CAPWAP Working Group
Internet-Draft
Expires: December 2, 2005

P. Narasimhan
Aruba Networks
D. Harkins
Trapeze Networks
S. Ponnuswamy
Aruba Networks
May 31, 2005

# SLAPP : Secure Light Access Point Protocol
# draft-narasimhan-ietf-slapp-01

## Status of this Memo



## Copyright Notice



## Abstract

The CAPWAP problem statement [3] describes a problem that needs to be addressed before a wireless LAN (WLAN) network designer can construct a solution composed of Wireless Termination Points (WTP) and Access Controllers (AC) from multiple, different vendors. One of the primary goals is to find a solution that solves the interoperability

between the two classes of devices (WTPs and ACs) which then enables an AC from one vendor to control and manage a WTP from another.

The interoperability problem is more general than as stated in the CAPWAP problem statement [3] because it can arise out of other networks that do not necessarily involve WLAN or any wireless devices. A similar problem exists in any network that is composed of network elements that are managed by a centralized controller where these two classes of devices are from different vendors and need to interoperate with each other such that the network elements can be controlled and managed by the controller.

A possible solution to this problem is to split it into two parts - one that is technology or application independent which serves as a common framework across multiple underlying technologies, and another that is dependent on the underlying technology that is being used in the network. For example, methods and parameters used by an 802.11 AC to configure and manage a network of 802.11 WTPs are expected to be quite different than that used by an equivalent 802.16 controller to manage a network of 802.16 base stations. The architectural choices for these two underlying technologies may also be significantly different.

In this draft, we present a protocol that forms the common technology-independent framework and the ability to negotiate and add, on top of this framework, a control protocol that contains a technology-dependent component to arrive at a complete solution. We have also presented two such control protocols - an 802.11 Control protocol, and another a more generic image download protocol, in this draft.

Even though the text in this draft is written to specifically address the problem stated in [3], the solution can be applied to any problem that has a controller (equivalent to the AC) managing one or more network elements (equivalent to the WTP).

Table of Contents

# 1. Definitions

## 1.1 Conventions used in this document

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

## 2. Introduction

The need for a protocol by which wireless LAN (WLAN) Access Controllers (AC) can control and manage Wireless Termination Points (WTP) from a different vendor has been presented in the CAPWAP problem statement [3]. We believe that this problem is more general than as stated in [3] and can be found in any application, including non-wireless ones, that requires a central controller to control and manage one or more network elements from a different vendor.

One way to solve the CAPWAP problem is to define a complete control protocol that enables an AC from one vendor to control and manage a WTP from a different vendor. But a solution that is primarily focused towards solving the problem for one particular underlying technology (IEEE 802.11, in this case) may find it difficult to address other underlying technologies. Different underlying technologies may differ on the set of configurable options, and different architectural choices that are specific to that underlying technology (similar to the local MAC vs. split MAC architectures in 802.11). The architectural choices that are good for one underlying technology may not necessarily work for another. Not to forget that there may be multiple architectural choices [2] even for the same underlying technology. A monolithic control protocol that strives to solve this problem for multiple technologies runs the risk of adding too much complexity and not realizing the desired goals, or it runs the risk of being too rigid and hampering technological innovation.

A different way to solve this problem is to split the solution space into two components - one that is technology agnostic or independent, and another that is specific to the underlying technology or even different approaches for the same underlying technology. The technology-independent component would be a common framework that would be an important component of the solution to this class of problems without any dependency on the underlying technology (i.e. 802.11, 802.16, etc.) being used. The technology-specific component would be a control protocol that would be negotiated using this common framework and can be easily defined to be relevant to that technology without the need for having any dependency on other underlying technologies. This approach also lends itself easily to extend the solution as new technologies arise or as new innovative methods to solve the same problem for an existing technology present themselves later in the future.

In this draft, we present secure light access point protocol (SLAPP), a technology-independent protocol by which network elements that are meant to be centrally managed by a controller can discover one or more controllers, perform a security association with one of them, and negotiate a control protocol that they would use to perform the

technology-specific components of the control and provisioning protocol. We have also presented two control protocols in this draft - an 802.11 control protocol for provisioning and managing a set of 802.11 WTPs, and an image download protocol that is very generic and can be applied to any underlying technology.

Figure 1 shows the model by which a technology-specific control protocol can be negotiated using SLAPP to complete a solution for a certain underlying technology. The figure shows a control protocol each for 802.11 and 802.16 technology components, but the SLAPP model does not preclude multiple control protocols within a certain technology segment. For example, a certain technology-specific control protocol may choose to support only the local MAC architecture [2] while deciding not to support the split MAC architecture [2]. While the image download protocol is presented in this draft, a SLAPP implementation MUST NOT assume that this control protocol is supported by other SLAPP implementations.

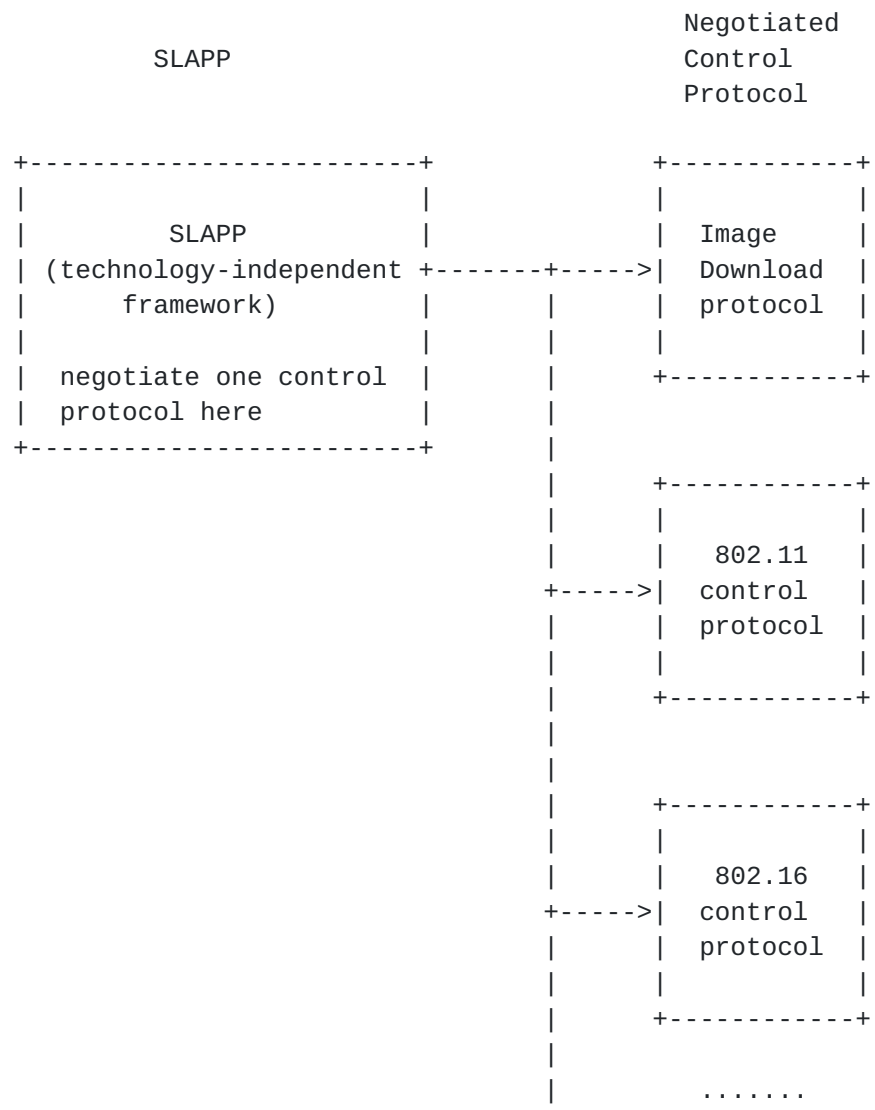



Figure 1: SLAPP Protocol Model

The control protocols that are negotiable using SLAPP are expected to be published ones that have gone through a review process in standards bodies such as the IETF. The control protocols can either re-use the security association created during SLAPP or have the option of clearing all SLAPP state and restarting with whatever mechanisms are defined in the control protocol.

Recently, there was a significant amount of interest in a similar problem in the RFID space that has led to the definition of a simple lightweight RFID reader protocol (SLRRP) [10]. It is quite possible that SLRRP could be a technology-specific (RFID, in this case) control protocol negoriated during a common technology-independent framework.

All of the text in the draft would seem to be written with a WLAN problem in mind. Please note that while the letter of the draft does position the solution to solve a CAPWAP-specific problem, the spirit of the draft is to address the more general problem.

## 3. Topology

The SLAPP protocol supports multiple topologies for interconnecting WTPs and ACs as indicated in Figure 2.

In Figure 2, we have captured four different interconnection topologies.

1. The WTP is directly connected to the AC without any intermediate nodes. Many WTPs are deployed in the plenum of buildings and are required to be powered over the Ethernet cable that is connecting it to the network. Many ACs in the marketplace can supply power over etherent and in the case where the AC is the one powering the WTP, the WTP is directly connected to the AC.

2. The WTP is not directly connected to the AC, but both the AC and the WTP are in the same L2 (broadcast) domain.

3. The WTP is not directly connected to the AC, and they are not present in the same L2 (broadcast) domain. They are on two different broadcast domains and have a node on the path that routes between two or more subnets.

4. The fourth case is a subset of the third one with the exception that the intermediate nodes on the path from the WTP to the AC may not necessarily be in the same administrative domain. The intermediate network may also span one or more WAN links that may have lower capacity than if both the AC and the WTP are within the same building or campus.

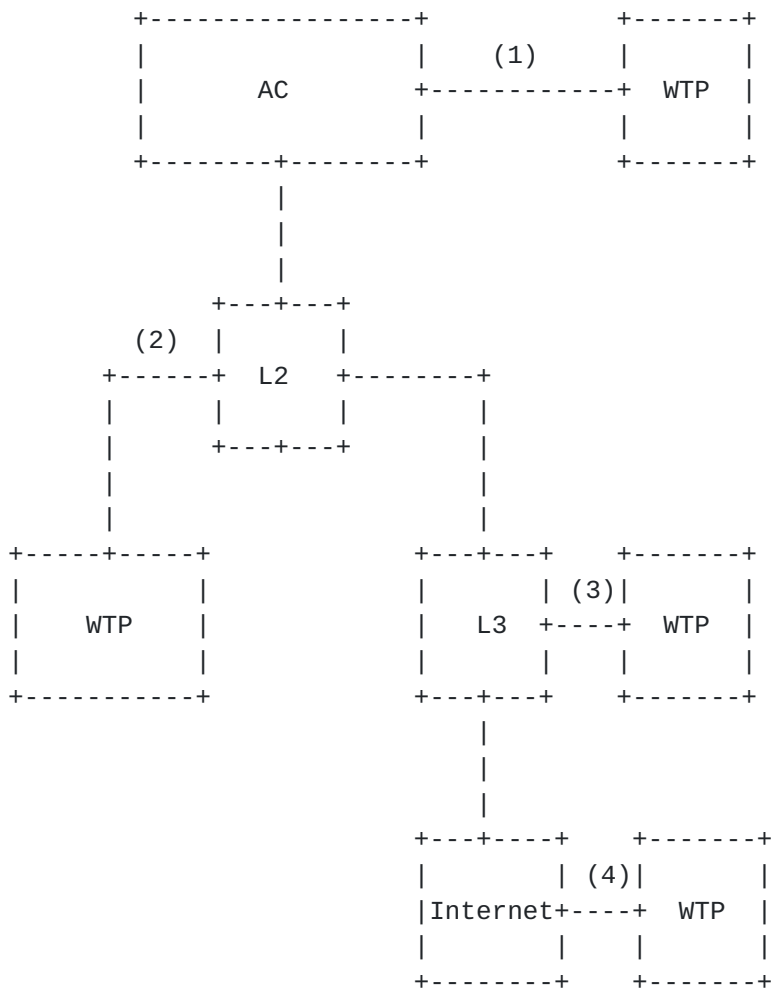



Figure 2: SLAPP Topology

# 4. Protocol

## 4.1 Protocol Description

The SLAPP state machine for both the WTP and AC is shown in Figure 3. Both the WTP and the AC discover each other, negotiate a control protocol, perform a secure handshake to establish a secure channel between them, and then use that secure channel to protect a negotiated control protocol.

The WTP maintains the following variable for its state machine:

abandon: a timer which sets the maximum amount of time the WTP will wait for an acquired AC to begin the DTLS handshake.

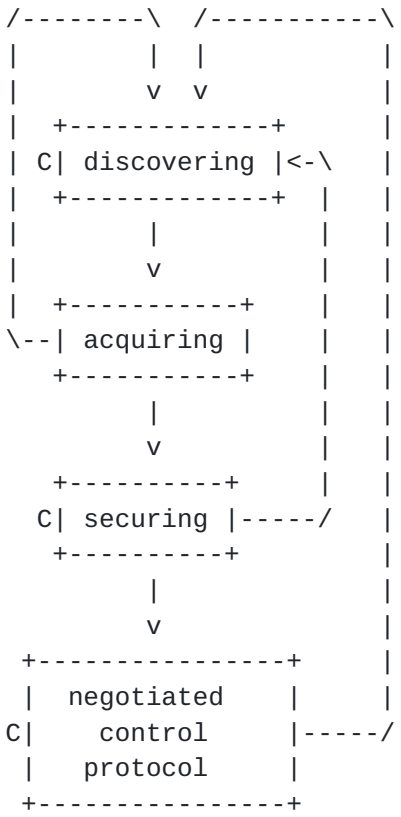



Figure 3: SLAPP State Machine

### 4.1.1 State Machine Explanation

Note: the symbol "C" indicates an event which results in the state remaining the same.

Discovering

AC: this is a quiescent state for the AC in which it waits for WTPs to request its acquisition. When a request is received the AC transitions to Acquiring.

WTP: the WTP is actively discovering an AC. When the WTP receives a response to its discovery request it transitions to Acquiring.

Acquiring

AC: a discover request from a WTP has been received. If the request is invalid or the AC wishes to not acquire the WTP it drops the packet and transitions back to Discovering. Otherwise a discovery response is sent and the AC transitions to Securing.

WTP: a discover response from an AC has been received. If the response is not valid the WTP transitions to Discovering, otherwise it sets the abandon timer to a suitable value to await a DTLS exchange. If the timer fires in Acquiring the WTP transitions back to Discovering. If a DTLS "client hello" is received the WTP transitions to Securing and cancels the abandon timer.

Securing

AC: The AC performs the "client end" of the DTLS exchange. Any error in the DTLS exchange results in the AC transitioning to Discovering. When the DTLS exchange finishes the AC transitions to Negotiated Control Protocol.

WTP: The WTP performs the "server end" of the DTLS exchange. Any error in the DTLS exchange results in the WTP transitioning to Discovering. When the DTLS exchange finishes the WTP transitions to the Negotiated Control Protocol.

Negotiated Control Protocol

AC: the AC performs its side of the protocol agreed to during the discovery process. Please refer to Section 6.1 for the SLAPP 802.11 control protocol. For the Image Download Protocol example see section Section 6.2.

WTP: the WTP performs its side of the protocol agreed to during the discovery process. Please refer to Section 6.1 for the SLAPP 802.11 control protocol. For the Image Download Protocol example see section Section 6.2.

## 4.2 Format of a SLAPP Header

All SLAPP packets begin with the same header as shown in Figure 4.

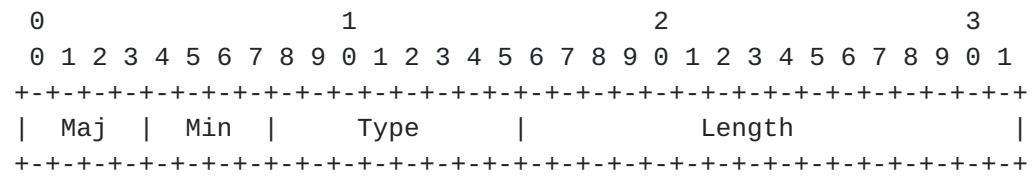



Figure 4: SLAPP Header

Where:

Maj (4 bits): the major number of the SLAPP version

Min (4 bits): the minor number of the SLAPP version

Type (1 octet): the type of SLAPP message

Length (two octets): the length of the SLAPP message, including the entire SLAPP header

The following types of SLAPP messages have been defined:

| name | type |
|---|---|
| discovery request | 1 |
| discovery response | 2 |
| image download control | 3 |
| control protocol packet | 4 |
| reserved | 5-255 |

## 4.3 Version

SLAPP messages include a version in the form of major.minor. This document describes the 1.0 version of SLAPP, that is the major version is one (1) and the minor version is zero (0).

Major versions are incremented when the format of a SLAPP message changes or the meaning of a SLAPP message changes such that it would

not be properly parsed by an older, existing version of SLAPP. Minor versions are incremented when some incremental additions have been made to SLAPP that enhance its capabilities or convey additional information in a way that does not change the format or meaning of the SLAPP message.

Future versions of SLAPP MAY NOT mandate support for earlier major versions of SLAPP so an implementation MUST NOT assume that a peer that supports version "n" will therefore support version "n - i" (where both "n" and "i" are non-zero integers and "n" is greater than "i").

A SLAPP implementation that receives a SLAPP message with a higher major version number MUST drop that message. A SLAPP implementation that receives a SLAPP message with a lower major version SHOULD drop down to the version of SLAPP the peer supports. If that version of SLAPP is not supported the message MUST be dropped. There may be valid reasons for which a peer wishes to drop a SLAPP message with a supported major version though.

A SLAPP implementation that receives a SLAPP message with a higher minor version number MUST NOT drop that message. It MUST respond with the minor version number that it supports and will necessarily not support whatever incremental capabilities were added that justified the bump in the minor version. A SLAPP implementation that receives a SLAPP message with a lower minor version MUST NOT drop that message. It SHOULD revert back to the minor version which the peer supports and not include any incremental capabilities that were added which justified the bump in the minor version.

### 4.4 Retransmission

SLAPP is a request response protocol. Discovery and security handshake requests are made by the WTP and responses to them are made by the AC. Image download packets are initiated by the AC and acknowledged by the WTP (in a negative fashion, see Section 6.2).

Retransmissions are handled solely by the initiator of the packet. After each packet for which a response is required is transmitted, the sender MUST set a retransmission timer and resend the packet upon its expiry. The receiver MUST be capable of either regenerating a previous response upon receipt of a retransmitted packet or caching a previous response and resending upon receipt of a retransmitted packet.

The retransmission timer MUST be configurable and default to one (1) second. No maximum or minimum for the timer is specified by this version of SLAPP.

Each time a retransmission is made a counter SHOULD be incremented and the number of retransmissions attempted by a sender before giving up and declaring a SLAPP failure SHOULD be four (4)-- that is, the number of attempts made for each packet before declaring failure is five (5).

The exception to this rule is Image Download packets which are not individually acknowledged by the WTP (see Section 6.2). The final packet is acknowledged and lost packets are indicated through Image Download Requests.

### 4.5 Discovery

When a WTP boots up and wants to interoperate with an Access Controller so that it can be configured by the AC, one of the first things it needs to do is to discover one or more ACs in its network neighborhood. This section contains the details of this discovery mechanism.

As described in Section 3, an AC and a WTP could reside in the same layer 2 domain, or be separated by a layer 3 cloud including intermediate clouds that are not under the same administrative domain (for example, an AC and a WTP separated by a wide-area public network). So any proposed discovery mechanism should have provisions to enable a WTP to discover an AC across all these topologies.

We assume that a WTP prior to starting the discovery process has already obtained an IP address on its wired segment.

#### 4.5.1 SLAPP Discover Request

The SLAPP discovery process is initiated by sending a SLAPP discover request packet. The packet can be addressed to the broadcast IP address, a well known multicast address, or (if the IP address of an AC is either configured prior to the WTP booting up or is learned during the boot-up sequence) addressed to a unicast IP address. Lack of a response to one method of discovery SHOULD result in the WTP trying another method of discovery. The SLAPP discover request packet is a UDP packet addressed to port [TBD] designated as the SLAPP discovery port. The source port can be any random port. The payload of the SLAPP discover request packet is shown in Figure 5.

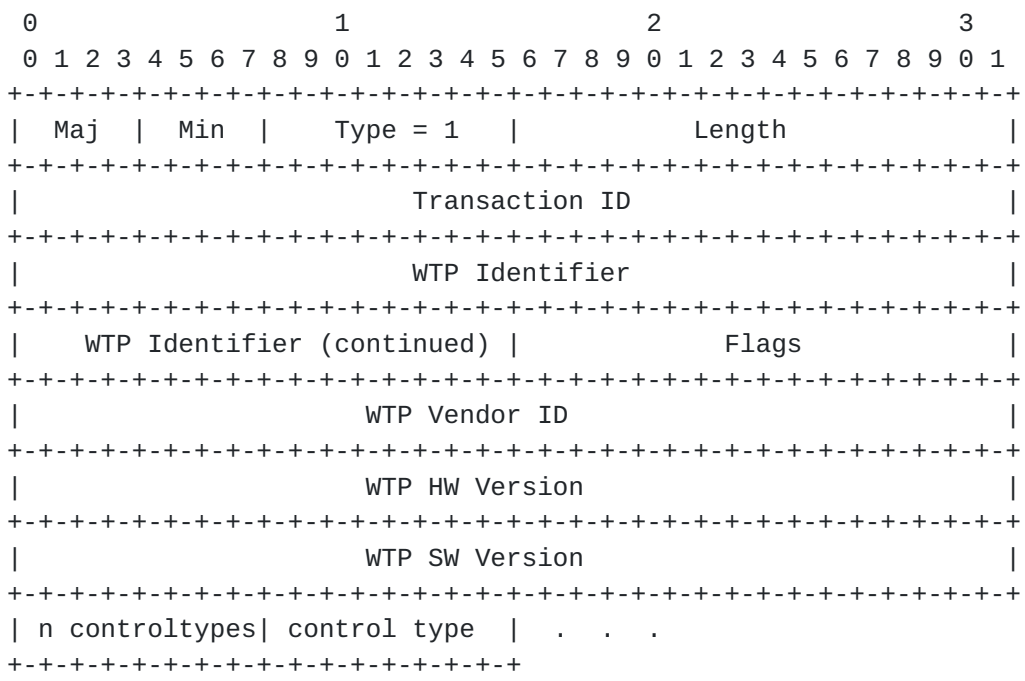



Figure 5: SLAPP Discover Request

#### 4.5.1.1 Transaction ID

The transaction ID is a randomly generated 32-bit number that is maintained during one phase of the SLAPP discovery process. It is generated by a WTP starting a discovery process. When one discovery method fails to find an AC and the WTP attempts another discovery method it MUST NOT reuse the Transaction ID. All ACs that intend to respond to a SLAPP discover request must use the same value for this field as in the request frame.

#### 4.5.1.2 WTP Identifier

This field allows the WTP to specify a unique identifier for itself. This MAY be, for instance, its 48-bit MAC address or it could be any other string such as a serial number.

#### 4.5.1.3 Flags

The flags field is used to indicate certain things about the discover request. For example, bit 0 in the flags field indicates whether the discover request packet is being sent to the AC, if unicast, based on a configuration at the WTP or based on some other means of discovery. This bit should always be set to the discover mode if the SLAPP discover request packet is being sent to either a broadcast or

multicast address. Here are the valid values for various bits in the Flags field.

Bit 0:
0 - Configuration mode
1 - Discover mode

Bits 1-15:
Must always be set to 0 by the transmitter
Must be ignored by the receiver

#### 4.5.1.4 WTP Vendor ID

This 32-bit field is the WTP vendor's SMI enterprise code in network octet order (these enterprise codes can be obtained from, and registered with, IANA).

#### 4.5.1.5 WTP HW Version

This 32-bit field indicates the version of hardware present in the WTP. This is a number that is totally left to the WTP vendor to choose.

#### 4.5.1.6 WTP SW Version

This 32-bit field indicates the version of software present in the WTP. This is a number that is totally left to the WTP vendor to choose.

#### 4.5.1.7 number of control types

This 8-bit field indicates the number of 8-bit control protocol indicators that follow it and therefore implicitly indicates the number of different control protocols the the WTP is capable of supporting. This number MUST be at least one (1).

#### 4.5.1.8 control types

This 8-bit field indicates the type of control protocol the WTP supports and is willing to use when communicating with an AC. There MAY be multiple "control type" indicators in a single SLAPP Discover Request.

```
Valid control types
-------------------
0      - RESERVED (MUST not be used)
1      - Image Download Control Protocol
2      - 802.11 SLAPP control protocol
3-255  - RESERVED (to IANA)
```

### 4.5.2 SLAPP Discover Response

An AC that receives a SLAPP discover request packet from a WTP can choose to respond with a SLAPP discover response packet. The format of the SLAPP discover response packet is shown in Figure 6.

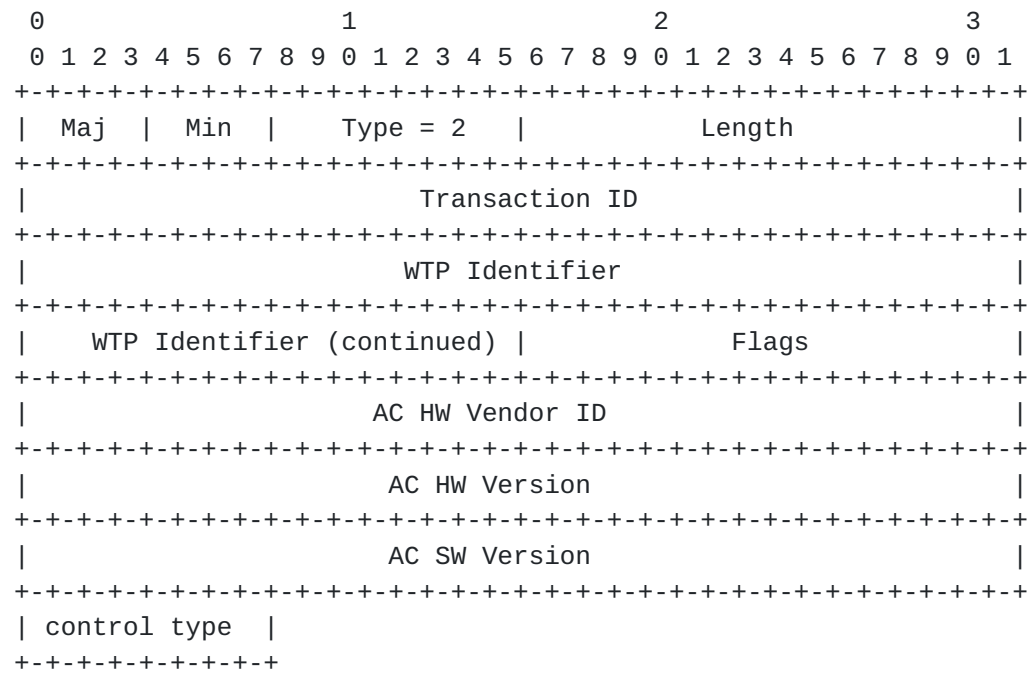



Figure 6: SLAPP Discover Response

The SLAPP discover response packet is a UDP packet. It is always unicast to the WTP's IP address. The source IP address is that of the AC sending the response. The source port is the SLAPP discover port [TBD] and the destination port is the same as the source port used in the SLAPP discover request. The WTP's MAC address and the transaction ID must be identical to the values contained in the SLAPP discover request. The Status field indicates to the WTP whether the AC is either accepting the discover request and is willing to allow the WTP to proceed to the next stage (ACK) or whether it is denying the WTP's earlier request (NACK). The AC includes its own vendor ID, hardware and software versions in the response.

#### 4.5.2.1 Transaction ID

The value of the Transaction ID field should be identical to its value in the SLAPP discover request packet sent by the WTP.

#### 4.5.2.2 WTP Identifier

The WTP Identifier that was sent in the corresponding SLAPP discover request frame.

#### 4.5.2.3 Flags

This field is unused by this version of SLAPP. It MUST be set to zero (0) on transmission and ignored upon receipt.

#### 4.5.2.4 AC Vendor ID

If the value of the status field is a 1, indicating that the AC is sending a successful response, then the values in this field and the following two are valid. The 32-bit AC Vendor ID points to the vendor ID of the AC. If the value of the status field is not 1, then this field should be set to 0 by the AC and ignored by the WTP.

#### 4.5.2.5 AC HW Version

If the value of the status field is 1, then this 32-bit field contains the value of the AC's hardware version. This value is chosen by the AC vendor. If the value of the status field is not a 1, then this field should be set to 0 by the AC and ignored by the WTP.

#### 4.5.2.6 AC SW Version

If the value of the status field is 1, then this 32-bit field contains the value of the AC's software version. This value is chosen by the AC vendor. If the value of the status field is not a 1, then this field should be set to 0 by the AC and ignored by the WTP.

#### 4.5.2.7 Control Type

The control type the AC will use to communicate with the WTP. This value MUST match one of the control types passed in the corresponding SLAPP Discover Request.

### 4.6 SLAPP Discovery Process

### 4.6.1 WTP

There are multiple ways in which a WTP can discover an AC.

1. Static configuration: An administrator, prior to deploying a WTP, can configure an IP address of an AC on the WTP's non-volatile memory. If this is the case, then the SLAPP discover request packet is addressed to the configured IP address.

2. DHCP options: As part of the DHCP response, the DHCP server could be configured to use option 43 to deliver the IP address of an AC to which the WTP should address the SLAPP discover request packet. If the IP address of an AC is handed to the WTP as part of the DHCP response, then the WTP should address the SLAPP discover request packet to this IP address.

3. DNS configuration: Instead of configuring a static IP address on the WTP's non-volatile memory, an administrator can configure a FQDN of an AC. If the FQDN of an AC is configured, then the WTP queries its configured DNS server for the IP address associated with the configured FQDN of the AC. If the DNS query is successful and the WTP acquires the IP address of an AC from the DNS server, then the above discover request packet is addressed to the unicast address of the AC.

4. Broadcast: The WTP sends a discover request packet addressed to the broadcast IP address with the WTP's IP address as the source. A network administrator, if necessary, could configure the default router for the subnet that the WTP is on with a helper address and unicast it to any address on a different subnet.

5. IP Multicast: A WTP can send the above payload to a SLAPP IP multicast address [TBD].

6. DNS: If there is no DNS FQDN configured on the WTP, and the WTP is unable to discover an AC by any of the above methods, then it should attempt to query the DNS server for a well known FQDN of an AC [TBD]. If this DNS query succeeds, then the WTP should address the SLAPP discover request packet to the unicast address of the AC.

The above process is summarized in the sequence shown in Figure 7.

```
    SLAPP discovery start:
       Static IP address config option:
         Is a static IP address for an AC configured?
           If yes, send SLAPP discover request to that unicast IP address
             SLAPP discover response within discovery_timer?
               If yes, go to "done"
               If not, go to "Static FQDN config option"
           If not, go to "Static FQDN config option"
       Static FQDN config option:
         Is a static FQDN configured?
           If yes, send a DNS query for the IP address for the FQDN.
           Is DNS query successful?
             If yes, send SLAPP discover request to that IP address
             SLAPP discover response within discovery timer?
               If yes, go to "done"
               If not, go to "DHCP options option"
             If not, go to "DHCP options option"
        DHCP options option:
          Is the IP address of an AC present in the DHCP response?
            If yes, send SLAPP discover request to the AC's IP addr
            SLAPP discover response within discovery timer?
              If yes, go to "done"
              If not, go to "Broadcast option"
            If not, go to "Broadcast option"
        Broadcast option:
          Send SLAPP discover packet to the broadcast address
          SLAPP discover response within discovery timer?
            If yes, go to "done"
            If not, go to "Multicast option"
        Multicast option:
          Send SLAPP discover packet to the SLAPP multicast address
          SLAPP discover response within discovery timer?
            If yes, go to "done"
            If not, go to "DNS discovery option"
        DNS discovery option:
          Query the DNS server for a well known DNS name
          Is the DNS discovery successful?
            If yes, send SLAPP discover request to that IP addr
            SLAPP discover response within discovery timer?
              If yes, go to "done"
              If not, go to "SLAPP discovery restart"
            If not, go to "SLAPP discovery restart"
        SLAPP discovery restart:
          Set timer for SLAPP discovery idle timer
          When timer expires, go to "SLAPP discovery start"
        done:
          go to the next step
```

Figure 7

### 4.6.2 AC

When an AC receives a SLAPP discover request it must determine whether it wishes to acquire the WTP or not. An AC MAY only agree to acquire those WTPs whose WTP Identifiers are statically configured in its configuration. Or an AC that is willing to gratuitously acquire WTPs MAY accept any request pending authentication. An AC MUST only choose to acquire WTPs that speak a common Negotiated Control protocol but other factors may influence its decision. For instance, if the Negotiated Control Protocol is the Image Download protocol defined in this memo the AC MUST NOT acquire a WTP for which it does not have a compatible image to download as determined by the WTP's HW Vendor ID, HW Version and Software Version. Whatever its decision, the AC MUST respond one of two ways.

1. The AC sends a SLAPP discover response indicating its agreement to acquire the WTP.

2. The AC silently drops the SLAPP discover request and does not respond at all.

## 5. Security Association

Once an AC has been discovered by a WTP and agreed to acquire it (by sending a Discovery Response) it will initiate a DTLS [7] [9] exchange with the WTP by assuming the role of the "client". The WTP assumes the role of the "server". The port used by both the WTP and AC for this exchange will be [TBD].

An obvious question is "why is the AC acting as a client?" The reason is to allow for non-mutual authentication in which the WTP is authenticated by the AC (see Section 5.1.2).

Informational note: DTLS is used because it provides a secure and connectionless channel using a widely accepted and analyzed protocol. In addition, the myriad of authentication options in DTLS allows for a wide array of options with which to secure the channel between the WTP and the AC-- mutual and certificate based; asymmetric or non-mutual authentication; anonymous authentication; etc. Furthermore, DTLS defines its own fragmentation and reassembly techniques as well as ways in which peers agree on an effective MTU. Using DTLS obviates the need to redefine these aspects of a protocol and therefore lessens code bloat as the same problem doesn't need to be solved yet again in another place.

Failure of the DTLS handshake protocol will cause both parties to abandon the exchange. The AC SHOULD blacklist this WTP for a period of time to prevent a misconfigured WTP from repeatedly discovering and failing authentication. The WTP MUST return to the discovery state of SLAPP to locate another suitable AC with which it will initiate a DTLS exchange.

Once the DTLS handshake has succeeded the WTP and AP transition into "image download state" and protect all further SLAPP messages with the DTLS-negotiated cipher suite.

### 5.1 Example Authentication Models (Informative)

Any valid cipher suite in [8] can be used to authenticate the WTP and/or the AC. Different scenarios require different authentication models. The following examples are illustrative only and not meant to be exhaustive.

Since neither side typically involves a human being a username/ password based authentication is not possible.

Zero-config requirements on certain WTP deployments can predicate certain authentication options and eliminate others.

### 5.1.1 Mutual Authentication

When mutually authenticating, the WTP authenticates the AC, thereby ensuring that the AC to which it is connecting is a trusted AC, and the AC authenticates the WTP, thereby ensuring that the WTP that is connecting is a trusted WTP.

Mutual authentication is typically achieved by using certificates on the WTP and AC which ensure public keys each party owns. These certificates are digitally signed by a Certification Authority, a trusted third party.

Enrolling each WTP in a Certification Authority is outside the scope of this document but it should be noted that a manufacturing Certification Authority does not necessarily provide the level of assurance necessary as it will only guarantee that a WTP or AC was manufactured by a particular company and cannot distinguish between a trusted WTP and a WTP which is not trusted but was purchased from the same manufacturer as the AC.

### 5.1.2 WTP-only Authentication

Some deployments may only require the WTP to authenticate to the AC and not the other way around.

In this case the WTP has a keypair which can uniquely identify it (for example, using a certificate) and that keypair is used in a "server-side authentication" [8] exchange.

This authentication model does not authenticate the AC and a rogue AC could assert control of a valid WTP. It should be noted, though, that this will only allow the WTP to provide service for networks made available by the rogue AC. No unauthorized network access is possible.

### 5.1.3 Anonymous Authentication

In some deployments it MAY just be necessary to foil the casual snooping of packets. In this case an unauthenticated, but encrypted, connection can suffice. Typically a Diffie-Hellman exchange is performed between the AC and WTP and the resulting unauthenticated key is used to encrypt traffic between the AC and WTP.

## 6. SLAPP Control Protocols

In this section, we describe two extensions for SLAPP - one that is specific to 802.11 WLANs and another that is a technology neutral protocol by which an AC can download a bootable image to a WTP.

### 6.1 802.11 Control Protocol for SLAPP

This section describes a SLAPP extension that is targeted towards WTPs and ACs implementing the IEEE 802.11 WLAN standard. This extension contains all the technology-specific component that will be used by an AC to control and manage 802.11 WTPs.

#### 6.1.1 Suppported CAPWAP Architectures

The CAPWAP architecture taxonomy document [2] describes multiple architectures that are in use today in the WLAN industry. While there is a wide spectrum of variability present in these documented architectures, supporting every single variation or choice would lead to a complex protocol and negotiation phase. In the interest of limiting the complexity of the 802.11 component, we have limited the negotiation to four different architectural choices as listed below.

Local MAC, bridged mode : This mode of operation falls under the Local MAC architecture. The 802.11 MAC is terminated at the WTP. The WTP implements an L2 bridge that forwards packets between its WLAN interface and its ethernet interface.

Local MAC, tunneled mode : This mode of operation also falls under the Local MAC architecture where the 802.11 MAC is terminated at the WTP. The difference between this mode and the previous one is that in this mode, the WTP tunnels 802.3 frames to the AC using the mechanisms defined in Section 6.1.2.

Split MAC, L2 crypto at WTP : This mode of operation falls under the split MAC architecture. The 802.11 MAC is split between the WTP and the AC, the exact nature of the split is described in Section 6.1.1.2. The L2 crypto functions are implemented in the WTP are the ones used to satisfy this function irrespective of whether the AC is also capable of this function or not. The WTP tunnels L2 frames to the AC using mechanisms defined in Section 6.1.2.

Split MAC, L2 crypto at AC : This mode of operation also falls under the split MAC architecture. The difference between this one and the previous one is that the L2 crypto functions implemented in the AC are used to satisfy this function irrespective of whether these functions are also available at the WTP or not. The WTP

tunnels L2 frames to the AC using mechanisms defined in Section 6.1.2.

#### 6.1.1.1 Local MAC

The Local MAC architecture as documented in the CAPWAP architecture taxonomy document [2] performs all 802.11 frame processing at the WTP. The conversion from 802.11 to 802.3 and vice-versa is also implemented at the WTP. This would mean that other functions like fragmentation/reassembly of 802.11 frames, encryption/decryption of 802.11 frames is implemented at the WTP.

##### 6.1.1.1.1 Bridged Mode

In this sub-mode of the Local MAC architecture, the 802.11 frames are converted to 802.3 frames and bridged onto the Ethernet interface of the WTP. These frames may be tagged with 802.1Q VLAN tags assigned by the AC.

##### 6.1.1.1.2 Tunneled Mode

In this sub-mode of the Local MAC architecture, the 802.11 frames are converted to 802.3 frames and are tunneled (using the tunneling mechanism defined in Section 6.1.2) to the AC that the WTP is attached to. These frames may be tagged with 802.1Q VLAN tags assigned by the AC.

#### 6.1.1.2 Split MAC

In the split MAC architecture, the MAC functions of an 802.11 AP are split between the WTP and the AC. The exact nature of the split is dependent upon the sub-modes listed in this section. In both cases, frames are tunneled to the AC using the mechanism defined in Section 6.1.2.

Some of these split MAC architectures convert the 802.11 frames into 802.3 frames, which may be 802.1Q tagged using tags assigned by the AC, while other of these split MAC architectures will tunnel the entire 802.11 frame to the AC. The AC and WTP agree on what type of frame will be tunneled during the control protocol registration Section 6.1.3

##### 6.1.1.2.1 L2 Crypto at the WTP

For this sub-mode of the split MAC architecture, the 802.11 AP functions are split as follows:

At the WTP

802.11 control frame processing

802.11 encryption and decryption

802.11 fragmentation and reassembly

Rate Adaptation

802.11 beacon generation

Power-save buffering and TIM processing

At the AC

802.11 Management frame processing

802.11 DS and portal

Split MAC implementations of this kind may tunnel either 802.11 or 802.3 frames between the AC and the WTP.

#### 6.1.1.2.2 L2 Crypto at the AC

For this sub-mode of the split MAC architecture, the 802.11 AP functions are split as follows:

At the WTP

802.11 control frame processing

Rate Adaptation

802.11 beacon generation

Power-save buffering and TIM processing

At the AC

802.11 Management frame processing

802.11 encryption and decryption

802.11 fragmentation and reassembly

802.11 DS and portal

Split MAC implementations of this kind tunnel 802.11 frames between the AC and the WTP.

### 6.1.2 Transport

The 802.11 Control protocol has two components, one for transporting the specific control and provisioning messages and another to tunnel data traffic from the WTP to the AC.

The SLAPP 802.11 Control Protocol uses the Generic Routing Encapsulation (GRE) [4] to encapsulate L2 frames. Depending on whether and how an architecture splits its MAC some architectures may tunnel 802.11 frames directly to the AC while others may tunnel 802.3 frames which may be optionally 802.1Q tagged using tags assigned by the AC.

The delivery mechanism of these GRE packets is IP. Therefore the IP protocol of the outer packet is 47, indicating a GRE header follows. When GRE encapsulates 802.11 frames the ether type in the GRE header is TBD; when GRE encapsulates 802.3 frames the ether type in the GRE header is TBD2.

Since IP is the delivery mechanism all issues governing fragmentation and reassembly are handled by [5].

#### 6.1.2.1 SLAPP 802.11 Control Protocol Header

When using the 802.11 control protocol the type of SLAPP message is four (4), "control protocol packet". In this case a two (2) octet field is appended to the SLAPP header to indicate the control protocol type as shown in Figure 8. The SLAPP 802.11 Control Protocol takes place in the "negoatiated control protocol" phase of Section 4.1 and all SLAPP 802.11 Control Protocol messages are therefore secured by the security association created immediately prior to entering that phase.

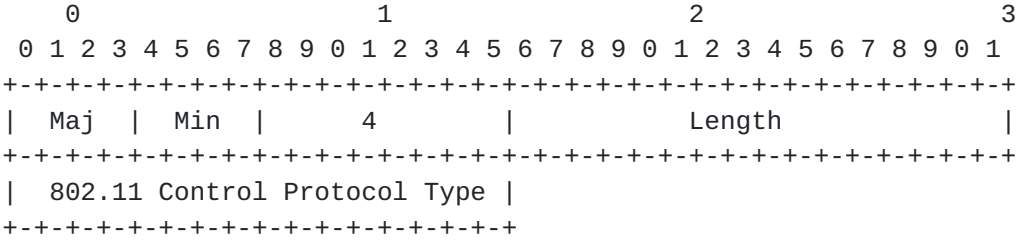



Figure 8: SLAPP Control Protocol Header

Where valid 802.11 Control Protocol Types are:

1 : Registration request - sent from WTP to AC

2 : Registration response - sent from AC to WTP

3 : De-registration request - sent by either WTP or AC

4 : De-registration response - sent by the recipient of the corresponding request

5 : Configuration request - sent by WTP to AC

6 : Configuration response - sent by AC to WTP

7 : Configuration update - sent by AC to WTP

8 : Configuration acknowledgment - sent by the WTP to AC

9 : Status request - sent by the AC to the WTP

10 : Status response - sent by the WTP to the AC

11 : Statistics request - sent by the AC to the WTP

12 : Statistics response - sent by the WTP to the AC

13 : Event - sent by the WTP to the AC

14 : Keepalive - sent either way

15 : Key Config Request - sent by the AC to the WTP

16 : Key Config Response - sent by the WTP to the AC

### 6.1.3 Provisioning and Configuration of WTP

All basic configuration functions are applicable per-ESSID per-radio in a WTP. Some WTPs MAY support more than one ESSID per-radio, while all WTPs MUST support at least one ESSID per-radio, which may be considered the primary ESSID in case of multiple ESSID support. All per-WTP configurations and capabilities (e.g., number of radios) are handled as part of the discovery and initialization process.

The provisioning of the regulatory domain of a WTP is beyond the scope of this document. A WTP, once provisioned for a specific regulatory domain MUST restrict the operational modes, channel,

transmit power and any other necessary limits based on the knowledge contained within its software image and hardware capabilities. The WTP MUST communicate its capabilities limited by the regulatory domain as well as by the WTP hardware, if any, to the AC during the capability exchange.

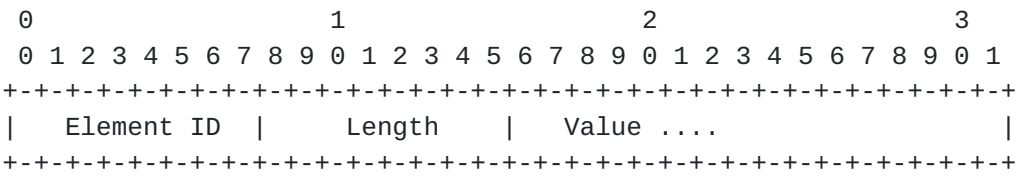
The allocation and assignment of BSSIDs to the primary interface and to the virtual AP interfaces, if supported, are outside the scope of this document.

#### 6.1.3.1 Information Elements

Information elements are used to communicate capability, configuration, status, and statistics information between the AC and the WTP.

##### 6.1.3.1.1 Structure of an Information Element

The structure of an information element is show below. The element ID starts with an element ID octet, followed by a 1-octet length and the value of the element ID whose length is indicated in the Length field. The maximum length of an element is 255 octets.




##### 6.1.3.1.2 CAPWAP Mode

This element defines the MAC architecture modes (Section 6.1.1).

Element ID : 1

Length : 1

Value : The following values are defined.

Bit 0 : CAPWAP mode 1 - Local MAC, bridged mode

Bit 1 : CAPWAP mode 2 - Local MAC, tunneled mode

Bit 2 : CAPWAP mode 3 - Split MAC, WTP encryption, 802.3 tunneling

Bit 3 : CAPWAP mode 4 - Split MAC, WTP encryption, 802.11 tunneling

Bit 4 : CAPWAP mode 5 - Split MAC, AC encryption, 802.11 tunneling

Bits 5-7 : set to 0

When this element is included in the capabilities message, then the setting of a bit indicates the support for this CAPWAP mode at the WTP. When this element is used in configuration and status messages, then exactly one of bits 0-4 MUST be set.

#### 6.1.3.1.3 Number of WLAN Interfaces

This element refers to the number of 802.11 WLAN present in the WTP.

Element ID : 2

Length : 1

Value : 0-255

#### 6.1.3.1.4 WLAN Interface Index

This element is used to refer to a particular instance of a WLAN interface when used in configuration and status messages. When used within a recursion element, the elements within the recursion element correspond to the WLAN interface specified in this element.

Element ID : 3

Length : 1

Value : 0 - (Number of WLAN interfaces - 1)

#### 6.1.3.1.5 WLAN Interface Hardware Vendor ID

This element is the WLAN Interface hardware vendor's SMI enterprise code in network octet order (these enterprise codes can be obtained from, and registered with, IANA). This field appears once for each instance of WLAN interface present in the WTP.

Element ID : 4

Length : 4

Value : 32-bit value

#### 6.1.3.1.6 WLAN Interface Type ID

This element is an ID assigned by the WLAN Interface hardware vendor to indicate the type of the WLAN interface. It is controlled by the hardware vendor and the range of possible values is beyond the scope of this document. This field appears once for each instance of WLAN interface present in the WTP.

Element ID : 5

Length : 4

#### 6.1.3.1.7 Regulatory Domain

If a regulatory domain is provisioned in the WTP, then the WTP indicates this by including this element in the capabilities list. If this information is not available at the WTP, then this element SHOULD not be included in the capabilities list. The process by which this information is provisioned into the WTP is beyond the scope of this document.

Element ID : 6

Length : 4

Value : ISO code assigned to the regulatory domain

#### 6.1.3.1.8 802.11 PHY mode and Channel Information

This element indicates the list of 802.11 PHY modes supported by the WTP along with a list of channels and maximum power level supported for this mode. This element appears once for each instance of WLAN interface at the WTP. There could be multiple instances of this element if the WLAN interface supports multiple PHY types.

Element ID : 7

Length : Variable

Valid : This field consists of

PHY mode : With a length of 1 octet with value values as follows:

0 : Radio Disabled/Inactive

1 : IEEE 802.11b

2 : IEEE 802.11g

3 : IEEE 802.11a

4-255 : Reserved

Power Level : In the capabilities messages, this indicates the maximum power level supported in this mode by the WTP, while in the configuration and status messages this field indicates the desired power level or the current power level that the WTP is operating at. The field has a length of 1 octet and the power level is indicated in dBm.

Channel Information : A variable number of 2-octet values that indicate the center frequencies (in KHz) of all supported channels in this PHY mode.

When this element is used in configuration and status messages, the power level field indicates the desired or current operating power level. The channel field has exactly one 2-octet value indicating the desired or current operating frequency.

#### 6.1.3.1.9 Cryptographic Capability

In the capabilities message, this element contains the list of cryptographic algorithms that are supported by the WTP. This appears once for each instance of the WLAN interface present in the WTP. In configuration and status messages, this element is used to indicate the configured cryptographic capabilities at the WTP.

Element ID : 8

Length : 1

Value : The following bits are defined.

Bit 0 : WEP

Bit 1 : TKIP

Bit 2 : AES-CCMP

Bits 3-7 : Reserved

#### 6.1.3.1.10 Other IEEE 802.11 Standards Support

This element contains a bitmap indicating support at the WTP for various IEEE 802.11 standards.

Element ID : 9

Length : 4

Value : A bitmap as follows

Bit 0 : WPA

Bit 1 : 802.11i

Bit 2 : WMM

Bit 3 : WMM-SA

Bit 4 : U-APSD

Bits 5-32 : Reserved

#### 6.1.3.1.11 Antenna Information Element

In the capabilities message, this element is formatted as follows

Element ID : 10

Length : 4

Value : Formatted as follows

Bits 0-7 : Number of Antennae

Bit 8 : Individually Configurable, 0 = No, 1 = Yes

Bit 9 : Diversity support, 0 = No, 1 = Yes

Bit 10 : 0 = Internal, 1 = External

Bits 11-31 : Reserved

In configuration and status messages, this element is formatted as

follows:

Element ID : 10

Length : 4

Value : Formatted as follows

Bits 0-7 : Antenna Number - is a number between 0 and the number of antennae indicated by the WTP. The value is valid only if Bit 8 is set, otherwise it MUST be ignored

Bit 8 : Antenna Select - if this bit is reset then the antenna selection is left to the algorithm on the WTP. If this bit is set, then the Antenna Number field indicates the antenna that should be used for transmit and receive.

Bits 9-31 : Reserved

#### 6.1.3.1.12 Number of BSSIDs

This element indicates the number of BSSIDs supported by the WLAN interface. This element is optional in the capabilities part of the registration request message and if it is absent, then the number of BSSIDs is set to 1. This element appears once for each instance of a WLAN interface present in the WTP.

Element ID : 11

Length : 1

Value : The number of BSSIDs that the WLAN interface is capable of supporting.

#### 6.1.3.1.13 BSSID Index

This element is used when sending configuration or status specific to a certain BSSID in the WTP.

Element ID : 12

Length : 1

Valid values are from 0 to (Number of BSSIDs -1)

#### 6.1.3.1.14 ESSID

This element is used in configuration and status messages to either configure the ESSID on a certain BSSID or report the current operating value.

Element ID : 13

Length : Variable, between 0 and 32 both inclusive

Value : Variable, contains ASCII characters.

There is no default value for this parameter.

#### 6.1.3.1.15 ESSID Announcement Policy

This element is used in configuration and status messages to control the announcement of the ESSID in 802.11 beacons. For the local MAC modes of operation, this field is also used to control whether the WTP should respond to probe requests that have a NULL ESSID in them.

Element ID : 14

Length : 1

Value : Defined as follows

Bit 0 : ESSID announcement, 0 = Hide ESSID, 1 = Display ESSID in 802.11 beacons. The default value for this bit is 1.

Bit 1 : Probe Response policy, 0 = Respond to probe requests that contain a NULL ESSID, 1 = Respond only to probe requests that match the configured ESSID. The default value for this bit is 0.

Bit 2-7 : Reserved

#### 6.1.3.1.16 Beacon Interval

This element is used to configure the beacon interval on a BSSID on the WTP.

Element ID : 15

Length : 2

Value : Valid values for the beacon interval as allowed by IEEE 802.11

The default value for this parameter is 100.

#### 6.1.3.1.17 DTIM period

This element is used to configure the DTIM period on a BSSID present on the WTP.

Element ID : 16

Length : 2

Value : Valid values for the DTIM period as allowed by IEEE 802.11

The default value for this parameter is 1.

#### 6.1.3.1.18 Basic Rates

Configure or report the configured set of basic rates.

Element ID : 17

Length : 4

Value : Each of the bits in the following list is interpreted as follows. If the bit is set, then that particular rate is to be configured as a basic rate. If the bit is reset, then the rate is not to be configured as a basic rate.

Bit 0 : 1 Mbps

Bit 1 : 2 Mbps

Bit 2 : 5.5 Mbps

Bit 3 : 11 Mbps

Bit 4 : 6 Mbps

Bit 5 : 9 Mbps

Bit 6 : 12 Mbps

Bit 7 : 18 Mbps

Bit 8 : 24 Mbps

Bit 9 : 36 Mbps

Bit 10 : 48 Mbps

Bit 11 : 54 Mbps

Bits 12-31 : Reserved

#### 6.1.3.1.19 Supported Rates

Configure or report the configured set of basic rates.

Element ID : 18

Length : 4

Value : Each of the bits in the following list is interpreted as follows. If the bit is set, then that particular rate is to be configured as a supported rate. If the bit is reset, then the rate is not to be configured as a supported rate.

Bit 0 : 1 Mbps

Bit 1 : 2 Mbps

Bit 2 : 5.5 Mbps

Bit 3 : 11 Mbps

Bit 4 : 6 Mbps

Bit 5 : 9 Mbps

Bit 6 : 12 Mbps

Bit 7 : 18 Mbps

Bit 8 : 24 Mbps

Bit 9 : 36 Mbps

Bit 10 : 48 Mbps

Bit 11 : 54 Mbps

Bits 12-31 : Reserved

#### 6.1.3.1.20 802.11 Retry Count

This element is used to configure long and short retries for each BSSID present on the WTP.

Element ID : 19

Length : 2

Value : as follows

Bits 0-7 : Short retry count, default value is 3.

Bits 8-15 : Long retry count, default value is 3.

#### 6.1.3.1.21 Fragmentation Threshold

This element is used to configure the fragmentation threshold on a BSSID present on the WTP.

Element ID : 20

Length : 2

Value : Valid values for the fragmentation threshold as allowed by IEEE 802.11.

The default value for this parameter is 2346.

#### 6.1.3.1.22 RTS Threshold

This element is used to configure the RTS threshold on a BSSID present on the WTP.

Element ID : 21

Length : 2

Value : Valid values for RTS threshold as allowed by IEEE 802.11.

The default value for this parameter is 2346.

#### 6.1.3.1.23 Short/Long Preamble

This element is used to configure the preamble type used for transmission in 802.11b mode.

Element ID : 22

Length : 1

Value : defined as follows

0 : Disable Short preamble

1 : Enable Short preamble

2-255 : Reserved

The default value for this parameter is 0.

#### 6.1.3.1.24 802.1Q Tag

This element is used to configure the tagging of packets belonging to a particular SSID when transfered between the AC and the WTP in CAPWAP modes 2-3, or before the WTP bridges the 802.3 frame to its wired interface when operating in CAPWAP mode 1.

Element ID : 23

Length : 2

Value : 802.1Q tag

If this element is absent in the configuration, then the WTP MUST assume that no tagging is required and should expect to receive untagged frames on frames destined towards the wireless interface.

#### 6.1.3.1.25 SLAPP Registration ID

A successful registration response from an AC to a WTP MUST contain this element. It is used in messages between the WTP and the AC on all other messages during the duration for which the registration is active.

Element ID : 24

Length : 4

Value : a 32-bit unsigned number allocated by the AC

#### 6.1.3.1.26 WTP Name

The AC uses this element to assign a string of ASCII characters to the WTP.

Element ID : 25

Length : Variable, between 0 and 64 both inclusive

Value : A variable length string of ASCII characters

#### 6.1.3.1.27 Event Filter

The AC uses this element to assign importance to events, enable or disable notification, and to configure the global event notification policy. When the Event Identifier is 0, this element serves as a global notification policy message. The bitmap indicates the types of events that require the WTP to generate a notification. When the Event Identifier is non-zero, this element is used to configure a specific event for notification and its importance level. The importance level is specified by setting exactly one bit in the bitmap. If none of the bits are set in the bitmap, the element should be interpreted as a cancellation request. The WTP should stop sending notifications for the corresponding event specified in the Element Identifier.

Element ID : 26

Length : 4

Value : defined as follows

Bits 0 - 15: Event Identifier

Bit 16: Fatal - The system is not usable

Bit 17: Alert - Immediate action is required

Bit 18: Critical

Bit 19: Error

Bit 20: Warning

Bit 21: Notification

Bit 22: Informational

Bit 23: Debug

Bits 24 - 31: Reserved

#### 6.1.3.1.28 Radio Mode

The AC uses this element to indicate the mode of operation for the radio for each WLAN interface.

Element ID : 27

Length : 1

Value : The following are valid values.

0 : Radio is disabled

1 : Radio is enabled

2-255 : Reserved

#### 6.1.3.1.29 IEEE 802.11e Element

The AC uses this element to configure 802.11e functions at the WTP

Element ID : 28

Length : 4

Value : A bitmap as follows

Bit 0 : WMM

Bit 1 : WMM-SA

Bit 2 : U-APSD

Bits 3-32 : Reserved

#### 6.1.3.1.30 Configuration Statistics

This element defines the statistics relating to configuration and registration events as seen by the WTP.

Element ID : 29

Length : 32

Value : The value is as follows.

* Configuration Requests : 4 octets - Number of configuration request messages sent by the WTP since the last reboot or reset of the counters

* Configuration Responses : 4 octets

* Configuration Updates : 4 octets

* Configuration ACKs : 4 octets

* Registration Requests : 4 octets

* Registration Responses : 4 octets

* De-registration requests : 4 octets

* De-registration responses : 4 octets

#### 6.1.3.1.31 Transmit Frame Counters

This information element contains a set of counters relating to the transmit side of the wireless link at the WTP. These counters apply to either a BSS or an Access Category (if WMM is enabled).

Element ID : 30

Length : 112 octets

Value : The value of this element is defined as follows.

* Total received from the network : 4 octets

* Successfully transmitted frames (total) : 4 octets

* Successfully transmitted 802.11 Mgmt frames : 4 octets

* Successfully transmitted 802.11 Data frames : 4 octets

* Transmitted 802.11 Control frames : 4 octets

* Frames that reached max-retry limit : 4 octets

* Transmitted frames with 1 retry attempt : 4 octets

* Transmitted frames with 2 retry attempts : 4 octets

* Transmitted frames with more than 2 retry attempts : 4 octets

* Frames transmitted at each 802.11 PHY rate : 12*4 octets - The counters indicate the number of frames at each of the following rates respectively : 1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54 Mbps

* Total frame dropped : 4 octets

* Frames dropped due to insufficient resources : 4 octets

* Frames dropped due to power-save timeouts : 4 octets

* Frames dropped due to other reasons : 4 octets

* Fragments Transmitted : 4 octets

* Fragments dropped : 4 octets

* Power-save multicast frames : 4 octets

* Power-save unicast frames : 4 octets

#### 6.1.3.1.32 Received Frame Counters

This information element includes all statistics related to the reception of the frames by WTP. These counters apply to either a BSS or an Access Category (if WMM is enabled).

Element ID : 31

Length : 108 octets

Value : The value of this element is defined as follows.

* Total Frames received : 4 octets

* Frames with the retry bit set : 4 octets

* 802.11 Data frames received : 4 octets

* 802.11 Mgmt frames received : 4 octets

* 802.11 Control frames received : 4 octets

* CRC errors : 4 octets

* PHY errors : 4 octets

* Total Fragments received : 4 octets

* Reassembled frames : 4 octets

* Reassembly failures : 4 octets

* Successful Decryption : 4 octets

* Decryption failures : 4 octets

* Rate statistics : 48 octets - the number of frames received at each of the 802.11 PHY rates respectively - 1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 49, 54 Mbps

* Total frames dropped : 4 octets

* Frames dropped due to insufficient resources : 4 octets

* Frames dropped due to other reasons : 4 octets

#### 6.1.3.1.33 Association Statistics

This element includes information about the current stations associated with the BSS.

Element ID : 32

Length : Variable

Value : The value is defined as follows.

* Total association requests : 4 octets

* Total associations accepted : 4 octets

* Total associations rejected : 4 octets

* Current associations : 4 octets

* For each associated station,

  + Station MAC address : 6 octets

  + Power save state : 1 octet

  + Current Tx rate : 1 octet

  + Rate of last packet : 1 octet

  + Preamble type : 1 octet

  + WMM/U-APSD state : ?? octet

#### 6.1.3.1.34 Status Element

The status IE is included in the status response message sent by the WTP to the AC. It contains a set of fields that are used to indicate the status of various states at the WTP or each BSS configured in the WTP.

Element ID : 33

Length : 2 octets

Value : The value is defined as follows.

ERP element, if applicable. If not applicable, then this field MUST be set to 0

Noise Floor : 1 octet

#### 6.1.3.1.35 Event Configuration

This element is used by the AC to configure the set of events that it wants to be notified by the WTP.

Element ID : 34

Length : 4 octets

Value : The value is defined as follows.

* Radar Detection - 1 octet
  + Bit 0 : 1 = notify on detecting radar interference, 0 otherwise
  + Bit 1 : 1 = notify of channel change due to radar interference, 0 otherwise
  + All other bits are reserved.
* Excessive Retry Event - 1 octet. Number of successive frames that have not been acknowledged by a client. A value of 0 disables notification.
* Noise Floor Threshold - 1 octet. Defines the threshold above which an event would be generated by the WTP.
* 802.11 Management and Action Frame Notification - 1 octet.
  + Bit 0 : If set, notify AC of probe requests from stations (please use with caution). If reset, then no probe response notification is needed.
  + Bit 1 : If set, the WTP should notify the AC of all other management frames from stations.
  + All other bits are reserved.

#### 6.1.3.1.36 Radar Detection Event

This element is used by the WTP to notify the AC of detection of radar interference and any channel changes as a result of this detection.

Element ID : 35

Length : 10 octets

Value : defined as follows.

BSSID : 6 octets. The BSSID of the WLAN interface that detected the radar interference.

Channel : 2 octets. The channel on which radar interference was detected.

New Channel : 2 octets. The new channel to which the WTP moved as a result of detection of radar interference.

#### 6.1.3.1.37 Excessive Retry Event

This element is used by the WTP to indicate excessive retry events on transmission to an associated station.

Element ID : 36

Length : 14 octets

Value : defined as follows.

Station MAC : 6 octets

Associated BSSID : 6 octets

Length of last burst of excessive retries : 2 octets.

#### 6.1.3.1.38 Noise Floor Event

This element is used by the WTP to notify the AC of the current noise floor at one of the WLAN interfaces exceeding the configured noise floor threshold.

Element ID : 37

Length : 10 octets

Value : defined as follows.

BSSID : 6 octets

Current Channel : 2 octets

Current Noise Floor : 2 octets

#### 6.1.3.1.39 Raw 802.11 Frame

This element provides a generic capability for either a WTP or an AC to send a raw 802.11 frame to the other party. For example, it can be used to notify the AC of station association/disassociation events in the case of local MAC architectures.

Element ID : 252

Length : Variable

Value : a raw 802.11 frame

#### 6.1.3.1.40 Vendor-Specific Element

This element is used to transfer vendor-specific information between the WTP and the AC.

Element ID : 253

Length : Variable, > 3

Value : This variable length element starts with a 3-octet OUI, followed by a series of octets that are specific to the vendor represented by the OUI.

#### 6.1.3.1.41 Recursion Element

This element type can be used to recursively define a variable length element that should be interpreted as a series of other elements defined in this section. It can be used to bound a set of elements as a unit.

Element ID : 254

Length : Variable

Value : A variable length element that contains a set of one or more elements defined in this section.

#### 6.1.3.1.42 Pad Element

This is a generic element type which can be used to pad the packets, if necessary.

Element ID : 255

Length : Variable

Value : A variable length element that MUST be filled with all 0s at the source and MUST be ignored at the destination.

### 6.1.3.2 SLAPP 802.11 Control Protocol Messages

#### 6.1.3.2.1 Registration Request

At the start of the SLAPP 802.11 control protocol, the WTP sends a registration request to the AC that it authenticated with. The registration request carries a list of information elements indicating the WTP's capabilities to the AC. The message starts with the SLAPP 802.11 control protocol header (Figure 8) with a SLAPP control protocol message type of 1.

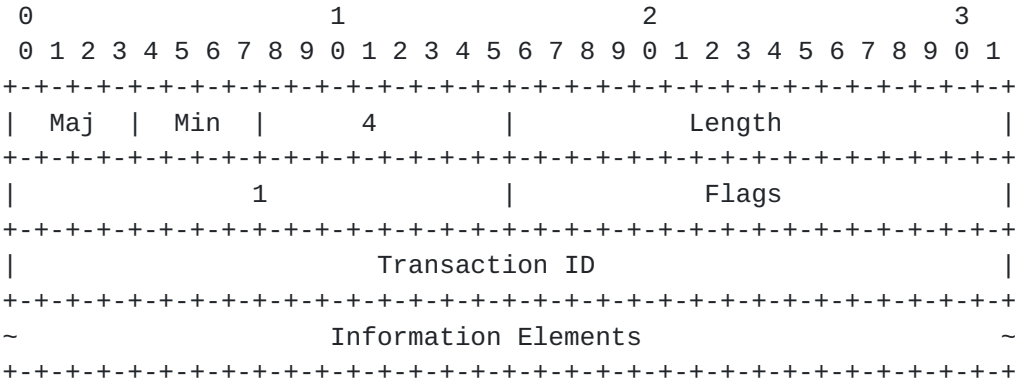



Figure 9: SLAPP 802.11 Registration Request

Flags : Reserved

Transaction ID : a 32-bit random number chosen by the WTP at the start of a new registration phase. This number is used in the registration response by the AC to match the response to the corresponding request.

The following information elements are mandatory in the capabilities exchange.

1 : CAPWAP mode

2 : Number of WLAN interfaces

For each WLAN interface:

7 : 802.11 PHY mode and Channel Information

8 : Cryptographic Capability

9 : Other 802.11 standards support

The following information elements may be optionally included in the

registration request.

For each WLAN interface:

4 : WLAN Interface HW Vendor ID

5 : WLAN Interface Type ID

6 : Regulatory Domain

10 : Antenna Information Element

11 : Number of BSSIDs

253 : Vendor-specific Element

#### 6.1.3.2.2 Registration Response

Upon receiving a registration request, the AC may either chose to accept the WTP

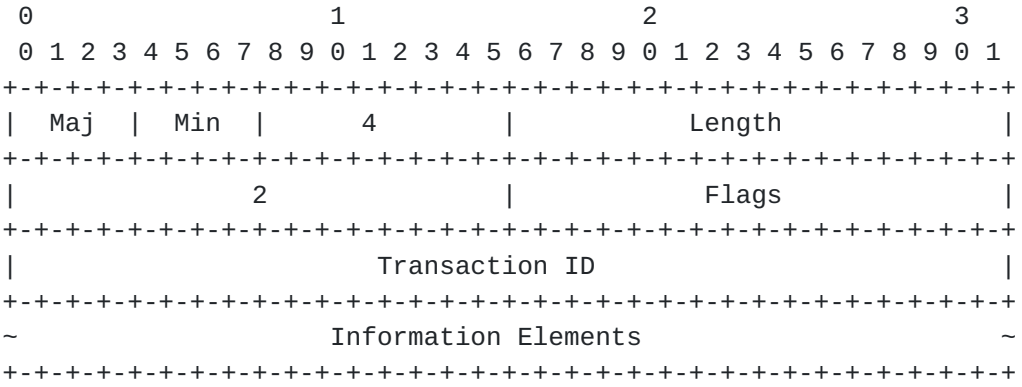



Figure 10: SLAPP 802.11 Registration Response

Flags :

Bit 0 : indicates the status of the transaction, 0 = successful response from the AC, 1 = the registration request is being rejected by the AC.

Bits 1-7 : Reserved

Bits 8-15 : If bit 0 = 1 (i.e., the registration request is being rejected by the AC), then this field contains a reason code. Otherwise, these bits are currently set to 0. The following reason codes are currently defined.

0 : Reserved

1 : Unspecified reason

2 : Unable to handle more WTPs

3 : Incompatible capabilities

4-255 : Reserved

Transaction ID : a 32-bit random number chosen by the WTP at the start of a new registration phase. This number is used in the registration response by the AC to match the response to the corresponding request.

The following information elements are mandatory if the transaction is successful.

1 : CAPWAP mode - the mode that the AC chooses from among the list of supported modes sent by the WTP in the registration request.

24 : SLAPP registration ID

#### 6.1.3.2.3 De-registration Request

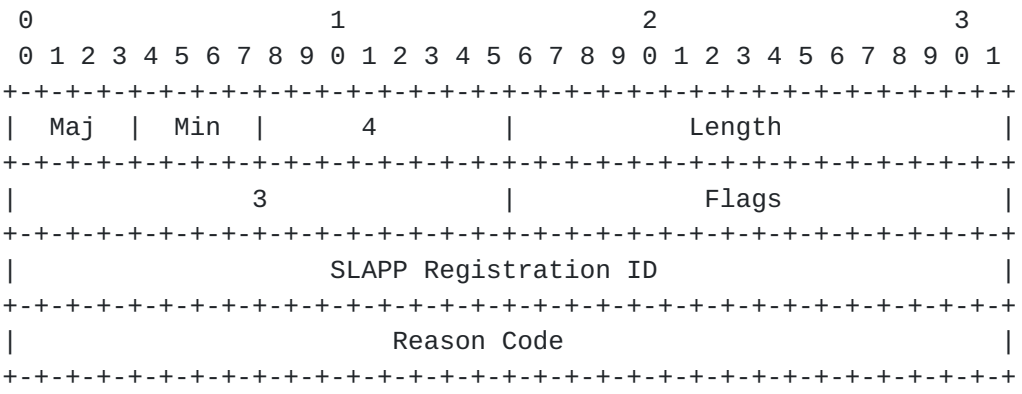



Figure 11: SLAPP 802.11 Deregistration Request

Flags : Reserved

SLAPP Registration ID : The registration ID assigned by the AC upon successful registration

Reason Code : The following are valid values.

0 : Unspecified reason

1 : The device that is the source of the frame is going down

All other values are reserved

#### 6.1.3.2.4 De-registration Response

The De-registration response is a simple ACK from the recipient of the corresponding de-registration request.

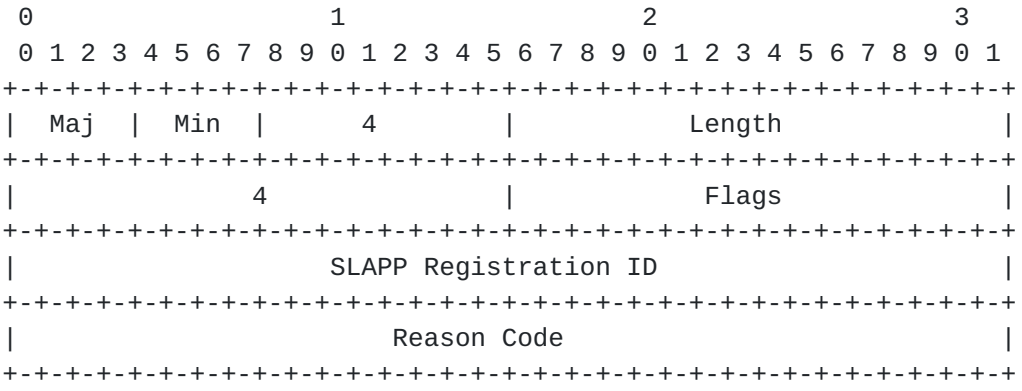



Figure 12: SLAPP 802.11 Deregistration Response

Flags : Reserved

SLAPP Registration ID : The registration ID assigned by the AC upon successful registration

Reason Code : The same reason code used in the corresponding request

#### 6.1.3.2.5 Configuration Request

The configuration request message is used by the WTP to request a set of configuration for each BSS that the AC wishes to configure at the WTP.

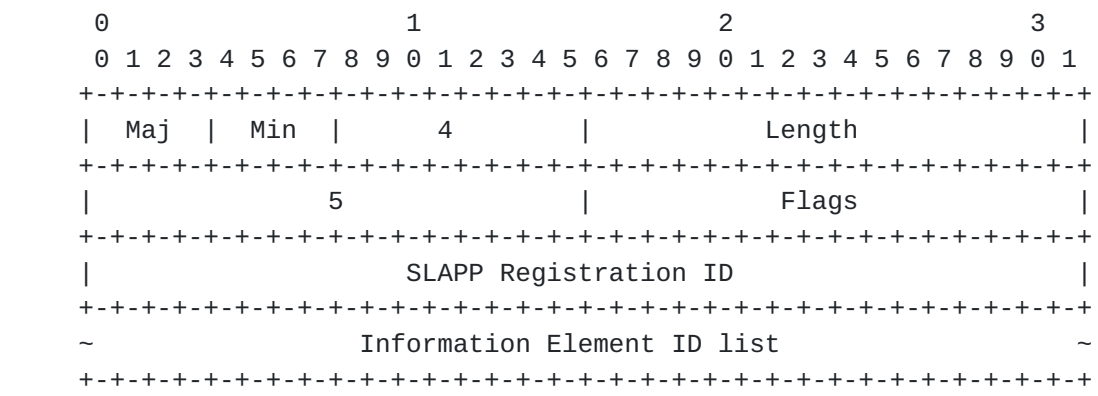



Figure 13: SLAPP 802.11 Configuration Request

The Information Element ID list field contains the list of IEs that the WTP is interested in obtaining configuration information for.

#### 6.1.3.2.6 Configuration Response

The Configuration response message is used by the AC to respond to a configuration request by the WTP.

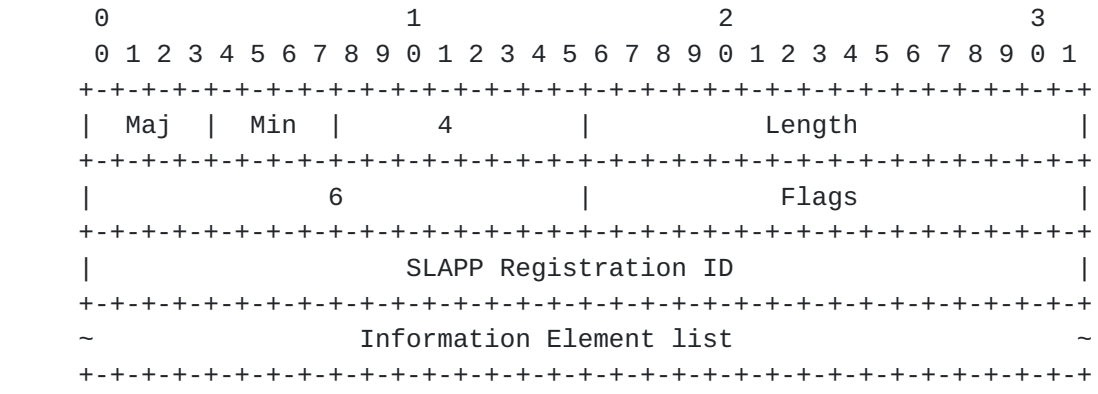



Figure 14: SLAPP 802.11 Configuration Response

The following information elements are mandatory in the configuration response.

01: CAPWAP mode

For each WLAN interface:

03: WLAN Interface Index

27: Radio Mode

07: 802.11 PHY mode and Channel Selection

For each BSSID:

12: BSSID Index

13: ESSID

08: Cryptographic Selection

The following information elements may be optionally included in the configuration response.

10: Antenna Information Element

25: WTP Name

For each WLAN interface:

For each BSSID:

14: ESSID Announcement Policy

15: Beacon Interval

16: DTIM Period

17: Basic Rates

18: Supported rates

19: Retry Count

20: Fragmentation Threshold

21: RTS Threshold

22: Short/Long Preamble

23: 802.1Q Tag

253: Vendor specific element

If any of the optional IEs is absent in the configuration response message, then their default values are applied by the WTP.

#### 6.1.3.2.7 Configuration Update

The Configuration Update message is initiated by the AC to push modified or updated configuration to the WTP. It has a format similar to that of the configuration response message defined above.

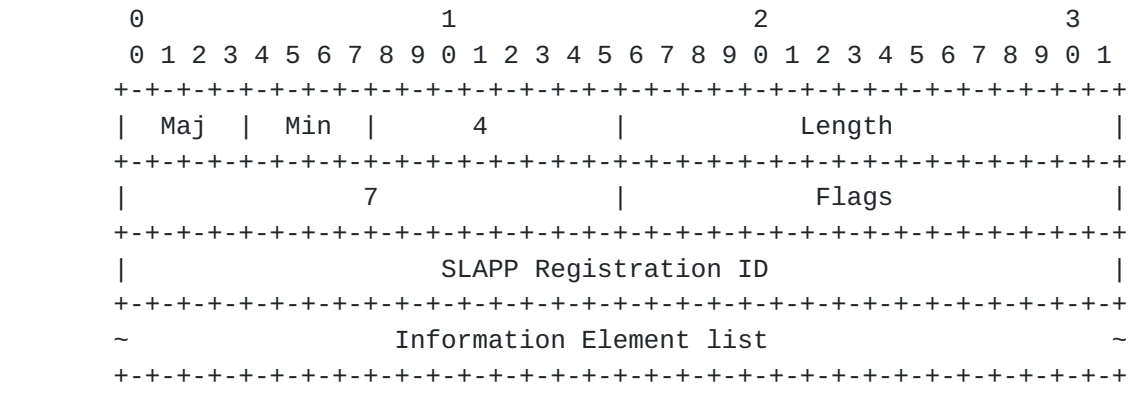



Figure 15: SLAPP 802.11 Configuration Update

The list of mandatory and optional IEs for the configuration update message is the same as that for the configuration response message.

#### 6.1.3.2.8 Configuration Acknowledgment

The Configuration Acknowledgment message is used by the WTP to inform the AC whether it has accepted the prior configuration update or configuration response message. The WTP can reject the configuration sent by the AC, in which case it MUST return to the discovery state.

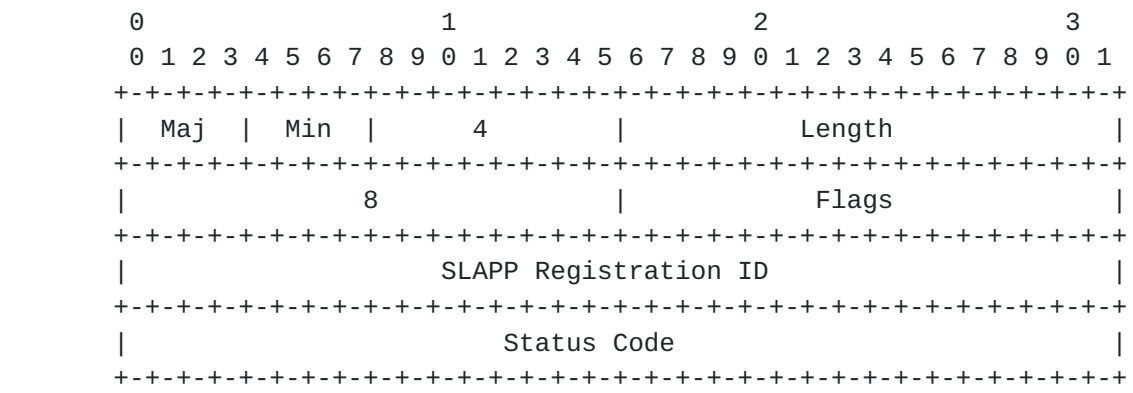



Figure 16: SLAPP 802.11 Configuration ACK

The Status Code field contains one of the following values:

0 : Success - The WTP accepts that the configuration pushed by the AC and has applied it.

1 : Failure - The WTP did not accept the configuration pushed by the AC and MUST be de-registered at the AC.

#### 6.1.3.2.9 Status Request

The Status request message is used by the AC to request the configuration and operational status from the WTP.

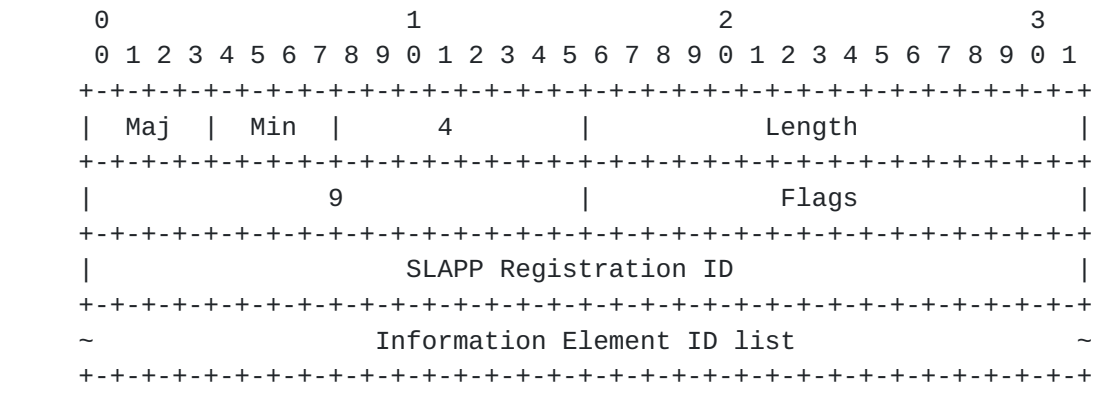



Figure 17: SLAPP 802.11 Status Request

The Information Element ID list contains the list of IEs that the AC requests status for.

#### 6.1.3.2.10 Status Response

The Status response message is used by the WTP to respond to a status request from the AC.

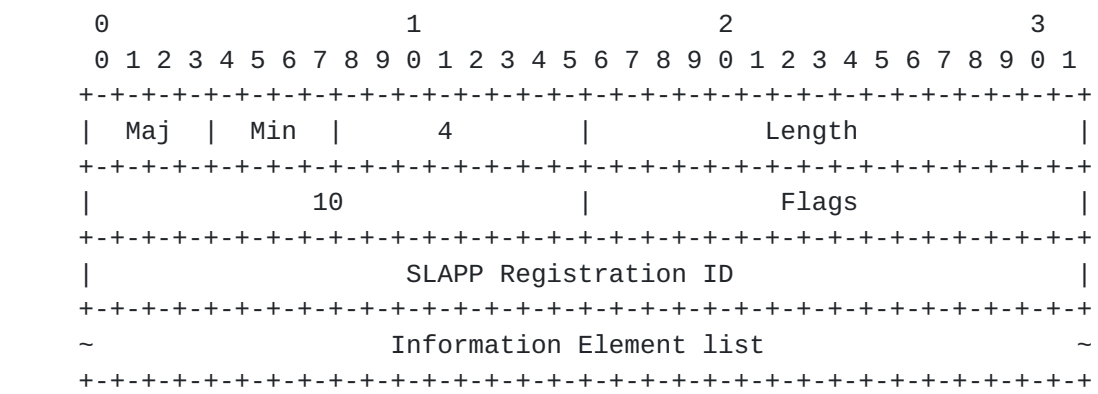



Figure 18: SLAPP 802.11 Status Response

The Flags field contains one of the following values:

Bit 0 : If set, Unknown AC or SLAPP registration ID. If this bit is reset, then this indicates a successful response.

Bit 1 : If set, WTP indicates that it has not been configured yet, otherwise the WTP is in a configured state.

All other values are reserved

The Status IE is mandatory in a Status Response message.

#### 6.1.3.2.11 Statistics Request

The Statistics request message is used by the AC to request statistics information from the WTP.

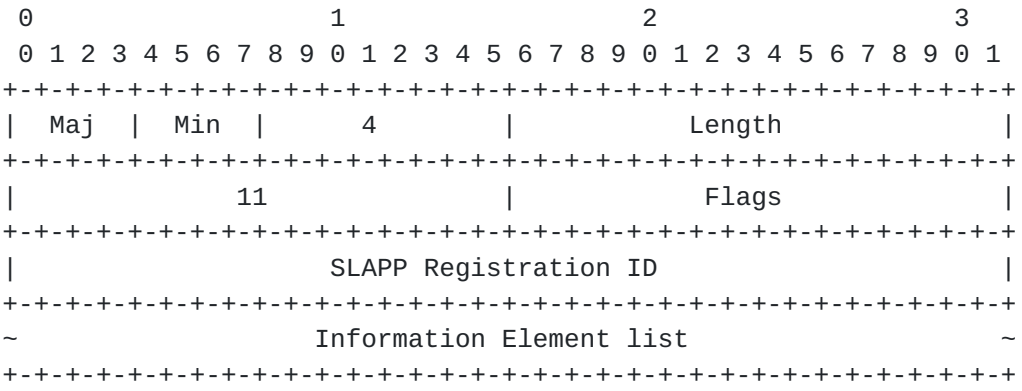



Figure 19: SLAPP 802.11 Statistics Request

The Flags field contains the following bits:

Bit 0 : If set to 1, then the WTP should reset the counters after sending the statistics response message.

All other bits are reserved and MUST be set to 0 by the source and ignored by the destination.

#### 6.1.3.2.12 Statistics Response

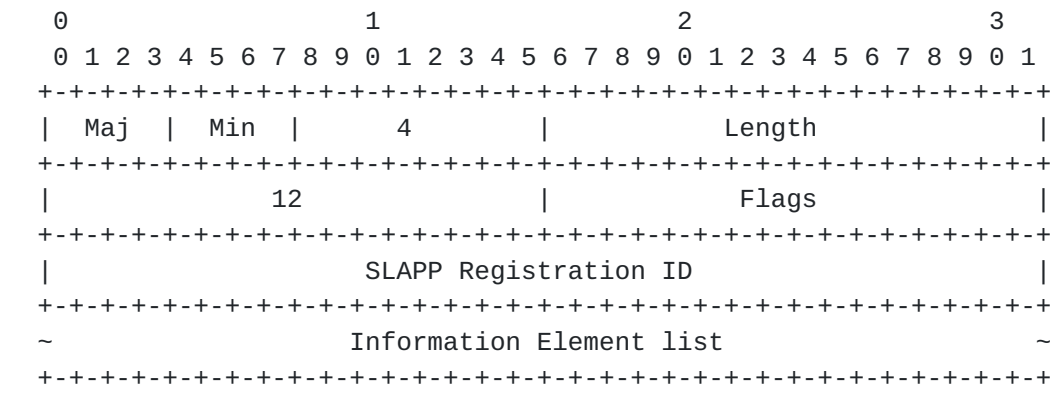



Figure 20: SLAPP 802.11 Statistics Response

The Flags field contains the following bits:

Bit 0 : if set then the counters have been reset as requested by the AC.

Bit 1 : If set, then the WTP has encountered a statistics request from either an unknown AC or with an unknown SLAPP registration ID.

Bit 2 : If set, WTP indicates that it has not been configured yet, otherwise the WTP is in a configured state.

All other bits are reserved.

### 6.1.3.2.13 Keepalive

The keepalive messages can be initiated by either the WTP or the AC. It is used to probe the availability of the other party and the path between them. The initial message is termed the keepalive request, while the response to that message is termed the keepalive response.

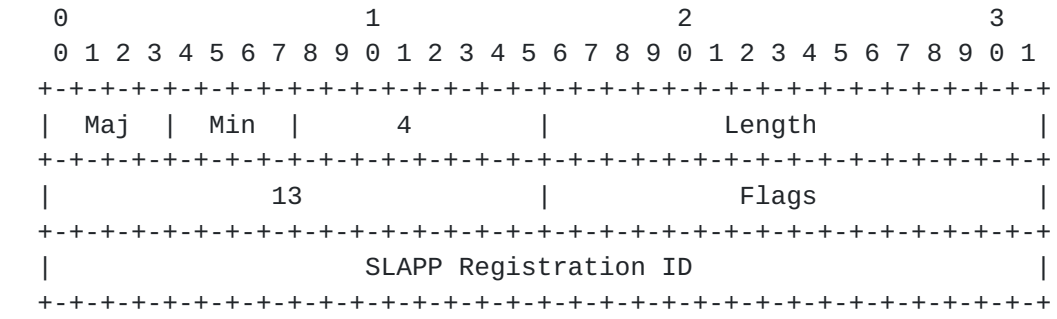



Figure 21: SLAPP Keepalive Packet

The Flags field has the following values:

Bit 0 : Set to 0 in a keepalive request message, set to 1 in a keepalive response message.

Bit 1 : Set to 0 in a keepalive request message, set to 1 in a keepalive response message if the initiator of the keepalive request is unknown or the SLAPP registration ID is incorrect, and set to 0 otherwise.

All other bits are reserved and must be set to 0 by the source and ignored at the destination.

#### 6.1.3.2.14 Key Configuration

In CAPWAP mode 5, the 802.11 crypto functions are performed at the AC. So there is no need for the AC to send PTKs/GTKs to the WTP. When one of CAPWAP Modes 1-4 has been negotiated between the AC and WTP it is necessary for the AC to send both unicast and broadcast/ multicast keys to the WTP. This is accomplished after the 802.1x authenticator (which resides on the AC) has successfully authenticated the supplicant. Key configuration requests are differentiated-- unicast or broadcast-- by setting or clearing the high-order bit of the "Flags" field. The setting of this bit determine the contents of the Key configuration request following the SLAPP Registration ID.

##### 6.1.3.2.14.1 Unicast Key Configuration Request

The Unicast Key Configuration Request is used by the AC to inform the WTP of the key to use when protecting unicast frames to and from a specified supplicant.

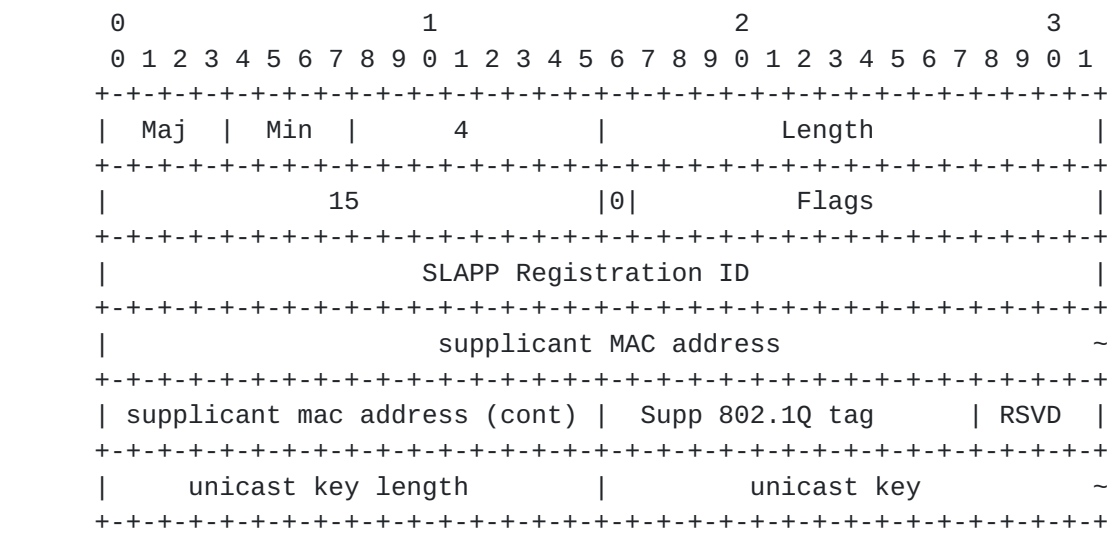



Figure 22

Note the high order bit of the "Flags" field is cleared to indicate a unicast key is being sent. The 802.1Q tag field is used to indicate to the WTP which VLAN this supplicant is in and which broadcast/ multicast key to use when communicating to it with broadcast/ multicast frames.

### 6.1.3.2.14.2 Broadcast/Multicast Key Configuration Request

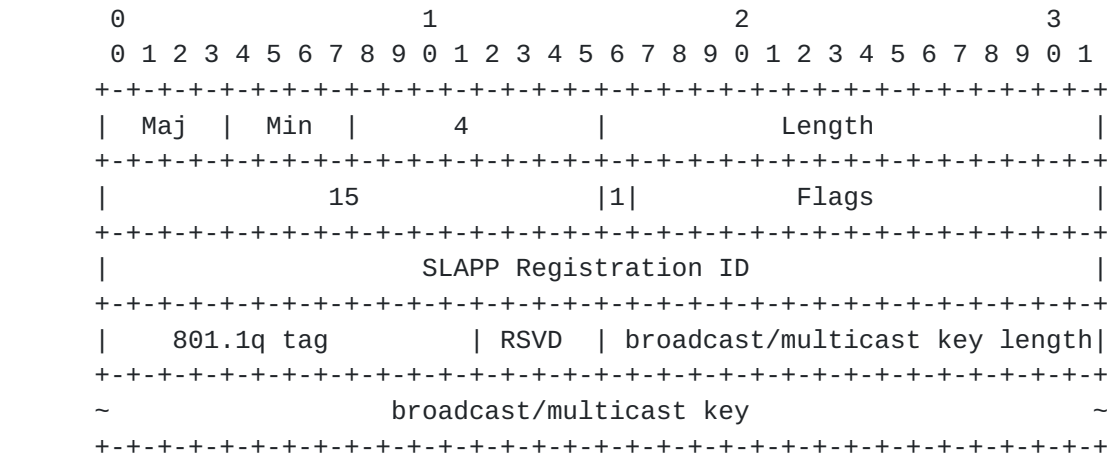



Figure 23

Note the high-order bit of the "Flags" field is set indicating a broadcast/multicast key is being sent. The bits marked "RSVD" are reserved and MUST be set to zero by the AC and ignored by the WTP.

#### 6.1.3.2.14.3 Unicast Key Configuration Response

The WTP acknowledges receipt of a Unicast Key Configuration Request by sending a Unicast Key Configuration Response. This response mirrors the request but does not send back the key length or the key itself. (The RSVD bits are returned for alignment purposes and MUST be set to zero by the WTP and ignored by the AC).

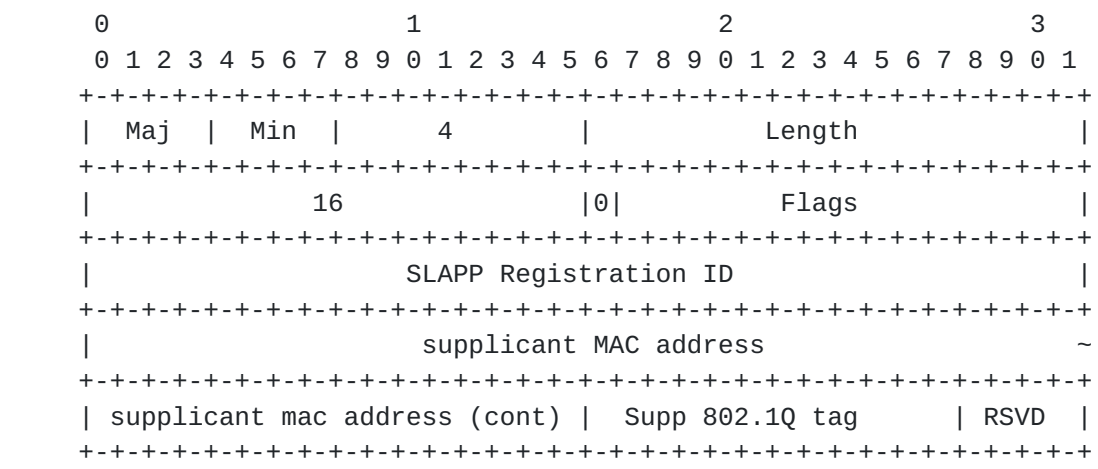



Figure 24

#### 6.1.3.2.14.4 Multicast Key Configuration Response

The WTP acknowledges receipt of a Multicast Key Configuration Request by sending a Multicast Key Configuration Response. This response mirrors the request but does not send back the key length or the key itself. (The RSVD bits are returned for alignment purposes and MUST be set to zero by the WTP and ignored by the AC).

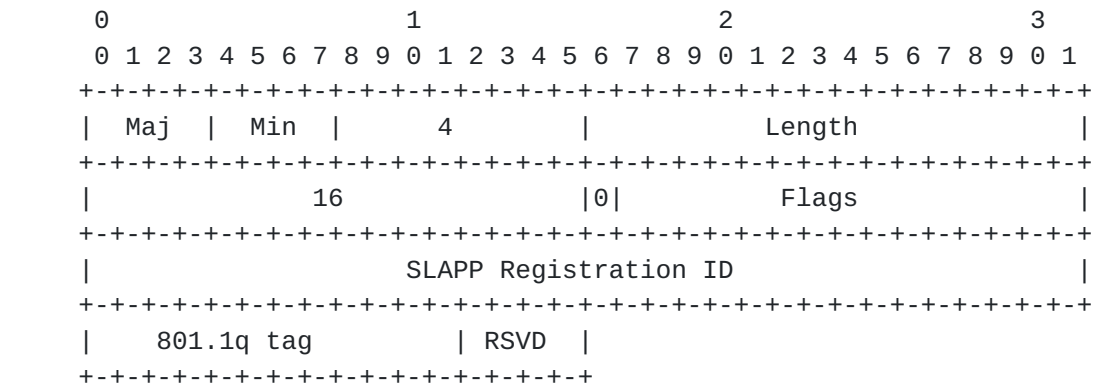



Figure 25

#### 6.1.3.3 Monitoring and Statistics

An AC may want to periodically monitor the health of a WTP, collect the necessary information for diagnostics, and get notifications on pre-defined events at the WTP that may be of interest. This section defines a set of WTP statistics and events and describes the process of collecting statistics from WTPs and configuring the event notification mechanism at the WTP. It is beyond the scope of this document to describe what should/could be done with the collected information.

##### 6.1.3.3.1 Statistics Collection Procedure

The simple statistics collection procedure defined here does not require the WTP to maintain any timers or any similar mechanisms. A WTP is responsible only for maintaining the statistics defined in Information Elements 29, 30, 31, and 32. The WTP must also respond to a statistics request message from the AC by delivering the appropriate statistics to the AC using a statistics response message. For example, if an AC is interested in gathering periodic statistics about some specific statistics, it is the responsibility of the AC to poll the WTP at the appropriate intervals.

##### 6.1.3.3.2 Events Procedure

The event notification process includes the following: 1) Event Registration: The registration of events of interest at the WTP by the AC and 2) Notification: The communication of event-related information by the WTP to AC whenever the conditions for a specific registered event has occurred. The set of events supported by a WTP and the event-specific parameters that may be configured as part of a event registration are given in Section 6.1.3.3.3.

##### 6.1.3.3.3 WTP Events

This section defines a set of WTP events along with the event-specific parameters that may be configured by ACs and the event-related information that should be delivered to the ACs by WTPs when the conditions for a particular configured event has occurred.

Radar Detection Event: Configure whether the AC is interested in receiving a notification whenever a radar event is detected. The WTP may notify the AC about the type of radar interference and the new channel that the WTP has moved to as a result, if any, using the Radar Detection Event Element (element ID : 35).

Excessive Retry Event: Configure the number of consecutive transmission failures before a notification is generated. The WTP

may notify the MAC address of the STA and the number of consecutive unacknowledged frames so far using the Excessive Retry Event Element (element ID : 36).

Noise Floor Event: Configure the noise floor threshold above which an event notification would be generated by the WTP. The WTP may notify the AC with the most recent measured noise floor that exceeded the configured threshold using the Noise Floor Event Element (element ID : 37).

De-authentication Event: Configure whether the AC is interested in receiving a notification whenever a station has been de-authenticated by WTP. The WTP may notify the AC with the MAC address of the STA along with a reason code (inactivity, etc).

Association Event: Needed in local MAC architecture

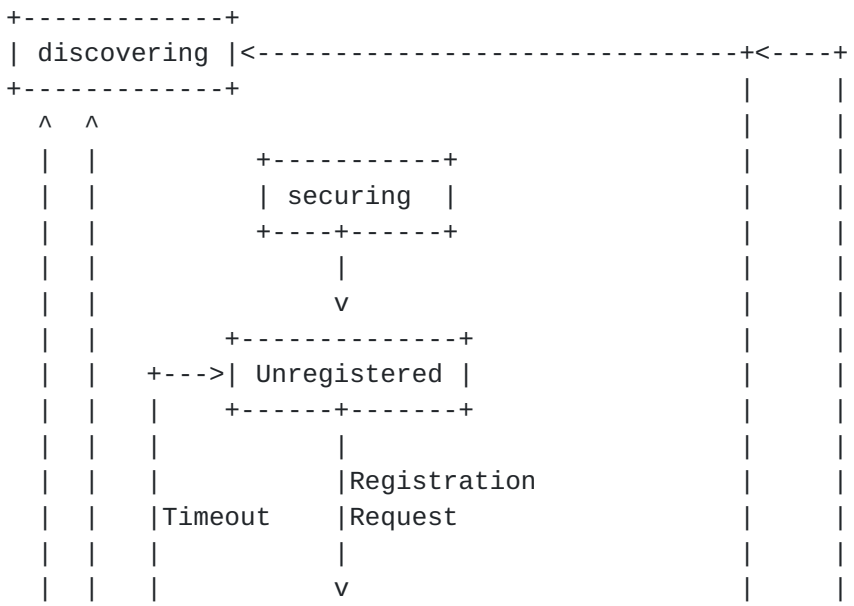
Disassociation Event: Needed in local MAC architecture

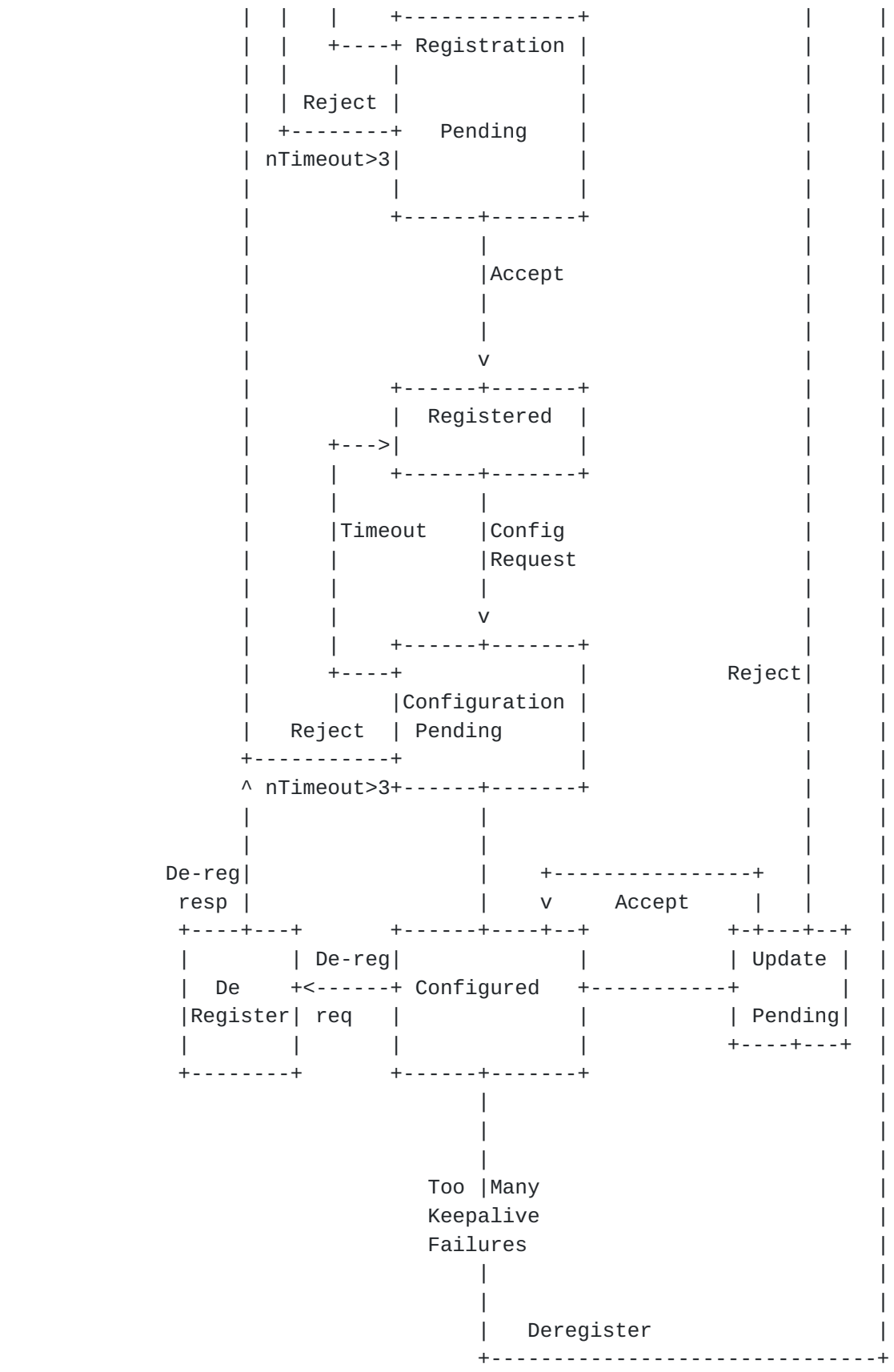
### 6.1.4 Protocol Operation

The SLAPP 802.11 control protocol operation is described in this section.

#### 6.1.4.1 SLAPP 802.11 Control Protocol State Machine

##### 6.1.4.1.1 At the WTP




Registration
Reject
Pending
nTimeout>3
Accept
Registered
Timeout
Config
Request
Reject
Configuration
Pending
Reject
nTimeout>3
De-reg
resp
Accept
De-reg
De
Register
req
Configured
Update
Pending
Too
Many
Keepalive
Failures
Deregister

In Configured and/or Registered states, respond to
Status Requests, Statistics Requests, Keepalives, Key Config

Figure 26: SLAPP 802.11 Control Protocol at the WTP

#### 6.1.4.1.1.1 State Machine Explanation

Unregistered : The transition into this state is from the securing state (Figure 3). Send registration request message to move to Registration Pending state, set timer for registration response.

Registration Pending : On a registration response from AC, cancel registration timer. If response is successful, move to Registered state. If not, move to discovering state (Figure 3). If timer expires, if nTimeout >3, then move to discovering state. If not, return to Unregistered state.

Registered : Send Configuration request message to AC to move to Configuration Pending state, and set timer for configuration response. In this state, respond to status request, statistics request, and keepalive messages from AC.

Configuration Pending : If configuration response received from AC, cancel configuration response timer. If response is successful and the configuration is acceptable, then send Configuration ACK message to AC, and move to Configured state. If configuration request is rejected or configuration is not acceptable, then send a deregister request to AC and move to discovering. If configuration response timer expires, move to Registered state unless nTimeout >3, in which case move to discovering state.

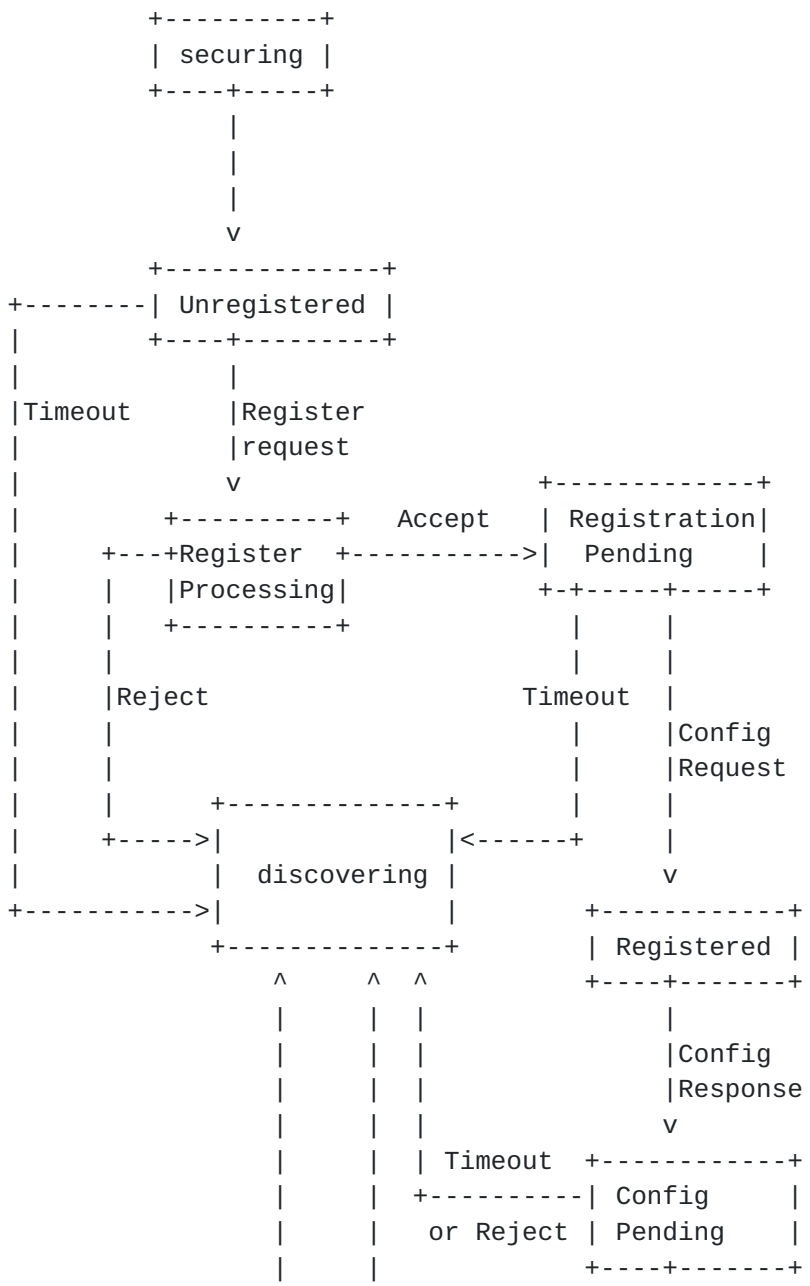
Configured : In the Configured state, the WTP responds to Status request, Statistics Request, and Keepalive messages from the AC. If it receives a de-register request message from the AC, then it sends a de-register response to the AC and moves to the discovering state. If the WTP receives a Configuration Update message, then it moves to the Update Pending state. If it receives too many consecutive Keepalive failures (no responses from the AC to Keepalive requests), then it sends a deregister message to the AC and moves to the discovering state.

Update Pending : In the Update Pending state, the WTP analyzes the configuration information received in the Configuration Update message. If the configuration is found to be acceptable, then it applies the configuration and returns to the Configured state. If the WTP chooses to reject the configuration update, then it sends

a deregister request to the AC and moves to the discovering state.

De-register : From the Configured state, the WTP moves to the De-register state when it receives a deregister request message from the AC. It sends a deregister response to the AC and moves to the discovering state.

#### 6.1.4.1.2 At the AC



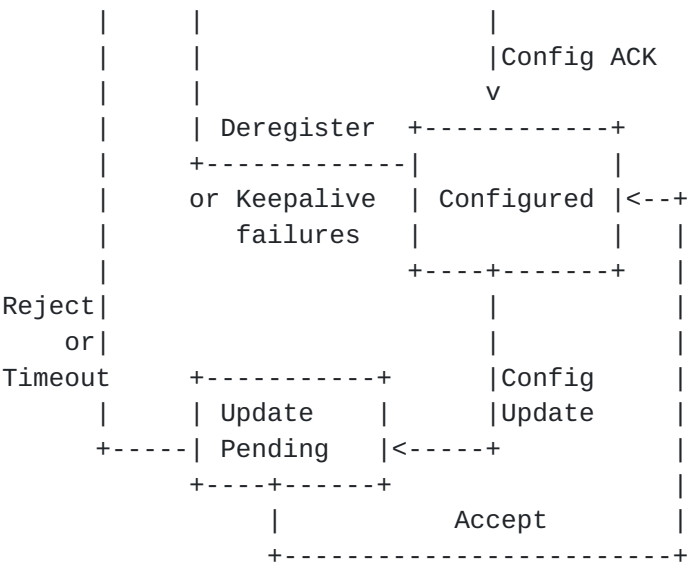



Figure 27: SLAPP 802.11 Control Protocol at the AC

#### 6.1.4.1.2.1 State Machine Explanation

The states "securing" and "discovering" are described in Figure 3.

Unregistered : This state is entered from the securing state described in Figure 3. In this state, the AC is waiting for a registration request message from the WTP. Upon receiving the registration request message, it moves into the Registration Processing state.

Registration Processing : In this state, the AC must determine if it can accept the new WTP or not. If the AC decides to accept the WTP, it must pick a CAPWAP mode to operate in and send a registration response message with a success code and moves to the Registration Pending state. If the AC chooses to reject the current registration request from the WTP, it must send a registration response with a failure code and move to the discovering state.

Registration Pending : If the timer expires before a response from the WTP is received, then the AC destroys the registration state and moves to the discovering state. If a configuration request message is received from the WTP, then the AC moves into the Registered state and processes the configuration request message. It sends a configuration response message to the WTP with the appropriate IEs and moves into the Configuration Pending state.

Configuration Pending : If the timer expires before a response is received from the WTP, then the AC destroys the current registration and moves into the discovering state. If a configuration ACK is received from the WTP, but contains a failure code, then the AC again destroys the registration state and moves into the discovering state. If the configuration ACK from the WTP is successful, then the AC moves to the Configured state.

Configured : In the Configured state, the AC can send status request, statistics request, keepalive, key configuration messages to the WTP. Any response to these messages from the WTP that indicates an unknown SLAPP registration ID or an unknown AC causes the AC to destroy any registration or configuration state and move to the discovering state. From the configured state, the AC can send a configuration update message and move into the Update Pending state. If it receives a deregister request from the WTP then it destroys all current registration and configuration state and moves into the discovering state. If a number of successive keepalive messages go unacknowledged by the WTP, then the AC moves into the discovering state.

Update Pending : When the AC receives an configuration ACK message with a success code, then it returns to the Configured state. If the status code is a failure or if the timer expires before the configuration ACK is received from the WTP, the AC destroys all registaration and configuration state for the WTP and moves into the discovering state.

## 6.2 Image Download Protocol

The Image Download protocol is a control protocol defined in this draft that is generic enough to be agnostic to the underlying technology.

In the image download protocol, the WTP obtains a bootable image from the AC by receiving a series of image transfer packets. Missed image data packets are re-requested by the WTP by sending image data request packets indicating the missing packets.

The image to download is divided into slices of equal size (except for the last slice which can be less than the slice size provided it is also greater than zero). The size of each slice depends on the MTU determined by the DTLS exchange and SHOULD be the realized MTU minus the size of an Image Download Request (Figure 29).

Note that the image download packet and download image request is encapsulated in a DTLS header which secures the image download.

### 6.2.1 Image Download Packet

The format of an image download packet is shown in Figure 28.

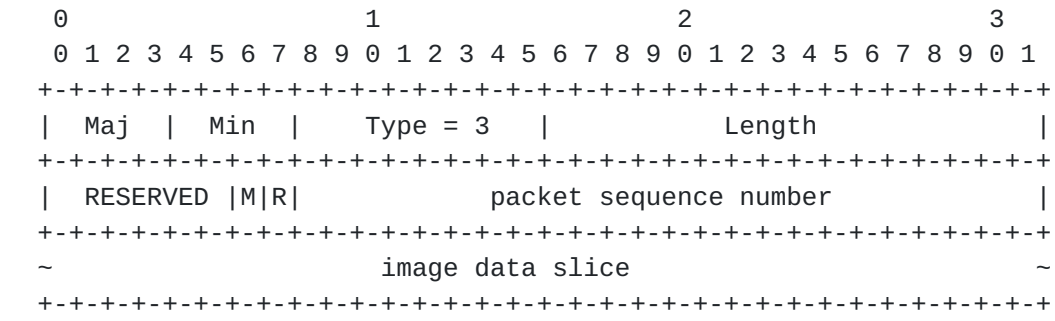



Figure 28: SLAPP Image Download Packet

where:

length: variable

RESERVED: unused in this version of SLAPP, MUST be zero (0) on transmission and ignored upon receipt

M: the "More" bit indicating that the current packet is not the final one

R: the "Request" bit. This bit MUST be set to one (1) when the packet is the response to a request and zero (0) otherwise.

packet sequence number: a monotonically increasing counter which assigns a unique number to each slice of the image

image data slice: a portion of the bootable image.

### 6.2.2 Image Download Request

The format of an image download request is show in Figure 29.

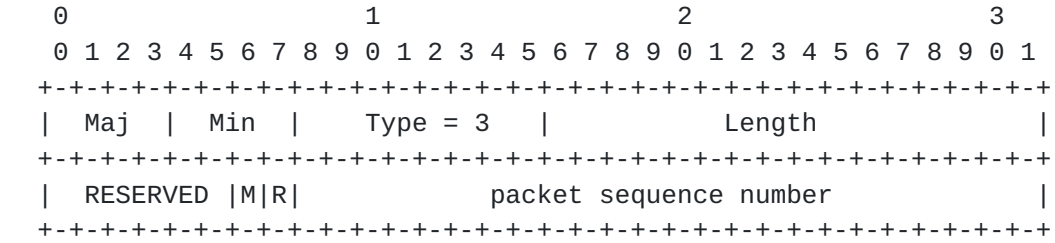



Figure 29: SLAPP Image Download Request Packet

where:

length: eight (8) octets

RESERVED: unused in this version of SLAPP, MUST be zero on transmission and ignored upon receipt

M: the "More" bit. This MUST be equal to the one (1) when negatively acknowledging a missed packet and set to zero (0) when indicating the end of the Image Download protocol.

R: the "Request" bit. This MUST be one in an Image Download Request.

packet sequence number: the packet sequence number of the missing image data slice.

### 6.2.3 Image Download Process

The AC will divides the bootable image into a series of slices and sends each slice as an Image Download packet. The size of each image data slice (and therefore the size of each image download packet) depends on the MTU of the connection determined during the DTLS handshake. With the transmission of each slice the AC MUST increment the packet sequence number.

Image Download packets are negatively ack'd. An AC MUST NOT assume anything about the reception of packets it sends based upon negative acks. One could naively assume that since the packets are sent sequentially that all packets with a sequence number of "n - 1" are implicitly ack'd by the receipt of a request for the packet with sequence number "n" to be retransmitted. Such an assumption would be incorrect since previous requests could, themselves, have been dropped.

The image download process is initiated by the WTP requesting a packet with the packet sequence number of zero (0). The AC sets the packet sequence counter for this WTP to one (1) and sends the first slice. The "Request" bit for the first slice sent by the AC MUST be set to zero (0) since the first slice was technically not requested.

The WTP sets a periodic timer that, when it fires, causes the WTP to sends Image Download requests for slices that have been missed since the last periodic timer had fired. Since individual Image Download packets are not ack'd the AC MUST NOT set a timer when each one is sent.

If a WTP notices missed image transfer packets-- when the difference

between the packet sequence number of a received image transfer packet and the packet sequence number of the last image transfer packet previously received is greater than one-- it will note that fact in a bitmask. When the periodic timer fires the WTP will request the slices that are absent from that bitmask. Each slice will be requested by sending a Download Request with a length of eight (8) and indicating the sequence number of the packet requested. The AC MUST interleave these retransmissions with packets in the sequence.

Since both sides implicitly agree upon the MTU of the link the WTP will know the slice size that the AC will use during the Image Download process. A dropped packet will therefore result in an internal buffer pointer on the WTP being incremented by the slice size and the lost packet requested. When the lost packet is received it can be inserted into the buffer in the space provided by the pointer increment when its loss was first detected. That is, loss of packet <n> will result in packet <n> being re-requested and when received inserted into the buffer at an offset of <n-1> * <slicesize> from the start of the buffer.

The final packet sent by the AC will not have the "more" bit set and this indicates to the WTP that the end of the image has been received. This final packet is acknowledged by the WTP indicating the end of the Image Download process.

A lost final packet will result in the AC resending the final packet again (see Section 4.4).

### 6.2.4 Image Download State Machine

The Image Download protocol is a negotiated control protocol defined for SLAPP. Transitions to it come from the "secure" state and transitions out of it go to "acquire". See Figure 3.

#### 6.2.4.1 AC

The AC's state machine for the Image Download protocol is shown in Figure 30. The AC maintains the following variables for its state machine:

seq_num: the current slice that is being sent

nslices: the total number of slices in the image

req_num: the number of the slice that was requested

more: whether the "More bit" in the packet should be set

starved: a timer which sets the maximum amount of time in which an AC will attempt to download an image.

Note: the symbol "C" indicates an event in a state which results in the state remaining the same.

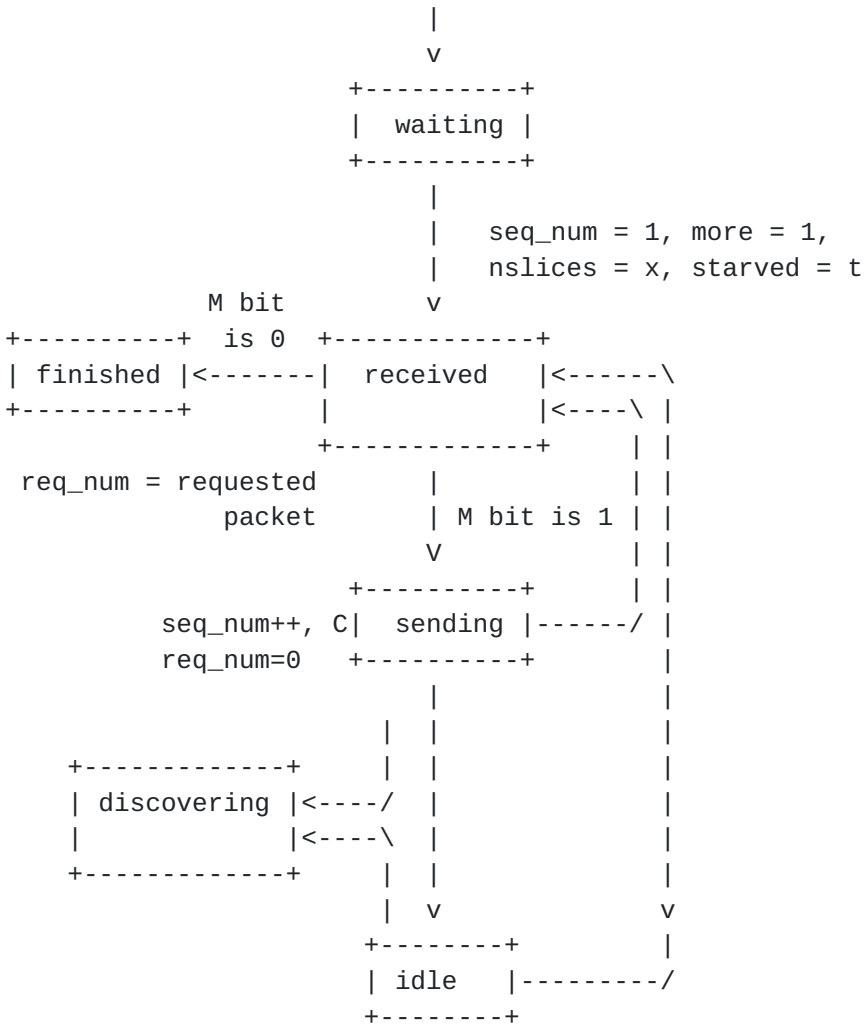



Figure 30: SLAPP Image Download Protocol State Machine at the AC

The following states are defined:

Waiting: When the AC leaves the SLAPP state of "Secure" it enters the "Waiting" state of the Image Download protocol. seq_num is set to one (1), more is set to one (1), and nslices is set to the number of slices in the particular image to download, and starved is set

to the maximum amount of time the AC will devote to downloading a particular image.

Received: The AC enters this state when it has received an Image Download request. If the sequence number of the packet is zero (0) it sets seq_num to one (1) and transitions to Sending, else if the M bit is set it sets req_num to the sequence number of the request and transitions to Sending else (if the M bit is clear) it transitions to Finished.

Sending: The AC is sending a slice to the WTP. If req_num is equal to zero (0) it sends the slice indicated by seq_num and increments seq_num. If req_num is greater than zero (0) it sends the slice indicated by req_num and sets req_num to zero (0). The "More" bit in either case is set depending on the value of more. As long as no request packets are received Sending transitions to Sending. When seq_num equals nslices "More" is set to zero (0) and the state transitions to Idle. If the starved timer expires the AC transitions to the SLAPP state of Discovering.

Idle: The AC has sent all the slices in the image and is just waiting for requests. If the starved timer expires the AC transitions to the SLAPP state of Discovering.

Finished: The Image Download protocol has terminated. The starved timer is canceled.

#### 6.2.4.2 WTP

The WTP's state machine for the Image Download protocol is shown in Figure 31. The WTP maintains the following variables for its state machine:

recv_num: the sequence number of the last received slice

req: a bitmask whose length equals the number of slices in the image

retry: a timer

giveup: a timer

final: the sequence number of the last slice

Note: the symbol "C" indicates an event in a state which results in the state remaining the same.

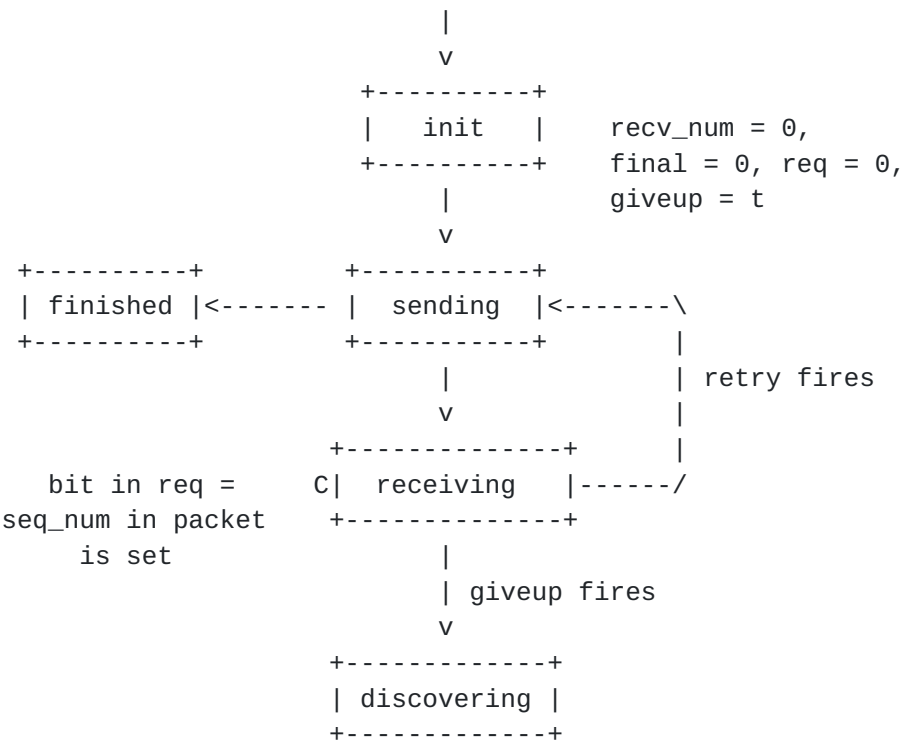



Figure 31: SLAPP Image Download Protocol State Machine at the WTP

The following states are defined:

Init:

When the WTP leaves the SLAPP state of "Secure" it enters the "Init" state of the Image Download Protocol. recv_num, final, and the req bitmask are set to zero (0), and the giveup timer is set to a suitably large number. The WTP transitions directly to Sending.

Sending:

If recv_num is zero (0) the WTP sends a request for a packet with sequence number of zero (0) and the "More" bit set to one (1). Otherwise for every unset bit in req between one (1) and recv_num a request packet is sent with the sequence number corresponding to the unset bit in req and the "More" bit set to more.

If there are no unset bits in req and final is non-zero a request packet is sent for the sequence number represented by final with the "More" bit cleared, giveup is cleared and the state machine transitions to Finished. Otherwise retry is set to a suitable value and the WTP transitions to Receiving.

Receiving:

In this state the WTP receives Image Download packets. The bit in req corresponding to the sequence number in the received packet is set indicating this packet has been received. If the sequence number of the received packet has already been received the packet is silently dropped, otherwise the data in the packet is stored as the indicated slice in a file which represents the downloaded image. If the received packet has the "More" bit cleared final is set to the sequence number in that packet. When the retry timer fires the WTP transitions to Sending. If the giveup timer fires the WTP transitions to the SLAPP state of Discovering.

Finished:

The image download protocol has finished.

## 7. Security Considerations

This document describes a protocol, SLAPP, which uses a different protocol, DTLS, to provide for authentication, key exchange, and bulk data encryption of a negotiated control protocol. It's security considerations are therefore those of DTLS.

The AC creates state upon receipt of an acceptable Discovery Request. AC implementations of SLAPP SHOULD therefore take measures to protect themselves from denial of service attacks which attempt to exhaust resources on target machines. These measures could take the form of randomly dropping connections when the number of open connections reaches a certain threshold.

The WTP exposes information about itself during the discovery phase. Some of this information could not be gleaned by other means.

## 8. Extensibility to other technologies

The SLAPP protocol can be considered to be a technology-independent protocol that can be extended with technology-specific components to solve an interoperability problem where a central controller from one vendor is expected to control and manage network elements from a different vendor.

While the description of the SLAPP protocol in this draft assumes that it is meant to solve the multi-vendor interoperability problem as defined in the CAPWAP problem statement [3], splitting the solution to two components where technology-dependent control protocols are negotiated using a technology-independent framework enables the use of SLAPP as the common framework for multiple underlying technologies that are vastly different from one another.

## 9. References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997, <ftp://ftp.isi.edu/in-notes/rfc2119>.

[2] "Architecture Taxonomy for Control and Provisioning of Wireless Access Points(CAPWAP)", August 2004, <ftp://ftp.isi.edu/internet-drafts/draft-ietf-capwap-arch-06.txt>.

[3] "Configuration and Provisioning for Wireless Access Points (CAPWAP) Problem Statement", February 2005, <http://www.ietf.org/rfc/rfc3990.txt>.

[4] "Generic Routing Encapsulation", March 2000, <http://www.ietf.org/rfc/rfc2784.txt>.

[5] "Requirements for Internet Hosts - Communication Layers", October 1989, <http://www.ietf.org/rfc/rfc1122.txt>.

[6] Govindan, S., "Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)", November 2004, <http://www.ietf.org/internet-drafts/draft-ietf-capwap-objectives-00.txt>.

[7] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", February 2004, <http://www.ietf.org/internet-drafts/draft-rescorla-dtls-03.txt>.

[8] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999, <ftp://ftp.isi.edu/in-notes/rfc2246.txt>.

[9] Modadugu, N. and E. Rescorla, "The Design and Implementation of Datagram TLS", <http://crypto.stanford.edu/~nagendra/papers/dtls.pdf>.

[10] Krishna, P. and D. Husak, "Simple Lightweight RFID Reader Protocol", March 2005, <http://www.ietf.org/internet-drafts/draft-krishna-slrrp-01.txt>.

## Authors' Addresses

Partha Narasimhan
Aruba Networks
1322 Crossman Ave
Sunnyvale, CA 94089

Phone: +1 408-480-4716
Email: partha@arubanetworks.com

Dan Harkins
Trapeze Networks
5753 W. Las Positas Blvd
Pleasanton, CA 94588

Phone: +1-925-474-2212
Email: dharkins@trpz.com

Subbu Ponnuswamy
Aruba Networks
1322 Crossman Ave
Sunnyvale, CA 94089

Phone: +1 408-754-1213
Email: subbu@arubanetworks.com

## Intellectual Property Statement



## Disclaimer of Validity



## Copyright Statement



## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.