

SDX55 and IPQ8074 rmnet loopback function user guide

一、 SDX55 Loopback function introduction

SDX55 support rmnet loopback function, which means the host sends data to SDX55 through the rmnet network card and SDX55 can loop back the data to the host. You also could define loop multiples. For example, if you define the multiples is 7, the result is that if the host sends 10M data to SDX55, SDX55 loops back 70M data to the host.

二、 Preparation before testing

2.1 The test tool version

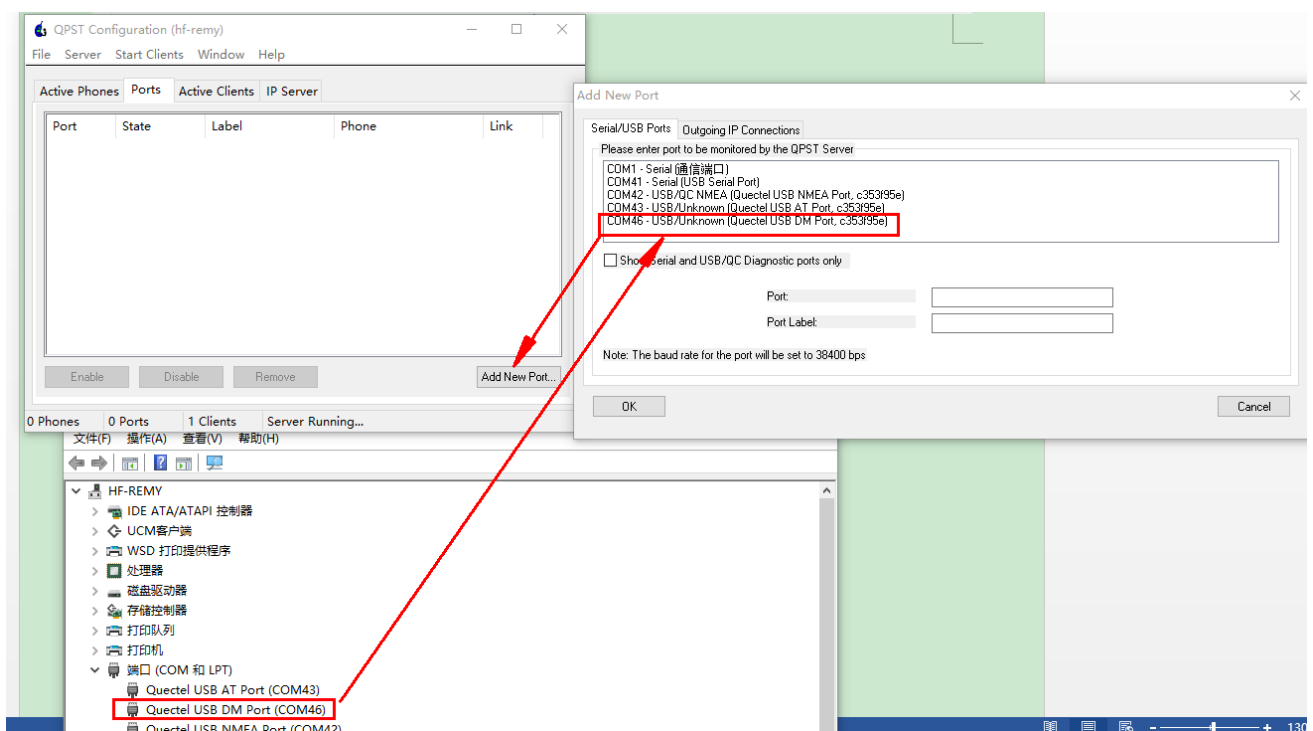
You should use below or newer version tool than below to make the test:

- qmi_wwan_q version: Quectel_Linux&Android_QMI_WWAN_Driver_V1.2.0.4
- quectel-CM version: Quectel_QConnectManager_Linux_V1.6.0.5
- PCIE driver version: Quectel_Linux_PCIE_MHI_Driver_V1.3.0.6
- X55 firmware version: RM500QGLAAR03A01M4G_BETA_20200107F
- IPQ8074 version: ipq8074-ilq-11-0-1/r00050.1

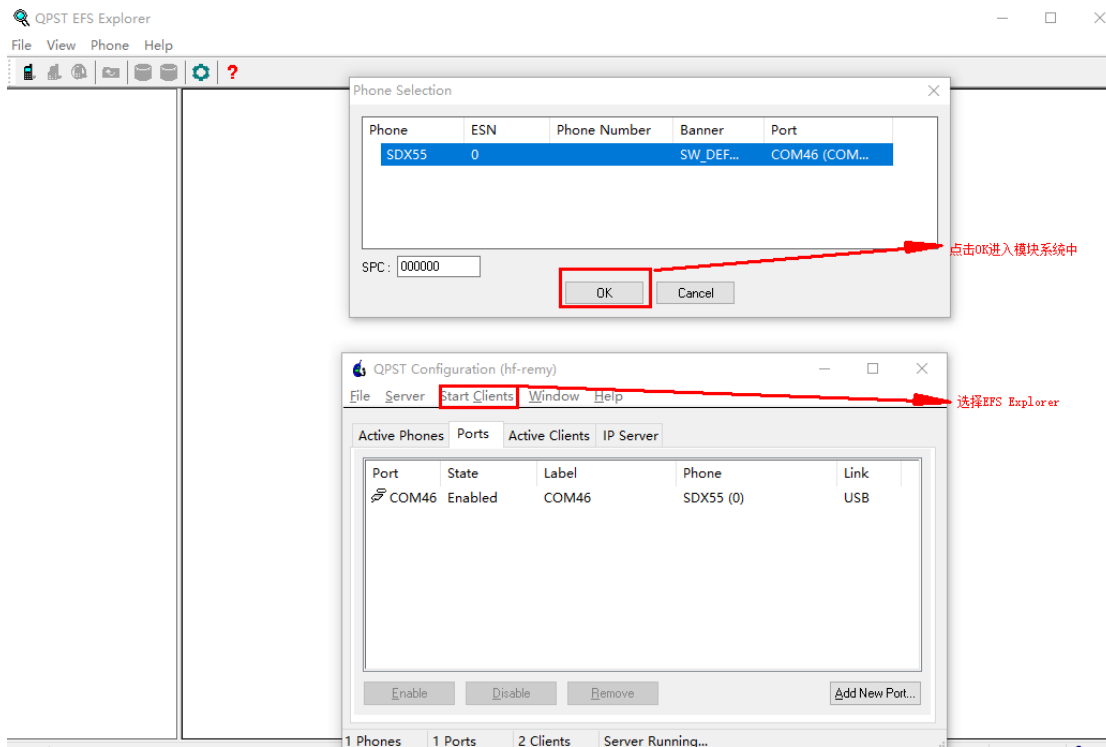
2.2 Enable the loopback function

In order to enable the loopback function, you need to set below NV value with QPST tool.

First: follow below step to connect QPST:



Second: Start Clients > EFS Explorer > OK get into the module file system.



Third: Create the file **loopback_config.txt** at below path:

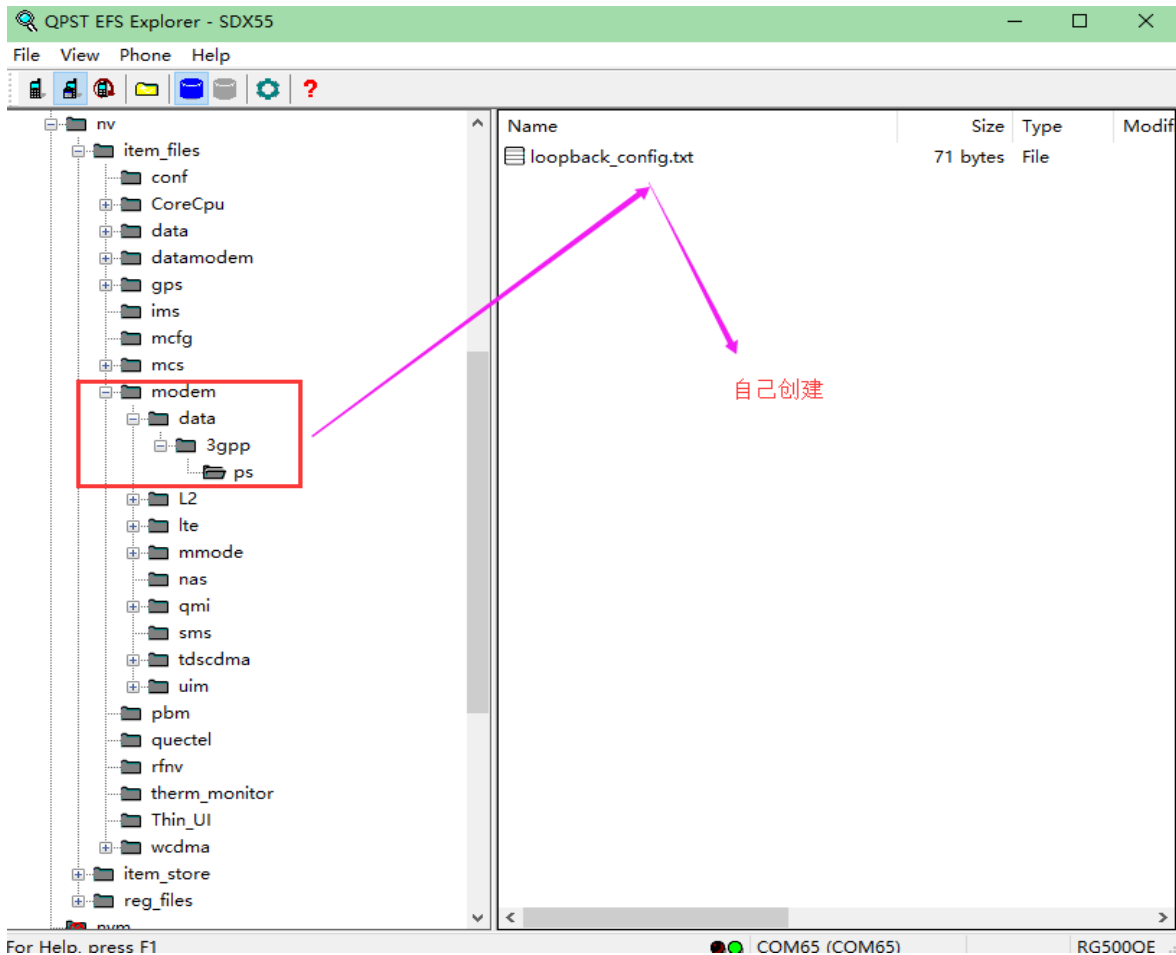
/nv/item_files/modem/data/3gpp/ps/loopback_config.txt

Note: This /data/3gpp/ps/ need to create by yourself.

Input below information into the loopback_config.txt document.

```
LOOPBACK_ENABLED:1;  
L2_LOOPBACK_ENABLED:0;  
SYS_MODE:9;  
REP_FACTOR:14;
```

REP_FACTOR is the multiples for loopback, you could select 7 or 14.



2.3 Dial up with Quectel-CM

quectel-CM -l 14

-l means the multiples for loopback.

Note: The real multiples for loopback is decided by the NV value. Here is just define Quectel-CM is under loopback test.

Test log is as below:

The driver will generate two network card as below:

For Qmi_wwan_q: rmnet_usb0 and rmnet_usb0.1, rmnet_usb0.1 is used for dial up.

For Pcie_mhi: rmnet_mhi0 and rmnet_mhi0.1, rmnet_mhi0.1 is used for dial up.

The reference dial log was as below:

```
[ 2113.488982] rmnet_nss_init(): initializing rmnet_nss
[ 2113.502858] qmi_wwan_q 4-1:1.4: cdc-wdm0: USB WDM device
[2113.502886] qmi_wwan_q 4-1:1.4: Quectel
EC25&EC21&EG91&EG95&EG06&EP06&EM06&EG12&EP12&EM12&EG16&EG18&BG96&AG35 work on RawIP
mode
[ 2113.507950] qmi_wwan_q 4-1:1.4: rx_urb_size = 31744
[ 2113.519883] qmi_wwan_q 4-1:1.4 rmnet_usb0: register 'qmi_wwan_q' at usb-xhci-hcd.1.auto-1, RMNET/USB device,
fa:9a:e4:de:55:ab
```

[2113.524647] net rmnet_usb0: qmap_register_device(rmnet_usb0.1)=0

[2113.535856] net rmnet_usb0 rmnet_usb0.1: NSS context created

[2113.546753] net rmnet_usb0: qmap_register_device rmnet_usb0.1

2. CM dial up LOG:

[04-02_04:17:05:647] Quectel_QConnectManager_Linux_V1.6.0.2

[04-02_04:17:05:647] Find /sys/bus/usb/devices/4-1 idVendor=0x2c7c idProduct=0x800, bus=0x004, dev=0x002

[04-02_04:17:05:648] Auto find qmichannel = /dev/cdc-wdm0

[04-02_04:17:05:648] Auto find usbnet_adapter = rmnet_usb0

[04-02_04:17:05:648] netcard driver = qmi_wwan_q, driver version = V1.2.0.4

[04-02_04:17:05:648] qmap_mode = 1, qmap_version = 9, qmap_size = 31744, muxid = 0x81, qmap_netcard = rmnet_usb0.1

[04-02_04:17:05:648] Modem works in QMI mode

[04-02_04:17:05:656] cdc_wdm_fd = 7

[04-02_04:17:06:656] QmiThreadSendQMIMessage pthread_cond_timeout_np timeout

[04-02_04:17:07:814] Get clientWDS = 19

[04-02_04:17:07:846] Get clientDMS = 1

[04-02_04:17:07:878] Get clientNAS = 3

[04-02_04:17:07:910] Get clientUIM = 1

[04-02_04:17:07:942] Get clientWDA = 1

[04-02_04:17:07:973] requestBaseBandVersion RM500QGLAAR03A01M4G_BETA_20200107F 1 [Dec 30 2019 17:00:00]

[04-02_04:17:08:005] qmap_settings.rx_urb_size = 31744

[04-02_04:17:08:006] qmap_settings.ul_data_aggregation_max_datagrams = 16

[04-02_04:17:08:006] qmap_settings.ul_data_aggregation_max_size = 4096

[04-02_04:17:08:006] qmap_settings.dl_minimum_padding = 16

[04-02_04:17:08:101] requestSetLoopBackState(loopback_state=1, replication_factor=14)

[04-02_04:17:08:197] requestGetSIMStatus SIMStatus: SIM_ABSENT

[04-02_04:17:08:229] requestGetProfile[1] ///0

[04-02_04:17:08:261] requestRegistrationState2 MCC: 0, MNC: 0, PS: Detached, DataCap: UNKNOWN

[04-02_04:17:08:293] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED

[04-02_04:17:08:293] ifconfig rmnet_usb0.1 down

[04-02_04:17:08:298] ifconfig rmnet_usb0.1 0.0.0.0

[04-02_04:17:08:302] SetLoopBackInd: loopback_state=1, replication_factor=14

[04-02_04:17:08:325] requestSetupDataCall WdsConnectionIPv4Handle: 0x670ecf00

[2116.378340] net rmnet_usb0: link_state 0x0 -> 0x1

[04-02_04:17:08:453] ifconfig rmnet_usb0 up

[04-02_04:17:08:458] ifconfig rmnet_usb0.1 up

[04-02_04:17:08:465] you are use OpenWrt?

[04-02_04:17:08:466] should not calling udhcpc manually?

[04-02_04:17:08:466] should modify /etc/config/network as below?

[04-02_04:17:08:466] config interface wan

[04-02_04:17:08:466] option ifname rmnet_usb0.1

[04-02_04:17:08:466] option proto dhcp

[04-02_04:17:08:466] should use "/sbin/ifstatus wan" to check rmnet_usb0.1 's status?

[04-02_04:17:08:466] busybox udhcpc -f -n -q -t 5 -i rmnet_usb0.1

udhcpc: started, v1.28.3

udhcpc: sending discover

udhcpc: sending select for 192.168.48.181

udhcp: lease of 192.168.48.181 obtained, lease time 7200

[04-02_04:17:08:642] udhcp: ifconfig rmnet_usb0.1 192.168.48.181 netmask 255.255.255.252 broadcast +

[04-02_04:17:08:651] udhcp: setting default routers: 192.168.48.182

三、 Testing process and results

3.1 CPU load balancing adjustment

USB/PCIE interrupts run on CPU0 by default

First: Transfer the load of the two network CARDS to CPU1 and CPU2 respectively

For Qmi_wwan_q driver:

```
echo 2 > /sys/class/net/rmnet_usb0/queues/rx-0/rps_cpus
```

```
echo 4 > /sys/class/net/rmnet_usb0.1/queues/rx-0/rps_cpus
```

For pcie_mhi driver:

```
echo 2 > /sys/class/net/rmnet_mhi0/queues/rx-0/rps_cpus
```

```
echo 4 > /sys/class/net/rmnet_mhi0.1/queues/rx-0/rps_cpus
```

Second: Increasing the size of the network card to be processed SKB queue prevents the network card driver from receiving SKB too large for the system to process in a short time.

```
echo 2000 > /proc/sys/net/core/netdev_max_backlog
```

Third: If you use IPQ8074, please make below configuration:

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

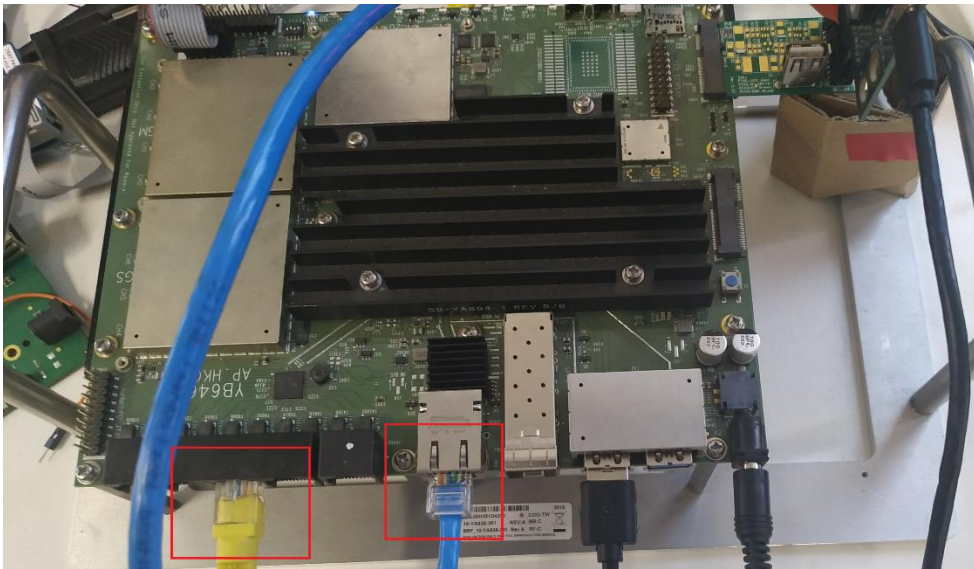
```
echo 8 > /proc/irq/100/smp_affinity
```

```
echo 8 > /proc/irq/144/smp_affinity
```

3.2 Set up the test environment

1. Prepare 2 PCS, one PC has 2.5g /10G network card, one PC has 1G network card is enough. For the connection, please refer to below picture.

The blue cable is the 10G adapter socket of IPQ, and the yellow cable is the 1G adapter socket.



2. The router sets `rmnet_usb0.1/ rmnet_mhi0.1` as WAN, and sets the port to forward UDP 5001 to the PC with blue network cable (with 2.5g /10G network card).
3. The PC with the blue cable (with a 2.5g /10G nic) installs a speed monitor tool to monitor the network speed.

For windows, you could use TrafficMonitor.

For Ubuntu, you could use nload.

4. Run the following iperf command on the PC with the yellow cable (with a 1G network card) and upload the data to SDX55.

```
iperf -u -c 192.168.48.172 -b 180M -t 72000
```

5. When measuring speed, you need to monitor whether the CPU load of IPQ is balanced

```
root@OpenWrt:~# mpstat -P ALL 2
```

```
Linux 4.4.60 (OpenWrt) 04/02/20 _armv7l_ (4 CPU)
```

04:18:59	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
04:19:01	all	0.59	0.00	0.00	0.00	1.90	13.03	0.00	0.00	0.00	84.48
04:19:01	0	0.00	0.00	0.00	0.00	7.58	10.10	0.00	0.00	0.00	82.32
04:19:01	1	0.00	0.00	0.00	0.00	0.50	6.03	0.00	0.00	0.00	93.47
04:19:01	2	0.00	0.00	0.00	0.00	0.00	31.33	0.00	0.00	0.00	68.67
04:19:01	3	2.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	97.47

6. Test result

The test results of the PC connected with the blue network cable (with a 2.5g /10G network card)

pcie_mhi test result:

Time	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
04:11:50											
04:11:52	all	0.00	0.00	0.38	0.00	0.63	32.25	0.00	0.00	0.00	66.75
04:11:52	0	0.00	0.00	1.02	0.00	2.04	40.82	0.00	0.00	0.00	56.12
04:11:52	1	0.00	0.00	0.00	0.00	0.00	40.69	0.00	0.00	0.00	59.31
04:11:52	2	0.00	0.00	0.00	0.00	0.00	45.23	0.00	0.00	0.00	54.77
04:11:52	3	0.00	0.00	0.51					0.00	0.00	96.97

上传: 925.3 Kb/s 下载: 2.31 Gb/s

Qmi_wwan_q test result:

Time	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
04:26:22											
04:26:24	all	0.51	0.00	0.89	0.00	4.94	30.29	0.00	0.00	0.00	63.37
04:26:24	0	0.00	0.00	1.52	0.00	19.70	39.39	0.00	0.00	0.00	39.39
04:26:24	1	1.01	0.00	1.51	0.00	0.00	35.18	0.00	0.00	0.00	62.31
04:26:24	2	0.00	0.00	0.00	0.00	0.00	45.96	0.00	0.00	0.00	54.04
04:26:24	3	1.03	0.00	0.52					0.00	0.00	98.45

上传: 922.6 Kb/s 下载: 2.32 Gb/s

If the module is directly connected to the Ubuntu PC and the loopback test results:

```

root@dtw-ThinkPad-E480: /home/dtw/Desktop/Kawhi/Test/LoopBack/X55_Q...
File Edit View Search Terminal Help
root@dtw-ThinkPad-E480:~# iperf -u -c 192.168.48.172 -b 250M -l1 -t 70
0000 -p 5001
Client connecting to 192.168.48.172, UDP port 5001
Sending 1470 byte datagrams, IPG target: 44.86 us (kalman adjust)
UDP buffer size: 208 KByte (default)
[ 3] local 192.168.48.171 port 44925 connected with 192.168.48.172 p
ort 5001
[ 3] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec   31.3 MBytes   262 Mb/s/sec
[ 3] 1.0- 2.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 2.0- 3.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 3.0- 4.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 4.0- 5.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 5.0- 6.0 sec   31.3 MBytes   262 Mb/s/sec
[ 3] 6.0- 7.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 7.0- 8.0 sec   31.3 MBytes   262 Mb/s/sec
[ 3] 8.0- 9.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 9.0-10.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 10.0-11.0 sec   31.3 MBytes   262 Mb/s/sec
[ 3] 11.0-12.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 12.0-13.0 sec   31.2 MBytes   262 Mb/s/sec
[ 3] 13.0-14.0 sec   31.2 MBytes   262 Mb/s/sec

root@dtw-ThinkPad-E480: /home/dtw/Desktop/Kawhi/Test/LoopBack/X55_QMAP_...
File Edit View Search Terminal Help
device rrmnet_usb0.1 [192.168.48.171] (4/4):
incoming:
#####
#####
#####
#####
##### Curr: 3.30 GBit/s
##### Avg: 2.06 GBit/s
##### Min: 0.00 Bit/s
##### Max: 3.30 GBit/s
##### Ttl: 293.88 GByte
outgoing:
#####
#####
#####
#####
##### Curr: 257.16 MB/s
##### Avg: 160.37 MB/s
##### Min: 0.00 Bit/s
##### Max: 257.17 MB/s
##### Ttl: 47.07 GByte

```