NRC7292 Evaluation Kit Application Note

(802.11s Mesh Network)

Ultra-low power & Long-range Wi-Fi

Ver 1.0
Nov 20, 2020

NEWRACOM, Inc.

NRC7292 Evaluation Kit Application Note (802.11s Mesh Network)
Ultra-low power & Long-range Wi-Fi

© 2020 NEWRACOM, Inc.

Newracom reserves the right to change in its products or product specification to improve function or design at any time without notice.

Office

Newracom, Inc.
25361 Commercentre Drive, Lake Forest, CA 92630 USA
http://www.newracom.com

Contents

List of Figures

1 Overview

IEEE 802.11s is an amendment to the 802.11 standard that enables wireless mesh network in the wireless local area network (WLAN). This amendment introduces a routing protocol, routing capabilities at the medium access control (MAC) layer, and security.

NRC7292 supports the wireless mesh network based on the IEEE 802.11s and provides many benefits including:

- **Easy coverage/range extension**

  In general, the coverage of the Wi-Fi network is limited by the output power of the access point (AP), and the transmission power is restricted by the regional regulation. Multi-hop routing of the mesh network makes it possible to extend the coverage of the Wi-Fi network.

- **Secure ad-hoc network**

  IEEE 802.11s standard adopts the simultaneous authentication of equals (SAE) as a mandatory authentication protocol. This protocol is the core component of WPA3-Personal so that users can deploy a secure ad-hoc network compared to the legacy 802.11 ad-hoc networks.

- **Self-organizing and resilient network**

  IEEE 802.11s standard set the hybrid wireless mesh protocol (HWMP) as a default routing protocol. This protocol is hybrid because it supports both proactive and reactive routing. The proactive routing establishes the routing path before any data flow, maintains and periodically updates the routing table, while the reactive routing establishes the path as needed for packet forwarding. Both routing technique enables a 11s mesh network to adapt network changes.

2 IEEE 802.11s Mesh Networks

2.1 Types of 802.11s Mesh Device

IEEE 802.11s standard defines a new type of service set called mesh basic service set (MBSS). In the MBSS, three different kinds of device can form the mesh network as shown in Figure 2.1.

- **Mesh Point (MP)**

  A mesh point is a station that has the 11s mesh capabilities to establish peer links with other MP neighbors in a range. It provides strong security based on SAE mechanism which is the core authentication technology of WFA3-Personal.

- **Mesh Point Portal (MPP)**
  A mesh point portal is a device that has mesh point functionality and provides inter-networking connectivity between the 11s mesh network and wired or other wireless networks such as long-term evolution (LTE).

- **Mesh Access Point (MAP)**
  A mesh assess point is equipped with the functionalities for mesh point as well as access point (AP) which provides BSS services to support communication with non-mesh stations. It may also connect to a wired network and act as an MPP.

⚠ ***In this release, two EVKs are needed and bridged to act as a MAP in the 802.11s mesh network. Detailed information is described in chapter 3.4.3.***

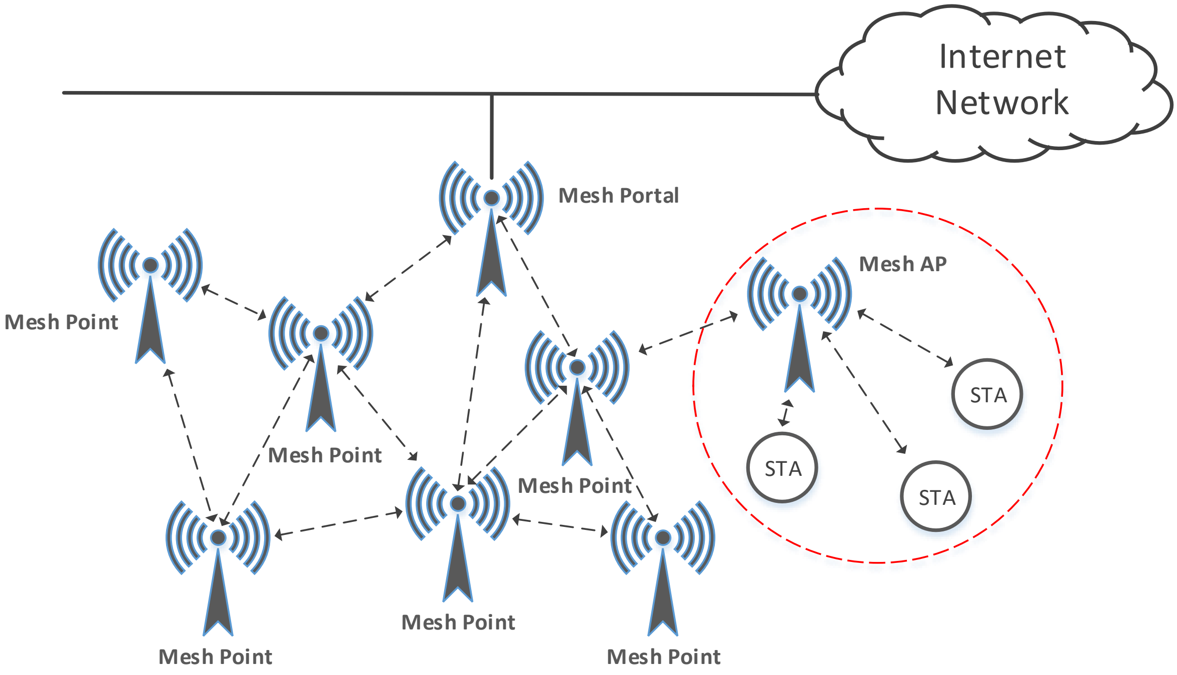


Figure 2.1 802.11s mesh network

2.2 Types of 802.11s Mesh Network

IEEE 802.11s standard enables two types of wireless mesh networks, ad-hoc (or mobile) and relatively fixed network. As shown in Figure 2.2, ad-hoc network consists of only mesh points. However, it is possible that one of the mesh points acts as a mesh point portal by providing inter-networking with wired or wireless networks.

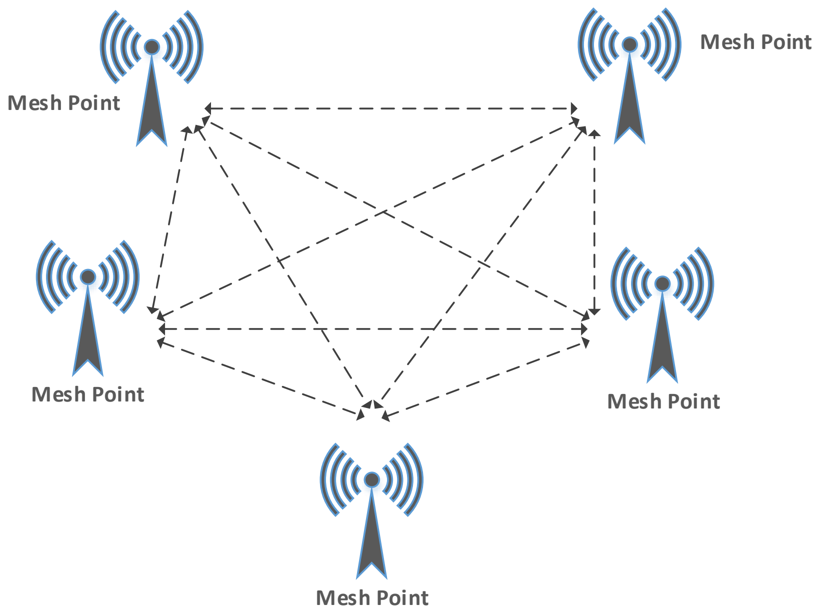


Figure 2.2 Example of 802.11s ad-hoc mesh network

The relatively fixed network represents the network that a fixed mesh point portal, mesh access points, and multiple mesh points form a wireless mesh network as shown in Figure 2.1. Non-mesh stations also can utilize the mesh network through the mesh access point.

2.3 802.11s Mesh Network Routing and Mesh Peering

As mentioned earlier, the IEEE 802.11s standard set the HWMP as a default routing protocol. Linux has provided this protocol in its kernel since 2013. The reference architecture of NRC7292 uses Linux as its host so that it provides all services that Linux's HWMP can provide.

Mesh peering is a logical relationship between two mesh points that are required to communicate directly over a single instance of the wireless medium (WM). A mesh point can establish a mesh peering with multiple neighbor mesh points manually or automatically.

- Manual peering
- Automatic peering

When selecting manual peering, the mesh point establishes a mesh peering with mesh points designated by users manually. On the other hand, automatic peering tries to set up a mesh peering to all mesh points in the range.

3 NRC7292 EVK 802.11s Mesh Test Bed Configuration

This chapter describes the steps to configure the IEEE 802.11s mesh network as shown in Figure 3.1. This simple mesh network consists of a mesh portal, a mesh access point, and a non-mesh station. As described above, the MAP is made of two EVKs connected over the bridge for mesh point and access point.

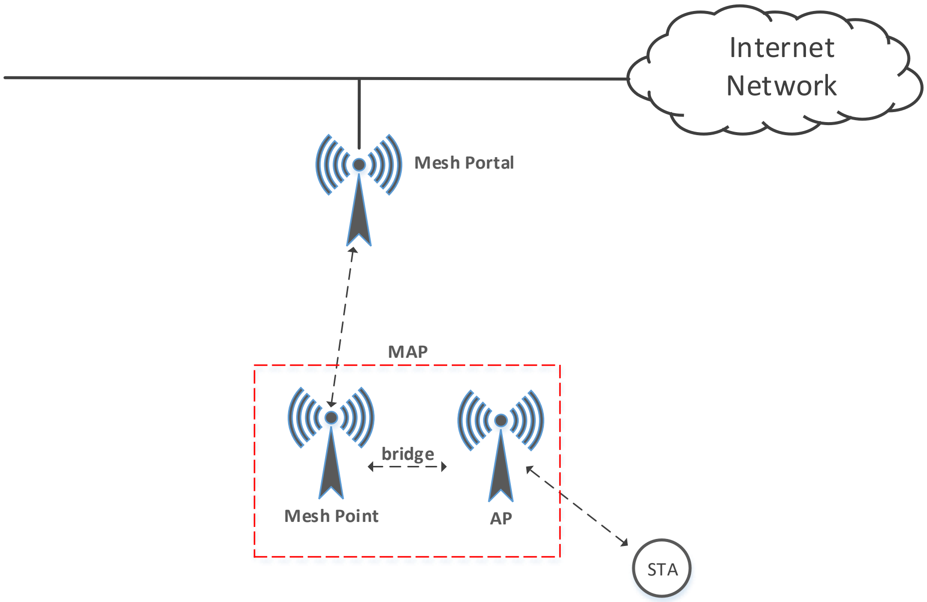


Figure 3.1 802.11s mesh network configuration example (MPP/MAP/STA network)

3.1 Prerequisite: Enable CONFIG_MESH

There is no dependency on the wpa_supplicant version, but it is recommended to use version 2.9 to enable WFA3-Personal (SAE) and CONFIG_MESH.

```
$ wget https://w1.fi/releases/wpa_supplicant-2.9.tar.gz
$ tar zxf wpa_supplicant-2.9.tar.gz
$ sudo apt-get update
$ sudo apt-get install libnl-3-dev libnl-genl-3-dev libssl-dev
$ sudo apt-get install libdbus-1-dev libdbus-glib-1-dev
$ cd wpa_supplicant-2.9/wpa_supplicant
$ cp defconfig .config

Edit .config to include below:
CONFIG_IEEE80211N=y
CONFIG_OWE=y
CONFIG_SAE=y
CONFIG_MESH=y
CONFIG_IEEE80211W=y

$ make -j4
$ sudo make install
```

3.2 Prerequisite: Install bridge tools

Since the MPP and MAP require bridge function, bridge tools should be installed beforehand as below.

```
$ sudo apt-get install bridge-utils
```

3.3 Run script: run_mesh.sh

The run_mesh.sh is the script that helps users to easily configure an EVK as MPP, MP, or MAP. It can be found in the `nrc_pkg/script/mesh` directory and following is the usage of `run_mesh.sh` script. The default security mode of 802.11s is WPA3-Personal (SAE).

```
run_mesh.sh -m mp -c <channel> -s <ssid> -p <password> -a <ip address> -k <peer MP's MAC Address>
```

- -a : static IP address (not necessary when DHCP service available)
- -k : peer MP's MAC address (only for manual peering)

As introduced earlier, two peering methods, manual and automatic peering, are to be used to form a mesh network. To use manual peering, users need to append the "no_auto_peer=1" at the end of network configuration in the write_config_mp() of `run_mesh.sh` file as shown below.

```
write_config_mp() {
        local FILENAME=$1
cat << EOF > $FILENAME
ctrl_interface=/var/run/wpa_supplicant
country=US
network={
        ssid="${SSID0}"
        mode=5
        frequency=${FREQ}
        key_mgmt=SAE
        psk="${PASSWORD0}"
        no_auto_peer=1
}
p2p_disabled=1
EOF
}
```

Besides, user should give the MAC address of the peer MP by using '-k' option as an example below.

Example usage for manual peering

```
$ ./run_mesh.sh -m mp -c 159 -s halow_mesh -p 12345678 -a 192.168.50.1 -k 8c:0f:fa:00:27:93
```

For automatic peering, the 'no_auto_peer=1' should not be in the `run_mesh.sh` script file and '-k' option is not necessary.

Example usage for automatic peering

```
$ ./run_mesh.sh -m mp -c 159 -s halow_mesh -p 12345678 -a 192.168.50.1
```

3.4 802.11s Mesh Device Configuration

This chapter describes detailed steps to configure IEEE 802.11s mesh network as shown in Figure 3.1 with NRC7292 EVKs. This example assumes that each device uses a static IP address. Users can assign a static IP address by using -a option of `run_mesh.sh` script.

3.4.1 MP Configuration

Example parameters for MP:

- MAC address: 00:01:02:03:04:52
- IP address: 192.168.222.2
- Channel: 159
- SSID: halow_mesh
- Password: 12345678

Following is the usage of the run_mesh.sh script for MP configuration with above parameters.

```
$ ./run_mesh.sh -m mp -c 159 -s halow_mesh -p 12345678 -a 192.168.222.2
```

The SAE is the default security for IEEE 802.11s mesh network. However, users can configure a different mode such as open and WPA2 by changing the `key_mgmt` to NONE and WPA-PSK, respectively as below, but it is for only test purposes.

```
write_config_mp() {
        local FILENAME=$1
cat << EOF > $FILENAME
ctrl_interface=/var/run/wpa_supplicant
country=US
network={
        ssid="${SSID0}"
        mode=5
        frequency=${FREQ}
        #key_mgmt=NONE
        key_mgmt=WPA-PSK
        psk="${PASSWORD0}"
        #no_auto_peer=1
}
p2p_disabled=1
EOF
}
```

The run_mesh.sh script provides a method to establish a manual peering, but to only one neighbor MP. To add more manual peering, users can use the `wpa_cli` command as below.

```
$ sudo wpa_cli -i wlan0 mesh_peer_add {Peer MP's MAC_Address}
```

3.4.2 MPP Configuration

Example parameters for MPP:

- MAC address: 00:01:02:03:04:51
- IP address: 192.168.222.1
- Channel: 159
- SSID: halow_mesh
- Password: 12345678

Like MP's configuration, users can configure MPP by using the `run_mesh.sh` script as below.

```
$ ./run_mesh.sh -m mp -c 159 -s halow_mesh -p 12345678 -a 192.168.222.1
```

To provide inter-networking with a wired network, users should set up a NAT by the below procedures. Eth0 is used for the backhaul in this example.

```
$ sudo iptables -F
$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
$ sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

To enable proactive routing, following configuration is needed. This makes other MPs create paths to MPP, and all MPs would send all traffic to MPP if they failed to resolve the destination in mesh network. But this also introduces some management and data traffic overhead. So, it is recommended to keep the number of MPP down.

```
$ sudo iw wlan0 set mesh_param mesh_hwmp_rootmode=3
$ sudo iw wlan0 set mesh_param mesh_gate_announcements=1
```

3.4.3 MAP Configuration

As described in chapter 2, MAP is composed of two NRC7292 EVKs: one for MP and the other for AP.

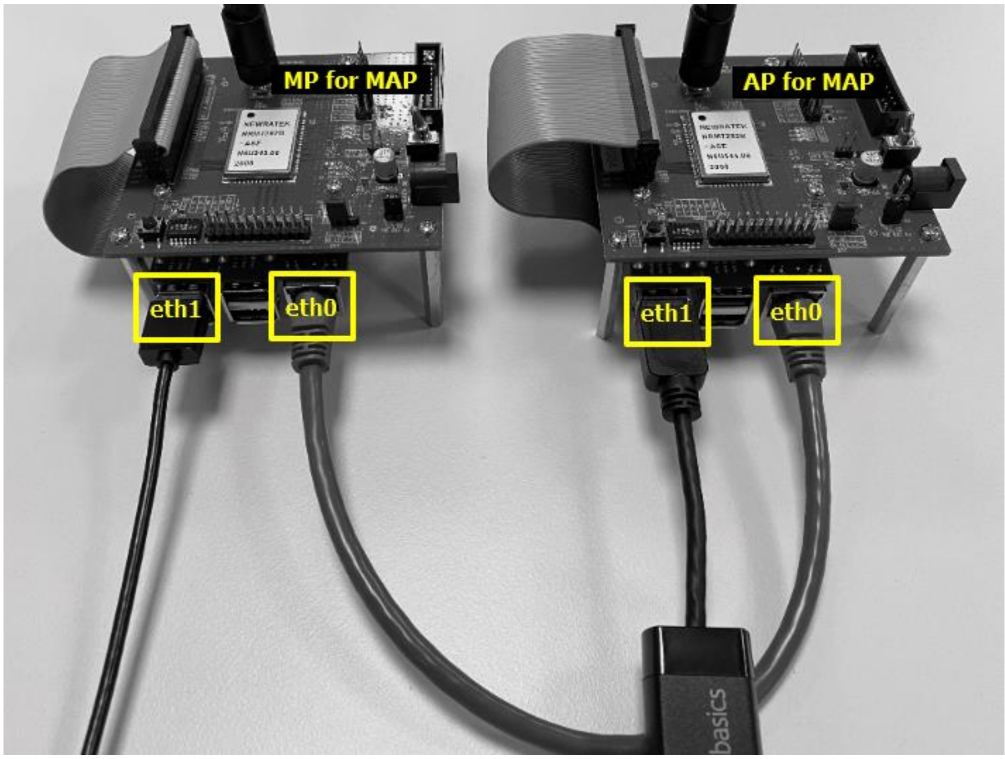


Figure 3.2 NRC7292 EVKs for MAP Configuration

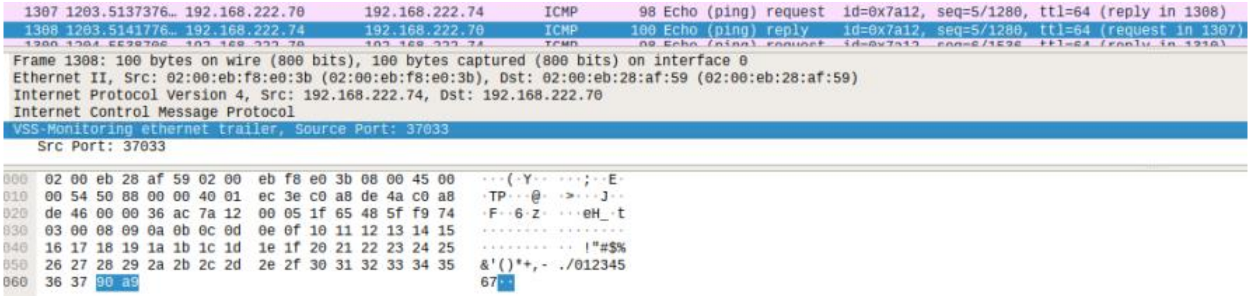
In this figure, there are two ethernet ports: eth0 for the bridge interface between MP and AP and eth1 for the host platform (Raspberry Pi) control interface. The eth1 interface is only for test purposes. So, this extra ethernet interface and cable can be removed in the final mesh device deployment.

Different combinations of ethernet interfaces can be used, but please be cautious about the below check point when an external USB-to-serial ethernet adapter is used for the bridge interface.

Ethernet Bridge Check Point

For the bridge connection between MP and AP, USB-to-serial ethernet adapter can be used, but there could be a problem as follows. **If you meet any bridge problem at MAP, please check this first.**

In some vendors, VSS-Monitoring ethernet trailer is appended at the end of packet and this trailer can trigger the ICMP/IP packet delivery problem.



```
 1307 1203.5137376… 192.168.222.70       192.168.222.74       ICMP       98 Echo (ping) request  id=0x7a12, seq=5/1280, ttl=64 (reply in 1308)
 1308 1203.5141776… 192.168.222.74       192.168.222.70       ICMP      100 Echo (ping) reply    id=0x7a12, seq=5/1280, ttl=64 (request in 1307)
 1309 1204.5538786… 192.168.222.70       192.168.222.74       ICMP       98 Echo (ping) request  id=0x7a12, seq=6/1536, ttl=64 (reply in 1310)
Frame 1308: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
Ethernet II, Src: 02:00:eb:f8:e0:3b (02:00:eb:f8:e0:3b), Dst: 02:00:eb:28:af:59 (02:00:eb:28:af:59)
Internet Protocol Version 4, Src: 192.168.222.74, Dst: 192.168.222.70
Internet Control Message Protocol
VSS-Monitoring ethernet trailer, Source Port: 37033
   Src Port: 37033

000  02 00 eb 28 af 59 02 00  eb f8 e0 3b 08 00 45 00   ···(·Y·· ···;··E·
010  00 54 50 88 00 00 40 01  ec 3e c0 a8 de 4a c0 a8   ·TP···@· ·>···J··
020  de 46 00 00 36 ac 7a 12  00 05 1f 65 48 5f f9 74   ·F··6·z· ···eH_·t
030  03 00 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ········ ········
040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
060  36 37 90 a9                                       67··
```

Figure 3.3 VSS-Monitoring ethernet trailer

If user use Raspbian Linux 9 (Stretch) and AmazonBasics USB 3.0 to 10/100/1000 Gigabit Ethernet Internet Adapter, user can meet this issue. Default driver for AX88xxx device on Raspbian Linux 9 has this problem. So, you should update AX88xxx driver with following driver.

- Driver URL: https://www.asix.com.tw/en/product/USBEthernet/Super-Speed_USB_Ethernet/AX88179
- We verified its proper operation with Linux kernel 5.x/4.x/3.x/2.6.x Driver v1.20.0 (4/27/2020) as below.

```
$ tar jxf AX88179_178A_LINUX_DRIVER_v1.20.0_SOURCE.tar.bz2
$ cd AX88179_178A_Linux_Driver_v1.20.0_source/
$ make
$ sudo make install
```

Example parameters for MP or MAP:

- MAC address: 00:01:02:03:04:53
- IP address: 192.168.222.3
- Channel: 159
- SSID: halow_mesh
- Password: 12345678

MP configuration of MAP device is the same as MP only device except bridge setup.

```
$ ./run_mesh.sh -m mp -c 159 -s halow_mesh -p 12345678 -a 192.168.222.3
```

Example parameters for AP of MAP:

- IP address: 192.168.222.4
- Channel: 161
- SSID: halow_mesh_ap
- Password: 12345678

Users can allocate the same channel for both the mesh and the infrastructure network. However, the performance may better when using a different channel. Moreover, users can change the parameters for AP of MAP by modifying the `ap_sdk_map.conf` file as below.

```
ssid=halow_mesh_ap
channel=161
wpa=2
wpa_key_mgmt=SAE
wpa_pairwise=CCMP
rsn_pairwise=CCMP
wpa_passphrase=12345678
```

To configure bridge on both MP and AP device, following steps are necessary.

- Step 1: bridge setup on MP for MAP

```
$ sudo vi /etc/dhcpcd.conf → add following two lines
denyinterfaces wlan0
denyinterfaces eth0
$ sudo brctl addbr br0
$ sudo ifconfig wlan0 0.0.0.0
$ sudo ifconfig eth0 0.0.0.0
$ sudo brctl addif br0 wlan0
$ sudo brctl addif br0 eth0
$ sudo ifconfig br0 192.168.222.4
```

- Step 2: bridge setup on AP for MAP (map.sh script used for starting AP)

```
$ sudo vi /etc/network/interfaces → add following three lines
auto br0
iface br0 inet manual
bridge_ports br0 eth0 wlan0
$ sudo vi /etc/dhcpcd.conf → add following two lines
denyinterfaces wlan0
denyinterfaces eth0
$ ./map.sh
```

3.4.4 STA Configuration

To run STA, users can use start.py scripts located in `nrc_pkg/script/conf` directory.

For more information about the usage of start.py script, please refer to the NRC7292 Evaluation Kit User Guide (Host Mode) document.

```
$ ./start.py 0 3 US
```

3.4.5 802.11s Mesh Commands

Following two commands can be used to check the 802.11s mesh connection status.

- To see a list of mesh paths: `sudo iw dev wlan0 mpath dump`
- To get current WPA status: `sudo wpa_cli status`

```
$ sudo iw dev wlan0 mpath dump
DEST ADDR         NEXT HOP          IFACE     SN     METRIC  QLEN    EXPTIME       DTIM    DRET    FLAGS
02:00:eb:4e:31:58 02:00:eb:4e:31:58 wlan0      182     1366    0      0       100     0       0x14
$ sudo wpa_cli status
Selected interface 'wlan0'
bssid=00:00:00:00:00:00
freq=5780
ssid=halow_mesh
id=0
mode=mesh
pairwise_cipher=UNKNOWN
group_cipher=UNKNOWN
key_mgmt=UNKNOWN
wpa_state=COMPLETED
address=02:00:eb:f8:e0:3b
uuid=21bf22ea-9b6d-5a4f-86a8-a1643d608da9
```

Following iw commands also provide the way to handle mesh path and mesh peer link.

- To delete a mesh path entry: `sudo iw dev wlan0 mpath del <MP MAC address>`
- To force a specific mesh network topology: `sudo iw dev wlan0 mpath new <MP MAC address> next_hop <Next hop MP MAC address>`
- To list all mesh peer links: `sudo iw dev wlan0 station dump`
- To open/block a specific mesh peer link: `sudo iw dev wlan0 station set <MP MAC address> plink_action [open|block]`
- To delete a mesh peer link: `sudo iw dev wlan0 station del <MP MAC address>`
- To adjust path request/reply action frames interval: `sudo iw dev wlan0 set mesh_hwmp_root_interval <time in the unit of millisecond(ms)>`

⚠ ***In this release, only wlan0 interface should be used for mesh network. So, it is not permissible to use a virtual interface by using iw interface add command.***

4 Revision history

Revision No	Date	Comments
Ver 1.0	11/20/2019	Initial version