NRC7292 Application Note
(Dual-band)

Ultra-low power & Long-range Wi-Fi

Ver 1.0
Apr 12, 2019

NEWRACOM, Inc.

NRC7292 Application Note (Dual-band)
Ultra-low power & Long-range Wi-Fi Module

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

Contents

List of Figures

1 Overview

This document describes 802.11ah HaLow dual-band mode (AP and Bridge), and how to run them on NRC7292 EVK. Basic operations as HaLow Singla-band STA and AP are described in the document "NRC7292_EVK_User_Guide_HM", therefore we just focus on HaLow Dual-band mode in this document. Figure 1.1 shows HaLow Single-band STA/AP and network topology where HaLow STA connects to HaLow AP and Internet access is possible via NAT (Network Address Translation) on Ethernet as other Wi-Fi 4 (IEEE802.11n) or Wi-Fi 5 (IEEE802.11ac) APs do. Dual-band HaLow AP is almost the same as the single-band (Sub1G) HaLow AP except that it can support another 2.4G Wi-Fi 4 Band. Figure 1.2 shows it with possible network topology.

> ⚠ ***Onboard Wi-Fi on Raspberry Pi 3 Model B we provide as a Host of NRC7292 EVK only supports 2.4GHz (i.e. Do NOT support 5GHz)***

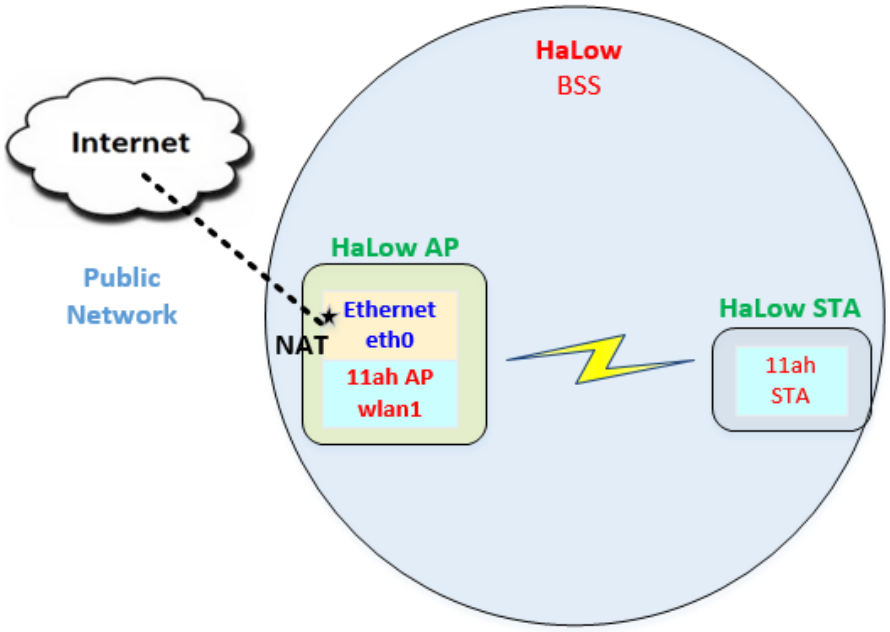

Figure 1.1 Single-band (SubS1G) HaLow AP and STA

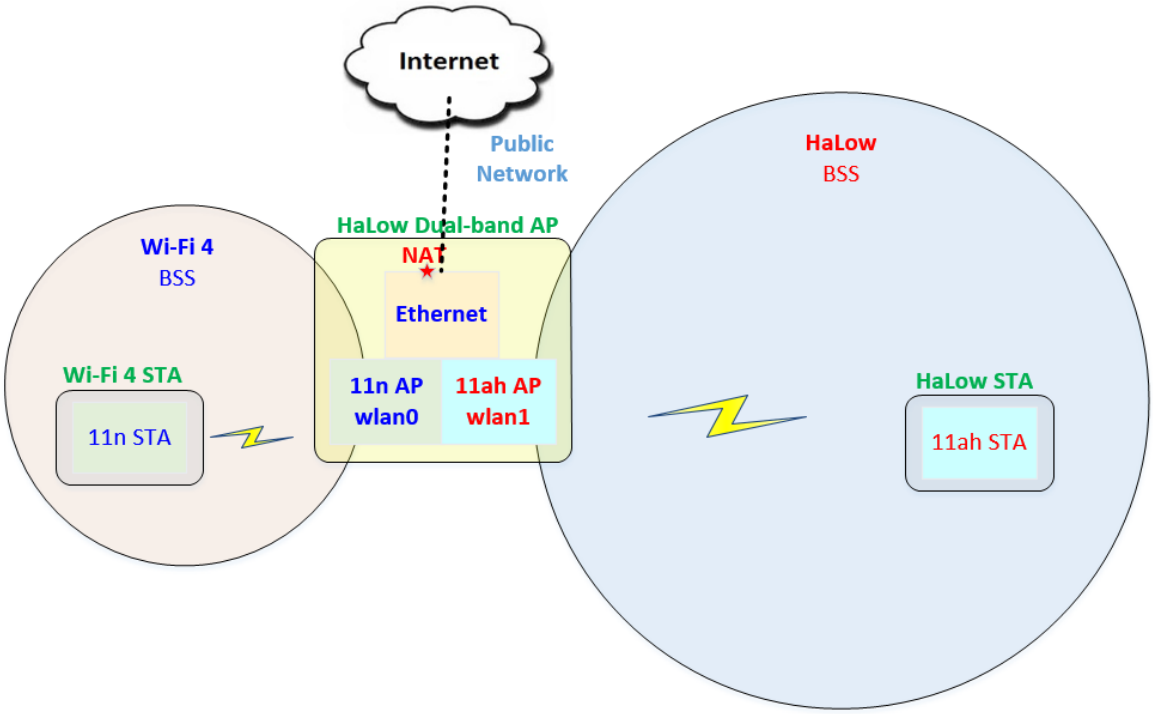

Figure 1.2 Dual-Band(SubS1G + 2.4G) HaLow AP

HaLow Bridge mode is the same concept as Dual-band HaLow AP since it supports both HaLow and Wi-Fi 4. However, it does NOT have Ethernet Interface for Internet access because the main goal of HaLow Bridge is just to forward frames from Wi-Fi 4 network to HaLow network and vice versa. Using HaLow bridge, Wi-Fi 4 devices can extend their coverage or HaLow devices can take advantage of legacy Wi-Fi 4 Infrastructure already installed in many venues. Figure 1.3 ~ Figure 1.5 show network topologies with HaLow bridge where many use cases can be made like wireless backhaul, long-range Ad-hoc, etc.

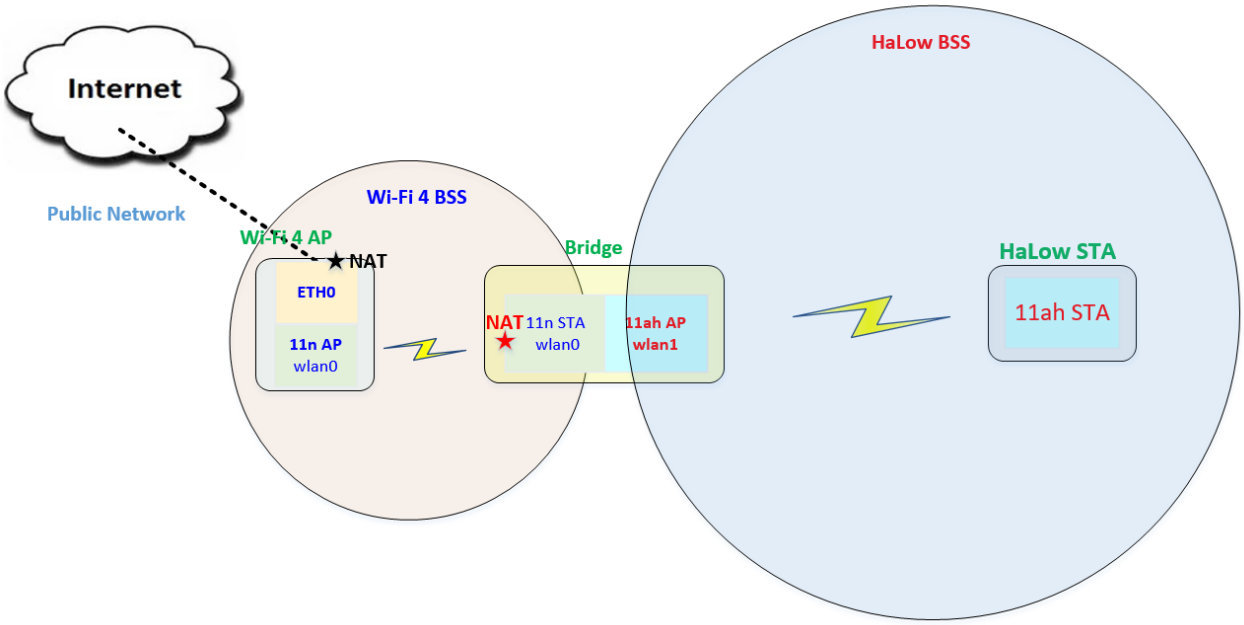

Figure 1.3 Network Topology with HaLow Bridge #1

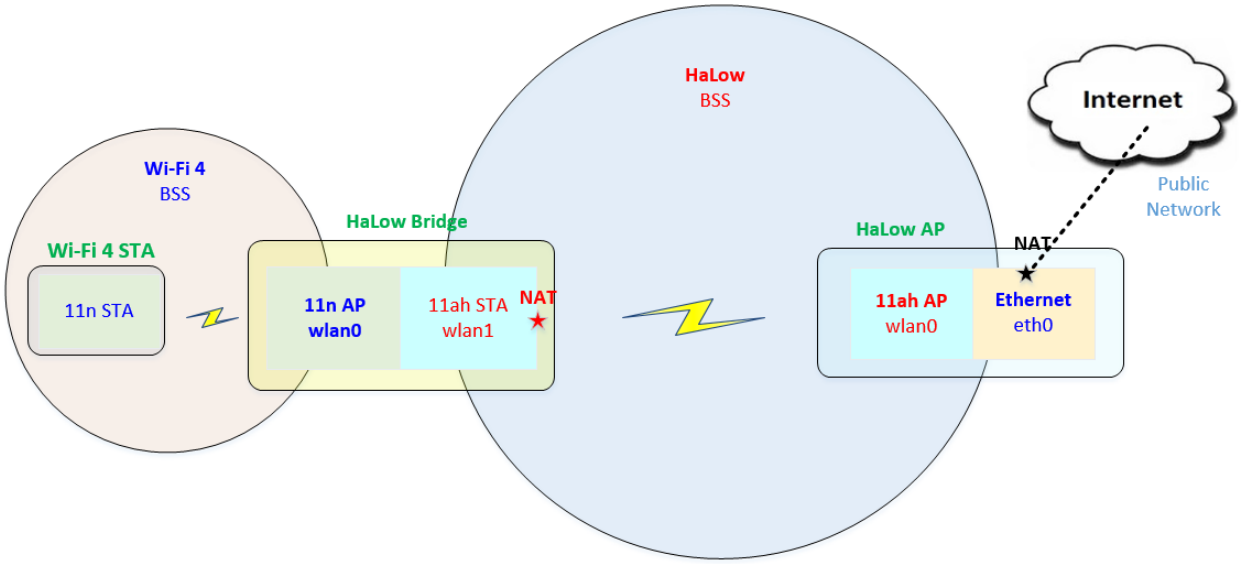

Figure 1.4 Network Topology with HaLow Bridge #2

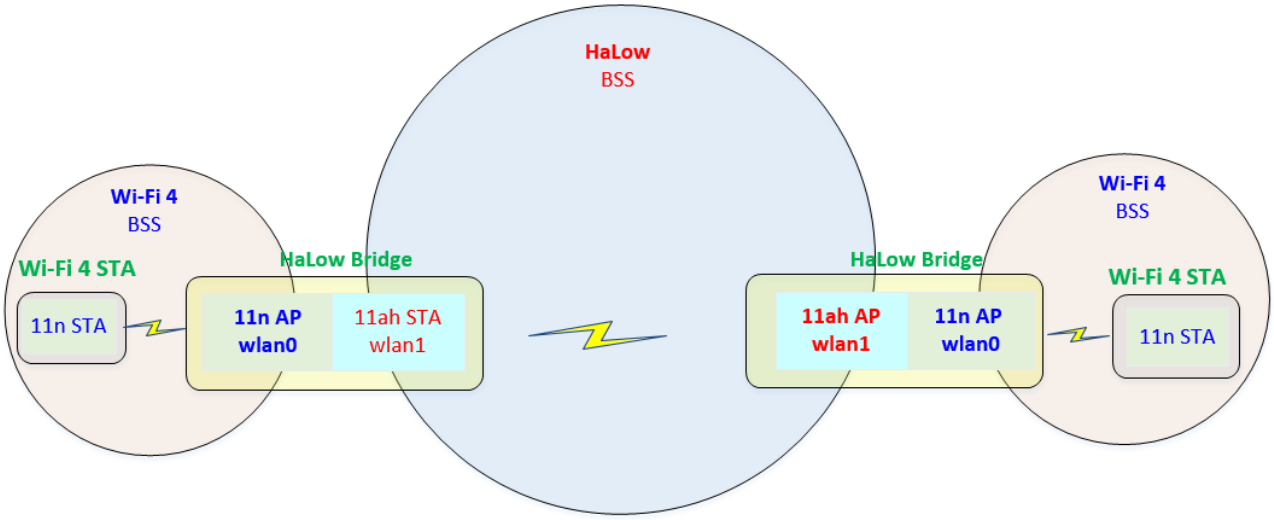

Figure 1.5 Network Topology with HaLow Bridge #3

2 Enable Wi-Fi 4

NRC7292 EVK we provide is just for Single-band (Sub1G) HaLow AP so internal Wi-Fi 4 on RPi is disabled as default. Therefore, it should be revived before enabling HaLow Dual-band mode. We describe how to enable it in this chapter.

2.1 Check Wi-Fi 4 overlay

First users check that there is “overlay for disabling onboard Wi-Fi” that should be commented or removed to use Wi-Fi 4. Figure 2.1 shows how to uncomment it in /boot/config.txt. If there is no line for overlay of Wi-Fi in config.txt file, just skip this procedure.

```
# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
enable_uart=1
dtoverlay=pi3-disable-bt
#dtoverlay=pi3-disable-wifi
dtoverlay=newracom
gpu_mem=128
```

Figure 2.1 config.txt

2.2 Enable Wi-Fi 4 modules

Modules related to Wi-Fi 4 should be run while booting procedure of RPi. brcmfmac and brcmutil are modules for Wi-Fi 4. Therefore, users should uncomment or remove them if they are on blacklist, “/etc/modprobe.d/raspi-blacklist.conf”. Figure 2.2 shows raspi-blacklist.conf file where brcmfmac and brcmutils are removed from blacklist

```
pi@raspberrypi:~ $ cd /etc/modprobe.d/
pi@raspberrypi:/etc/modprobe.d $ cat raspi-blacklist.conf
#blacklist brcmfmac
#blacklist brcmutil
```

Figure 2.2 raspi-blacklist.conf

2.3 Reboot Raspberry Pi 3

Once two procedures above are done, Wi-Fi 4 is now ready to use. Users should reboot the system to apply it. Users can check and see there is wlan0 interface for Wi-Fi 4 via ifconfig command on RPi.

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.4  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::a2d4:3e84:764f:35e1  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:4d:63:fb  txqueuelen 1000  (Ethernet)
        RX packets 107  bytes 12169 (11.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 130  bytes 22332 (21.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 34  bytes 4932 (4.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34  bytes 4932 (4.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:18:36:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 2.3 wlan0 interface

3 NRC7292 EVK Bridge operation

This chapter describes how to run HaLow dual-band AP and Bridge operation. Parameters related to dual-band mode in ~/nrc_pkg/script/start.py, a script used to initiate HaLow STA or AP, should be set properly according to the role of NRC7292 EVK such as HaLow Dual-band AP, HaLow AP/AP Bridge, HaLow AP/STA Bridge, etc.

3.1 Start HaLow Dual-band AP

To start dual-band AP, Wi-Fi 4 and NAT should be enabled but also role of Wi-Fi 4 should be set “AP”. Figure 3.1 shows the parameters in ~/nrc_pkg/script/start.py set for dual-band AP. Both of network interfaces, wlan0 and wlan1, have static IP. Plus, DHCP server and DNS server should be run to fulfill the duty of AP. Configuration files related to Static IP, DHCP server, and DNS server are in ~/nrc_pkg/etc/dhcpcd and ~/nrc_pkg/etc/dhcpcd/dnsmasq folder. Each of configuration file is different according to STA type (AP or STA) of each interface. However, there is no worry to set manually because configurations are set automatically while start.py is running according to parameters set by users.

```
################################################################################
# Default Configuration (you can change value you want here)
model       = 7292     # 7292 or 7192
hif_speed   = 16000000 # HSPI Clock
gain_type   = 'phy'    # 'phy' or 'nrf(legacy)'
txpwr_val   = 17       # TX Power
maxagg_num  = 8        # 0(AMPDU off) or  >2(AMPDU on)
cqm_off     = 0        # 0(CQM on) or 1(CQM off)
fw_download = 1        # 0(FW Download off) or 1(FW Download on)
fw_name     = 'uni_s1g.bin'
guard_int   = 'long'   # 'long'(LGI) or 'short'(SGI)
concurrent  = 0        # 0(Concurrent Mode off) or 1(Coucurrent Mode on)
interface_11ah   = 'wlan1'  # 11ah driver interface : 'wlan0' or 'wlan1'
supplicant_debug = 0   # WPA Supplicant debug option : 0(off) or 1(on)
hostapd_debug    = 0   # Hostapd debug option    : 0(off) or 1(on)
max_cpuclock     = 1   # RPi Max CPU Clock : 0(off) or 1(on)
# Optional Configuration for dual-band (11N + 11AH)
enable_11n        = 1        # Enable Dual-Band (11ah + 11n)
role_11n          = 'AP'     # Role of 11N ('AP' or 'STA')
interface_11n     = 'wlan0'  # 11n driver interface : 'wlan0'
disable_NAT       = 0  # 1(NAT disable) 0(NAT enable)
power_save        = 0  # power save : 0(off) or 1(on)
################################################################################
```

Figure 3.1 Parameters for HaLow Dual-band AP

Figure 3.2 and Figure 3.3 show logs displayed while start.py is running. Start.py script first makes HaLow AP started and then Wi-Fi 3 AP. Each of SSID for wlan0 and wlan1 is set as “nrc_11n_ap_1” and “halow_dual_ap” respectively. SSID can be changed via conf. file, ~/nrc_pkg/script/con/US/ap_halow_open.conf and ~/nrc_pkg/script/conf/11N/hostapd_AP.conf. DHCP and DNS server are also started at the same time and then EVK is finally ready to be operated as a HaLow Dual-band AP. Figure 3.4 and Figure 3.5 shows ping results from Wi-Fi 4 STA to google.com

and HaLow STA to google.com. Of course, ping between Wi-Fi 4 STA and HaLow STA is successful and Figure 3.6 shows it.

```
pi@raspberrypi:~/nrc_pkg/script $ ./start.py 1 0 US
------------------------------
Model            : 7292
STA Type         : AP
Security Mode    : OPEN
Country Selected : US
Download FW      : uni_s1g.bin
Interface        : wlan1
TX Gain          : 17
RX Gain          : 85
------------------------------
11n Interface    : wlan0
11n Role         : AP
------------------------------
NRC AP setting for HaLow...
[0] Clear
wpa_supplicant: no process found
wireshark-gtk: no process found
11n inteface (wlan0) down
[1] Copy
total 428
drwxr-xr-x 2 pi pi   4096 Apr 15 09:48 .
drwxr-xr-x 4 pi pi   4096 Apr 15 09:48 ..
-rwxr-xr-x 1 pi pi    218 Apr 15 09:48 copy
-rwxr-xr-x 1 pi pi 212560 Apr 15 09:48 nrc7292_cspi.bin
-rwxr-xr-x 1 pi pi 212560 Apr 15 11:38 uni_s1g.bin
-rw-r--r-- 1 root root 212560 Apr 15 11:38 /lib/firmware/uni_s1g.bin
[2] Loading module
sudo insmod ~/nrc_pkg/sw/driver/nrc.ko fw_name=uni_s1g.bin disable_cqm=0 hifspeed=16000000
[3] Set trx gain
phy rxgain 85
success
phy txgain 17
success
[4] Set aggregation number
set maxagg 1 0
success
[5] Set guard interval
set gi long
success
[6] Start hostapd
Configuration file: /home/pi/nrc_pkg/script/conf/US/ap_halow_open.conf
wlan1: interface state UNINITIALIZED->COUNTRY_UPDATE
Using interface wlan1 with hwaddr 02:00:eb:4d:63:fb and ssid "halow_dual_ap"
wlan1: interface state COUNTRY_UPDATE->ENABLED
wlan1: AP-ENABLED
[7] Setting NAT
INPUT:wlan1 OUTPUT:eth0
[8] ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.4  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::a2d4:3e84:764f:35e1  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:4d:63:fb  txqueuelen 1000  (Ethernet)
        RX packets 3205  bytes 430514 (420.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5126  bytes 596916 (582.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 84  bytes 8494 (8.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 8494 (8.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::20b9:2098:da53:7110  prefixlen 64  scopeid 0x20<link>
        ether 02:00:eb:4d:63:fb  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 2191 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

HaLow AP ready
```

Figure 3.2 Results of running HaLow dual-band AP (1/2)

```
[*] Enable 11n interface(wlan0)
[*] Start 11n AP
Configuration file: /home/pi/nrc_pkg/script/conf/11N/hostapd_AP.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:18:36:ae and ssid "nrc_11n_ap_1"
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
[*] Start dnsmasq service
Done.
pi@raspberrypi:~/nrc_pkg/script $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.4  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::a2d4:3e84:764f:35e1  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:4d:63:fb  txqueuelen 1000  (Ethernet)
        RX packets 3241  bytes 432728 (422.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5163  bytes 600818 (586.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 84  bytes 8494 (8.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 8494 (8.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.1  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::35b7:d589:491a:f9f7  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:18:36:ae  txqueuelen 1000  (Ethernet)
        RX packets 5385  bytes 486087 (474.6 KiB)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 2906  bytes 903049 (881.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.1  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::20b9:2098:da53:7110  prefixlen 64  scopeid 0x20<link>
        ether 02:00:eb:4d:63:fb  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29  bytes 4438 (4.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.3 Results of running HaLow dual-band AP (2/2)

```
pi@raspberrypi:~/nrc_pkg/script $ ping google.com
PING google.com (216.58.221.238) 56(84) bytes of data.
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=1 ttl=48 time=46.1 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=2 ttl=48 time=46.1 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=3 ttl=48 time=46.8 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=4 ttl=48 time=47.2 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=5 ttl=48 time=61.1 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=6 ttl=48 time=47.2 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=7 ttl=48 time=50.5 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=8 ttl=48 time=62.2 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=9 ttl=48 time=46.9 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=10 ttl=48 time=48.5 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=11 ttl=48 time=45.5 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=12 ttl=48 time=59.2 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=13 ttl=48 time=49.2 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=14 ttl=48 time=45.7 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=15 ttl=48 time=46.5 ms
64 bytes from hkg07s21-in-f238.1e100.net (216.58.221.238): icmp_seq=16 ttl=48 time=44.8 ms
```

Figure 3.4 Results of ping to google.com on HaLow STA

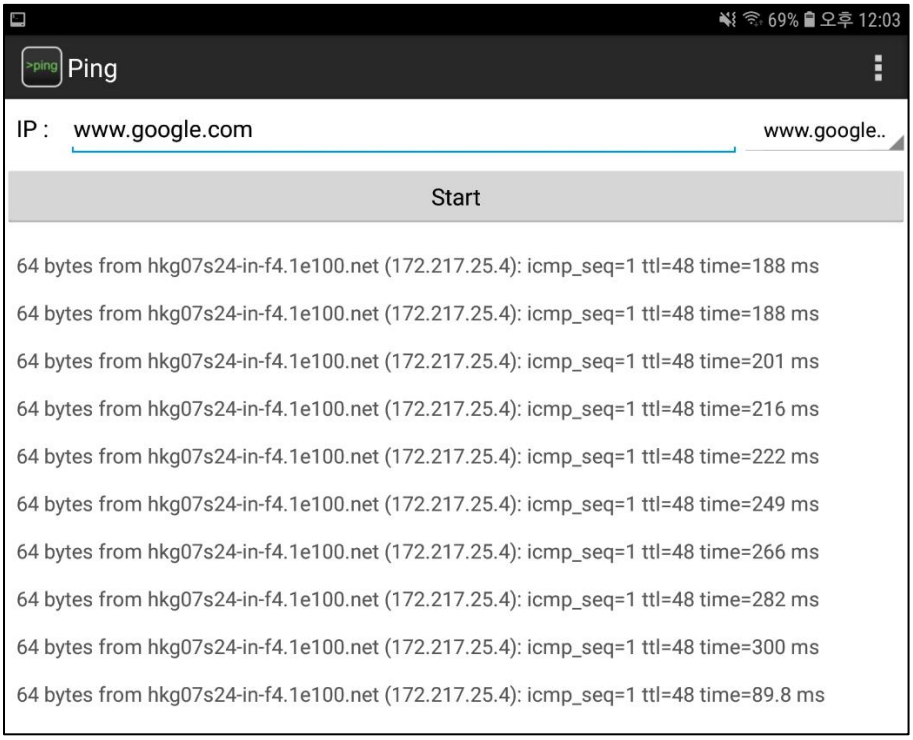

Figure 3.5 Results of ping to google.com on Wi-Fi 4 STA

```
pi@raspberrypi:~/nrc_pkg/script $ ping 192.168.100.37
PING 192.168.100.37 (192.168.100.37) 56(84) bytes of data.
64 bytes from 192.168.100.37: icmp_seq=1 ttl=63 time=121 ms
64 bytes from 192.168.100.37: icmp_seq=2 ttl=63 time=143 ms
64 bytes from 192.168.100.37: icmp_seq=3 ttl=63 time=164 ms
64 bytes from 192.168.100.37: icmp_seq=4 ttl=63 time=57.0 ms
64 bytes from 192.168.100.37: icmp_seq=5 ttl=63 time=211 ms
64 bytes from 192.168.100.37: icmp_seq=6 ttl=63 time=242 ms
64 bytes from 192.168.100.37: icmp_seq=7 ttl=63 time=49.7 ms
64 bytes from 192.168.100.37: icmp_seq=8 ttl=63 time=73.2 ms
64 bytes from 192.168.100.37: icmp_seq=9 ttl=63 time=95.9 ms
64 bytes from 192.168.100.37: icmp_seq=10 ttl=63 time=118 ms
64 bytes from 192.168.100.37: icmp_seq=11 ttl=63 time=140 ms
```

Figure 3.6 Results of ping between HaLow STA and Wi-Fi 3 STA

3.2 Start HaLow Bridge

Figure 3.7 shows sample parameters of “start.py” script for HaLow Bridge without NAT. Parameter, enable_11n, must be set as 1 to use HaLow Bridge. Setting parameters, role_11n and disable_NAT, depend on network configuration users want to make. (refer to Figure 1.3, Figure 1.4 and Figure 1.5)

```
################################################################################
# Default Configuration (you can change value you want here)
model       = 7292     # 7292 or 7192
hif_speed   = 16000000 # HSPI Clock
gain_type   = 'phy'    # 'phy' or 'nrf(legacy)'
txpwr_val   = 17       # TX Power
maxagg_num  = 8        # 0(AMPDU off) or  >2(AMPDU on)
cqm_off     = 0        # 0(CQM on) or 1(CQM off)
fw_download = 1        # 0(FW Download off) or 1(FW Download on)
fw_name     = 'uni_s1g.bin'
guard_int   = 'long'   # 'long'(LGI) or 'short'(SGI)
concurrent  = 0        # 0(Concurrent Mode off) or 1(Coucurrent Mode on)
interface_11ah   = 'wlan1'  # 11ah driver interface : 'wlan0' or 'wlan1'
supplicant_debug = 0   # WPA Supplicant debug option : 0(off) or 1(on)
hostapd_debug    = 0   # Hostapd debug option    : 0(off) or 1(on)
max_cpuclock     = 1   # RPi Max CPU Clock : 0(off) or 1(on)
# Optional Configuration for dual-band (11N + 11AH)
enable_11n        = 1        # Enable Dual-Band (11ah + 11n)
role_11n          = 'AP'     # Role of 11N ('AP' or 'STA')
interface_11n     = 'wlan0'  # 11n driver interface : 'wlan0'
disable_NAT       = 1  # 1(NAT disable) 0(NAT enable)
power_save        = 0  # power save : 0(off) or 1(on)
################################################################################
```

Figure 3.7 Parameters for HaLow Bridge Mode

Figure 3.8 and Figure 3.9 show the whole procedures of starting HaLow Bridge mode where HaLow AP and Wi-Fi 4 AP are enabled in order. Each SSID of HaLow AP and Wi-Fi 4 AP is assigned as “halow_dual_ap” and “nrc_11n_ap_1” as default. HaLow STA and Wi-Fi 4 STA can connect to its own AP using each SSID.

```
pi@raspberrypi:~/nrc_pkg/script $ ./start.py 1 0 US
------------------------------
Model             : 7292
STA Type          : AP
Security Mode     : OPEN
Country Selected  : US
Download FW       : uni_s1g.bin
Interface         : wlan1
TX Gain           : 17
RX Gain           : 85
------------------------------
11n Interface     : wlan0
11n Role          : AP
------------------------------
NRC AP setting for HaLow...
[0] Clear
wpa_supplicant: no process found
wireshark-gtk: no process found
11n inteface (wlan0) down
[1] Copy
total 428
drwxr-xr-x 2 pi pi   4096 Apr 11 20:08 .
drwxr-xr-x 4 pi pi   4096 Apr 11 20:08 ..
-rwxr-xr-x 1 pi pi    218 Apr 11 20:08 copy
-rwxr-xr-x 1 pi pi 212560 Apr 11 20:08 nrc7292_cspi.bin
-rwxr-xr-x 1 pi pi 212560 Apr 12 16:36 uni_s1g.bin
-rw-r--r-- 1 root root 212560 Apr 12 16:36 /lib/firmware/uni_s1g.bin
[2] Loading module
sudo insmod ~/nrc_pkg/sw/driver/nrc.ko fw_name=uni_s1g.bin disable_cqm=0 hifspeed=16000000
[3] Set trx gain
phy rxgain 85
success
phy txgain 17
success
[4] Set aggregation number
set maxagg 1 0
success
[5] Set guard interval
set gi long
success
[6] Start hostapd
Configuration file: /home/pi/nrc_pkg/script/conf/US/ap_halow_open.conf
wlan1: interface state UNINITIALIZED->COUNTRY_UPDATE
Using interface wlan1 with hwaddr 02:00:eb:0e:07:04 and ssid "halow_dual_ap"
wlan1: interface state COUNTRY_UPDATE->ENABLED
wlan1: AP-ENABLED
wlan1: STA 02:00:eb:cc:66:41 IEEE 802.11: authenticated
wlan1: STA 02:00:eb:cc:66:41 IEEE 802.11: associated (aid 1)
wlan1: AP-STA-CONNECTED 02:00:eb:cc:66:41
wlan1: STA 02:00:eb:cc:66:41 RADIUS: starting accounting session 5CB03FFD-00000000
[7] Setting NAT
Skip setting NAT table
[8] ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.5  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::b331:2b4a:6f0e:3bb2  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:0e:07:04  txqueuelen 1000  (Ethernet)
        RX packets 4135  bytes 289470 (282.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3327  bytes 389471 (380.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 655  bytes 61392 (59.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 655  bytes 61392 (59.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::c39a:42cd:6b13:96d0  prefixlen 64  scopeid 0x20<link>
        ether 02:00:eb:0e:07:04  txqueuelen 1000  (Ethernet)
        RX packets 3  bytes 218 (218.0 B)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 17  bytes 2180 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

HaLow AP ready
```

Figure 3.8 Results of running 1st HaLow Bridge (1/2)

```
[*] Enable 11n interface(wlan0)
[*] Start 11n AP
Configuration file: /home/pi/nrc_pkg/script/conf/11N/hostapd_AP.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:5b:52:51 and ssid "nrc_11n_ap_1"
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
[*] Start dnsmasq service
Done.
pi@raspberrypi:~/nrc_pkg/script $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.5  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::b331:2b4a:6f0e:3bb2  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:0e:07:04  txqueuelen 1000  (Ethernet)
        RX packets 4166  bytes 291166 (284.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3360  bytes 393425 (384.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 655  bytes 61392 (59.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 655  bytes 61392 (59.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.1  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::a618:79b4:c5e5:bfe4  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:5b:52:51  txqueuelen 1000  (Ethernet)
        RX packets 5589  bytes 479110 (467.8 KiB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 1226  bytes 182458 (178.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.1  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::c39a:42cd:6b13:96d0  prefixlen 64  scopeid 0x20<link>
        ether 02:00:eb:0e:07:04  txqueuelen 1000  (Ethernet)
        RX packets 30  bytes 4016 (3.9 KiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 60  bytes 9082 (8.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.9 Results of running 1st HaLow Bridge (2/2)

If users want to implement the network topology in Figure 1.5, they should run 2nd HaLow Bridge where HaLow **STA** and Wi-Fi 4 AP are enabled. Figure 3.10 and Figure 3.11 shows these procedures. While running another HaLow Bridge, HaLow STA tries to connect HaLow AP already run by 1st Halow Bridge. Moreover, another Wi-Fi 4 BSS (SSID: "nrc_11n_ap_2) is ready. It means that two Wi-Fi 4 BSSs are ready and waiting for each Wi-Fi 4 STA's connection.

```
pi@raspberrypi:~/nrc_pkg/script $ ./start.py 0 0 US
------------------------------
Model            : 7292
STA Type         : STA
Security Mode    : OPEN
Country Selected : US
Download FW      : uni_s1g.bin
Interface        : wlan1
TX Gain          : 17
RX Gain          : 85
------------------------------
11n Interface    : wlan0
11n Role         : AP
------------------------------
NRC STA setting for HaLow...
[0] Clear
wireshark-gtk: no process found
11n inteface (wlan0) down
[1] Copy
total 428
drwxr-xr-x 2 pi pi   4096 Apr 11 20:19 .
drwxr-xr-x 4 pi pi   4096 Apr 11 20:19 ..
-rwxr-xr-x 1 pi pi    218 Apr 11 20:19 copy
-rwxr-xr-x 1 pi pi 212560 Apr 11 20:19 nrc7292_cspi.bin
-rwxr-xr-x 1 pi pi 212560 Apr 12 17:15 uni_s1g.bin
-rwxr-xr-x 1 root root 212560 Apr 12 17:15 /lib/firmware/uni_s1g.bin
copy dhcpcd_sta_ap.conf & dnsmasq_sta_ap.conf
[2] Loading module
sudo insmod ~/nrc_pkg/sw/driver/nrc.ko fw_name=uni_s1g.bin disable_cqm=0 hifspeed=16000000
[3] Set trx gain
phy rxgain 85
success
phy txgain 17
success
[4] Set aggregation number
set maxagg 1 0
success
[5] Set guard interval
set gi long
success
[6] Start wpa_supplicant
Successfully initialized wpa_supplicant
nl80211: Could not set interface 'p2p-dev-wlan1' UP
nl80211: deinit ifname=p2p-dev-wlan1 disabled_11b_rates=0
p2p-dev-wlan1: Failed to initialize driver interface
P2P: Failed to enable P2P Device interface
wlan1: SME: Trying to authenticate with 02:00:eb:0e:07:04 (SSID='halow_dual_ap' freq=5805 MHz)
wlan1: Trying to associate with 02:00:eb:0e:07:04 (SSID='halow_dual_ap' freq=5805 MHz)
wlan1: Associated with 02:00:eb:0e:07:04
wlan1: CTRL-EVENT-CONNECTED - Connection to 02:00:eb:0e:07:04 completed [id=0 id_str=]
wlan1: CTRL-EVENT-REGDOM-CHANGE init=COUNTRY_IE type=COUNTRY alpha2=US
[7] Connect and DHCP
Waiting for IP
ip_address=192.168.200.10
IP assigned. HaLow 11ah STA ready
[8] ifconfig
```

Figure 3.10 Results of running 2nd HaLow Bridge (1/2)

```
[*] Enable 11n interface(wlan0)
[*] Start 11n AP
Configuration file: /home/pi/nrc_pkg/script/conf/11N/hostapd_STA.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:99:33:14 and ssid "nrc_11n_ap_2"
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
[*] Setting NAT
Skip setting NAT table
[*] Start dnsmasq service
Done.
pi@raspberrypi:~/nrc_pkg/script $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.112  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::9ce:257e:8305:3d43  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:cc:66:41  txqueuelen 1000  (Ethernet)
        RX packets 308734  bytes 29924108 (28.5 MiB)
        RX errors 0  dropped 3895  overruns 0  frame 0
        TX packets 1614  bytes 196174 (191.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 439  bytes 37890 (37.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 439  bytes 37890 (37.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.110.1  netmask 255.255.255.0  broadcast 192.168.110.255
        inet6 fe80::320c:75bf:e0ff:4b53  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:99:33:14  txqueuelen 1000  (Ethernet)
        RX packets 888  bytes 60232 (58.8 KiB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 1284  bytes 120112 (117.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.10  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::ef38:11b9:fcb5:e9f6  prefixlen 64  scopeid 0x20<link>
        ether 02:00:eb:cc:66:41  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 4616 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.11 Results of running 2nd HaLow Bridge (2/2)

Our SW package for HaLow Bridge makes network configuration including IP and routing information like Figure 3.12 as default.

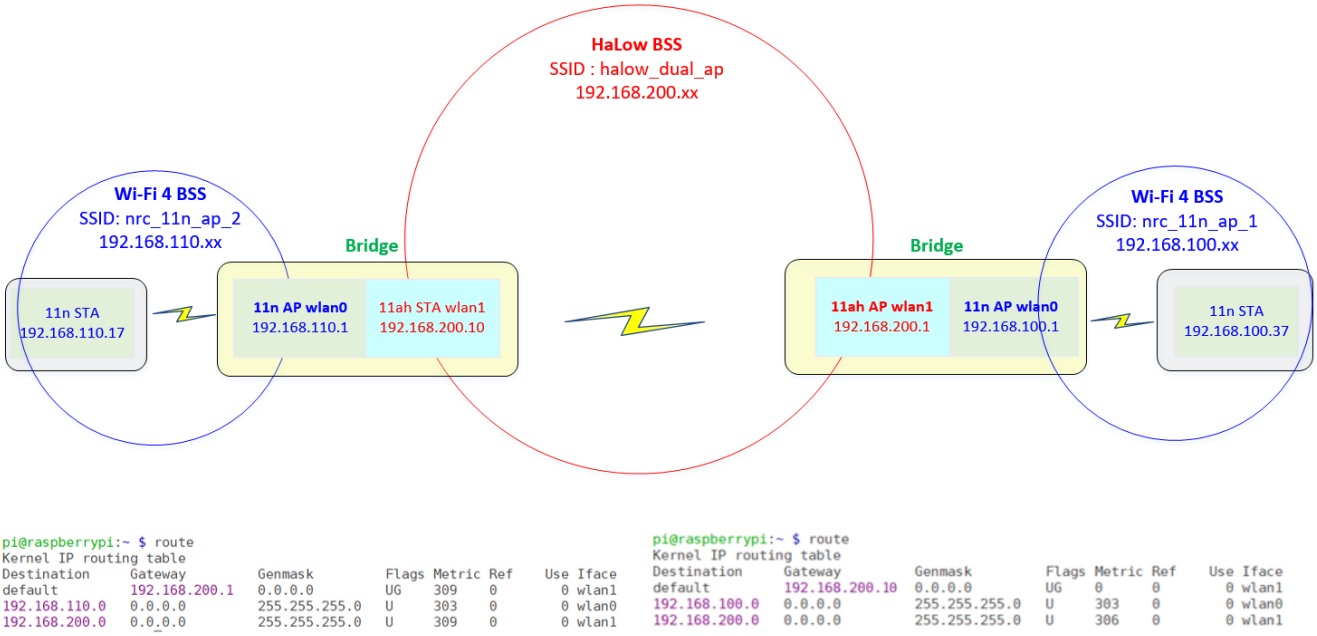

Figure 3.12 Network Topology with two HaLow Bridges

After starting two HaLow Bridges, Wi-Fi 4 STA in BSS1 can communicate to another Wi-Fi 4 STA in BSS2 via HaLow Backhaul like Figure 3.12. Ping and iperf test results are shown in Figure 3.13 and Figure 3.14 respectively.

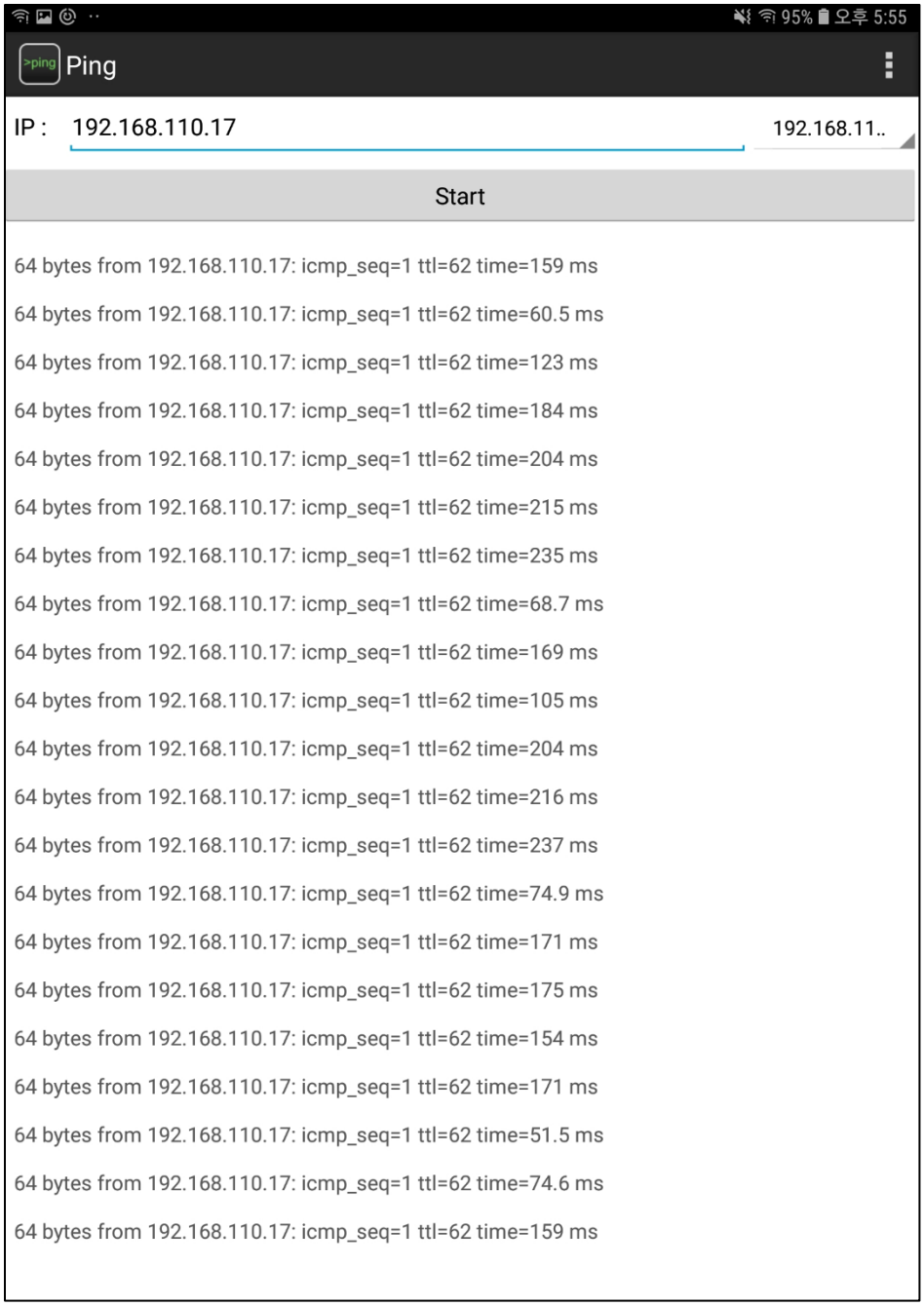

Figure 3.13 Ping Test Results

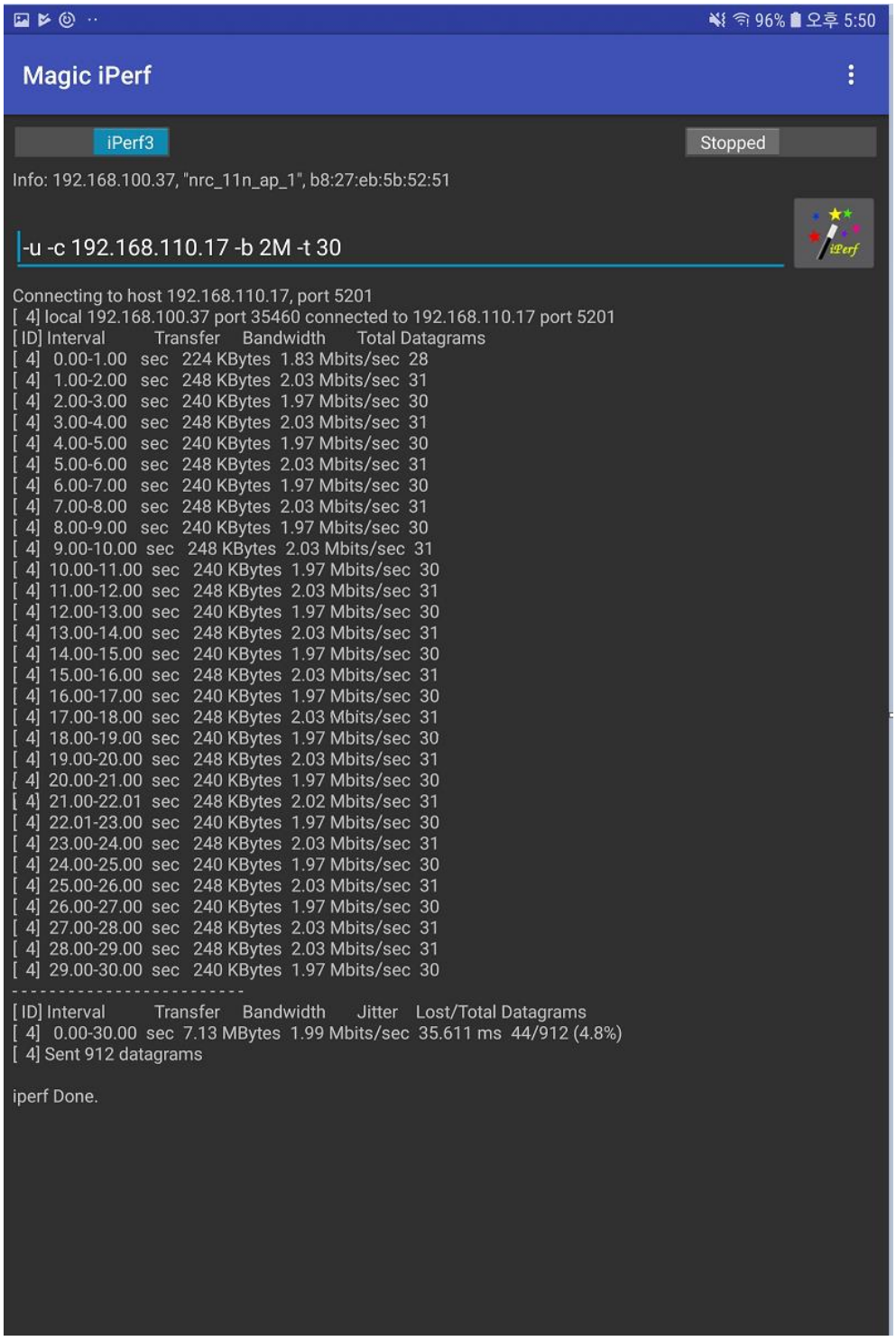

Figure 3.14 iperf3 Test Results

4 Revision History

Revision No	Date	Comments
Ver 1.0	04/12/2019	Initial version for customer release created