MEDIATEK

MT7986 Concurrent WPS AN

2021/10/14

Nishank Aggarwal

Version History

Version	Date	Author (Optional)	Description
0.1	2021-9-27	Nishank Aggarwal	Initial draft
1.0	2021-10-14	Micheal Su	Official release

Outline

- □ Feature Description
- □ How to Configure – runtime command
- □ WPS Trigger Flow

Feature Description

WPS Architecture

A device with the authority to issue and revoke access to a network; it may be integrated into a wireless access point (AP), or provided as a separate device.

A device seeking to join a wireless network.

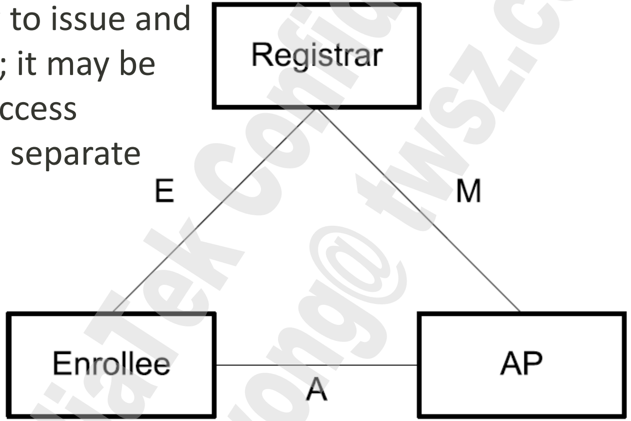


Figure 1 – Components and Interfaces

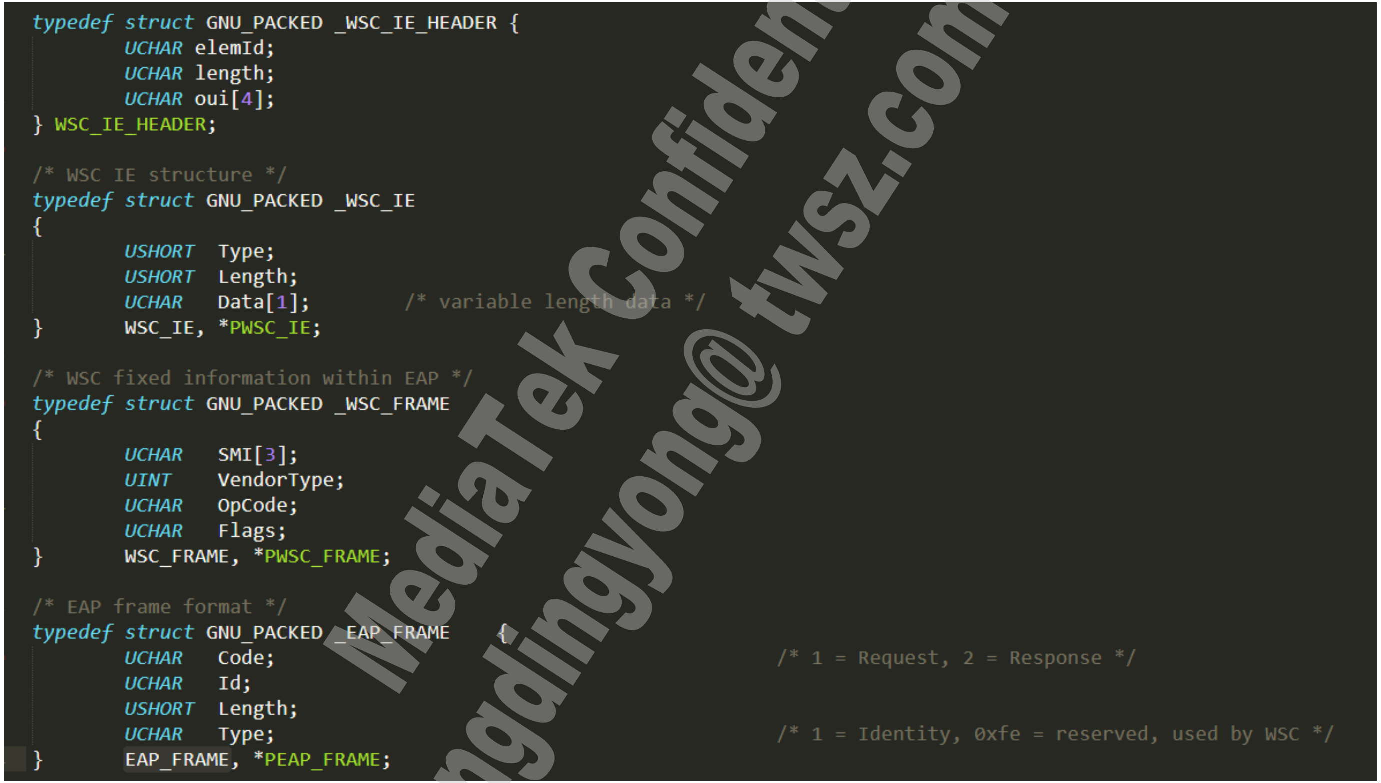
An access point functioning as a **proxy** between a registrar and an enrollee.

Internal/Standalone Registrar: Registrar & AP are integrated.
External Registrar (ER): AP and Registrar are separated.

Driver WSC File Description

- **wsc.h – WSC Data Structure Definitions**
- **wsc_tlv.h – WSC Data Element Definitions**
- **wsc.c – WSC Function State Machine**
- **wsc_tlv.c – WSC Messages Build/Process**
- **wsc_v2.c – New API for WSC V2**
- **wsc_ufd.c – Parse WSC data from USB Flash Drives (UFD)**
- **nfc.c – WSC Function for Near-Field Communication (NFC)**

WSC IE and EAP FORMAT



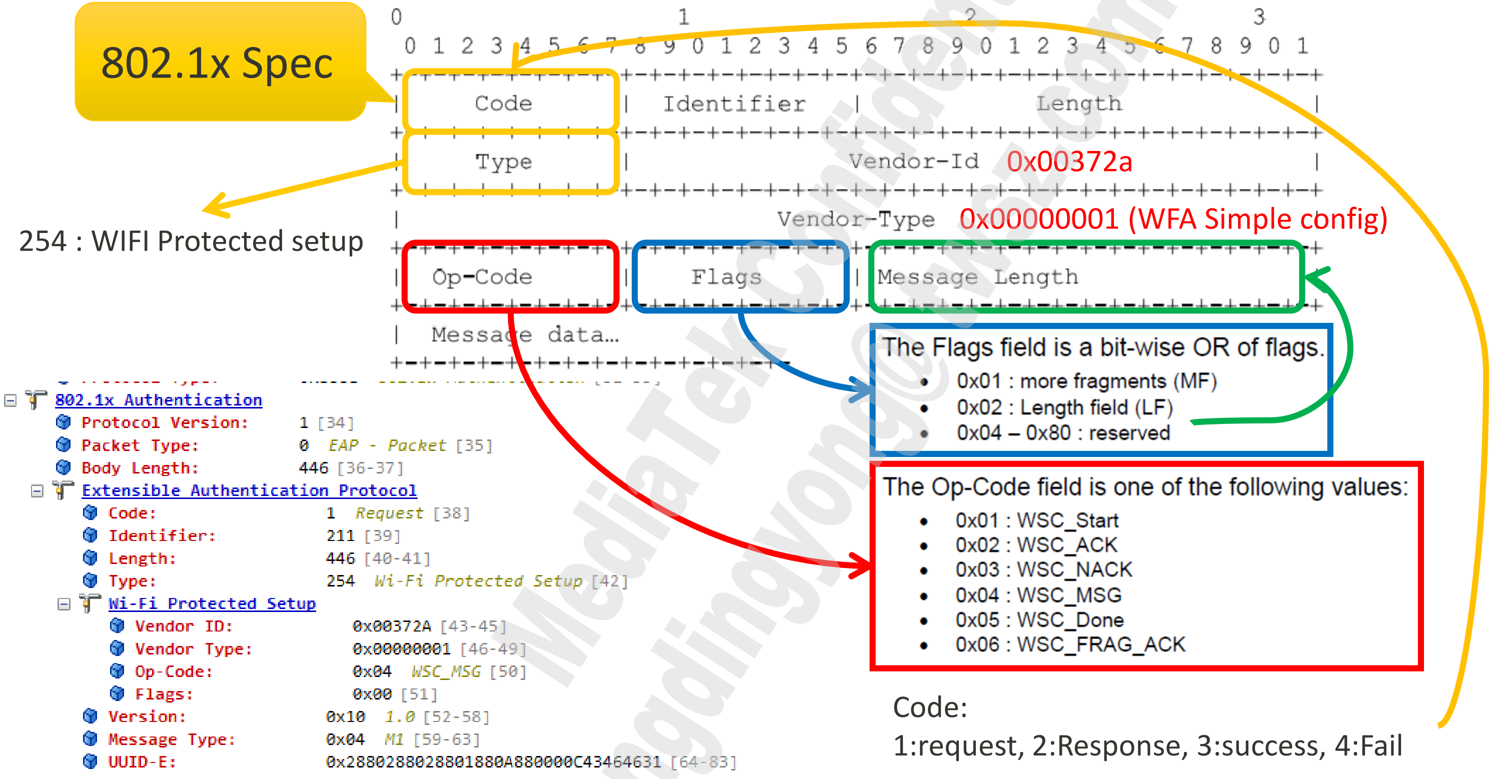
```
typedef struct GNU_PACKED _WSC_IE_HEADER {
        UCHAR elemId;
        UCHAR length;
        UCHAR oui[4];
} WSC_IE_HEADER;

/* WSC IE structure */
typedef struct GNU_PACKED _WSC_IE
{
        USHORT  Type;
        USHORT  Length;
        UCHAR   Data[1];        /* variable length data */
}       WSC_IE, *PWSC_IE;

/* WSC fixed information within EAP */
typedef struct GNU_PACKED _WSC_FRAME
{
        UCHAR   SMI[3];
        UINT    VendorType;
        UCHAR   OpCode;
        UCHAR   Flags;
}       WSC_FRAME, *PWSC_FRAME;

/* EAP frame format */
typedef struct GNU_PACKED _EAP_FRAME    {
        UCHAR   Code;                                   /* 1 = Request, 2 = Response */
        UCHAR   Id;
        USHORT  Length;
        UCHAR   Type;                                   /* 1 = Identity, 0xfe = reserved, used by WSC */
}       EAP_FRAME, *PEAP_FRAME;
```

EAP-WSC Packet Format



802.1x Spec

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |               Vendor-Id                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Vendor-Type                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Op-Code      |    Flags      | Message Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

0x00372a

0x00000001 (WFA Simple config)

254 : WIFI Protected setup

The Flags field is a bit-wise OR of flags.

- 0x01 : more fragments (MF)
- 0x02 : Length field (LF)
- 0x04 – 0x80 : reserved

The Op-Code field is one of the following values:

- 0x01 : WSC_Start
- 0x02 : WSC_ACK
- 0x03 : WSC_NACK
- 0x04 : WSC_MSG
- 0x05 : WSC_Done
- 0x06 : WSC_FRAG_ACK

Code:
1:request, 2:Response, 3:success, 4:Fail

```
⊟ 802.1x Authentication
      Protocol Version:      1 [34]
      Packet Type:           0  EAP - Packet [35]
      Body Length:           446 [36-37]
  ⊟ Extensible Authentication Protocol
        Code:                  1  Request [38]
        Identifier:            211 [39]
        Length:                446 [40-41]
        Type:                  254  Wi-Fi Protected Setup [42]
    ⊟ Wi-Fi Protected Setup
          Vendor ID:             0x00372A [43-45]
          Vendor Type:           0x00000001 [46-49]
          Op-Code:               0x04  WSC_MSG [50]
          Flags:                 0x00 [51]
        Version:               0x10  1.0 [52-58]
        Message Type:          0x04  M1 [59-63]
        UUID-E:                0x2880288028801880A880000C43464631 [64-83]
```

802.1x Authentication (EAP-WSC)

Byte 0	Byte 1	Byte 2	Byte 3
Protocol ver	Packet type(EAP type)	Body length	
Code	Identifier	Length	
Type	Vender ID (0x00372A)		
Vender type (0x1)			
OP-code	Flag		

Packet type:
0: EAP-Packet
1:EAPOL-Start

Code:
1: request
2:Response
3:Success
4:Fail

Type:
254: wifi protected setup

The Op-Code field is one of the following values:

- 0x01: WSC_Start
- 0x02: WSC_ACK
- 0x03: WSC_NACK
- 0x04: WSC_MSG
- 0x05: WSC_Done
- 0x06: WSC_FRAG_ACK

WSC Frame checklist (From AP view 1/2)

Frame name	wsc.c, wsc_tlv.c
Beacon (WSC IE) (TX)	WscBuildBeaconIE
Probe req (WSC IE)	
Probe rsp (WSC IE)(TX)	WscBuildProbeRespIE
Auth req	
Auth rsp	
Assoc req (WSC IE)	
Assoc rsp (WSC IE)(TX)	WscBuildAssocRespIE

WSC Frame checklist (From AP view 2/2)

Frame name	wsc.c, wsc_tlv.c
EAPOL-START (RX)	WscEAPOLStartAction
EAP-request-identity (TX)	WscSendEapReqId
EAP-response-identity	
EAP-Request (start,M2,M4,M6,M8)	BuildMessageM2,BuildMessageM4,BuildMessageM6,BuildMessageM8
EAP-response (M1, M3, M5,M7,Done) RX	ProcessMessageM1, ProcessMessageM3, ProcessMessageM5, ProcessMessageM7, WscEapRegistrarAction :WSC_MSG_WSC_DONE
EAP-FAIL	WscSendEapFail
Deauth	

WSC Frame checklist (From APCLI view)

Frame name	comment
Beacon (WSC IE)	WscBuildBeaconIE
Probe req (WSC IE)	
Probe rsp (WSC IE)	WscBuildProbeRespIE
Auth req	
Auth rsp	
Assoc req (WSC IE)	
Assoc rsp (WSC IE)	
EAPOL-START	
EAP-request-identity	
EAP-response-identity	
EAP-Request (start,M2,M4,M6,M8)	
EAP-response (M1, M3, M5,M7,Done)	
EAP-FAIL	
Deauth	

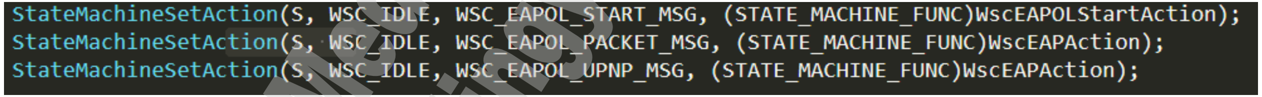
MlmeInit

MlmeQueueInit	
ApMlmeInit	
ApCliMlmeInit	APCLI_SUPPORT
WscStateMachineInit	WSC_INCLUDED
WpaStateMachineInit	
RTMPInitTimer	MlmePeriodicExecTimer, AsicRxAntEvalTimeout, APSDPeriodicExec, APQuickResponeForRateUpExec,
RTMP_OS_**TASK**_INIT(pTask, "RtmpMlmeTask", pAd);	/* Creat MLME Thread */ pTask = &pAd->mlmeTask;
RtmpOSTaskAttach (Attach kernel thread)	•RtmpOSTaskAttach(pTask, **MlmeThread**, (ULONG)pTask); •RtmpOSTaskAttach => __RtmpOSTaskAttach

Note: ps –ef can see all kthread

EAP-WSC RX PATH

- **RX:**
  - **WpsSmProcess**
    - StateMachinePerformAction (This will perform State machine transition function)
    - The transition function was register by **WscStateMachineInit**
  - **WscEAPOLStartAction (EAPOL-Start)**
  - **WscEAPAction (EAP_REQ / EAP_RSP / EAP_FAIL)**
    - WscEapRegistrarAction
    - WscEapApProxyAction
    - WscEapEnrolleeAction



```
StateMachineSetAction(S, WSC_IDLE, WSC_EAPOL_START_MSG, (STATE_MACHINE_FUNC)WscEAPOLStartAction);
StateMachineSetAction(S, WSC_IDLE, WSC_EAPOL_PACKET_MSG, (STATE_MACHINE_FUNC)WscEAPAction);
StateMachineSetAction(S, WSC_IDLE, WSC_EAPOL_UPNP_MSG, (STATE_MACHINE_FUNC)WscEAPAction);
```

EAP-WSC TX PATH

- **TX:**
  - **WscEapRegistrarAction (e.g. Receive M1 then build M2 or M2D)**
  - **RTMPSendWirelessEvent (MSG_PATH)**
    - WscSendMessage
      - MAKE_802_3_HEADER
      - sizeof(EAP_FRAME) + sizeof(WSC_FRAME) + Len
  - **(After Received Eapol-start)WscEAPOLStartAction => WscSendEapReqId (Send Identity path)**
  - **RTMPToWirelessSta**
    - **wdev->tx_pkt_ct_handle = FullOffloadFrameTx** (register in wdev_init)

How to Configure – runtime command

WPS command

- **CLI: iwpriv ra0 set**

Command	Purpose	**Function**
WscConfMode	=5 Registrar Enrollee (CONPWS) =4 Registrar(AP) =2 PROXY (AP) =1 Enrollee (STA)	Set_AP_WscConfMode_Proc
WscMode	=1 (PIN) =2 (PBC)	Set_AP_WscMode_Proc
WscConfStatus	=2	Set_AP_WscConfStatus_Proc (0x1044 wifi simple configuration state AP must =2)
WscGetConf	=1	Trigger WPS 2 mins timer

Normal WPS (AP)

- AP PIN
  - iwpriv ra0 set WscConfMode=4 //Registrar
  - iwpriv ra0 set WscMode=1
  - iwpriv ra0 set WscConfStatus=2
  - iwpriv ra0 set WscPinCode=12044085
  - iwpriv ra0 set WscGetConf=1
- AP PBC
  - iwpriv ra0 set WscConfMode=4 //Registrar
  - iwpriv ra0 set WscMode=2
  - iwpriv ra0 set WscConfStatus=2
  - iwpriv ra0 set WscGetConf=1

Normal WPS (APCLI)

- ApCli PIN
  - iwpriv apcli0 set ApCliEnable=1
  - ifconfig apcli0 up
  - brctl addif br0 apcli0
  - iwpriv apcli0 set WscConfMode=1   //Enrollee
  - iwpriv apcli0 set WscMode=1   //PIN method
  - iwpriv apcli0 show WscPin
  - iwpriv apcli0 set ApCliWscSsid=XXXXXXX
  - iwpriv apcli0 set WscGetConf=1   //Trigger
- ApCli PBC
  - iwpriv apclii0 set ApCliEnable=1
  - ifconfig apclii0 up
  - brctl addif br0 apclii0
  - iwpriv apclii0 set WscConfMode=1   //Enrollee
  - iwpriv apclii0 set WscMode=2   //PBC
  - iwpriv apclii0 set WscGetConf=1   //Trigger

iwpriv command - WscAutoTriggerDisable

- To disable AP Enrollee auto trigger capability
  - iwpriv ra0 set WscAutoTriggerDisable=1
- To enable AP Enrollee auto trigger capability again
  - iwpriv ra0 set WscAutoTriggerDisable=0

CONCURRENT WPS

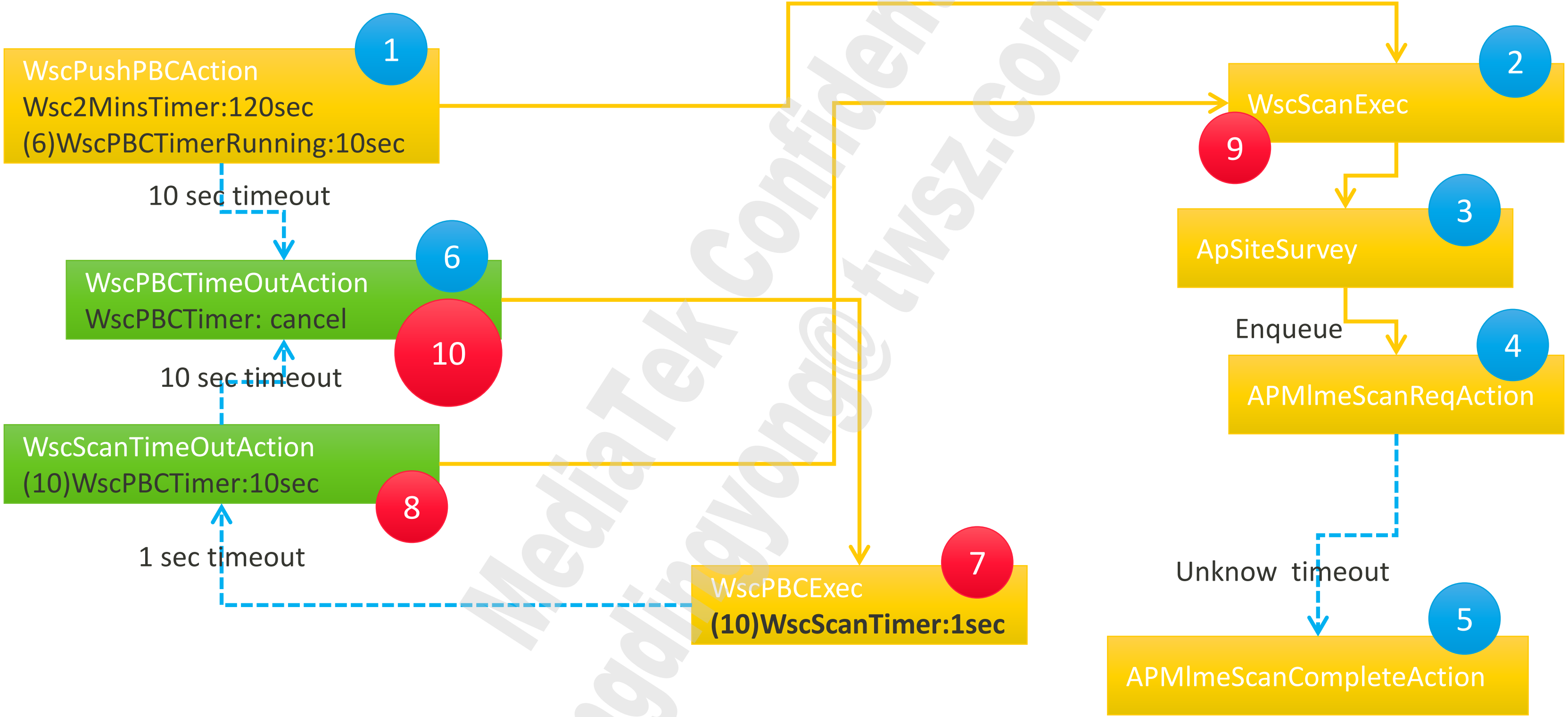
- Quick set up Dual Band concurrent WPS
  - ifconfig apcli0 up
  - ifconfig apclii0 up
  - iwpriv apcli0 set ApCliEnable=1
  - iwpriv apclii0 set ApCliEnable=1
  - iwpriv ra0 set ConWpsApcliPreferIface=1
  - iwpriv apcli0 set ConWpsApCliMode=0 //Enable the Auto selection CON_WPS
  - //Push one button
  - iwpriv ra0 set WscConfMode=5 //Trigger
  - iwpriv rai0 set WscConfMode=5 //Trigger

CONCURRENT WPS

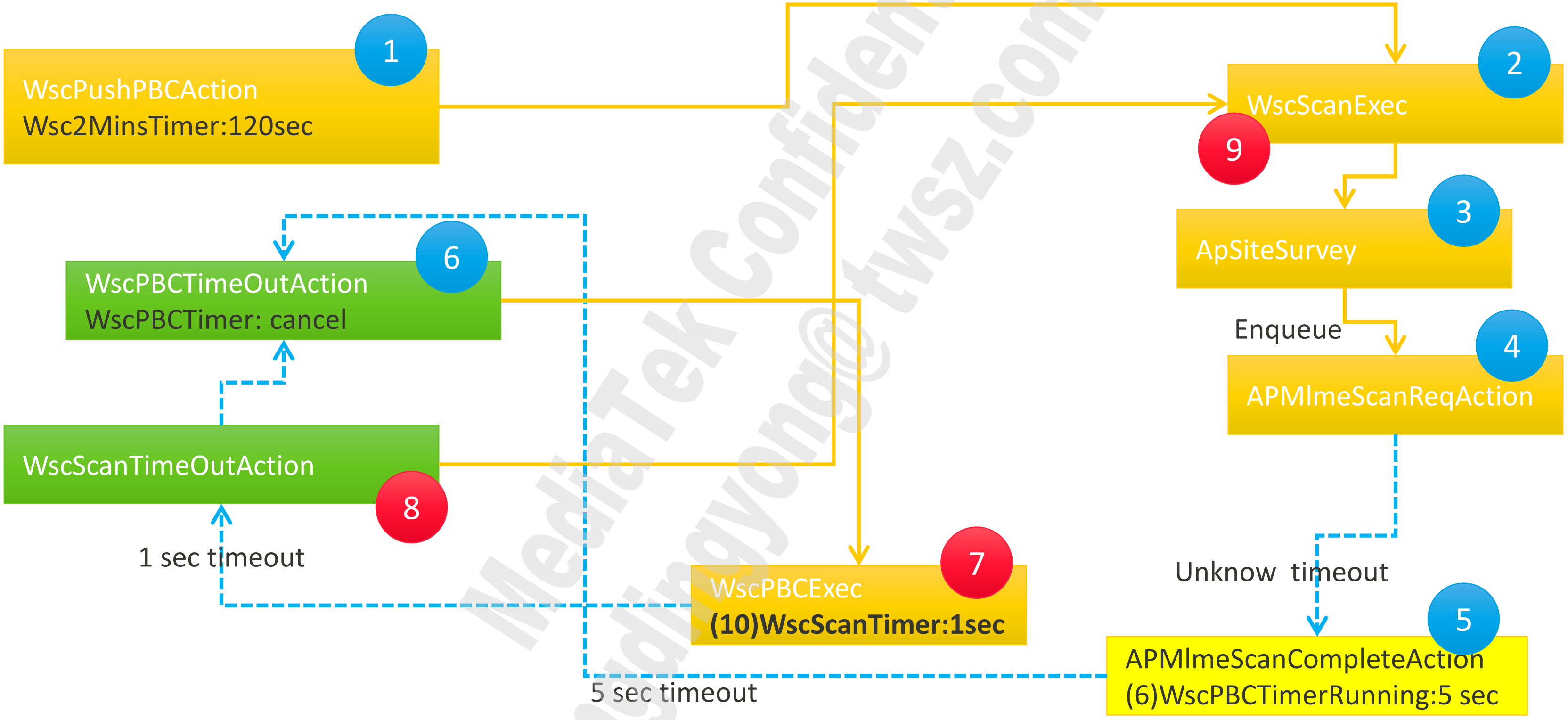
- iwpriv apcli0 set ConWpsApCliMode=0
  - 0: Auto Band Selection (probe req without wps IE)
  - 1: 2G Band Preferred
  - 2: 5G Band Preferred
- iwpriv ra0 set ConWpsApcliPreferIface=1 (Auto prefer 1)
- iwpriv apcli0 set ConWpsApCliDisabled=0
  - 0: means disabled (Default)
  - 1: means enabled the behavior as "Extender must not acts as Enrollee if it's connected to an AP already"

WPS Trigger Flow

Normal WPS


WscPushPBCAction
Wsc2MinsTimer:120sec
(6)WscPBCTimerRunning:10sec
1
10 sec timeout
WscPBCTimeOutAction
WscPBCTimer: cancel
6
10
10 sec timeout
WscScanTimeOutAction
(10)WscPBCTimer:10sec
8
1 sec timeout
WscPBCExec
(10)WscScanTimer:1sec
7
WscScanExec
2
9
ApSiteSurvey
3
Enqueue
APMlmeScanReqAction
4
Unknow timeout
APMlmeScanCompleteAction
5

CONCURRENT WPS



MediaTek Proprietary and Confidential

This document has been prepared solely for informational purposes. The content herein is made available to a restricted number of clients or partners, for internal use, pursuant to a license agreement or any other applicable agreement and subject to this notice. THIS DOCUMENT AND ANY ORAL INFORMATION PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT (COLLECTIVELY THIS “DOCUMENT”), IF ANY, ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS OR GUARANTEE REGARDING THE USE OR THE RESULT OF THE USE OF THIS DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, TIMELINESS, RELIABILITY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTIES ARISING OUT OF COURSE OF PERFORMANCE, COURSE OF DEALING OR USAGE OF TRADE. This Document must be held in strict confidence and may not be communicated, reproduced, distributed or disclosed to any third party or to any other person, or being referred to publicly, in whole or in part at any time except with MediaTek’s prior written consent, which MediaTek reserves the right to deny for any reason. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for your unauthorized use or disclosure of this Document, in whole or in part. If you are not the intended recipient of this document, please delete and destroy all copies immediately.

MEDIATEK

everyday genius