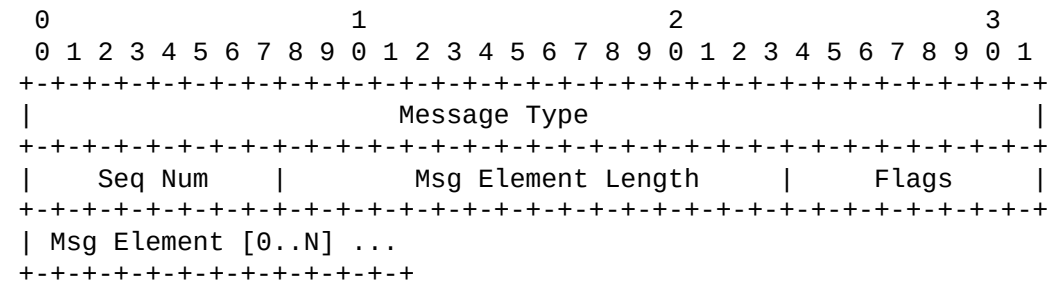
Binding Specific CAPWAP Management Messages: Messages in this category are used by the AC and the WTP to exchange protocol-specific CAPWAP management messages. These messages may or may not be used to change the link state of a station.

Discovery, Join, Control Channel Management, WTP Configuration Management and Station Session Management CAPWAP control messages MUST be implemented. Device Management Operations messages MAY be implemented.

CAPWAP control messages sent from the WTP to the AC indicate that the WTP is operational, providing an implicit keep-alive mechanism for the WTP. The Control Channel Management Echo Request and Echo Response messages provide an explicit keep-alive mechanism when other CAPWAP control messages are not exchanged.

4.5.1. Control Message Format

All CAPWAP control messages are sent encapsulated within the CAPWAP header (see Section 4.3). Immediately following the CAPWAP header, is the control header, which has the following format:



4.5.1.1. Message Type

The Message Type field identifies the function of the CAPWAP control message. The Message Type field is comprised of an IANA Enterprise Number and an enterprise specific message type number. The first three octets contain the enterprise number in network byte order, with zero used for CAPWAP protocol defined message types and the IEEE 802.11 IANA assigned enterprise number 13277 is used for IEEE 802.11 technology specific message types. The last octet is the enterprise specific message type number, which has a range from 0 to 255.

The message type field is defined as:

```
Message Type =
        IANA Enterprise Number * 256 +
            Enterprise Specific Message Type Number
```

The CAPWAP protocol reliability mechanism requires that messages be defined in pairs, consisting of both a Request and a Response message. The Response message MUST acknowledge the Request message. The assignment of CAPWAP control Message Type Values always occurs in pairs. All Request messages have odd numbered Message Type Values, and all Response messages have even numbered Message Type Values. The Request value MUST be assigned first. As an example, assigning a Message Type Value of 3 for a Request message and 4 for a Response message is valid, while assigning a Message Type Value of 4 for a Response message and 5 for the corresponding Request message is invalid.

When a WTP or AC receives a message with a Message Type Value field that is not recognized and is an odd number, the number in the Message Type Value Field is incremented by one, and a Response message with a Message Type Value field containing the incremented value and containing the Result Code message element with the value (Unrecognized Request) is returned to the sender of the received message. If the unknown message type is even, the message is ignored.

The valid values for CAPWAP Control Message Types are specified in the table below:

CAPWAP Control Message	Message Type Value
Discovery Request	1
Discovery Response	2
Join Request	3
Join Response	4
Configuration Status	5
Configuration Status Response	6
Configuration Update Request	7
Configuration Update Response	8
WTP Event Request	9
WTP Event Response	10
Change State Event Request	11
Change State Event Response	12
Echo Request	13
Echo Response	14
Image Data Request	15
Image Data Response	16
Reset Request	17
Reset Response	18
Primary Discovery Request	19
Primary Discovery Response	20
Data Transfer Request	21
Data Transfer Response	22
Clear Configuration Request	23
Clear Configuration Response	24
Station Configuration Request	25
Station Configuration Response	26

4.5.1.2. Sequence Number

The Sequence Number Field is an identifier value used to match Request and Response packets. When a CAPWAP packet with a Request Message Type Value is received, the value of the Sequence Number field is copied into the corresponding Response message.

When a CAPWAP control message is sent, the sender’s internal sequence number counter is monotonically incremented, ensuring that no two pending Request messages have the same Sequence Number. The Sequence Number field wraps back to zero.

4.5.1.3. Message Element Length

The Length field indicates the number of bytes following the Sequence Number field.

4.5.1.4. Flags

The Flags field MUST be set to zero.

4.5.1.5. Message Element[0..N]

The message element(s) carry the information pertinent to each of the control message types. Every control message in this specification specifies which message elements are permitted.

When a WTP or AC receives a CAPWAP message without a message element that is specified as mandatory for the CAPWAP message, then the CAPWAP message is discarded. If the received message was a Request message for which the corresponding Response message carries message elements, then a corresponding Response message with a Result Code message element indicating "Failure - Missing Mandatory Message Element" is returned to the sender.

When a WTP or AC receives a CAPWAP message with a message element that the WTP or AC does not recognize, the CAPWAP message is discarded. If the received message was a Request message for which the corresponding Response message carries message elements, then a corresponding Response message with a Result Code message element indicating "Failure - Unrecognized Message Element" and one or more Returned Message Element message elements is included, containing the unrecognized message element(s).

4.5.2. Control Message Quality of Service

It is recommended that CAPWAP control messages be sent by both the AC and the WTP with an appropriate Quality of Service precedence value, ensuring that congestion in the network minimizes occurrences of CAPWAP control channel disconnects. Therefore, a Quality of Service enabled CAPWAP device SHOULD use the following values:

802.1P: The precedence value of 7 SHOULD be used.

DSCP: The DSCP tag value of 46 SHOULD be used.

4.5.3. Retransmissions

The CAPWAP control protocol operates as a reliable transport. For each Request message, a Response message is defined, which is used to acknowledge receipt of the Request message. In addition, the control header Sequence Number field is used to pair the Request and Response messages (see Section 4.5.1).

Response messages are not explicitly acknowledged, therefore if a

Response message is not received, the original Request message is retransmitted. Implementations MAY cache Response messages to respond to a retransmitted Request messages with minimal local processing. Retransmitted Request messages MUST NOT be altered by the sender. The sender MUST assume that the original Request message was processed, but that the Response message was lost. Any alterations to the original Request message MUST have a new Sequence Number, and be treated as a new Request message by the receiver.

After transmitting a Request message, the RetransmitInterval (see Section 4.7) timer and MaxRetransmit (see Section 4.8) variable are used to determine if the original Request message needs to be retransmitted. The RetransmitInterval timer is used the first time the Request is retransmitted. The timer is then doubled every subsequent time the same Request message is retransmitted, up to MaxRetransmit but no more than half the EchoInterval timer (see Section 4.7.5). Response messages are not subject to these timers.

When a Request message is retransmitted, it MUST be re-encrypted via the DTLS stack. If the peer had received the Request message, and the corresponding Response message was lost, it is necessary to ensure that retransmitted Request messages are not identified as replays by the DTLS stack. Similarly, any cached Response messages that are retransmitted as a result of receiving a retransmitted Request message MUST be re-encrypted via DTLS.

Duplicate Response messages, identified by the Sequence Number field in the CAPWAP control message header, SHOULD be discarded upon receipt.

4.6. CAPWAP Protocol Message Elements

This section defines the CAPWAP Protocol message elements which are included in CAPWAP protocol control messages.

Message elements are used to carry information needed in control messages. Every message element is identified by the Type Value field, defined below. The total length of the message elements is indicated in the message element Length field.

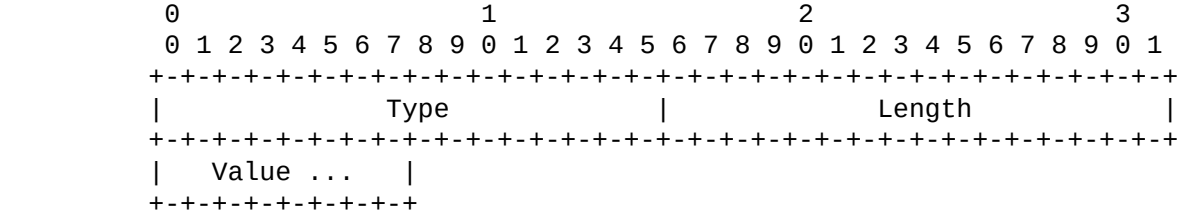
All of the message element definitions in this document use a diagram similar to the one below in order to depict its format. Note that to simplify this specification, these diagrams do not include the header fields (Type and Length). The header field values are defined in the message element descriptions.

Unless otherwise specified, a control message that lists a set of supported (or expected) message elements MUST not expect the message

elements to be in any specific order. The sender MAY include the message elements in any order. Unless otherwise noted, one message element of each type is present in a given control message.

Additional message elements may be defined in separate IETF documents.

The format of a message element uses the TLV format shown here:



The 16 bit Type field identifies the information carried in the Value field and Length (16 bits) indicates the number of bytes in the Value field. Type field values are allocated as follows:

Usage	Type Values
CAPWAP Protocol Message Elements	1-1023
IEEE 802.11 Message Elements	1024-2047
IEEE 802.16 Message Elements	2048 - 3071
EPCGlobal Message Elements	3072 - 4095
Reserved for Future Use	4096 - 65024

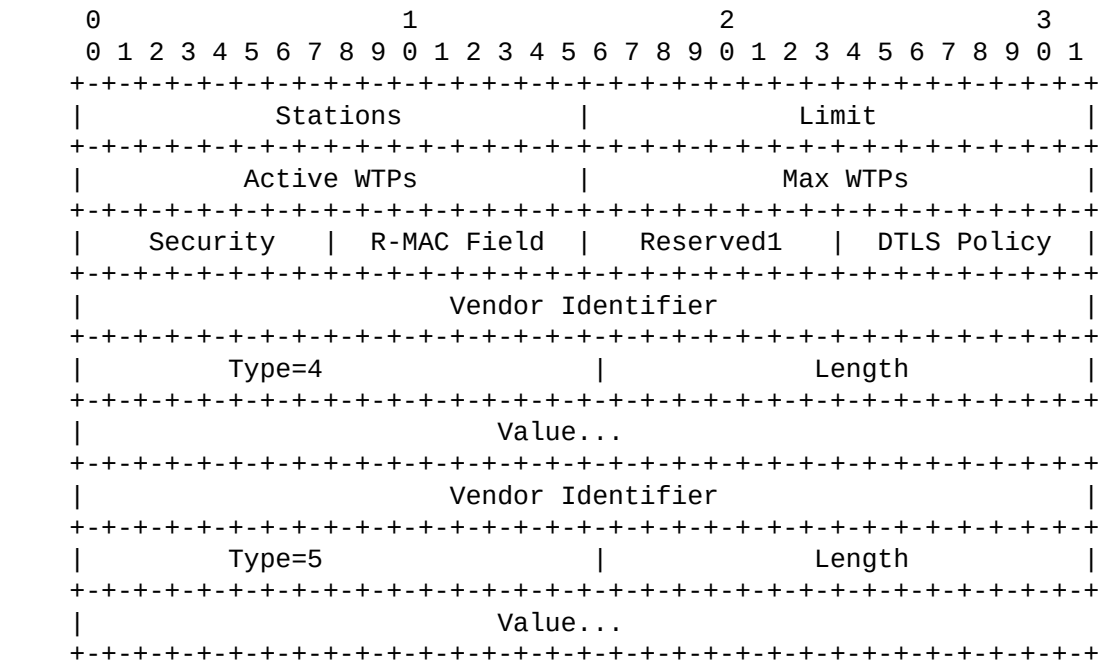
The table below lists the CAPWAP protocol Message Elements and their Type values.

CAPWAP Message Element	Type Value
AC Descriptor	1
AC IPv4 List	2
AC IPv6 List	3
AC Name	4
AC Name with Index	5
AC Timestamp	6
Add MAC ACL Entry	7
Add Station	8
Add Static MAC ACL Entry	9
CAPWAP Control IPV4 Address	10
CAPWAP Control IPV6 Address	11
CAPWAP Timers	12
Data Transfer Data	13
Data Transfer Mode	14

Decryption Error Report	15
Decryption Error Report Period	16
Delete MAC ACL Entry	17
Delete Station	18
Delete Static MAC ACL Entry	19
Discovery Type	20
Duplicate IPv4 Address	21
Duplicate IPv6 Address	22
Idle Timeout	23
Image Data	24
Image Identifier	25
Image Info	26
Initiate Download	27
Location Data	28
Maximum Message Length	29
MTU Discovery Padding	30
Radio Administrative State	31
Radio Operational State	32
Result Code	33
Returned Message Element	34
Session ID	35
Statistics Timer	36
Vendor Specific Payload	37
WTP Board Data	38
WTP Descriptor	39
WTP Fallback	40
WTP Frame Tunnel Mode	41
WTP IPv4 IP Address	42
WTP IPv6 IP Address	43
WTP MAC Type	44
WTP Name	45
WTP Operational Statistics	46
WTP Radio Statistics	47
WTP Reboot Statistics	48
WTP Static IP Address Information	49

4.6.1. AC Descriptor

The AC Descriptor message element is used by the AC to communicate its current state. The value contains the following fields.



Type: 1 for AC Descriptor

Length: >= 12

Stations: The number of stations currently served by the AC

Limit: The maximum number of stations supported by the AC

Active WTPs: The number of WTPs currently attached to the AC

Max WTPs: The maximum number of WTPs supported by the AC

Security: A 8 bit bit mask specifying the authentication credential type supported by the AC. The following values are supported (see Section 2.4.4):

1 - X.509 Certificate Based

2 - Pre-Shared Secret

R-MAC Field: The AC supports the optional Radio MAC Address field in the CAPWAP transport Header (see Section 4.3).

Reserved: A set of reserved bits for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

DTLS Policy: The AC communicates its policy on the use of DTLS for the CAPWAP data channel. The AC MAY communicate more than one supported option, represented by the bit field below. The WTP MUST abide by one of the options communicated by AC. The following bit field values are supported:

1 - Clear Text Data Channel Supported

2 - DTLS Enabled Data Channel Supported

Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes"

Type: Vendor specific encoding of AC information. The following values are supported. The Hardware and Software Version values MUST be included.

4 - Hardware Version: The AC's hardware version number.

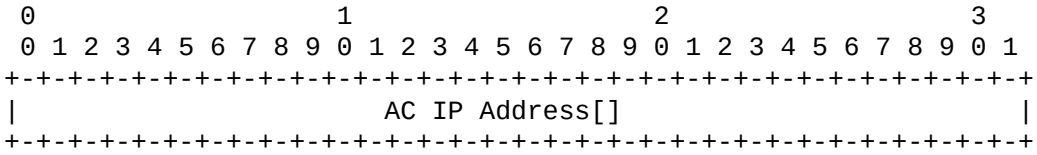
5 - Software Version: The AC's Software (firmware) version number.

Length: Length of vendor specific encoding of AC information.

Value: Vendor specific encoding of AC information.

4.6.2. AC IPv4 List

The AC IPv4 List message element is used to configure a WTP with the latest list of ACs available for the WTP to join.



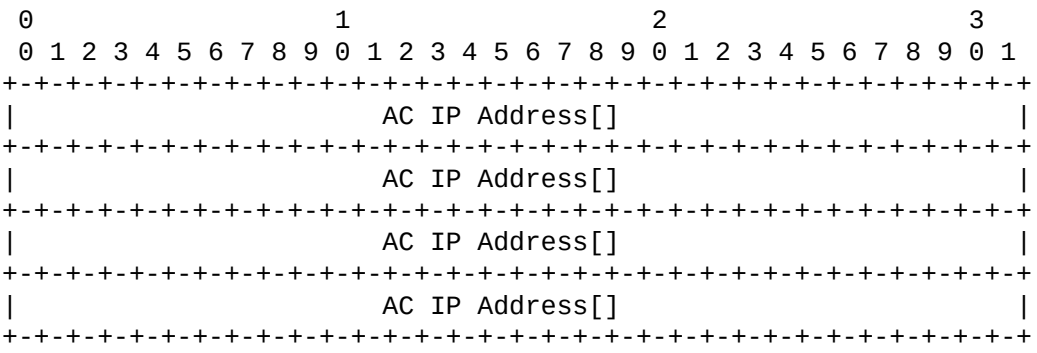
Type: 2 for AC IPv4 List

Length: >= 4

The AC IP Address: An array of 32-bit integers containing AC IPv4 Addresses.

4.6.3. AC IPv6 List

The AC IPv6 List message element is used to configure a WTP with the latest list of ACs available for the WTP to join.



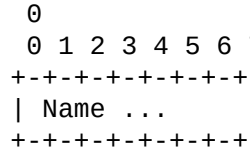
Type: 3 for AC IPV6 List

Length: >= 16

The AC IP Address: An array of 128-bit integers containing AC IPv6 Addresses.

4.6.4. AC Name

The AC Name message element contains an UTF-8 representation of the AC identity. The value is a variable length byte string. The string is NOT zero terminated.



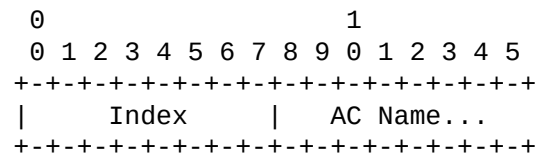
Type: 4 for AC Name

Length: > 0

Name: A variable length UTF-8 encoded string containing the AC's name

4.6.5. AC Name with Index

The AC Name with Index message element is sent by the AC to the WTP to configure preferred ACs. The number of instances of this message element is equal to the number of ACs configured on the WTP.



Type: 5 for AC Name with Index

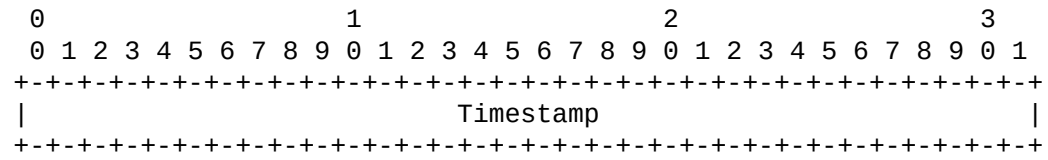
Length: > 2

Index: The index of the preferred server (1=primary, 2=secondary).

AC Name: A variable length UTF-8 encoded string containing the AC name.

4.6.6. AC Timestamp

The AC Timestamp message element is sent by the AC to synchronize the WTP clock.



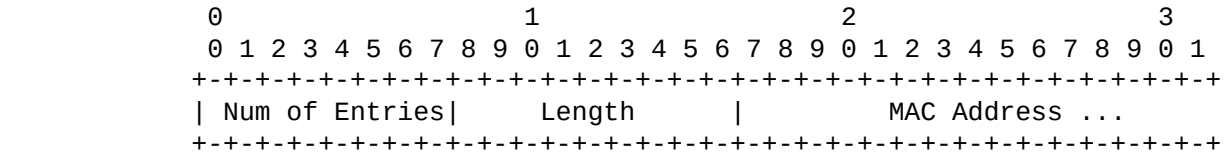
Type: 6 for AC Timestamp

Length: 4

Timestamp: The AC's current time, allowing all of the WTPs to be time synchronized in the format defined by Network Time Protocol (NTP) in RFC 1305 [3].

4.6.7. Add MAC ACL Entry

The Add MAC Access Control List (ACL) Entry message element is used by an AC to add a MAC ACL list entry on a WTP, ensuring that the WTP no longer provides service to the MAC addresses provided in the message. The MAC Addresses provided in this message element are not expected to be saved in non-volatile memory on the WTP.



Type: 7 for Add MAC ACL Entry

Length: >= 8

Num of Entries: The number of instances of the Type/MAC Addresses fields in the array.

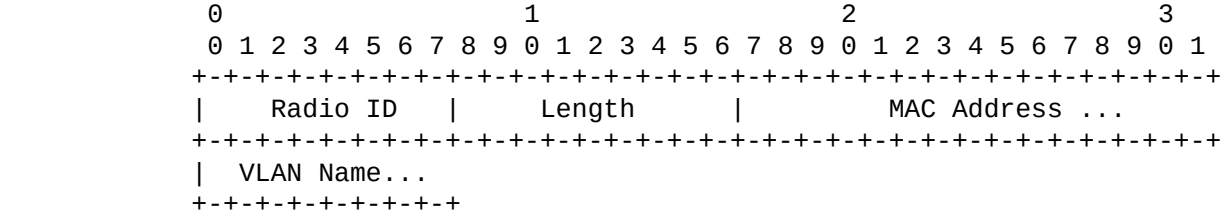
Length: The length of the MAC Address field.

MAC Address: MAC Addresses to add to the ACL.

4.6.8. Add Station

The Add Station message element is used by the AC to inform a WTP that it should forward traffic for a station. The Add Station message element is accompanied by technology specific binding information element(s) which may include security parameters. Consequently, the security parameters MUST be applied by the WTP for the station.

After station policy has been delivered to the WTP through the Add Station message element, an AC MAY change any policies by sending a modified Add Station message element. When a WTP receives an Add Station message element for an existing station, it MUST override any existing state for the station.



Type: 8 for Add Station

Length: >= 8

Radio ID: An 8-bit value representing the radio

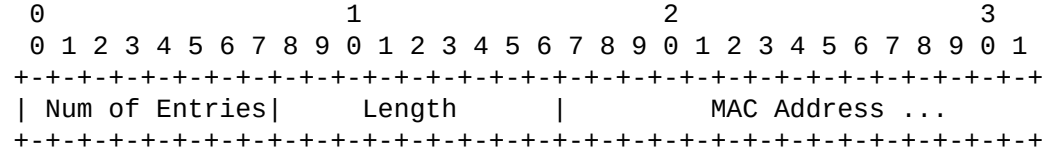
Length: The length of the MAC Address field.

MAC Address: The station's MAC Address

VLAN Name: An optional variable length UTF-8 encoded string containing the VLAN Name on which the WTP is to locally bridge user data. Note this field is only valid with WTPs configured in Local MAC mode.

4.6.9. Add Static MAC ACL Entry

The Add Static MAC ACL Entry message element is used by an AC to add a permanent ACL entry on a WTP, ensuring that the WTP no longer provides any service to the MAC addresses provided in the message. The MAC Addresses provided in this message element are expected to be saved in non-volative memory on the WTP.



Type: 9 for Add Static MAC ACL Entry

Length: >= 8

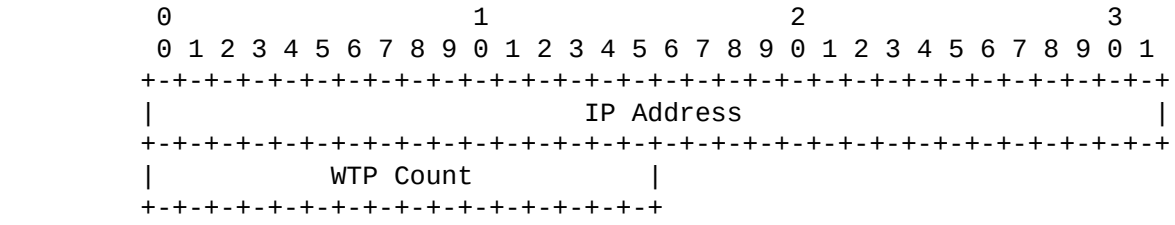
Num of Entries: The number of instances of the Type/MAC Addresses fields in the array.

Length: The length of the MAC Address field.

MAC Address: MAC Addresses to add to the permanent ACL.

4.6.10. CAPWAP Control IPv4 Address

The CAPWAP Control IPv4 Address message element is sent by the AC to the WTP during the discovery process and is used by the AC to provide the interfaces available on the AC, and the current number of WTPs connected. When multiple CAPWAP Control IPV4 Address message elements are returned, the WTP SHOULD perform load balancing across the multiple interfaces.



Type: 10 for CAPWAP Control IPv4 Address

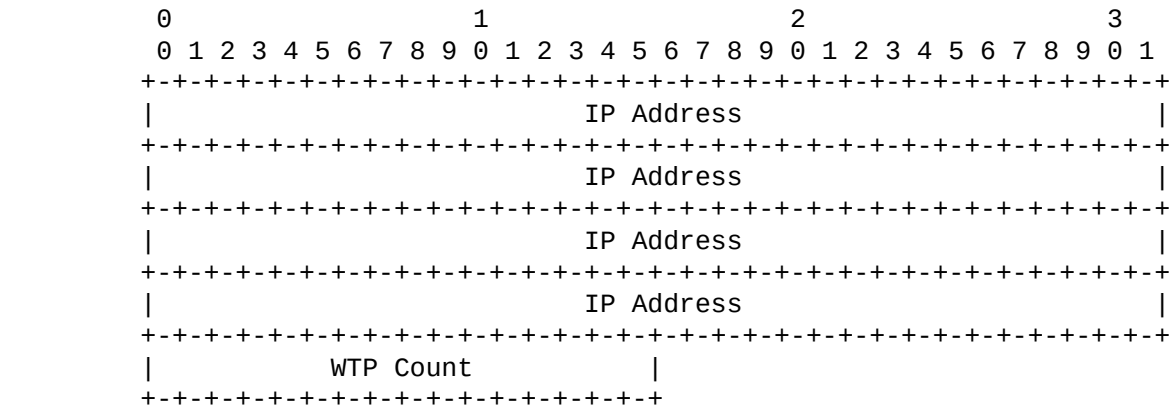
Length: 6

IP Address: The IP Address of an interface.

WTP Count: The number of WTPs currently connected to the interface.

4.6.11. CAPWAP Control IPv6 Address

The CAPWAP Control IPv6 Address message element is sent by the AC to the WTP during the discovery process and is used by the AC to provide the interfaces available on the AC, and the current number of WTPs connected. This message element is useful for the WTP to perform load balancing across multiple interfaces.



Type: 11 for CAPWAP Control IPv6 Address

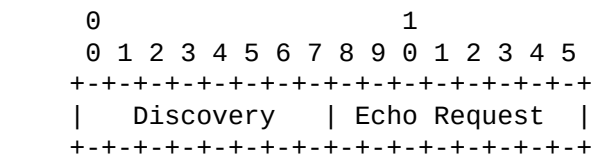
Length: 18

IP Address: The IP Address of an interface.

WTP Count: The number of WTPs currently connected to the interface.

4.6.12. CAPWAP Timers

The CAPWAP Timers message element is used by an AC to configure CAPWAP timers on a WTP.



Type: 12 for CAPWAP Timers

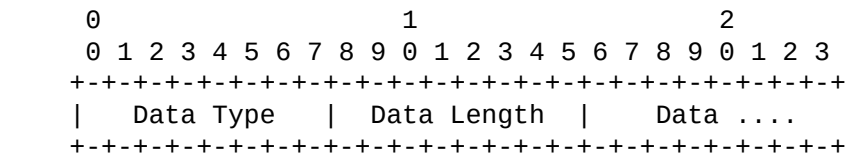
Length: 2

Discovery: The number of seconds between CAPWAP Discovery messages, when the WTP is in the discovery phase.

Echo Request: The number of seconds between WTP Echo Request CAPWAP messages. The default value for this message element is specified in Section 4.7.5.

4.6.13. Data Transfer Data

The Data Transfer Data message element is used by the WTP to provide information to the AC for debugging purposes.



Type: 13 for Data Transfer Data

Length: >= 3

Data Type: An 8-bit value the type of information being sent. The following values are supported:

1 - WTP Crash Data

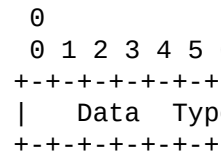
2 - WTP Memory Dump

Data Length: Length of data field.

Data: Debug information.

4.6.14. Data Transfer Mode

The Data Transfer Mode message element is used by the WTP to indicate the type of data transfer information it is sending to the AC for debugging purposes.



Type: 14 for Data Transfer Mode

Length: 1

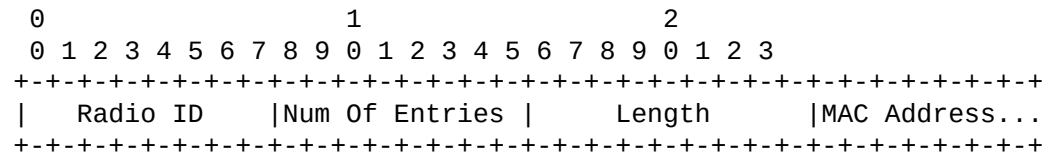
Data Type: An 8-bit value the type of information being requested. The following values are supported:

1 - WTP Crash Data

2 - WTP Memory Dump

4.6.15. Decryption Error Report

The Decryption Error Report message element value is used by the WTP to inform the AC of decryption errors that have occurred since the last report. Note that this error reporting mechanism is not used if encryption and decryption services are provided in the AC.



Type: 15 for Decryption Error Report

Length: >= 9

Radio ID: The Radio Identifier refers to an interface index on the WTP.

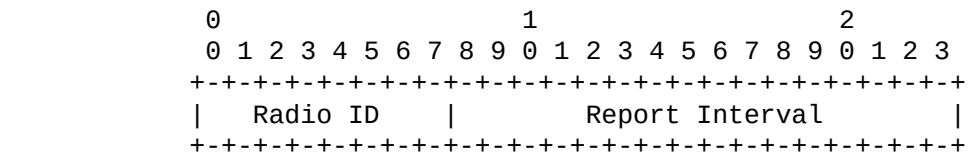
Num of Entries: The number of instances of the Type/MAC Addresses fields in the array.

Length: The length of the MAC Address field.

MAC Address: MAC addresses of the station that has caused decryption errors.

4.6.16. Decryption Error Report Period

The Decryption Error Report Period message element value is used by the AC to inform the WTP how frequently it should send decryption error report messages. Note that this error reporting mechanism is not used if encryption and decryption services are provided in the AC.



Type: 16 for Decryption Error Report Period

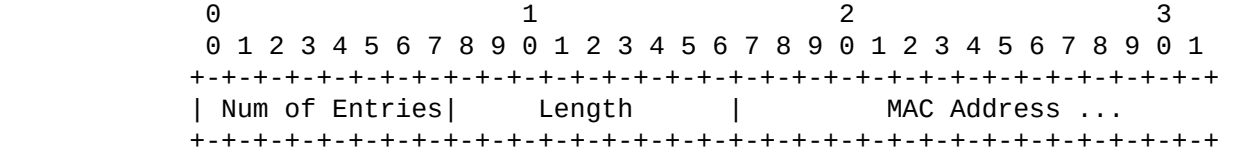
Length: 3

Radio ID: The Radio Identifier refers to an interface index on the WTP.

Report Interval: A 16-bit unsigned integer indicating the time, in seconds. The default value for this message element can be found in Section 4.8.8.

4.6.17. Delete MAC ACL Entry

The Delete MAC ACL Entry message element is used by an AC to delete a MAC ACL entry on a WTP, ensuring that the WTP provides service to the MAC addresses provided in the message.



Type: 17 for Delete MAC ACL Entry

Length: >= 8

Num of Entries: The number of instances of the Type/MAC Addresses fields in the array.

Length: The length of the MAC Address field.

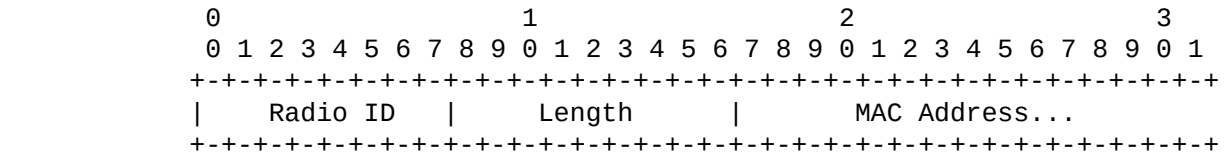
MAC Address: An array of MAC Addresses to delete from the ACL.

4.6.18. Delete Station

The Delete Station message element is used by the AC to inform a WTP that it should no longer provide service to a particular station. The WTP MUST terminate service to the station immediately upon receiving this message element.

The transmission of a Delete Station message element could occur for various reasons, including for administrative reasons, or if the station has roamed to another WTP.

The Delete Station message element MAY be sent by the WTP, in the WTP Event Request message, to inform the AC that a particular station is no longer being provided service. This could occur as a result of an Idle Timeout (see section 4.4.43), due to internal resource shortages or for some other reason.



Type: 18 for Delete Station

Length: >= 8

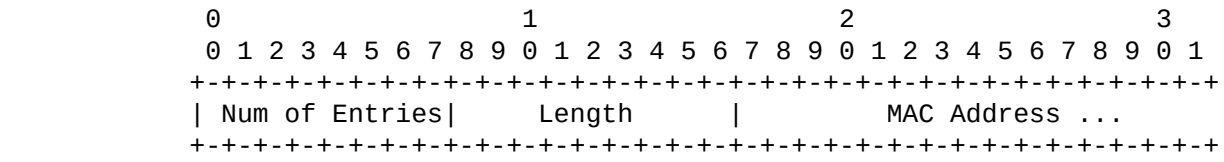
Radio ID: An 8-bit value representing the radio

Length: The length of the MAC Address field.

MAC Address: The station’s MAC Address

4.6.19. Delete Static MAC ACL Entry

The Delete Static MAC ACL Entry message element is used by an AC to delete a previously added static MAC ACL entry on a WTP, ensuring that the WTP provides service to the MAC addresses provided in the message.



Type: 19 for Delete Static MAC ACL Entry

Length: >= 8

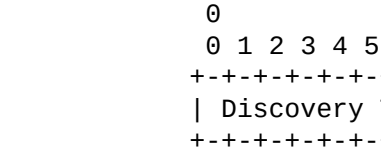
Num of Entries: The number of instances of the Type/MAC Addresses fields in the array.

Length: The length of the MAC Address field.

MAC Address: An array of MAC Addresses to delete from the static MAC ACL entry.

4.6.20. Discovery Type

The Discovery Type message element is used by the WTP to indicate how it has come to know about the existence of the AC to which it is sending the Discovery Request message.



Type: 20 for Discovery Type

Length: 1

Discovery Type: An 8-bit value indicating how the WTP discovered the AC. The following values are supported:

0 - Unknown

1 - Static Configuration

2 - DHCP

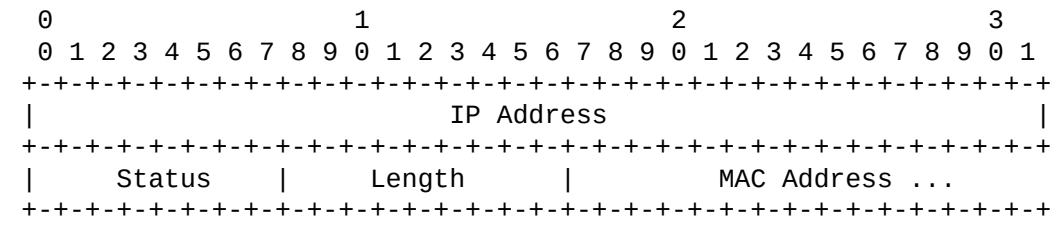
3 - DNS

4 - AC Referral (used when the AC was configured either through the AC IPv4 List or AC IPv6 List message element)

4.6.21. Duplicate IPv4 Address

The Duplicate IPv4 Address message element is used by a WTP to inform an AC that it has detected another IP device using the same IP address that the WTP is currently using.

The WTP MUST transmit this message element with the status set to 1 after it has detected a duplicate IP address. When the WTP detects that the duplicate IP address has been cleared, it MUSY send this message element with the status set to 0.



Type: 21 for Duplicate IPv4 Address

Length: >= 12

IP Address: The IP Address currently used by the WTP.

Status: The status of the duplicate IP address. The value MUST be set to 1 when a duplicate address is detected, and 0 when the duplicate address has been cleared.

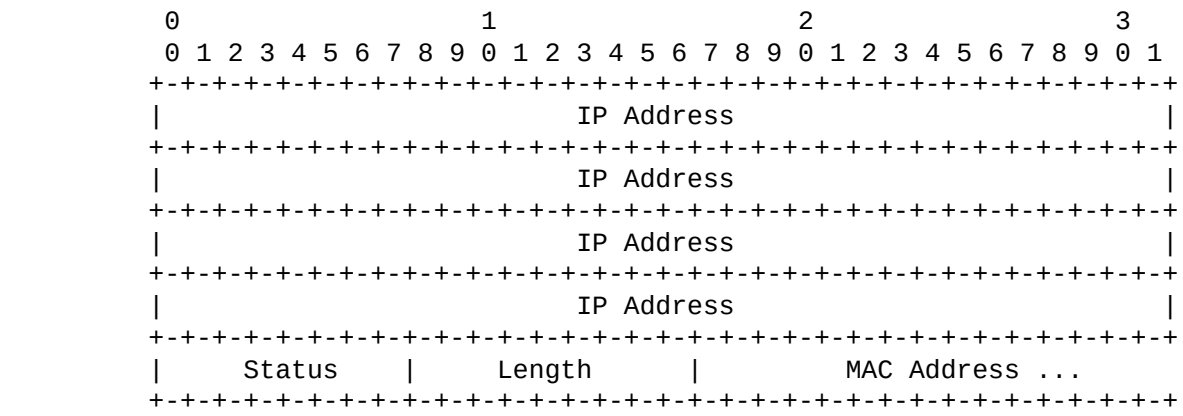
Length: The length of the MAC Address field.

MAC Address: The MAC Address of the offending device.

4.6.22. Duplicate IPv6 Address

The Duplicate IPv6 Address message element is used by a WTP to inform an AC that it has detected another host using the same IP address that the WTP is currently using.

The WTP MUST transmit this message element with the status set to 1 after it has detected a duplicate IP address. When the WTP detects that the duplicate IP address has been cleared, it MUST send this message element with the status set to 0.



Type: 23 for Duplicate IPv6 Address

Length: >= 24

IP Address: The IP Address currently used by the WTP.

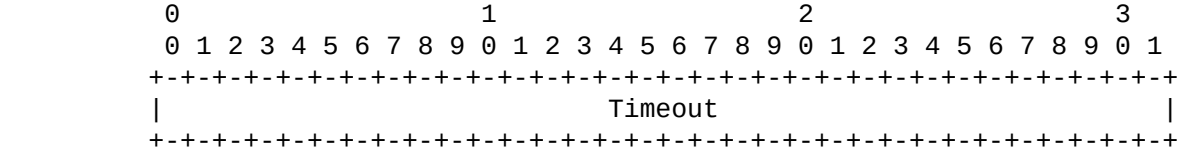
Status: The status of the duplicate IP address. The value MUST be set to 1 when a duplicate address is detected, and 0 when the duplicate address has been cleared.

Length: The length of the MAC Address field.

MAC Address: The MAC Address of the offending device.

4.6.23. Idle Timeout

The Idle Timeout message element is sent by the AC to the WTP to provide the idle timeout value that the WTP SHOULD enforce for its active stations. The value applies to all radios on the WTP.



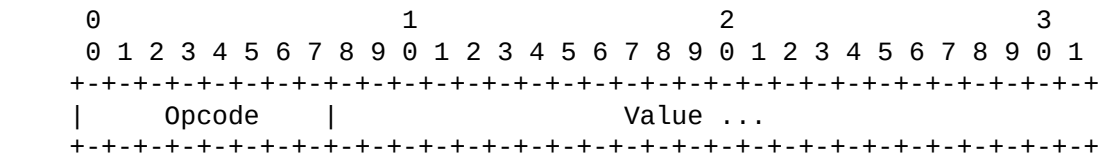
Type: 23 for Idle Timeout

Length: 4

Timeout: The current idle timeout to be enforced by the WTP. The default value for this message element is specified in Section 4.8.5.

4.6.24. Image Data

The Image Data message element is present in the Image Data Request message sent by the AC and contains the following fields.



Type: 24 for Image Data

Length: >= 1

Opcode: An 8-bit value representing the transfer opcode. The following values are supported:

1 - Image data is included

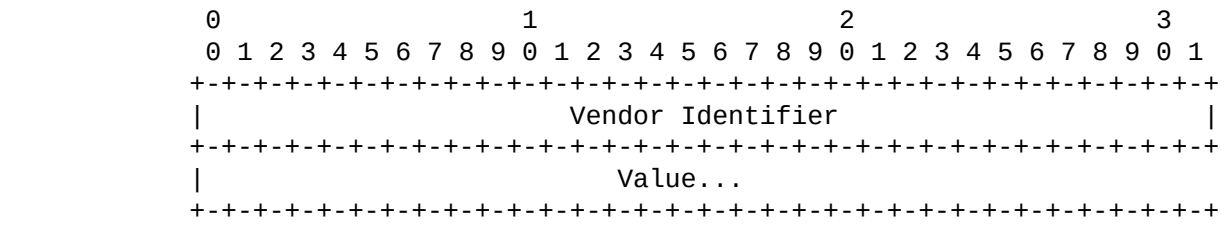
2 - Last Image Data Block is included (EOF)

5 - An error occurred. Transfer is aborted

Value: The Image Data field contains up to 1024 characters. If the block being sent is the last one, the Opcode is set to 2. The AC MAY opt to abort the data transfer by setting the Opcode to 5. When the Opcode is 5, the Value field has a zero length.

4.6.25. Image Identifier

The Image Identifier message element is sent by the AC to the WTP and is used to indicate the expected active software version that is to be run on the WTP. The value is a variable length UTF-8 encoded string, which is NOT zero terminated.



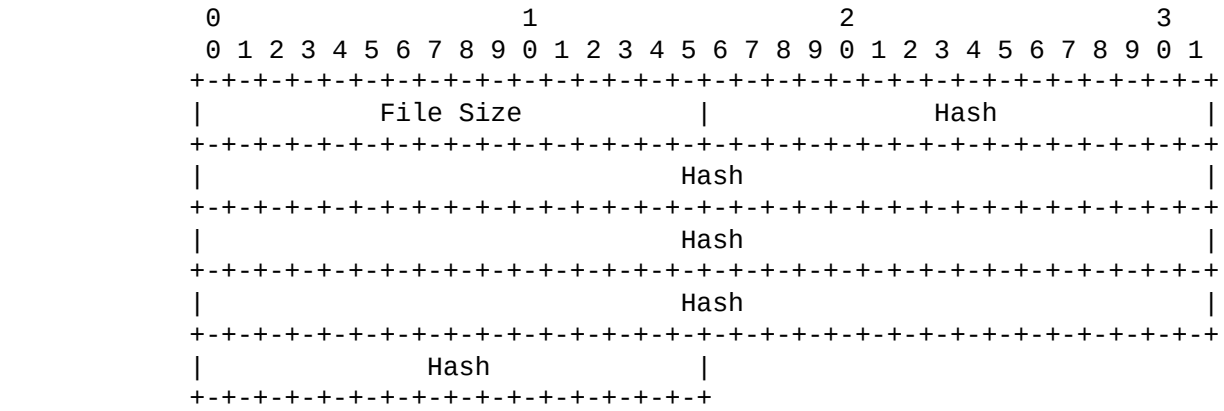
Type: 25 for Image Identifier

Length: >= 1

Value: A variable length UTF-8 encoded string containing the firmware identifier to be run on the WTP.

4.6.26. Image Information

The Image Information message element is present in the Image Data Response message sent by the AC to the WTP and contains the following fields.



Type: 26 for Image Information

Length: 18

File Size: A 16-bit value containing the size of the file that will be transfered by the AC to the WTP.

Hash: A 16 octet hash of the image. The hash is computed using MD5, using the following pseudo-code:

```
#include <md5.h>
CapwapCreateHash(char *hash, char *image, int image_len)
{
        MD_CTX context;

        MDInit (&context);
        MDUpdate (&context, buffer, len);
        MDFinal (hash, &context);
}
```

4.6.27. Initiate Download

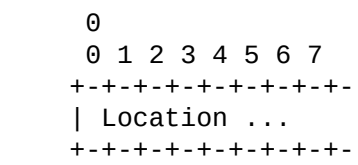
The Initiate Download message element is used by the AC to inform the WTP that the WTP SHOULD initiate a firmware upgrade. The WTP subsequently transmits an Image Data Request message which includes the Image Download message element. This message element does not contain any data.

Type: 27 for Initiate Download

Length: 0

4.6.28. Location Data

The Location Data message element is a variable length byte UTF-8 encoded string containing user defined location information (e.g. "Next to Fridge"). This information is configurable by the network administrator, and allows the WTP location to be determined. The string is not zero terminated.



Type: 28 for Location Data

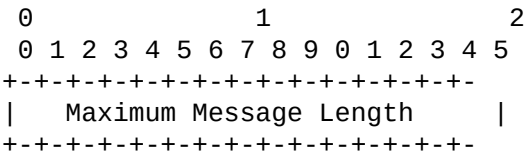
Length: > 0

Location: A non-zero terminated UTF-8 encoded string containing the WTP location.

4.6.29. Maximum Message Length

The Maximum Message Length message element is included in the Join Request message by the WTP to indicate the maximum CAPWAP message length that it supports to the AC. The Maximum Message Length

message element is optionally included in Join Response message by the AC to indicate the maximum CAPWAP message length that it supports to the WTP.



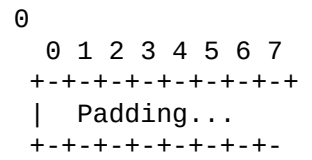
Type: 29 for Maximim Message Length

Length: 2

Maximum Message Length An 16-bit unsigned integer indicating the maximum message length.

4.6.30. MTU Discovery Padding

The MTU Discovery Padding message element is used as padding to perform MTU discovery, and MUST contain octets of value 0xFF, of any length



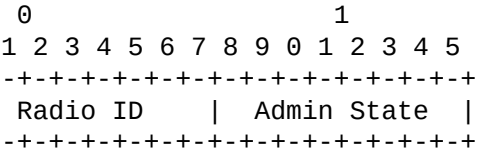
Type: 30 for MTU Discovery Padding

Length: variable

Pad: A variable length pad.

4.6.31. Radio Administrative State

The Radio Administrative State message element is used to communicate the state of a particular radio. The Radio Administrative State message element is sent by the AC to change the state of the WTP. The WTP saves the value, to ensure that it remains across WTP resets. The WTP communicates this message element during the configuration phase, in the Configuration Status Request message, to ensure that AC has the WTP radio current administrative state settings. The message element contains the following fields.



Type: 31 for Radio Administrative State

Length: 2

Radio ID: An 8-bit value representing the radio to configure. The Radio ID field MAY also include the value of 0xff, which is used to identify the WTP. If an AC wishes to change the administrative state of a WTP, it includes 0xff in the Radio ID field.

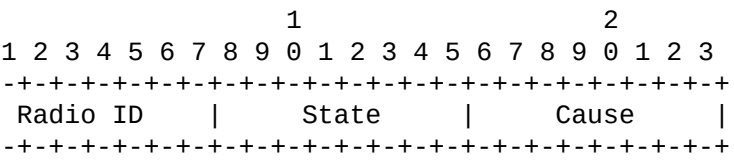
Admin State: An 8-bit value representing the administrative state of the radio. The default value for the Admin State field is listed in Section 4.8.1. The following values are supported:

1 - Enabled

2 - Disabled

4.6.32. Radio Operational State

The Radio Operational State message element is sent by the WTP to the AC to communicate a radio's operational state. This message element is included in the Configuration Update Response message by the WTP if it was requested to change the state of its radio, via the Radio Administrative State message element, but was unable to comply to the request. This message element is included in the Change State Event message when a WTP radio state was changed unexpectedly. This could occur due to a hardware failure. Note that the operational state setting is not saved on the WTP, and therefore does not remain across WTP resets. The value contains three fields, as shown below.



Type: 32 for Radio Operational State

Length: 3

Radio ID: The Radio Identifier refers to an interface index on the WTP. A value of 0xFF is invalid, as it is not possible to change the WTP's operational state.

State: An 8-bit boolean value representing the state of the radio. A value of one disables the radio, while a value of two enables it.

Cause: When a radio is inoperable, the cause field contains the reason the radio is out of service. The following values are supported:

0 - Normal

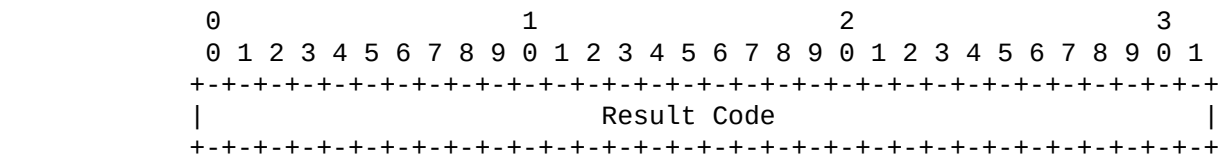
1 - Radio Failure

2 - Software Failure

3 - Administratively Set

4.6.33. Result Code

The Result Code message element value is a 32-bit integer value, indicating the result of the Request message corresponding to the Sequence Number included in the Response message.



Type: 33 for Result Code

Length: 4

Result Code: The following values are defined:

0 Success

1 Failure (AC List message element MUST be present)

2 Success (NAT detected)

3 Join Failure (unspecified)

4 Join Failure (Resource Depletion)

5 Join Failure (Unknown Source)

6 Join Failure (Incorrect Data)

7 Join Failure (Session ID already in use)

8 Join Failure (WTP Hardware not supported)

9 Join Failure (Binding Not Supported)

10 Reset Failure (Unable to Reset)

11 Reset Failure (Firmware Write Error)

12 Configuration Failure (Unable to Apply Requested Configuration - Service Provided Anyhow)

13 Configuration Failure (Unable to Apply Requested Configuration - Service Not Provided)

14 Image Data Error (Invalid Checksum)

15 Image Data Error (Invalid Data Length)

16 Image Data Error (Other Error)

17 Image Data Error (Image Already Present)

18 Message Unexpected (Invalid in current state)

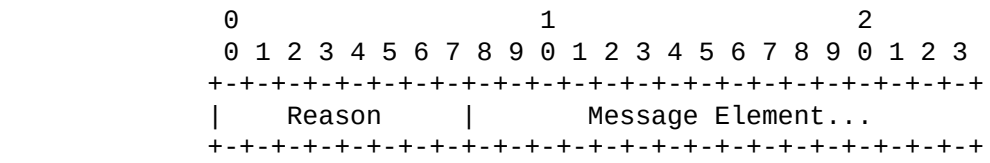
19 Message Unexpected (Unrecognized Request)

20 Failure - Missing Mandatory Message Element

21 Failure - Unrecognized Message Element

4.6.34. Returned Message Element

The Returned Message Element is sent by the WTP in the Change State Event Request message to communicate to the AC which message elements in the Configuration Status Response it was unable to apply locally. The Returned Message Element message element contains a result code indicating the reason that the configuration could not be applied, and encapsulates the failed message element.



Type: 34 for Returned Message Element

Length: >= 1

Reason: The reason why the configuration in the offending message element could not be applied by the WTP.

1 - Unknown Message Element

2 - Unsupported Message Element

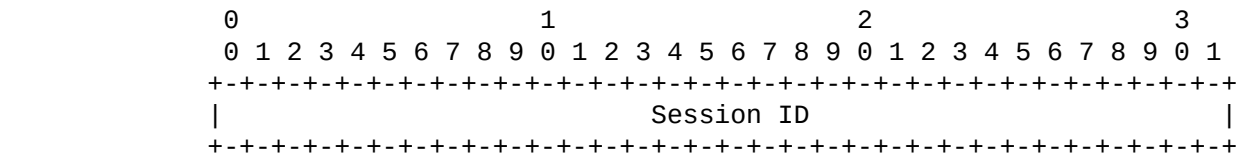
3 - Unknown Message Element Value

4 - Unsupported Message Element Value

Message Element: The Message Element field encapsulates the message element sent by the AC in the Configuration Status Response message that caused the error.

4.6.35. Session ID

The Session ID message element value contains a randomly generated unsigned 32-bit integer.



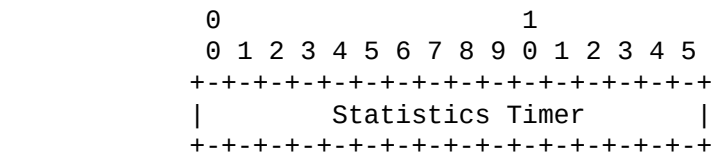
Type: 35 for Session ID

Length: 16

Session ID: A 32-bit unsigned integer used as a random session identifier

4.6.36. Statistics Timer

The Statistics Timer message element value is used by the AC to inform the WTP of the frequency with which it expects to receive updated statistics.



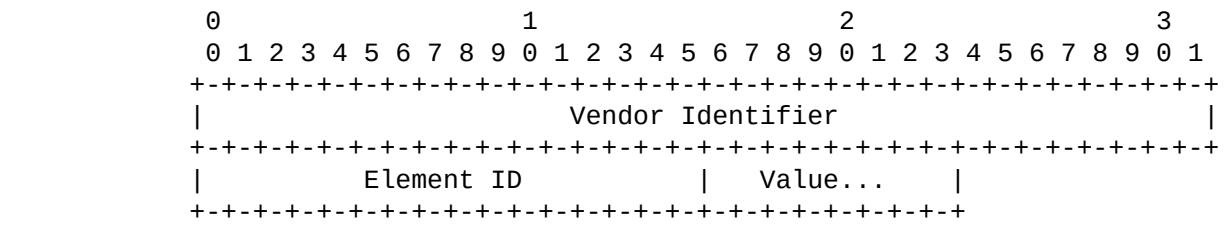
Type: 36 for Statistics Timer

Length: 2

Statistics Timer: A 16-bit unsigned integer indicating the time, in seconds. The default value for this timer is specified in Section 4.7.12.

4.6.37. Vendor Specific Payload

The Vendor Specific Payload message element is used to communicate vendor specific information between the WTP and the AC. The message element uses the following format:



Type: 37 for Vendor Specific

Length: >= 7

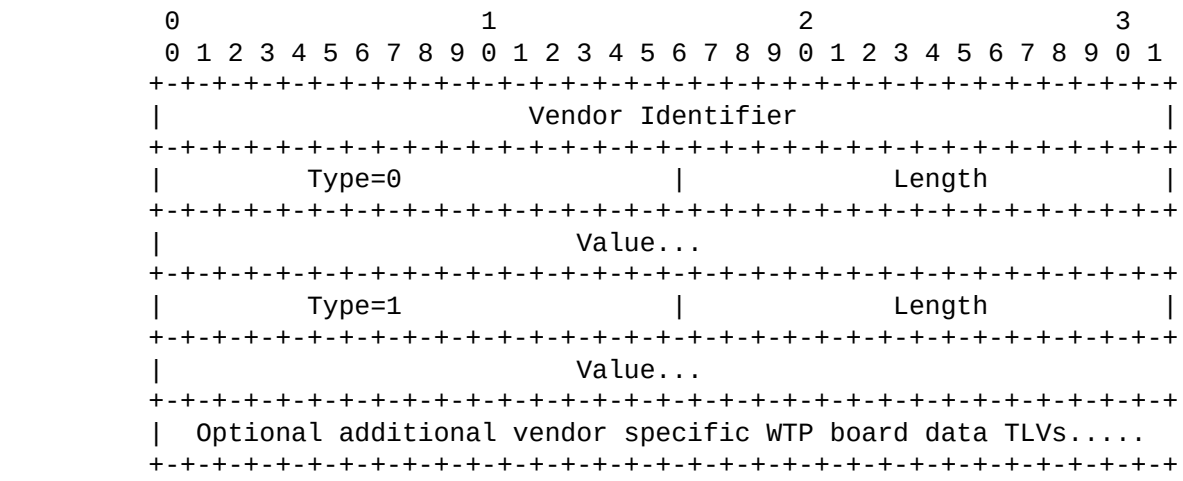
Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes" [14]

Element ID: A 16-bit Element Identifier which is managed by the vendor.

Value: The value associated with the vendor specific element.

4.6.38. WTP Board Data

The WTP Board Data message element is sent by the WTP to the AC and contains information about the hardware present.



Type: 38 for WTP Board Data

Length: >=14

Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes"

Type: The following values are supported:

0 - WTP Model Number: The WTP Model Number MUST be included in the WTP Board Data message element.

1 - WTP Serial Number: The WTP Serial Number MUST be included in the WTP Board Data message element.

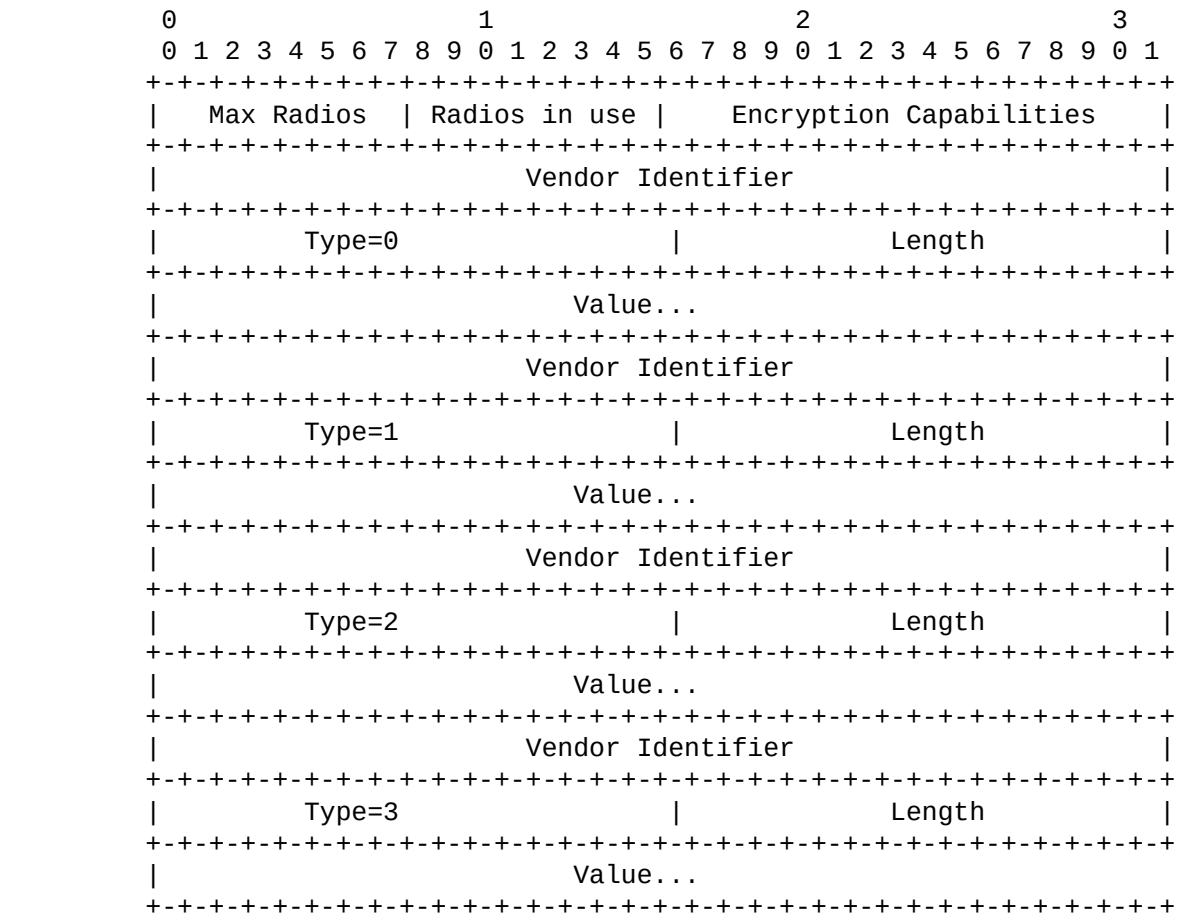
2 - Board ID: A hardware identifier, which MAY be included in the WTP Board Data mesage element.

3 - Board Revision A revision number of the board, which MAY be included in the WTP Board Data message element.

4 - Base MAC Addres The WTP’s Base MAC Address, which MAY be assigned to the primary Ethernet interface.

4.6.39. WTP Descriptor

The WTP Descriptor message element is used by a WTP to communicate its current hardware and software (firmware) configuration. The value contains the following fields.



Type: 39 for WTP Descriptor

Length: >= 31

Max Radios: An 8-bit value representing the number of radios (where each radio is identified via the Radio ID field) supported by the WTP.

Radios in use: An 8-bit value representing the number of radios in use in the WTP.

Encryption Capabilities: This 16-bit field is used by the WTP to communicate its capabilities to the AC. A WTP that does not have any encryption capabilities sets this field to zero (0). Refer to the specific wireless binding for further specification of the Encryption Capabilities field.

Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes".

Type: The following values are supported. The Hardware Version, Active Software Version, and Boot Version values MUST be included. Zero or more Other Software Version values MAY be included.

0 - Hardware Version: The WTP hardware version number.

1 - Active Software Version: The WTP running software version number.

2 - Boot Version: The WTP boot loader version number.

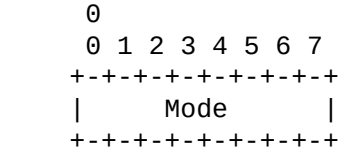
3 - Other Software Version: The WTP non-running software (firmware) version number.

Length: Length of vendor specific encoding of WTP information.

Value: Vendor specific data of WTP information encoded in the UTF-8 format.

4.6.40. WTP Fallback

The WTP Fallback message element is sent by the AC to the WTP to enable or disable automatic CAPWAP fallback in the event that a WTP detects its preferred AC, and is not currently connected to it.



Type: 40 for WTP Fallback

Length: 1

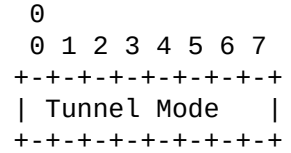
Mode: The 8-bit value indicates the status of automatic CAPWAP fallback on the WTP. When enabled, if the WTP detects that its primary AC is available, and that the WTP is not connected to the primary AC, the WTP SHOULD automatically disconnect from its current AC and reconnect to its primary AC. If disabled, the WTP will only reconnect to its primary AC through manual intervention (e.g., through the Reset Request message). The default value for this field is specified in Section 4.8.10. The following values are supported:

1 - Enabled

2 - Disabled

4.6.41. WTP Frame Tunnel Mode

The WTP Frame Tunnel Mode message element allows the WTP to communicate the tunneling modes of operation which it supports to the AC. A WTP that advertises support for all types allows the AC to select which type will be used, based on its local policy.



Type: 41 for WTP Frame Tunnel Mode

Length: 1

Frame Tunnel Mode: The Frame Tunnel mode specifies the tunneling modes for station data that are supported by the WTP. The following values are supported:

1 - Local Bridging: When Local Bridging is used, the WTP does not tunnel user traffic to the AC; all user traffic is locally bridged. This value MUST NOT be used when the WTP MAC Type is set to Split-MAC.

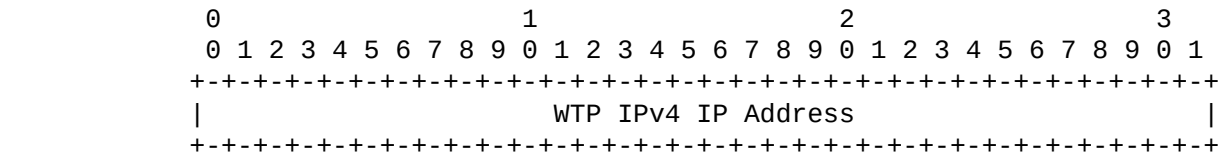
2 - 802.3 Frame Tunnel Mode: The 802.3 Frame Tunnel Mode requires the WTP and AC to encapsulate all user payload as native IEEE 802.3 frames (see Section 4.4). All user traffic is tunneled to the AC. This value MUST NOT be used when the WTP MAC Type is set to Split-MAC.

4 - Native Frame Tunnel Mode: Native Frame Tunnel mode requires the WTP and AC to encapsulate all user payloads as native wireless frames, as defined by the wireless binding (see for example Section 4.4).

7 - All: The WTP is capable of supporting all frame tunnel modes.

4.6.42. WTP IPv4 IP Address

The WTP IPv4 address is used to perform NAT detection.



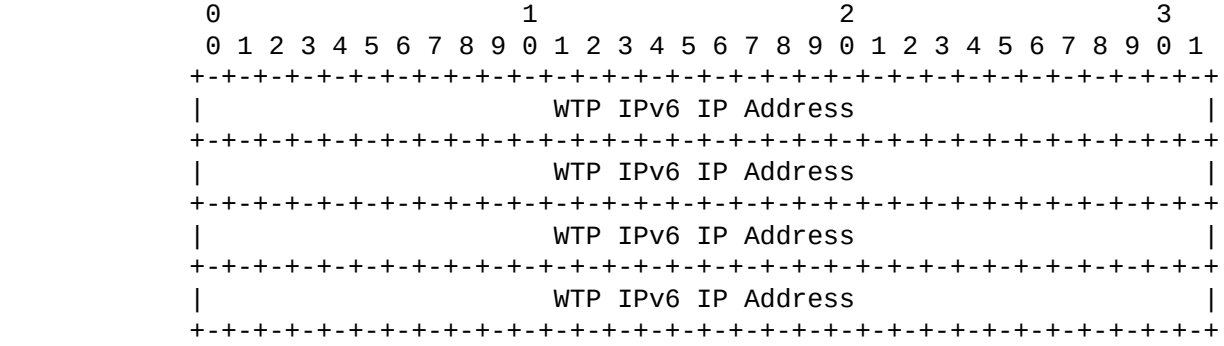
Type: 42 for WTP IPv4 IP Address

Length: 4

WTP IPv4 IP Address: The IPv4 address from which the WTP is sending packets. This field is used for NAT detection.

4.6.43. WTP IPv6 IP Address

The WTP IPv6 address is used to perform NAT detection (e.g., IPv4 to IPv6 NAT to help with technology transition).



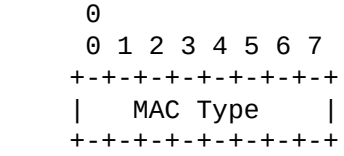
Type: 43 for WTP IPv6 IP Address

Length: 32

WTP IPv6 IP Address: The IPv6 address from which the WTP is sending packets. This field is used for NAT detection.

4.6.44. WTP MAC Type

The WTP MAC-Type message element allows the WTP to communicate its mode of operation to the AC. A WTP that advertises support for both modes allows the AC to select the mode to use, based on local policy.



Type: 44 for WTP MAC Type

Length: 1

MAC Type: The MAC mode of operation supported by the WTP. The following values are supported

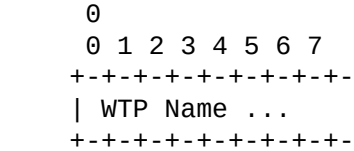
0 - Local-MAC: Local-MAC is the default mode that MUST be supported by all WTPs.

1 - Split-MAC: Split-MAC support is optional, and allows the AC to receive and process native wireless frames.

2 - Both: WTP is capable of supporting both Local-MAC and Split-MAC.

4.6.45. WTP Name

The WTP Name message element is a variable length byte UTF-8 encoded string. The string is not zero terminated.



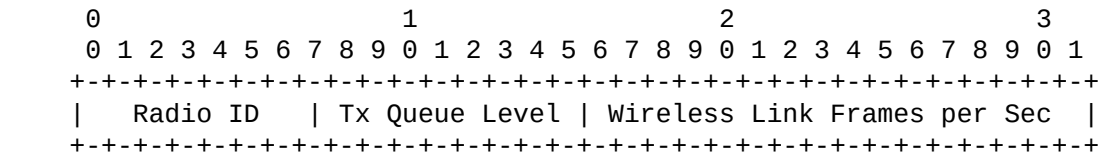
Type: 45 for WTP Name

Length: variable

WTP Name: A non-zero terminated UTF-8 encoded string containing the WTP name.

4.6.46. WTP Operational Statistics

The WTP Operational Statistics message element is sent by the WTP to the AC to provide statistics related to the operation of the WTP.



Type: 46 for WTP Operational Statistics

Length: 4

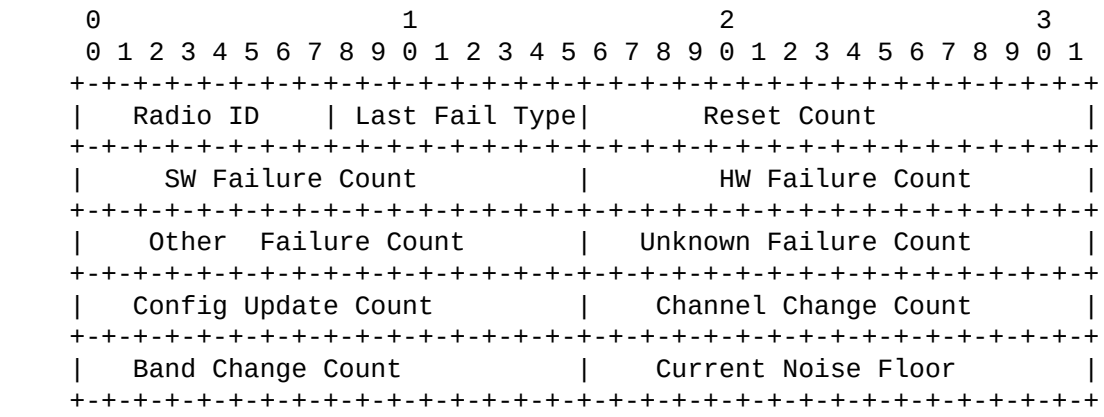
Radio ID: The radio ID of the radio to which the statistics apply.

Wireless Transmit Queue Level: The percentage of Wireless Transmit queue utilization, calculated as the sum of utilized transmit queue lengths divided by the sum of maximum transmit queue lengths, multiplied by 100. The Wireless Transmit Queue Level is representative of congestion conditions over wireless interfaces between the WTP and stations.

Wireless Link Frames per Sec: The number of frames transmitted or received per second by the WTP over the air interface.

4.6.47. WTP Radio Statistics

The WTP Radio Statistics message element is sent by the WTP to the AC to communicate statistics on radio behavior and reasons why the WTP radio has been reset.



Type: 47 for WTP Radio Statistics

Length: 20

Radio ID: The radio ID of the radio to which the statistics apply.

Last Failure Type: The last WTP failure. The following values are supported:

0 - Statistic Not Supported

1 - Software Failure

2 - Hardware Failure

3 - Other Failure

255 - Unknown (e.g., WTP doesn’t keep track of info)

Reset Count: The number of times that that the radio has been reset.

SW Failure Count: The number of times that the radio has failed due to software related reasons.

HW Failure Count: The number of times that the radio has failed due to hardware related reasons.

Other Failure Count: The number of times that the radio has failed due to known reasons, other than software or hardware failure.

Unknown Failure Count: The number of times that the radio has failed for unknown reasons.

Config Update Count: The number of times that the radio configuration has been updated.

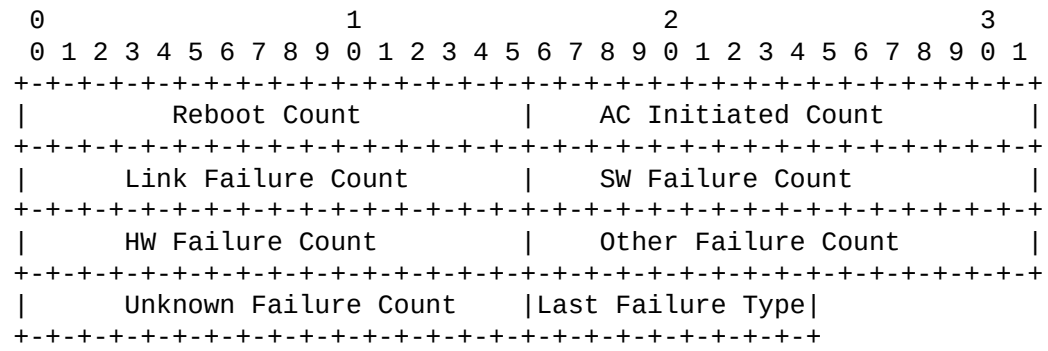
Channel Change Count: The number of times that the radio channel has been changed.

Band Change Count: The number of times that the radio has changed frequency bands.

Current Noise Floor: A signed integer which indicates the noise floor of the radio receiver in units of dBm.

4.6.48. WTP Reboot Statistics

The WTP Reboot Statistics message element is sent by the WTP to the AC to communicate reasons why WTP reboots have occurred.



Type: 48 for WTP Reboot Statistics

Length: 15

Reboot Count: The number of reboots that have occurred due to a WTP crash. A value of 65535 implies that this information is not available on the WTP.

AC Initiated Count: The number of reboots that have occurred at the request of a CAPWAP protocol message, such as a change in configuration that required a reboot or an explicit CAPWAP protocol reset request. A value of 65535 implies that this information is not available on the WTP.

Link Failure Count: The number of times that a CAPWAP protocol connection with an AC has failed due to link failure.

SW Failure Count: The number of times that a CAPWAP protocol connection with an AC has failed due to software related reasons.

HW Failure Count: The number of times that a CAPWAP protocol connection with an AC has failed due to hardware related reasons.

Other Failure Count: The number of times that a CAPWAP protocol connection with an AC has failed due to known reasons, other than AC initiated, link, SW or HW failure.

Unknown Failure Count: The number of times that a CAPWAP protocol connection with an AC has failed for unknown reasons.

Last Failure Type: The failure type of the most recent WTP failure. The following values are supported:

0 - Not Supported

1 - AC Initiated (see Section 9.2)

2 - Link Failure

3 - Software Failure

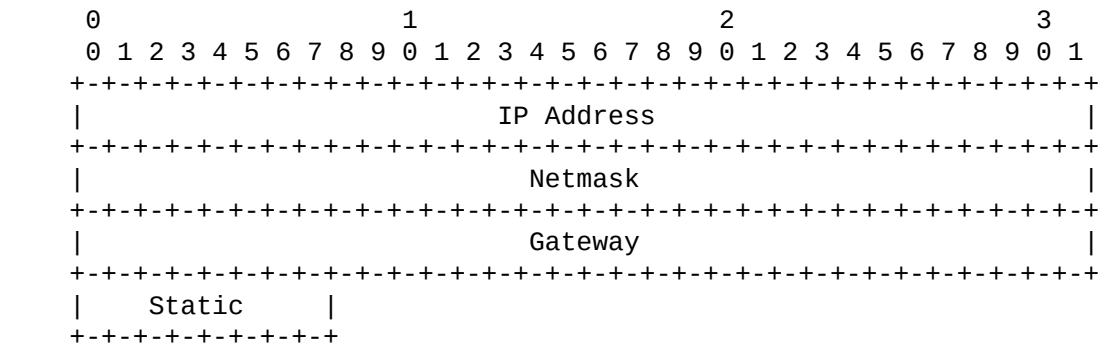
4 - Hardware Failure

5 - Other Failure

255 - Unknown (e.g., WTP doesn't keep track of info)

4.6.49. WTP Static IP Address Information

The WTP Static IP Address Information message element is used by an AC to configure or clear a previously configured static IP address on a WTP.



Type: 49 for WTP Static IP Address Information

Length: 13

IP Address: The IP Address to assign to the WTP. This field is only valid if the static field is set to one.

Netmask: The IP Netmask. This field is only valid if the static field is set to one.

Gateway: The IP address of the gateway. This field is only valid if the static field is set to one.

Netmask: The IP Netmask. This field is only valid if the static field is set to one.

Static: An 8-bit boolean stating whether the WTP should use a static IP address or not. A value of zero disables the static IP address, while a value of one enables it.

4.7. CAPWAP Protocol Timers

This section contains the CAPWAP timers.

4.7.1. ChangeStatePendingTimer

The maximum time, in seconds, the AC will wait for the Change State Event Request from the WTP after having transmitted a successful Configuration Status Response message. The default value is 25 seconds.

4.7.2. DataChannelDeadInterval

The minimum time, in seconds, a WTP MUST wait without having received a Data Channel Keep Alive packet before the destination for the Data Channel Keep Alive packets may be considered dead. The value of this

timer MUST be no less than 2*DataChannelKeepAlive seconds and no greater that 240 seconds.

Default: 5

4.7.3. DiscoveryInterval

The minimum time, in seconds, that a WTP MUST wait after receiving a Discovery Response message, before initiating a DTLS handshake.

Default: 5

4.7.4. DTLSSessionDelete

The minimum time, in seconds, a WTP MUST wait for DTLS session deletion.

Default: 5

4.7.5. EchoInterval

The minimum time, in seconds, between sending Echo Request messages to the AC with which the WTP has joined.

Default: 30

4.7.6. MaxDiscoveryInterval

The maximum time allowed between sending Discovery Request messages, in seconds. This value MUST be no less than 2 seconds and no greater than 180 seconds.

Default: 20 seconds.

4.7.7. MaxFailedDTLSSessionRetry

The maximum number of failed DTLS session establishment attempts before the CAPWAP device enters a silent period.

Default: 3.

4.7.8. NeighborDeadInterval

The minimum time, in seconds, a WTP MUST wait without having received an Echo Response message to its Echo Request message, before the destination for the Echo Request may be considered dead. This value MUST be no less than 2*EchoInterval seconds and no greater than 240 seconds.

Default: 60

4.7.9. ResponseTimeout

The minimum time, in seconds, in which the WTP or AC MUST respond to a CAPWAP Request message.

Default: 1

4.7.10. RetransmitInterval

The minimum time, in seconds, in which a non-acknowledged CAPWAP packet will be retransmitted.

Default: 3

4.7.11. SilentInterval

For a WTP, this is the minimum time, in seconds, a WTP MUST wait before it MAY again send Discovery Request messages or attempt to a establish DTLS session. For an AC, this is the minimum time, in seconds, during which the AC SHOULD ignore all CAPWAP and DTLS packets received from the WTP that is in the Sulking state.

Default: 30

4.7.12. StatisticsTimer

The default Statistics Interval is 120 seconds.

4.7.13. WaitDTLS

The maximum time, in seconds, a WTP MUST wait without having received a DTLS Handshake message from an AC. This timer MUST be greater than 30 seconds.

Default: 60

4.7.14. WaitJoin

The maximum time, in seconds, after which the DTLS session has been established that the AC will wait before receiving a Join Request message. This timer MUST be greater than 30 seconds.

Default: 60

4.8. CAPWAP Protocol Variables

A WTP or AC that implements the CAPWAP Discovery phase MUST allow for the following variables to be configured by system management; default values are specified, making explicit configuration unnecessary in many cases. If the default values are explicitly overriden by the AC, the WTP MUST save the values sent by the AC.

4.8.1. AdminState

The default Administrative State value is enabled (1).

4.8.2. DiscoveryCount

The number of Discovery Request messages transmitted by a WTP to a single AC. This is a monotonically increasing counter.

4.8.3. FailedDTLSAuthFailCount

The number of failed DTLS session establishment attempts due to authentication failures.

4.8.4. FailedDTLSSessionCount

The number of failed DTLS session establishment attempts.

4.8.5. IdleTimeout

The default Idle Timeout is 300 seconds.

4.8.6. MaxDiscoveries

The maximum number of Discovery Request messages that will be sent after a WTP boots.

Default: 10

4.8.7. MaxRetransmit

The maximum number of retransmissions for a given CAPWAP packet before the link layer considers the peer dead.

Default: 5

4.8.8. ReportInterval

The default Report Interval is 120 seconds.

4.8.9. RetransmitCount

The number of retransmissions for a given CAPWAP packet. This is a monotonically increasing counter.

4.8.10. WTPFallBack

The default WTP Fallback value is enabled (1).

4.9. WTP Saved Variables

In addition to the values defined in Section 4.8, the following values SHOULD be saved on the WTP in non-volatile memory. CAPWAP wireless bindings MAY define additional values that SHOULD be stored on the WTP.

4.9.1. AdminRebootCount

The number of times the WTP has rebooted administratively, defined in Section 4.6.48.

4.9.2. FrameEncapType

For WTPs that support multiple Frame Encapsulation Types, it is useful to save the value configured by the AC. The Frame Encapsulation Type is defined in Section 4.6.41.

4.9.3. LastRebootReason

The reason why the WTP last rebooted, defined in Section 4.6.48.

4.9.4. MacType

For WTPs that support multiple MAC Types, it is useful to save the value configured by the AC. The MACType is defined in Section 4.6.44.

4.9.5. PreferredACs

The preferred ACs, with the index, defined in Section 4.6.5.

4.9.6. RebootCount

The number of times the WTP has rebooted, defined in Section 4.6.48.

4.9.7. Static ACL Table

The static ACL table saved on the WTP, as configured by the Add Static MAC ACL Entry message element, see Section 4.6.9.

4.9.8. Static IP Address

The static IP Address assigned to the WTP, as configured by the WTP Static IP Address Information message element (see Section 4.6.49).

4.9.9. WTPLinkFailureCount

The number of times the link to the AC has failed, see Section 4.6.48.

4.9.10. WTPLocation

The WTP Location, defined in Section 4.6.28.

4.9.11. WTPName

The WTP Name, defined in Section 4.6.45.

5. CAPWAP Discovery Operations

The Discovery messages are used by a WTP to determine which ACs are available to provide service, and the capabilities and load of the ACs.

5.1. Discovery Request Message

The Discovery Request message is used by the WTP to automatically discover potential ACs available in the network. The Discovery Request message provides ACs with the primary capabilities of the WTP. A WTP must exchange this information to ensure subsequent exchanges with the ACs are consistent with the WTP's functional characteristics.

Discovery Request messages MUST be sent by a WTP in the Discover state after waiting for a random delay less than MaxDiscoveryInterval, after a WTP first comes up or is (re)initialized. A WTP MUST send no more than the maximum of MaxDiscoveries Discovery Request messages, waiting for a random delay less than MaxDiscoveryInterval between each successive message.

This is to prevent an explosion of WTP Discovery Request messages. An example of this occurring is when many WTPs are powered on at the same time.

Discovery Request messages MUST be sent by a WTP when no Echo Response messages are received for NeighborDeadInterval and the WTP returns to the Idle state. Discovery Request messages are sent after NeighborDeadInterval. They MUST be sent after waiting for a random delay less than MaxDiscoveryInterval. A WTP MAY send up to a maximum of MaxDiscoveries Discovery Request messages, waiting for a random delay less than MaxDiscoveryInterval between each successive message.

If a Discovery Response message is not received after sending the maximum number of Discovery Request messages, the WTP enters the Sulking state and MUST wait for an interval equal to SilentInterval before sending further Discovery Request messages.

Upon receiving a Discovery Request message, the AC will respond with a Discovery Response message sent to the address in the source address of the received Discovery Request message.

It is possible for the AC to receive a cleartext Discovery Request message while a DTLS session is already active with the WTP. This is most likely the case if the WTP has rebooted, perhaps due to a software or power failure, but could also be caused by a DoS attack. In such cases, any WTP state, including the state machine instance,

MUST NOT be cleared until another DTLS session has been successfully established, communicated via the DTLSSessionEstablished DTLS notification (see Section 2.3.2.2).

The binding specific WTP Radio Information message element (see Section 2.1) is included in the Discovery Request message to advertise WTP support for one or more CAPWAP bindings.

The Discovery Request message is sent by the WTP when in the Discovery State. The AC does not transmit this message.

The following message elements MUST be included in the Discovery Request message:

- Discovery Type, see Section 4.6.20
- WTP Board Data, see Section 4.6.38
- WTP Descriptor, see Section 4.6.39
- WTP Frame Tunnel Mode, see Section 4.6.41
- WTP MAC Type, see Section 4.6.44
- WTP Radio Information message element(s)that the WTP supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1).

5.2. Discovery Response Message

The Discovery Response message provides a mechanism for an AC to advertise its services to requesting WTPs.

When a WTP receives a Discovery Response message, it MUST wait for an interval not less than DiscoveryInterval for receipt of additional Discovery Response messages. After the DiscoveryInterval elapses, the WTP enters the DTLS-Init state and selects one of the ACs that sent a Discovery Response message and send a DTLS Handshake to that AC.

One or more binding specific WTP Radio Information message elements (see Section 2.1) are included in the Discovery Request message to advertise AC support for the CAPWAP bindings. The AC MAY include only the bindings it shares in common with the WTP, known through the WTP Radio Information message elements received in the Discovery Request message, or it MAY include all of the bindings supported. The WTP MAY use the supported bindings in its AC decision process. Note that if the WTP joins an AC that does not support a specific

CAPWAP binding, service for that binding MUST NOT be provided by the WTP.

The Discovery Response message is sent by the AC when in the Idle State. The WTP does not transmit this message.

The following message elements MUST be included in the Discovery Response Message:

- AC Descriptor, see Section 4.6.1
- AC Name, see Section 4.6.4
- WTP Radio Information message element(s)that the AC supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1 for more information).
- One of the following message elements MUST be included in the Discovery Response Message:
  - CAPWAP Control IPv4 Address, see Section 4.6.10
  - CAPWAP Control IPv6 Address, see Section 4.6.11

5.3. Primary Discovery Request Message

The Primary Discovery Request message is sent by the WTP to determine whether its preferred (or primary) AC is available.

A Primary Discovery Request message is sent by a WTP when it has a primary AC configured, and is connected to another AC. This generally occurs as a result of a failover, and is used by the WTP as a means to discover when its primary AC becomes available. Since the WTP only has a single instance of the CAPWAP state machine, the Primary Discovery Request is sent by the WTP when in the Run State. The AC does not transmit this message.

The frequency of the Primary Discovery Request messages should be no more often than the sending of the Echo Request message.

Upon receipt of a Primary Discovery Request message, the AC responds with a Primary Discovery Response message sent to the address in the source address of the received Primary Discovery Request message.

The following message elements MUST be included in the Primary Discovery Request message.

- Discovery Type, see Section 4.6.20
- WTP Board Data, see Section 4.6.38
- WTP Descriptor, see Section 4.6.39
- WTP Frame Tunnel Mode, see Section 4.6.41
- WTP MAC Type, see Section 4.6.44
- WTP Radio Information message element(s)that the WTP supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1 for more information).

5.4. Primary Discovery Response

The Primary Discovery Response message enables an AC to advertise its availability and services to requesting WTPs that are configured to have the AC as its primary AC.

The Primary Discovery Response message is sent by an AC after receiving a Primary Discovery Request message.

When a WTP receives a Primary Discovery Response message, it may establish a CAPWAP protocol connection to its primary AC, based on the configuration of the WTP Fallback Status message element on the WTP.

The Primary Discovery Response message is sent by the AC when in the Idle State. The WTP does not transmit this message.

The following message elements MUST be included in the Primary Discovery Response message.

- AC Descriptor, see Section 4.6.1
- AC Name, see Section 4.6.4
- WTP Radio Information message element(s)that the AC supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1 for more information).

One of the following message elements MUST be included in the Discovery Response Message:

- CAPWAP Control IPv4 Address, see Section 4.6.10

- CAPWAP Control IPv6 Address, see Section 4.6.11

6. CAPWAP Join Operations

The Join Request message is used by a WTP to request service from an AC after a DTLS connection is established to that AC. The Join Response message is used by the the AC to indicate that it will or will not provide service.

6.1. Join Request

The Join Request message is used by a WTP to request service through the AC. A Join Request message is sent by a WTP after (optionally) receiving one or more Discovery Response messages, and completion of DTLS session establishment. When an AC receives a Join Request message it responds with a Join Response message.

Upon completion of the DTLS handshake, and receiving the DTLSEstablished notification, the WTP sends the Join Request message to the AC. When the AC is notified of the DTLS session establishment, it does not clear the WaitDTLS timer until it has received the Join Request message, at which time it sends a Join Response message to the WTP, indicating success or failure.

One or more WTP Radio Information message elements (see Section 2.1) are included in the Join Request to request service for the CAPWAP bindings by the AC. Including a binding that is unsupported by the AC will result in a failed Join Response.

If the AC rejects the Join Request, it sends a Join Response message with a failure indication and initiates an abort of the DTLS session via the DTLSAbort command.

If an invalid (i.e. malformed) Join Request message is received, the message MUST be silently discarded by the AC. No response is sent to the WTP. The AC SHOULD log this event.

The Join Request is sent by the WTP when in the Join State. The AC does not transmit this message.

The following message elements MUST be included in the Join Request message.

- Location Data, see Section 4.6.28
- WTP Board Data, see Section 4.6.38
- WTP Descriptor, see Section 4.6.39

- WTP Name, see Section 4.6.45
- Session ID, see Section 4.6.35
- WTP Frame Tunnel Mode, see Section 4.6.41
- WTP MAC Type, see Section 4.6.44
- WTP Radio Information message element(s)that the WTP supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1 for more information).

At least one of the following message element MUST be included in the Join Request message.

- WTP IPv4 IP Address, see Section 4.6.42
- WTP IPv6 IP Address, see Section 4.6.43

The following message element MAY be included in the Join Request message.

- Maximum Message Length, see Section 4.6.29
- WTP Reboot Statistics, see Section 4.6.48
- WTP IPv4 IP Address, see Section 4.6.42
- WTP IPv6 IP Address, see Section 4.6.43

6.2. Join Response

The Join Response message is sent by the AC to indicate to a WTP that it is capable and willing to provide service to the WTP.

The WTP, receiving a Join Response message, checks for success or failure. If the message indicates success, the WTP clears the WaitDTLS timer for the session and proceeds to the Configure state.

If the WaitDTLS Timer expires prior to reception of the Join Response message, the WTP MUST terminate the handshake, deallocate session state and initiate the DTLSAbort command.

If an invalid (malformed) Join Response message is received, the WTP SHOULD log an informative message detailing the error. This error MUST be treated in the same manner as AC non-responsiveness. The WaitDTLS timer will eventually expire, and the WTP MAY (if it is so configured) attempts to join a new AC.

If one of the WTP Radio Information message elements (see Section 2.1) in the Join Request message requested support for a CAPWAP binding which the AC does not support, the AC sets the Result Code message element to "Binding Not Supported".

The AC includes the Image Identifier message element to indicate the software version it expects the WTP to run. This information is used to determine whether the WTP MUST either change its currently running firmware image, or download a new version (see Section 9.1.1).

The Join Response message is sent by the AC when in the Join State. The WTP does not transmit this message.

The following message elements MAY be included in the Join Response message.

- AC IPv4 List, see Section 4.6.2
- AC IPv6 List, see Section 4.6.3
- Image Identifier, see Section 4.6.25
- Maximum Message Length, see Section 4.6.29

The following message elements MUST be included in the Join Response message.

- Result Code, see Section 4.6.33
- AC Descriptor, see Section 4.6.1
- AC Name, see Section 4.6.4
- WTP Radio Information message element(s)that the AC supports; These are defined by the individual link layer CAPWAP Binding Protocols (see Section 2.1).

One of the following message elements MUST be included in the Discovery Response Message:

- CAPWAP Control IPv4 Address, see Section 4.6.10
- CAPWAP Control IPv6 Address, see Section 4.6.11

7. Control Channel Management

The Control Channel Management messages are used by the WTP and AC to maintain a control communication channel. CAPWAP control messages, such as the WTP Event Request message sent from the WTP to the AC indicate to the AC that the WTP is operational. When such control messages are not being sent, the Echo Request and Echo Response messages are used to maintain the control communication channel.

7.1. Echo Request

The Echo Request message is a keep-alive mechanism for CAPWAP control messages.

Echo Request messages are sent periodically by a WTP in the Run state (see Section 2.3) to determine the state of the control connection between the WTP and the AC. The Echo Request message is sent by the WTP when the EchoInterval timer expires. The WTP MUST start its NeighborDeadInterval timer when the EchoInterval timer expires.

The Echo Request message is sent by the WTP when in the Run State. The AC does not transmit this message.

The Echo Request message carries no message elements.

When an AC receives an Echo Request message it responds with an Echo Response message.

7.2. Echo Response

The Echo Response message acknowledges the Echo Request message.

An Echo Response message is sent by an AC after receiving an EchoRequest message. After transmitting the Echo Response message, the AC SHOULD reset its EchoInterval timer. If another Echo Request message or other control message is not received by the AC when the timer expires, the AC SHOULD consider the WTP to be no longer reachable.

The Echo Response message is sent by the AC when in the Run State. The WTP does not transmit this message.

The Echo Response message carries no message elements.

When a WTP receives an Echo Response message it stops the NeighborDeadInterval timer, and initializes the EchoInterval to the configured value.

If the NeighborDeadInterval timer expires prior to receiving an Echo Response message, or other control message, the WTP enters the Idle state.

8. WTP Configuration Management

WTP Configuration messages are used to exchange configuration information between the AC and the WTP.

8.1. Configuration Consistency

The CAPWAP protocol provides flexibility in how WTP configuration is managed. A WTP has two options:

1. The WTP retains no configuration and accepts the configuration provided by the AC.

2. The WTP retains the configuration of parameters provided by the AC that are non-default values.

If the WTP opts to save configuration locally, the CAPWAP protocol state machine defines the Configure state, which allows for configuration exchange. In the Configure state, the WTP sends its current configuration overrides to the AC via the Configuration Status message. A configuration override is a non-default parameter. As an example, in the CAPWAP protocol, the default antenna configuration is internal omni antenna. A WTP that either has no internal antennas, or has been explicitly configured by the AC to use external antennas, sends its antenna configuration during the configure phase, allowing the AC to become aware of the WTP's current configuration.

Once the WTP has provided its configuration to the AC, the AC sends its configuration to the WTP. This allows the WTP to receive configuration and policies from the AC.

The AC maintains a copy of each active WTP configuration. There is no need for versioning or other means to identify configuration changes. If a WTP becomes inactive, the AC MAY delete the inactive WTP configuration. If a WTP fails, and connects to a new AC, the WTP provides its overridden configuration parameters, allowing the new AC to be aware of the WTP configuration.

This model allows for resiliency in case of an AC failure, ensuring another AC can provide service to the WTP. A new AC would be automatically updated with WTP configuration changes, eliminating the need for inter-AC communication and the need for all ACs to be aware of the configuration of all WTPs in the network.

Once the CAPWAP protocol enters the Run state, the WTPs begin to provide service. It is common for administrators to require that configuration changes be made while the network is operational.

Therefore, the Configuration Update Request is sent by the AC to the WTP to make these changes at run-time.

8.1.1. Configuration Flexibility

The CAPWAP protocol provides the flexibility to configure and manage WTPs of varying design and functional characteristics. When a WTP first discovers an AC, it provides primary functional information relating to its type of MAC and to the nature of frames to be exchanged. The AC configures the WTP appropriately. The AC also establishes corresponding internal state for the WTP.

8.2. Configuration Status

The Configuration Status message is sent by a WTP to deliver its current configuration to the AC.

The Configuration Status message carries binding specific message elements. Refer to the appropriate binding for the definition of this structure.

When an AC receives a Configuration Status message it acts upon the content of the message and responds to the WTP with a Configuration Status Response message.

The Configuration Status message includes multiple Radio Administrative State message elements, one for the WTP, and one for each radio in the WTP.

The Configuration Status message is sent by the WTP when in the Configure State. The AC does not transmit this message.

The following message elements MUST be included in the Configuration Status message.

- AC Name, see Section 4.6.4
- AC Name with Index, see Section 4.6.5
- Radio Administrative State, see Section 4.6.31
- Statistics Timer, see Section 4.6.36
- WTP Reboot Statistics, see Section 4.6.48

The following message elements MAY be included in the Configuration Status message.

- WTP Static IP Address Information, see Section 4.6.49

8.3. Configuration Status Response

The Configuration Status Response message is sent by an AC and provides a mechanism for the AC to override a WTP's requested configuration.

A Configuration Status Response message is sent by an AC after receiving a Configuration Request message.

The Configuration Status Response message carries binding specific message elements. Refer to the appropriate binding for the definition of this structure.

When a WTP receives a Configuration Status Response message it acts upon the content of the message, as appropriate. If the Configuration Status Response message includes a Radio Operational State message element that causes a change in the operational state of one of the radios, the WTP transmits a Change State Event to the AC, as an acknowledgement of the change in state.

The Configuration Status Response message is sent by the AC when in the Configure State. The WTP does not transmit this message.

The following message elements MUST be included in the Configuration Status Response message.

- AC IPv4 List, see Section 4.6.2
- AC IPv6 List, see Section 4.6.3
- CAPWAP Timers, see Section 4.6.12
- Decryption Error Report Period, see Section 4.6.16
- Idle Timeout, see Section 4.6.23
- WTP Fallback, see Section 4.6.40

The following message element MAY be included in the Configuration Status Response message.

- WTP Static IP Address Information, see Section 4.6.49

8.4. Configuration Update Request

Configuration Update Request messages are sent by the AC to provision the WTP while in the Run state. This is used to modify the configuration of the WTP while it is operational.

When a WTP receives a Configuration Update Request message, it responds with a Configuration Update Response message, with a Result Code message element indicating the result of the configuration request.

The AC includes the Image Identifier and Initiate Download message elements to force the WTP to update its firmware while in the Run state. The WTP MAY proceed to download the requested firmware if it determines the version specified in the Image Identifier message element is not in its non-volatile storage (see Section 9.1.1).

The Configuration Update Request is sent by the AC when in the Run State. The WTP does not transmit this message.

One or more of the following message elements MAY be included in the Configuration Update message.

- AC Name with Index, see Section 4.6.5
- AC Timestamp, see Section 4.6.6
- Add MAC ACL Entry, see Section 4.6.7
- Add Static MAC ACL Entry, see Section 4.6.9
- CAPWAP Timers, see Section 4.6.12
- Decryption Error Report Period, see Section 4.6.16
- Delete MAC ACL Entry, see Section 4.6.17
- Delete Static MAC ACL Entry, see Section 4.6.19
- Idle Timeout, see Section 4.6.23
- Location Data, see Section 4.6.28
- Radio Administrative State, see Section 4.6.31
- Statistics Timer, see Section 4.6.36

- WTP Fallback, see Section 4.6.40
- WTP Name, see Section 4.6.45
- WTP Static IP Address Information, see Section 4.6.49
- Image Identifier, see Section 4.6.25
- Initiate Download, see Section 4.6.27

8.5. Configuration Update Response

The Configuration Update Response message is the acknowledgement message for the Configuration Update Request message.

The Configuration Update Response message is sent by a WTP after receiving a Configuration Update Request message.

When an AC receives a Configuration Update Response message the result code indicates if the WTP successfully accepted the configuration.

The Configuration Update Response message is sent by the WTP when in the Run State. The AC does not transmit this message.

The following message element MUST be present in the Configuration Update message.

Result Code, see Section 4.6.33

The following message elements MAY be present in the Configuration Update Response message.

- Radio Operational State, see Section 4.6.32

8.6. Change State Event Request

The Change State Event Request message is used by the WTP for two main purposes:

- When sent by the WTP following the reception of a Configuration Status Response message from the AC, the WTP uses the Change State Event Request message to provide an update on the WTP radio's operational state and to confirm that the configuration provided by the AC was successfully applied.
- When sent during the Run state, the WTP uses the Change State Event Request message to notify the AC of an unexpected change in

the WTP's radio operational state.

When an AC receives a Change State Event Request message it responds with a Change State Event Response message and modifies its data structures for the WTP as needed. The AC MAY decide not to provide service to the WTP if it receives an error, based on local policy, and to transition to the Reset state.

The Change State Event Request message is sent by a WTP to acknowledge or report an error condition to the AC for a requested configuration in the Configuration Status Response message. The Change State Event Request message includes the Result Code message element, which indicates whether the configuration was successfully applied. If the WTP is unable to apply a specfic configuration request, it indicates the failure by including one or more Returned Message Element message elements (see Section 4.6.34).

The Change State Event Request message is sent by the WTP in the Configure or Run State. The AC does not transmit this message.

The WTP MAY save its configuration to persistent storage prior to transmitting the response. However, this is implementation specific and is not required.

The following message elements MUST be present in the Change State Event Request message.

- Radio Operational State, see Section 4.6.32
- Result Code, see Section 4.6.33

One or more of the following message elements MAY be present in the Change State Event Request message.

- Returned Message Element(s), see Section 4.6.34

8.7. Change State Event Response

The Change State Event Response message acknowledges the Change State Event Request message.

A Change State Event Response message is sent by an AC in response to a Change State Event Request message.

The Change State Event Response message is sent by the AC when in the Configure or Run state. The WTP does not transmit this message.

The Change State Event Response message carries no message elements.

The WTP does not take any action upon receipt of the Change State Event Response message.

8.8. Clear Configuration Request

The Clear Configuration Request message is used to reset the WTP configuration.

The Clear Configuration Request message is sent by an AC to request that a WTP reset its configuration to the manufacturing default configuration. The Clear Config Request message is sent while in the Run state.

The Clear Configuration Request is sent by the AC when in the Run State. The WTP does not transmit this message.

The Clear Configuration Request message carries no message elements.

When a WTP receives a Clear Configuration Request message it resets its configuration to the manufacturing default configuration.

8.9. Clear Configuration Response

The Clear Configuration Response message is sent by the WTP after receiving a Clear Configuration Request message and resetting its configuration parameters to the manufacturing default values.

The Clear Configuration Response is sent by the WTP when in the Run State. The AC does not transmit this message.

The Clear Configuration Request message MUST include the following message element.

- Result Code, see Section 4.6.33

9. Device Management Operations

This section defines CAPWAP operations responsible for debugging, gathering statistics, logging, and firmware management.

9.1. Firmware Management

This section describes the firmware download procedures used by the CAPWAP protocol. Firmware download can occur during the Image Data or Run state.

Figure 4 provides an example of a WTP that performs a firmware upgrade while in the Image Data state. In this example, the WTP does not already have the requested firmware (Image Identifier = x), and downloads the image from the AC.

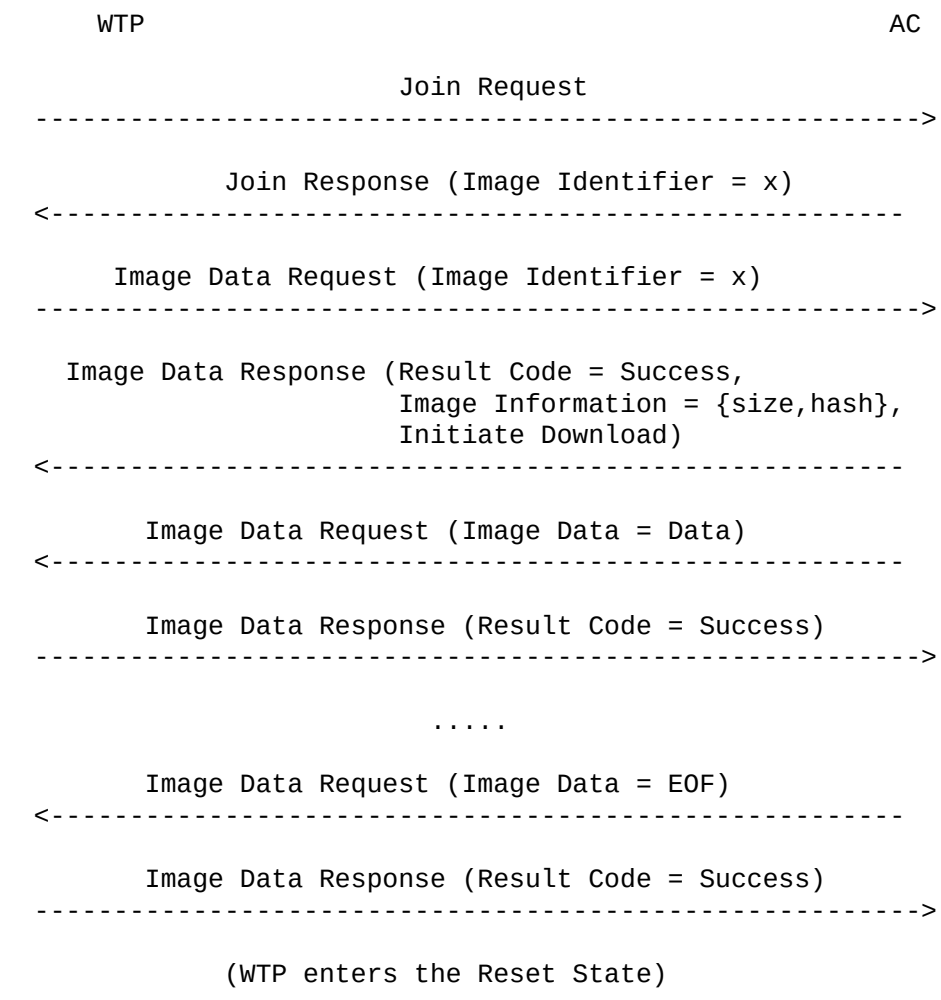


Figure 4: WTP Firmware Download Case 1

Figure 5 provides an example in which the WTP has the image specified by the AC in its non-volative storage. The WTP opts to NOT download the firmware and immediately reset.

Figure 5: WTP Firmware Download Case 2

Figure 6 provides an example of a WTP that performs a firmware upgrade while in the Run state. This mode of firmware upgrade allows the WTP to download its image while continuing to provide service. The WTP will not automatically reset until it is notified by the AC, with a Reset Request message.

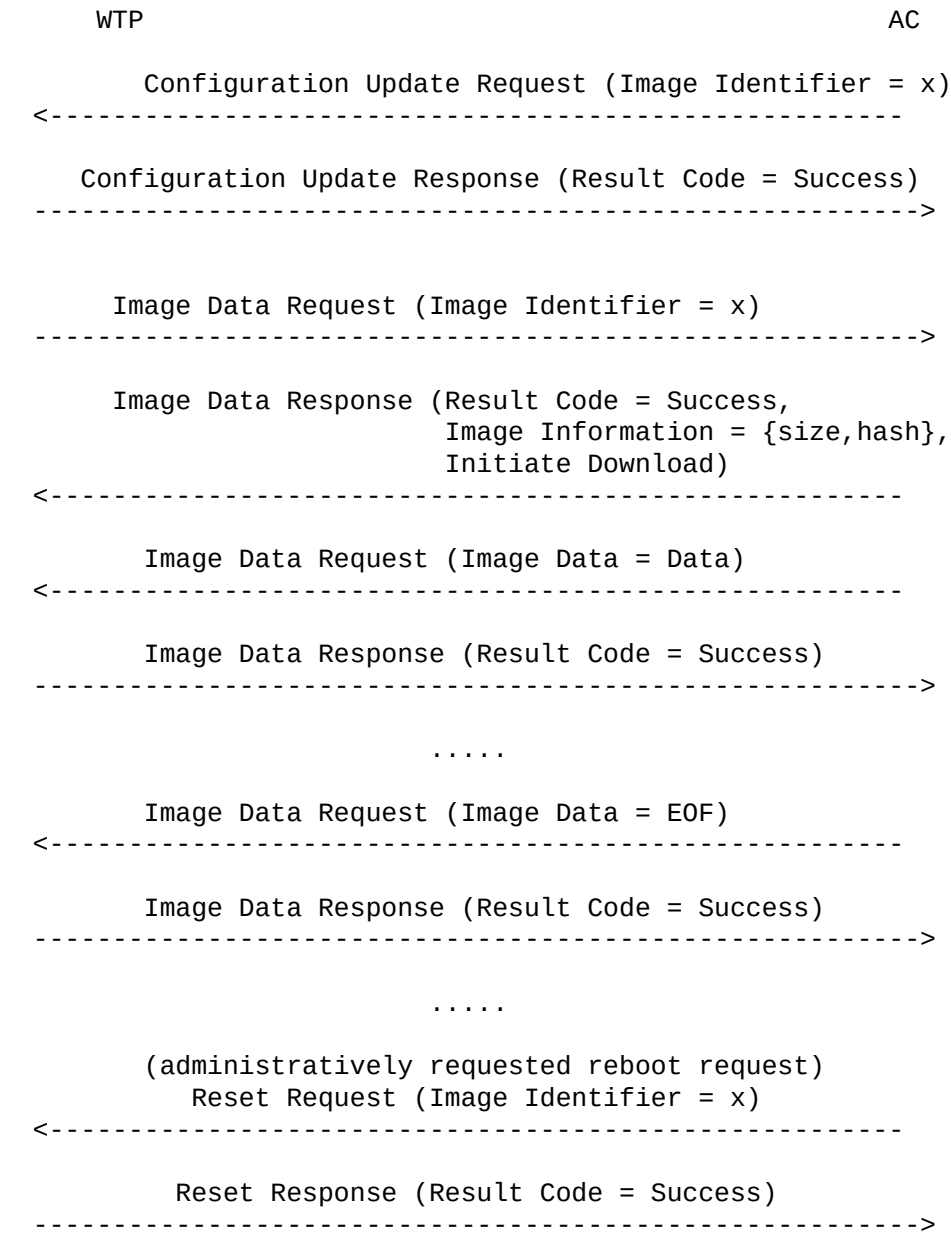


Figure 6: WTP Firmware Download Case 3

Figure 7 provides another example of the firmware download while in the Run state. In this example, the WTP already has the image specified by the AC in its non-volative storage. The WTP opts to NOT download the firmware. The WTP resets upon receipt of a Reset Request message from the AC.

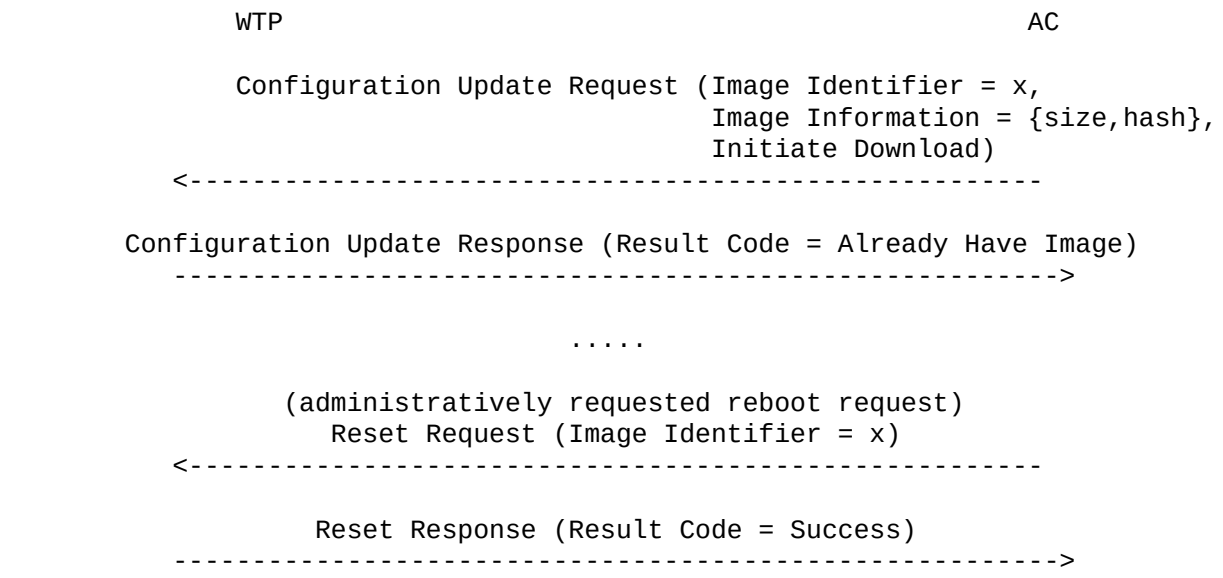


Figure 7: WTP Firmware Download Case 4

9.1.1. Image Data Request

The Image Data Request message is used to update firmware on the WTP. This message and its companion Response message are used by the AC to ensure that the image being run on each WTP is appropriate.

Image Data Request messages are exchanged between the WTP and the AC to download a new firmware image to the WTP. When a WTP or AC receives an Image Data Request message it responds with an Image Data Response message. The message elements contained within the Image Data Request message are required to determine the intent of the request.

The decision that new firmware is to be downloaded to the WTP can occur in one of two ways:

When the WTP joins the AC, the Join Response message includes the Image Identifier message element, which informs the WTP of the firmware it is expected to run. if the WTP does not currently have the requested firmware version, it transmits an Image Data Request message, with the appropriate Image Identifier message element. If the WTP already has the requested firmware, it simply resets.

Once the WTP is in the Run state, it is possible for the AC to cause the WTP to initiate a firmware download by sending a Configuration Update Request message with the Initiate Download and and Image Identifier message elements. The WTP then transmits

the Image Data Request message, which includes the Image Identifier message element to start the download process. Note that when the firmware is downloaded in this way, the WTP does not automatically reset after the download is complete. The WTP will only reset when it receives a Reset Request message from the AC. If the WTP already had the requested firmware version in its non-volatile storage, the WTP does not transmit the Image Data Request message and responds with a Configuration Update Response message with the Result Code set to Image Already Present.

Regardless of how the download was initiated, once the AC receives an Image Data Request message with the Image Identifier message element, it begins the transfer process by transmitting an Image Data Request message that includes the Image Data message element. This continues until the firmware image has been transfered.

The Image Data Request message is sent by the WTP or the AC when in the Image Data or Run State.

The following message elements MAY be included in the Image Data Request message.

- Image Data, see Section 4.6.24
- Image Identifier, see Section 4.6.25

9.1.2. Image Data Response

The Image Data Response message acknowledges the Image Data Request message.

An Image Data Response message is sent in response to a received Image Data Request message. Its purpose is to acknowledge the receipt of the Image Data Request message. The Result Code is included to indicate whether a previously sent Image Data Request message was invalid.

The Image Data Response message is sent by the WTP or the AC when in the Image Data or Run State.

The following message element MUST be included in the Image Data Response message.

- Result Code, see Section 4.6.33

The following message elements MAY be included in the Image Data Response message.

- Image Information, see Section 4.6.26
- Initiate Download, see Section 4.6.27

Upon receiving an Image Data Response message indicating an error, the WTP MAY retransmit a previous Image Data Reqest message, or abandon the firmware download to the WTP by transitioning to the Reset state.

9.2. Reset Request

The Reset Request message is used to cause a WTP to reboot.

A Reset Request message is sent by an AC to cause a WTP to reinitialize its operation.

The Reset Request is sent by the AC when in the Run State. The WTP does not transmit this message.

The following message elements MUST be included in the Reset Request message.

- Image Identifier, see Section 4.6.25

When a WTP receives a Reset Request message, it responds with a Reset Response message indicating success and then reinitialize itself. If the WTP is unable to write to its non-volatile storage, to ensure that it runs the requested software version indicated in the Image Identifier message element, it MAY send the appropriate Result Code message element, but MUST reboot. If the WTP is unable to reset, including a hardware reset, it sends a Reset Response message to the AC with a Result Code message element indicating failure. The AC no longer provides service to the WTP.

9.3. Reset Response

The Reset Response message acknowledges the Reset Request message.

A Reset Response message is sent by the WTP after receiving a Reset Request message.

The Reset Response is sent by the WTP when in the Run State. The AC does not transmit this message.

The following message element MAY be included in the Image Data Request message.

- Result Code, see Section 4.6.33

When an AC receives a successful Reset Response message, it is notified that the WTP will reinitialize its operation. An AC that receives a Reset Response message indicating failure may opt to no longer provide service to the WTP.

9.4. WTP Event Request

The WTP Event Request message is used by a WTP to send information to its AC. The WTP Event Request message MAY be sent periodically, or sent in response to an asynchronous event on the WTP. For example, a WTP MAY collect statistics and use the WTP Event Request message to transmit the statistics to the AC.

When an AC receives a WTP Event Request message it will respond with a WTP Event Response message.

The presence of the Delete Station message element is used by the WTP to inform the AC that it is no longer providing service to the station. This could be the result of an Idle Timeout (see Section 4.6.23), due to to resource shortages, or some other reason.

The WTP Event Request message is sent by the WTP when in the Run State. The AC does not transmit this message.

The WTP Event Request message MUST contain one of the message elements listed below, or a message element that is defined for a specific wireless technology. More than one of each messsage element listed MAY be included in the WTP Event Request message.

- Decryption Error Report, see Section 4.6.15
- Duplicate IPv4 Address, see Section 4.6.21
- Duplicate IPv6 Address, see Section 4.6.22
- WTP Operational Statistics, see Section 4.6.46
- WTP Radio Statistics, see Section 4.6.47
- WTP Reboot Statistics, see Section 4.6.48
- Delete Station, see Section 4.6.18

9.5. WTP Event Response

The WTP Event Response message acknowledges receipt of the WTP Event Request message.

A WTP Event Response message is sent by an AC after receiving a WTP Event Request message.

The WTP Event Response message is sent by the AC when in the Run State. The WTP does not transmit this message.

The WTP Event Response message carries no message elements.

9.6. Data Transfer Request

The Data Transfer Request message is used to deliver debug information from the WTP to the AC.

Data Transfer Request messages are sent by the WTP to the AC when the WTP determines that it has important information to send to the AC. For instance, if the WTP detects that its previous reboot was caused by a system crash, it can send the crash file to the AC. The remote debugger function in the WTP also uses the Data Transfer Request message to send console output to the AC for debugging purposes.

When the AC receives a Data Transfer Request message it responds to the WTP with a Data Transfer Response message. The AC MAY log the information received.

The Data Transfer Request message is sent by the WTP when in the Run State. The AC does not transmit this message.

The Data Transfer Request message MUST contain one of the message elements listed below.

- Data Transfer Data, see Section 4.6.13
- Data Transfer Mode, see Section 4.6.14

9.7. Data Transfer Response

The Data Transfer Response message acknowledges the Data Transfer Request message.

A Data Transfer Response message is sent in response to a received Data Transfer Request message. Its purpose is to acknowledge receipt of the Data Transfer Request message.

The Data Transfer Response message is sent by the AC when in the Run State. The WTP does not transmit this message.

The Data Transfer Response message carries no message elements.

Upon receipt of a Data Transfer Response message, the WTP transmits more information, if more information is available.

10. Station Session Management

Messages in this section are used by the AC to create, modify or delete station session state on the WTPs.

10.1. Station Configuration Request

The Station Configuration Request message is used to create, modify or delete station session state on a WTP. The message is sent by the AC to the WTP, and MAY contain one or more message elements. The message elements for this CAPWAP control message include information that is generally highly technology specific. Refer to the appropriate binding document for definitions of the messages elements that are included in this control message.

The Station Configuration Request message is sent by the AC when in the Run State. The WTP does not transmit this message.

The following CAPWAP Control message elements MAY be included in the Station Configuration Request message. More than one of each message element listed MAY be included in the Station Configuration Request message.

- Add Station, see Section 4.6.8
- Delete Station, see Section 4.6.18

10.2. Station Configuration Response

The Station Configuration Response message is used to acknowledge a previously received Station Configuration Request message.

The Station Configuration Response message is sent by the WTP when in the Run State. The AC does not transmit this message.

The following message element MUST be present in the Station Configuration Response message.

- Result Code, see Section 4.6.33

The Result Code message element indicates that the requested configuration was successfully applied, or that an error related to processing of the Station Configuration Request message occurred on the WTP.

11. NAT Considerations

There are three specific situations in which a NAT deployment may be used in conjunction with a CAPWAP-enabled deployment. The first consists of a configuration in which a single WTP is behind a NAT system. Since all communication is initiated by the WTP, and all communication is performed over IP using two UDP ports, the protocol easily traverses NAT systems in this configuration.

In the second case, two or more WTPs are deployed behind the same NAT system. Here, the AC would receive multiple connection requests from the same IP address, and cannot differentiate the originating WTP of the connection requests. The CAPWAP Data Check state, which establishes the data plane connection and communicates the Data Keepalive, includes the Session Identifier message element, which is used to bind the control and data plane. Use of the Session Identifier message element enables the AC to match the control and data plane flows from multiple WTPs behind the same NAT system (multiple WTPs sharing the same IP address).

In the third configuration, the AC is deployed behind a NAT. Two issues exist in this situation. First, an AC communicates its interfaces and corresponding WTP load using the CAPWAP Control IP(v4/v6) Address message element. This message element is currently mandatory, and if NAT compliance becomes an issue, it is possible to either:

1. Make the CAPWAP Control IP (v4/v6) Address optional, allowing the WTP to use the known IP Address. Note that this approach eliminates the ability to perform load balancing of WTP across ACs, and therefore is not the recommended approach.

2. Allow an AC to configure a NAT’ed address for every AC that would otherwise be communicated in the CAPWAP Control IP (v4/v6) Address message element.

3. Require that if a WTP determines that the AC List message element contains a set of IP Addresses that are different from the AC IP Address the WTP is currently using, then assume that NAT is present, and require that the WTP communicate with the AC IP Address (and ignore the CAPWAP Control IP (v4/v6) Address message element(s)).

The CAPWAP protocol allows for all of the AC identities supporting a group of WTPs to be communicated through the AC List message element. This feature MUST be disabled when the AC is behind a NAT and the IP Address that is embedded is invalid.

The CAPWAP protocol allows an AC to configure a static IP address on a WTP using the WTP Static IP Address Information message element. This message element SHOULD NOT be used in NAT'ed environments, unless the administrator is familiar with the internal IP addressing scheme within the WTP's private network, and does not rely on the public address seen by the AC.

When a WTP detects the duplicate address condition, it generates a message to the AC, which includes the Duplicate IP Address message element. The IP Address embedded within this message element is different from the public IP address seen by the AC.

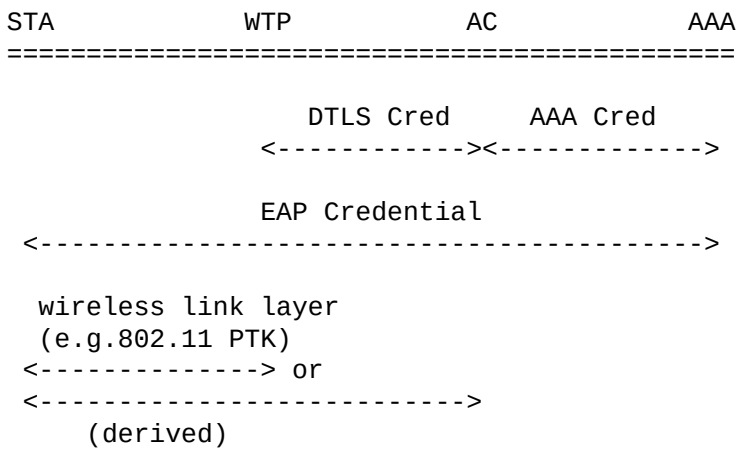
When CAPWAP is run over IPv6, NAT support can only be provided if the IPv6 NAT system is capable of performing address translation over the UDP-Lite 3828 protocol [11]. A protocol interoperability issues will exist if the NAT system is being utilized for IPv4/IPv6 address translation.

12. Security Considerations

This section describes security considerations for the CAPWAP protocol. It also provides security recommendations for protocols used in conjunction with CAPWAP.

12.1. CAPWAP Security

As it is currently specified, the CAPWAP protocol sits between the security mechanisms specified by the wireless link layer protocol (e.g.IEEE 802.11i) and AAA. One goal of CAPWAP is to bootstrap trust between the STA and WTP using a series of preestablished trust relationships:



Within CAPWAP, DTLS is used to secure the link between the WTP and AC. In addition to securing control messages, it's also a link in this chain of trust for establishing link layer keys. Consequently, much rests on the security of DTLS.

In some CAPWAP deployment scenarios, there are two channels between the WTP and AC: the control channel, carrying CAPWAP control messages, and the data channel, over which client data packets are tunneled between the AC and WTP. Typically, the control channel is secured by DTLS, while the data channel is not.

The use of parallel protected and unprotected channels deserves special consideration, but does not create a threat. There are two potential concerns: attempting to convert protected data into un-protected data and attempting to convert un-protected data into protected data. These concerns are addressed below.

12.1.1. Converting Protected Data into Unprotected Data

Since CAPWAP does not support authentication-only ciphers (i.e. all supported ciphersuites include encryption and authentication), it is not possible to convert protected data into unprotected data. Since encrypted data is (ideally) indistinguishable from random data, the probability of an encrypted packet passing for a well-formed packet is effectively zero.

12.1.2. Converting Unprotected Data into Protected Data (Insertion)

The use of message authentication makes it impossible for the attacker to forge protected records. This makes conversion of unprotected records to protected records impossible.

12.1.3. Deletion of Protected Records

An attacker could remove protected records from the stream, though not undetectably so, due the built-in reliability of the underlying CAPWAP protocol. In the worst case, the attacker would remove the same record repeatedly, resulting in a CAPWAP session timeout and restart. This is effectively a DoS attack, and could be accomplished by a man in the middle regardless of the CAPWAP protocol security mechanisms chosen.

12.1.4. Insertion of Unprotected Records

An attacker could inject packets into the unprotected channel, but this may become evident if sequence number desynchronization occurs as a result. Only if the attacker is a MiM can packets be inserted undetectably. This is a consequence of that channel's lack of protection, and not a new threat resulting from the CAPWAP security mechanism.

12.2. Session ID Security

Since DTLS does not export a unique session identifier, there can be no explicit protocol binding between the DTLS layer and CAPWAP layer. As a result, implementations MUST provide a mechanism for performing this binding. For example, an AC MUST NOT associate decrypted DTLS control packets with a particular WTP session based solely on the Session ID in the packet header. Instead, identification should be done based on which DTLS session decrypted the packet. Otherwise one authenticated WTP could spoof another authenticated WTP by altering the Session ID in the encrypted CAPWAP header.

It should be noted that when the CAPWAP data channel is unencrypted, the WTP Session ID is exposed and possibly known to adversaries and

other WTPs. This would allow the forgery of the source of data-channel traffic. This, however, should not be a surprise for unencrypted data channels. When the data channel is encrypted, the Session ID is not exposed, and therefore can safely be used to associate a data and control channel. The 64-bit length of the Session ID mitigates online guessing attacks where an adversarial, authenticated WTP tries to correlate his own data channel with another WTP's control channel. Note that for encrypted data channels, the Session ID should only be used for correlation for the first packet immediately after the initial DTLS handshake. Future correlation should instead be done via identification of a packet's DTLS session.

12.3. Discovery Attacks

Since the Discovery Request messages are sent in the clear, it is important that AC implementations NOT assume that receiving such a request from a WTP implies that it has rebooted, and consequently tear down any active DTLS sessions. Discovery Request messages can easily be spoofed by malicious devices, so it is important that the AC maintain two separate sets of states for the WTP until the DTLSSessionEstablished notification is received, indicating that the WTP was authenticated. Once a new DTLS session is successfully established, any state referring to the old session can be cleared.

12.4. Interference with a DTLS Session

If a WTP or AC repeatedly receives packets which fail DTLS authentication or decryption, this could indicate a DTLS desynchronization between the AC and WTP, a link prone to undetectable bit errors, or an attacker trying to disrupt a DTLS session.

In the state machine (section 2.3), transitions to the DTLS tear down state can be triggered by frequently receiving DTLS packets with authentication or decryption errors. The threshold or technique for deciding when to move to the tear down state should be chosen carefully. Being able to easily transition to DTLS TD allows easy detection of malfunctioning devices, but allows for denial of service attacks. Making it difficult to transition to DTLS TD prevents denial of service attacks, but makes it more difficult to detect and reset a malfunctioning session. Implementers should set this policy with care.

12.5. Use of Preshared Keys in CAPWAP

While use of preshared keys may provide deployment and provisioning advantages not found in public key based deployments, it also

introduces a number of operational and security concerns. In particular, because the keys must typically be entered manually, it is common for people to base them on memorable words or phrases. These are referred to as "low entropy passwords/passphrases".

Use of low-entropy preshared keys, coupled with the fact that the keys are often not frequently updated, tends to significantly increase exposure. For these reasons, the following recommendations are made:

- When DTLS is used with a preshared-key (PSK) ciphersuite, each WTP SHOULD have a unique PSK. Since WTPs will likely be widely deployed, their physical security is not guaranteed. If PSKs are not unique for each WTP, key reuse would allow the compromise of one WTP to result in the compromise of others

- Generating PSKs from low entropy passwords is NOT RECOMMENDED.

- It is RECOMMENDED that implementations that allow the administrator to manually configure the PSK also provide a capability for generation of new random PSKs, taking RFC 4086 [2] into account.

- Preshared keys SHOULD be periodically updated. Implementations MAY facilitate this by providing an administrative interface for automatic key generation and periodic update, or it MAY be accomplished manually instead.

Every pairwise combination of WTP and AC on the network SHOULD have a unqiue PSK. This prevents the domino effect (see Guidance for AAA Key Management [16]). If PSKs are tied to specific WTPs, then knowledge of the PSK implies a binding to a specified identity that can be authorized.

If PSKs are shared, this binding between device and identity is no longer possible. Compromise of one WTP can yield compromise of another WTP, violating the CAPWAP security hierarchy. Consequently, sharing keys between WTPs is NOT RECOMMENDED.

12.6. Use of Certificates in CAPWAP

For public-key-based DTLS deployments, each device SHOULD have unique credentials, with an extended key usage authorizing the device to act as either a WTP or AC. If devices do not have unique credentials, it is possible that by compromising one device, any other device using the same credential may also be considered to be compromised.

Certificate validation involves checking a large variety of things.

Since the necessary things to validate are often environment-specific, many are beyond the scope of this document. In this section, we provide some basic guidance on certificate validation.

Each device is responsible for authenticating and authorizing devices with which they communicate. Authentication entails validation of the chain of trust leading to the peer certificate, followed by the the peer certificate itself. At a minimum, devices SHOULD use SSH-style certificate caching to guarantee consistency. If devices have access to a certificate authority, they SHOULD properly validate the trust chain. Implementations SHOULD also provide a secure method for verifying that the credential in question has not been revoked.

Note that if the WTP relies on the AC for network connectivity (e.g. the AC is a layer 2 switch to which the WTP is directly connected), the WTP may not be able to contact an OCSP server or otherwise obtain an up to date CRL if a compromised AC doesn't explicitly permit this. This cannot be avoided, except through effective physical security and monitoring measures at the AC.

Proper validation of certificates typically requires checking to ensure the certificate has not yet expired. If devices have a real-time clock, they SHOULD verify the certificate validity dates. If no real-time clock is available, the device SHOULD make a best-effort attempt to validate the certificate validity dates through other means. Failure to check a certificate's temporal validity can make a device vulnerable to man-in-the-middle attacks launched using compromised, expired certificates, and therefore devices should make every effort to perform this validation.

12.7. AAA Security

The AAA protocol is used to distribute EAP keys to the ACs, and consequently its security is important to the overall system security. When used with TLS or IPsec, security guidelines specified in RFC 3539 [5] SHOULD be followed.

In general, the link between the AC and AAA server SHOULD be secured using a strong ciphersuite keyed with mutually authenticated session keys. Implementations SHOULD NOT rely solely on Basic RADIUS shared secret authentication as it is often vulnerable to dictionary attacks, but rather SHOULD use stronger underlying security mechanisms.

13. Management Considerations

The CAPWAP protocol assumes that it is the only configuration interface to the WTP to configure parameters that are specified in the CAPWAP specifications. While the use of a separate management protocol MAY be used for the purposes of monitoring the WTP directly, configuring the WTP through a separate management interface is not recommended. Configuring the WTP through a separate protocol, such as via a CLI or SNMP, could lead to the AC state being out of sync with the WTP.

14. IANA Considerations

A separate UDP port for data channel communications is (currently) the selected demultiplexing mechanism, and a port must be assigned for this purpose in Section 3.1. The UDP port numbers are listed by IANA at http://www.iana.org/assignments/port-numbers.

IANA needs to assign an organization local multicast address called the "All ACs multicast address" from the IPv6 multicast address registry in Section 3.3

14.1. CAPWAP Message Types

The Message Type field in the CAPWAP header (Section 4.5.1.1) is used to identify the operation performed by the message. There are multiple namespaces, which is identified via the first three octets of the field containing the IANA Enterprise Number [10]. When the Enterprise Number is set to zero, the message types are reserved for use by the base CAPWAP specification which are controlled and maintained by IANA and requires a Standards Action.

14.2. Wireless Binding Identifiers

The Wireless Binding Identifier (WBID) field in the CAPWAP header (Section 4.3) is used to identify the wireless technology associated with the packet. Due to the limited address space available, a new WBID request requires Standards Action.

15. Acknowledgements

The following individuals are acknowledged for their contributions to this protocol specification: Puneet Agarwal, Saravanan Govindan, Peter Nilsson, and David Perkins.

Michael Vakulenko contributed text to describe how CAPWAP can be used over layer 3 (IP/UDP) networks.

16. References

16.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[3] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[4] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[5] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.

[6] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[7] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[8] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[9] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[10] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[11] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[12] Calhoun, P., Montemurro, M., Stanley, D., "CAPWAP Protocol Binding for IEEE 802.11", draft-ietf-capwap-protocol-binding-ieee80211-04 (work in progress), June 2007.

[13] Calhoun, P., "CAPWAP Access Controller DHCP Option", draft-ietf-capwap-dhc-ac-option-00 (work in progress), June 2007.

16.2. Informational References

[14] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[15] Manner, J. and M. Kojo, "Mobility Related Terminology", RFC 3753, June 2004.

[16] Housley, R. and B. Aboba, "Guidance for AAA Key Management", draft-housley-aaa-key-mgmt-09 (work in progress), February 2007.

[17] Modadugu et al, N., "The Design and Implementation of Datagram TLS", Feb 2004.

[18] IEEE, "Guidelines for use of a 48-bit Extended Unique Identifier", Dec 2005.

[19] IEEE, "GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY".

Editors' Addresses

Pat R. Calhoun
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134

Phone: +1 408-853-5269
Email: pcalhoun@cisco.com

Michael P. Montemurro
Research In Motion
5090 Commerce Blvd
Mississauga, ON L4W 5M4
Canada

Phone: +1 905-629-4746 x4999
Email: mmontemurro@rim.com

Dorothy Stanley
Aruba Networks
1322 Crossman Ave
Sunnyvale, CA 94089

Phone: +1 630-363-1389
Email: dstanley@arubanetworks.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).