AN1278: Using RS9116N to Measure Wi-Fi Throughput

Version 0.4

10/21/2020

Table of Contents

1 About this Document

This document explains the procedure to measure throughput for RS9116N Module.

2 Introduction

RS9116N n-link Open Source Driver implemented for RS9116N family of modules, it uses netlink sockets.

The document will walk you through the steps with which user can bring up the RS9116N device to get the maximum throughput along with the details about software, Hardware, tools required for this.

3 Prerequisites

3.1 Software

- WLAN driver – RS9116N n-link Open Source Driver for Linux

  Download the driver package from the below link:
  https://www.silabs.com/wireless/wi-fi/rs9116-wi-fi-transceiver-modules

- Measurement Tool – Iperf

3.2 Hardware

1. RS9116-wi-fi-transceiver-modules with USB / SDIO connector
2. Linux PC (kernel versions 2.6.35 and above).

4 Terminology (Acronyms/Abbreviations)

- NL 80211 - NL80211 is the new 802.11 netlink interface public header.
- TCP - Transmission Control protocol.
- UDP - User datagram protocol.
- DHCP - Dynamic Host Configuration protocol.
- Uplink - Data transfer from station to access point
- Downlink- Data transfer from access point to station.

5 Throughput

5.1 Description

Throughput refers to how much data can be transferred from one device to another in a given amount of time. It is used to measure the performance wireless networks.

5.2 Block Diagram

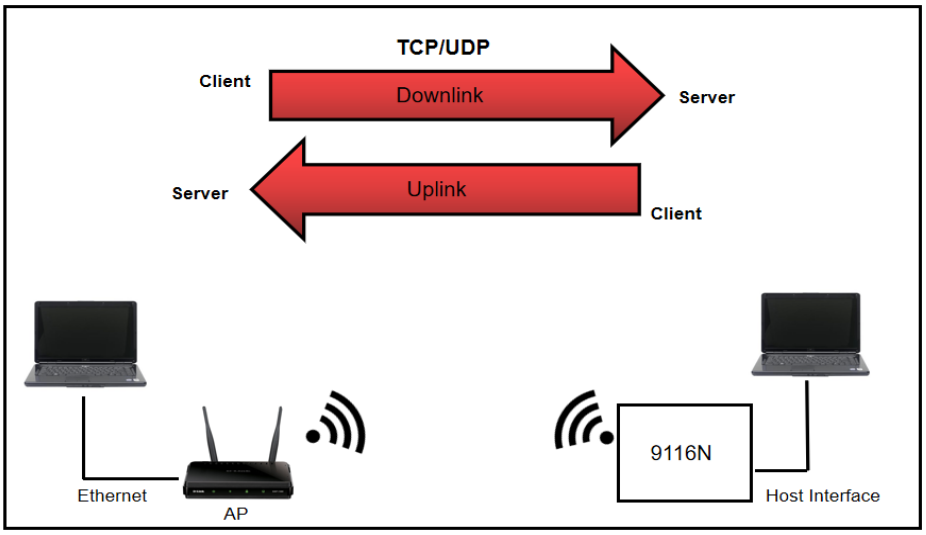


Figure 1: Throughput Setup Diagram

5.3 Use Case

Throughput test will give assurance of the application performance like audio/video data using 9116N module

5.4 Benefits

By measuring throughput, user can determine network performance, which will help in designing end-product.

6 Execution Steps

6.1 Configuration Parameters for Driver Package

1. Download the RS9116N driver package.
   https://www.silabs.com/wireless/wi-fi/rs9116-wi-fi-transceiver-modules
2. Unzip the driver.

```
# unzip RS9116.NB0.NL.GNU.LNX.OSD.x.x.x.x.zip
```

3. Refer the section “Compilation Steps” of RS9116N Open Source Driver Technical Reference Manual which is available under TECH-DOCS section of
   https://www.silabs.com/wireless/wi-fi/rs9116-wi-fi-transceiver-modules.

6.2 Compiling the Driver

Change the working directory to directory of the driver package and following are compilation steps.

1. Configure build flags in driver source.

```
# cd rsi
```

2. Open Makefile and configure build flags. Below are the build flags to be set based on the usage of driver. Selecting the required options shall reduce the binary size which is important for kernel modules particularly on embedded platforms.
   a. **KERNELDIR**: Provide the kernel source path here. For example, on X-86 below path is used.

```
KERNELRELEASE=$(Shell uname -r)
KERNELDIR=/lib/modules/$(KERNELRELEASE)/build
```

   b. **CONFIG_RSI_COEX_MODE**: Enable this flag when Wi-Fi and BT coexistence mode is used.
   c. **CONFIG_RSI_DEBUGFS**: debugfs is used by driver to take dynamic configuration from user. Supported debugfs based configurations are listed in the corresponding feature sections in RS9116N Open Source Driver Technical Reference Manual.
   d. **CONFIG_RSI_BT_ALONE:** Enable this flag when only BT EDR/ BT LE only mode is used.

3. Build the driver using make command.

```
# make
```

After completion of compilation, the driver generates the following modules in the rsi folder according to the configuration.
They are outlined below:

- rsi_91x.ko
- rsi_usb.ko
- rsi_sdio.ko

6.3 Installation of Driver

In order to install the driver, use the following commands:
1.) Before installing driver install the dependencies using below commands:

```
# modprobe mac80211
# modprobe cfg80211
# modprobe Bluetooth
```

Insert rsi_91x.ko with the required module params (configuration) as shown below:

```
# insmod rsi_91x.ko dev_oper_mode=<mode> rsi_zone_enabled=<val> . . .
```

Select dev_oper_mode 1 for STA/AP modes. For all other supported modes please refer RS9116N Open Source Driver Technical Reference Manual.

2.) For USB interface, enter the below command:

```
# insmod rsi_usb.ko
```

3.) For SDIO interface, enter the below command:

```
# insmod rsi_sdio.ko sdio_clock=<clk_val>
```

Here the clk_val is 1 to 50 in Mhz.
You can install either USB or SDIO or both depending upon the selection of the interface while compiling the driver.
After successful installation, a new wireless interface shall be created or WLAN and/or BT/BLE as per the dev_oper_mode selection.
If **WLAN** is selected, interface details can be verified using below commands.
Name of the Wi-Fi interface created after successful installation of the driver can be seen using 'ifconfig' command.

```
# ifconfig -a
```

You should expect an output like the sample shown below with all other available interfaces included.

```
wlan0  flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet6 fe80::8da:1aff:fe1e:d1c8  prefixlen 64 scopeid 0x20<link>
       ether 88:da:1a:1e:d1:c8  txqueuelen 1000  (Ethernet)
       RX packets: 3 bytes 372 (372.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets: 6 bytes 696 (696.0 B)
       TX errors 0 dropped 0 overruns 0 collisions:0
```

6.4 Testing WiFi STA Mode

This section provides steps to configure Wi-Fi station mode using wpa_supplicant.

1. Before running supplicant, stop the existing network manager and unblock WLAN from rfkill. Below commands are used to stop the network-manager on different Linux distribution.

   For ubuntu, we need to use below command.

```
# service network-manager stop
```

For fedora, we need to use below command.

```
# service NetworkManager stop
```

To stop rfkill blocking WLAN, we need to use below command.

```
# rfkill unblock wlan (or) #rfkill unblock all
```

For station mode connectivity, ensure that the dev_oper_mode is set equal to 1.

- dev_oper_mode=1

2. Create a **sta_settings.conf** file with below information. Also please fill the information like ssid, psk etc corresponding to the AP you intend to connect in this file. Sample sta_settings.conf file is available within scripts directory of release package with basic configurations required. User may use this file and edit the information as explained below. For the details of all configurations available please refer open source supplicant wpa_supplicant.conf file.

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
```

Also add network block to the sta_settings.conf file as per the AP security. Example network block for different security modes are listed below.

i. For Open (non-Secure) mode:

```
network={
        ssid="<SSID of Access Point>"
        key_mgmt=NONE
        priority=3
}
```

ii. For WPA2-PSK (CCMP) mode:

```
network={
         ssid="<SSID of Access Point>"
         key_mgmt=WPA-PSK
         psk=<passphrase specified in the Access Point>
         proto=WPA2
         pairwise=CCMP
         group=CCMP
}
```

The pass phrase can be input either in ASCII or Hexadecimal formats:
ASCII Format: psk="very secret passphrase"
Hexadecimal Format: psk= 06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c7

iii. For WPA3 security mode

To connect in WPA3, we need to compile the latest supplicant with below flags enabled in wpa_supplicant .config file

```
CONFIG_SAE=y
CONFIG_IEEE80211W=y
```

```
pmf=2
network={
        ssid="<SSID of Access Point>"
        key_mgmt=SAE
        psk=<passphrase specified in the Access Point>
        ieee80211w=2
}
```

Note: WPA3 Enterprise security mode is not supported in this release.

iv. For WPA2-EAP TLS (Enterprise mode) mode:

```
  network={
        ssid="<SSID of Access Point>"
        key_mgmt=WPA-EAP
        eap=TLS
        anonymous_identity="tlsuser"
        identity="test"
        password=<passphrase specified in the Access Point>
        ca_cert="/etc/certs/wifiuser.pem"
        client_cert="/etc/certs/wifiuser.pem"
        private_key_passwd=<private key password>
        private_key="/etc/certs/wifiuser.key"
        pairwise=CCMP TKIP
        group=CCMP TKIP
        proto=WPA2 WPA
        priority=20
  }
```

In EAP-TLS user has to copy client certificates in a path and the path need to be configured in network block as given above.

v. For WPA2-EAP PEAP (Enterprise mode) mode:

```
  network={
         ssid="<SSID of Access Point>"
         key_mgmt=WPA-EAP
         eap=PEAP
         anonymous_identity="peapuser"
         identity="test"
         password=<passphrase specified in the Access Point>
         pairwise=CCMP TKIP
         group=CCMP TKIP
         proto=WPA2 WPA
         priority=20
  }
```

vi. For WPA2-EAP TTLS (Enterprise mode) mode:

```
network={
        ssid="<SSID of Access Point>"
        key_mgmt=WPA-EAP
        eap=TTLS
        anonymous_identity="ttlsuser"
        identity="test"
        password=<passphrase specified in the Access Point>
        pairwise=CCMP TKIP
        group=CCMP TKIP
        proto=WPA2 WPA
        priority=20
}
```

To connect to an Access Point whose SSID is not broadcast (Hidden), add the following line to the network block.

```
  scan_ssid=1
```

For example**:**

```
  network={
          ssid="<SSID of Access Point>"
          scan_ssid=1
          key_mgmt=NONE
   }
```

3. Start the supplicant using below command:

```
  # wpa_supplicant -i <interface_name> -D nl80211 -c sta_settings.conf -dttt > supp.log &
```

- "–i" option specifies the Wi-Fi interface name
- <interface name> - This name as listed in iw dev output
- "-D" specifies the driver interface to be used. In open source driver it is nl80211.
- "-c" specifies the supplicant configuration file.
- "-d" specifies the log level of supplicant. You can apend more d's to it for more detailed logs.

Note: For configuring supplicant conf for 4.19.97 kernel please refer section " WPA_Supplicant configuration for 4.19.97 kernel on RPI "

1. To check the scan results please use below command.

```
  # wpa_cli -i <interface_name> -p /tmp/wpa scan_results
```

For example, above command will give scan results output as follows.

```
bssid               / frequency   / signal level    / flags                 / ssid
50:d4:f7:1e:5a:40   2457            -21             [WPA2-PSK-CCMP][ESS]    TP_LINK
04:79:70:72:03:e7   2412            -31             [ESS]                   honor_9i
```

2. To check whether the connection is successful or not use below command

```
# iwconfig <interface_name>
```

For example, if connection is successful, we will see below output.

```
wlan0     IEEE 802.11bgn  ESSID:"Range"  Nickname:""
          Mode:Managed  Frequency:2.412 GHz  Access Point: 38:A4:ED:DE:BB:06
          Bit Rate:39 Mb/s   Tx-Power=16 dBm
          Retry short limit:7   RTS thr:2353 B   Fragment thr:2352 B
          Encryption key:off
          Power Management:off
          Link Quality=80/80  Signal level=-28 dBm  Noise level:0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

If the connection is successful, then the connected Access point SSID along with the MAC address is displayed as shown above. If it is not connected to an Access point, a message **"Not Associated"** is displayed as shown below.

```
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=0 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

3. IP for the device can be set in two ways either get IP dynamically from AP or set static IP. To obtain dynamic IP from AP, use below commands.

```
# dhclient < interface_name > -r
# dhclient < interface_name > -v
```

To set static IP to STA use below command.

```
# ifconfig <interface_name> <IP_address>
```

4. To check whether IP is assigned or not use below command.

```
# ifconfig <interface_name>
```

Output:

```
wlan0:  flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.114  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::224:d7ff:fe56:54dc  prefixlen 64  scopeid 0x20<link>
        ether 00:24:d7:56:54:dc  txqueuelen 1000  (Ethernet)
        RX packets 31160  bytes 31082515 (29.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23356  bytes 3367496 (3.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now user can data transfer tests like ping, iperf … etc

6.5 Access Point Configuration

Parameters should be enabled on AP side is the following:

- 11n feature should be enabled to get High throughput
- QoS feature
- Bandwidth (40 MHz will get better throughput than 20 MHz)

7 Iperf Test for Throughput

Iperf: Iperf was developed as a modern tool for measuring maximum TCP and UDP bandwidth performance. Iperf allows the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss. Iperf can run as a client or as a server according to the arguments passed to it.

Install Iperf using the below mentioned commands:

 # sudo apt-get install iperf (for Ubuntu)

 # sudo yum install iperf (for Fedora)

Or download and install the iperf application(**for Windows**) from https://iperf.fr

To run iperf test, on both the client and server the iperf application should be installed and both should have an IP address.
Here,
IP of RS9116N station is 192.168.0.100
IP of PC connected to access point is 192.168.0.101

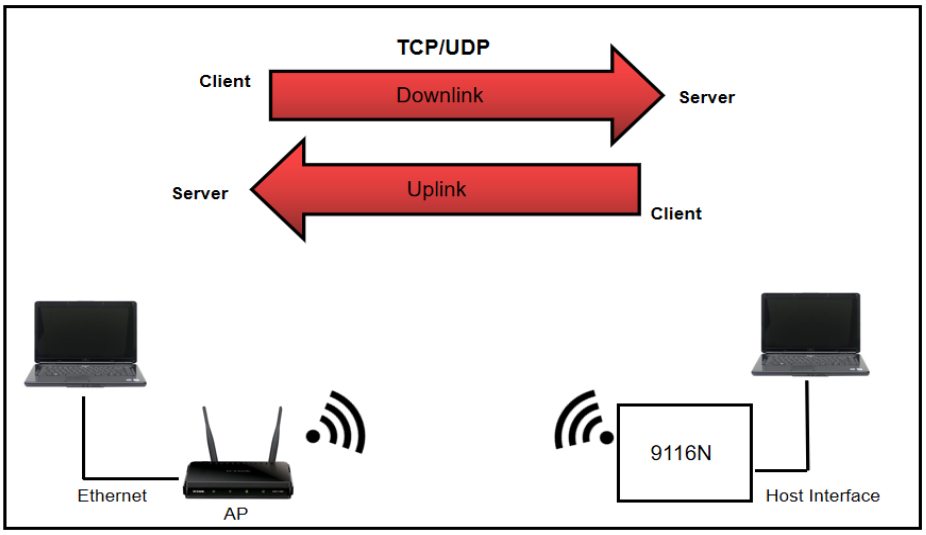


Figure 2: Throughput Setup Diagram

Where RS9116N EVK is configured as station and connected to AP. The station is connected via WLAN. We will measure the upload and download bandwidth, using both TCP and UDP.

7.1 TCP Uplink

At RS9116N station side give below command

 # iperf -c <ip address of server> -i 1 -t 25 (Note:'-t' is for time in seconds to transmit for)

Ex: iperf -c 192.168.0.101 -i 1 -t 25

Where:

The IP address is the server IP address we ran before

-c means run as a client.

-t means run for certain number of seconds (in the example, it is 25 seconds).

-i means the interval between the traffic reports (in the example it is set to 1 second).

At AP side give this command

 # iperf -s -i 1

Where:

-s means run as a server.

-i means the interval between the traffic reports (in the example it is set to 1 second).

Refer to **iperf –help** for more options

Note: Make sure the server is started first.

Iperf output should show as below example:

```
root@arm:/home/ubuntu/release# iperf -c 192.168.0.101 -i 1 -t 25
------------------------------------------------------------
Client connecting to 192.168.0.101, TCP port 5001
TCP window size: 43.8 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.100 port 52692 connected with 192.168.0.101 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  1.0- 2.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  2.0- 3.0 sec  5.75 MBytes  48.2 Mbits/sec
[  3]  3.0- 4.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  4.0- 5.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3]  5.0- 6.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  6.0- 7.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3]  7.0- 8.0 sec  5.62 MBytes  47.2 Mbits/sec
[  3]  8.0- 9.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3]  9.0-10.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 10.0-11.0 sec  5.38 MBytes  45.1 Mbits/sec
[  3] 11.0-12.0 sec  6.12 MBytes  51.4 Mbits/sec
[  3] 12.0-13.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 13.0-14.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 14.0-15.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 15.0-16.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 16.0-17.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 17.0-18.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 18.0-19.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 19.0-20.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 20.0-21.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 21.0-22.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 22.0-23.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 23.0-24.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 24.0-25.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  0.0-25.0 sec   146 MBytes  48.9 Mbits/sec
root@arm:/home/ubuntu/release#
```

Figure 3: Iperf Running on RS9116N

```
root@lapt064:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release# iperf -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.168.0.101 port 5001 connected with 192.168.0.100 port 52692
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 1.0 sec  5.63 MBytes  47.3 Mbits/sec
[  4]  1.0- 2.0 sec  5.74 MBytes  48.2 Mbits/sec
[  4]  2.0- 3.0 sec  5.67 MBytes  47.6 Mbits/sec
[  4]  3.0- 4.0 sec  5.81 MBytes  48.7 Mbits/sec
[  4]  4.0- 5.0 sec  5.77 MBytes  48.4 Mbits/sec
[  4]  5.0- 6.0 sec  5.94 MBytes  49.8 Mbits/sec
[  4]  6.0- 7.0 sec  5.87 MBytes  49.2 Mbits/sec
[  4]  7.0- 8.0 sec  5.68 MBytes  47.6 Mbits/sec
[  4]  8.0- 9.0 sec  5.91 MBytes  49.6 Mbits/sec
[  4]  9.0-10.0 sec  5.74 MBytes  48.2 Mbits/sec
[  4] 10.0-11.0 sec  5.46 MBytes  45.8 Mbits/sec
[  4] 11.0-12.0 sec  5.59 MBytes  46.9 Mbits/sec
[  4] 12.0-13.0 sec  5.90 MBytes  49.5 Mbits/sec
[  4] 13.0-14.0 sec  5.76 MBytes  48.3 Mbits/sec
[  4] 14.0-15.0 sec  5.69 MBytes  47.7 Mbits/sec
[  4] 15.0-16.0 sec  5.82 MBytes  48.8 Mbits/sec
[  4] 16.0-17.0 sec  5.91 MBytes  49.6 Mbits/sec
[  4] 17.0-18.0 sec  5.74 MBytes  48.1 Mbits/sec
[  4] 18.0-19.0 sec  5.49 MBytes  46.0 Mbits/sec
[  4] 19.0-20.0 sec  5.99 MBytes  50.2 Mbits/sec
[  4] 20.0-21.0 sec  5.79 MBytes  48.5 Mbits/sec
[  4] 21.0-22.0 sec  5.98 MBytes  50.1 Mbits/sec
[  4] 22.0-23.0 sec  5.70 MBytes  47.9 Mbits/sec
[  4] 23.0-24.0 sec  6.02 MBytes  50.5 Mbits/sec
[  4] 24.0-25.0 sec  5.91 MBytes  49.6 Mbits/sec
[  4]  0.0-25.2 sec   146 MBytes  48.5 Mbits/sec
```

Figure 4: Iperf running on PC Connected to AP

7.2 TCP Downlink

At RS9116N station side run the below command

 # iperf -s -i 1

At AP side run below command

 # iperf -c <IP address> -i 1 -t 25

Iperf output at RS9116N STA should show as below example:

```
root@arm:/home/ubuntu/release# iperf -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.0.100 port 5001 connected with 192.168.0.101 p
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 1.0 sec  5.39 MBytes  45.2 Mbits/sec
[  4]  1.0- 2.0 sec  5.86 MBytes  49.1 Mbits/sec
[  4]  2.0- 3.0 sec  5.88 MBytes  49.3 Mbits/sec
[  4]  3.0- 4.0 sec  5.87 MBytes  49.2 Mbits/sec
[  4]  4.0- 5.0 sec  5.87 MBytes  49.3 Mbits/sec
[  4]  5.0- 6.0 sec  5.90 MBytes  49.5 Mbits/sec
[  4]  6.0- 7.0 sec  5.83 MBytes  48.9 Mbits/sec
[  4]  7.0- 8.0 sec  5.88 MBytes  49.3 Mbits/sec
[  4]  8.0- 9.0 sec  5.83 MBytes  48.9 Mbits/sec
[  4]  9.0-10.0 sec  5.86 MBytes  49.2 Mbits/sec
[  4] 10.0-11.0 sec  5.88 MBytes  49.3 Mbits/sec
[  4] 11.0-12.0 sec  5.80 MBytes  48.6 Mbits/sec
[  4] 12.0-13.0 sec  5.87 MBytes  49.2 Mbits/sec
[  4] 13.0-14.0 sec  5.84 MBytes  48.9 Mbits/sec
[  4] 14.0-15.0 sec  5.88 MBytes  49.3 Mbits/sec
[  4] 15.0-16.0 sec  5.89 MBytes  49.4 Mbits/sec
[  4] 16.0-17.0 sec  5.81 MBytes  48.7 Mbits/sec
[  4] 17.0-18.0 sec  5.88 MBytes  49.3 Mbits/sec
[  4] 18.0-19.0 sec  5.83 MBytes  48.9 Mbits/sec
[  4] 19.0-20.0 sec  5.91 MBytes  49.5 Mbits/sec
[  4] 20.0-21.0 sec  5.90 MBytes  49.5 Mbits/sec
[  4] 21.0-22.0 sec  5.91 MBytes  49.6 Mbits/sec
[  4] 22.0-23.0 sec  5.40 MBytes  45.3 Mbits/sec
[  4] 23.0-24.0 sec  6.26 MBytes  52.5 Mbits/sec
[  4] 24.0-25.0 sec  5.85 MBytes  49.1 Mbits/sec
[  4]  0.0-25.2 sec   148 MBytes  49.0 Mbits/sec
```

Figure 5: Iperf Results for RS9116N

```
root@lapt864:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release# iperf -c 192.168.0.100 -i 1 -t 25
------------------------------------------------------------
Client connecting to 192.168.0.100, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.101 port 55782 connected with 192.168.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3]  1.0- 2.0 sec  6.38 MBytes  53.5 Mbits/sec
[  3]  2.0- 3.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3]  3.0- 4.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  4.0- 5.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  5.0- 6.0 sec  5.75 MBytes  48.2 Mbits/sec
[  3]  6.0- 7.0 sec  5.75 MBytes  48.2 Mbits/sec
[  3]  7.0- 8.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  8.0- 9.0 sec  5.75 MBytes  48.2 Mbits/sec
[  3]  9.0-10.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 10.0-11.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 11.0-12.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 12.0-13.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 13.0-14.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 14.0-15.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 15.0-16.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 16.0-17.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 17.0-18.0 sec  5.88 MBytes  49.3 Mbits/sec
[  3] 18.0-19.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 19.0-20.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 20.0-21.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3] 21.0-22.0 sec  6.12 MBytes  51.4 Mbits/sec
[  3] 22.0-23.0 sec  5.50 MBytes  46.1 Mbits/sec
[  3] 23.0-24.0 sec  6.12 MBytes  51.4 Mbits/sec
[  3] 24.0-25.0 sec  6.00 MBytes  50.3 Mbits/sec
[  3]  0.0-25.1 sec   148 MBytes  49.4 Mbits/sec
root@lapt864:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release#
```

Figure 6: Iperf Running on PC Connected to AP

7.3 UDP Uplink

At RS9116N station side run below command

 # iperf -c <IP address> -i 1 -u -b 50M -t 25

Ex: iperf -c 192.168.0.101 -i 1 -u 50M -t 25

Where:

The IP address is the server IP address we ran before.

-c means run as a client.

-t means run for certain number of seconds (in the example, it is 25 seconds).

-i means the interval between the traffic reports (in the example it is set to 1 second).

-u means set the client is a UDP server.

-b means the UDP bandwidth.

At AP side give below command

 # iperf -s -u -i 1

-s means run as a server.

-u means set the client is a UDP server.

-i means the interval between the traffic reports (in the example it is set to 1 second).

Refer to **iperf –help** for more options

Iperf output should show as below example

```
root@arm:/home/ubuntu/release# iperf -c 192.168.0.101 -i 1 -t 25 -u -b 50M
------------------------------------------------------------
Client connecting to 192.168.0.101, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  160 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.100 port 38221 connected with 192.168.0.101 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  1.0- 2.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  2.0- 3.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  3.0- 4.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  4.0- 5.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  5.0- 6.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  6.0- 7.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  7.0- 8.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  8.0- 9.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  9.0-10.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 10.0-11.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 11.0-12.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 12.0-13.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 13.0-14.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 14.0-15.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 15.0-16.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 16.0-17.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 17.0-18.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 18.0-19.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 19.0-20.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 20.0-21.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 21.0-22.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 22.0-23.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 23.0-24.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec
[  3] Sent 106383 datagrams
[  3] Server Report:
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec   0.079 ms    0/106382 (0%)
[  3]  0.0-25.0 sec  1 datagrams received out-of-order
root@arm:/home/ubuntu/release#
```

Figure 7: Iperf Results for RS9116N

```
root@lapt064:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release# iperf -s -i 1 -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.101 port 5001 connected with 192.168.0.100 port 38221
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  5.97 MBytes  50.1 Mbits/sec   0.079 ms    0/ 4257 (0%)
[  3]  1.0- 2.0 sec  5.96 MBytes  50.0 Mbits/sec   0.081 ms    0/ 4254 (0%)
[  3]  2.0- 3.0 sec  5.97 MBytes  50.1 Mbits/sec   0.090 ms    0/ 4256 (0%)
[  3]  3.0- 4.0 sec  5.96 MBytes  50.0 Mbits/sec   0.092 ms    0/ 4254 (0%)
[  3]  4.0- 5.0 sec  5.97 MBytes  50.0 Mbits/sec   0.093 ms    0/ 4255 (0%)
[  3]  5.0- 6.0 sec  5.97 MBytes  50.1 Mbits/sec   0.088 ms    0/ 4256 (0%)
[  3]  6.0- 7.0 sec  5.97 MBytes  50.0 Mbits/sec   0.083 ms    0/ 4255 (0%)
[  3]  7.0- 8.0 sec  5.97 MBytes  50.1 Mbits/sec   0.091 ms    0/ 4256 (0%)
[  3]  8.0- 9.0 sec  5.97 MBytes  50.0 Mbits/sec   0.088 ms    0/ 4255 (0%)
[  3]  9.0-10.0 sec  5.97 MBytes  50.0 Mbits/sec   0.090 ms    0/ 4255 (0%)
[  3] 10.0-11.0 sec  5.97 MBytes  50.0 Mbits/sec   0.100 ms    0/ 4255 (0%)
[  3] 11.0-12.0 sec  5.97 MBytes  50.0 Mbits/sec   0.088 ms    0/ 4255 (0%)
[  3] 12.0-13.0 sec  5.97 MBytes  50.1 Mbits/sec   0.078 ms    0/ 4256 (0%)
[  3] 13.0-14.0 sec  5.97 MBytes  50.0 Mbits/sec   0.091 ms    0/ 4255 (0%)
[  3] 14.0-15.0 sec  5.97 MBytes  50.0 Mbits/sec   0.087 ms    0/ 4255 (0%)
[  3] 15.0-16.0 sec  5.97 MBytes  50.0 Mbits/sec   0.088 ms    0/ 4255 (0%)
[  3] 16.0-17.0 sec  5.96 MBytes  50.0 Mbits/sec   0.120 ms    0/ 4253 (0%)
[  3] 17.0-18.0 sec  5.97 MBytes  50.0 Mbits/sec   0.116 ms    0/ 4255 (0%)
[  3] 18.0-19.0 sec  5.97 MBytes  50.1 Mbits/sec   0.079 ms    0/ 4258 (0%)
[  3] 19.0-20.0 sec  5.97 MBytes  50.0 Mbits/sec   0.091 ms    0/ 4255 (0%)
[  3] 20.0-21.0 sec  5.97 MBytes  50.0 Mbits/sec   0.088 ms    0/ 4255 (0%)
[  3] 21.0-22.0 sec  5.96 MBytes  50.0 Mbits/sec   0.110 ms    0/ 4252 (0%)
[  3] 22.0-23.0 sec  5.97 MBytes  50.1 Mbits/sec   0.083 ms    0/ 4258 (0%)
[  3] 23.0-24.0 sec  5.97 MBytes  50.1 Mbits/sec   0.099 ms    0/ 4256 (0%)
[  3] 24.0-25.0 sec  5.97 MBytes  50.0 Mbits/sec   0.079 ms    0/ 4255 (0%)
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec   0.079 ms    0/106382 (0%)
[  3]  0.0-25.0 sec  1 datagrams received out-of-order
```

Figure 8: Iperf Running on PC Connected to AP

7.4 UDP Downlink

At RS9116N station side give below command

 # iperf -s -i -u 1

At AP side give below command

 # iperf -c <IP address> -i 1 -u -b 50M -t 25

```
^Croot@arm:/home/ubuntu/release# iperf -s -i 1 -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  160 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.100 port 5001 connected with 192.168.0.101 port 58767
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  5.99 MBytes  50.3 Mbits/sec   0.116 ms    0/ 4275 (0%)
[  3]  1.0- 2.0 sec  5.95 MBytes  49.9 Mbits/sec   0.108 ms    0/ 4246 (0%)
[  3]  2.0- 3.0 sec  5.97 MBytes  50.1 Mbits/sec   0.115 ms    0/ 4256 (0%)
[  3]  3.0- 4.0 sec  5.97 MBytes  50.0 Mbits/sec   0.122 ms    0/ 4255 (0%)
[  3]  4.0- 5.0 sec  5.97 MBytes  50.1 Mbits/sec   0.103 ms    0/ 4256 (0%)
[  3]  5.0- 6.0 sec  5.97 MBytes  50.0 Mbits/sec   0.086 ms    0/ 4255 (0%)
[  3]  6.0- 7.0 sec  5.97 MBytes  50.0 Mbits/sec   0.127 ms    0/ 4255 (0%)
[  3]  7.0- 8.0 sec  5.97 MBytes  50.1 Mbits/sec   0.104 ms    0/ 4256 (0%)
[  3]  8.0- 9.0 sec  5.97 MBytes  50.0 Mbits/sec   0.112 ms    0/ 4255 (0%)
[  3]  9.0-10.0 sec  5.97 MBytes  50.1 Mbits/sec   0.110 ms    0/ 4256 (0%)
[  3] 10.0-11.0 sec  5.97 MBytes  50.0 Mbits/sec   0.124 ms    0/ 4255 (0%)
[  3] 11.0-12.0 sec  5.97 MBytes  50.0 Mbits/sec   0.085 ms    0/ 4255 (0%)
[  3] 12.0-13.0 sec  5.97 MBytes  50.1 Mbits/sec   0.113 ms    0/ 4256 (0%)
[  3] 13.0-14.0 sec  5.96 MBytes  50.0 Mbits/sec   0.113 ms    0/ 4254 (0%)
[  3] 14.0-15.0 sec  5.96 MBytes  50.0 Mbits/sec   0.105 ms    0/ 4250 (0%)
[  3] 15.0-16.0 sec  5.97 MBytes  50.1 Mbits/sec   0.100 ms    0/ 4261 (0%)
[  3] 16.0-17.0 sec  5.97 MBytes  50.1 Mbits/sec   0.113 ms    0/ 4256 (0%)
[  3] 17.0-18.0 sec  5.97 MBytes  50.0 Mbits/sec   0.115 ms    0/ 4255 (0%)
[  3] 18.0-19.0 sec  5.97 MBytes  50.1 Mbits/sec   0.118 ms    0/ 4256 (0%)
[  3] 19.0-20.0 sec  5.97 MBytes  50.0 Mbits/sec   0.095 ms    0/ 4255 (0%)
[  3] 20.0-21.0 sec  5.97 MBytes  50.1 Mbits/sec   0.105 ms    0/ 4256 (0%)
[  3] 21.0-22.0 sec  5.97 MBytes  50.1 Mbits/sec   0.095 ms    0/ 4256 (0%)
[  3] 22.0-23.0 sec  5.96 MBytes  50.0 Mbits/sec   0.091 ms    0/ 4254 (0%)
[  3] 23.0-24.0 sec  5.97 MBytes  50.0 Mbits/sec   0.118 ms    0/ 4255 (0%)
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec   0.112 ms    0/106373 (0%)
[  3]  0.0-25.0 sec  1 datagrams received out-of-order
```

Figure 9: Iperf Running on RS9116N

```
DHCPOFFER of 192.168.0.101 from 192.168.0.1
DHCPACK of 192.168.0.101 from 192.168.0.1
RTNETLINK answers: File exists
bound to 192.168.0.101 -- renewal in 2948 seconds.
root@lapt064:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release# iperf -c 192.168.0.100 -i 1 -t 25 -u -b 50M
------------------------------------------------------------
Client connecting to 192.168.0.100, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.101 port 58767 connected with 192.168.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  1.0- 2.0 sec  5.95 MBytes  49.9 Mbits/sec
[  3]  2.0- 3.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  3.0- 4.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  4.0- 5.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  5.0- 6.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  6.0- 7.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  7.0- 8.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  8.0- 9.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  9.0-10.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 10.0-11.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 11.0-12.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 12.0-13.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 13.0-14.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 14.0-15.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 15.0-16.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 16.0-17.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 17.0-18.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 18.0-19.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 19.0-20.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 20.0-21.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 21.0-22.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3] 22.0-23.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 23.0-24.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3] 24.0-25.0 sec  5.97 MBytes  50.1 Mbits/sec
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec
[  3] Sent 106374 datagrams
[  3] Server Report:
[  3]  0.0-25.0 sec   149 MBytes  50.0 Mbits/sec   0.111 ms    0/106373 (0%)
[  3]  0.0-25.0 sec  1 datagrams received out-of-order
root@lapt064:/work/test/murali/RS9113.NBZ.NL.GENR.LNX.1.6.3/source/host/release#
```

Figure 10: Iperf Running on PC Connected to AP

8 Expected Results

Application data throughput up to 100 Mbps (Hosted Mode) in 802.11n with 40MHz bandwidth and up to 50 Mbps with 20 MHz bandwidth.

SNo	Operating mode	Actual Operation Mode (This depend on User test case)	Band (GHz)	Chnl Width	Protocol			
					TCP		UDP	
					Uplink	Downlink	Uplink	Downlink
1	Wi-Fi_STA Only	STA mode	2.4	20	45.4	41.5	50	48.3
2	Wi-Fi_AP Only	AP mode	2.4	20	26.3	39.1	45.5	46.6
3	STA+AP	STA mode	2.4	20	43.7	40.7	50	49
4		AP mode	2.4	20	42.6	32.8	47.2	44.7
5	Wi-Fi_STA + BT	STA mode + BT classic (Connected)	2.4	20	33.6	20.6	24.8	35.4
6		STA mode + BT classic (Connected)	2.4	20	41.2	40.2	48.7	45.3
7	Wi-Fi_AP + BT	AP mode + BT classic (Connected)	2.4	20	40.5	40.8	45.9	46
8		AP mode + BT classic (Connected)	2.4	20	23.1	24.7	43.5	34.8
9	Wi-Fi_STA + BT LE	STA mode + BT LE (Advertising mode)	2.4	20	32.9	28.6	50	34
10	Wi-Fi_AP + BT LE	AP mode + BT LE (Advertising mode)	2.4	20	24.7	21.9	42.4	34.8
11	Wi-Fi_STA + BT_Classic + BT LE	STA mode + BT LE (Advertising mode)	2.4	20	27.8	27.1	50	34.6
12	Wi-Fi_AP + BT_Classic +BT LE	AP mode + BT LE (Advertising mode)	2.4	20	28.9	24.3	35.7	28.8
13	WiFi STA Only	STA mode	5	40	84	78	99	98
14	Wi-Fi_STA + BT LE	STA mode + BT LE (Advertising mode)	5	40	71	58	71	86
15	Wi-Fi_STA + BT LE	Station mode + BT LE (L2 test data transfer)	5	40	39	32	49	43
16	WiFi AP Only	AP mode	5	40	73	74	76	84
17	Wi-Fi_AP + BT LE	AP mode + BT LE (Advertising mode)	5	40	56	56	74	69

Table 1: Expected Throughput Results

Note:

1. The above mentioned throughput numbers are verified using USB interface on X86. The numbers may vary slightly with SDIO interface.
2. Wireless throughput varies with the environment of the test setup - distance, obstacles, type of obstacles, speed and performance of the platform.
3. The throughput is also dependant on whether all the Wi-Fi components in the test support 40 MHz and 20MHz or only 20 MHz bandwidth.
4. It is recommended to choose a Wi-Fi channel which has less interference preferably in the 5Ghz band if the EVK is for a dual band, to observe optimal throughput of the module.
5. For operating _mode, please refer to the below table for operating mode configuration.

S.NO	Protocols Support	Operating Mode
1	Wi-Fi_alone_STA	1
2	Wi-Fi_alone_AP	1
3	BT_Classic	4
4	Wi-Fi_STA + BT_Classic	5
5	Wi-Fi_AP + BT_Classic	6
6	BT LE	8
7	BT_Classic+BT LE	12
8	Wi-Fi_STA + BT_Classic + BT LE	13
9	Wi-Fi_AP + BT_Classic +BT LE	14

Table 2: Operating Mode Configuration

9 Summary

By following the above procedure, RS9116N STA can connect to AP and can calculate the maximum throughput by running iperf.

10 References and Related Documentation

- Refer to our RS9116N Open Source Driver Technical Reference Manual, Data sheet and Module integration guide which are available in our Technical Resource Search site:

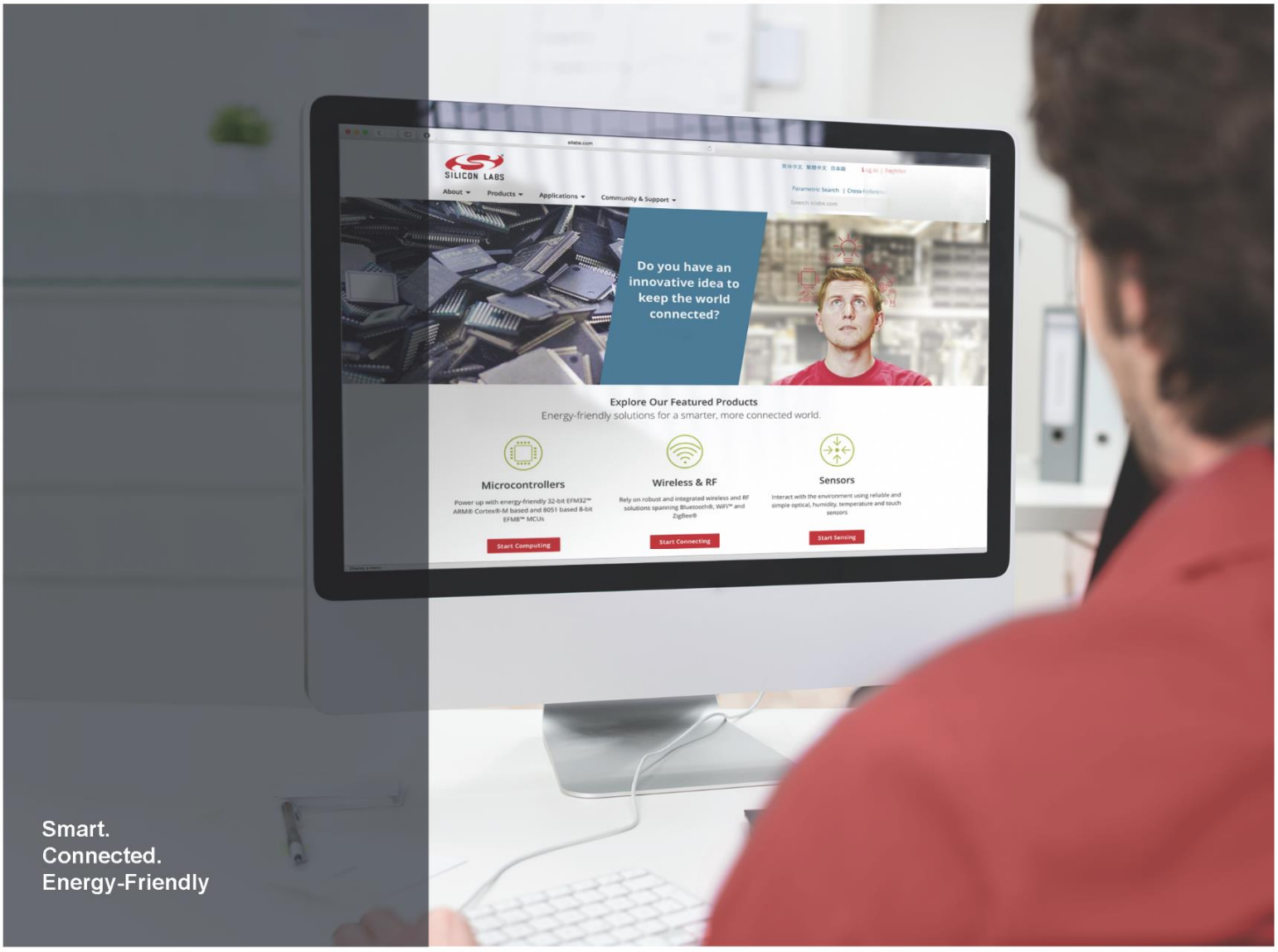
  https://www.silabs.com/wireless/wi-fi/rs9116-wi-fi-transceiver-modules

11 Troubleshooting

1. Make sure that IP address is assigned for both client and server before running iperf test.
2. Make sure that station is connected to AP ,by giving the command "iwconfig".
3. For running Iperf, server should listen first, and client must send the data.

12 Revision History

Revision No.	Version No	Date	Changes
1	0.1	April, 2020	Preliminary version
2	0.2	April 23, 2020	Added Figure captions Modified screen shots Changed URLs
3	0.3	May 13, 2020	Updated iperf URLs Updated screen shots
4	0.4	October 16, 2020	Updated Links Updated code blocks Updated OSD related changes



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com