# Table of Content

1. **What is TIP OpenWiFi - A Basic Introduction** ........................................ 2  
   1.1 The Benefits of OpenWiFi ................................................................. 2  
   1.2 OpenWiFi’s Unique Capabilities ....................................................... 2  

2. **How to Become a TIP OpenWiFi Member** ...................................... 4  

3. **Edgecore OpenWiFi Solution** ......................................................... 6  
   3.1 Edgecore TIP OpenWiFi Ready Devices ............................................ 6  
   3.2 Edgecore Devices Supported by OpenWiFi ....................................... 7  
   3.3 Edgecore ecOpen Cloud Controller .................................................. 10  
   3.4 Edgecore OpenWiFi Expert Team - Oxherd ...................................... 10  
   3.5 Edgecore WiFi Community ............................................................... 10  

4. **Get on Board with ecOpen Cloud Controller** .................................. 11  
   4.1 ecOpen Cloud Controller Login ......................................................... 11  
   4.2 Site Creation ...................................................................................... 13  
   4.3 OpenWiFi Device Preparation .......................................................... 15  
   4.4 Configuration .................................................................................... 17  
   4.5 Information Display .......................................................................... 23  

5. **Appendix - Build a Test Lab From Scratch** ........................................ 27  
   5.1 Get AP’s Latest Firmware ................................................................. 27  
   5.2 Upgrade AP’s Firmware .................................................................... 27  
   5.3 Install CloudSDK Controller ............................................................ 28  
   5.4 Connect AP to CloudSDK Controller .............................................. 29  
   5.5 Configure Managed Aps ................................................................. 30  
   5.6 Build AP’s Firmware ....................................................................... 34  
   5.7 Modify AP’s Firmware ..................................................................... 35  
   5.8 Flash TIP OpenWiFi AP Image from Non-TIP Devices ...................... 36
1. What is TIP OpenWiFi - A Basic Introduction

OpenWiFi is a community-developed, disaggregated Wi-Fi software system, offered as free open-source software that includes both a cloud controller SDK and an Enterprise-grade Access Point (AP) firmware, designed and validated to work seamlessly together.

1.1 The Benefits of OpenWiFi

- Lower R&D cost to develop enterprise-grade advanced Wi-Fi solutions with OpenWiFi’s robust, and single codebase
- Accelerated development, cost effective integration, and open innovation due to OpenWiFi’s common control, data, and management layers
- Favorable economics for Service Providers and Enterprises by bringing significant reduction in Total Cost of Ownership (TCO) over current proprietary solutions and diverse multi-vendor selection of cloud controllers and access points

1.2 OpenWiFi’s Unique Capabilities

- Community Driven Development
  The OpenWiFi community empowers all stakeholders to participate and contribute, see all the source code and nightly test results. This open collaboration model enables rapid development and release of customer-defined features, minimizing the time to get the new releases from the vendors.
- Fully Disaggregated Open Tech Stack
  The OpenWiFi software tech stack incorporates advanced Enterprise and Carrier-
grade features, including open, standardized APIs and data models (AP & Controller); scalable mobility & Wi-Fi meshing; extensible Radio Resource Management (RRM); Passpoint (802.11u) and OpenRoaming; configuration, telemetry & analytics, and much more.

- **Commercial-Grade Automated Testing**
  The TIP OpenWiFi project has applied the best practices of web-scale software development and the Wi-Fi OEM industry to deliver commercial-grade quality. E.g., CI/CD software builds for all hardware platforms; DevOps managed release engineering; and community-developed automated testing.

- **Product Compliance Validation**
  The community oversees and ensures OpenWiFi based products are compliant with the software tech stack. Compliance testing includes:
  1. Each access point hardware SKU passes software capability testing
  2. Cloud solutions can interwork with any OpenWiFi compliant access point
  3. Software products properly implement the OpenWiFi cloud controller SDK northbound API
  4. Multi-vendor OpenWiFi networks interoperate as expected (Meshing, RRM, etc.)

- **Globally Secure Zero Touch Provisioning**
  OpenWiFi includes Zero Touch Provisioning (ZTP) based on Public Key Infrastructure (PKI) over the public internet, a key requirement for deploying disaggregated network systems in a secure way.

For more info, please visit [OpenWiFi’s official site](https://www.openwifi.org).
2. How to Become a TIP OpenWiFi Member

Getting the TIP products and managed with CloudSDK is an excellent start to participate the TIP projects. TIP is not only a OpenWiFi solution, but also a community that everyone contributes, so let’s apply for the TIP members.

(1) Visit Telecom Infra Project | Global Community Connectivity collaboration and go to ‘Get Involved’.

(2) Based on the purpose, there are different Tiers for you to apply such as the Full Participation Tier, the Associate Participation Tier, and the Software Participation Tier. To register your organization as a TIP partner, you’ll need to prepare some information.

Steps to join TIP

You will be asked to:

1. Confirm your work email address
2. Provide your own profile information

If your organization is NOT already a participant of TIP, you will also need to provide:

3. Your organization’s contact information
4. A Billing Contact
5. Desired participation tier
6. An Authorized Representative to be sent the General Participation Agreement for signature

After the participation agreement is signed, TIP will review your application and send an invoice for any participation dues.
(3) Once approved to join the TIP community, you’ll be benefited with the following community resources:

- **GitHub Account**: OpenWiFi publishes open-sourced firmware to GitHub. It is best to connect your GitHub account together with TIP community.

- **Atlassian**: TIP uses Confluence and JIRA of Atlassian products as their knowledge base. You’ll find detailed technical information in the Confluence, tracking test cases of nightly build testbeds in JIRA.

- **Slack**: You’ll also be able to reach ISPs & other system integrators worldwide on the Slack chat. It connects users together and brings more business opportunities.
3. Edgecore OpenWiFi Solution

As an active contributing member of the Telecom Infra Project (TIP) Open Converged Wireless Software Project Group, Edgecore works with other TIP members to accelerate the pace of innovation in the Wi-Fi ecosystem by creating a cost-effective disaggregated open-source solution aimed to improve global connectivity.

3.1 Edgecore TIP OpenWiFi Ready Devices

Edgecore provides a range of access points that comes preinstalled with TIP’s Open AP software that offers users an open platform for further customization. From SMB to MDUs to larger venues, Edgecore TIP-ready access points can be accustomed to various usage and requirements.

The all-in-one package allows users to save on time and the hassle of having to go through initial setup. If required, Edgecore products provide endless possibilities on its open platform for clients to expand accordingly.

Edgecore is endeavoring to bring more TIP-ready products into the market and to develop, build, test, and deploy open, disaggregated, and standards-based solutions that deliver the high-quality connectivity that the world needs.
There are two versions of CloudSDK provided by TIP OpenWiFi, both supported by Edgecore devices, and they are TIP 1.x which is based on OpenSync and TIP 2.x which is based on uCentral. In this document we focus on Edgecore’s OpenWiFi solution for TIP 2.x uCentral version for the CloudSDK.

3.2 Edgecore Devices Supported by OpenWiFi

Wi-Fi 6

EAP101 (Indoor Access Point)

- 802.11ax
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.7Gbps

EAP102 (Indoor Access Point)

- 802.11ax
- 4x4:4 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 2.9Gbps

EAP104 (Indoor Access Point)

- 802.11ax
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.7Gbps
Wi-Fi 5 Wave2
ECW5211-L (Indoor Access Point)
- 802.11ac Wave2
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.2Gbps

ECW5410-L (Indoor Access Point)
- 802.11ac Wave2
- 4x4:4 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.73Gbps

SP-W2-AC1200 (Indoor/Outdoor Access Point)
- 802.11ac Wave2
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.3Gbps
- IP55
SS-W2-AC2600 (Indoor/Outdoor Access Point)
- 802.11ac Wave2
- 4x4:4 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 2.5Gbps
- IP55

SP-W2M-AC1200 (Indoor Access Point)
- 802.11ac Wave2
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.3Gbps

OAP100 (Outdoor Access Point)
- 802.11ac Wave2
- 2x2:2 MU-MIMO
- Dual Radio 2.4Ghz + 5Ghz
- Up to 1.3Gbps
- IP68

For more product information, please visit https://wifi.edge-core.com/openwifi
3.3 Edgecore ecOpen Cloud Controller

By converging with CloudSDK 2.0, the highlight of OpenWiFi 2.0, ecOpen can provide central management and visibility to all TIP OpenWiFi devices. Being the world’s first launched TIP OpenWiFi Cloud Controller, ecOpen can support the full range of TIP OpenWiFi hardware devices and software applications, which realizes the dream of commercializing TIP OpenWiFi and making it a sustainable business model.

In the next chapter, we will show you how to get on board with ecOpen (Chapter 4).

Visit ecOpen Cloud Controller: https://cloud.openwifi.ignitenet.com/

3.4 Edgecore OpenWiFi Expert Team - Oxherd

The Oxherd team consists of elites from Edgecore’s System Architects, R&D, Quality Assurance (QA), and Technical Support. From adopting an OpenWiFi environment to configuring specific access point features, the Oxherd team is here to assist clients at every step and accelerate the process in every phase. With the Oxherd team by their side, ISPs/MSPs can smoothly build up their OpenWiFi environments, while software companies can release all kinds of innovative possibilities through embedded system development support, cloud API integration, and resource for development. With the help of Oxherd, clients can face complex environments and multiple variables with simple methods that speed the process of implementing OpenWiFi; the software development time can be shortened from months to weeks before testing, or even running.

Reach the team: oxherd@edge-core.com

3.5 Edgecore WiFi Community

Since Edgecore is one of the biggest participants and the contributors of TIP community. You can reach to Edgecore Technical Support Team and Edgecore partners on the Edgecore OpenWiFi Forum. Welcome to visit the community here: https://community.wifi.edge-core.com

For more information, please visit Edgecore OpenWiFi site
4. Get on Board with ecOpen Cloud Controller

In this chapter, we will introduce ecOpen Cloud Controller, including account registration, device registration, inheritance policy, configuration, and useful information.

4.1 ecOpen Cloud Controller Login

a. From a web browser, go to cloud.openwifi.ignitenet.com to register an account and start creating your own cloud networks and sites.

Click “I want to register” to create a new account.
b. Enter your email address and specify your first and last name. Set a password to protect access to your account, tick the “I am not a robot” checkbox, and then click REGISTER.

c. The ecOpen Cloud Controller will send you a verification email to the email you registered. When you received the email, click the link to activate your account.

The ecOpen Cloud Controller uses a cloud-like account – it houses a group of sites, which are logical groupings of your managed devices. Each cloud will have its own set of users and configuration settings. As an end-user, you can join many clouds, with different roles on each cloud. Once you are registered as a user on the ecOpen Cloud Controller, you are given the option to create a cloud when you first log in.
4.2 Site Creation

a. After cloud is created, ecOpen will navigate user to create site.

b. After setting the regulatory country and local logins, click “Save” to save your configuration.
c. After saving configuration, you are prompted to add devices to your new site. Click “ADD DEVICES” to continue.

d. Fill-in your device serial number, MAC address and device name to add device. If you have any problem when adding device, please contact our support and provide your device information.
4.3 OpenWiFi Device Preparation

In this section, we will guide you on how to let the OpenWiFi devices connect to the ecOpen Cloud Controller.

4.3.1 Redirector URL

The redirector URL in certificate is the cloud URL (backend cloud URL) that the AP would connect to. To manage devices on ecOpen, please change the redirector URL in OpenWiFi device.
The redirector URL for ecOpen is "owsvc.openwifi.ignitenet.com".
If the Redirector URL is correct and correctly connect to ecOpen, then you will see device register state changes to Registered and start synchronization.
4.3.2 Edgecore OpenWiFi Product

The Edgecore AP has already set the default redirector URL to ecOpen Cloud Controller.

**Note:** If you would like to check the redirector URL on AP, please SSH into device and check value "server" in `/etc/config/ucentral` and `/etc/config-shadow/ucentral`. If the value "server" in your device is not "owsvc.openwifi.ignitenet.com", please contact our support to change remotely in Digicert server, after certificate updated, please reset your device to default to download updated certificate.

The correct setting on AP shows as follows:
```bash
cfg ucentral 'config'
  option port '15002'
  option debug '1'
  option reporting '10'
  option serial '98192c8d5825'
  option server 'owsvc.openwifi.ignitenet.com'
```

4.3.3 Non-Edgecore OpenWiFi Product

Please contact your product vendor to change the redirector URL to "owsvc.openwifi.ignitenet.com" and reset device to default to download updated certificate.

Or you can change redirector URL for temporary purpose by editing `/etc/config/ucentral` and `/etc/config-shadow/ucentral`, changing the value "server" and then reboot.

**Note:** This value will not be kept after device reset to default.

The correct setting on AP shows as follows:
```bash
cfg ucentral 'config'
  option port '15002'
  option debug '1'
  option reporting '10'
  option serial '98192c8d5825'
  option server 'owsvc.openwifi.ignitenet.com'
```
4.4 Configuration

In this section, we will introduce ecOpen Cloud Controller configuration architecture and inheritance policy.

4.4.1 OpenWiFi Configuration Style

We implement JSON configuration architecture in ecOpen. In JSON format, there’s a key-value pair. To make sure the Configuration Inheritance correctly, we use “Key” as root node.

```
"ssids": [  
  
  
  "name": "SSID01",  
  "wifi-bands": [  
    "5G",  
    "2G"
  ]
]
```

4.4.2 Configuration Inheritance

When a new device is added to the Cloud, the device’s “Site-level configuration inheritance” behavior must also be selected. This “Inheritance Policy” determines how the Cloud configures a device.

Cloud Configuration is very flexible; it allows Device-level configuration overrides to be setup when there is a need to inherit only a subset of Site-level settings. The Site-level Inheritance Policy is set when you first register a device, but this can also be changed later at any time.

There are two Inheritance Policy options for devices:

- **Inherit Site-level Settings** — Select this Inheritance Policy if you want to manage devices at a site like a single unit with a common configuration.
- **Don’t Inherit Site-level Settings** — Select this Inheritance Policy if you do not want a device to inherit any settings from the Site level. You would normally
choose this Inheritance Policy if a device is used for infrastructure, backhaul, or needs to be configured independently from the other devices at a site.

When Site-level inheritance is enabled for a device, the device’s final configuration will include the following:

- Settings inherited from the Site-level device configuration.
- Settings initially inherited from the Site-level configuration that “value” of JSON KEY-VALUE pair have been since been modified as a Device-level” override.”
- Settings unique to the Device-level configuration. That is, device-specific settings that are not configurable at the Site-level.

**WARNING:** the “KEY” in Site JSON configuration, Interface name and SSID name cannot be modified or removed in Device-level configuration.

Here’s an example for Configuration merging, we can change the VVALUE of KEY-VALUE pair in Device level configuration and configuration will be merged into device.

Here’s another example if we change SSID name device level configuration, original SSID01 will be reserved and another SSID02 will append, if we do not want SSID01 in this AP, we should remove it from Site-level.
4.4.3 Import JSON Configuration

In SITE MENU-Configuration-OpenWiFi page, click import the JSON configuration file, ecOpen Cloud Controller will automatically separate the configuration into different sections.

If device does not inherit Site-level Settings, you could also import JSON in Device-level configuration.
4.4.4 Configuration Section

ecoOpen Cloud Controller separate JSON configuration into several functional sections, user can easily monitor different function in each section:

- **Unit:** Hostname location time zone leds-active
- **Ethernet:** Ethernet port-related configuration. e.g.: select-ports speed duplex services
- **Radios:** Radio-related configuration, e.g.: bandwidth, channel, country, tx power
- **Interfaces:** Interface name, role, bridge, SSID
- **Services:** LLDP / SSH / NTP / Mdns, etc.
- **Third-Party:** Third-party Coova-Chilli login page configuration

4.4.5 Device-level Configuration

In Device-level configuration, there’re two parts in each section:

**CURRENT CONFIG:** Display current device configuration for user reference.

**CUSTOMIZED CONFIG:** You can edit your configuration here.
4.4.6 How to Configure in Site-level

a. Switch to SITE MENU, click Configuration, and then click OpenWiFi.

b. In Site-level configuration, click Import JSON subpage, you can upload your JSON file or just copy and paste.

c. Click Apply, the imported JSON configuration will be separated to each section.
d. Click Save. If the configuration is correct, ecOpen Cloud Controller will show **CHANGES SAVED**.

4.4.7 How to Configure in Device-level

a. Switch to DEVICE MENU, click Configuration.

b. In configuration page, if device does not inherit Site-level settings, there will be an Import JSON subpage, you can upload your JSON file or just copy and paste here.

c. You can also edit in each section, then click SAVE to push config into device.
4.5  Information Display

4.5.1  Dashboard

- Cloud-level dashboard displays high-level aggregated Site information, user can monitor the Site health status clearly.

- The Site-level dashboard provides status information for configured devices, client activity, most active clients, most active clients and application, gateway interface, site maps, and site activity.
- Device-level dashboard displays basic information. Include firmware version, model, configuration state, WAN IP and CPU utilization.

### 4.5.2 Statistics

ecOpen Cloud Controller provides Wireless, Network and System level historical statistics. User can monitor detailed information of OpenWiFi devices such as SSID throughput, associate clients and CPU utilization.
4.5.3 Activities

Click “Activity” on the Cloud menu to display all logged system alerts, maintenance tasks, and logged events. You can click the filtering button on the left to specify a date range selection.

The displayed messages can also be sorted by clicking on the ascending or descending arrows at the top of the Date column.
4.5.4 Notification

Click “Notifications” on the Site menu to access the notification settings for the selected site, such as device unreachable, device rebooted, configuration failed. These settings are used for any email sent for this site. You can disable the creation of individual alerts using the toggle switches on the Notification Settings page.
5. Appendix - Build a Test Lab From Scratch

If you are not using the ecOpen Cloud Controller to manage your OpenWiFi devices, you can build a test lab from scratch, please follow the instructions in this section.

5.1 Get AP's Latest Firmware

The formal released firmware can be obtained from TIP JFrog’s uCentral directory.

Download *-upgrade.bin file for firmware upgrade and download *.tar.gz for all files including firmware upgrade image and factory bin image. The factory bin image is for flashing the entire image.

5.2 Upgrade AP’s Firmware

There are two ways to upgrade OpenWiFi AP to another version locally before it is managed by any cloud controller:

(1) Upgrade by AP’s web UI

In the onboarding mode (default setting and not managed by cloud controller), access the AP’s web UI from LAN port or WiFi. Firmware can be manually upgraded in the upgrade page.

Notes:
If the AP is already managed by controller, the local web UI would be hidden.

- Default IP address is 192.168.1.1
- Default SSID is maverick (for the version greater than 2.6, the SSID would be maverick-<AP’s MAC address>)
- Default credentials: root/openwifi

(2) Upgrade by AP’s command line prompt
   I. Connect a PC to AP’s LAN port or WiFi
   II. Transfer firmware file to the /tmp directory of the AP by SCP
   III. Login to the console of the AP by SSH
   IV. Run sysupgrade command to upgrade the AP
      
      # sysupgrade -o /tmp/0220701-edgecore_eap101-v2.6.0-rc5-4c21f5c-upgrade.bin

Note:
- Default IP address is 192.168.1.1
- Default SSID is maverick (for the version greater than 2.6, the SSID would be maverick-<AP’s MAC address>)
- Default credentials: root/openwifi

5.3 Install CloudSDK Controller

Recommend using non-LB deployment with self-signed certificates for test lab.

Please find the steps below,
(1) Clone the repository from https://github.com/Telecominfraproject/wlan-cloud-ucentral-deploy/tree/main/docker-compose

      # git clone https://github.com/Telecominfraproject/wlan-cloud-ucentral-deploy

(2) Enable docker environment in your OS, here we use Ubuntu as an example.

      # sudo apt-get install docker.io

(3) Switch into the project directory to docker-compose directory

      # cd docker-compose/

(4) Deploy and run the docker container

      # docker-compose up -d

(5) Check if containers are up and running
# docker-compose ps

(6) Add a DNS entry from your PC to direct openwifi.wlan.local to the IP address of the host which installs the CloudSDK controller containers

```bash
# Host Database
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
# 127.0.0.1 localhost
255.255.255.255 broadcasthost
::1 localhost
10.2.142.25 openwifi.wlan.local
```

(7) Add SSL certificate checking exception in the browser for


(8) Login to CloudSDK UI and follow the instruction to change the default password

For more information, please refer to the document here:


### 5.4 Connect AP to CloudSDK Controller

All Edgecore OpenWiFi APs have a pre-installed DigiCert certificate, it redirects the AP to Edgecore OpenWiFi controller by default. If you want to build your own test environment with self-installed CloudSDK controller, please follow the steps here to change the AP’s redirect URL.

(1) Connect a PC to LAN port of the AP

(2) SSH to AP by default IP address 192.168.1.1 and default credentials root/openwifi.

(3) Change the value of `option server` option for /etc/config/ucentral and /etc/config-shadow/ucentral

```bash
config ucentral 'config'
  option port '15002'
  option debug '1'
  option reporting '10'
  option serial '98192c8d5825'
  option server 'wlan.openwifi.local'
```
(4) Reboot the device

(5) After the device boots up, configure the /etc/hosts file in the AP for wlan.openwifi.local

```
# Host Database
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
#...
127.0.0.1 localhost
255.255.255.255 broadcasthost
::1 localhost
10.2.142.25 openwifi.wlan.local
```

(6) To verify if the AP receives response from the controller

```
# logread | grep ucentral
```

- If successful, it would show: “daemon.info ucentral: running health task”
- If unsuccessful, it would show: “daemon.info ucentral: connection not successful”

(7) Check the cloud controller UI for the device

5.5 Configure Managed Aps

There are several device features that can be configured from the cloud controller, as documented in OpenWiFi Release 2.5 GA’s Device Feature Configuration Examples, such as

- Bridge Mode SSID
- NAT Gateway Mode SSID
- Multi-VLAN SSID
- Zero Touch Provisioning
- DHCP Relay
- Services
- Metrics
- GRE
- L2TP
- VxLAN
- WDS
- Mesh
- QoS
- Dynamic Air Time Fairness
- Dynamic Subscriber QoS
- Captive Portal
  - External Captive Portal
- ExpressWiFi
- Roaming RRM and SON
- RADIUS Authenticated SSID
  - Dynamic VLANs with RADIUS
- Multi-PSK (MDU Shared Key)
- Dynamic Air-Time Policy
- Passpoint®
- Switching
  - Port Speed

For example, if can set up the WDS (Wireless Distribution System) in an AP and a Station. In this example, the LAN side of the AP at the top if the topology will be wireless bridged to the LAN side of the AP Station at the bottom of the topology.

**WDS-AP**

```
"interfaces": [  
  {  
    "name": "WAN",  
    "role": "upstream",  
    "services": ["lldp"],  
    "ethernet": [  
      {  
      }
  }
```
"select-ports": [
  "WAN*"
],
"ipv4": {
  "addressing": "dynamic"
},
"name": "LAN",
"role": "downstream",
"services": ["ssh", "lldp"],
"ethernet": [
  {
    "select-ports": [
      "LAN*"
    ]
  }
],
"ssids": [
  {
    "name": "OpenWifi_WDS_AP",
    "wifi-bands": [
      "5G"
    ],
    "bss-mode": "wds-ap",
    "encryption": {
      "proto": "psk2",
      "key": "OpenWifi",
      "ieee80211w": "optional"
    },
    "roaming": {
      "message-exchange": "ds",
      "generate-psk": true
    }
  }
],
"ipv4": {
    "addressing": "static",
    "subnet": "192.168.10.1/24",
    "dhcp": {
        "lease-first": 10,
        "lease-count": 100,
        "lease-time": "6h"
    }
},

"interfaces": [
    {
        "name": "WAN",
        "role": "upstream",
        "services": ["lldp"],
        "ethernet": [
            {
                "select-ports": ["WAN*"
            }
        ]
    },
    {
        "name": "LAN",
        "role": "downstream",
        "services": ["ssh", "lldp"],
        "ethernet": [
            {
                "select-ports": ["LAN*"
            }
        ]
    }
],

"WDS-STA"
In this configuration, LAN clients of the WDS station AP receive IP addresses from the WDS Access Point AP from its LAN side DHCP service, via WDS ink at 5GHz.

5.6 Build AP’s Firmware

The open-source AP codes are available at Telecominfraproject/wlan-ap

To build AP firmware

(1) Setup build machine, and install packages below, (Python 3.7 or above is required)

```bash
# sudo apt install build-essential libncurses5-dev gawk libssl-dev gettext zlib1g-dev swig unzip time rsync python3 python3-setuptools python3-yaml openvswitch-common
```

(2) Get the repository from TIP’s GitHub

```bash
# git clone https://github.com/Telecominfraproject/wlan-ap
# cd wlan-ap
# git checkout release/v2.6.0
```
(3) Recommend using docker build to avoid many host dependencies (please check profile/ directory for supported AP models), using EAP101 as an example:

```
# TARGET=edgecore_eap101 make -j8
```

5.7 Modify AP’s Firmware

Here is an example to add a new package which is existing in OpenWRT.

(1) In profiles/, find the yml file for target AP

(2) Add package name as below

![Configuration File Example]

(3) To verify the name of the package and related packages that might need to be enabled as well

I. Copy the original configuration file

```
# cp openwrt/.config openwrt/.config.bak
```

II. Use menuconfig to select the packages you want to add

```
# cd openwrt/
# make menuconfig
```

III. After leaving menuconfig, a new .config will be generated, and then you can use diff to check additional packages needed to be installed. For example, if there is additional config option below,

```
CONFIG_PACKAGE_wwan=y
```

IV. Then you can add config option “wwan” to the yml file
(4) Then when you rebuild the firmware, the new packages will be included.

For more details, refer to https://openwrt.org/docs/guide-developer/packages

5.8 Flash TIP OpenWiFi AP Image from Non-TIP Devices

Pre-requisites:

- Use the Serial Console cable to connect to the device console port with BAUD=115200
- Prepare the TFTP server to your computer and put the downloaded files into the TFTP server. Set your computer ethernet interface to 192.168.1.99, and connect the ethernet to the EAP101 LAN port
- Power on or reboot the device and fast hit any key to stop autoboot, and access the U-boot terminal
EAP101

(1) After accessing the U-boot, proceed with the following command

```
# setenv active 1
# setenv upgrade available 1
# saveenv
# setenv ipaddr 192.168.1.11
# setenv serverip 192.168.1.99   //Your TFTP Server IP
# ping 192.168.1.99
# tftpboot openwrt-ipq807x-edgecore_eap101-squashfs-nand-factory.bin && imgaddr=$fileaddr
# imxtract $imgaddr ubi && nand device 0
# nand erase 0x00000000 0x03c00000
# nand write $fileaddr 0x00000000 $filesize
# nand erase 0x3c000000 0x3c000000
# nand write $fileaddr 0x3c000000 $filesize
# reset   //AP reboots, but we don’t access to U-boot
```

(2) Let the device boots into the AP Network Operating System (AP NOS)

(3) Login using default username ‘root’ and password ‘openwifi’

EAP102

(1) After accessing the U-boot, proceed with the following command

```
# setenv active 1
# setenv upgrade available 1
# saveenv
# setenv ipaddr 192.168.1.11
# setenv serverip 192.168.1.99   //Your TFTP Server IP
# ping 192.168.1.99
# tftpboot openwrt-ipq807x-edgecore_eap102-squashfs-nand-factory.bin && imgaddr=$fileaddr
# imxtract $imgaddr ubi && nand device 0
# nand erase 0x00000000 0x03400000
# nand write $fileaddr 0x00000000 $filesize
# nand erase 0x3c000000 0x34000000
# nand write $fileaddr 0x3c000000 $filesize
# reset
```

(2) Let the device boots into the AP Network Operating System (AP NOS)
(3) Login using default username ‘root’ and password ‘openwifi’

**EAP104**

(1) After accessing the U-boot, proceed with the following command

```bash
# tftpboot openwrt-ipq807x-edgecore_eap104-squashfs-nand-factory.bin && imgaddr=$fileaddr
# imxtract $imgaddr ubi && nand device 0
# nand erase 0x00080000 0x3e000000
# nand write $fileaddr 0x00080000 $filesize
# nand erase 0x3e80000 0x3e000000
# nand write $fileaddr 0x3e80000 $filesize
# reset
```

(2) Let the device boots into the AP Network Operating System (AP NOS)

(3) Login using default username ‘root’ and password ‘openwifi’

**ECW5211-L**

(1) After accessing the U-boot, proceed with the following command

```bash
# sf probe && sf erase 0xe0000 0x10000
# tftpboot 0x84000000 openwrt-ipq40xx-generic-edgecore_ecw5211-squashfs-nand-factory.ubi
# nand device 1 && nand erase 0x00000000 0x04000000
# nand write $fileaddr 0x00000000 0x012a0000
# reset
```

(2) Let the device boots into the AP Network Operating System (AP NOS)

(3) Login using default username ‘root’ and password ‘openwifi’

**ECW5410-L**

(1) After accessing the U-boot, proceed with the following command

```bash
# set ipaddr 192.168.1.11 && set serverip 192.168.1.99 && ping
```
192.168.1.99
# sf probe && sf erase 0x170000 0x10000
# tftpboot 0x44000000 openwrt-ipq806x-generic-edgecore_ecw5410-squashfs-nand-factory.bin
# nand device 0 && nand erase 0x00000000 0x04000000
# nand write $fileaddr 0x00000000 $filesize
# nand device 0 && nand erase 0x04000000 0x04000000
# nand write $fileaddr 0x04000000 $filesize
# reset

(2) Let the device boots into the AP Network Operating System (AP NOS)

(3) Login using default username ‘root’ and password ‘openwifi’

OAP100

(1) After accessing the U-boot, proceed with the following command

# sf probe && sf erase 0xe0000 0x10000
# tftpboot 0x84000000 openwrt-ipq40xx-generic-edgecore_oap100-squashfs-nand-factory.ubi
# nand device 0 && nand erase 0x00000000 0x04000000
# nand write $fileaddr 0x00000000 0x01740000
# reset

(2) Let the device boots into the AP Network Operating System (AP NOS)

(3) Login using default username ‘root’ and password ‘openwifi’