**CAPWAP(AP)**

**Functional Specification**

Published by:

Software Department

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Issue Date** | **Remarks** |
| 1.0 |  |  |  |
| 2.0 |  |  |  |
| 3.0 |  |  |  |

Company Confidential

The information in this document is confidential and proprietary. This document is the property of and shall not be reproduced or copied or used in whole or in part without the written permission. This is an unpublished work protected under the Federal copyright laws.

**Contents**

[1 Introduction 1](#_Toc345251222)

[1.1 Document Purpose 1](#_Toc345251223)

[1.2 Intended audience 1](#_Toc345251224)

[1.3 Reference Document 1](#_Toc345251225)

[1.4 Reference Standard 1](#_Toc345251226)

[2 Function Realization 2](#_Toc345251227)

[2.1 Overview 2](#_Toc345251228)

[2.2 Function Mechanism & Operation 2](#_Toc345251229)

[2.3 Function design 2](#_Toc345251230)

[2.4 Data Structure 2](#_Toc345251231)

[2.5 Special Consideration 2](#_Toc345251232)

[2.6 Relationship with Other Modules 2](#_Toc345251233)

[3 User Interface 3](#_Toc345251234)

[3.1 CLI 3](#_Toc345251235)

[3.2 Log 5](#_Toc345251236)

[3.3 MIB 5](#_Toc345251237)

[3.4 Trap 5](#_Toc345251238)

[4 Unit Pre-test 6](#_Toc345251239)

[5 Appendix 7](#_Toc345251240)

# Introduction

本文档主要描述CAPWAP协议在AP端的设计和实现。

# FunctionRealization

## Overview

CAPWAP协议是基于WLAN体系结构的无线接入点控制协议， 为AP和AC之间的交互提供统一的标准化接口，用于对无线网络中AP 的控制和管理。通过CAPWAP协议，AP能够加入到AC控制的WLAN网络，同时与AC建立可靠连接。

CAPWAP协议将整个WLAN网络分成位于前端接入点AP和无线控制器AC两个部分，AP可看作是AC的远程射频端口，接受AC的控制,CAPWAP协议为AC和AP之间的互通性提供一个通用封装和传输机制，如图：

## Function Mechanism & Operation

### 2.2.1 CAPWAP 帧格式

CAPWAP协议规定AC和AP通信报文分为控制报文和数据报文两种，控制报文在AC和WTP之间传输，实现配置、管理、监控等功能，数据报文则是将被转发的用户数据帧。

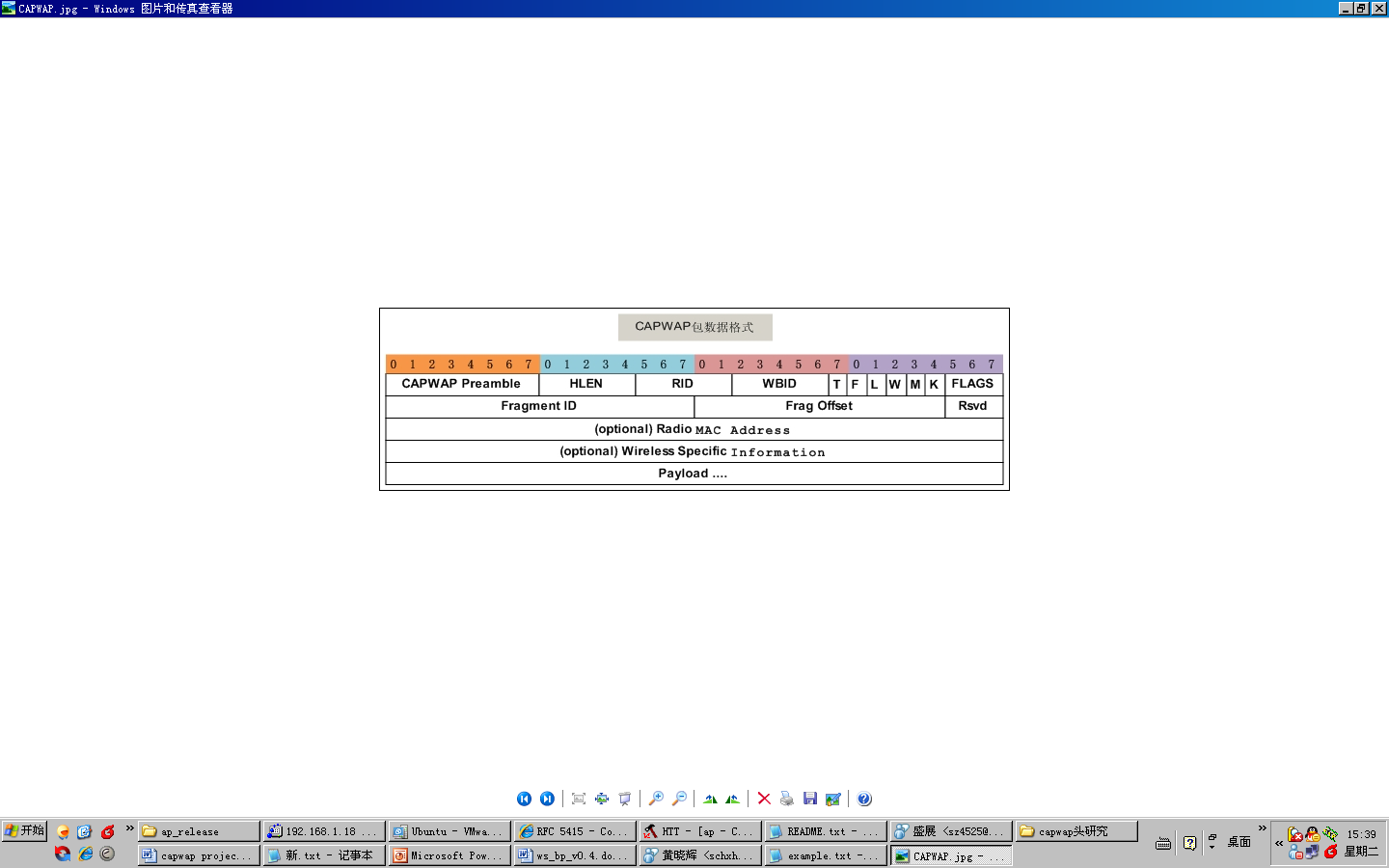
**CAPWAP控制报文结构：**



**CAPWAP数据报文结构：**



**CAPWAP协议头如下图所示**

****

Preamble：分为两个部分version(capwap版本号)和type(0-capwap header, 1-capwap dtls header)

HLEN：capwap头的长度

RID：radio ID

WBID：wireless binding id

T：传输的报文的格式，1表示负载为由WBID定义的类型，0表示负载为802.3帧。

F：当前报文是否分片。1表示当前报文是分片。

L：只有F标志置位才有效。1表示是最后一个分片，0表示后面还有分片。

W：表示optional Wireless Specific Information字段是否存在，1为存在。

M：表示Radio MAC Address optional是否存在。

K：表示是否是数据通道的keep-alive包。

Flags：保留字段。

Fragment ID：分片时标识每个分片的序列号。

Fragment Offset：分片时标识每个分片的偏移（8字节为单位）。

Reserved：保留字段。

Radio MAC Address：当报文中是802.3帧时保留wtp的mac。

Wireless Specific Information：无线扩展信息。

Payload: 这个字段包含CAPWAP的数据消息和控制消息。

**CAPWAP控制报文如下**



Message Type：

CAPWAP Control Message Message Type Value

Discovery Request 1

Discovery Response 2

Join Request 3

Join Response 4

Configuration Status Request 5

Configuration Status Response 6

Configuration Update Request 7

Configuration Update Response 8

WTP Event Request 9

WTP Event Response 10

Change State Event Request 11

Change State Event Response 12

Echo Request 13

Echo Response 14

Image Data Request 15

Image Data Response 16

Reset Request 17

Reset Response 18

Primary Discovery Request 19

Primary Discovery Response 20

Data Transfer Request 21

Data Transfer Response 22

Clear Configuration Request 23

Clear Configuration Response 24

Station Configuration Request 25

Station Configuration Response 26

Wlan Configuration Request 27

Wlan Configuration Response 28

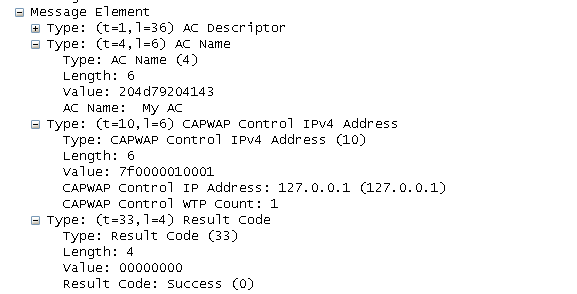
Seq Num：对于需要重传的CAPWAP控制报文，依次递增。Response报文的Seq Num应与对应的Request报文的序列号相同。

Msg Element Length：为Seq Num后的报文长度。

**Msg Element 报文：**



具体内容如下图所示：

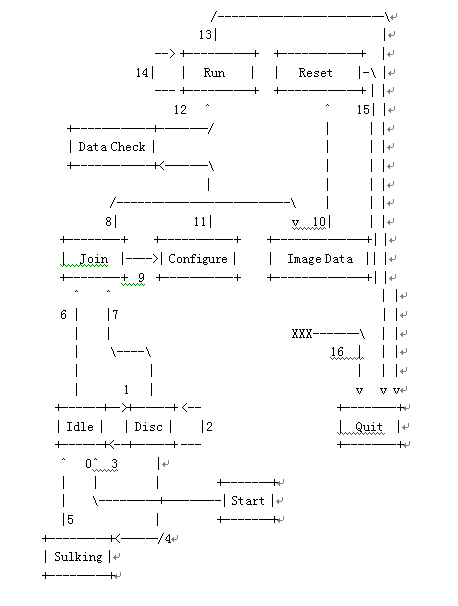


### 2.2.2 CAPWAP消息交互流程和状态机定义

#### 2.2.2.1 AC – AP消息交互流程



#### 2.2.2.2 CAPWAP状态机流程图



### 3. AP系统总体框架

#### 3.1控制通道总体架构



AC和AP之间的CAPWAP隧道只是他们通信的接口，当AP收到AC对其配置的变更，如是对WLAN的配置变更(wlan config request)，WTPD将对其CAPWAP报文解封装，将报文通过PAPI发送至SAPD，SAPD通过ioctl或者系统调用设置更新的配置，然后返回OK给WTPD，WTPD送response给AC。

#### 3.2 数据通道转发模式

CAPWAP按照数据转发的格式不同分为两种转发模式： Split Mac 和 Local Mac

不同转发模式下功能和服务的处理分配情况：

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | | **Local MAC** | **Split MAC** |
| Function | Distribution Service（分发服务） | AP/AC | AC |
| Integration Service（集中服务） | AP | AC |
| Beacon Generation（Beacon处理） | AP | AP |
| Probe Response Generation（Probe处理） | AP | AP |
| Power Mgmt/Packet Buffering（节电管理/报文缓冲） | AP | AP |
| Fragmentation/Defragmentation（分片、重组） | AP | AP/AC |
| Assoc/Disassoc/Reassoc（关联/解关联/重关联） | AP/AC | AC |
| IEEE 802.11 QoS | Classifying（WMM流分类） | AP | AC |
| Scheduling（WMM调度） | AP | AP/AC |
| Queuing（WMM队列） | AP | AP |
| IEEE 802.11 RSN(WPA2) | IEEE 802.1X/EAP | AC | AC |
| RSNA Key Management（RSNA密匙管理） | AC | AC |
| IEEE 802.11 Encryption/Decryption（加密/解密） | AP | AP/AC |

Split Mac：

1. 802.11控制报文（ RTS、CTS, etc) 终结在AP
2. 实时管理报文（probe/beacon）终结在AP，非实时管理报文(Assoc/auth)终结至AC
3. CAPWAP数据隧道直接转发所有802.11数据报文至AC

Local Mac:

1. 802.11控制报文（ RTS、CTS, etc) 终结在AP
2. 管理报文基本终结在AP,部分终结在AC
3. AP完成802.11数据报文向802.3报文格式的转变，CAPWAP数据隧道转发802.3数据帧至AC

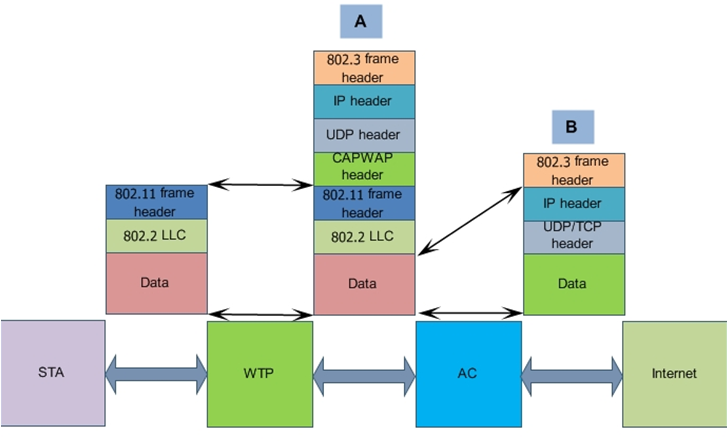
Local Mac必须在AP上完成802.11数据报文向802.3报文格式的转变，然后通过CAPWAP数据隧道转发到AC。

## CAPWAP模块设计和实现

### 4.1 CAPWAP数据通道的设计和实现

1. Split Mac模式：

在split mac模式下，AP完全接受AC的控制，只有实时帧片理在AP端，而验证、安全管理以及移动功能由AC处理，AP收到前端用户STA数据，通过CAPWAP协议封装传至AC，AC通过CAPWAP控制通道实现对AP的集中控制和管理。



WLAN中数据的传输过程如上图所示，数据在station和AP之间进行加解密，以802.11无线帧传输，并在AP与AC之间封装CAPWAP协议头，经由CAPWAP数据隧道进行传输，而在AC上进行802.11与802.3帧的转换。

基于要求，建立快速转发模块，由快速转发模块完成对数据的CAPWAP打包和解包工作如图所示。具体设计如下：



数据通道架构图

1. SAPD根据系统工作模式，启动快速转发，将需要的参数进行配置;
2. 用户数据由无线驱动保持802.11格式转发给快速转发模块，快速转发模块收到802.11帧，进行CAPWAP封装，根据快速转发设置的参数进行转发;
3. 快速转发模块收到AC下行的用户CAPWAP数据报文，进行CAPWAP解封装，根据802.11帧BSSID进行相应转发;

2. Split Mac 实现：

1> STA发送数据报文到AC

在802.11报文转成802.3之前，调用快速转发模块：

IF (vap == split mac && p\_type != beacon &&

p\_type != probe && p\_type != power)

{

1. 封装CAPWAP头SKB

2. 用eth0转发至AC

}

当STA发送数据报文到AC时，在ieee80211\_input函数中，对数据报文和管理报文进行check, 除了beacon, probe, power manage报文和控制报文外，其他封装CAPWAP头，发送到AC。

2> AC发送数据报文到STA

在函数udp\_rcv中调用快速转发模块判断capwap协议

IF(protocol == CAPWAP && p\_type == 802.11)

{

1. 解封装CAPWAP头

2. 获取VAP device

3. 调用此VAP device发送数据

}

当AC发送数据报文到STA时，在TCP/IP协议栈udp\_rcv函数中，对数据报文进行check，如果其协议是CAPWAP，并且封装报文是802.11帧，则对报文进行解封装，发送至STA。

2. Local Mac:

### CAPWAP控制通道的设计和实现

CAPWAP协议作为AP和AC的接口，规范AP与AC的通信行为，实现无线网络AP的集中控制和管理，该协议主要功能包括AP自动发现AC，AC对AP进行安全认证，AP从AC获取软件映像，AP从AC获得初始和动态配置等。

CAPWAP控制报文流程:



#### capwap状态机机制实现





#### Discovery过程的设计和实现

AP发现AC共有4种方式，发现的次序依次为：

1. 静态发现
2. 动态发现
3. DSN发现
4. 主播发现
5. 参数说明

AP-DiscoverInterval： AP接收到Discovery Response后进行下一步操作之前的最小等待时间

AP-MaxDiscoveryInterval： AP前后两次发送Discovery Request的最大时间间隔

AP-DiscoveryCount： AP向一个AC发送Discovery Request的次数

AP-MaxDiscoveries： AP启动后发送Discovery Request报文的最大次数

1. 状态变更
2. Discovery -> Discovery

AP-DiscoveryInterval计时器超时, AP应该增加DiscoveryCount计数器

1. Discovery -> Sulking

当AP-DiscoveryInterval超时并且DiscoveryCount超过MaxDiscoveries，AP就进入Sulking状态

1. Discovery -> Join

AP发送完Discovery request请求报文,如果接收到一个或者多个响应报文,就

根据配置策略选择一个AC进行接入，发送Join request请求报文。

1. Discovery 静态/动态流程图



CAPWAP发现过程

Pick AC的方法：

1. 选择优先级高的AC;
2. 如果优先级相同，选择负载最小的AC进行关联, 负载包括AC上已关联的AP数量和Client数量，先比较AP数量，AP数量相同的情况下比较Client数量;
3. 收到发现响应的先后次序;
4. 发现过程必须携带的元素列表

Discovery Request报文必须携带的消息元素：

[***Discovery Type***](#_发现类型（Discovery_Type）_1)

[***WTP Board Data***](#_WTP_Board_Data_1)

[***WTP Descriptor***](#_WTP描述符（WTP_Descriptor）_1)

[***WTP Frame Tunnel Mode***](#_WTP_Frame_Tunnel_Mode)

[***WTP MAC Type***](#_WTP_MAC_Type_1)

***[IEEE 802.11 WTP Radio Information](#_Radio信息（IEEE_802.11_WTP_Radio Infor)***

Discovery Response必须携带的消息元素：

***[AC Descriptor](#_AC描述符（AC_Descriptor）_1)***

[***AC Name***](#_AC名字（AC_Name）_1)

[***Control IPv4 Address***](#_控制隧道IPv4地址（Control_IPv4_Address）_1)***或***[***Control IPv6 Address***](#_控制隧道IPv6地址（Control_IPv6_Address）_1)

***[AC Priority](#_AC_Priority_1)***

#### Sulking状态

当AP在discovery状态得不到AC的响应时，AP进入本状态，在此状态下，WTPD保持监听socket端口的状态，如果有新的数据包到来，就执行接收动作，但是接收的数据包会被丢掉，当SlientInterval计时器超时，AP就重新进入Discovery状态,

#### Join过程的设计和实现

1. 状态变更
2. Join -> configuration

当AP接收到AC的Join response报文,判断AC支持的固件版本和当前运行的固件版本一致,就进入configuration状态。

1. Join -> Image-Data

当AP收到AC发过来的Join response报文,进行解析，并且判断AC支持的AP的固件版本和自己的固件版本比较,如果版本不同，进入Image-Data状态。

1. Join -> discovery

如果定时器超时还有没收到Join Response消息，AP就必须重新进入Discovery 状态，与**已发现AC列表中的下一台**发起关联。

注意：如果AP接收了一个**无效的（恶意的）**Join Response。AP应该记录这个错误信息的详细。处理这个错误的方法和没有收到这个Join Response一样，等待定时器超时，

1. Join过程流程图



1. Join过程必须携带的元素列表

Join Request报文必须包含以下元素：

[***Location Data***](#_位置（Location_Data）_1)

[***WTP Board Data***](#_WTP_Board_Data_1)

[***WTP Descriptor***](#_WTP描述符（WTP_Descriptor）_1)

[***WTP Name***](#_WTP_Name_1)

[***Session ID***](#_连接标识（Session_ID）_1)

[***WTP Frame Tunnel Mode***](#_WTP_Frame_Tunnel_Mode)

[***WTP MAC Type***](#_WTP_MAC_Type_1)

***[Local IPv4 Address或Local IPv6 Address](#_Local_IPv4_Address_1)***

***[IEEE 802.11 WTP Radio Information](#_Radio信息（IEEE_802.11_WTP_Radio Infor)***

Join Response中：

[***Result Code***](#_错误码（Result_Code）_1)

[***AC Descriptor***](#_AC描述符（AC_Descriptor）_1)

[***AC Name***](#_AC名字（AC_Name）_1)

[***Control IPv4 Address***](#_控制隧道IPv4地址（Control_IPv4_Address）_1) ***/*** [***Control IPv6 Address***](#_控制隧道IPv6地址（Control_IPv6_Address）_1)

Image Identifier(可选)：

1. 若报文中携带该元素，AP比较该版本和自身版本，结果相同时，进入Configure状态，否则进入Image Data状态；
2. 若报文中不携带该元素，AP直接进入Configure状态。

#### Image-Data过程的设计和实现

1. 状态变更
2. Image-Data -> Reset

当AP下载固件成功后,就给AC发送image date response,表示下载固件成功,等待AC下发的Reset Request，然后进入reset状态.

当AP下载固件失败，则直接进入reset状态，重启后重新关联AC。

1. Image-Data序列图

CAPWAP状态机在Image Data阶段：



AP Run之后以手工方式升级新版本：



1. Image-Data流程图

不论是AP是RUN状态还是Image data状态，在升级固件时流程是一样的。

AP-IntervalForImageDataRequest：

AP等待AC发送Image Data Request的最大时间，超时则RESET.



#### Configure过程的设计和实现

1. 状态变更：
2. configure –> Data-Check

当AP接收到AC的 Configuration status response响应报文,表明交换配置成功,AP就进入date check状态.

1. configure –> Reset

当AP接收不到AC的 Configuration status response响应报文或响应报文状态为Error, AP就进入Reset状态.

1. configure过程流程图：

在这个阶段中，AC和AP将交换配置信息，并对其进行设置，以使AP按照AC的规定运行。



1. configure过程必须携带的元素列表：

Configuration Status Request报文必须包含以下消息元素：

[***AC Name***](#_AC名字（AC_Name）_1)

[***Radio Administrative State***](#_Radio_Administrative_State_1)

[***Statistics Timer***](#_统计定时器（Statistics_Timer）_1)

[***WTP Reboot Statistics***](#_WTP_Reboot_Statistics_1)

[***IEEE 802.11 WTP Radio Configuration***](#_IEEE_802.11_WTP_Radio Configuration)

[***IEEE 802.11 WTP Radio Information***](#_Radio信息（IEEE_802.11_WTP_Radio Infor)

[***IEEE 802.11 Supported Rates***](#_IEEE_802.11_Supported_Rates_1)

下面的消息元素可能包含在Configure Status Request消息中：

[***AC Name with Priority***](#_AC带优先级的名字（AC_Name_with_Priority）)

[***IEEE 802.11 Direct Sequence Control***](#_IEEE_802.11_Direct_Sequence Control)

[***IEEE 802.11 OFDM Control***](#_IEEE_802.11_OFDM_Control_1)

[***IEEE 802.11 Tx Power***](#_IEEE_802.11_Tx_Power)

[***Vendor Specific Payload***](#_Vendor_Specific_Payload)

Configure Status Response报文必须包含以下消息元素：

***[CAPWAP Timers](#_CAPWAP_Timers_1)***

***[Decryption Error Report Period](#_Decryption_Error_Report_Period_1)***

***[Idle Timeout](#_Idle_Timeout_1)***

下列消息可选包含在Configure Status Response报文中：

***[AC IPv4 List（内容必须支持，AP可以不做处理）](#_AC描述符（AC_Descriptor）)***

***[AC IPv6 List（内容必须支持，AP可以不做处理）](#_AC_IPv6地址列表（AC_IPv6)***

[***IEEE 802.11 Direct Sequence Control***](#_IEEE_802.11_Direct_Sequence Control)

[***IEEE 802.11 OFDM Control***](#_IEEE_802.11_OFDM_Control_1)

[***IEEE 802.11 Tx Power***](#_IEEE_802.11_Tx_Power)

[***IEEE 802.11 Rate Set***](#_IEEE_802.11_Rate_Set)

[***IEEE 802.11 WTP Radio Information***](#_Radio信息（IEEE_802.11_WTP_Radio Infor)

[***IEEE 802.11 MAC Operation***](#_IEEE_802.11_MAC_Operation)

***[IEEE 802.11 WTP Quality of Service](#_IEEE_802.11_WTP_Quality of Service)***

***[IEEE 802.11 WTP Radio Configuration](#_IEEE_802.11_WTP_Radio Configuration)***

[***IEEE 802.11 Multi-Domain Capability***](#_IEEE_802.11_Multi-Domain_Capability_1)

[***IEEE 802.11 Antenna***](#_IEEE_802.11_Antenna_1)

***[Vendor Specific Payload](#_Vendor_Specific_Payload)***

#### Data-Check过程的设计和实现

1. 状态变更：
2. Data-Check -> Run

当AP接收到change state event response响应报文,并且表明是成功,AP就进入run状态.

1. Data-Check -> Reset

当接收不到change state event response响应报文，或者报文响应Error,AP将进入Reset状态。

1. Data-Check程序流程图

Change State Event Request报文是AP发送给AC，用来确认或上报错误状况，这个状况可能是AP收到AC的Configuration Status Response报文后配置失败。Change State Event Request报文包含消息元素Result Code，表明AP对于配置执行是否成功。

Change State Event Request报文必须包含以下元素：

***[Radio Operational State](#_Radio_Operational_State_1)***

***[Result Code](#_错误码（Result_Code）_1)***



Data-Check程序流程图

#### Run过程的设计和实现

Run过程总流程图：



以下的报文类型都将在Run状态检测和处理。

* + - 1. 配置更新请求（Configuration Update Request）
      2. 事件请求（WTP Event Request）
      3. 状态改变请求（Change State Event Request）
      4. 回声请求（Echo Request）
      5. 重置请求（Reset Request）
      6. 站点配置请求（Station Configuration Request）
      7. 无线局域网配置请求（WLAN Configuration Request）
      8. 清空配置请求（Clear Configuration Request）
      9. 数据传输请求（Data Transfer Request）

## Data Structure

## Special Consideration

## Relationship with Other Modules

## User Interface

## CLI

**Command List**

|  |  |  |
| --- | --- | --- |
| **Item** | **Command Name** | **WebUI** |
| 1 | xxx |  |
| 2 | xxx |  |

1. **aaa-xxx**

**Description**

**Parameter**

**Usage Guideline**

**Example**

1. **aaa-xxx**

**Description**

**Parameter**

**Usage Guideline**

**Example**

## Log

## MIB

## Trap

## Unit Pre-test

# Appendix