Cypress Docking Station Test

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev.** | **Change Description** | **Reason for Change** | **Author** | **Effective Date** |
| 1 | Initial Draft | Initial writeup | Ashweeja H L | 14th April 2020 |
| 2 | Addition of sections, content and description | For clarity on steps and commands for test execution | Meghana S Murthy | 23rd April 2020 |

**Abbreviations and Acronyms**

|  |  |
| --- | --- |
| **Acronyms and Terminologies** | **Description** |
| **Cobalt** | **On-board Wi-Fi chipset in STA mode which connects to enterprise AP forming the backhaul network** |
| **DHCP** | **Dynamic Host Control Protocol** |
| **Hastings prime** | **PCIe based Wi-Fi chipset in soft AP/P2P GO mode to which the Windows laptop connects and this forms the fronthaul network** |
| **LAN** | **Local Area Network** |
| **Linux endpoint and radius server** | **Backend for the enterprise AP. This also has the radius authentication server running on it** |
| **MA-USB** | **Media Agnostic USB stack which allows wireless devices to communicate over the USB protocol without needing a physical connection. This is also a part of the fronthaul network.** |
| **WAN** | **Wide Area Network** |
| **Windows endpoint and Test server** | **802.11ax compliant Windows 10 laptop which connects to the dock wirelessly. It also acts as the backend for the Cypress board as well as the test server for the test automation** |
|  |  |
|  |  |
|  |  |

**Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | **Name** | **Signature** | **Date** |
| **Software Engineering** | Ian Carvalho, Vinay Krishna, Kiran Kumar |  |  |
| **Product Marketing** |  |  |  |

Contact Information

Altran India (Global Edge), Global Village, RVCE Post, Mysore Road, Bengaluru, Karnataka 560059

Phone: 08071 006 677

**Table of Contents**

[1. Introduction 6](#_Toc41980779)

[2. Image compilation and flashing 6](#_Toc41980780)

[3. Test Setup 6](#_Toc41980781)

[3.1. Router mode 6](#_Toc41980782)

[3.2. Bridge mode 7](#_Toc41980783)

[3.3. Display-Link Mode 8](#_Toc41980784)

[4. Interface management 8](#_Toc41980785)

[5. Test Configurations 8](#_Toc41980786)

[5.1. Fronthaul configurations 8](#_Toc41980787)

[5.1.1. Configuring AP in open mode 8](#_Toc41980788)

[5.1.2. Configuring AP in WPA2 mode 9](#_Toc41980789)

[5.1.3. Configuring AP in WPA3 mode 10](#_Toc41980790)

[5.1.3.1. Opportunistic Wireless Encryption 10](#_Toc41980791)

[5.1.3.2. Simultaneous Authentication of Equals 10](#_Toc41980792)

[5.1.3.3. Device Provisioning Protocol 11](#_Toc41980793)

[5.1.4. WPS configuration 11](#_Toc41980794)

[5.1.5. WFA Easy Connect 12](#_Toc41980795)

[5.2. Backhaul configurations 12](#_Toc41980796)

[5.2.1. Configuring STA to connect in open mode 12](#_Toc41980797)

[5.2.2. Configuring STA to connect in WPA2 mode 13](#_Toc41980798)

[5.2.3. Configuring STA to connect in WPA3 mode 14](#_Toc41980799)

[5.2.3.1. Opportunistic Wireless Encryption 14](#_Toc41980800)

[5.2.3.2. Simultaneous Authentication of Equals 14](#_Toc41980801)

[5.2.3.3. Device Provisioning Protocol 15](#_Toc41980802)

[5.2.4. WPS configuration 15](#_Toc41980803)

[5.2.5. Configuring STA to connect in WPA2 enterprise mode 16](#_Toc41980804)

[5.2.5.1. EAP-AKA 16](#_Toc41980805)

[5.2.5.2. EAP-TLS 16](#_Toc41980806)

[5.2.5.3. EAP-TTLS 16](#_Toc41980807)

[5.2.5.4. EAP-MSCHAPv2 16](#_Toc41980808)

[5.2.6. Configuring STA to connect in WPA3 enterprise mode 16](#_Toc41980809)

[5.2.6.1. Suite B 16](#_Toc41980810)

[5.2.7. Wi-Fi Certified Enhanced Open 16](#_Toc41980811)

[5.3. Concurrency 16](#_Toc41980812)

[5.3.1. Configuring DUT in router mode with both fronthaul and backhaul in wireless mode 16](#_Toc41980813)

[5.3.2. Configuring DUT in bridge mode with both fronthaul and backhaul in wireless mode 18](#_Toc41980814)

[5.3.3. Configuring DUT in router mode with fronthaul in wireless mode and backhaul in wired mode 20](#_Toc41980815)

[5.3.4. Configuring DUT in bridge mode with fronthaul in wireless mode and backhaul in wired mode 21](#_Toc41980816)

[5.3.5. Configuring DUT in router mode with fronthaul in wired mode and backhaul in wireless mode 23](#_Toc41980817)

[5.3.6. Configuring DUT in bridge mode with fronthaul in wired mode and backhaul in wireless mode 24](#_Toc41980818)

[5.3.7. Configuring DUT in router mode with both fronthaul and backhaul in wired mode 26](#_Toc41980819)

[5.3.8. Configuring DUT in bridge mode with both fronthaul and backhaul in wired mode 27](#_Toc41980820)

[5.4. Miscellaneous configurations 29](#_Toc41980821)

[5.4.1. FreeRADIUS server bring up and configuration 29](#_Toc41980822)

[5.4.2. Configuring opensource hostapd in WPA3 SAE mode 30](#_Toc41980823)

[5.4.3. Configuring opensource wpa\_supplicant to associate to AP in WPA3 SAE mode 31](#_Toc41980824)

[6. Test Automation 31](#_Toc41980825)

[*6.1.* Description 31](#_Toc41980826)

[6.2. Automation Testbed 32](#_Toc41980827)

[6.2.1. Concurrent in router mode in WPA2 mode 32](#_Toc41980828)

[6.2.2. Concurrent in bridge mode in WPA2 mode 34](#_Toc41980829)

[6.2.3. Fronthaul only validation 35](#_Toc41980830)

[6.2.4. Backhaul only validation 36](#_Toc41980831)

[6.2.5. Concurrent in bridge mode with wired backhaul and WPA2 fronthaul 37](#_Toc41980832)

[6.2.6. Concurrent in router mode with wired backhaul and WPA2 fronthaul 38](#_Toc41980833)

[6.2.7. Concurrent in bridge mode with wired backhaul and fronthaul 38](#_Toc41980834)

[6.2.8. Concurrent in router mode with wired backhaul and fronthaul 39](#_Toc41980835)

[6.3. Source Code Directory structure 39](#_Toc41980836)

[6.4. System Requirements 39](#_Toc41980837)

[6.5. Software Installation And Configuration 40](#_Toc41980838)

[6.6. Configuration and execution Procedure 42](#_Toc41980839)

[6.6.1. Test bed preparation 42](#_Toc41980840)

[6.6.2. Download and Install Automation Framework 42](#_Toc41980841)

[6.6.3. Cypress Docking Station Automation Testbed Configuration 42](#_Toc41980842)

[6.7. Test Scripts 46](#_Toc41980843)

[6.7.1. Execution 46](#_Toc41980844)

[6.7.2. PASS/FAIL Criteria 46](#_Toc41980845)

# Introduction

Cypress wireless docking station houses two Wi-Fi chipsets. Integrated cobalt supporting 802.11ax in 2x2 mode at 5GHz would connect to the corporate enterprise Wi-Fi network using 802.1x, forming the backhaul network.

PCIe Hastings prime module using 802.11ax @ 6GHz to which the Dell laptop running Windows 10 would be connecting wirelessly, forming the fronthaul network.

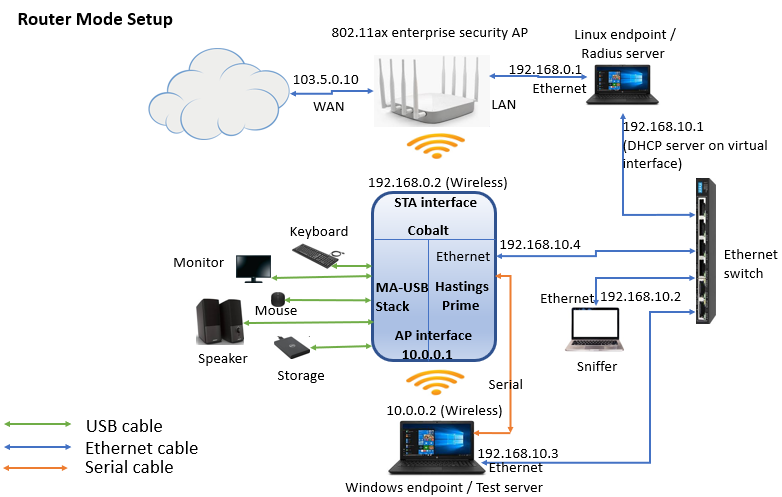
This document is a project specific user guide for use by Dell automation development projects. It provides guidance/instructions which is intended to assist the relevant stakeholder.

# Image compilation and flashing

# Test Setup

## Router mode

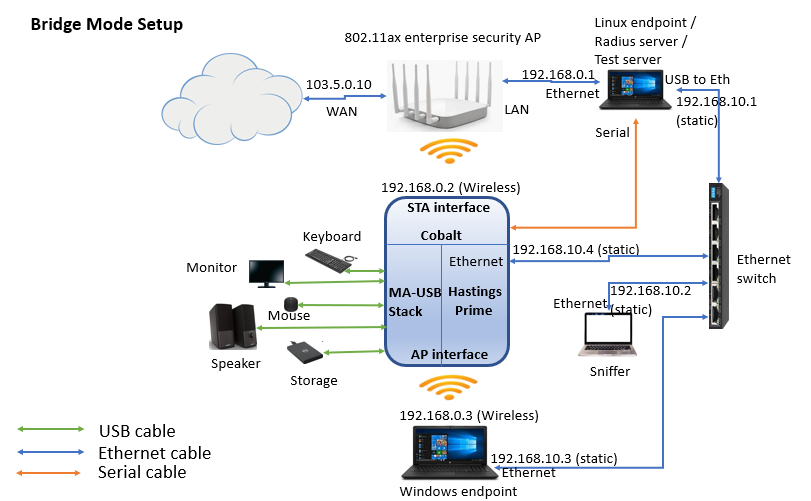
When DUT is in router mode, fronthaul endpoint gets the IP from the DHCP server running on AP interface of DUT



## Bridge mode

When DUT is in bridge mode, fronthaul endpoint gets the IP from the DHCP server running on backend AP.

A virtual interface is created on backend Linux endpoint and DHCP server runs on it. DUT ethernet interface, sniffer and Windows end point ethernet interface uses this DHCP server to get IP. IP forwarding rules are added in backend linux to reach the AP from test server. Cypress docking system is controlled by the test server using serial module and/or ethernet interface. Wireless interface of test server is used to associate with the fronthaul of docking station.



## Display-Link Mode

# Interface management

# Test Configurations

## Fronthaul configurations

### Configuring AP in open mode

1. Edit the configuration in /etc/config/wireless for the AP interface in the fronthaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Currently using AP in backhaul (ath0) as the AP on fronthaul (ath1) is being enumerated but cannot be scanned.

config wifi-device 'wifi0'

option type 'qcawificfg80211'

option channel '36'

option macaddr '04:f0:21:52:81:53'

option hwmode '11ac’

option disabled '0'

config wifi-iface

option device 'wifi0'

option network 'lan'

option mode 'ap'

option ssid 'Backhaul\_AP'

option encryption 'none'

config wifi-device 'wifi1'

option type 'qcawificfg80211'

option channel '1'

option macaddr '04:f0:21:3a:5b:ac'

option hwmode '11ng'

option disabled ‘0’

config wifi-iface

option device 'wifi1'

option network 'lan'

option mode 'ap'

option ssid 'Fronthaul\_AP'

option encryption 'none'

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Issue “iwconfig” command and ensure AP interface is configured in ‘master’ mode correctly.
4. Scan for the configured AP from a third-party station and connect.
5. As the DHCP server is enabled on the DUT, associated station shall get an IP address automatically.
6. Run ping and iperf between associated station and backend of the DUT for configured AP to verify successful connectivity. Ensure the same through sniffer as well.

### Configuring AP in WPA2 mode

1. Edit the configuration in /etc/config/wireless for the AP interface in the fronthaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Currently using AP in backhaul (ath0) as the AP on fronthaul (ath1) is being enumerated but cannot be scanned.

config wifi-device 'wifi0'

option type 'qcawificfg80211'

option channel '36'

option macaddr '04:f0:21:52:81:53'

option hwmode '11ac’

option disabled '0'

config wifi-iface

option device 'wifi0'

option network 'lan'

option mode 'ap'

option ssid 'Backhaul\_AP'

option encryption 'psk2+ccmp'

option key ‘1234567890’

config wifi-device 'wifi1'

option type 'qcawificfg80211'

option channel '1'

option macaddr '04:f0:21:3a:5b:ac'

option hwmode '11ng'

option disabled ‘0’

config wifi-iface

option device 'wifi1'

option network 'lan'

option mode 'ap'

option ssid 'Fronthaul\_AP'

option encryption 'psk2+ccmp'

option key ‘1234567890’

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Issue “iwconfig” command and ensure AP interface is configured in ‘master’ mode and security is set correctly.
4. Scan for the configured AP from a third-party station and connect by entering the configured password.
5. As the DHCP server is enabled on the DUT, associated station shall get an IP address automatically.
6. Run ping and iperf between associated station and backend of the DUT for configured AP to verify successful connectivity. Ensure the same through sniffer as well.

### Configuring AP in WPA3 mode

#### Opportunistic Wireless Encryption

#### Simultaneous Authentication of Equals

1. Edit the configuration in /etc/config/wireless for the AP interface in the fronthaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Currently using AP in backhaul (ath0) as the AP on fronthaul (ath1) is being enumerated but cannot be scanned.
* Reference: Wireless LAN Access Point (Driver Version 11.0) Command Reference Guide Section 8.40

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel '36'

        option macaddr '04:f0:21:52:81:53'

        option hwmode '11ac’

        option disabled '0'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'ap'

        option ssid 'WPA3\_AP'

        option encryption 'psk2+ccmp'

option key ‘1234567890’

option sae ‘1’

option sae\_password ‘1234567890’

option ieee80211w ‘2’

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel '1'

        option macaddr '04:f0:21:3a:5b:ac'

        option hwmode '11ng'

option disabled ‘0’

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'Fronthaul\_AP'

        option encryption 'psk2+ccmp'

option key ‘1234567890’

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Issue “iwconfig” command and ensure AP interface is configured in ‘master’ mode and security is set correctly.
4. Scan for the configured AP from a third-party station which supports SAE and connect by entering the configured password (Refer [Section 5.3.3](#_Configuring_opensource_wpa_supplica) for help on configuring wpa\_supplicant with WPA3 SAE support).
5. As the DHCP server is enabled on the DUT, associated station shall get an IP address automatically.
6. Run ping and iperf between associated station and backend of the DUT for configured AP to verify successful connectivity. Ensure the same through sniffer as well.

#### Device Provisioning Protocol

### WPS configuration

1. Edit the configuration in /etc/config/wireless for the AP interface in the fronthaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Only push-button method is verified for WPS
* Currently using AP in backhaul (ath0) as the AP on fronthaul (ath1) is being enumerated but cannot be scanned.

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel '36'

        option macaddr '04:f0:21:52:81:53'

        option hwmode '11ac’

        option disabled '0'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'ap'

        option ssid 'Backhaul\_AP'

        option encryption 'psk2+ccmp'

option key ‘1234567890’

option wps\_config ‘push\_button’

option wps\_pushbutton ‘1’

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel '1'

        option macaddr '04:f0:21:3a:5b:ac'

        option hwmode '11ng'

option disabled ‘0’

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'Fronthaul\_AP'

        option encryption 'psk2+ccmp'

option key ‘1234567890’

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Issue “iwconfig” command and ensure AP interface is configured in ‘master’ mode and security is set correctly.
4. Open hostapd\_cli by issuing the command “hostapd\_cli -i ath0 -p /var/run/hostapd-wifi0” and then give “wps\_pbc”.
5. On a WPS supported third party station, push button within 2 minutes and ensure the connection is established successfully.
6. As the DHCP server is enabled on the DUT, associated station shall get an IP address automatically.
7. Run ping and iperf between associated station and backend of the DUT for configured AP to verify successful connectivity. Ensure the same through sniffer as well.

### WFA Easy Connect

## Backhaul configurations

### Configuring STA to connect in open mode

1. Edit the configuration in /etc/config/wireless for the station interface in the backhaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel '36'

        option macaddr '04:f0:21:52:81:53'

        option hwmode '11ac'

        option htmode 'HE80'

        option disabled '0'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'ap'

        option ssid 'Backhaul\_AP'

        option encryption 'none'

        option extap '1'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'sta'

        option ssid 'Backhaul\_STA'

        option encryption 'none'

        option extap '1'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel '1'

        option macaddr '04:f0:21:3a:5b:ac'

        option hwmode '11ng'

option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'Fronthaul\_AP'

        option encryption 'none'

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Have a third-party AP configured for the SSID, channel and security method mentioned above for the station interface to which the DUT should be able to associate automatically.
4. Issue “iwconfig” command and ensure station interface ath01 is associated to the desired AP.
5. Remove the ath01 interface from the br-lan bridge using the command “brctl delif br-lan ath01” and confirm the interface is no longer bridged (“brctl show”), so that a static IP address can be assigned to the ath01 interface.
6. Run ping and iperf between backend of the third-party AP and station interface ath01 to verify successful connectivity. Ensure the same through sniffer as well.

### Configuring STA to connect in WPA2 mode

1. Edit the configuration in /etc/config/wireless for the station interface in the backhaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel '36'

        option macaddr '04:f0:21:52:81:53'

        option hwmode '11ac'

        option htmode 'HE80'

        option disabled '0'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'ap'

        option ssid 'Backhaul\_AP'

        option encryption 'none'

        option extap '1'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'sta'

        option ssid 'Backhaul\_STA'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option extap '1'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel '1'

        option macaddr '04:f0:21:3a:5b:ac'

        option hwmode '11ng'

option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'Fronthaul\_AP'

        option encryption 'none'

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Have a third-party AP configured for the SSID, channel and security method mentioned above for the station interface to which the DUT should be able to associate automatically.
4. Issue “iwconfig” command and ensure station interface ath01 is associated to the desired AP.
5. Remove the ath01 interface from the br-lan bridge using the command “brctl delif br-lan ath01” and confirm the interface is no longer bridged (“brctl show”), so that a static IP address can be assigned to the ath01 interface.
6. Run ping and iperf between backend of the third-party AP and station interface ath01 to verify successful connectivity. Ensure the same through sniffer as well.

### Configuring STA to connect in WPA3 mode

#### Opportunistic Wireless Encryption

#### Simultaneous Authentication of Equals

1. Edit the configuration in /etc/config/wireless for the station interface in the backhaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Reference: Wireless LAN Access Point (Driver Version 11.0) Command Reference Guide Section 8.40

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel '36'

        option macaddr '04:f0:21:52:81:53'

        option hwmode '11ac'

        option htmode 'HE80'

        option disabled '0'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'sta'

        option ssid 'WPA3\_STA'

        option encryption 'ccmp'

option sae ‘1’

        option sae\_password '1234567890'

        option extap '1'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel '1'

        option macaddr '04:f0:21:3a:5b:ac'

        option hwmode '11ng'

option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'Fronthaul\_AP'

        option encryption 'none'

1. Save the configuration and issue “wifi down” and “wifi up” commands for the new changes to take effect.
2. Issue “iw dev” and verify the interfaces are up and the configuration changes made are reflected.
3. Have a third-party AP configured for the SSID, channel and security method mentioned above for the station interface to which the DUT should be able to associate automatically (Refer to [Section 5.3.2](#_Configuring_opensource_hostapd) for bringing up hostapd configured in WAP3 SAE mode).
4. Issue “iwconfig” command and ensure station interface ath1 is associated to the desired AP.
5. Remove the ath1 interface from the br-lan bridge using the command “brctl delif br-lan ath01” and confirm the interface is no longer bridged (“brctl show”), so that a static IP address can be assigned to the ath1 interface.
6. Run ping and iperf between backend of the third-party AP and station interface ath1 to verify successful connectivity. Ensure the same through sniffer as well.

#### Device Provisioning Protocol

### WPS configuration

**NOTE:** Only push-button method is verified for WPS.

1. Edit the configuration file as per section 5.2.2.
2. Issue the command “wpa\_cli -i ath01 -p /var/run/wpa\_supplicant-ath01” to open the wpa\_cli utility for the station interface.
3. If the station is already connected to the third-party AP, issue “disconnect”.
4. Now start “push button” on the third-party AP either through the button on the AP (Ensure the third-party AP supports connection through WPS)
5. Within 2 minutes, issue “wps\_pbc” command on the wpa\_cli interface of the station and verify that station can associate successfully.
6. Remove the ath01 interface from the br-lan bridge using the command “brctl delif br-lan ath01” and confirm the interface is no longer bridged (“brctl show”), so that a static IP address can be assigned to the ath01 interface.
7. Run ping and iperf between backend of the third-party AP and station interface ath01 to verify successful connectivity. Ensure the same through sniffer as well.

### Configuring STA to connect in WPA2 enterprise mode

#### EAP-AKA

#### EAP-TLS

#### EAP-TTLS

#### EAP-MSCHAPv2

### Configuring STA to connect in WPA3 enterprise mode

#### Suite B

### Wi-Fi Certified Enhanced Open

## Concurrency

### Configuring DUT in router mode with both fronthaul and backhaul in wireless mode

1. Edit the configuration in /etc/config/wireless for enabling both the AP and station interfaces in the fronthaul and backhaul respectively as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board
* Note in the below example that both AP and STA interfaces are not added as part of network ‘lan’ so they will not be a part of bridge

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-iface

        option device 'wifi0’

        option mode 'ap'

        option ssid 'bandwidth160'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Run /etc/init.d/network restart so that fronthaul AP and backhaul STA are brought out of bridge
2. Verify by issuing “brctl show” that the bridge does not include ath0 (AP) and ath01 (STA) interfaces.
3. Bring down the bridge interface and delete the bridge
   1. ifconfig brctl down
   2. brctl delbr br-lan
4. Assign static IP address 192.168.1.1 to fronthaul AP interface ath0 and have the windows laptop connected to it and assign static IP 192.168.1.130
5. Connect backhaul sta interface ath01 to third party AP 192.168.2.1 and assign a static IP 192.168.2.100. Have a backend linux laptop connected to third party AP (192.168.2.120)
6. Apply the routing rules between AP (ath0) and STA(ath01) interfaces as below:

* iptables -F
* iptables -X
* iptables -A FORWARD -i ath01 -o ath0 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i ath0 -o ath01 -j ACCEPT
* iptables -A FORWARD -i ath0 -o ath01 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i ath01 -o ath0 -j ACCEPT
* iptables -t nat -A POSTROUTING -o ath01 -j MASQUERADE
* iptables -t nat -A POSTROUTING -o ath0 -j MASQUERADE

1. Ping from windows laptop 192.168.1.130 to fronthaul AP 192.168.1.1, STA interface 192.168.2.100, 3rd party AP 192.168.2.1 and its backend 192.168.2.120
2. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in bridge mode with both fronthaul and backhaul in wireless mode

1. Edit the configuration in /etc/config/wireless for enabling both the AP and station interfaces in the fronthaul and backhaul respectively as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-iface

        option device 'wifi0’

option network 'lan'

        option mode 'ap'

        option ssid 'bandwidth160'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

        option network 'lan'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Edit the network configuration file as per below example so that the bridge is configured to get the IP address from the DHCP server configured on the third party AP (ensure to take the backup of the default config file in /etc/config/network)

config interface 'loopback'

option ifname 'lo'

option proto 'static'

option ipaddr '127.0.0.1'

option netmask '255.0.0.0'

config globals 'globals'

option ula\_prefix 'auto'

config interface 'lan'

option ifname 'eth1'

option force\_link '1'

option type 'bridge'

option proto 'dhcp'

option ip6assign '60'

option multicast\_querier '0'

option igmp\_snooping '0'

config interface 'wan'

option ifname 'eth0'

option proto 'dhcp'

config interface 'wan6'

option ifname 'eth0'

option proto 'dhcpv6'

config switch

option name 'switch0'

option reset '1'

option enable\_vlan '1'

config switch\_vlan

option device 'switch0'

option vlan '1'

option ports '6 1 2 3 4'

config switch\_vlan

option device 'switch0'

option vlan '2'

option ports '0 5'

1. Run /etc/init.d/network restart so that fronthaul AP and backhaul STA are included in the bridge.
2. Verify by issuing “brctl show” that the bridge includes ath0 (AP) and ath01 (STA) interfaces.
3. Ensure the bridge is up and station interface of the dock is connected to third party AP on which DHCP is enabled.
4. Verify that the bridge on the DUT gets the IP address from the third party AP as the station interface is connected to third party AP.
5. Connect windows 10 laptop to the fronthaul AP and verify that it has acquired the IP address dynamically from the third party AP.
6. Ping from Windows laptop to the backend of the 3rd party AP.
7. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in router mode with fronthaul in wireless mode and backhaul in wired mode

1. Edit the configuration in /etc/config/wireless for enabling only the AP interface in the fronthaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

1. Note in the below example that AP interface is not added as part of network ‘lan’ so it will not be a part of bridge. Also note station interface is not being brought up.

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-iface

        option device 'wifi0’

        option mode 'ap'

        option ssid 'bandwidth160'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

1. Run /etc/init.d/network restart so that configuration takes effect.
2. Connect an ethernet cable between WAN port (eth0) of the DUT to the LAN port of the third party AP so as to form the wired backhaul.
3. Verify by issuing “brctl show” that the bridge does not include ath0 (AP) interface.
4. Bring down the bridge interface and delete the bridge
   1. ifconfig brctl down
   2. brctl delbr br-lan
5. Assign static IP address 192.168.1.1 to fronthaul AP interface ath0 and have the windows laptop connected to it and assign static IP 192.168.1.130
6. Ensure that the eth0 interface of the DUT has an IP address assigned by the DHCP server running in the third party AP in 192.168.2.x subnet
7. Apply the routing rules between AP (ath0) and WAN(eth0) interfaces as below:

* iptables -F
* iptables -X
* iptables -A FORWARD -i eth0 -o ath0 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i ath0 -o eth0 -j ACCEPT
* iptables -A FORWARD -i ath0 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i eth0 -o ath0 -j ACCEPT
* iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
* iptables -t nat -A POSTROUTING -o ath0 -j MASQUERADE

1. Ping from windows laptop 192.168.1.130 to fronthaul AP 192.168.1.1, eth0 interface 192.168.2.100, 3rd party AP 192.168.2.1 and its backend 192.168.2.120
2. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in bridge mode with fronthaul in wireless mode and backhaul in wired mode

1. Edit the configuration in /etc/config/wireless for enabling both the AP and WAN interfaces in the fronthaul and backhaul respectively as per the below example.  Backhaul wired configuration is in Step 2 below

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-iface

        option device 'wifi0’

option network 'lan'

        option mode 'ap'

        option ssid 'bandwidth160'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

1. Connect an ethernet cable between WAN port (eth0) of the DUT to the LAN port of the third party AP so as to form the wired backhaul.
2. Edit the network configuration file as per below example so that the bridge is configured to get the IP address from the DHCP server configured on the third party AP and eth0 interface is also part of the bridge (ensure to take the backup of the default config file in /etc/config/network)

config interface 'loopback'

option ifname 'lo'

option proto 'static'

option ipaddr '127.0.0.1'

option netmask '255.0.0.0'

config globals 'globals'

option ula\_prefix 'auto'

config interface 'lan'

option ifname 'eth1 eth0'

option force\_link '1'

option type 'bridge'

option proto 'dhcp'

option ip6assign '60'

option multicast\_querier '0'

option igmp\_snooping '0'

config interface 'wan'

config interface 'wan6'

option ifname 'eth0'

option proto 'dhcpv6'

config switch

option name 'switch0'

option reset '1'

option enable\_vlan '1'

config switch\_vlan

option device 'switch0'

option vlan '1'

option ports '6 1 2 3 4'

config switch\_vlan

option device 'switch0'

option vlan '2'

option ports '0 5'

1. Run /etc/init.d/network restart so that fronthaul AP and WAN interface eth0 are included in the bridge.
2. Verify by issuing “brctl show” that the bridge includes ath0 (AP) and eth0 (WAN) interfaces.
3. Ensure the bridge is up and it gets the IP address from the third party AP as the WAN interface eth0 is part of the bridge.
4. Connect windows 10 laptop to the fronthaul AP and verify that it has acquired the IP address dynamically from the third party AP.
5. Ping from Windows laptop to the backend of the 3rd party AP.
6. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in router mode with fronthaul in wired mode and backhaul in wireless mode

1. Edit the configuration in /etc/config/wireless for enabling only the STA interface in the backhaul as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

1. Note in the below example that STA interface is not added as part of network ‘lan’ so it will not be a part of bridge. Also note AP interface is not being brought up and hence STA interface comes up on ath0.

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Run /etc/init.d/network restart so that configuration takes effect.
2. Connect an ethernet cable between LAN port (eth1) of the DUT to the LAN port of the third party AP so as to form the wired fronthaul.
3. Verify by issuing “brctl show” that the bridge does not include ath0 (STA) interface.
4. Bring down the bridge interface and delete the bridge
5. ifconfig brctl down
6. brctl delbr br-lan
7. Have the STA interface ath0 connected to the third party AP so that it gets the IP address from the DHCP server running in the third party AP in 192.168.2.x subnet.
8. Ensure that the eth1 interface of the DUT has a static IP 192.168.1.1 and the windows laptop acting as its wired backend also has an IP in same subnet.
9. Apply the routing rules between STA(ath0) and LAN(eth1) interfaces as below:

* iptables -F
* iptables -X
* iptables -A FORWARD -i eth1 -o ath0 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i ath0 -o eth1 -j ACCEPT
* iptables -A FORWARD -i ath0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i eth1 -o ath0 -j ACCEPT
* iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
* iptables -t nat -A POSTROUTING -o ath0 -j MASQUERADE

1. Ping from windows laptop 192.168.1.130 to LAN interface of DUT 192.168.1.1, ath0 STA interface 192.168.2.100, 3rd party AP 192.168.2.1 and its backend 192.168.2.120
2. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in bridge mode with fronthaul in wired mode and backhaul in wireless mode

1. Edit the configuration in /etc/config/wireless for enabling both the STA and LAN interfaces in the bakhaul and fronthaul respectively as per the below example.  Fronthaul wired configuration is in Step 2 below

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '0'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

option network 'lan'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Edit the network configuration file as per below example so that the bridge is configured to get the IP address from the DHCP server configured on the third party AP as the STA interface ath0 is part of the bridge and so is lan interface eth1 (ensure to take the backup of the default config file in /etc/config/network)

config interface 'loopback'

option ifname 'lo'

option proto 'static'

option ipaddr '127.0.0.1'

option netmask '255.0.0.0'

config globals 'globals'

option ula\_prefix 'auto'

config interface 'lan'

option ifname 'eth1'

option force\_link '1'

option type 'bridge'

option proto 'dhcp'

option ip6assign '60'

option multicast\_querier '0'

option igmp\_snooping '0'

config interface 'wan'

config interface 'wan6'

option ifname 'eth0'

option proto 'dhcpv6'

config switch

option name 'switch0'

option reset '1'

option enable\_vlan '1'

config switch\_vlan

option device 'switch0'

option vlan '1'

option ports '6 1 2 3 4'

config switch\_vlan

option device 'switch0'

option vlan '2'

option ports '0 5'

1. Run /etc/init.d/network restart so that backhaul STS and LAN interface eth1 are included in the bridge.
2. Verify by issuing “brctl show” that the bridge includes ath0 (STA) and eth1 (LAN) interfaces.
3. Ensure the bridge is up and it gets the IP address from the third party AP by having the backhaul STA interface connected to it.
4. Connect windows 10 laptop as the wired interface to the LAN port eth1 of the DUT and verify that it has acquired the IP address dynamically from the third party AP.
5. Ping from Windows laptop to the backend of the 3rd party AP.
6. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in router mode with both fronthaul and backhaul in wired mode

1. Edit the configuration in /etc/config/wireless for disabling both AP and the STA interfaces as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '1'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Run /etc/init.d/network restart so that configuration takes effect and no wifi interfaces are up.
2. Connect an ethernet cable between WAN port (eth0) of the DUT to the LAN port of the third party AP so as to form the wired backhaul. Also, it gets the IP from the third party AP in 192.168.2.x subnet
3. Connect an ethernet cable between LAN port (eth1) of the DUT to the LAN port of the third party AP so as to form the wired fronthaul.
4. Ensure that the eth1 interface of the DUT has a static IP 192.168.1.1 and the windows laptop acting as its wired backend also has an IP in same subnet.
5. Apply the routing rules between WAN(eth0) and LAN(eth1) interfaces as below:

* iptables -F
* iptables -X
* iptables -A FORWARD -i eth1 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
* iptables -A FORWARD -i eth0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
* iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
* iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
* iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

1. Ping from windows laptop 192.168.1.130 to LAN interface of DUT 192.168.1.1, eth0 WAN interface 192.168.2.100, 3rd party AP 192.168.2.1 and its backend 192.168.2.120
2. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

### Configuring DUT in bridge mode with both fronthaul and backhaul in wired mode

1. Edit the configuration in /etc/config/wireless for disabling both the STA and AP interfaces in the bakhaul and fronthaul respectively as per the below example.

NOTE:

* Ensure to take a backup of the current configuration
* Change the MAC address as per the board

config wifi-device 'wifi0'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:52:81:4a'

        option hwmode '11ac'

        option disabled '1'

config wifi-device 'wifi1'

        option type 'qcawificfg80211'

        option channel 'auto'

        option macaddr '04:f0:21:3a:5b:98'

        option hwmode '11ng'

        option disabled '0'

config wifi-iface

        option device 'wifi1'

        option network 'lan'

        option mode 'ap'

        option ssid 'OpenWrt'

        option encryption 'none'

config wifi-iface

        option device 'wifi0'

option network 'lan'

        option mode 'sta'

        option extap '1'

        option ssid 'Enterprise\_AP'

        option encryption 'psk2+ccmp'

        option key '1234567890'

        option sae '0'

        option sae\_password 'none'

1. Connect an ethernet cable between WAN port (eth0) of the DUT to the LAN port of the third party AP so as to form the wired backhaul.
2. Connect an ethernet cable between LAN port (eth1) of the DUT to the LAN port of the third party AP so as to form the wired fronthaul.
3. Edit the network configuration file as per below example so that the bridge is configured to get the IP address from the DHCP server configured on the third party AP as the WAN interface eth0 is part of the bridge and so is lan interface eth1 (ensure to take the backup of the default config file in /etc/config/network)

config interface 'loopback'

option ifname 'lo'

option proto 'static'

option ipaddr '127.0.0.1'

option netmask '255.0.0.0'

config globals 'globals'

option ula\_prefix 'auto'

config interface 'lan'

option ifname 'eth0 eth1'

option force\_link '1'

option type 'bridge'

option proto 'dhcp'

option ip6assign '60'

option multicast\_querier '0'

option igmp\_snooping '0'

config interface 'wan'

config interface 'wan6'

option ifname 'eth0'

option proto 'dhcpv6'

config switch

option name 'switch0'

option reset '1'

option enable\_vlan '1'

config switch\_vlan

option device 'switch0'

option vlan '1'

option ports '6 1 2 3 4'

config switch\_vlan

option device 'switch0'

option vlan '2'

option ports '0 5'

1. Run /etc/init.d/network restart so that backhaul STS and LAN interface eth1 are included in the bridge.
2. Verify by issuing “brctl show” that the bridge includes eth0 (WAN) and eth1 (LAN) interfaces.
3. Ensure the bridge is up and it gets the IP address from the third party AP by having the wired backhaul connected to it.
4. Connect windows 10 laptop as the wired interface to the LAN port eth1 of the DUT and verify that it has acquired the IP address dynamically from the third party AP.
5. Ping from Windows laptop to the backend of the 3rd party AP.
6. For end to end iperf traffic, start iperf client on the windows laptop and iperf server on the backend of the third party AP. Use “-R” option on the iperf client to verify traffic in reverse mode (Client receives and server sends traffic)

## Miscellaneous configurations

### FreeRADIUS server bring up and configuration

1. Download the latest version of the FreeRadius server (current version 3.0.21) from <https://freeradius.org/releases/>
2. Untar the file:
   1. tar -xvf freeradius-server-3.0.21.tar.gz
   2. cd freeradius-server-3.0.21
3. Download dependencies required:
   1. apt-get install libssl-dev
   2. apt-get install linux-headers-$(uname -r)
   3. apt-get install libtalloc-dev -y
4. Follow below steps to install freeradius
   1. ./configure
   2. make
   3. make install
5. All the configuration files required for free radius will be generated in directory **/usr/local/etc/raddb** generated post installation
   1. cd /usr/local/etc/raddb
6. Below three files need to be configured in raddb directory:
   1. clients.conf
   2. users
   3. modules-enabled/eap
7. **Edit the clients.conf file to add the IP address and secret for the authenticator as below:**
8. # Add below lines at the last.
9. # Give ip address of Authenticator and seceret which is shared b/w Authenticator and server.
10. client 192.168.3.11 {
11. ipaddr = 192.168.3.11
12. secret = AuthPassword
13. }
14. **2. vim users : Username and Password for wpa\_supplicants**
15. # Uncomment the following line
16. bob Cleartext-Password := "testing123"
17. Reply-Message := "Hello, %{User-Name}"
19. N**ote** : If any spaces are there fot Username include it in double quotes.
20. This is the Username and Password which will be given on supplicant
21. Change Username and Password according to your need
23. **3. vim eap.conf**
25. For eap configuration- select the method whichever you want.
26. Note-1 : In eap change default-eap-type to ttls or peap or tls
27. Note-2 : In ttls and peap make sure deafult-eap-type should to mschapv2

### Configuring opensource hostapd in WPA3 SAE mode

1. Get the latest version hostapd-2.9
2. Enable the below configurations in .config (copy of defconfig)

CONFIG\_IEEE80211N=y

CONFIG\_IEEE80211AC=y

CONFIG\_SAE=y

1. For the successful compilation of hostapd, the following packages need to be installed (use apt-get install)

pkg-config

libnl-3-dev

libssl-dev

libnl-genl-3-dev

libdbus-1-dev

1. Compile hostapd (make).
2. Edit or create hostapd.conf with the below configuration

driver=nl80211

interface=wlan0

ssid=WPA3\_AP

hw\_mode=a

ieee80211ac=1

channel=36

wpa=2

wpa\_passphrase=1234567890

wpa\_key\_mgmt=SAE

rsn\_pairwise=CCMP

ieee80211w=2

1. Ensure to kill wpa\_supplicant and stop network-manager before starting hostapd.
2. Start hostpad using the command “./hostapd hostapd.conf -dd -K”

NOTE:

If hostapd cannot bring up AP in 5 GHz band, make sure that the regulatory domain is set correctly. To verify this, issue “iw list” and check for frequencies for wlan interface. If “no IR” is observed for 5 GHz channels, change the country to US by using the command “iw reg set US”.

### Configuring opensource wpa\_supplicant to associate to AP in WPA3 SAE mode

1. Get the latest version wpa\_supplicant-2.9
2. Enable the below configuration in .config (copy of defconfig)

CONFIG\_SAE=y

1. For the successful compilation of wpa\_supplicant, the following packages need to be installed (use apt-get install)

pkg-config

libnl-3-dev

libssl-dev

libnl-genl-3-dev

libdbus-1-dev

1. Compile wpa\_supplicant (make).
2. Edit or create wpa\_supplicant.conf with the below configuration

ctrl\_interface=/var/run/wpa\_supplicant

update\_config=1

network={

ssid=”WPA3\_AP”

psk=”1234567890”

key\_mgmt=SAE

ieee80211w=2

}

1. Ensure to kill wpa\_supplicant and stop network-manager before starting wpa\_supplicant.
2. Start wpa-supplicant using the command “./wpa\_supplicant -D nl80211 -i wlan0 -c wpa\_supplicant.conf -dd -K”

# Test Automation

## Description

This is a generic test automation framework developed by GlobalEdge in Python v3.6.

TODO: Need to add

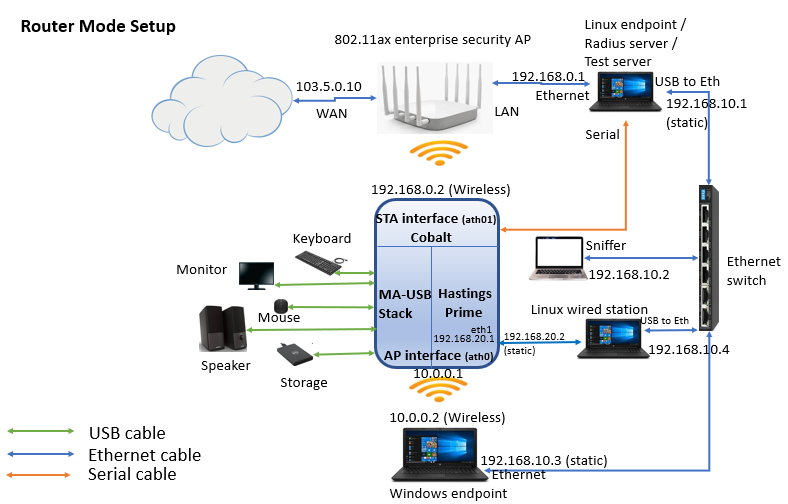
This document serves as a both user guide and release notes.

As a user guide, this document briefs about standalone automation test bed, system requirements, setup configuration and execution.

As a release notes, this document details about test automation framework releases including release version, supported features, limitations and JIRA status.

## Automation Testbed

### Concurrent in router mode in WPA2 mode

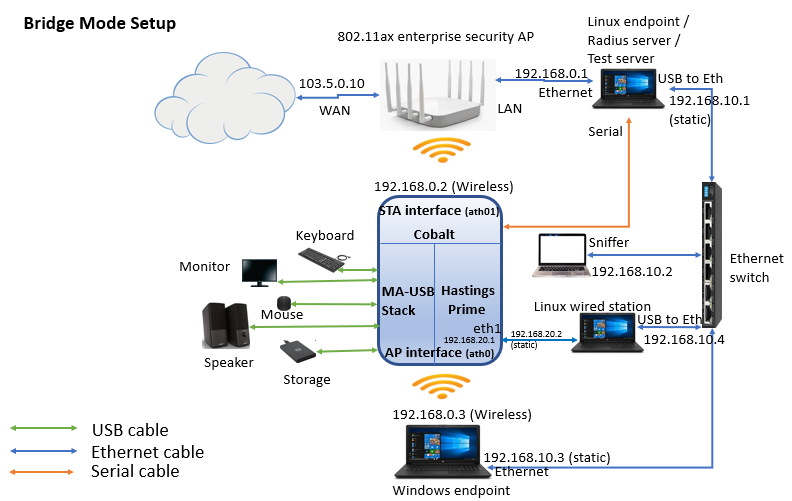


* Configure DUT AP interface WPA2 mode
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Set channel on wifi0 interface
    - uci set wireless.wifi0.channel="36’
  + Set SSID on AP interface
    - uci set wireless.@wifi-iface[0].ssid="Fronthaul\_SSID"
  + Configure security type
    - uci set wireless.@wifi-iface[0].encryption="psk+ccmp"
  + Configure password
    - uci set wireless.@wifi-iface[0].key="1234567890"
  + Remove any WPA3 specific configurations
    - uci set wireless.@wifi-iface[0].extap="1"
    - uci delete wireless.@wifi-iface[0].sae
    - uci delete wireless.@wifi-iface[0].sae\_password
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate configured SSID is beaconing after network restart
* Configure 3rd party AP in WPA2 mode

Use browser to configure 3rd party AP in WPA2 security mode

* Configure DUT STA interface in WPA2 mode and associate to 3rd party AP
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Bring up new virtual interface to configure as STA
    - uci add wireless wifi-iface
  + Bring newly created interface under wifi0. Note the index 2
    - uci set [wireless.@wifi-iface[2].device=wifi0](mailto:wireless.@wifi-iface[2].device=wifi0)
  + Add to lan option
    - uci set [wireless.@wifi-iface[2].network=lan](mailto:wireless.@wifi-iface[2].network=lan)
  + Configure interface in STA mode
    - uci set [wireless.@wifi-iface[2].mode=sta](mailto:wireless.@wifi-iface[2].mode=sta)
  + Configure the 3rd party AP SSID
    - uci set [wireless.@wifi-iface[2].ssid="AP\_SSID](mailto:wireless.@wifi-iface[2].ssid=%22AP_SSID)"
  + Configure WPA2 mode
    - uci set [wireless.@wifi-iface[2].encryption="psk2+ccmp](mailto:wireless.@wifi-iface[2].encryption=%22psk2+ccmp)"
  + Configure 3rd party AP password
    - uci set [wireless.@wifi-iface[2].key="1234567890](mailto:wireless.@wifi-iface[2].key=%221234567890)"
    - uci set [wireless.@wifi-iface[2].extap="1](mailto:wireless.@wifi-iface[2].extap=%221)"
  + Cleanup if any WPA3 setting is present
    - uci delete [wireless.@wifi-iface[2].sae](mailto:wireless.@wifi-iface[2].sae)
    - uci delete [wireless.@wifi-iface[2].sae\_password](mailto:wireless.@wifi-iface[2].sae_password)
  + Commit and restart the network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate DUT station interface ‘ath01’ is up and associated to 3rd party AP using iwconfig command. It may take around 2 minutes for ath01 to come up and associate to 3rd party AP.
* Configure router mode in DUT
  + Enable bridge br-wan by adding STA interface under ‘wan’ section in /etc/config/network
    - uci set network.wan.ifname=ath01
    - uci set network.wan.type=bridge
    - uci set network.wan.proto=dhcp
  + Commit and restart network
    - uci commit network
    - /etc/init.d/network restart
  + Wait for backhaul sta is associated to 3rd party AP after network restart and gets IP address for br-wan interface
* Associate Windows station to AP interface of DUT
* Ping from Windows to AP interface of DUT
* Ping from Windows to AP and AP backend
* Run uplink traffic from Windows station to AP backend (Downlink traffic is not working as unable to ping from AP endpoint to Windows)

### Concurrent in bridge mode in WPA2 mode



* Configure DUT AP interface WPA2 mode
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Set channel on wifi0 interface
    - uci set wireless.wifi0.channel="36’
  + Set SSID on AP interface
    - uci set wireless.@wifi-iface[0].ssid="Fronthaul\_SSID"
  + Configure security type
    - uci set wireless.@wifi-iface[0].encryption="psk2+ccmp"
  + Configure password
    - uci set wireless.@wifi-iface[0].key="1234567890"
  + Remove any WPA3 specific configurations
    - uci set wireless.@wifi-iface[0].extap="1"
    - uci delete wireless.@wifi-iface[0].sae
    - uci delete wireless.@wifi-iface[0].sae\_password
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate configured SSID is beaconing after network restart
* Configure 3rd party AP in WPA2 mode

Use browser to configure 3rd party AP in WPA2 security mode

* Configure DUT STA interface in WPA2 mode and associate to 3rd party AP
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Bring up new virtual interface to configure as STA
    - uci add wireless wifi-iface
  + Bring newly created interface under wifi0. Note the index 2
    - uci set [wireless.@wifi-iface[2].device=wifi0](mailto:wireless.@wifi-iface[2].device=wifi0)
  + Add to lan option
    - uci set [wireless.@wifi-iface[2].network=lan](mailto:wireless.@wifi-iface[2].network=lan)
  + Configure interface in STA mode
    - uci set [wireless.@wifi-iface[2].mode=sta](mailto:wireless.@wifi-iface[2].mode=sta)
  + Configure the 3rd party AP SSID
    - uci set [wireless.@wifi-iface[2].ssid="AP\_SSID](mailto:wireless.@wifi-iface[2].ssid=%22AP_SSID)"
  + Configure WPA2 mode
    - uci set [wireless.@wifi-iface[2].encryption="psk2+ccmp](mailto:wireless.@wifi-iface[2].encryption=%22psk2+ccmp)"
  + Configure 3rd party AP password
    - uci set [wireless.@wifi-iface[2].key="1234567890](mailto:wireless.@wifi-iface[2].key=%221234567890)"
    - uci set [wireless.@wifi-iface[2].extap="1](mailto:wireless.@wifi-iface[2].extap=%221)"
  + Cleanup if any WPA3 setting is present
    - uci delete [wireless.@wifi-iface[2].sae](mailto:wireless.@wifi-iface[2].sae)
    - uci delete [wireless.@wifi-iface[2].sae\_password](mailto:wireless.@wifi-iface[2].sae_password)
  + Commit and restart the network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate DUT station interface ‘ath01’ is up and associated to 3rd party AP using iwconfig command. It may take around 2 minutes for ath01 to come up and associate to 3rd party AP.
* Configure bridge mode in DUT
  + Validate AP and STA interface are under bridge
    - brctl show br-lan
  + Assign DHCP to bridge
    - uci set network.lan.proto=dhcp
  + Remove static IP, netmask from bridge
    - uci delete network.lan.ipaddr
    - uci delete network.lan.netmask
  + Commit and restart network
    - uci commit network
    - /etc/init.d/network restart
  + Wait for backhaul sta is associated to 3rd party AP after network restart
  + Wait for DHCP IP to STA interface ath01 from 3rd party AP
* Associate Windows station to AP interface of DUT. Validate Windows will get IP from 3rd party AP (IP should be in 3rd party AP subnet)
* Ping from Windows to AP interface of DUT
* Ping from Windows to AP and AP backend
* Run traffic from Windows station to AP backend

### Fronthaul only validation

* Configure DUT AP interface WPA2 mode
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Set channel on wifi0 interface
    - uci set wireless.wifi0.channel="36’
  + Set SSID on AP interface
    - uci set wireless.@wifi-iface[0].ssid="Fronthaul\_SSID"
  + Configure security type
    - uci set wireless.@wifi-iface[0].encryption="psk+ccmp"
  + Configure password
    - uci set wireless.@wifi-iface[0].key="1234567890"
  + Remove any WPA3 specific configurations
    - uci set wireless.@wifi-iface[0].extap="1"
    - uci delete wireless.@wifi-iface[0].sae
    - uci delete wireless.@wifi-iface[0].sae\_password
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate configured SSID is beaconing after network restart
* Connect a linux wired backend to DUT and assign static IP in 3rd party AP subnet (*192.168.0.131*)
* Configure DUT in bridge
  + Remove network lan option for AP interface from /etc/config/wireless
    - uci delete [wireless.@wifi-iface[0].network](mailto:wireless.@wifi-iface[0].network)
  + Bring up the bridge in 3rd party AP subnet instead of default 192.168.1.1
    - uci set network.lan.ipaddr=*192.168.0.10*
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate bridge gets the IP as 192.168.0.10
    - ifconfig br-lan
  + Ping from DUT to DUT backend
* Associate Windows to AP interface of DUT. Windows should get IP in 192.168.0.X.
* Ping from Windows to DUT backend
* Run traffic between Windows and DUT backend

### Backhaul only validation

* Configure 3rd party AP in WPA2 mode

Use browser to configure 3rd party AP in WPA2 security mode

* Configure DUT STA interface in WPA2 mode and associate to 3rd party AP
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Bring up new virtual interface to configure as STA
    - uci add wireless wifi-iface
  + Bring newly created interface under wifi0. Note the index 2
    - uci set [wireless.@wifi-iface[2].device=wifi0](mailto:wireless.@wifi-iface[2].device=wifi0)
  + Add to lan option
    - uci set [wireless.@wifi-iface[2].network=lan](mailto:wireless.@wifi-iface[2].network=lan)
  + Configure interface in STA mode
    - uci set [wireless.@wifi-iface[2].mode=sta](mailto:wireless.@wifi-iface[2].mode=sta)
  + Configure the 3rd party AP SSID
    - uci set [wireless.@wifi-iface[2].ssid="AP\_SSID](mailto:wireless.@wifi-iface[2].ssid=%22AP_SSID)"
  + Configure WPA2 mode
    - uci set [wireless.@wifi-iface[2].encryption="psk+ccmp](mailto:wireless.@wifi-iface[2].encryption=%22psk+ccmp)"
  + Configure 3rd party AP password
    - uci set [wireless.@wifi-iface[2].key="1234567890](mailto:wireless.@wifi-iface[2].key=%221234567890)"
    - uci set [wireless.@wifi-iface[2].extap="1](mailto:wireless.@wifi-iface[2].extap=%221)"
  + Cleanup if any WPA3 setting is present
    - uci delete [wireless.@wifi-iface[2].sae](mailto:wireless.@wifi-iface[2].sae)
    - uci delete [wireless.@wifi-iface[2].sae\_password](mailto:wireless.@wifi-iface[2].sae_password)
  + Commit and restart the network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate DUT station interface ‘ath01’ is up and associated to 3rd party AP using iwconfig command. It may take around 2 minutes for ath01 to come up and associate to 3rd party AP.
* Connect a linux wired backend to DUT and assign static IP in 3rd party AP subnet (*192.168.0.131*)
* Configure DUT in bridge
  + Remove network lan option for STA interface from /etc/config/wireless
    - uci delete [wireless.@wifi-iface[2].network](mailto:wireless.@wifi-iface[2].network)
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Delete br-lan interface
    - ifconfig br-lan down
    - brctl delbr br-lan
  + Create bridge between STA interface (ath01) and DUT backend (eth1)
    - brctl addbr br-sta
    - brctl addif br-sta ath01
    - brctl addif br-sta eth1
    - brctl show
  + Assign IP to bridge br-sta in 3rd party AP subnet
    - ifconfig br-sta *192.168.0.10* up
  + Ping from DUT to DUT backend
* Assign static IP to the linux laptop connected to AP backend in 3rd party AP subnet.
* Ping from DUT backend to AP backend
* Run traffic between Windows and DUT backend

### Concurrent in bridge mode with wired backhaul and WPA2 fronthaul

* Configure DUT AP interface WPA2 mode
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Set channel on wifi0 interface
    - uci set wireless.wifi0.channel="36’
  + Set SSID on AP interface
    - uci set wireless.@wifi-iface[0].ssid="Fronthaul\_SSID"
  + Configure security type
    - uci set wireless.@wifi-iface[0].encryption="psk2+ccmp"
  + Configure password
    - uci set wireless.@wifi-iface[0].key="1234567890"
  + Remove any WPA3 specific configurations
    - uci set wireless.@wifi-iface[0].extap="1"
    - uci delete wireless.@wifi-iface[0].sae
    - uci delete wireless.@wifi-iface[0].sae\_password
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate configured SSID is beaconing after network restart
* Connect DUT eth0 port to 3rd party AP and validate it gets IP from AP
* Configure bridge mode in DUT
  + Add eth0 to bridge
    - uci set network.lan.ifname=eth0
  + Delete eth0 interface from wan
    - uci delete network.wan.ifname
    - uci delete network.wan.proto
  + Commit and restart network
    - uci commit network
    - uci commit network
    - /etc/init.d/network restart
  + Validate AP and STA interface are under bridge
    - brctl show br-lan
  + Assign DHCP to bridge
    - uci set network.lan.proto=dhcp
  + Remove static IP, netmask from bridge
    - uci delete network.lan.ipaddr
    - uci delete network.lan.netmask
  + Commit and restart network
    - uci commit network
    - /etc/init.d/network restart
  + Wait for br-lan gets IP from 3rd party AP after network restart
  + Wait for DHCP IP to STA interface ath01 from 3rd party AP
* Associate Windows station to AP interface of DUT. Validate Windows will get IP from 3rd party AP (IP should be in 3rd party AP subnet)
* Ping from Windows to AP interface of DUT
* Ping from Windows to AP and AP backend

Run traffic from Windows station to AP backend

### Concurrent in router mode with wired backhaul and WPA2 fronthaul

* Configure DUT AP interface WPA2 mode
  + Enable WiFi interface
    - uci delete wireless.wifi0.disabled
  + Set channel on wifi0 interface
    - uci set wireless.wifi0.channel="36’
  + Set SSID on AP interface
    - uci set wireless.@wifi-iface[0].ssid="Fronthaul\_SSID"
  + Configure security type
    - uci set wireless.@wifi-iface[0].encryption="psk+ccmp"
  + Configure password
    - uci set wireless.@wifi-iface[0].key="1234567890"
  + Remove any WPA3 specific configurations
    - uci set wireless.@wifi-iface[0].extap="1"
    - uci delete wireless.@wifi-iface[0].sae
    - uci delete wireless.@wifi-iface[0].sae\_password
  + Commit and restart network
    - uci commit wireless
    - /etc/init.d/network restart
  + Validate configured SSID is beaconing after network restart
* Connect ethernet cable from DUT eth0 interface to 3rd party AP
* Configure router mode in DUT
  + Add eth0 from wan and enable br-wan
    - uci set network.wan.ifname=eth0
    - uci set network.wan.proto=dhcp
    - uci set network.wan.type=bridge
  + Assign AP interface ‘atho’ to lan
    - uci set network.lan.ifname=’ath0’
  + Commit and restart network
    - uci commit network
    - /etc/init.d/network restart
  + Wait for backhaul sta is associated to 3rd party AP after network restart and br-wan gets IP address
* Associate Windows station to AP interface of DUT
* Ping from Windows to AP interface of DUT
* Ping from Windows to AP and AP backend
* Run uplink traffic from Windows station to AP backend (Downlink traffic is not working as unable to ping from AP endpoint to Windows)

### Concurrent in bridge mode with wired backhaul and fronthaul

* Connect ethernet cable from DUT WAN port to 3rd party AP LAN port
* Connect ethernet cable from DUT LAN port to Windows PC
* Configure bridge mode
  + Remove br-wan
    - uci delete network.wan.ifname
    - uci delete network.wan.type
    - uci delete network.wan.proto
  + Add DUT wired STA interface under br-lan
    - uci set network.lan.ifname=’eth0 eth1’
    - uci set network.lan.proto=dhcp
    - uci delete network.lan.ipaddr
    - uci delete network.lan.netmask
  + Commit and restart network
    - uci commit network
    - /etc/init.d/network restart
* Wait for br-lan and Windows PC gets IP from 3rd party AP
* Ping from Windows PC to 3rd party AP and AP backend
* Run traffic between Windows PC and AP backend

### Concurrent in router mode with wired backhaul and fronthaul

* Connect ethernet cable from DUT WAN port to 3rd party AP LAN port
* Connect ethernet cable from DUT LAN port to Windows PC
* Configure router mode
  + Enable br-wan
    - uci set network.wan.ifname=eth0
    - uci set network.wan.proto=dhcp
    - uci set network.wam.type=bridge
  + Enable br-lan
    - uci set network.lan.ifname=eth1
    - uci set network.wan.proto=static
    - uci set network.wam.type=bridge
    - uci set network.lan.ipaddr=192.168.1.1
    - uci set network.lan.netmask=255.255.255.0
  + Commit and restart network
    - uci commit network
    - /etc/.init.d/network restart
* Wait for br-wan gets IP from 3rd party AP and Windows PC gets IP from DUT
* Ping from Windows PC to 3rd party AP and AP backend
* Run traffic between Windows PC and AP backend

## Source Code Directory structure

TODO: NEED TO ADD

## System Requirements

Mandatory steps to be followed before automation execution,

1. Read all the sections from this release notes thoroughly and get clarified from the automation dev. engineer on any doubts, before automation execution starts.
2. Open internet connection is required for the test controller to install software packages during setup/testbed configuration. Refer section #1.6 for more information.

Below table shows required hardware with software to be install in detail for cypress docking station automation setup construction. Refer section #1.3 and #1.4

|  |  |  |
| --- | --- | --- |
| Sl. No. | Test Bed Devices | Hardware specifications |
| 1 | Test Controller | X86/Linux 64bit, 8GB RAM, 500GB HDD with Linux-Ubuntu 18.04 |
| 2 | Sniffer machine |  |
| 3 | Access Point |  |
| 4 | Linux Endpoint | Laptop with Linux OS |
| 5 | Windows Endpoint | Laptop with Windows 10 |
| 6 | Ethernet Switch (>=8 port) |  |
| 7 | Power adapter |  |
| 8 | Type C to multi-function LAN adapter (White)  Type C to 3 USB adapter |  |

## Software Installation And Configuration

|  |  |
| --- | --- |
| **Ubuntu OS installation and configuration for TestController** | |
| Hardware | x86/Linux 64bit |
| Software | 18.04-Ubuntu |
| Configuration | 1. Install Ubuntu OS 18.04 2. Install system packages    1. sudo apt-get install vim    2. sudo apt-get install python3.6    3. sudo apt-get install python3-pip    4. sudo apt-get install firefox (version should be >= 67.0.4)    5. sudo apt-get install minicom    6. sudo apt-get install sshpass    7. sudo apt-get install iperf    8. sudo apt-get install tshark (version should be = 2.6.8) 3. Install python modules    1. sudo pip3 install paramiko    2. sudo pip3 install pymongo    3. sudo pip3 install colorlog    4. sudo pip3 install openpyxl    5. sudo pip3 install xmltodict    6. sudo pip3 install robotframework 4. Geckodriver (web driver for firefox browser) installation,    1. Go to url - https://github.com/mozilla/geckodriver/releases    2. Download geckodriver-v0.24.0-linux64.tar.gz    3. Untar the package, # tar -xvf geckodriver-v0.18.0-linux64.tar.gz    4. Move gecodriver to bin, # sudo mv geckodriver /usr/bin    5. Use the below command to check the version,   #geckodriver --version |

|  |  |
| --- | --- |
| Linux EndPoint configuration and Installation | |
| Hardware | x86/Linux 64bit laptop |
| Software |  |
| Configuration | 1. Install python3.7    1. sudo apt-get install python3.7 2. Install system packages    1. sudo apt-get install python3-pip    2. sudo apt-get install xterm 3. Install python modules    1. sudo pip3 install netifaces    2. sudo pip3 install xmltodict    3. sudo pip3 install colorlog    4. sudo pip3install selenium 4. Install utilities    1. sudo apt-get install ping    2. sudo apt-get install fping    3. sudo apt-get install iperf    4. sudo apt-get install iperf3 5. Install firefox browser and geckodriver    1. Download Firefox package - firefox-65.0.tar.bz2   URL: <https://www.mozilla.org/en-US/firefox/65.0/system-requirements/>   * 1. Untar the package - $tar xvz firefox-65.0.tar.bz2   2. Move the file to bin - $sudo mv firefox/ /usr/bin/firefox  1. Install the geckodriver    1. Download the geckodriver from internet    2. URL: <https://github.com/mozilla/geckodriver>    3. Untar the package and move to bin folder,   $ sudo mv geckodriver /usr/bin |

|  |  |
| --- | --- |
| Windows EndPoint configuration and Installation | |
| Hardware | Windows laptop |
| Software |  |
| Configuration | 1. Install python3.7    1. Go to URL- <https://www.python.org/downloads/windows/>    2. Under Python-3.7.4 check for the suitable package   \* If 64-bit Download "Windows x86-64 executable installer"  \* If 32-bit Download "Windows x86 executable installer"   1. Install Iperf3    1. Go to URL- <https://iperf.fr/iperf-download.php#windows>    2. Dowload iperf-3.1.3    3. Extract and run as administrator    4. Add path to environment    5. Open new command prompt and enter iperf3    6. If Iperf3 got successfully installed,then you can see the man page of iperf3 2. TODO:Need to add steps to install wtee package |

|  |  |
| --- | --- |
| **Sniffer machine configuration** | |
| Hardware |  |
| Software |  |
| Configuration | Wireshark installation  Go to - <https://www.wireshark.org/download.html>  Go to – stable release (3.0.2) and click on ‘[macOS 10.12 and later Intel 64-bit .dmg](https://2.na.dl.wireshark.org/osx/Wireshark%203.0.2%20Intel%2064.dmg)’ package and save it.  Right click on Wireshark.3.0.2 Intel 64.dmg and extract it.  After extract double click on Wireshark 3.0.2 Intel 64.pkg file  Follow installation steps.  Cherrypy software package installation   * 1. Install six software package by download six-1.12.0.tar.gz from [https://pypi.org/project/six/#files](https://pypi.org/project/six/#_blank) path and extract.   2. Go to folder and execute setup.py to install using below command   # python setup.py install   * 1. sudo pip install cherrypy --ignore-installed six |

## Configuration and execution Procedure

### Test bed preparation

TODO

### Download and Install Automation Framework

TODO

### Cypress Docking Station Automation Testbed Configuration

Cypress docking station automation testbed configuration is carried out via .xml format. All .xml configurations are found at **TODO :Need to add .xml path**

There are separate .xml file is created. This name should not be changed.

Latest working .xml files is embedded below, **TODO:What to keep in this ?**

1. Copy the Controller folder into TestServer PC as mentioned in the diagram.

2. Copy the EndPoints folder in respective endpoints laptop.

3. Update all the available TestBed device details in resource manager database.

4. Below is the detailed steps for configuration of xml file:

5. Configure Mangodb server

To Create ROLE

use Resource\_manager

db.createRole(

{

role: "Gwats",

privileges: [

{ resource: { db: "Resource\_manager", collection: "" }, actions:

[ "find", "update", "insert"] },

],

roles: [

{ role: "read", db: "Resource\_manager" }

]

},

{ w: "majority" , wtimeout: 5000 }

)

* 1. To create USER

use admin

db.createUser(

{

user: "Gwats\_admin",

pwd: "Gwats\_pass", // Or "<cleartext password>"

roles: [ { role: "userAdminAnyDatabase", db: "admin" },

{ role: "readWriteAnyDatabase", db: "admin" },

],

})

use Resource\_manager

db.createUser(

{

user: "Gwats\_user",

pwd: "Gwats123",

roles: [ { role: "Gwats", db: "Resource\_manager" }]

}

)

Commandline access

$ mongo Resource\_manager admin -u Gwats\_user -p Gwats123

Configure MasterTestBed.xml

Go to the folder cd Controller/Lib/system/resource\_manager

vim MasterTestBed.xml

Under <TestBed> tag,create the DUT data

For Example, Add station related info as below,

<GSTA1>

<os>Linux</os>

<socket\_type>simplesocket</socket\_type>

<ctrl\_ip>192.168.1.11</ctrl\_ip>

<test\_ip>172.16.1.24</test\_ip>

<port>12345</port>

<mac>c0:3f:d5:fe:42:77</mac>

<ctrl\_username>gtest</ctrl\_username>

<ctrl\_password>p@ssw0rd</ctrl\_password>

<support>IEEE802.11a/b/g/n/ac</support>

</GSTA1>

Add AP related info as below,

TODO:Need to add AP config parameters

Add Sniffer related info as below,

TODO:Need to add Sniffer config parameters

Run $python3 dbserver.py

3. Modify the windows EndPoint configuration file as follows,

Go to the folder: /home\_directory/GWATS/EndPoints/STA\_EndPoint/Config/config.xml

<ConfigInformation>

<System>

<OS>Operating system<OS>

<Version>Operating system Version<Version>

<Architecture>64bit/32bit<Architecture>

</System>

<CtrlLine>

<interface>

<name>Interface name connected to controller</name>

<type>Type of interface (wired/wireless)</type>

<ip>IP address of the system</ip>

<host>IP address of the system/Host name(Eg: STA1)</host>

<port>Port number to connect to controller</port>

<mac>mac address of the interface</mac>

<sock\_type>IP address of the system(simplesocket)</sock\_type>

</interface>

</CtrlLine>

<TestLine>

<interface>

<name>Interface name to be tested (wireless interface)</name>

<type>Type of interface (wired/wireless/Android/IOS)</type>

<ip>IP address to be configured</ip>

<nmask>Net mask IP address to be configured</nmask>

<id>Andoid Device ID connected</id>

<port>The port no. to communicate with Device</port>

<mac>mac address of the interface</mac>

<sock\_type>IP address of the system(simplesocket)</sock\_type>

</interface>

<configutil>

<util>Utility used in the system for wifi</util>

<path>Path of the Utility present</path>

<driver>wifi driver present</driver>

<register>Register Id/No. if present(XXXXX/Re)</register>

</configutil>

</TestLine>

<log>

<path>Log path to save logs</path>

<loglevel>Log level to br printed(info/debug)</loglevel>

</log>

</ConfigInformation>

Steps to start windows endpoint:

1. Open the terminal
2. Go to the folder

$ cd /home\_directory/GWATS/EndPoints/STA\_EndPoint/bin

1. start start.py as follows

$ sudo python3.7 start.py

1. check for the environment variables

$ cat exprtenv.sh

$ env

check for HOMEPATH,LOGPATH,PORT etc

4.In the same way configure Linux Endpoint in the mentioned config file

Go to the folder: /home\_directory/GWATS/EndPoints/STA\_EndPoint/Config/config.xml

Steps to start Linux endpoint:

1. Open the terminal
2. Go to the folder

$ cd /home\_directory/GWATS/EndPoints/AP\_EndPoint/bin

1. start start.py as follows

$ sudo python3.7 start.py

1. Check for the environment variables

$ cat exprtenv.sh

$ env

check for HOMEPATH,LOGPATH,PORT etc

1. Update the Testsuite details ,to execute the required testsuite in Controller/Config

vim APTestManifest.xml ---> for APUT tests.

a.If Sanity testsuite want to run,

i. Update the devices which are required like GSTA1, GSTA2.

ii. In UserPreference tag, Options given for

-> Change the SNF enable or Disable.

-> Prefixed channel for all the testcases.

-> LogLevel option can change to DEBUG or INFO

iii. Wip\_range tag is for : Assign the static ips for stations from 100 on words.

iv. Testjob : for sanity, Testcase : ALL/ <Particular testcase ID>

v. Testjob : ALL/ <particular Testjob tag name>

b. StaTestManifest.xml for STAUT Tests.

Similar to APUT tests.

## Test Scripts

All test suites are found in **TODO: Need to add**

### Execution

To start the testcase:

a. cd ~/GWATS/Controller/bin/

b. ./Intiator.py <APTestManifest.xml / StaTestManifest.xml>

### PASS/FAIL Criteria

TODO:Need to update based on robot output