EayMesh Controller & Agent that works on every SoC

UniFIED Wifi Mesh

User Manual

Munshi, Soumya

Table of Contents

Introduction

Unified-Wifi-Mesh is an end-to-end Wi-Fi mesh stack that facilitates configuration of parameters of Wi-Fi mesh network along with retrieval of different statistics and telemetry of different components of the network including devices, radios, Access Points, Client devices and neighbors. The stack is compatible with EasyMesh and WiFi-7 specifications. Currently, the implementation is compliant with Wi-Fi Alliance published Wi-Fi EasyMesh Specifications Version 5.0. An upgrade to Version 6.0 is planned by middle of 2025.

Building Unified-Wifi-Mesh

OneWifiMesh consists of 5 constituent software components that need to be individually built and executed to achieve end-to-end functionality of Wi-Fi mesh. All these components can be built on simple Linux based platform like RaspberryPi4.

Building OneWifi

The location of OneWifi is *"Name of Repository Here".* After downloading the code from the repository, please locate the makefile.inc file under generic folder. Please verify the directory names against the MACROS specified in the file. The definitions of the following must be accurate to build OneWifi for mesh functionalities.
ONE_WIFI_EM_HOME must be the directory location of the code downloaded from https://github.com/rdkcentral/unified-wifi-mesh.
WIFI_HAL_INTERFACE must be the directory location of the *code of halinterface*.
WIFI_HOSTAP_BASE must be the directory location of the code of *hostap*.
Under this directory "make clean" and "make" commands can be executed to build OneWifi.

Building Mesh Components

Agent

After downloading the code from https://github.com/rdkcentral/unified-wifi-mesh, please locate the build directory. Please verify the MACROS definitions of the locations of code are accurate. Under build directory please locate the agent directory. Mesh Agent can be built by executing the "make clean" and "make" commands in this agent directory.

Controller

Prerequisite of building and running controller is that mysql database must be installed in your Linux based system. In case of RaspberryPi4, you can install mysql by executing *"sudo apt install mariadb-server"* followed by *"sudo apt-get install libmysqlcppconn-dev"* commands. After downloading the code from https://github.com/rdkcentral/unified-wifi-

mesh, please locate the build directory. Please verify the MACROS definitions of the locations of code are accurate. Under build directory please locate the ctrl directory. Mesh Controller can be built by executing the *"make clean"* and *"make"* commands in this ctrl directory.

CLI

After downloading the code from https://github.com/rdkcentral/unified-wifi-mesh, please locate the build directory. Please verify the MACROS definitions of the locations of code are accurate. Under build directory please locate the cli directory. Mesh CLI can be built by executing the *"make clean"* and *"make"* commands in this ctrl directory.

Building 1905.1 Abstraction and Component

Running Unified-Wifi-Mesh

To run the end-to-end stack, as mentioned before, five constituent components must be built and run.

Running Controller

To run mesh controller, you must configure and run mysql database. To configure mysql database for controller, you should execute the following. To configure your custom user account and password to be used by controller in mysql, run mysql CLI by executing *"sudo mysql"*. In the mysql CLI prompt, execute the command *"ALTER USER 'pi'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_new_password'; FLUSH PRIVILEGES;"*. Verify that the user account and password is configured by exiting the mysql program and running it again by executing *"mysql -u pi -p"*. When password prompt comes, type in the password. You need to create a database for controller. Execute the commad *"create database OneWifiMesh;"*. Verify the database is created by executing *"show databases;"* and *"use OneWifiMesh;"*. You can get more information about mysql at https://www.basedash.com/blog/how-to-install-mysql-on-a-raspberry-pi. Now you can run controller by executing *"sudo ./onewifi_em_ctrl pi@"your password""*.

Running CLI

After building cli, please locate the install/bin directory under mesh code directory. To run cli, execute *"sudo ./onewifi_em_cli"* command. The CLI command prompt will appear as *<<OneWifiMeshCli>>*.

Initial Database Reset

Please use the command *"reset eth0 pi"* to setup the controller to accept and register mesh agents. In the above example, the eth0 argument indicates the 1905.1 abstraction layer interface name of the controller device. This can be substituted by any other interface

name that you would like to be the 1905.1 AL interface. The second argument indicates the name of the device.

Running OneWifi

After building OneWifi, please locate the install/bin directory. To run OneWifi, execute *“sudo ./OneWifi -c”* command.

Running Agent

After building agent, please locate the install/bin directory under mesh code directory. To run agent, execute *“sudo ./onewifi_em_agent”* command. Once this is executed, the mesh agent will continuously transmit 1905.1 Autoconfig Search messages to find and register with the controller. If the controller is up and running in the network, a 1905.1 Autoconfig Response, 1905.1 Autoconfig WSC, 1905.1 Topology Query and 1905.1 Topology Response will follow. At this point, the RspberryPi4 device will broadcast the Wi-Fi SSID specified in the system.

Running 1905.1 Abstraction and Components

Architecture

The fundamentals of Unified-Wifi-Mesh architecture is a layered non monolithic model. The different components of this layered model communicate with each other using standardized API sets that are published by standardized bodies like Wi-Fi Alliance or IEEE or Broadband Forum. A simplified model of this architecture is depicted below.

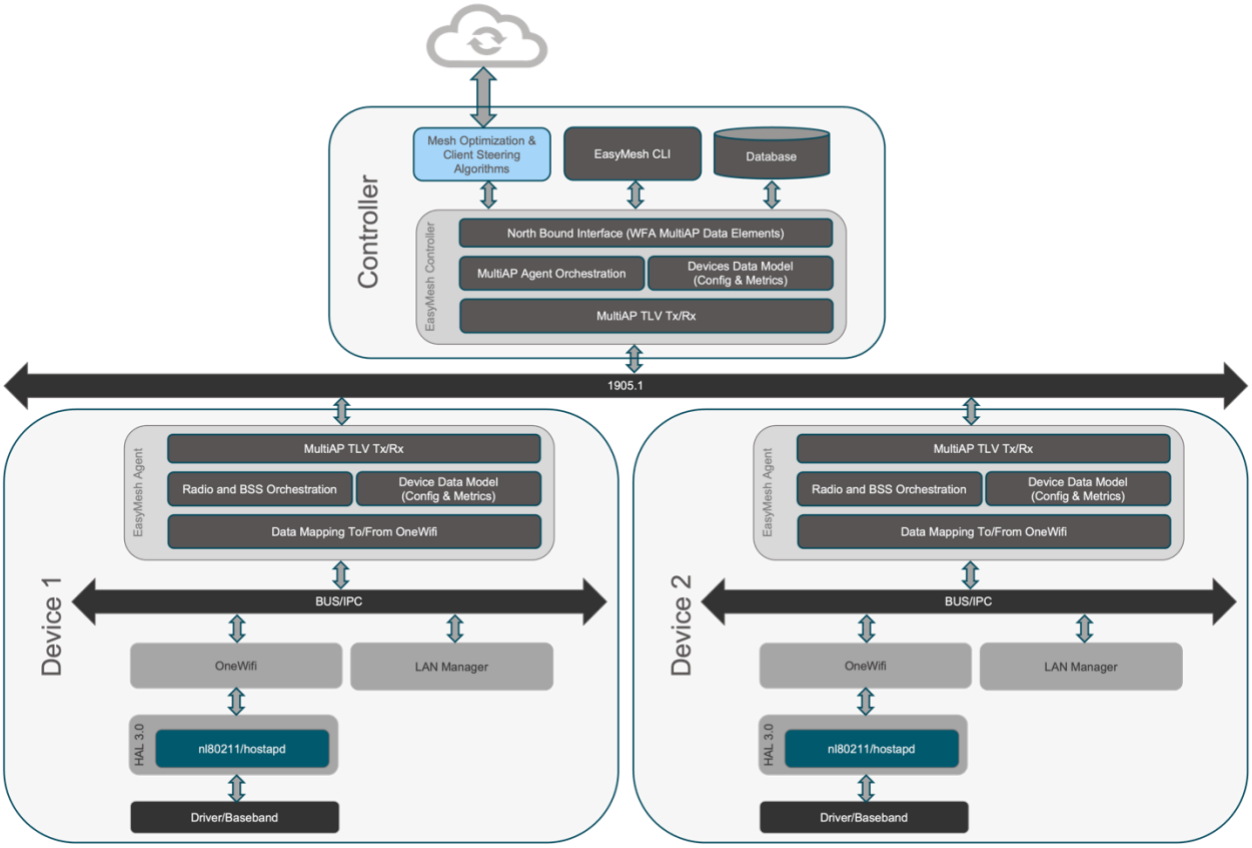

Figure 1: High Level EasyMesh Controller & Agent Architecture

Controller

The controller architecture consists of five components.

EasyMesh Objects

Each of this object represents a remote device peer radio. The object is created when a remote or collocated Agent EasyMesh object performs auto-configuration via M1/M2 process. These objects are responsible for constructing and transmitting MultiAP frames. These objects are also responsible for receiving and analyzing MultiAP frames. The state of these objects is updated from data received from peer objects via MultiAP frame exchange. The data is updated in the data model object that is described in the next section.

Data Models List

The data models list is a list of data model objects. Each data model object represents the aggregated data model of a device and bears a direct correlation with the WFA data element referenced by wfa-dataelements.Network.DeviceList.{i}. All the leaves parameters of the Device data element object are part of this data model object. As and when data from peer agents are received via MultiAP frames, these objects are updated.

Orchestrator

Orchestrator is responsible for orchestrating the states of peer objects when a command is applied from external entities like CLI or Optimizer. The orchestrator initiates the necessary actions for any specific command and completes and destroys the command after successful application to all possible peer objects. In case of failure, the command is timed out and destroyed, the states of the peer objects remain unchanged.

Database

Nearly all the data elements of data model objects are stored persistently in MySql database. Whenever the database is updated, the corresponding data is loaded in running memory of the data models list that is described above.

Manager

The controller is essentially a manager entity that coordinates the activities among the objects described above. The manager also listens for command requests from CLI or other North Bound interfaces with Optimizer.

Controller State Machines

Agent (Device) Discovery & Configuration

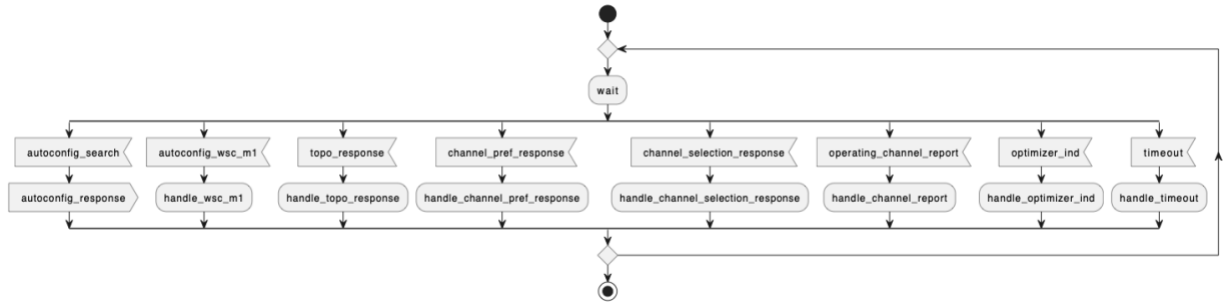

Figure 2: Controller Discovery & Configuration State Diagram

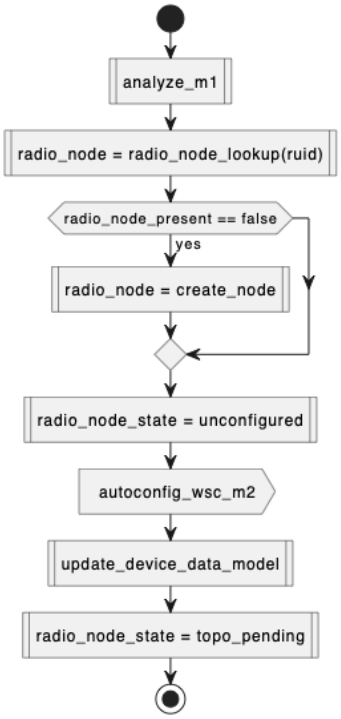

Figure 3: WSC M1 State Diagram

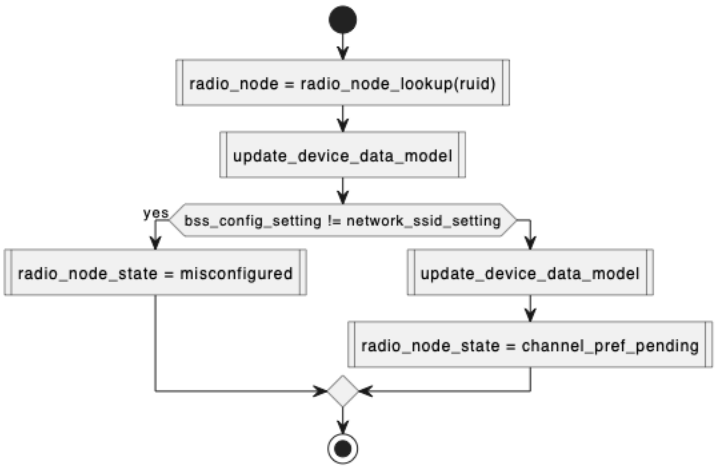

Figure 4: Topology Response State Diagram

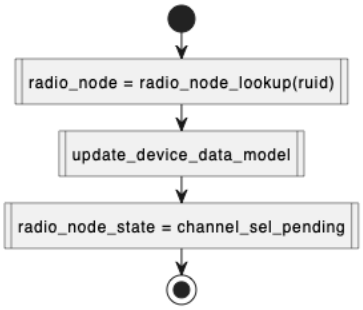

Figure 5: Channel Preference Response State Diagram

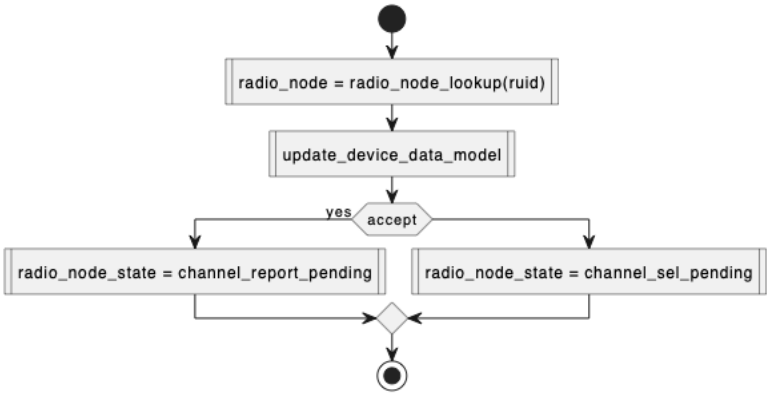

Figure 6: Channel Selection Response State Diagram

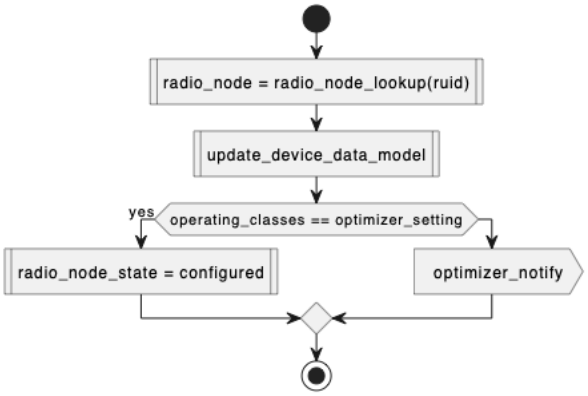

Figure 7: Channel Report State Diagram

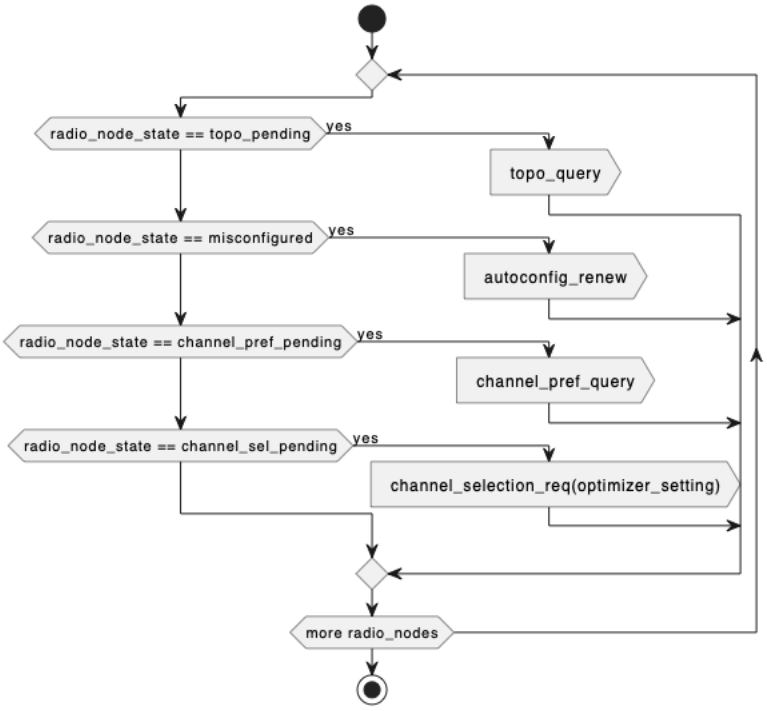

Figure 8: Timeout State Diagram

Agent

The agent architecture consists of three components

EasyMesh Objects

This object represents a radio on the device. If a device has three radios, three such objects will be created. These objects are responsible for sending and receiving MultiAP frames. As and when frames are received from controller, the object is responsible for analyzing the frame and updating OneWifi with the latest configuration. The state of the object is updated accordingly. The data corresponding to the updated state is stored in the data model object and updates are sent to the peer object of controller via MultiAP frames.

Data Model

The data model object represents the device data model. This data model object has a direct correlation with the data element described by wfa-dataelements.Network.DeviceList.{i}.Device.

OneWifi Data Translator

This entity is responsible for translating configuration and statistics related data between OneWifi and Agent. The agent data model is essentially derived from WFA data elements definition and OneWifi data model is derived from HAL 3.0 specifications.

CLI

The CLI entity is used primarily for development testing purposes. The supported commands are described below.

Getting and Setting SSID/Passphrase of Mesh Network

The initial reset installs default SSID and passphrases for fronthaul and backhaul networks in the system. To view the defaults, please use the *"get_ssid OneWifiMesh"* command in CLI. The output will look like following.

```
<<OneWifiMeshCli>>: get_ssid OneWifiMesh
{
        "Status": "Success",
        "Result": {
                "NetworkSSIDList":      [{
                        "SSID":   "private_ssid",
                        "PassPhrase":      "test-fronthaul",
                        "Band":   ["2.4", "5", "6"],
                        "Enable":true,
                        "AKMsAllowed":    ["dpp"],
                        "SuiteSelector":    "",
                        "AdvertisementEnabled":    true,
                        "MFPConfig":       "Optional",
                        "MobilityDomain": "00:01:02:03:04:05",
                        "HaulType":        ["Fronthaul"]
                }, {
                        "SSID":   "iot_ssid",
                        "PassPhrase":      "test-backhaul",
                        "Band":   ["2.4", "5", "6"],
```

```
                        "Enable":true,
                        "AKMsAllowed":    ["dpp", "sae", "SuiteSelector"],
                        "SuiteSelector":    "00010203",
                        "AdvertisementEnabled":    true,
                        "MFPConfig":        "Required",
                        "MobilityDomain": "00:01:02:03:04:05",
                        "HaulType":         ["IoT"]
                    }, {
                        "SSID":    "lnf_radius",
                        "PassPhrase":       "test-backhaul",
                        "Band":   ["2.4", "5", "6"],
                        "Enable":true,
                        "AKMsAllowed":    ["dpp", "sae", "SuiteSelector"],
                        "SuiteSelector":    "00010203",
                        "AdvertisementEnabled":    true,
                        "MFPConfig":        "Required",
                        "MobilityDomain": "00:01:02:03:04:05",
                        "HaulType":         ["Configurator"]
                    }, {
                        "SSID":    "mesh_backhaul",
                        "PassPhrase":       "test-backhaul",
                        "Band":   ["2.4", "5", "6"],
                        "Enable":true,
                        "AKMsAllowed":    ["dpp", "sae", "SuiteSelector"],
                        "SuiteSelector":    "00010203",
                        "AdvertisementEnabled":    true,
                        "MFPConfig":        "Required",
                        "MobilityDomain": "00:01:02:03:04:05",
                        "HaulType":         ["Backhaul"]
                    }]
            }
}
```


Getting and Removing Devices in Mesh Network

<<OneWifiMeshCli>>: *get_device OneWifiMesh*


```
{
        "Status": "Success",
        "Result": {
                "Network":          {
                        "ID":       "OneWifiMesh",
                        "DeviceList":       [{
                                        "ID":       "dc:a6:32:d5:91:62",
                                        "MultiAPCapabilities":          "",
                                        "CollectionInterval":           0,
                                        "ReportUnsuccessfulAssociations":   false,
                                        "MaxReportingRate":             0,
                                        "APMetricsReportingInterval":           0,
                                        "Manufacturer":     "Raspberry Pi 4 Model B Rev 1.4",
                                        "SerialNumber":     "100000007a43d805",
                                        "ManufacturerModel":        "Raspberry Pi 4 Model B
                                        "SoftwareVersion":          "",
                                        "ExecutionEnv":     "",
                                        "DSCPMap":          "",
                                        "MaxPrioritizationRules":       0,
                                        "MaxVIDs":          0,
                                        "CountryCode":      "",
                                        "PrioritizationSupport":        false,
```

```
        "ReportIndependentScans":           false,
        "TrafficSeparationAllowed": false,
        "ServicePrioritizationAllowed":     false,
        "DFSEnable":        false,
        "MaxUnsuccessfulAssociationReportingRate": 0,
        "STASteeringState":         false,
        "CoordinatedCACAllowed": false,
        "ControllerOperationMode":          "",
        "BackhaulMACAddress":       "00:00:00:00:00:00",
        "BackhaulDownMACAddress":       [],
        "BackhaulPHYRate":          0,
        "BackhaulALID":     "00:00:00:00:00:00",
        "TrafficSeparationCapability":      false,
        "EasyConnectCapability":    false,
        "TestCapabilities": 0,
        "APMLDMaxLinks":            0,
        "TIDLinkMapping": "",
        "AssociatedSTAReportingInterval":       0,
        "MaxNumMLDs":   0,
        "bSTAMLDMaxLinks":          0
    }, {
        "ID":       "e4:5f:01:40:70:5b",
        "MultiAPCapabilities":          "",
        "CollectionInterval":           0,
        "ReportUnsuccessfulAssociations":   false,
        "MaxReportingRate":         0,
        "APMetricsReportingInterval":       0,
        "Manufacturer":     "",
        "SerialNumber":     "",
        "ManufacturerModel":        "",
        "SoftwareVersion":          "",
        "ExecutionEnv":     "",
        "DSCPMap":          "",
        "MaxPrioritizationRules":       0,
        "MaxVIDs":          0,
        "CountryCode":      "",
        "PrioritizationSupport":        false,
        "ReportIndependentScans":           false,
        "TrafficSeparationAllowed": false,
        "ServicePrioritizationAllowed":     false,
        "DFSEnable":        false,
        "MaxUnsuccessfulAssociationReportingRate": 0,
        "STASteeringState":         false,
        "CoordinatedCACAllowed": false,
        "ControllerOperationMode":          "",
        "BackhaulMACAddress":       "00:00:00:00:00:00",
        "BackhaulDownMACAddress":       [],
        "BackhaulPHYRate":          0,
        "BackhaulALID":     "00:00:00:00:00:00",
        "TrafficSeparationCapability":      false,
        "EasyConnectCapability":    false,
        "TestCapabilities": 0,
        "APMLDMaxLinks":            0,
        "TIDLinkMapping": "",
        "AssociatedSTAReportingInterval":       0,
        "MaxNumMLDs":   0,
        "bSTAMLDMaxLinks":          0
    }]
```

```
                }
        }
}
```


Getting Information of Radios and Radio Statistics

<<OneWifiMeshCli>>: *get_radio OneWifiMesh*

```
{
        "Status":  "Success",
        "Result":  {
                "Network":          {
                        "ID":       "OneWifiMesh",
                        "NumberOfDevices":          12160,
                        "TimeStamp":        "\u0019",
                        "ControllerID":     "02:01:02:01:00:01",
                        "MSCSDisallowedStaList":    [],
                        "SCSDisallowedStaList":     [],
                        "CollocatedAgentID":        "00:0c:29:dc:63:48",
                        "DeviceList":       [{
                                "ID":       "00:0c:29:dc:63:48",
                                "RadioList":        [{
                                        "ID":       "00:ab:cd:ef:00:01",
                                        "Enabled":          true,
                                        "NumberOfUnassocSta":       0,
                                        "Noise":    20,
                                        "Utilization":          56,
                                        "TrafficSeparationCombinedFronthaul":true,
                                        "TrafficSeparationCombinedBackhaul": false,
                                        "SteeringPolicy":   1,
                                        "ChannelUtilizationThreshold":          200,
                                        "RCPISteeringThreshold":    190,
                                        "STAReportingRCPIThreshold":        180,
                                        "STAReportingRCPIHysteresisMarginOverride":     0,
                                        "ChannelUtilizationReportingThreshold":         190,
                                        "AssociatedSTATrafficStatsInclusionPolicy":     true,
                                        "AssociatedSTALinkMetricsInclusionPolicy":      true,
                                        "ChipsetVendor":    "testVendor"
                                },{
                                        "ID":       "00:ab:cd:ef:00:02",
                                        "Enabled":          true,
                                        "NumberOfUnassocSta":       0,
                                        "Noise":    20,
                                        "Utilization":          18,
                                        "TrafficSeparationCombinedFronthaul":true,
                                        "TrafficSeparationCombinedBackhaul": false,
                                        "SteeringPolicy":   1,
                                        "ChannelUtilizationThreshold":          200,
                                        "RCPISteeringThreshold":    190,
                                        "STAReportingRCPIThreshold":        180,
                                        "STAReportingRCPIHysteresisMarginOverride":     0,
                                        "ChannelUtilizationReportingThreshold":         190,
                                        "AssociatedSTATrafficStatsInclusionPolicy":     true,
                                        "AssociatedSTALinkMetricsInclusionPolicy":      true,
                                        "ChipsetVendor":    "testVendor"
                                },{
                                        "ID":       "00:ab:cd:ef:00:10",
                                        "Enabled":          true,
```

```
                                                "NumberOfUnassocSta":       0,
                                                "Noise":    20,
                                                "Utilization":      6,
                                                "TrafficSeparationCombinedFronthaul":true,
                                                "TrafficSeparationCombinedBackhaul": false,
                                                "SteeringPolicy":     1,
                                                "ChannelUtilizationThreshold":      200,
                                                "RCPISteeringThreshold":      190,
                                                "STAReportingRCPIThreshold":      180,
                                                "STAReportingRCPIHysteresisMarginOverride":    0,
                                                "ChannelUtilizationReportingThreshold":      190,
                                                "AssociatedSTATrafficStatsInclusionPolicy":      true,
                                                "AssociatedSTALinkMetricsInclusionPolicy":      true,
                                                "ChipsetVendor":    "testVendor"
                                        }]
                                }]
                        }
                }
}
```


Getting and Setting Operating Channels of Mesh Network

***<<OneWifiMeshCli>>**: get_channel OneWifiMesh*


```
{
        "Status": "Success",
        "Result": {
                "Network":          {
                        "ID":       "OneWifiMesh",
                        "DeviceList":       [{
                                        "ID":       "00:0c:29:dc:63:48",
                                        "RadioList":        [{
                                                "ID":       "00:ab:cd:ef:00:01",
                                                "CurrentOperatingClasses":[{
                                                        "Class":  81,
                                                        "Channel":      6,
                                                        "TxPower":      20,
                                                        "MaxTxPower":     0,
                                                        "NonOperable":    [0]
                                                }]
                                        }, {
                                                "ID":       "00:ab:cd:ef:00:02",
                                                "CurrentOperatingClasses":[{
                                                        "Class":  115,
                                                        "Channel":      36,
                                                        "TxPower":      20,
                                                        "MaxTxPower":     0,
                                                        "NonOperable":    [0]
                                                }]
                                        }, {
                                                "ID":       "00:ab:cd:ef:00:10",
                                                "CurrentOperatingClasses":[]
                                        }]
                                }]
                }
        }
}
```

Getting Information of Client (Station) Devices and Statistics