Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2009

P. Calhoun, Editor
Cisco Systems, Inc.
M. Montemurro, Editor
Research In Motion
D. Stanley, Editor
Aruba Networks
October 31, 2008

CAPWAP Protocol Specification
draft-ietf-capwap-protocol-specification-15

Status of this Memo

Abstract

This specification defines the Control And Provisioning of Wireless Access Points (CAPWAP) Protocol, meeting the objectives defined by the CAPWAP working group in RFC 4564. 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, while separate binding extensions will enable its use with additional wireless technologies.

Table of Contents

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, and meets the Objectives for Control and Provisioning of Wireless Access Points (CAPWAP) [RFC4564].

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 [I-D.ietf-capwap-protocol-binding-ieee80211] 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.

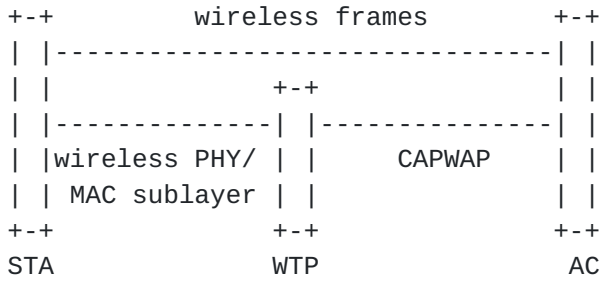


Figure 1: Representative CAPWAP Architecture for Split MAC

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 802.11 Integration 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.

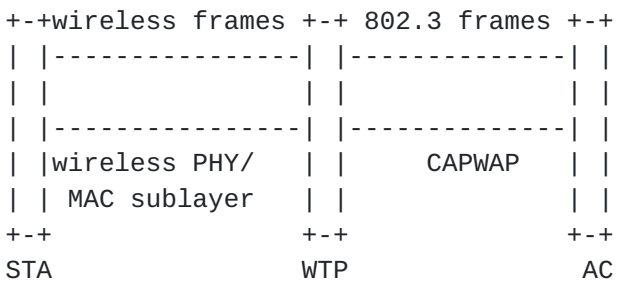


Figure 2: Representative CAPWAP Architecture for Local MAC

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 an extensible protocol that is not bound to a specific wireless technology. Extensibility is provided via 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 [RFC2119].

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) [I-D.ohara-capwap-lwapp] to be used as the basis of the CAPWAP protocol specification. The following people are authors of the LWAPP document:

Bob O'Hara
Email: bob.ohara@computer.org

Pat Calhoun, Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134
Phone: +1 408-902-3240, 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, Green Hills Software
825 Victors Way, Suite 100, Ann Arbor, MI 48108
Phone: +1 734 222 1610, Email: shares@ndzh.com

Datagram Transport Layer Security (DTLS) [RFC4347] 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 [I-D.narasimhan-ietf-slapp]. 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 WTP 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 interface to a 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 [RFC3753].

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) [RFC4347]. 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. In order to establish the secure DTLS connection, the WTP will need some amount of pre-provisioning, which is specified in Section 12.5. 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. MTU Discovery and Fragmentation are described in Section 3.

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, as well its associated wireless link layer protocol. 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 [I-D.ietf-capwap-protocol-binding-ieee80211], begins with "IEEE 802.11".

The CAPWAP binding concept MUST also be used in any future specification 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 between a CAPWAP WTP and AC. 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 asterisk in Figure 3.


Finished*
(-- AC callout for WTP authorization --)
ChangeCipherSpec,
Finished*
(-- DTLS session is established now --)
Join Request
Join Response
[-- Join State Complete --]
(-- assume image is up to date --)
Configuration Status Request
Configuration Status Response
[-- Configure State Complete --]
Change State Event Request
Change State Event Response
[-- Data Check State Complete --]
(-- enter RUN state --)
:
:
Echo Request
Echo Response
:
:
Event Request
Event Response

:
:

Figure 3: CAPWAP Control Protocol Exchange

At the end of the illustrated CAPWAP message exchange, the AC and WTP are securely exchanging CAPWAP control messages. This illustration is provided to clarify protocol operation, and does not include any possible error conditions. 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.

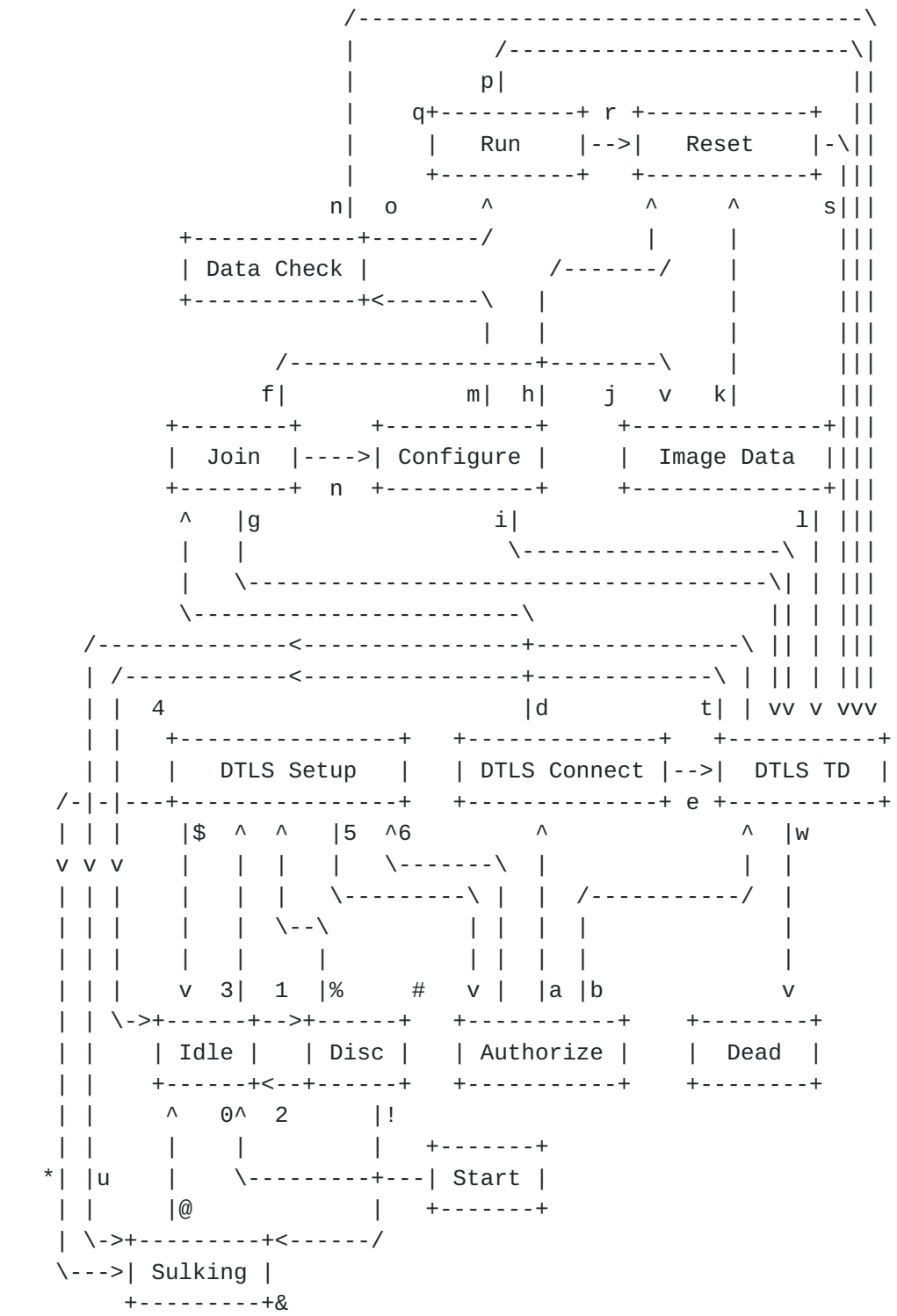


Figure 4: CAPWAP Integrated State Machine

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 message 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 state machine works differently on the AC since it communicates with many WTPs. The AC uses the concept of three threads. Note that the term thread used here does not necessarily imply that implementers must use threads, but it is one possible way of implementing the AC's state machine.

Listener Thread: The AC's Listener thread handles inbound DTLS session establishment requests, through the DTLSListen command. Upon creation, the Listener thread starts in the DTLS Setup state. Once a DTLS session has been validated, which occurs when the state machine enters the "Authorize" state, the Listener thread creates a WTP session specific Service thread and state context. The state machine transitions in Figure 4 are represented by numerals. It is necessary for the AC to protect itself against various attacks that exist with non-authenticated frames. See Section 12 for more information.

Discovery Thread: The AC's Discovery thread is responsible for receiving, and responding to, Discovery Request messages. The state machine transitions in Figure 4 are represented by numerals. Note that the Discovery thread does not maintain any per-WTP specific context information, and a single state context exists. It is necessary for the AC to protect itself against various attacks that exist with non-authenticated frames. See Section 12 for more information.

Service Thread: The AC's Service thread handles the per-WTP states, and one such thread exists per-WTP connection. This thread is created by the listener thread when the Authorize state is reached. When created, the Service thread inherits a copy of the state machine context from the Listener thread. When communication with the WTP is complete, the Service thread is terminated and all associated resources are released. The state machine transitions in Figure 4 are represented by alphabetic and punctuation characters.

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.

Start to Idle (0): This transition occurs once device initialization is complete.

WTP: This state transition is used to start the WTP's CAPWAP state machine.

AC: The AC creates the Discovery and Listener threads and starts the CAPWAP state machine.

Idle to Discovery (1): This transition occurs to support the CAPWAP discovery process .

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: This state transition is executed by the AC's Discovery thread, and occurs when a Discovery Request message is received. The AC SHOULD respond with a Discovery Response message (see Section 5.2).

Discovery to Discovery (#): 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 an invalid state transition for the AC.

Discovery to Idle (2): This transition occurs on the AC's Discovery thread when the Discovery processing is complete.

WTP: This is an invalid state transition for the WTP.

AC: This state transition is executed by the AC's Discovery thread when it has transmitted the Discovery Response, in response to a Discovery Request.

Discovery to Sulking (!): This transition occurs on a WTP when AC Discovery fails.

WTP: The WTP enters this state when the DiscoveryInterval timer expires and 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 MUST be ignored.

AC: This is an invalid state transition for the AC.

Sulking to Idle (@): 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: This is an invalid state transition for the AC.

Sulking to Sulking (&): 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: This is an invalid state transition for the AC.

Idle to DTLS Setup (3): 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 and the WaitDTLS timer is started (see Section 4.7). When the discovery phase is bypassed, it is assumed the WTP has locally configured ACs.

AC: Upon entering the Idle state from the Start state, the newly created Listener thread automatically transitions to the DTLS Setup and invokes the DTLSListen command (see Section 2.3.2.1), and the WaitDTLS timer is started (see Section 4.7).

Discovery to DTLS Setup (%): 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: This is an invalid state transition for the AC.

DTLS Setup to Idle ($): This transition occurs when the DTLS connection setup fails.

WTP: The WTP initiates this state transition when it receives a DTLSEstablishFail notification from DTLS (see Section 2.3.2.2), and the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter have not reached the value of the MaxFailedDTLSSessionRetry variable (see Section 4.8). This error notification aborts the secure DTLS session establishment. When this notification is received, the FailedDTLSSessionCount counter is incremented. This state transition also occurs if the WaitDTLS timer has expired.

AC: This is an invalid state transition for the AC.

DTLS Setup to Sulking (*): This transition occurs when repeated attempts to setup the DTLS connection have failed.

WTP: The WTP enters this state when the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter reaches the value of the 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: This is an invalid state transition for the AC.

DTLS Setup to DTLS Setup (4): This transition occurs when the DTLS Session failed to be established.

WTP: This is an invalid state transition for the WTP.

AC: The AC's Listener 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.

The Listener thread then invokes the DTLSListen command (see Section 2.3.2.1).

DTLS Setup to Authorize (5): 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 is handled by the AC's Listener thread when the DTLS module initiates the DTLSPeerAuthorize notification (see Section 2.3.2.2). The Listener thread forks an instance of the Service thread, along with a copy of the state context. Once created, the Service thread performs an authorization check against the WTP credentials. See Section 2.4.4 for more information on WTP authorization.

Authorize to DTLS Setup (6): This transition is executed by the Listener thread to enable it to listen for new incoming sessions.

WTP: This is an invalid state transition for the WTP.

AC: This state transition occurs when the AC's Listener thread has created the WTP context and the Service thread. The Listener thread then invokes the DTLSListen command (see Section 2.3.2.1).

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

WTP: This state transition occurs when the WTP has successfully authorized the AC's credentials (see Section 2.4.4). This is done by invoking the DTLSAccept DTLS command (see Section 2.3.2.1).

AC: This state transition occurs when the AC has successfully authorized the WTP's credentials (see Section 2.4.4). This is done by invoking the DTLSAccept DTLS command (see Section 2.3.2.1).

Authorize to DTLS Teardown (b): 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). This state transition also occurs if the WaitDTLS timer has expired. The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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). This state transition also occurs if the WaitDTLS timer has expired. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

DTLS Connect to DTLS Teardown (c): 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. This state transition also occurs if the WaitDTLS timer has expired. The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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, and both of the FailedDTLSAuthFailCount and FailedDTLSSessionCount counters have not reached the value of the MaxFailedDTLSSessionRetry variable (see Section 4.8). This state transition also occurs if the WaitDTLS timer has expired. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

DTLS Connect to Join (d): 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. The WTP enters the Join state by transmiting the

Join Request to the AC. The WTP stops the WaitDTLS timer.

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. The AC stops the WaitDTLS timer, and starts the WaitJoin timer.

Join to DTLS Teardown (e): 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, or 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. 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. The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

Join to Image Data (f): 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 that the software version in the 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, after having sent its Join Response to the WTP. The AC stops the WaitJoin timer. The AC MUST transmit an Image Data Response message (see Section 9.1.2) to the WTP, which includes a portion of the firmware.

Join to Configure (g): 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 message, and determines that the included Image Identifier message element is the same as its currently running image. The WTP transmits the Configuration Status Request message (see Section 8.2) to the AC with message elements describing its current configuration.

AC: This state transition occurs when it receives the Configuration Status Request message from the WTP (see Section 8.2), which MAY include specific message elements to override the WTP's configuration. The AC stops the WaitJoin timer. The AC transmits the Configuration Status Response message (see Section 8.3) and starts the ChangeStatePendingTimer timer (see Section 4.7).

Configure to Reset (h): 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. The CAPWAP Reset command is used to indicate to the peer that it will initiate a DTLS teardown.

WTP: The WTP enters the Reset state when it receives a Configuration Status Response message 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 (i): 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. The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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 AC MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

Image Data to Image Data (j): 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. This state transition also occurs when the WTP receives the subsequent Image Data Requests, at which time it resets the ImageDataStartTimer time to ensure it receives the next expected Image Data Request from the AC. This state transition can also occur when the WTP's EchoInterval timer (see Section 4.7.7) expires, in which case the WTP transmits an Echo Request message (see Section 7.1), and resets its EchoInterval timer. The state transition also occurs when the WTP receives an Echo Response from the AC (see Section 7.2.

AC: This state transition occurs either when the AC receives the Image Data Response message from the WTP while already in the Image Data state. This state transition also occurs when the AC receives an Echo Request (see Section 7.1) from the WTP, in which case it responds with an Echo Response (see Section 7.2), and resets its EchoInterval timer (see Section 4.7.7).

Image Data to Reset (k): 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, or if the ImageDataStartTimer timer expires, 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 either when the image transfer has successfully completed or an error occurs during the image download process.

Image Data to DTLS Teardown (l): 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. The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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 AC MAY tear down the DTLS session if it receives frequent DTLSDecapFailure notifications. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

Configure to Data Check (m): 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 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 DataCheckTimer timer and stops the ChangeStatePendingTimer timer (see Section 4.7).

Data Check to DTLS Teardown (n): This transition occurs when the WTP does not complete the Data Check exchange.

WTP: This state transition occurs if the WTP does not receive the Change State Event Response message before a CAPWAP retransmission timeout occurs. The WTP also transitions to this state if the underlying reliable transport's RetransmitCount counter has reached the MaxRetransmit variable (see Section 4.7). The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

AC: The AC enters this state when the DataCheckTimer timer expires (see Section 4.7). The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

Data Check to Run (o): This state transition occurs when the linkage between the control and data channels is established, 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 4.7.2) and transmits a Data Channel Keep Alive packet (see Section 4.4.1). The WTP then starts the EchoInterval timer and 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. The AC disables the DataCheckTimer timer. 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 (p): This state transition occurs when an error has occurred in the DTLS stack, causing the DTLS session to be torn down.

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). The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

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 AC 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). This state

transition also occurs when the AC's EchoInterval timer (see Section 4.7.7) expires. The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

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

WTP: This is the WTP's normal state of operation. The WTP resets its EchoInterval timer whenever it transmits a request to the AC. 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. When the WTP receives the Echo Response, it resets its EchoInterval timer (see Section 4.7.7).

Clear Config Request: The WTP receives a Clear Configuration Request message (see Section 8.8) and MUST generate a corresponding Clear Configuration Response message (see Section 8.9). 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 or Data Transfer Response message to the AC (see Section 9.6). The WTP receives a Data Transfer Request or Data Transfer Response message from the AC (see Section 9.6). Upon receipt of a Data Transfer Request, the WTP transmits a Data Transfer Response to the AC.

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. Note that the receipt of any Request from the WTP causes the AC to reset its EchoInterval timer (see Section 4.7.7).

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 sends a Clear Configuration Request message (see Section 8.8) to the WTP to clear its configuration. 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 sends a Data Transfer Request or Data Transfer Response message to the WTP (see Section 9.6). The AC receives a Data Transfer Request or Data Transfer Response message from the WTP (see Section 9.6). Upon receipt of a Data Transfer Request, the AC transmits a Data Transfer Response to the WTP.

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 (r): 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 (s): This transition occurs when the CAPWAP reset is complete, to terminate the DTLS session.

WTP: This state transition occurs when the WTP transmits a Reset Response message. The WTP does not invoke the DTLSShutdown command (see Section 2.3.2.1). The WTP starts the DTLSSessionDelete timer (see Section 4.7.6).

AC: This state transition occurs when the AC receives a Reset Response message. This causes the AC to initiate the DTLSShutdown command (see Section 2.3.2.1). The AC starts the DTLSSessionDelete timer (see Section 4.7.6).

DTLS Teardown to Idle (t): 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, or if the DTLSSessionDelete timer (see Section 4.7.6) expires. The data plane DTLS session is also shutdown, and all resources released, 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 is an invalid state transition for the AC.

DTLS Teardown to Sulking (u): This transition occurs when repeated attempts to setup the DTLS connection have failed.

WTP: The WTP enters this state when the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter reaches the value of the 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: This is an invalid state transition for the AC.

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

WTP: This is an invalid state transition for the WTP.

AC: This state transition occurs when the AC has successfully cleaned up all resources associated with the control plane DTLS session , or if the DTLSSessionDelete timer (see Section 4.7.6) expires. The data plane DTLS session is also shutdown, and all resources released, if a DTLS session was established for the data plane. Any timers set for the current instance of the state machine are also cleared. The AC's Service thread is terminated.

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. See Section 3.5 for more information on MTU Discovery. 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 4 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 entities; 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 [RFC4347]. 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 shown in Figure 5. (NOTE: this example uses certificates, but preshared keys are also supported):

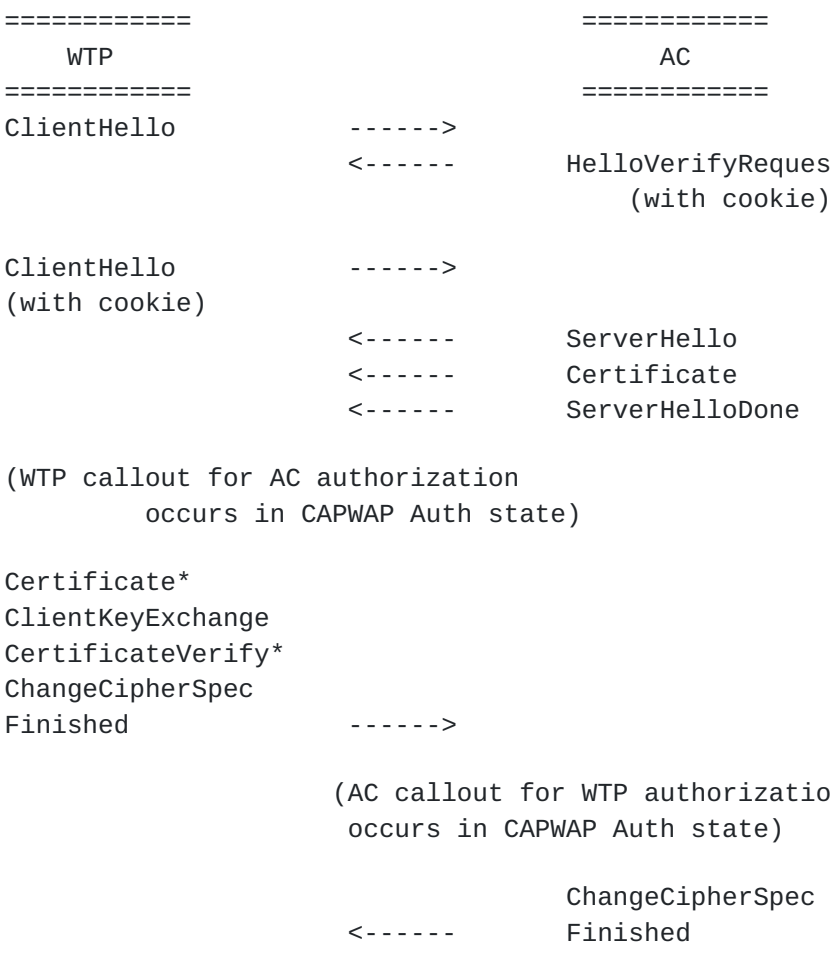


Figure 5: DTLS Handshake

DTLS, as specified, provides its own retransmit timers with an exponential back-off. [RFC4347] does not specify how long retransmissions should continue. Consequently, timing out incomplete DTLS handshakes is entirely the responsibility of the CAPWAP module.

The DTLS implementation used by CAPWAP MUST support TLS Session Resumption. Session resumption is typically used to establish the DTLS session used for the data channel. Since the data channel uses different port numbers than the control channel, the DTLS implementation on the WTP MUST provide an interface that allows the CAPWAP module to request session resumption despite the use of the different port numbers (TLS implementations usually attempt session resumption only when connecting to the same IP address and port number). Note that session resumption is not guaranteed to occur, and a full DTLS handshake may occur instead.

The DTLS implementation used by CAPWAP MUST use replay detection, per

Section 3.3 of [RFC4347]. Since the CAPWAP protocol handles retransmissions by re-encrypting lost frames, any duplicate DTLS frames are either unintentional or malicious, and should be silently discarded.

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 invoking the DTLSStart or DTLSListen commands, the WTP and AC, respectively, set 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 or WTP does not respond to any DTLS handshake messages sent by its peer, the DTLS specification calls for the message to be retransmited. Note that during the handshake, when both the AC and the WTP are expecting additional handshake messages, they both retransmit if an expected message has not been received (note that retransmissions for CAPWAP Control messages work differently: all CAPWAP Control messages are either requests or responses, and the peer who sent the request is responsible for retransmissions).

If the WTP or the AC does not receive an expected DTLS handshake message despite of retransmissions, the WaitDTLS timer will eventually expire, and the session will be terminated. This can happen if communication between the peers has completely failed, or if one of the peers sent a DTLS Alert message which was lost in transit (DTLS does not retransmit Alert messages).

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, and SHOULD treat the message as though no cookie were present.

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 value provided to the DTLS stack is the result of the MTU Discovery process, which is described in Section 3.5. 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

CAPWAP implementations only use cipher suites that are recommended for use with DTLS, see [DTLS-DESIGN]. At present, the following algorithms MUST be supported when using certificates for CAPWAP authentication:

- TLS_RSA_WITH_AES_128_CBC_SHA [RFC4346]

The following algorithms SHOULD be supported when using certificates:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA [RFC4346]

The following algorithms MAY be supported when using certificates:

- TLS_RSA_WITH_AES_256_CBC_SHA [RFC4346]

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA [RFC4346]

Additional ciphers MAY be defined in follow on CAPWAP specifications.

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 [RFC4279], and we focus on the following two:

- Pre-Shared Key (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 algorithm 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 [RFC4346]

- TLS_DHE_PSK_WITH_AES_128_CBC_SHA [RFC4346]

The following algorithms MAY be supported when using preshared keys:

- TLS_PSK_WITH_AES_256_CBC_SHA [RFC4346]

- TLS_DHE_PSK_WITH_AES_256_CBC_SHA [RFC4346]

Additional ciphers MAY be defined in follow on CAPWAP specifications.

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 [RFC5280].

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 described in [RFC5280] and [ISO.9834-1.1993] and 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 }
```

All capwap devices MUST support the ExtendedKeyUsage certificate extension if it is present in a certificate. If the extension is present, then the certificate MUST have either the id-kp-capwapAC or the id-kp-anyExtendedKeyUsage keyPurposeID to act as an AC. Similarly, if the extension is present, a device MUST have the id-kp-capwapWTP or id-kp-anyExtendedKeyUsage keyPurposeID to act as a WTP.

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. For instance, an AC SHOULD NOT accept a connection request from another AC, and therefore MUST verify that the id-kp-capwapWTP EKU is present in the certificate.

CAPWAP implementations MUST support certificates where the common name (CN) for both the WTP and AC is the MAC address of that device.

The MAC address MUST be encoded in the PrintableString format, using the well recognized MAC address format of 01:23:45:67:89:ab. The CN field MAY contain either of the EUI-48 [EUI-48] or EUI-64 [EUI-64] MAC Address formats. This seemingly unconventional use of the CN field is consistent with other standards that rely on device certificates that are provisioned during the manufacturing process, such as Packet Cable [PacketCable], Cable Labs [CableLabs] and WiMAX [WiMAX]. See Section 12.8 for more information on the use of the MAC Address in the CN field.

ACs and WTPs MUST authorize (e.g. through access control lists) certificates of devices to which they are connecting, e.g., based on the issuer, MAC address, or organizational information specified in the certificate. The identities specified in the certificates bind a particular DTLS session to a specific pair of mutually-authenticated and authorized MAC addresses. The particulars of authorization filter construction are implementation details which are, for the most part, not within the scope of this specification. However, at minimum, all devices MUST verify that the appropriate EKU bit is set according to the role of the peer device (AC vs. WTP), and that the issuer of the certificate is appropriate for the domain in question.

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 [RFC3828] transport protocols. When run over IPv4, UDP is used for the CAPWAP control and data channels.

When run over IPv6, the CAPWAP control channel always uses UDP, while the CAPWAP data channel may use either UDP or UDP-Lite. UDP-Lite is the default transport protocol for the CAPWAP data channel. However, if a middlebox or IPv4 to IPv6 gateway has been discovered, UDP is used for the CAPWAP data channel.

This section describes how the CAPWAP protocol is carried over IP and UDP/UDP-Lite transport protocols. The CAPWAP Transport Protocol message element Section 4.6.14 describes the rules to use in determining which transport protocol is to be used.

In order for CAPWAP to be compatible with potential middleboxes in the network, CAPWAP implementations MUST send return traffic from the same port on which it received traffic from a given peer. Further, any unsolicited requests generated by a CAPWAP node MUST be sent on the same port.

3.1. UDP Transport

One of the CAPWAP protocol requirements is to allow a WTP to reside behind a middlebox, 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. When CAPWAP is run over IPv4, 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 5246. 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 5247. If an AC permits the administrator to change the CAPWAP control port, the CAPWAP data port MUST be the next consecutive port number. 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 the default transport protocol, which reduces the checksum processing required for each packet (compared to the use of UDP over IPv6 [RFC2460]). When UDP-Lite is used, the checksum field MUST have a coverage of 8 [RFC3828].

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), the well known CAPWAP multicast address (xx.xx.xx.xx) 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. ACs, on the other hand, MUST support broadcast, multicast and unicast discovery.

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 [I-D.ietf-capwap-dhc-ac-option] for more information on the use of DHCP to discover AC IP addresses.

DNS: The WTP MAY support use of DNS SRV records [RFC2782] to discover the AC address(es). In this case, the WTP first obtains (e.g., from local configuration) the correct domain name suffix (e.g., "example.com") and performs a SRV lookup with Service name "capwap-control" and Proto "udp". Thus, the name resolved in DNS would be, e.g., "_capwap-control._udp.example.com". Note that the SRV record MAY specify a non-default port number for the control channel; the port number for the data channel is the next port number (control channel port + 1).

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 Priority 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 topology including firewall, NAT and middlebox devices.

It is important to note that the fragmentation mechanism employed by CAPWAP has known limitations and deficiencies, which are similar to those described in [RFC4963]. The limited size of the Fragment ID field (see Section 4.3 can cause wrapping of the field, and hence cause fragments from different datagrams to be incorrectly spliced together (known as "mis-associated"). For example, a 100Mpbs link with an MTU of 1500 (causing fragmentation at 1450 bytes) would cause the Fragment ID field wrap in 8 seconds. Consequently, CAPWAP implementors are warned to properly size their buffers for reassembly purposes based on the expected wireless technology throughput.

CAPWAP implementations SHOULD perform MTU Discovery (see Section 3.5), which can avoid the need for fragmentation. At the time of writing of this specification, most enterprise switching and routing infrastructure were capable of supporting "mini-jumbo" frames (1800 bytes), which eliminates the need for fragmentation (assuming the station's MTU is 1500 bytes). The need for fragmentation typically continues to exist when the WTP communicates with the AC over a Wide Area Network (WAN). Therefore, future versions of the CAPWAP protocol SHOULD either consider increasing the size of the Fragment ID field, or provide alternatives extensions.

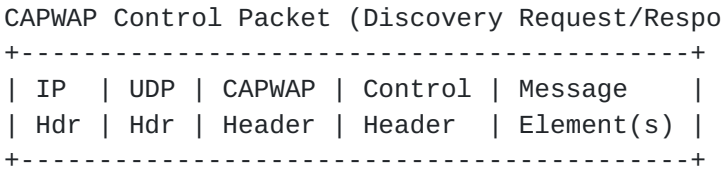
3.5. MTU Discovery

Once a WTP has discovered the AC it wishes to establish a CAPWAP session with, it SHOULD perform a Path MTU (PMTU) discovery. One recommendation for performing PMTU discovery is to have the WTP transmit Discovery Request (see Section 5.1) messages, and include the MTU Discovery Padding message element (see Section 4.6.32). The actual procedures used for PMTU discovery are described in [RFC1191], for IPv4, or for IPv6 [RFC1981] SHOULD be used. Alternatively, implementers MAY use the procedures defined in [RFC4821]. The WTP SHOULD also periodically re-evaluate the PMTU using the guidelines provided in these two RFCs, using the Primary Discovery Request (see Section 5.3) along with the MTU Discovery Padding message element (see Section 4.6.32). When the MTU is initially known, or updated in the case where an existing session already exists, the discovered PMTU is used to configure the DTLS component (see Section 2.3.2.1), while non-DTLS frames need to be fragmented to fit the MTU, defined in Section 3.4.

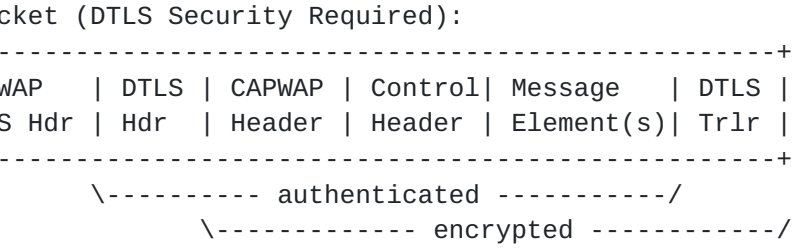
4. CAPWAP Packet Formats

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.

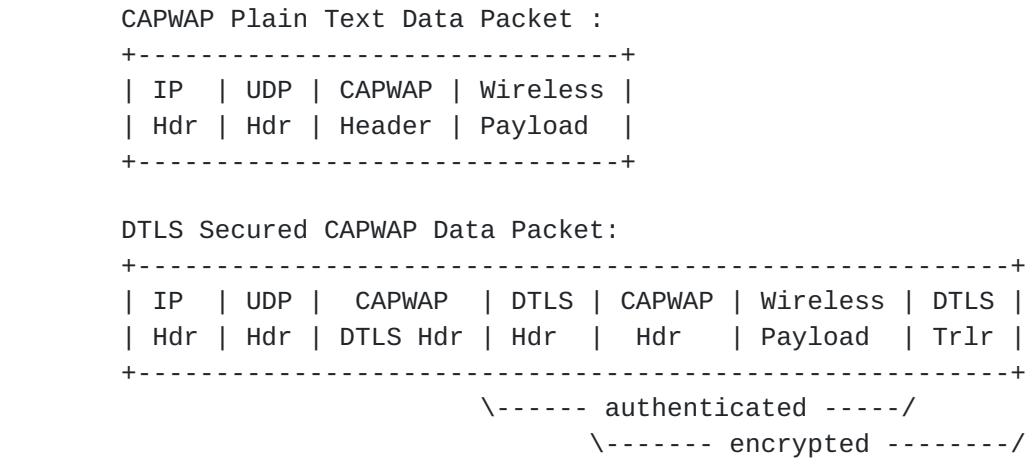
The CAPWAP Control protocol includes two messages that are never protected by DTLS: the Discovery Request message and the Discovery Response message. These messages need to be in the clear to allow the CAPWAP protocol to properly identify and process them. The format of these packets are as follows:



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:



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 [RFC4347].

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 signaling 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.

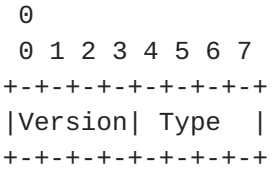
Message Elements: A CAPWAP Control packet includes one or more message elements, which are found immediately following the Control Header. These message elements are in a Type/Length/value style header, defined in Section 4.6.

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 4.6.31 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 preamble 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).

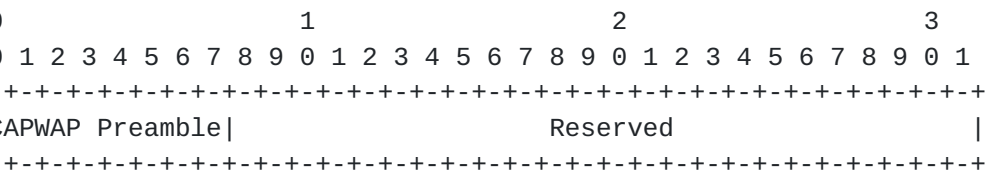
Type: A 4 bit field which specifies the payload type that follows the UDP header. The following values are supported:

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 [RFC4347] always immediately follows this header. The format of the CAPWAP DTLS Header is as follows:



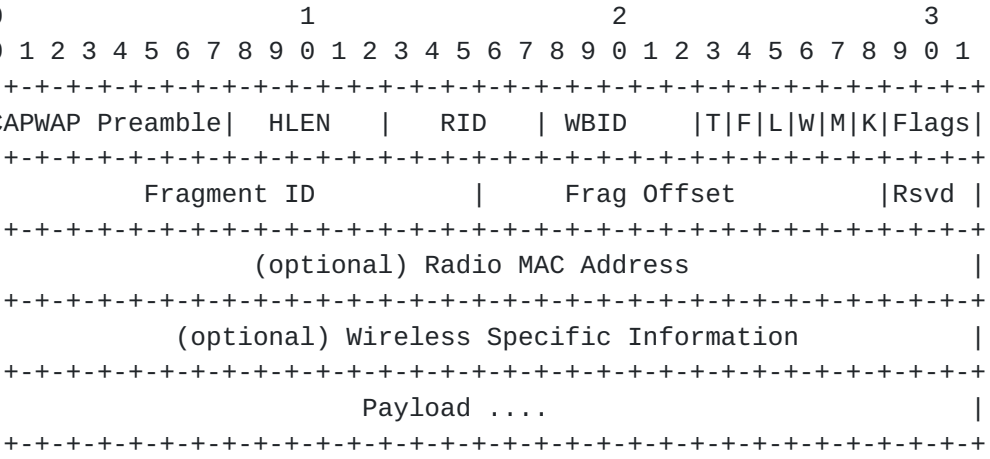
CAPWAP Preamble: The CAPWAP Preamble is defined in Section 4.1. The CAPWAP Preamble's Payload Type field MUST be set to one (1).

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, whose value is between one (1) and 31. Given that MAC Addresses are not necessarily unique across physical radios in a WTP, the Radio Identifier (RID) field is used to indicate 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 associated with the radio. The following values are defined:

0 - Reserved

1 - IEEE 802.11

2 - Reserved

3 - EPCGlobal [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.

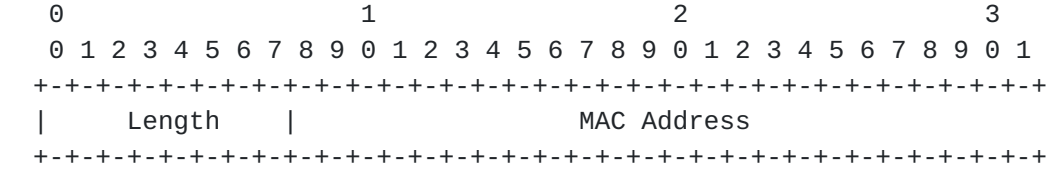
Fragment ID: A 16 bit field whose value is assigned to each group of fragments making up a complete set. The fragment ID space is managed individually for each direction for every WTP/AC pair. The value of Fragment ID is incremented with each new set of fragments. The Fragment ID wraps to zero after 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. Because the native wireless frame format to IEEE 802.3 format causes the MAC address of the WTP's radio to be lost, this field allows the address to be communicated to the AC. 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:

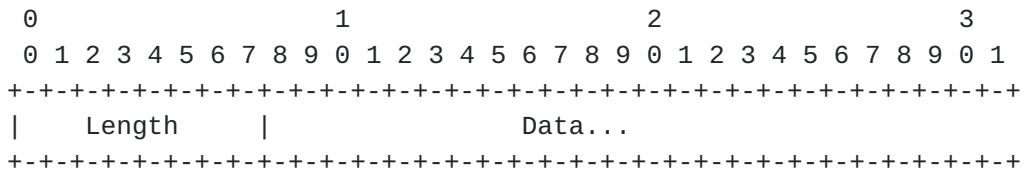


Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

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 WBID field in the CAPWAP header is used to identify the format of the wireless specific information optional field. 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:



Length: The 8 bit field contains the length of the data field, with a maximum size of 255.

Data: Wireless specific information, defined by the wireless specific binding specified in the CAPWAP Header's WBID field.

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 Channel Keepalive

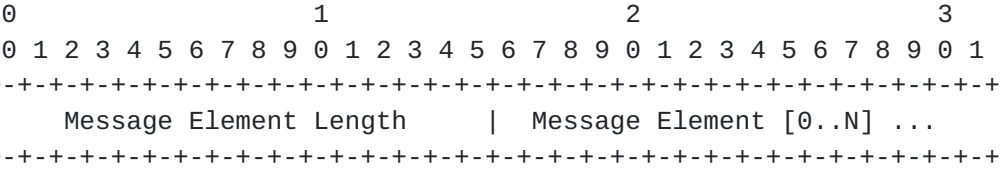
The CAPWAP Data Channel Keep Alive packet is used to bind the CAPWAP control channel with the data channel, and to maintain freshness of the data channel, ensuring that the channel is still functioning. The CAPWAP Data Channel Keep Alive packet is transmitted by the WTP

when the DataChannelKeepAlive timer expires (see Section 4.7.2). 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 16 bit Length field indicates the number of bytes following the CAPWAP Header, with a maximum size of 65535.

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

Session ID, see Section 4.6.37

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 for 802.11 frames, the 802.11 *Integration* 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.

For 802.3 payload frames, the 802.3 frame is encapsulated (excluding the IEEE 802.3 Preamble, Start Frame Delimiter (SFD) and Frame Check Sequence (FCS) Fields). 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. The CAPWAP protocol can support IEEE 802.3 frames whose length is defined in the IEEE 802.3as specification [FRAME-EXT].

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 SHOULD be initiated using the TLS session resumption feature [RFC4346].

The AC DTLS implementation MUST NOT initiate a data channel session for a DTLS session for which there is no active control channel session.

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.

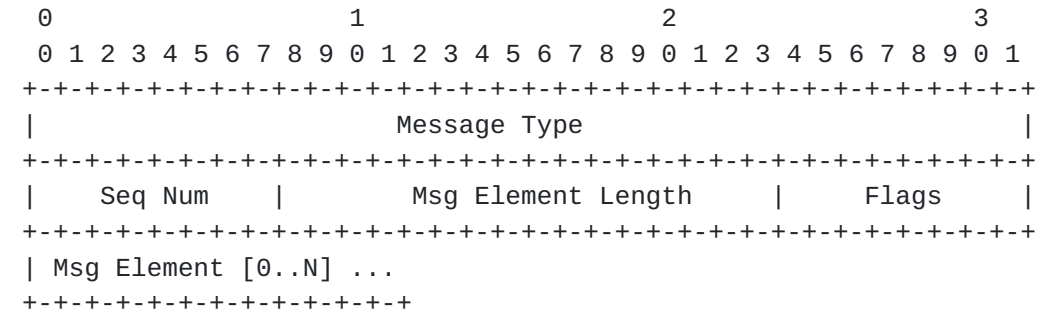
Binding Specific CAPWAP Management Messages: Messages in this category are used by the AC and the WTP to exchange protocol-specific CAPWAP management messages. These messages may or may not be used to change the link state of a station.

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. To provide extensibility, the Message Type field is comprised of an IANA Enterprise Number [RFC3232] and an enterprise

specific message type number. The first three octets contain the IANA Enterprise Number in network byte order, with zero used for CAPWAP base protocol (this specification) defined 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 Request	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. Quality of Service

The CAPWAP base protocol does not provide any Quality of Service (QoS) recommendations for use with the CAPWAP data messages. Any wireless specific CAPWAP binding specification that has QoS requirements MUST define the application of QoS to the CAPWAP data messages.

The IP header also includes the Explicit Congestion Notification (ECN) bits [RFC3168]. Section 9.1.1 of [RFC3168] describes two levels of ECN functionality, full functionality and limited functionality. CAPWAP ACs and WTPs SHALL implement the limited functionality and are RECOMMENDED to implement the full functionality described in [RFC3168].

4.5.2.1. Applying QoS to CAPWAP Control Message

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:

802.1Q: The priority tag of 7 SHOULD be used.

DSCP: The CS6 per-hop behavior Service Class SHOULD be used, which is described in [RFC2474]).

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 MUST keep track of the Sequence Number of the last received Request message, and MUST cache the corresponding Response message. If a retransmission with the same Sequence Number is received, the cached Response message MUST be retransmitted without re-processing the Request. If an older Request message is received, meaning one where the sequence number is smaller, it MUST be ignored. A newer Request message, meaning one whose sequence number is larger, is processed as usual.

Note: A sequence number is considered "smaller" when s1 is smaller than s2 modulo 256 if and only if (s1<s2 and (s2-s1)<128) or (s1>s2 and (s1-s2)>128)

Both the WTP and the AC can only have a single request outstanding at any given time. Retransmitted Request messages MUST NOT be altered by the sender.

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 4.7.7). Response messages are not subject to these timers.

If the sender stops retransmitting a Request message before reaching MaxRetransmit retransmissions (which leads to transition to DTLS Teardown, as described in Section 2.3.1), it cannot know whether the

recipient received and processed the Request or not. In most situations, the sender SHOULD NOT do this, and instead continue retransmitting until a Response message is received, or transition to DTLS Teardown occurs. However, if the sender does decide to continue the connection with a new or modified Request message, the new message MUST have a new Sequence Number, and be treated as a new Request message by the receiver. Note that there is a high chance that both the WTP and the AC's sequence numbers will become out of sync.

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.

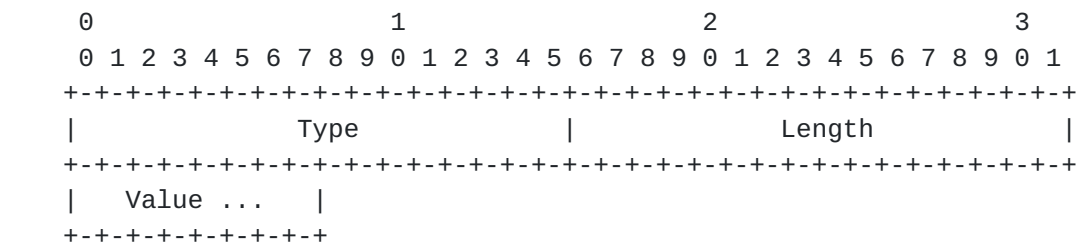
All of the message element definitions in this document use a diagram similar to the one below in order to depict its format. Note that to simplify this specification, these diagrams do not include the header fields (Type and Length). The header field values are defined in the message element descriptions.

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.

Unless otherwise specified, any configuration information sent by the AC to the WTP MAY be saved to non-volatile storage (see Section 8.1) for more information).

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. The value of zero (0) is reserved and MUST NOT be used. The rest of the Type field values are allocated as follows:

Usage	Type Values
CAPWAP Protocol Message Elements	1 - 1023
IEEE 802.11 Message Elements	1024 - 2047
Reserved for Future Use	2048 - 3071
EPCGlobal Message Elements	3072 - 4095
Reserved for Future Use	4096 - 65535

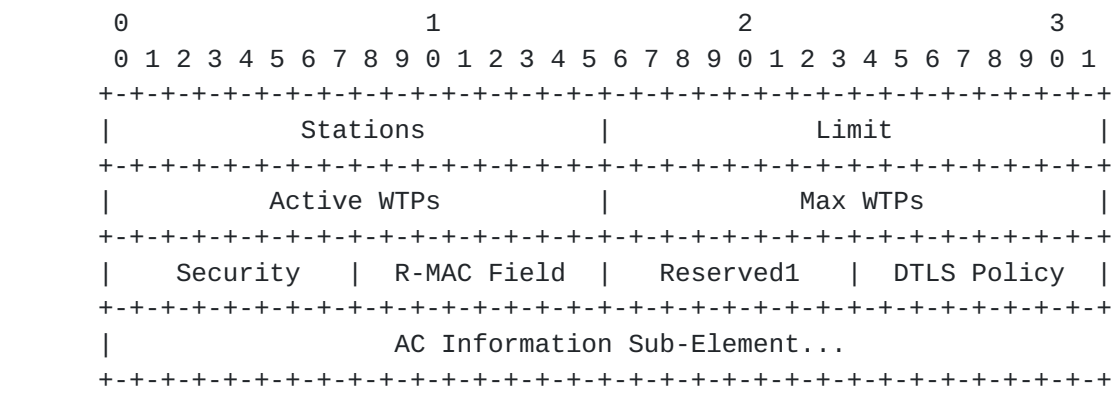
The table below lists the CAPWAP protocol Message Elements and their Type values.

CAPWAP Message Element	Type Value
AC Descriptor	1
AC IPv4 List	2
AC IPv6 List	3
AC Name	4
AC Name with Priority	5
AC Timestamp	6
Add MAC ACL Entry	7
Add Station	8
Reserved	9
CAPWAP Control IPV4 Address	10
CAPWAP Control IPV6 Address	11
CAPWAP Local IPV4 Address	30
CAPWAP Local IPV6 Address	50
CAPWAP Timers	12
CAPWAP Transport Protocol	51
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
Reserved	19
Discovery Type	20
Duplicate IPv4 Address	21
Duplicate IPv6 Address	22
ECN Support	53
Idle Timeout	23
Image Data	24
Image Identifier	25
Image Information	26
Initiate Download	27
Location Data	28
Maximum Message Length	29
MTU Discovery Padding	52
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
Reserved	42
Reserved	43
WTP MAC Type	44
WTP Name	45
Unused/Reserved	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

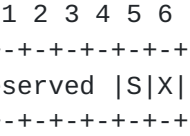
Stations: The number of stations currently served by the AC

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 mask specifying the authentication credential type supported by the AC (See Section 2.4.4). The field has the following format:



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.

S: The AC supports the pre-shared secret authentication, as described in Section 12.6.

X: The AC supports X.509 Certificate authentication, as described in Section 12.7.

R: A reserved bit 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.

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

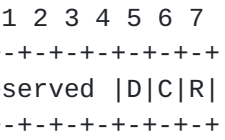
0 - Reserved

1 - Supported

2 - Not Supported

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 field has the following format:



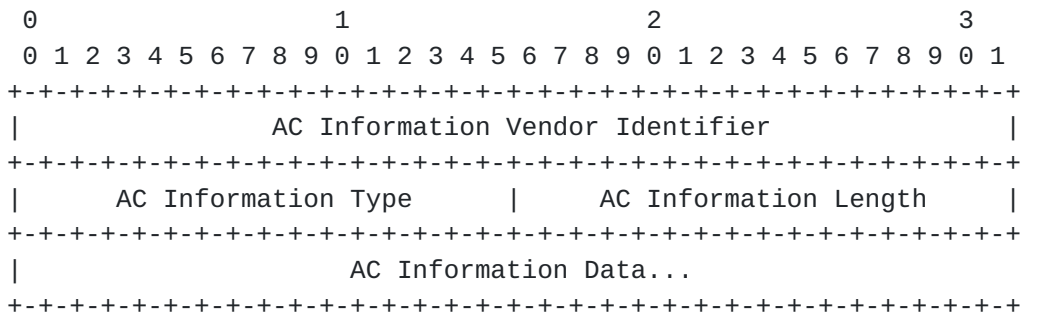
Reserved: A set of reserved bits for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

D: DTLS Enabled Data Channel Supported

C: Clear Text Data Channel Supported

R: A reserved bit 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.

AC Information Sub-Element: The AC Descriptor message element contains multiple AC Information sub-elements, and defines two sub-types, each of which MUST be present. The AC Information sub-element has the following format:



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

AC Information Type: Vendor specific encoding of AC information in the UTF-8 format [RFC3629]. The following enumerated values are supported. Both the Hardware and Software Version sub-elements MUST be included in the AC Descriptor message element. The values listed below are used in conjunction with the AC Information Vendor Identifier field, whose value MUST be set to zero (0). This field, combined with the AC Information Vendor Identifier set to a non-zero (0) value, allows vendors to use a private namespace.

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

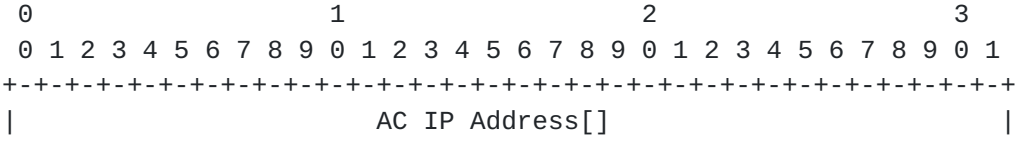
5 - Software Version: The AC's Software (firmware) version number.

AC Information Length: Length of vendor specific encoding of AC information, with a maximum size of 1024.

AC Information Data: 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.



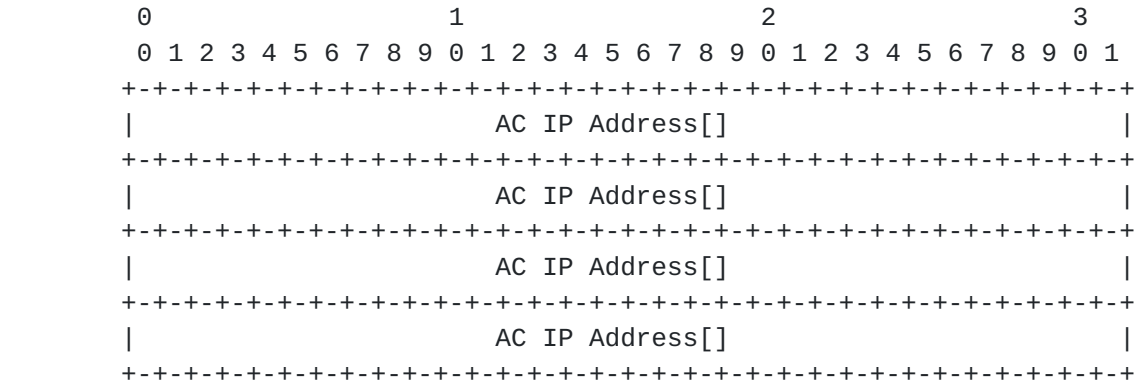
Type: 2 for AC IPv4 List

Length: >= 4

AC IP Address: An array of 32-bit integers containing AC IPv4 Addresses, containing no more than 1024 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.



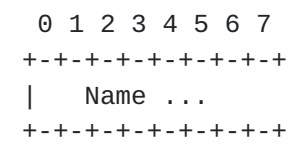
Type: 3 for AC IPV6 List

Length: >= 16

AC IP Address: An array of 128-bit integers containing AC IPv6 Addresses, containing no more than 1024 addresses.

4.6.4. AC Name

The AC Name message element contains an UTF-8 [RFC3629] representation of the AC identity. The value is a variable length byte string. The string is NOT zero terminated.



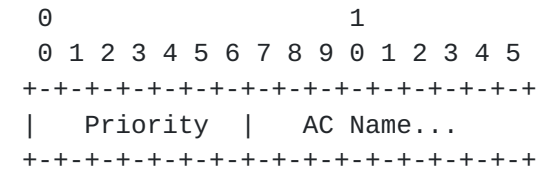
Type: 4 for AC Name

Length: >= 1

Name: A variable length UTF-8 encoded string [RFC3629] containing the AC's name, whose maximum size MUST NOT exceed 512 bytes.

4.6.5. AC Name with Priority

The AC Name with Priority 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. The WTP also uses this message element to send its configuration to the AC.



Type: 5 for AC Name with Priority

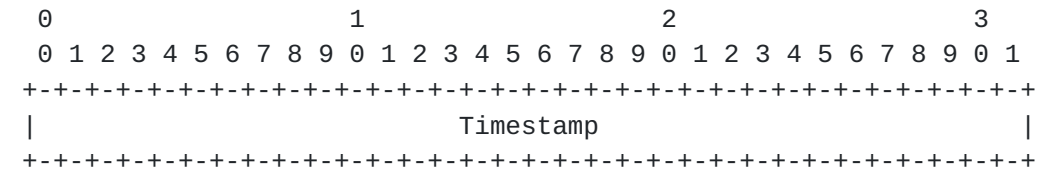
Length: >= 2

Priority: A value between 1 and 255 specifying the priority order of the preferred AC. For instance, the value of one (1) is used to set the primary AC, the value of two (2) is used to set the secondary, etc.

AC Name: A variable length UTF-8 encoded string [RFC3629] containing the AC name, whose maximum size MUST NOT exceed 512 bytes.

4.6.6. AC Timestamp

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



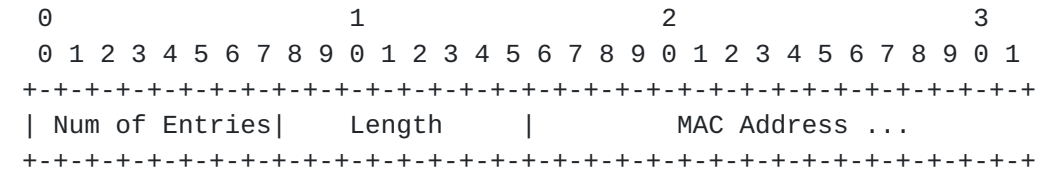
Type: 6 for AC Timestamp

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 [RFC1305]. Only the most significant 32 bits of the NTP time is included in this field.

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. The MAC ACL table on the WTP is cleared everytime the WTP establishes a new session with an AC.



Type: 7 for Add MAC ACL Entry

Length: >= 8

Num of Entries: The number of instances of the Length/MAC Addresses fields in the array. This value MUST NOT exceed 255.

Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

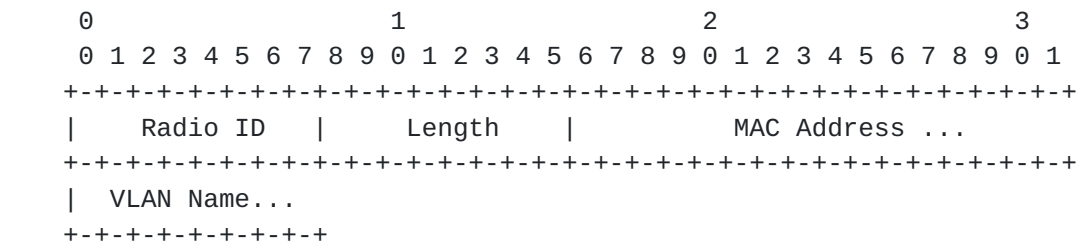
MAC Address: MAC Addresses to add to the ACL.

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, whose value is between one (1) and 31.

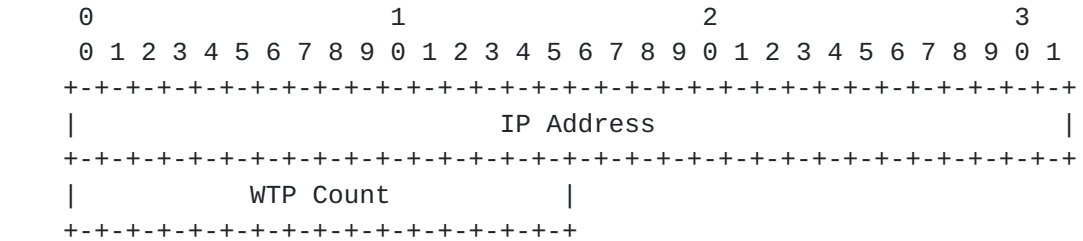
Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

MAC Address: The station's MAC Address

VLAN Name: An optional variable length UTF-8 encoded string[RFC3629], with a maximum length of 512 octets, 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. 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 (see Section 6.1).



Type: 10 for CAPWAP Control IPv4 Address

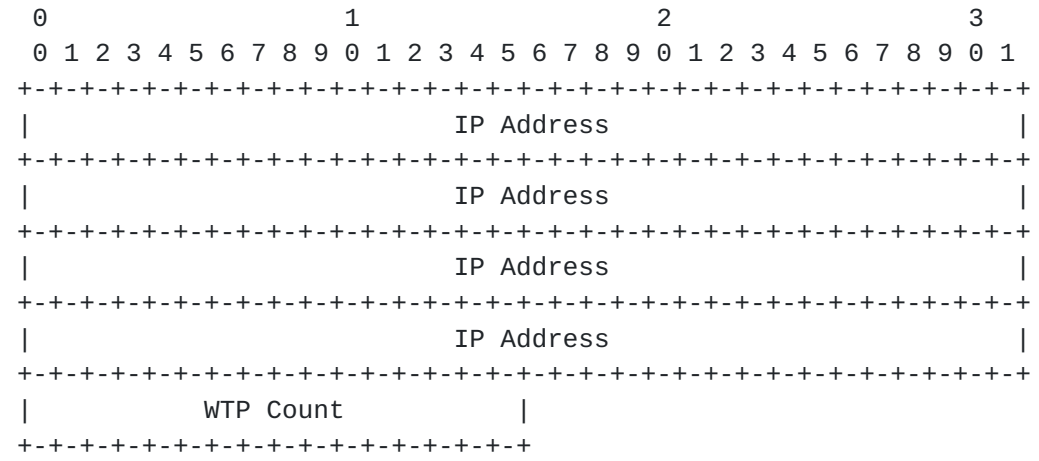
Length: 6

IP Address: The IP Address of an interface.

WTP Count: The number of WTPs currently connected to the interface, with a maximum value of 65535.

4.6.10. 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 (see Section 6.1).



Type: 11 for CAPWAP Control IPv6 Address

Length: 18

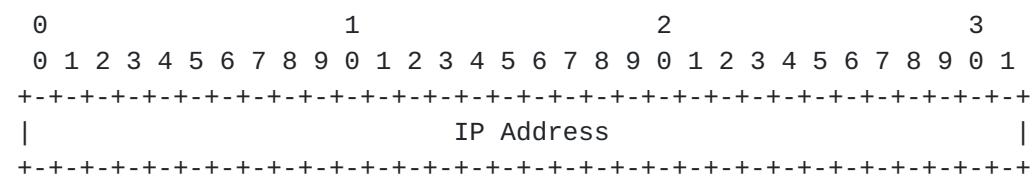
IP Address: The IP Address of an interface.

WTP Count: The number of WTPs currently connected to the interface, with a maximum value of 65535.

4.6.11. CAPWAP Local IPv4 Address

The CAPWAP Local IPv4 Address message element is sent by either the WTP, in the Join Request, or by the AC, in the Join Response. The CAPWAP Local IPv4 Address message element is used to communicate the IP Address of the transmitter. The receiver uses this to determine whether a middlebox exists between the two peers, by comparing the source IP address of the packet against the value of the message

element.



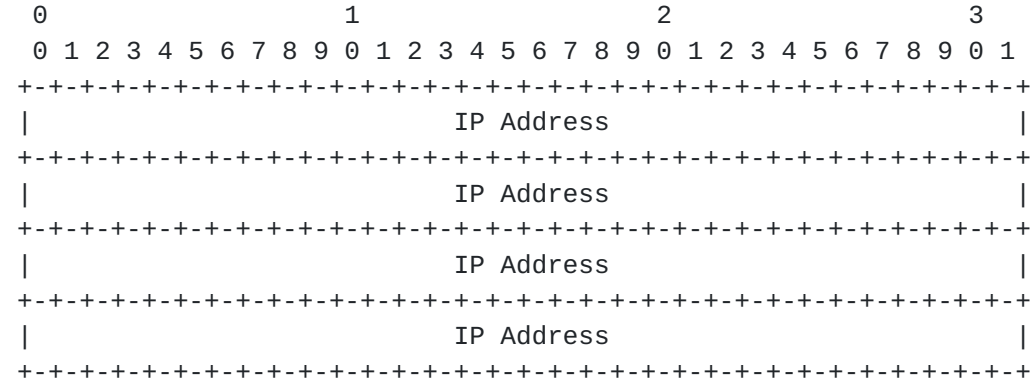
Type: 30 for CAPWAP Local IPv4 Address

Length: 4

IP Address: The IP Address of the sender.

4.6.12. CAPWAP Local IPv6 Address

The CAPWAP Local IPv6 Address message element is sent by either the WTP, in the Join Request, or by the AC, in the Join Response. The CAPWAP Local IPv6 Address message element is used to communicate the IP Address of the transmitter. The receiver uses this to determine whether a middlebox exists between the two peers, by comparing the source IP address of the packet against the value of the message element.



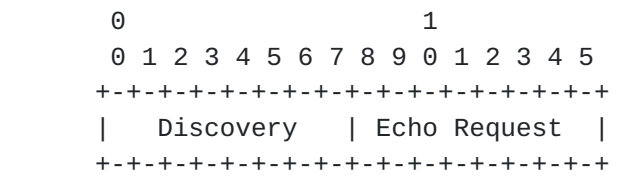
Type: 50 for CAPWAP Local IPv6 Address

Length: 16

IP Address: The IP Address of the sender.

4.6.13. 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. This value is used to configure the MaxDiscoveryInterval timer (see Section 4.7.10).

Echo Request: The number of seconds between WTP Echo Request CAPWAP messages. This value is used to configure the EchoInterval timer (see Section 4.7.7). The AC sets its EchoInterval timer to this value, plus the maximum retransmission time as described in Section 4.5.3.

4.6.14. CAPWAP Transport Protocol

When CAPWAP is run over IPv6, the UDP-Lite or UDP transports MAY be used (see Section 3). The CAPWAP IPv6 Transport Protocol message element is used by either the WTP or the AC to signal which transport protocol is to be used for the CAPWAP data channel.

Upon receiving the Join Request, the AC MAY set the CAPWAP Transport Protocol to UDP-Lite in the Join Response message if the CAPWAP message was received over IPv6, and the CAPWAP Local IPv6 Address message element (see Section 4.6.12) is present and no middlebox was detected (see Section 11).

Upon receiving the Join Response, the WTP MAY set the CAPWAP Transport Protocol to UDP-Lite in the Configuration Status Request or Image Data Request message if the AC advertised support for UDP-Lite, the message was received over IPv6, the CAPWAP Local IPv6 Address message element (see Section 4.6.12) and no middlebox was detected (see Section 11). Upon receiving either the Configuration Status Request or the Image Data Request, the AC MUST observe the preference indicated by the WTP in the CAPWAP Transport Protocol, as long as it is consistent with what the AC advertised in the Join Response.

For any other condition, the CAPWAP Transport Protocol MUST be set to UDP.

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|   Transport   |
+-+-+-+-+-+-+-+-+
```

Type: 51 for CAPWAP Transport Protocol

Length: 1

Transport: The transport to use for the CAPWAP data channel. The following enumerated values are supported:

1 - UDP-Lite: The UDP-Lite transport protocol is to be used for the CAPWAP data channel. Note that this option MUST NOT be used if the CAPWAP control channel is being used over IPv4.

2 - UDP: The UDP transport protocol is to be used for the CAPWAP data channel.

4.6.15. Data Transfer Data

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Data Type   |   Data Mode   |         Data Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Data ....
+-+-+-+-+-+-+-+-+
```

Type: 13 for Data Transfer Data

Length: >= 5

Data Type: An 8-bit value representing the transfer Data Type. The following enumerated values are supported:

1 - Transfer data is included

2 - Last Transfer Data Block is included (EOF)

5 - An error occurred. Transfer is aborted

Data Mode: An 8-bit value the type of information being transmitted. The following enumerated values are supported:

0 - Reserved

1 - WTP Crash Data

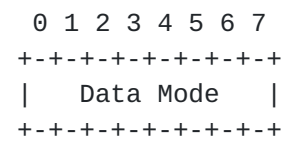
2 - WTP Memory Dump

Data Length: Length of data field, with a maximum size of 65535.

Data: Data being transferred from the WTP to the AC, whose type is identified via the Data Mode field.

4.6.16. 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 Mode: An 8-bit value the type of information being requested. The following enumerated values are supported:

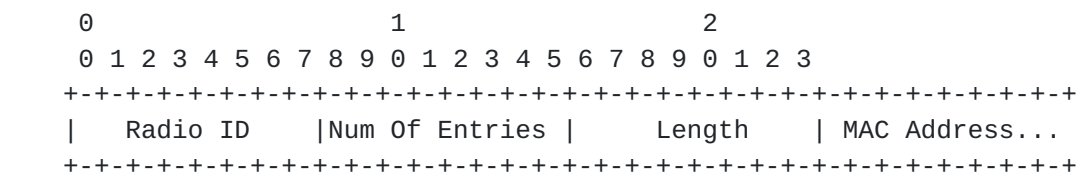
0 - Reserved

1 - WTP Crash Data

2 - WTP Memory Dump

4.6.17. 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, whose value is between one (1) and 31.

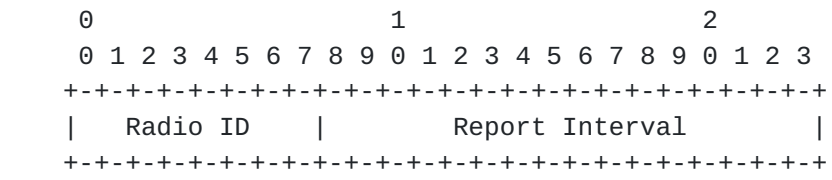
Num of Entries: The number of instances of the Length/MAC Addresses fields in the array. This field MUST NOT exceed the value of 255.

Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

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

4.6.18. 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

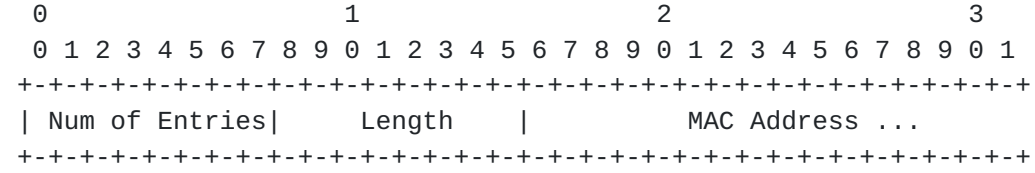
Length: 3

Radio ID: The Radio Identifier refers to an interface index on the WTP, whose value is between one (1) and 31.

Report Interval: A 16-bit unsigned integer indicating the time, in seconds. The default value for this message element can be found in Section 4.7.11.

4.6.19. 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 Length/MAC Addresses fields in the array. This field MUST NOT exceed the value of 255.

Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

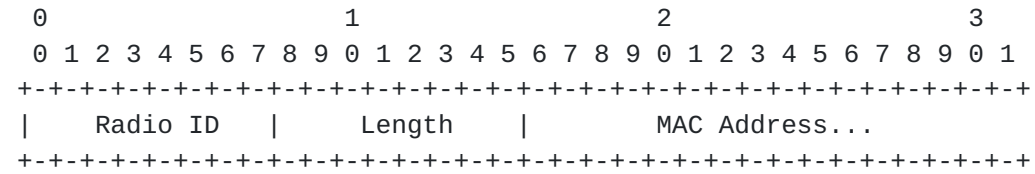
MAC Address: An array of MAC Addresses to delete from the ACL.

4.6.20. 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

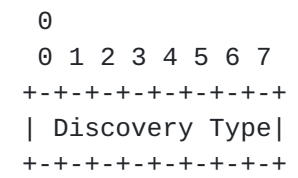
Radio ID: An 8-bit value representing the radio, whose value is between one (1) and 31.

Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

MAC Address: The station's MAC Address

4.6.21. 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 enumerated values are supported:

0 - Unknown

1 - Static Configuration

2 - DHCP

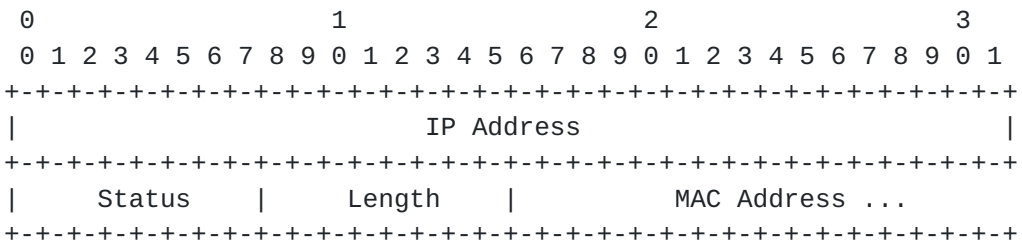
3 - DNS

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

4.6.22. 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.

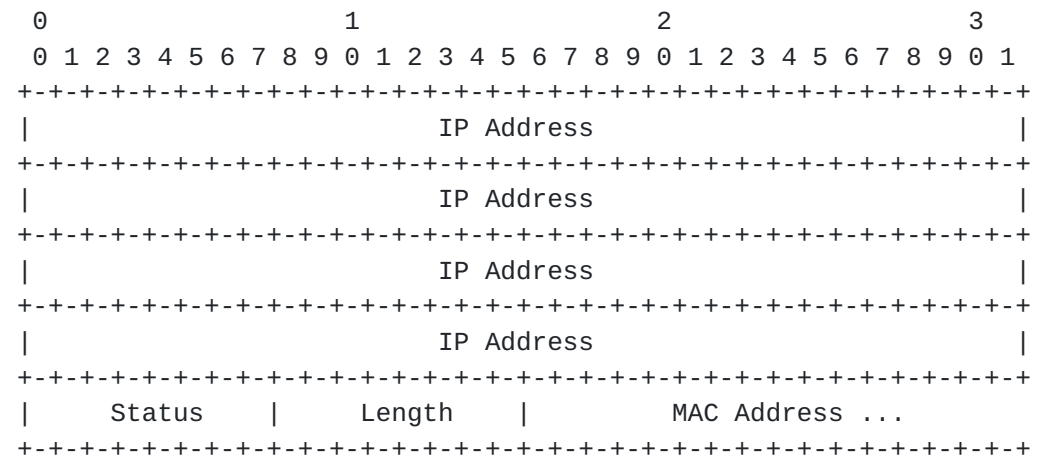
Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

MAC Address: The MAC Address of the offending device.

4.6.23. 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: 22 for Duplicate IPv6 Address

Length: >= 24

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

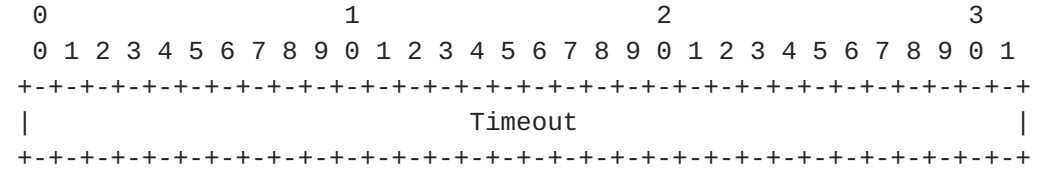
Status: The status of the duplicate IP address. The value MUST be set to 1 when a duplicate address is detected, and 0 when the duplicate address has been cleared.

Length: The length of the MAC Address field. The following formats, and lengths, are supported [EUI-48] and [EUI-64].

MAC Address: The MAC Address of the offending device.

4.6.24. 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.



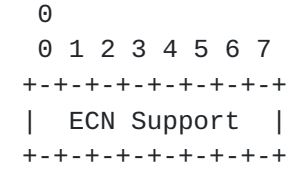
Type: 23 for Idle Timeout

Length: 4

Timeout: The current idle timeout, in seconds, to be enforced by the WTP. The default value for this message element is specified in Section 4.7.8.

4.6.25. ECN Support

The ECN Support message element is sent by both the WTP and the AC to indicate their support for the Explicit Congestion Notification (ECN) bits, as defined in [RFC3168].



Type: 53 for ECN Support

Length: 1

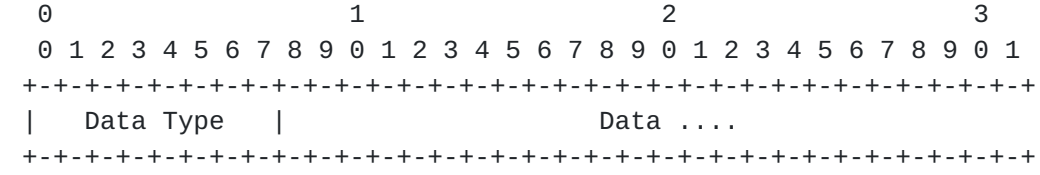
ECN Support: An 8-bit value representing the sender's support for ECN, as defined in [RFC3168].

0 - Limited ECN Support

1 - Full and Limited ECN Support

4.6.26. 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

Data Type: An 8-bit value representing the image Data Type. The following enumerated values are supported:

1 - Image data is included

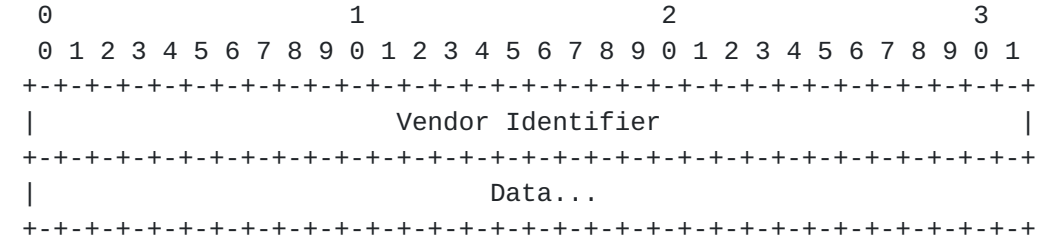
2 - Last Image Data Block is included (EOF)

5 - An error occurred. Transfer is aborted

Data: The Image Data field contains up to 1024 characters, and its length is inferred from this message element's length field. If the block being sent is the last one, the Data Type field is set to 2. The AC MAY opt to abort the data transfer by setting the Data Type field to 5. When the Data Type field is 5, the Value field has a zero length.

4.6.27. Image Identifier

The Image Identifier message element is sent by the AC to the WTP to indicate the expected active software version that is to be run on the WTP. The WTP sends the Image Identifier message element in order to request a specific software version from the AC. The actual download process is defined in Section 9.1. The value is a variable length UTF-8 encoded string [RFC3629], which is NOT zero terminated.



Type: 25 for Image Identifier

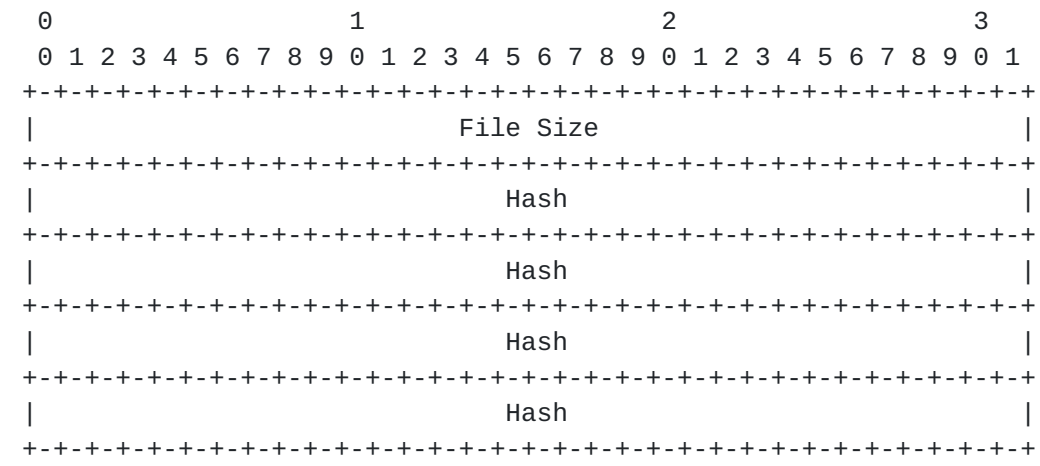
Length: >= 5

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

Data: A variable length UTF-8 encoded string [RFC3629] containing the firmware identifier to be run on the WTP, whose length MUST NOT exceed 1024 octets. The length of this field is inferred from this message element's length field.

4.6.28. 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: 20

File Size: A 32-bit value containing the size of the file, in bytes, that will be transferred by the AC to the WTP.

Hash: A 16 octet MD5 hash of the image using the procedures defined in [RFC1321].

4.6.29. Initiate Download

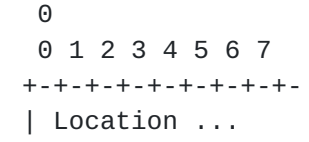
The Initiate Download message element is used by the WTP to inform the AC that the AC SHOULD initiate a firmware upgrade. The AC subsequently transmits an Image Data Request message which includes the Image Data message element. This message element does not contain any data.

Type: 27 for Initiate Download

Length: 0

4.6.30. Location Data

The Location Data message element is a variable length byte UTF-8 encoded string [RFC3629] 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: >= 1

Location: A non-zero terminated UTF-8 encoded string [RFC3629] containing the WTP location, whose maximum size MUST NOT exceed 1024.

4.6.31. 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.

```
      0               1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    Maximum Message Length     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 29 for Maximum Message Length

Length: 2

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

4.6.32. 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.

```
 0
   0 1 2 3 4 5 6 7
  +-+-+-+-+-+-+-+-+
  |  Padding...
  +-+-+-+-+-+-+-+-
```

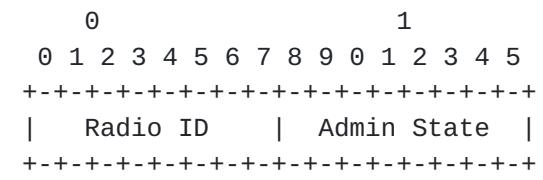
Type: 52 for MTU Discovery Padding

Length: variable

Pad: A variable length pad, filled with the value 0xFF.

4.6.33. 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, whose value is between one (1) and 31. 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 enumerated values are supported:

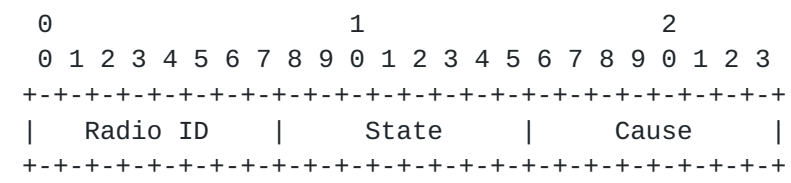
0 - Reserved

1 - Enabled

2 - Disabled

4.6.34. 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, whose value is between one (1) and 31. 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. The following enumerated values are supported:

0 - Reserved

1 - Enabled

2 - Disabled

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

0 - Normal

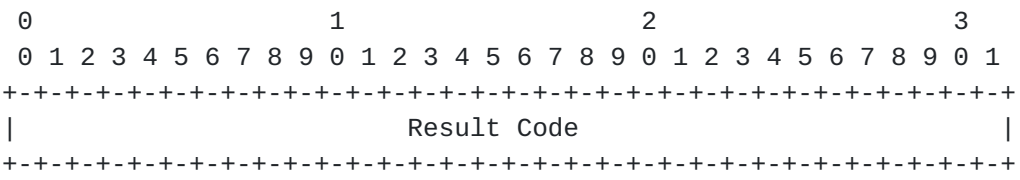
1 - Radio Failure

2 - Software Failure

3 - Administratively Set

4.6.35. 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 enumerated 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)

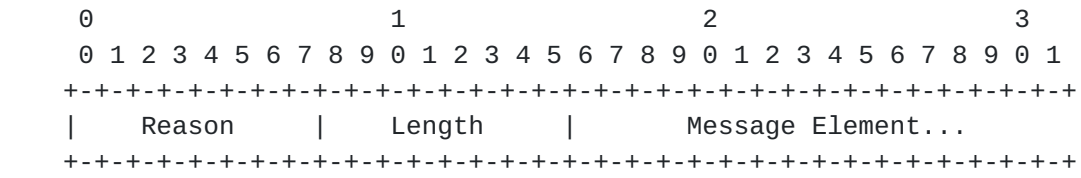
20 Failure - Missing Mandatory Message Element

21 Failure - Unrecognized Message Element

22 Data Transfer Error (No Information to Transfer)

4.6.36. 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: >= 6

Reason: The reason why the configuration in the offending message element could not be applied by the WTP. The following enumerated values are supported:

0 - Reserved

1 - Unknown Message Element

2 - Unsupported Message Element

3 - Unknown Message Element Value

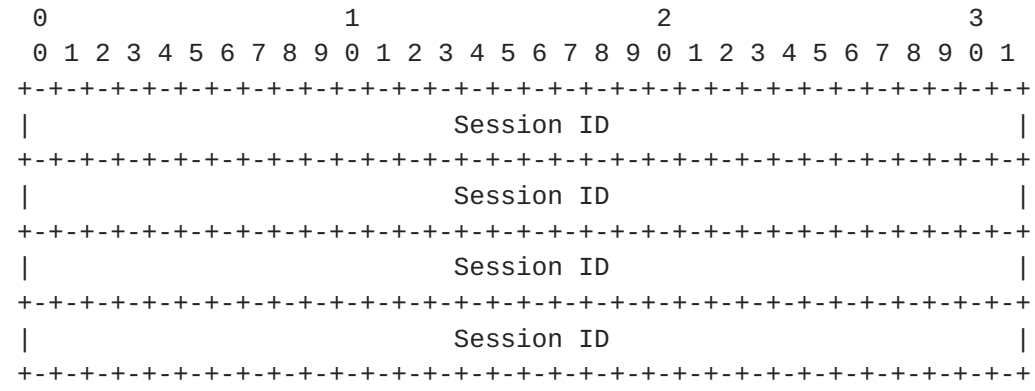
4 - Unsupported Message Element Value

Length: The length of the Message Element field, which MUST NOT exceed 255 octets.

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

4.6.37. Session ID

The Session ID message element value contains a randomly generated unsigned 128-bit integer.



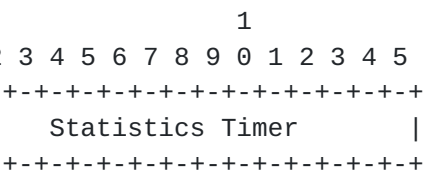
Type: 35 for Session ID

Length: 16

Session ID: A 128-bit unsigned integer used as a random session identifier

4.6.38. 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.



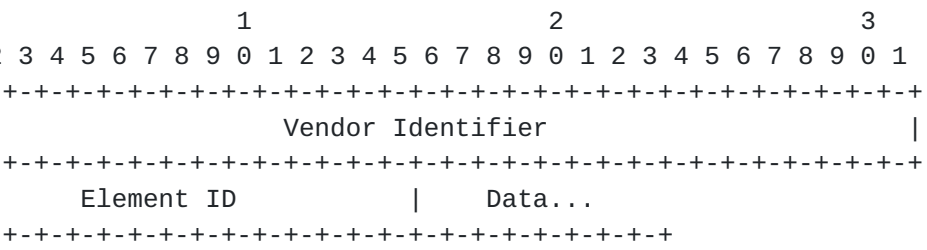
Type: 36 for Statistics Timer

Length: 2

Statistics Timer: A 16-bit unsigned integer indicating the time, in seconds. The default value for this timer is specified in Section 4.7.14.

4.6.39. Vendor Specific Payload

The Vendor Specific Payload message element is used to communicate vendor specific information between the WTP and the AC. The Vendor Specific Payload message element MAY be present in any CAPWAP message. The exchange of vendor specific data between the MUST NOT modify the behavior of the base CAPWAP protocol and state machine. The message element uses the following format:



Type: 37 for Vendor Specific Payload

Length: >= 7

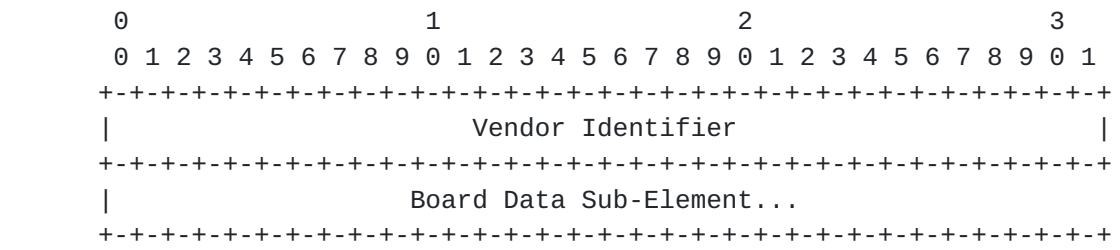
Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes" [RFC3232]

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

Data: Variable length vendor specific information, whose contents and format are proprietary and understood based on the Element ID field. This field MUST NOT exceed 2048 octets.

4.6.40. WTP Board Data

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

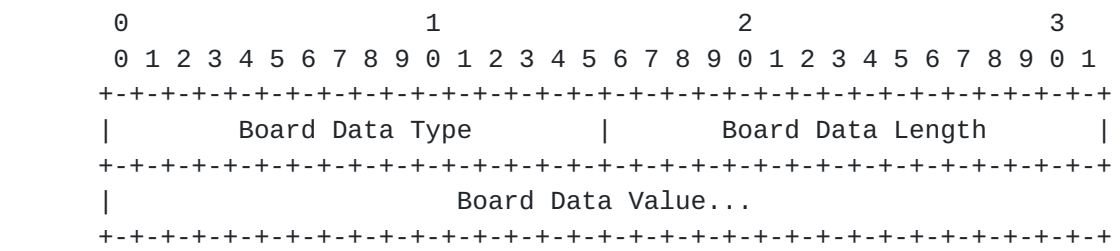


Type: 38 for WTP Board Data

Length: >=14

Vendor Identifier: A 32-bit value containing the IANA assigned "SMI Network Management Private Enterprise Codes", identifying the WTP hardware manufacturer. The Vendor Identifier field MUST NOT be set to zero.

Board Data Sub-Element: The WTP Board Data message element contains multiple Board Data sub-elements, some of which are mandatory and some are optional, as described below. The Board Data Type values are not extensible by vendors, and is therefore not coupled along with the Vendor Identifier field. The Board Data sub-element has the following format:



Board Data Type: The Board Data Type field identifies the data being encoded. The CAPWAP protocol defines the following values, and each of these types identify whether their presence is mandatory or optional:

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 message element.

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

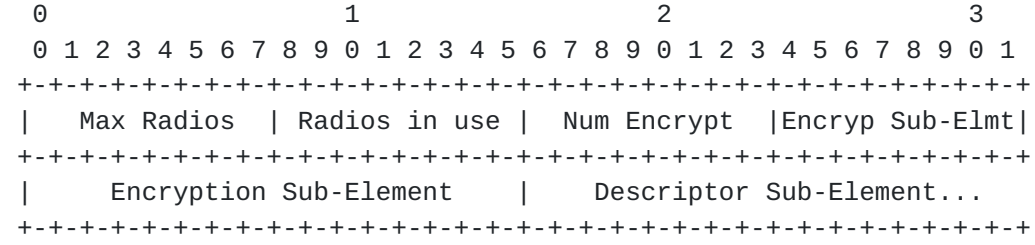
4 - Base MAC Address The WTP's Base MAC Address, which MAY be assigned to the primary Ethernet interface.

Board Data Length: The length of the data in the Board Data Value field, whose length MUST NOT exceed 1024 octets.

Board Data Value: The data associated with the Board Data Type field for this Board Data sub-element.

4.6.41. 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

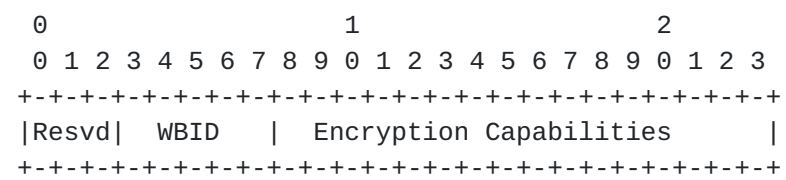
Length: >= 33

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.

Num Encrypt: The number of 3 byte Encryption Sub-Elements that follow this field. The value of the Num Encrypt field MUST be between one (1) and 255.

Encryption Sub-Element: The WTP Descriptor message element MUST contain at least one Encryption sub-element. One sub-element is present for each binding supported by the WTP. The Encryption sub-element has the following format:

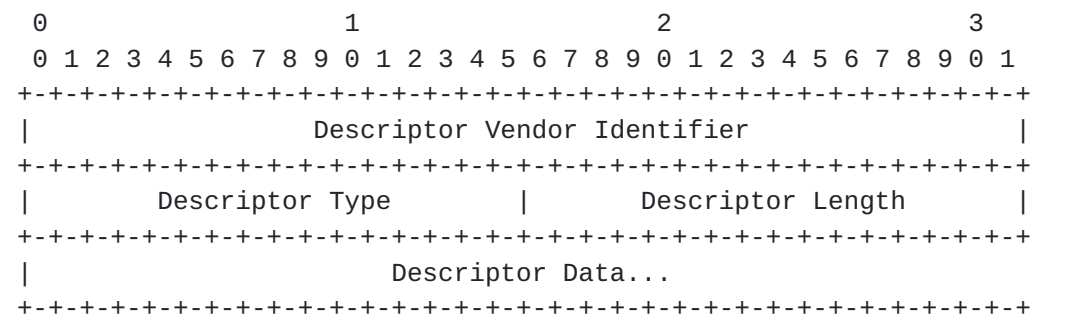


Resvd: The 3-bit field is reserved for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

WBID: A 5 bit field which is the wireless binding identifier. The identifier will indicate the type of wireless packet associated with the radio. The WBIDs defined in this specification can be found in Section 4.3

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.

Descriptor Sub-Element: The WTP Descriptor message element contains multiple Descriptor sub-elements, some of which are mandatory and some are optional, as described below. The Descriptor sub-element has the following format:



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

Descriptor Type: The Descriptor Type field identifies the data being encoded. The format of the data is vendor specific encoded in the UTF-8 format [RFC3629]. The CAPWAP protocol defines the following values, and each of these types identify whether their presence is mandatory or optional. The values listed below are used in conjunction with the Descriptor Vendor Identifier field, whose value MUST be set to zero (0). This field, combined with the Descriptor Vendor Identifier set to a non-zero (0) value, allows vendors to use a private namespace.

0 - Hardware Version: The WTP hardware version number MUST be present.

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

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

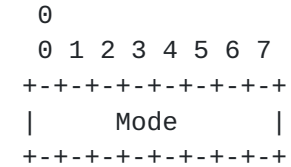
3 - Other Software Version: The WTP non-running software (firmware) version number MAY be present. This type is used to communicate alternate software versions that are available on the WTP's non-volatile storage.

Descriptor Length: Length of vendor specific encoding of Descriptor Data field, whose length MUST NOT exceed 1024 octets.

Descriptor Data: Vendor specific data of WTP information encoded in the UTF-8 format [RFC3629].

4.6.42. 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 4.8.9. The following enumerated values are supported:

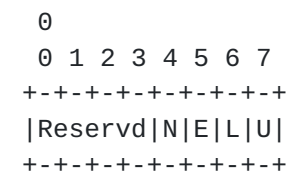
0 - Reserved

1 - Enabled

2 - Disabled

4.6.43. 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

Reservd: 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.

N: 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)

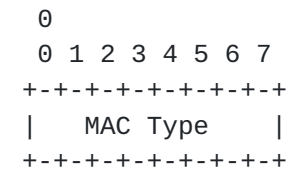
E: The 802.3 Frame Tunnel Mode requires the WTP and AC to encapsulate all user payload as native IEEE 802.3 frames (see Section 4.4). All user traffic is tunneled to the AC. This value MUST NOT be used when the WTP MAC Type is set to Split-MAC.

L: 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.

R: A reserved bit 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.

4.6.44. 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 enumerated values are supported:

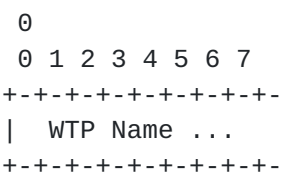
0 - Local-MAC: Local-MAC is the default mode that MUST be supported by all WTPs. When tunneling is enabled (see Section 4.6.43), the encapsulated frames MUST be in the 802.3 format (see Section 4.4.2), unless a wireless management or control frame which MAY be in its native format. Any CAPWAP binding needs to specify the format of management and control wireless frames.

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 [RFC3629]. The string is not zero terminated.



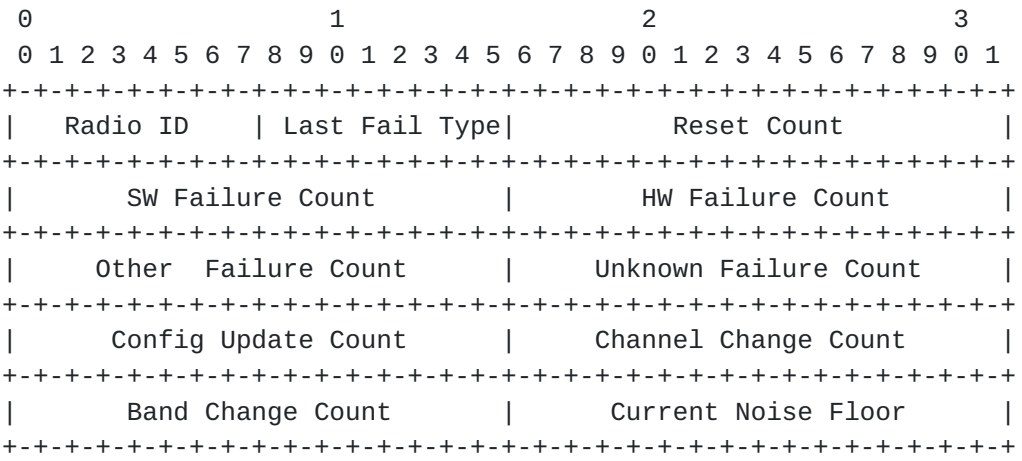
Type: 45 for WTP Name

Length: >= 1

WTP Name: A non-zero terminated UTF-8 encoded string [RFC3629] containing the WTP name, whose maximum size MUST NOT exceed 512 bytes.

4.6.46. 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. These counters are never reset on the WTP, and will therefore roll over to zero when the maximum size has been reached.



Type: 47 for WTP Radio Statistics

Length: 20

Radio ID: The radio ID of the radio to which the statistics apply, whose value is between one (1) and 31.

Last Failure Type: The last WTP failure. The following enumerated 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.

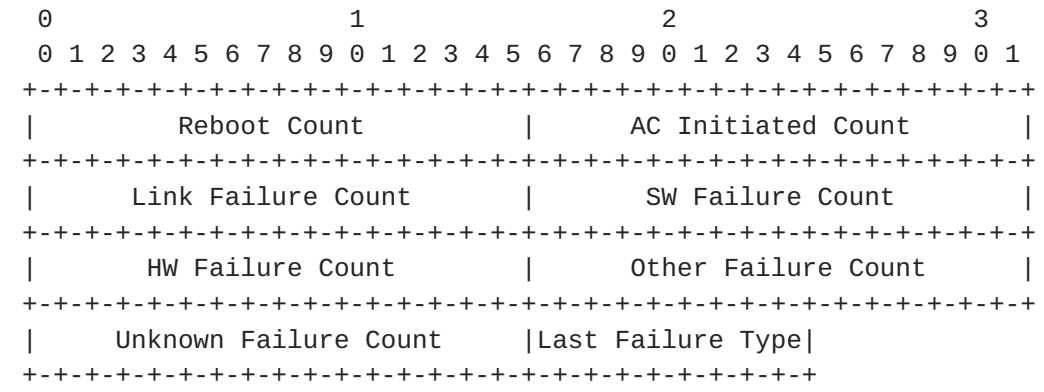
Channel Change Count: The number of times that the radio channel has been changed.

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.

4.6.47. 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. These counters are never reset on the WTP, and will therefore roll over to zero when the maximum size has been reached.



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 enumerated values are supported:

0 - Not Supported

1 - AC Initiated (see Section 9.2)

2 - Link Failure

3 - Software Failure

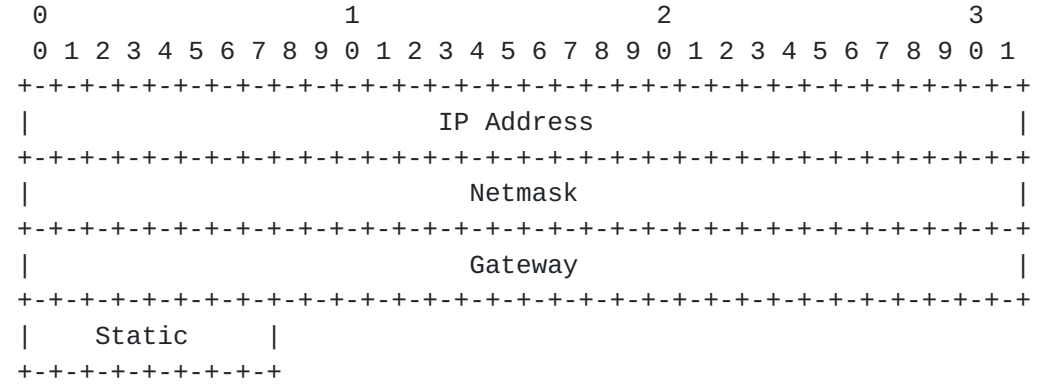
4 - Hardware Failure

5 - Other Failure

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

4.6.48. 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. IPv6 WTPs are expected to use dynamic addresses.



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.

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 definition of 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.

Default: 25 seconds

4.7.2. DataChannelKeepAlive

The DataChannelKeepAlive timer is used by the WTP to determine the next opportunity when it must transmit the Data Channel KeepAlive, in seconds.

Default: 30 seconds

4.7.3. 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: 60

4.7.4. DataCheckTimer

The number of seconds the AC will wait for the Data Channel Keep Alive, which is required by the CAPWAP state machine's Data Check state. The AC resets the state machine if this timer expires prior to transitioning to the next state.

Default: 30

4.7.5. DiscoveryInterval

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

Default: 5

4.7.6. DTLSSessionDelete

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

Default: 5

4.7.7. EchoInterval

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

Default: 30

4.7.8. IdleTimeout

The default Idle Timeout is 300 seconds.

4.7.9. ImageDataStartTimer

The number of seconds the WTP will wait for its peer to transmit the Image Data Request.

Default: 30

4.7.10. 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.

4.7.11. ReportInterval

The ReportInterval is used by the WTP to determine the interval the WTP uses between sending the Decryption Error message elements to inform the AC of decryption errors, in seconds.

The default Report Interval is 120 seconds.

4.7.12. RetransmitInterval

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

Default: 3

4.7.13. 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 seconds

4.7.14. StatisticsTimer

The StatisticsTimer is used by the WTP to determine the interval the WTP uses between the WTP Events Requests it transmits to the AC to communicate its statistics, in seconds.

Default: 120 seconds

4.7.15. 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

4.7.16. WaitJoin

The maximum time, in seconds, an AC will wait after the DTLS session has been established until it receives the Join Request from the WTP. This timer MUST be greater than 20 seconds.

Default: 60

4.8. CAPWAP Protocol Variables

This section defines the CAPWAP protocol variables, which are used for various protocol functions. Some of these variables are configurable, while others are counters or have a fixed value. For non counter related variables, default values are specified. However, when a WTP's variable configuration is explicitly overridden

by an AC, the WTP MUST save the new value.

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. MaxDiscoveries

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

Default: 10

4.8.6. MaxFailedDTLSSessionRetry

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

Default: 3.

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. RetransmitCount

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

4.8.9. 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 4.6.47.

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 4.6.43.

4.9.3. LastRebootReason

The reason why the WTP last rebooted, defined in Section 4.6.47.

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 4.6.47.

4.9.7. Static IP Address

The static IP Address assigned to the WTP, as configured by the WTP Static IP Address Information message element (see Section 4.6.48).

4.9.8. WTPLinkFailureCount

The number of times the link to the AC has failed, see Section 4.6.47.

4.9.9. WTPLocation

The WTP Location, defined in Section 4.6.30.

4.9.10. 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.

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. Once a Discovery Response has been received, if the WTP decides to establish a session with the responding AC, it SHOULD perform an MTU discovery, using the process described in Section 3.5.

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 4.6.21
- WTP Board Data, see Section 4.6.40
- WTP Descriptor, see Section 4.6.41
- WTP Frame Tunnel Mode, see Section 4.6.43
- 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).

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

- MTU Discovery Padding, see Section 4.6.32
- Vendor Specific Payload, see Section 4.6.39

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 4.6.9
  - CAPWAP Control IPv6 Address, see Section 4.6.10

The following message elements MAY be included in the Discovery Response message:

- Vendor Specific Payload, see Section 4.6.39

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 or to perform a Path MTU Discovery (see section Section 3.5.

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 4.6.21
- WTP Board Data, see Section 4.6.40
- WTP Descriptor, see Section 4.6.41
- WTP Frame Tunnel Mode, see Section 4.6.43
- 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).

The following message elements MAY be included in the Primary Discovery Request message:

- MTU Discovery Padding, see Section 4.6.32
- Vendor Specific Payload, see Section 4.6.39

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 4.6.9
- CAPWAP Control IPv6 Address, see Section 4.6.10

The following message elements MAY be included in the Primary Discovery Response message:

- Vendor Specific Payload, see Section 4.6.39

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 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. If the WTP is performing the optional AC discovery process (see Section 3.3), the join process occurs after the WTP has received one or more Discovery Response messages. During the discovery process, an AC MAY return more than one CAPWAP Control IPv4 Address or CAPWAP Control IPv6 Address message elements. When more than one such message element is returned, the WTP SHOULD perform "load balancing" by choosing the interface that is servicing the least number of WTPs (known through the WTP Count field of the message element). Note, however, that other load balancing algorithms are also permitted. Once the WTP has determined its preferred AC, and its associated interface, to connect to, it establishes the DTLS session, and transmits the Join Request over the secured control channel. 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 4.6.30
- WTP Board Data, see Section 4.6.40
- WTP Descriptor, see Section 4.6.41
- WTP Name, see Section 4.6.45
- Session ID, see Section 4.6.37
- WTP Frame Tunnel Mode, see Section 4.6.43
- 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).
- ECN Support, see Section 4.6.25

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

- CAPWAP Local IPv4 Address, see Section 4.6.11
- CAPWAP Local IPv6 Address, see Section 4.6.12

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

- CAPWAP Transport Protocol, see Section 4.6.14
- Maximum Message Length, see Section 4.6.31
- WTP Reboot Statistics, see Section 4.6.47
- Vendor Specific Payload, see Section 4.6.39

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 MUST be included in the Join Response message.

- Result Code, see Section 4.6.35
- 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).
- ECN Support, see Section 4.6.25

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

- CAPWAP Control IPv4 Address, see Section 4.6.9
- CAPWAP Control IPv6 Address, see Section 4.6.10

One of the following message elements MUST be included in the Join

Response Message:

- CAPWAP Local IPv4 Address, see Section 4.6.11
- CAPWAP Local IPv6 Address, see Section 4.6.12

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
- CAPWAP Transport Protocol, see Section 4.6.14
- Image Identifier, see Section 4.6.27
- Maximum Message Length, see Section 4.6.31
- Vendor Specific Payload, see Section 4.6.39

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

The following message elements MAY be included in the Echo Request message:

o Vendor Specific Payload, see Section 4.6.39

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 Echo Request message. After transmitting the Echo Response message, the AC SHOULD reset its EchoInterval timer (see Section 4.7.7. 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 following message elements MAY be included in the Echo Response message:

- Vendor Specific Payload, see Section 4.6.39

When a WTP receives an Echo Response message it initializes the EchoInterval to the configured value.

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 can behave in one of two ways, which is implementation specific:

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

2. The WTP saves the configuration of parameters provided by the AC that are non-default values into local non-volatile memory, and are enforced during the WTP's power up initialization phase.

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 Request 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 Request

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

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

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

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

The Configuration Status Request 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 Request message.

- AC Name, see Section 4.6.4
- Radio Administrative State, see Section 4.6.33
- Statistics Timer, see Section 4.6.38
- WTP Reboot Statistics, see Section 4.6.47

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

- AC Name with Priority, see Section 4.6.5
- CAPWAP Transport Protocol, see Section 4.6.14
- WTP Static IP Address Information, see Section 4.6.48
- Vendor Specific Payload, see Section 4.6.39

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 Status 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.

- CAPWAP Timers, see Section 4.6.13
- Decryption Error Report Period, see Section 4.6.18
- Idle Timeout, see Section 4.6.24
- WTP Fallback, see Section 4.6.42

One or both of 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

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

- WTP Static IP Address Information, see Section 4.6.48
- Vendor Specific Payload, see Section 4.6.39

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 message element (see Section 4.6.27) 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 by transmitting an Image Data Request (see Section 9.1.1) that includes the Initiate Download message element (see Section 4.6.29).

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 Priority, see Section 4.6.5
- AC Timestamp, see Section 4.6.6
- Add MAC ACL Entry, see Section 4.6.7
- CAPWAP Timers, see Section 4.6.13
- Decryption Error Report Period, see Section 4.6.18
- Delete MAC ACL Entry, see Section 4.6.19
- Idle Timeout, see Section 4.6.24

- Location Data, see Section 4.6.30
- Radio Administrative State, see Section 4.6.33
- Statistics Timer, see Section 4.6.38
- WTP Fallback, see Section 4.6.42
- WTP Name, see Section 4.6.45
- WTP Static IP Address Information, see Section 4.6.48
- Image Identifier, see Section 4.6.27
- Vendor Specific Payload, see Section 4.6.39

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 4.6.35

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

- Radio Operational State, see Section 4.6.34
- Vendor Specific Payload, see Section 4.6.39

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 specific configuration request, it indicates the failure by including one or more Returned Message Element message elements (see Section 4.6.36).

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 4.6.34

- Result Code, see Section 4.6.35

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

- Returned Message Element(s), see Section 4.6.36

- Vendor Specific Payload, see Section 4.6.39

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 following message elements MAY be included in the Change State Event Response message:

o Vendor Specific Payload, see Section 4.6.39

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 following message elements MAY be included in the Clear Configuration Request message:

o Vendor Specific Payload, see Section 4.6.39

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 Response message MUST include the following message element.

- Result Code, see Section 4.6.35

The following message elements MAY be included in the Clear Configuration Request message:

- Vendor Specific Payload, see Section 4.6.39

9. Device Management Operations

This section defines CAPWAP operations responsible for debugging, gathering statistics, logging, and firmware management. The management operations defined in this section are used by the AC to either push/pull information to/from the WTP, or request that the WTP reboot. This section does not deal with the management of the AC per se, and assumes that the AC is operational and configured.

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. The former allows the download to occur at boot time, while the latter is used to trigger the download while an active CAPWAP session exists. It is important to note that the CAPWAP protocol does not provide the ability for the AC to identify whether the firmware information provided by the WTP is correct, nor whether the WTP is properly storing the firmware (see Section 12.10 for more information.

Figure 6 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.

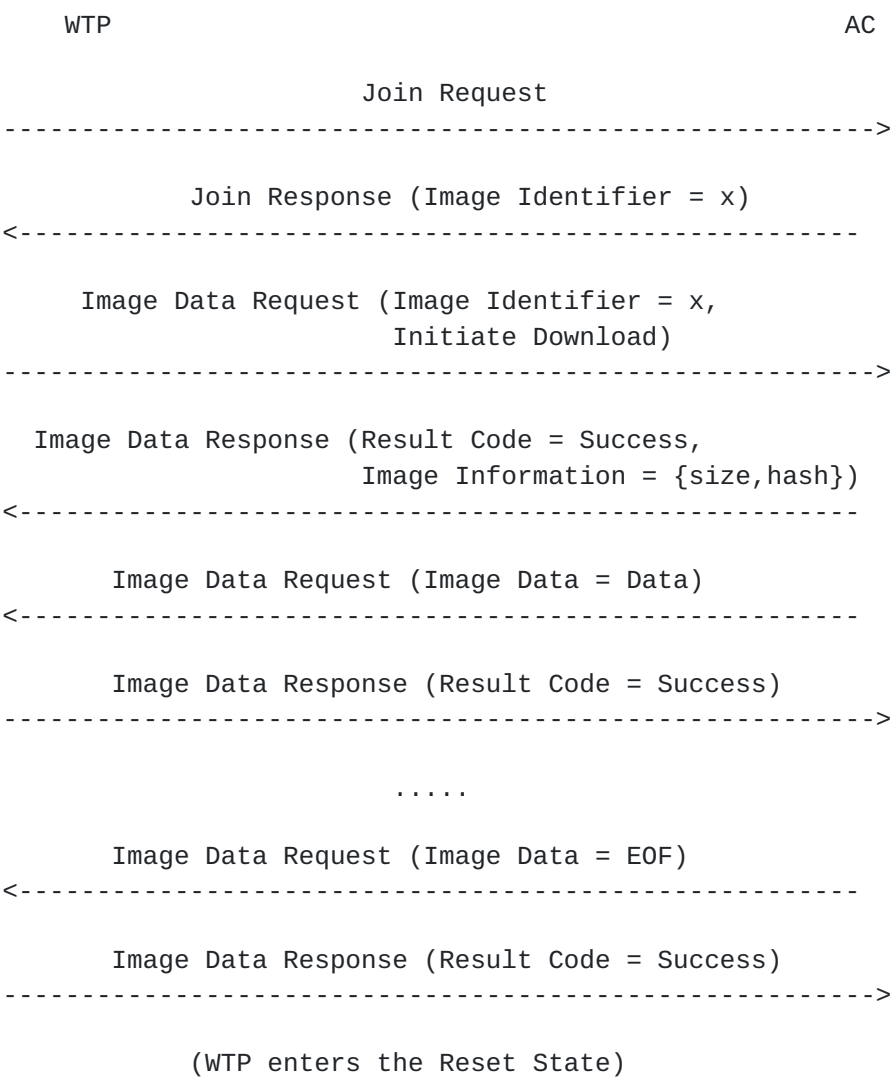


Figure 6: WTP Firmware Download Case 1

Figure 7 provides an example in which the WTP has the image specified by the AC in its non-volative storage, but is not its current running image. In this case, the WTP opts to NOT download the firmware and immediately reset to the requested image.

Figure 7: WTP Firmware Download Case 2

Figure 8 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 8: WTP Firmware Download Case 3

Figure 9 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.

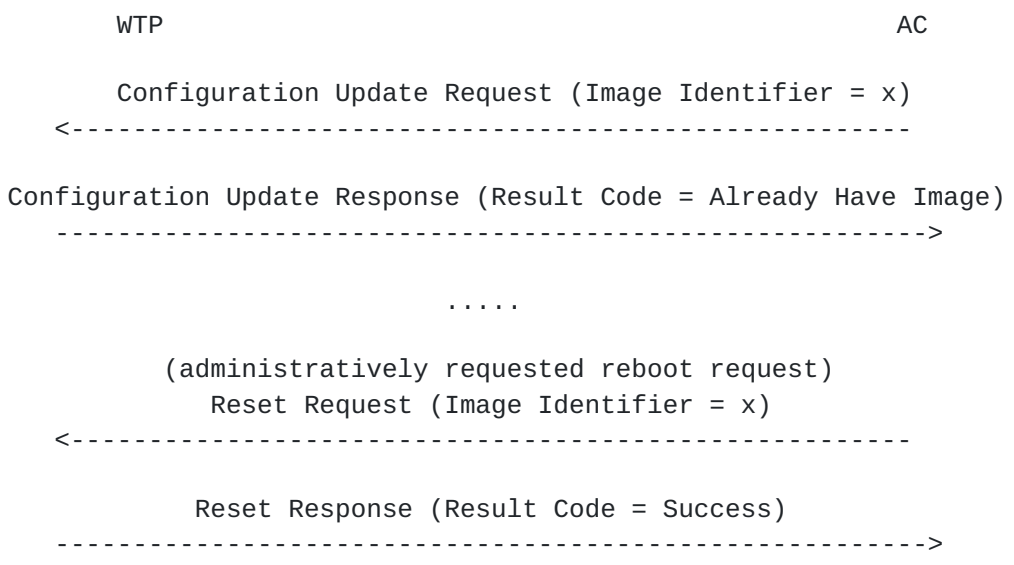


Figure 9: WTP Firmware Download Case 4

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 in its non-volatile flash, but is not its currently running image, it simply resets to run the proper firmware.

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 an Configuration Update Request message with the Image Identifier message elements. This will cause the WTP to transmit an Image

Data Request with the Image Identifier and the Initiate Download message elements. 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 transferred.

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.

- CAPWAP Transport Protocol, see Section 4.6.14
- Image Data, see Section 4.6.26
- Vendor Specific Payload, see Section 4.6.39

The following message elements MAY be included in the Image Data Request message when sent by the WTP.

- Image Identifier, see Section 4.6.27
- Initiate Download, see Section 4.6.29

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.

o Result Code, see Section 4.6.35

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

o Vendor Specific Payload, see Section 4.6.39

The following message elements MAY be included in the Image Data Response message when sent by the AC.

o Image Information, see Section 4.6.28

Upon receiving an Image Data Response message indicating an error, the WTP MAY retransmit a previous Image Data Request 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. If the AC includes the Image Identifier message element (see Section 4.6.27), it indicates to the WTP that it SHOULD use that version of software upon reboot.

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.

o Image Identifier, see Section 4.6.27

The following message elements MAY be included in the Reset Request message:

o Vendor Specific Payload, see Section 4.6.39

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 Reset Response message.

- Result Code, see Section 4.6.35
- Vendor Specific Payload, see Section 4.6.39

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 4.6.24), due 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 message element

listed MAY be included in the WTP Event Request message.

- Decryption Error Report, see Section 4.6.17
- Duplicate IPv4 Address, see Section 4.6.22
- Duplicate IPv6 Address, see Section 4.6.23
- WTP Radio Statistics, see Section 4.6.46
- WTP Reboot Statistics, see Section 4.6.47
- Delete Station, see Section 4.6.20
- Vendor Specific Payload, see Section 4.6.39

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 following message elements MAY be included in the WTP Event Response message:

- Vendor Specific Payload, see Section 4.6.39

9.6. Data Transfer

This section describes the data transfer procedures used by the CAPWAP protocol. The data transfer mechanism is used to upload information available at the WTP to the AC, such as crash or debug information. The data transfer messages can only be exchanged while in the Run state.

Figure 10 provides an example of an AC that requests that the WTP transfer its latest crash file. Once the WTP acknowledges that it has information to send, via the Data Transfer Response, it transmits its own Data Transfer Request. Upon receipt, the AC responds with a Data Transfer Response, and the exchange continues until the WTP transmits a Data Transfer Data message element that indicates an End of File (EOF).

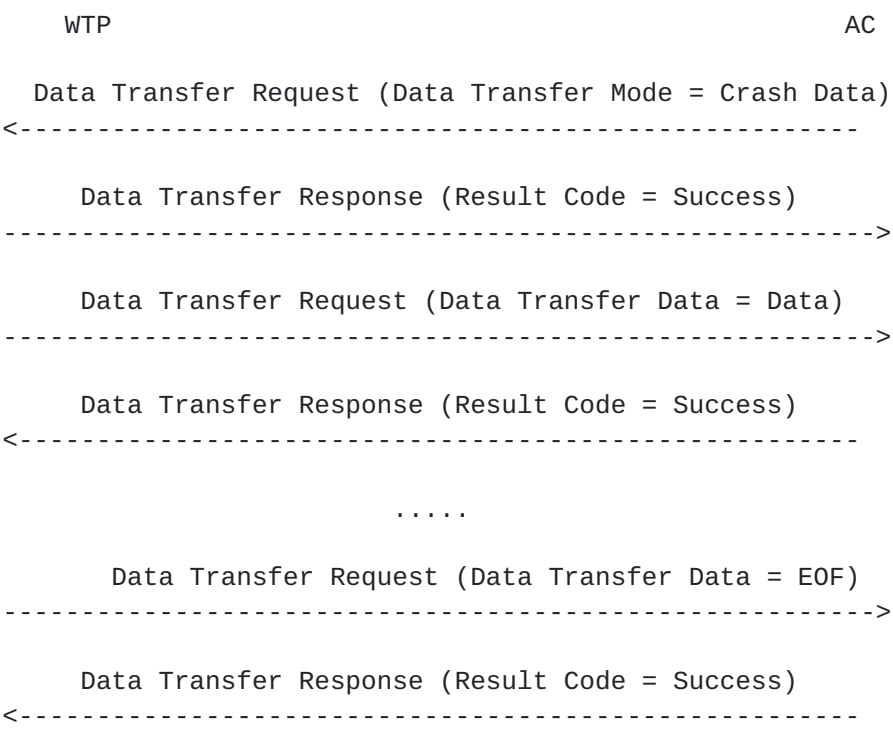


Figure 10: WTP Data Transfer Case 1

Figure 11 provides an example of an AC that requests that the WTP transfer its latest crash file. However, in this example, the WTP does not have any crash information to send, and therefore sends a Data Transfer Response with a Result Code indicating the error.

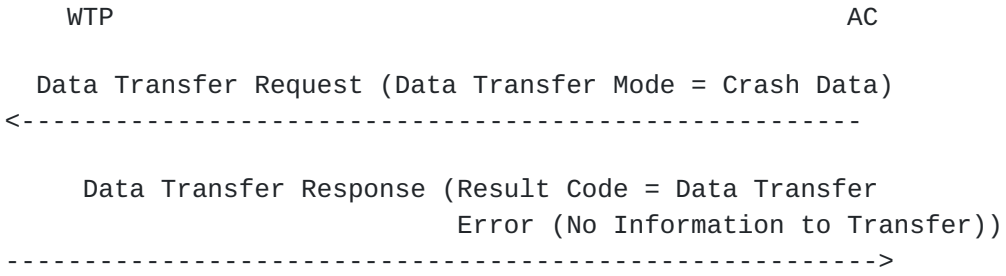


Figure 11: WTP Data Transfer Case 2

9.6.1. Data Transfer Request

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

The Data Transfer Request messages can be sent either by the AC or

the WTP. When sent by the AC, it is used to request that data be transmitted from the WTP to the AC, and includes the Data Transfer Mode message element, which specifies the information desired by the AC. The Data Transfer Request is sent by the WTP in order to transfer actual data to the AC, through the Data Transfer Data message element.

Given that the CAPWAP protocol minimizes the need for WTPs to be directly managed, the Data Transfer Request is an important troubleshooting tool used by the AC to retrieve information that may be available on the WTP. For instance, some WTPs implementations may store crash information to help manufacturers identify software faults. The Data Transfer Request message can be used to send such information from the WTP to the AC. Another possible use would be to allow a remote debugger function in the WTP to use the Data Transfer Request message to send console output to the AC for debugging purposes.

When the WTP or 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 through the Data Transfer Data message element.

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

When sent by the AC, the Data Transfer Request message MUST contain the following message elements:

o Data Transfer Mode, see Section 4.6.16

When sent by the WTP, the Data Transfer Request message MUST contain the following message elements:

o Data Transfer Data, see Section 4.6.15

Regardless of whether the Data Transfer Request is sent by the AC or WTP, the following message elements MAY be included in the Data Transfer Request message:

o Vendor Specific Payload, see Section 4.6.39

9.6.2. 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. When sent by the WTP, the Result Code message element is used to indicate whether the data transfer requested by the AC can be completed. When sent by the AC, the Result Code message element is used to indicate receipt of the data transfered in the Data Transfer Request message.

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

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

- Result Code, see Section 4.6.35

The following message elements MAY be included in the Data Transfer Response message:

- Vendor Specific Payload, see Section 4.6.39

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 4.6.20
- Vendor Specific Payload, see Section 4.6.39

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 4.6.35

The following message elements MAY be included in the Station Configuration Response message:

- Vendor Specific Payload, see Section 4.6.39

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 therefore cannot use the WTP's IP address alone to bind the CAPWAP control and data channel. The CAPWAP Data Check state, which establishes the data plane connection and communicates the CAPWAP Data Channel 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). CAPWAP implementations MUST also use DTLS session information on any encrypted CAPWAP channel to validate the source of both the control and data plane, as described in Section 12.2.

In the third configuration, the AC is deployed behind a NAT. In this case, the AC is not reachable by the WTP unless a specific rule has been configured on the NAT to translate the address and redirect CAPWAP messages to the AC. This deployment presents two issues. First, an AC communicates its interfaces and corresponding WTP load using the CAPWAP Control IPv4 Address and CAPWAP Control IPv6 Address message elements. This message element is mandatory, but contains IP addresses that are only valid in the private address space used by the AC, which is not reachable by the WTP. The WTP MUST NOT utilize the information in these message elements if it detects a NAT (as described in the CAPWAP Transport Protocol message element in Section 4.6.14). Second, since the addresses cannot be used by the WTP, this effectively disables the load balancing capabilities (see Section 6.1) of the CAPWAP protocol. Alternatively, the AC could have a configured NAT'ed address, which it would include in either of the two control address message elements, and the NAT would need to be configured accordingly.

In order for a CAPWAP WTP or AC to detect whether a middlebox is present, both the Join Request (see Section 6.1) and the Join Response (see Section 6.2) include either the CAPWAP Local IPv4 Address (see Section 4.6.11), or the CAPWAP Local IPv6 Address (see Section 4.6.12) message element. Upon receiving one of these messages, if the packet's source IP address differs from the address found in either one of these message elements, it indicates that a

middlebox is present.

In order for CAPWAP to be compatible with potential middleboxes in the network, CAPWAP implementations MUST send return traffic from the same port on which it received traffic from a given peer. Further, any unsolicited requests generated by a CAPWAP node MUST be sent on the same port.

Note that this middlebox detection technique is not foolproof. If the public IP address assigned to the NAT is identical to the private IP address used by the AC, detection by the WTP would fail. This failure can lead to various protocol errors, so it is therefore necessary for deployments to ensure that the NAT's IP address is not the same as the ACs.

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 ignored by the WTP when it detects the AC is behind a middlebox.

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.

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:

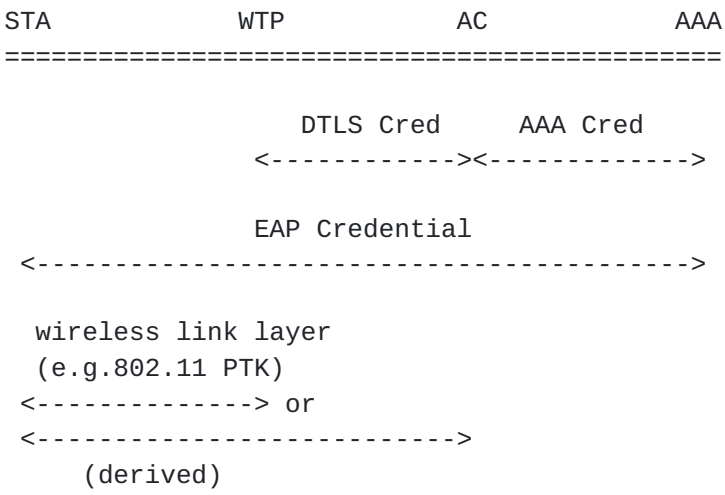


Figure 12: STA Session Setup

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.

12.1.5. Use of MD5

The Image Information Section 4.6.28) message element makes use of MD5 to compute the hash field. The authenticity and integrity of the image file is protected by DTLS, and in this context, MD5 is not used as a cryptographically secure hash, but just as a basic checksum. Therefore, the use of MD5 is not considered a security vulnerability, and no mechanisms for algorithm agility are provided.

12.1.6. CAPWAP Fragmentation

RFC 4963 [RFC4963] describes a possible security vulnerability where a malicious entity can "corrupt" a flow by injecting fragments. By sending "high" fragments (those with offset greater than zero) with a

forged source address, the attacker can deliberately cause corruption. The use of DTLS on the CAPWAP Data channel can be used to avoid this possible vulnerability.

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 128-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 or DTLS Setup Attacks

Since the Discovery Request messages are sent in the clear, it is important that AC implementations NOT assume that receiving a Discovery Request message from a WTP implies that the WTP 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.

Similarly, entering the DTLS Setup phase SHOULD NOT assume that the WTP has reset, and therefore should not discard active state until the DTLS session has been successfully established. While the HelloVerifyRequest provides some protection against denial of service

(DoS) attacks on the AC, an adversary capable of receiving packets at a valid address (or a malfunctioning or misconfigured WTP) may repeatedly attempt DTLS handshakes with the AC, potentially creating a resource shortage. If either the FailedDTLSSessionCount or the FailedDTLSAuthFailCount counter reaches the value of MaxFailedDTLSSessionRetry variable (see Section 4.8), implementations MAY choose to rate-limit new DTLS handshakes for some period of time. It is RECOMMENDED that implementations choosing to implement rate limiting use a random discard technique, rather than mimicking the WTP's sulking behavior. This will ensure that messages from valid WTPs will have some probability of eliciting a response, even in the face of a significant DoS attack.

Some CAPWAP implementations may wish to restrict the DTLS setup process to only those peers that have been configured in the access control list, authorizing only those clients to initiate a DTLS handshake. Note that the impact of this on mitigating denial of service attacks against the DTLS layer is minimal, because DTLS already uses client-side cookies to minimize processor consumption attacks.

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. CAPWAP Pre-Provisioning

In order for CAPWAP to establish a secure communication with a peer, some level of pre-provisioning on both the WTP and AC is necessary. This section will detail the minimal number of configuration parameters.

When using preshared keys, it is necessary to configure the preshared

key for each possible peer with which a DTLS session may be established. To support this mode of operation, one or more entries of the following table may be configured on either the AC or WTP:

- Identity: The identity of the peering AC or WTP. This format MAY be either in the form of an IP address or host name (the latter of which needs to be resolved to an IP address using DNS).

- Key: The pre-shared key for use with the peer when establishing the DTLS session (see Section 12.6 for more information).

- PSK Identity: Identity hint associated with the provisioned key (see Section 2.4.4.4 for more information).

When using certificates, the following items need to be pre-provisioned:

- Device Certificate: The local device's certificate (see Section 12.7 for more information)

- Trust Anchor: Trusted root certificate chain used to validate any certificate received from CAPWAP peers. Note that one or more root certificate MAY be configured on a given device.

Regardless of the authentication method, the following items need to be pre-provisioned:

- Access Control List: The access control list table contains the identities of one or more CAPWAP peers, along with a rule. The rule is used to determine whether communication with the peer is permitted (see Section 2.4.4.3 for more information).

12.6. 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 [RFC4086] 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 unique PSK. This prevents the domino effect (see Guidance for AAA Key Management [RFC4962]). 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.

12.7. 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 peer certificate itself. 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.

12.8. Use of MAC Address in CN Field

The CAPWAP protocol is an evolution of an existing protocol [I-D.ohara-capwap-lwapp] which is implemented on a large number of already deployed ACs and WTPs. Everyone of these devices have an existing X.509 certificate, which is provisioned at manufacturing time. These X.509 certificates use the device's MAC Address in the Common Name (CN) field. It is well understood that encoding the MAC Address in the CN field is less than optimal, and using the SubjectAltName field would be preferable. However, at the time of publication, there is no URN specification that allows for the MAC Address to be used in the SubjectAltName field. As such a specification is published by the IETF, future versions of the CAPWAP protocol MAY require support for the new URN scheme.

12.9. 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 [RFC3539] 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.

12.10. WTP Firmware

The CAPWAP protocol defines a mechanism by which the AC downloads new firmware to the WTP. During the session establishment process, the WTP provides information about its current firmware to the AC. The AC then decides whether the WTP's firmware needs to be updated. It is important to note that the CAPWAP specification makes the explicit assumption that the WTP is providing the correct firmware version to the AC, and is therefore not lying. Fruther, during the firmware download process, the CAPWAP protocol does not provide any mechanisms to recognize whether the WTP is actually storing the firmware for future use.

13. Operational 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.

The CAPWAP protocol does not deal with the management of the ACs. The AC is assumed to be configured through some separate management interface, which could be via a proprietary CLI, SNMP, NETCONF or some other management protocol.

The CAPWAP protocol's control channel is fairly light weight from a traffic perspective. Once the WTP has been configured, the WTP sends periodic statistics. Further, the specification calls for a keepalive packet to be sent on the protocol's data channel to make sure that any possible middleboxes (e.g., NAT) maintain their UDP state. The overhead associated with the control and data channel is not expected to impact network traffic. That said, the CAPWAP protocol does allow for the frequency of these packets to be modified through the DataChannelKeepAlive and StatisticsTimer (see Section 4.7.2 and Section 4.7.14, respectively).

14. Transport Considerations

The CAPWAP WG carefully considered the congestion control requirements of the CAPWAP protocol, both for the CAPWAP control and data channels.

CAPWAP specifies a single-threaded command/response protocol to be used on the control channel, and we have specified that an exponential back-off algorithm should be used when commands are retransmitted. When CAPWAP runs in its default mode (Local MAC), the control channel is the only CAPWAP channel.

However, CAPWAP can also be run in Split MAC mode, in which case there will be a DTLS-encrypted data channel between each WTP and the AC. The WG discussed various options for providing congestion control on this channel. However, due to performance problems with TCP when it is run over another congestion control mechanism and the fact that the vast majority of traffic run over the CAPWAP data channel is likely to be congestion-controlled IP traffic, the CAPWAP WG felt that specifying a congestion control mechanism for the CAPWAP data channel would be more likely to cause problems than to resolve any.

Because there is no congestion control mechanism specified for the CAPWAP data channel, it is RECOMMENDED that non-congestion-controlled traffic not be tunneled over CAPWAP. When a significant amount of non-congestion-controlled traffic is expected to be present on a WLAN, the CAPWAP connection between the AC and the WTP for that LAN should be configured to remain in Local MAC mode with Distribution function at the WTP.

The lock step nature of the CAPWAP protocol's control channel can cause the firmware download process to take some time, depending upon the RTT. This is not expected to be a problem since the CAPWAP protocol allows firmware to be downloaded while the WTP provides service to wireless clients/devices.

It is necessary for the WTP and AC to configure their MTU based on the capabilities of the path. See Section 3.5 for more information.

The CAPWAP protocol mandates support of the Explicit Congestion Notification (ECN) through a mode of operation named "limited functionality option", detailed in section 9.1.1 of [RFC3168]. Future versions of the CAPWAP protocol should consider mandating support for the "full functionality option".

15. IANA Considerations

This section details the actions to be taken by IANA during the publication of the specification. There are numerous registries that need to be created, and the contents, document action (see [RFC5226], and registry format are all included below. Note that in cases where bit fields are referred to, the bit numbering is left to right, where the leftmost bit is labelled as bit zero (0).

For registration requests where an Expert Review is required, a Designated Expert should be consulted, which is appointed by the responsible IESG area director. The intention is that any allocation will be accompanied by a published RFC, but given that other SDOs may want to create standards built on top of CAPWAP, a document the Designated Expert can review is also acceptable. IANA should allow for allocation of values prior to documents being approved for publication, so the Designated Expert can approve allocations once it seems clear that publication will occur. The Designated expert will post a request to the CAPWAP WG mailing list (or a successor designated by the Area Director) for comment and review. Before a period of 30 days has passed, the Designated Expert will either approve or deny the registration request and publish a notice of the decision to the CAPWAP WG mailing list or its successor, as well as informing IANA. A denial notice must be justified by an explanation, and in the cases where it is possible, concrete suggestions on how the request can be modified so as to become acceptable should be provided.

15.1. IPv4 Multicast Address

This document requires a new IPv4 multicast address called "capwap-ac" from the Local Network Control Block IPv4 multicast address registry [to be removed upon publication http://www.iana.org/assignments/multicast-addresses]. The new multicast address is to replace the text xx.xx.xx.xx in Section 3.3.

15.2. IPv6 Multicast Address

This document requires a new organization local multicast address called the "All ACs multicast address" from the Variable Scope IPv6 multicast address registry [to be removed upon publication http://www.iana.org/assignments/ipv6-multicast-addresses]. The new multicast address is to be inserted in Section 3.3.

15.3. UDP Port

This document requires a two UDP Ports organization local multicast address from the registered port numbers registry [to be removed upon

publication http://www.iana.org/assignments/port-numbers]. The new UDP Ports numbers have already been assigned and can be found in Section 3.1. The following values are being registered:

Keyword	Decimal	Description	References
capwap-control	5246/udp	CAPWAP Control Protocol	This Document
capwap-data	5247/udp	CAPWAP Data Protocol	This Document

15.4. CAPWAP Message Types

The Message Type field in the CAPWAP header (see 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 [RFC5226].

IANA will create and maintain the CAPWAP Message Types registry for all message types whose Enterprise Number is set to zero (0). The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) through 26 are allocated in this specification, and can be found in Section 4.5.1.1. Any new assignments of a CAPWAP Message Type, whose Enterprise Number is set to zero (0) requires a Expert Review. The format of the registry to be maintained by IANA has the following format:

CAPWAP Control Message	Message Type Value	Reference

15.5. CAPWAP Header Flags

The Flags field in the CAPWAP header (see Section 4.3) is 9 bits in length and is used to identify any special treatment related to the message. This specification defines bits zero (0) through five (5), while bits six (6) through eight (8) are reserved. There are currently three unused, reserved bits which are managed by IANA and whose assignment requires a Expert Review. IANA will create the CAPWAP Header Flags registry, whose format is:

Flag Field Name	Bit Position	Reference

15.6. CAPWAP Control Message Flags

The Flags field in the CAPWAP Control Message header (see Section 4.5.1.4) is used to identify any special treatment related to the control message. There are currently eight (8) unused, reserved bits. These bits whose assignment is managed by IANA and requires a Expert Review. IANA will create the CAPWAP Control Message Flags

registry, whose format is:

Flag Field Name	Bit Position	Reference

15.7. CAPWAP Message Element Type

The Type field in the CAPWAP Message Element header (see Section 4.6) is used to identify the data being transported. The namespace is 16 bits (0-65535), where the value of zero (0) is reserved and must not be assigned. The values one (1) through 53 are allocated in this specification, and can be found in Section 4.5.1.1.

The 16 bit namespace is further divided into blocks of addresses that are reserved for specific CAPWAP wireless bindings. The following blocks are reserved:

CAPWAP Protocol Message Elements	1 - 1023
IEEE 802.11 Message Elements	1024 - 2047
EPCGlobal Message Elements	3072 - 4095

This namespace is managed by IANA and assignments require a Expert Review. IANA will create the CAPWAP Message Element Type registry, whose format is:

CAPWAP Message Element	Type Value	Reference

15.8. Wireless Binding Identifiers

The Wireless Binding Identifier (WBID) field in the CAPWAP header (see Section 4.3) is used to identify the wireless technology associated with the packet. This specification allocates the values one (1) and three (3). Due to the limited address space available, a new WBID request requires Expert Review. IANA will create the CAPWAP Wireless Binding Identifier registry, whose format is:

CAPWAP Wireless Binding Identifier	Type Value	Reference

15.9. AC Security Types

The Security field in the AC Descriptor message element (see Section 4.6.1) is 8 bits in length and used to identify the authentication methods available on the AC. This specification defines bits five (5) and six (6), while bits zero (0) through four (4) as well as bit seven (7) are reserved and unused. These reserved bits are managed by IANA and assignment requires a Standards Action. IANA will create the AC Security Types registry, whose format is:

AC Security Type	Bit Position	Reference

15.10. AC DTLS Policy

The DTLS Policy field in the AC Descriptor message element (see Section 4.6.1) is 8 bits in length and used to identify whether the CAPWAP Data Channel is to be secured. This specification defines bits five (5) and six (6), while bits zero (0) through four (4) as well as bit seven (7) are reserved and unused. These reserved bits are managed by IANA and assignment requires a Standards Action. IANA will create the AC DTLS Policy registry, whose format is:

AC DTLS Policy	Bit Position	Reference

15.11. AC Information Type

The Information Type field in the AC Descriptor message element (see Section 4.6.1) is used to represent information about the AC. The namespace is 16 bits (0-65535), where the value of zero (0) is reserved and must not be assigned. This field, combined with the AC Information Vendor ID, allows vendors to use a private namespace. This specification defines the AC Information Type namespace when the AC Information Vendor ID is set to zero (0), for which the values four (4) and five (5) are allocated in this specification, and can be found in Section 4.6.1. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the AC Information Type registry, whose format is:

AC Information Type	Type Value	Reference

15.12. CAPWAP Transport Protocol Types

The Transport field in the CAPWAP Transport Protocol message element (see Section 4.6.14) is used to identify the transport to use for the CAPWAP Data Channel. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) and two (2) are allocated in this specification, and can be found in Section 4.6.14. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the CAPWAP Transport Protocol Types registry, whose format is:

CAPWAP Transport Protocol Type	Type Value	Reference

15.13. Data Transfer Type

The Data Type field in the Data Transfer Data message element (see Section 4.6.15) and Image Data message element (see Section 4.6.26) is used to provide information about the data being carried. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1), two (2) and five (5)

are allocated in this specification, and can be found in Section 4.6.15. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Data Transfer Type registry, whose format is:

Data Transfer Type	Type Value	Reference

15.14. Data Transfer Mode

The Data Mode field in the Data Transfer Data message element (see Section 4.6.15) and Data Transfer Mode message element (see Section 15.14) is used to provide information about the data being carried. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) and two (2) are allocated in this specification, and can be found in Section 15.14. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Data Transfer Mode registry, whose format is:

Data Transfer Mode	Type Value	Reference

15.15. Discovery Types

The Discovery Type field in the Discovery Type message element (see Section 4.6.21) is used by the WTP to indicate to the AC how it was discovered. The namespace is 8 bits (0-255). The values zero (0) through four (4) are allocated in this specification, and can be found in Section 4.6.21. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Discovery Types registry, whose format is:

Discovery Types	Type Value	Reference

15.16. ECN Support

The ECN Support field in the ECN Support message element (see Section 4.6.25) is used by the WTP to represent its ECN Support. The namespace is 8 bits (0-255). The values zero (0) and one (1) are allocated in this specification, and can be found in Section 4.6.25. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the ECN Support registry, whose format is:

ECN Support	Type Value	Reference

15.17. Radio Admin State

The Radio Admin field in the Radio Administrative State message element (see Section 4.6.33) is used by the WTP to represent the state of its radios. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) and two (2) are allocated in this specification, and can be found in Section 4.6.33. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Radio Admin State registry, whose format is:

Radio Admin State	Type Value	Reference

15.18. Radio Operational State

The State field in the Radio Operational State message element (see Section 4.6.34) is used by the WTP to represent the operational state of its radios. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) and two (2) are allocated in this specification, and can be found in Section 4.6.34. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Radio Operational State registry, whose format is:

Radio Operational State	Type Value	Reference

15.19. Radio Failure Causes

The Cause field in the Radio Operational State message element (see Section 4.6.34) is used by the WTP to represent the reason why a radio may have failed. The namespace is 8 bits (0-255), where the value of zero (0) through three (3) are allocated in this specification, and can be found in Section 4.6.34. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Radio Failure Causes registry, whose format is:

Radio Failure Causes	Type Value	Reference

15.20. Result Code

The Result Code field in the Result Code message element (see Section 4.6.35) is used to indicate the success, or failure, of a CAPWAP control message. The namespace is 32 bits (0-4294967295), where the value of zero (0) through 22 are allocated in this specification, and can be found in Section 4.6.35. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Result Code registry, whose format is:

Result Code	Type Value	Reference

15.21. Returned Message Element Reason

The Reason field in the Returned Message Element message element (see Section 4.6.36) is used to indicate the reason why a message element was not processed successfully. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be assigned. The values one (1) through four (4) are allocated in this specification, and can be found in Section 4.6.36. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the Returned Message Element Reason registry, whose format is:

Returned Message Element Reason	Type Value	Reference

15.22. WTP Board Data Type

The Board Data Type field in the WTP Board Data message element (see Section 4.6.40) is used to represent information about the WTP hardware. The namespace is 16 bits (0-65535). The WTP Board Data Type values zero (0) through four (4) are allocated in this specification, and can be found in Section 4.6.40. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP Board Data Type registry, whose format is:

WTP Board Data Type	Type Value	Reference

15.23. WTP Descriptor Type

The Descriptor Type field in the WTP Descriptor message element (see Section 4.6.41) is used to represent information about the WTP software. The namespace is 16 bits (0-65535). This field, combined with the Descriptor Vendor ID, allows vendors to use a private namespace. This specification defines the WTP Descriptor Type namespace when the Descriptor Vendor ID is set to zero (0), for which the values zero (0) through three (3) are allocated in this specification, and can be found in Section 4.6.41. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP Board Data Type registry, whose format is:

WTP Descriptor Type	Type Value	Reference

15.24. WTP Fallback Mode

The Mode field in the WTP Fallback message element (see Section 4.6.42) is used to indicate to the WTP the type of AC fallback mechanism it should employ. The namespace is 8 bits (0-255), where the value of zero (0) is reserved and must not be

assigned. The values one (1) and two (2) are allocated in this specification, and can be found in Section 4.6.42. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP Fallback Mode registry, whose format is:

WTP Fallback Mode	Type Value	Reference

15.25. WTP Frame Tunnel Mode

The Tunnel Type field in the WTP Frame Tunnel Mode message element (see Section 4.6.43) is 8 bits and is used to indicate the type of tunneling to use between the WTP and the AC. This specification defines bits four (4) through six (6), while bits zero (0) through four (4) as well as bit seven (7) are reserved and unused. These reserved bits are managed by IANA and assignment requires a Expert Review. IANA will create the AC DTLS Policy registry, whose format is:

WTP Frame Tunnel Mode	Bit Position	Reference

15.26. WTP MAC Type

The MAC Type field in the WTP MAC Type message element (see Section 4.6.44) is used to indicate the type of MAC to use in tunneled frames between the WTP and the AC. The namespace is 8 bits (0-255), where the value of zero (0) through two (2) are allocated in this specification, and can be found in Section 4.6.44. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP MAC Type registry, whose format is:

WTP MAC Type	Type Value	Reference

15.27. WTP Radio Stats Failure Type

The Last Failure Type field in the WTP Radio Statistics message element (see Section 4.6.46) is used to indicate the last WTP failure. The namespace is 8 bits (0-255), where the value of zero (0) through three (3) as well as the value 255 are allocated in this specification, and can be found in Section 4.6.46. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP Radio Stats Failure Type registry, whose format is:

WTP Radio Stats Failure Type	Type Value	Reference

15.28. WTP Reboot Stats Failure Type

The Last Failure Type field in the WTP Reboot Statistics message element (see Section 4.6.47) is used to indicate the last reboot reason. The namespace is 8 bits (0-255), where the value of zero (0) through five (5) as well as the value 255 are allocated in this specification, and can be found in Section 4.6.47. This namespace is managed by IANA and assignments require a Expert Review. IANA will create the WTP Reboot Stats Failure Type registry, whose format is:

WTP Reboot Stats Failure Type	Type Value	Reference

16. Acknowledgments

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

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

17. References

17.1. Normative References

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2598] Jacobson, V., Nichols, K., and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, July 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[ISO.9834-1.1993]
International Organization for Standardization, "Procedures for the operation of OSI registration authorities - part 1: general procedures", ISO Standard 9834-1, 1993.

[I-D.ietf-capwap-protocol-binding-ieee80211]
Montemurro, M., Stanley, D., and P. Calhoun, "CAPWAP Protocol Binding for IEEE 802.11", draft-ietf-capwap-protocol-binding-ieee80211-11 (work in progress), October 2008.

[I-D.ietf-capwap-dhc-ac-option]
Calhoun, P., "CAPWAP Access Controller DHCP Option", draft-ietf-capwap-dhc-ac-option-02 (work in progress), October 2008.

[FRAME-EXT]
IEEE, "IEEE Standard 802.3as-2006", 2005.

17.2. Informational References

[RFC3232] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[RFC3753] Manner, J. and M. Kojo, "Mobility Related Terminology", RFC 3753, June 2004.

[RFC4564] Govindan, S., Cheng, H., Yao, ZH., Zhou, WH., and L. Yang, "Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)", RFC 4564, July 2006.

[RFC4962] Housley, R. and B. Aboba, "Guidance for Authentication, Authorization, and Accounting (AAA) Key Management", BCP 132, RFC 4962, July 2007.

[I-D.ohara-capwap-lwapp] Calhoun, P., "Light Weight Access Point Protocol", draft-ohara-capwap-lwapp-04 (work in progress), March 2007.

[I-D.narasimhan-ietf-slapp] Narasimhan, P., "SLAPP : Secure Light Access Point Protocol", draft-narasimhan-ietf-slapp-01 (work in progress), March 2006.

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

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

[EUI-64] IEEE, "GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY".

[EPCGlobal] "See http://www.epcglobalinc.org/home".

[PacketCable] "PacketCable Security Specification PKT-SP-SEC-I12-050812", August 2005, <PacketCable>.

[CableLabs] "OpenCable System Security Specification OC-SP-SEC-I07-061031", October 2006, <CableLabs>.

[WiMAX] "WiMAX Forum X.509 Device Certificate Profile Approved

Specification V1.0.1", April 2008, <WiMAX>.

Editors' Addresses

Pat R. Calhoun
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134

Phone: +1 408-902-3240
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