

保密等级	审核人	批准人	实施时间
绝密文件 <input type="checkbox"/> 机密文件 <input type="checkbox"/> 一般文件 <input checked="" type="checkbox"/>	周涵	蔡珺	2017-07-26

SDX55 系列

Loopback 功能测试指导 手册

5G NR 系列

版本：SDX55 系列_Loopback 功能测试指导手册_V1.0

日期：2020-03-25



上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期 (B 区) 5/13 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2020，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2020.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2020-03-25	张德乐	初始版本
1.1	2020-5-19	张书尧	修改 USB&PCIE 模式下的 loopback 拨号

目录

文档历史	2
目录	3
1 引言	5
2 准备环境.....	6
3 QPST 设置回环倍数.....	6
4 修改源文件	错误!未定义书签。
4.1 Quectel-CM 拨号工具宏修改	错误!未定义书签。
4.2 USB 下使能 qmi_wwan_q.c 中的宏	错误!未定义书签。
4.3 PCIE 下使能 mhi_netdev_quectel.c 中的宏	11
5.测试步骤	11
5.1 quectel-CM 工具拨号	11
5.2 ping 192.168.48.172 是否可 ping 通	错误!未定义书签。
5.3 iperf 测试	13
6.常见问题	15
6.1 IPQ 上需要对导入的 pcie 驱动进行源码修改	15
6.2 IPQ 上测试 loopback 功能 iperf 打流需要指定网卡和端口.....	17

1 引言

本文档主要引导大家如何进行 Loopback 功能的测试，及模块 NV 中回环倍数的设置方法
X55 的回环功能简单介绍：

X55 自持 rmnet 网卡的 loopback，即上位机通过 rmnet 网卡发送数据给 X55，X55 可以回环给上位机。
而且可以定义回环倍数，假设是 7 倍，则效果就是发 10M 的数据给 X55，X55 回环 70M 的数据给你。

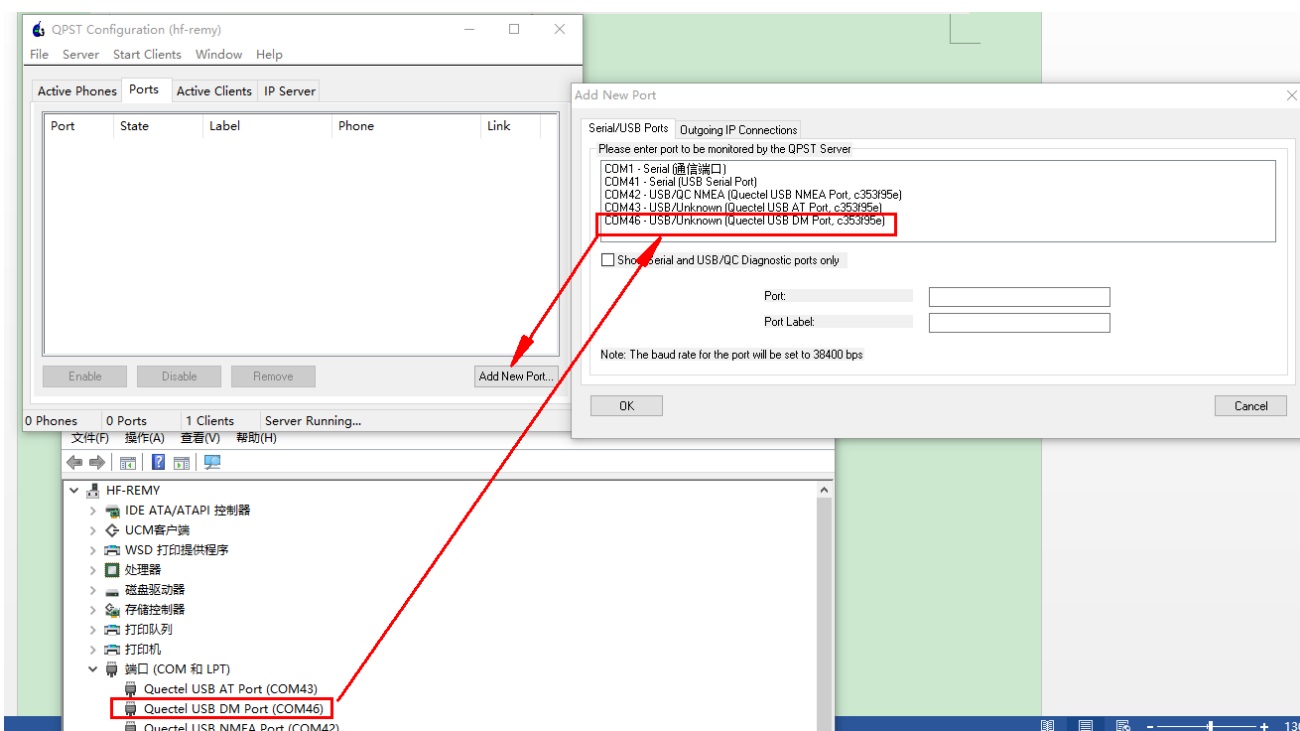
●

2 准备环境

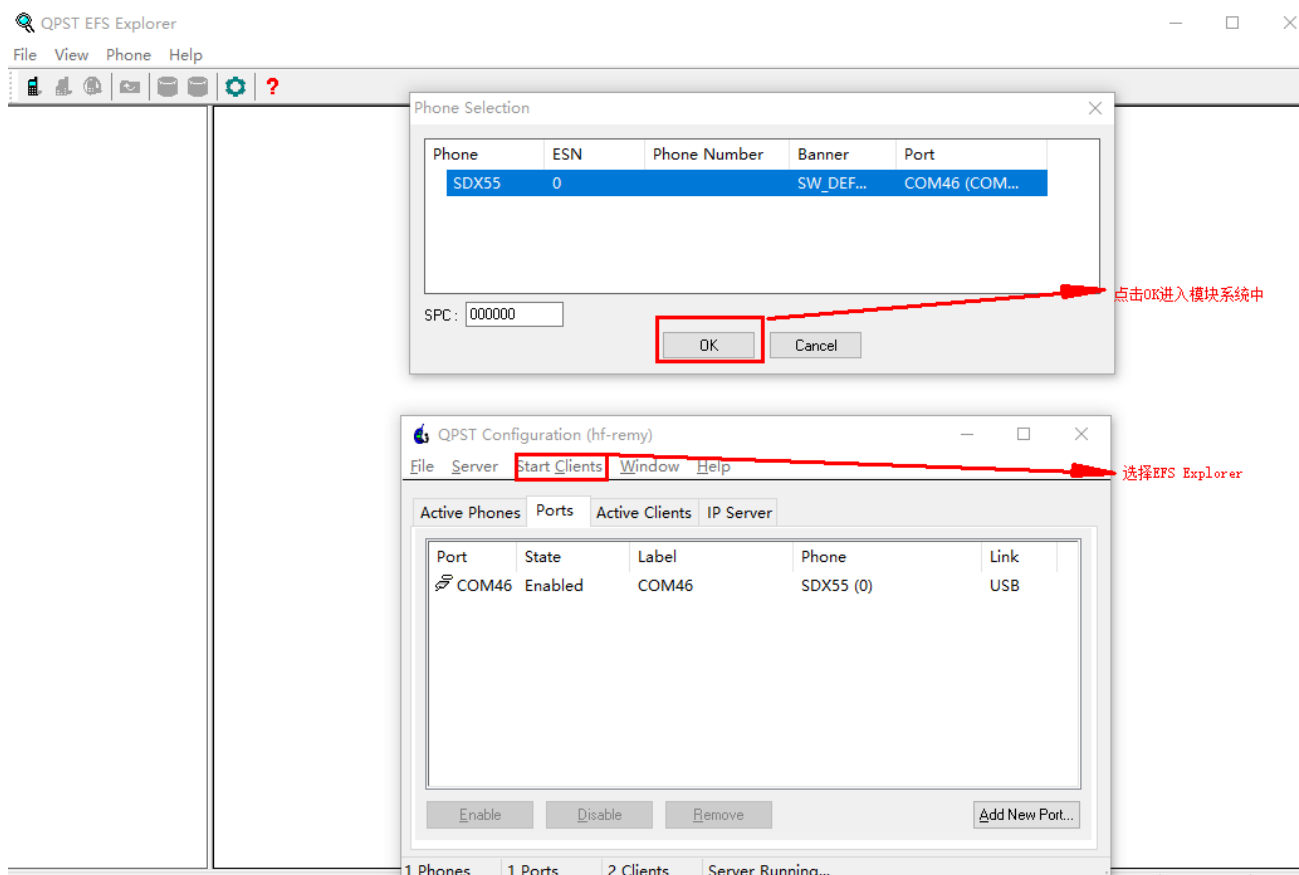
- Loopback 功能验证区分两种环境
 1. Linux 下 loopback 功能验证
 2. IPQ 下 loopback 功能验证（IPQ 下验证 loopback 功能需要在 PC 端进行 iperf 打流操作）
 3. 物资准备：双系统 PC、5GEVB、待测模块、网线（loopback 功能验证不需要 SIM 卡）

3 QPST 设置回环倍数

1. 选择 DM 口连接 QPST



2. Start Clients > EFS Explorer > OK 进入模块文件系统



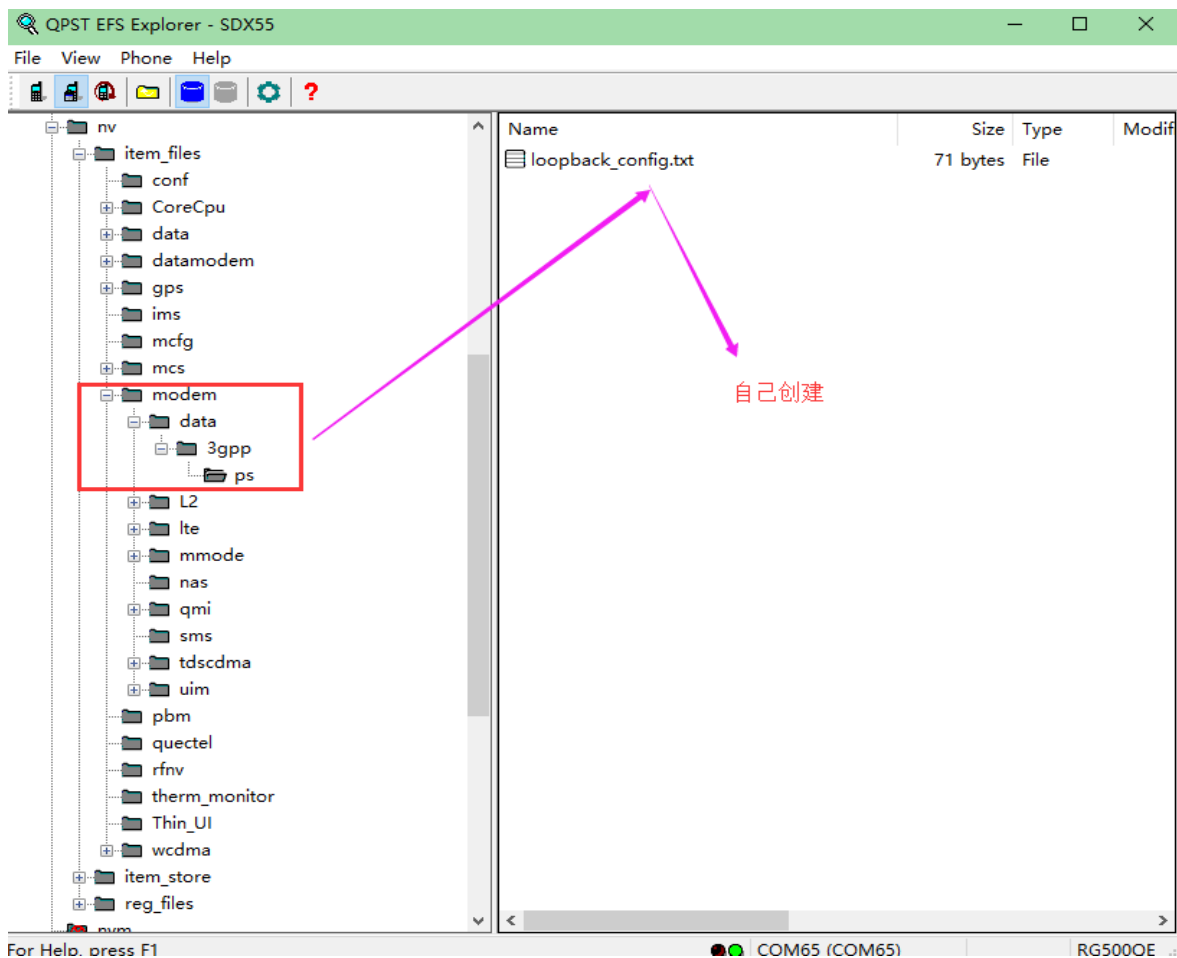
3. 在如下文件夹下面穿件文件/nv/item_files/modem/data/3gpp/ps/loopback_config.txt

(此处/data/3gpp/ps/文件夹需要自己创建)

loopback_config.txt 文档内容如下:

```
LOOPBACK_ENABLED:1;
L2_LOOPBACK_ENABLED:0;
SYS MODE:9;
REP_FACTOR:7;
```

REP_FACTOR:7 是回环倍数(可选择性设置 14 倍回环和 7 倍回环)



•

4 USB&PCIE 下的 loopback 功能

4.1 usb 下的 loopback 拨号

4.1.1 准备工作

- 使用最新的 LINUX 的 USB 驱动: Quectel_LTE&5G_Windows_USB_Driver_V2.2.2
- 使用最新的 loopback 拨号工具: Quectel_QConnectManager_Linux_V1.6.0.6 (开发提供)
- 验证版本: RG500QEAAAR01A01M4G_BETA_20200106G

备注: Linux 的 USB 驱动安装见文档 **SDx55 PCIE 测试指导**

4.1.2 查看 usb 驱动有没有加载成功

- ls /dev/ttyUSB*

```
root@q-OptiPlex-5060: /home/q
q@q-OptiPlex-5060:~$ sudo su
sudo: /etc/sudoers.d is world writable
[sudo] password for q:
root@q-OptiPlex-5060:/home/q# ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3
root@q-OptiPlex-5060:/home/q#
```

4.1.3 loopback 拨号

- 在 Quectel_QConnectManager_Linux_V1.6.0.6 驱动根目录执行 make clean→make→chmod 777 *
- 执行拨号工具 ./quectel-CM -l 14 (14 为回环倍数)

```

root@q-OptiPlex-5060: /home/q/jimmy/Quectel_QConnectManager_Linux_V1.6.0.6/Quec
1 [Dec 30 2019 02:00:00]
[05-20_00:33:45:734] qmap_settings.rx_urb_size = 31744
[05-20_00:33:45:830] requestSetLoopBackState(loopback_state=1, replication_factor=14)
[05-20_00:33:45:926] requestGetSIMStatus SIMStatus: SIM_ABSENT
[05-20_00:33:45:958] requestGetProfile[1] ///0
[05-20_00:33:45:990] requestRegistrationState2 MCC: 0, MNC: 0, PS: Detached, DataCap: UNKNOWN
[05-20_00:33:46:022] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[05-20_00:33:46:022] ifconfig rmnet_usb0.1 down
[05-20_00:33:46:027] ifconfig rmnet_usb0.1 0.0.0.0
SIOCSIFFLAGS: Network is down
[05-20_00:33:46:032] SetLoopBackInd: loopback_state=1, replication_factor=14
[05-20_00:33:46:054] requestSetupDataCall WdsConnectionIPv4Handle: 0x3d48d4e0
[05-20_00:33:46:182] ifconfig rmnet_usb0 up
[05-20_00:33:46:187] ifconfig rmnet_usb0.1 up
[05-20_00:33:46:188] busybox udhcpd -f -n -q -t 5 -i rmnet_usb0.1
[05-20_00:33:46:189] udhcpd (v1.22.1) started
[05-20_00:33:46:232] Sending discover...
[05-20_00:33:46:268] Sending select for 192.168.48.171...
[05-20_00:33:46:307] Lease of 192.168.48.171 obtained, lease time 7200
[05-20_00:33:46:314] /etc/udhcpd/default.script: Resetting default routes
SIOCDELRT: No such process

```

- 拨号成功后，ip route show 查看路由表信息，并 ping 192.168.48.172 默认网关地址

```

root@q-OptiPlex-5060: /home/q
q@q-OptiPlex-5060:~$ sudo su
sudo: /etc/sudoers.d is world writable
[sudo] password for q:
root@q-OptiPlex-5060:/home/q# ip route show
default via 192.168.48.172 dev rmnet_usb0.1
192.168.48.168/29 dev rmnet_usb0.1 proto kernel scope link src 192.168.48.171

root@q-OptiPlex-5060:/home/q# ping 192.168.48.172
PING 192.168.48.172 (192.168.48.172) 56(84) bytes of data.
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=2.36 ms
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.39 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.41 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.39 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.42 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.39 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.42 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=8.39 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=2.41 ms
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.48 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.52 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.48 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.52 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.48 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=8.53 ms (DUP!)

```

备注：loopback 功能验证不需要插 SIM 卡，拨号成功后，会获取一个固定的 IP 地址：192.168.48.171

4.2 PCIE 下 loopback 拨号

4.2.1 准备工作

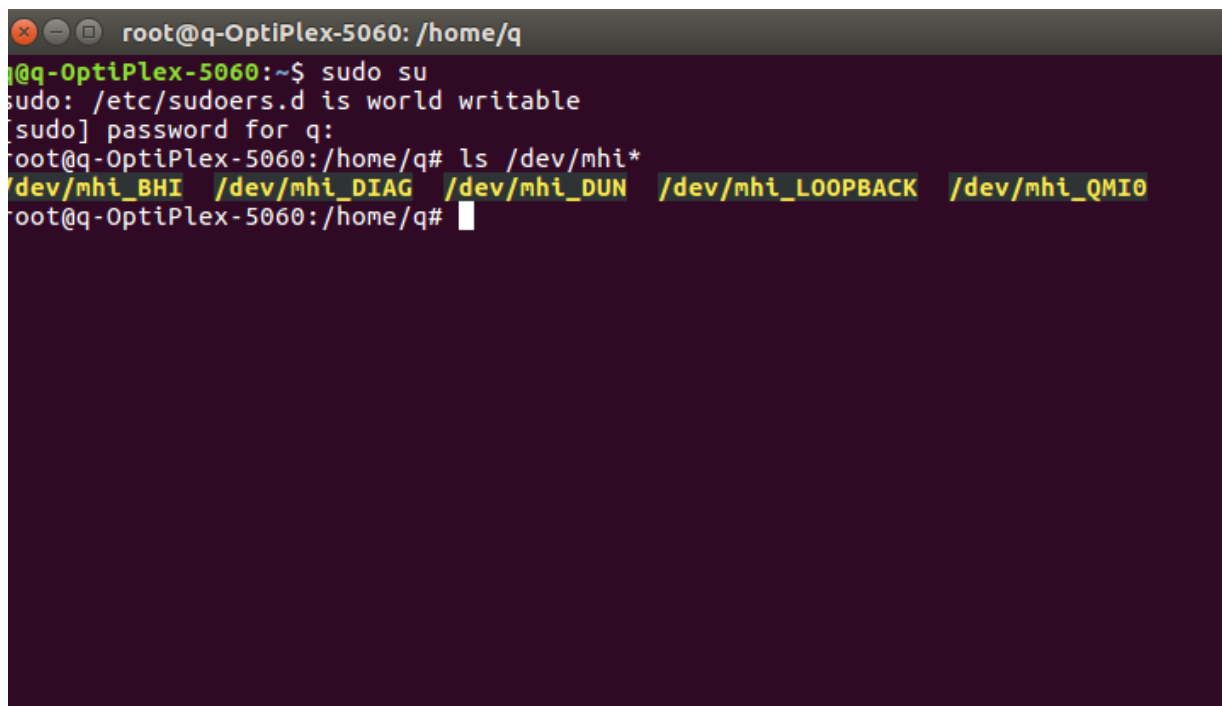
- 使用最新的 loopback 对应的 PCIE 驱动：Quectel_Linux_PCIE_MHI_Driver_V1.3.0.10（开发提供）
- 使用最新的 loopback 拨号工具：Quectel_QConnectManager_Linux_V1.6.0.6（开发提供）
- 验证版本：RG500QEAAAR01A01M4G_BETA_20200106G

4.2.2 PCIE 驱动安装

- 在 Quectel_Linux_PCIE_MHI_Driver_V1.3.0.10 驱动文件夹根目录执行 `make clean→make→chmod 777 *`
- 编译完成生成 `pcie_mhi.ko` 的可执行驱动工具，将此文件 `copy` 至 `/lib/modules/`uname -r`/kernel/drivers/PCI` 路径下，删除此路径下原有的驱动文件 `rm -rf pcie_mhi.ko`，进行授权 `chmod 777 pcie_mhi.ko`，再回到驱动文件夹下执行 `make install` 安装驱动，插入 PCIE 设备后，开机 `ls /dev/mhi*` 即可查询到 PCIE 口（或者在驱动文件夹下手动安装 `insmod pcie_mhi.ko`）

4.2.3 查看 loopback 对应的 PCIE 驱动有没有安装成功

- `ls /dev/mhi*`



```

root@q-OptiPlex-5060: /home/q
q@q-OptiPlex-5060:~$ sudo su
sudo: /etc/sudoers.d is world writable
[sudo] password for q:
root@q-OptiPlex-5060:/home/q# ls /dev/mhi*
/dev/mhi_BHI  /dev/mhi_DIAG  /dev/mhi_DUN  /dev/mhi_LOOPBACK  /dev/mhi_QMI0
root@q-OptiPlex-5060:/home/q#
  
```

4.2.4 loopback 拨号

- 在 Quectel_QConnectManager_Linux_V1.6.0.6 驱动根目录执行 make clean→make→chmod 777 *
- 执行拨号工具 ./quectel-CM -l 14 (14 为回环倍数)

```

root@q-OptiPlex-5060: /home/q/jimmy/Quectel_QConnectManager_Linux_V1.6.0.6/Quec
[Dec 30 2019 02:00:00]
05-19_19:10:38:803] qmap_settings.rx_urb_size = 16384
05-19_19:10:38:811] requestSetLoopBackState(loopback_state=1, replication_facto
=14)
05-19_19:10:38:816] requestGetSIMStatus SIMStatus: SIM_READY
05-19_19:10:38:819] requestGetProfile[1] ///0
05-19_19:10:38:822] requestRegistrationState2 MCC: 0, MNC: 0, PS: Detached, Dat
Cap: UNKNOWN
05-19_19:10:38:825] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
05-19_19:10:38:825] ifconfig rmnet_mhi0.1 down
05-19_19:10:38:830] ifconfig rmnet_mhi0.1 0.0.0.0
IOCSIFFLAGS: Network is down
05-19_19:10:38:834] SetLoopBackInd: loopback_state=1, replication_factor=14
05-19_19:10:38:846] requestSetupDataCall WdsConnectionIPv4Handle: 0x5b48a190
05-19_19:10:38:852] ifconfig rmnet_mhi0 up
05-19_19:10:38:858] ifconfig rmnet_mhi0.1 up
05-19_19:10:38:868] busybox udhcpc -f -n -q -t 5 -i rmnet_mhi0.1
05-19_19:10:38:927] udhcpc (v1.22.1) started
05-19_19:10:38:984] Sending discover...
05-19_19:10:39:027] Sending select for 192.168.48.171...
05-19_19:10:39:079] Lease of 192.168.48.171 obtained, lease time 7200
05-19_19:10:39:085] /etc/udhcpc/default.script: Resetting default routes
IOCDLRT: No such process

```

- 拨号成功后, ip route show 查看路由表信息, 并 ping 192.168.48.172 默认网关地址

```

root@q-OptiPlex-5060: /home/q
q@q-OptiPlex-5060:~$ sudo su
sudo: /etc/sudoers.d is world writable
[sudo] password for q:
root@q-OptiPlex-5060:/home/q# ip route show
default via 192.168.48.172 dev rmnet_mhi0.1 proto static metric 100
169.254.0.0/16 dev rmnet_mhi0.1 scope link metric 1000
192.168.48.168/29 dev rmnet_mhi0.1 proto kernel scope link src 192.168.48.171
metric 100
root@q-OptiPlex-5060:/home/q#

```



```

root@q-OptiPlex-5060: /home/q
192.168.48.168/29 dev rmnet_mhi0.1 proto kernel scope link src 192.168.48.171
metric 100
root@q-OptiPlex-5060: /home/q# ping 192.168.48.172
PING 192.168.48.172 (192.168.48.172) 56(84) bytes of data.
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=0.760 ms
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.77 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.78 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.78 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.79 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.79 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.79 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.79 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.80 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.80 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.80 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.80 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=6.81 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=0.967 ms
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=7.08 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=7.11 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=7.11 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=7.11 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=7.11 ms (DUP!)

```

备注：loopback 功能验证不需要插 SIM 卡，拨号成功后，会获取一个固定的 IP 地址：192.168.48.171

5 iperf 测试

通过 iperf 打流测试可测试模块 loopback 功能是否实现，操作步骤如下

- (1. 备注测试 IPQ 上 loopback 功能时请在 PC 端进行打流操作，操作流程不变)
- (2. IPQ 具体操作指导请参考文档【机密】SDx55 PCIE 测试指导文档_V1.9 此处不再赘述)

```
root@OpenWrt:~# iperf -u -s &
```

(iperf -u -s -i 1) 实时显示 server 端数据

```

root@OpenWrt:~# -----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----

```

```
root@OpenWrt:~# iperf -u -c 192.168.48.172 -b 100M
```

(iperf -u -c 192.168.48.172 -b 100M -t 1000 -i 1) Client 端

```

-----
Client connecting to 192.168.48.172, UDP port 5001
Sending 1470 byte datagrams, IPG target: 112.15 us (kalman adjust)
UDP buffer size: 160 KByte (default)
-----

```

```
[ 3] local 192.168.48.171 port 36832 connected with 192.168.48.172 port 5001
```

```
[ 3] local 192.168.48.171 port 5001 connected with 192.168.48.172 port 36832
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0-10.0 sec    741 MBytes    620 Mbits/sec   0.922 ms   0/89163 (0%)
[ 3] 0.00-10.02 sec  450339 datagrams received out-of-order
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec    125 MBytes    105 Mbits/sec
[ 3] Sent 89163 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec    87.1 KBytes    71.2 Kbits/sec   0.921 ms   0/ 0 (nan%)
[ 3] 0.00-10.02 sec  450339 datagrams received out-of-order
```

需要注意的是,这种 loopback,由于回环的数据,不是严格按照 iperf 协议来回环的,所以 iperf/server 的统计结果是不准的,可以直接查看网卡的统计数量。

由于我们设置的回环倍数是 7 倍,可以看到 RX 的是 TX 的 7 倍。

如果带宽再往上加,会出现回环的数据量会超过 AP(CPU/USB/PCIE)的处理能力,会出现 RX 少于 7 倍的 TX。

```
root@OpenWrt:~# ifconfig pcie_mhi0
pcie_mhi0 Link encap:Ethernet HWaddr 02:50:F4:00:00:00
    inet addr:192.168.48.171 P-t-P:192.168.48.171 Mask:255.255.255.248
    inet6 addr: fe80::50:f4ff:fe00:0/64 Scope:Link
    UP POINTOPOINT RUNNING NOARP MTU:1500 Metric:1
    RX packets:624276 errors:0 dropped:0 overruns:0 frame:0
    TX packets:89184 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:935336280 (892.0 MiB) TX bytes:133583980 (127.3 MiB)
```

iperf 打流统计的是总体数据,因存在丢包率(使用的 UDP)所以计算 loopback·回环的时候 RX TX 应减去前一次 RX TX 的数据

```

root@QUEAFKL:/home/qquefda/Linux/quectel-CM
root@QUEAFKL:/home/qquefda/Linux# clear

root@QUEAFKL:/home/qquefda/Linux# ls
root@QUEAFKL:/home/qquefda/Linux# cd quectel-CM/
root@QUEAFKL:/home/qquefda/Linux/quectel-CM# ls
default.script  mbin-cn.c      QMIThread.c      udhccp.c
device.c        MPQCTL.h       QMIWanCM.c       udhccp_netlink.c
ethtool-copy.h  MPQMIL.h       quectel-CM       util.c
GobiNetCM.c     MPQMIL.h       quectel-qmi-proxy  util.h
libmnl          MPQMIL.h       quectel-qmi-proxy
main.c          NOTICE        quectel-qmi-proxy.c
Makefile        qmap_bridge_mode.c  ReleaseNote.txt
root@QUEAFKL:/home/qquefda/Linux/quectel-CM# ./quectel-CM -l 7 &
[1] 2166
root@QUEAFKL:/home/qquefda/Linux/quectel-CM# [03-05-11:01:18:608] Quctel_QConnectManager_Linux_V1.5.
9
[03-05-11:01:18:609] Find /sys/bus/usb/devices/2-5 ldVendor=0x2c7c ldProduct=0x800, bus=0x002, dev=0x0
02
[03-05-11:01:18:609] network interface '' or qmldev '' is not exist
[03-05-11:01:18:609] qmldevice detect failed
[03-05-11:01:18:609] netcard driver = pcie_mhi, driver version = V1.2.6
[03-05-11:01:18:610] qmap_mode = 1, qmap_size = 16384, muxid = 0x81, qmap_netcard = pcie_mhi0
[03-05-11:01:18:610] Modem works in QMI mode
[03-05-11:01:18:642] cdc_wdm_fd = 7
[03-05-11:01:19:642] QMIThreadSendQMITimeout pthread_cond_timeout_np timeout
[03-05-11:01:20:652] Get clientMMS = 20
[03-05-11:01:20:655] Get clientDMS = 1
[03-05-11:01:20:658] Get clientNAS = 2
[03-05-11:01:20:661] Get clientUIM = 1
[03-05-11:01:20:664] Get clientWDA = 1
[03-05-11:01:20:667] requestBaseBandVersion RMS00QGLAAR81A01M4G_BETA_201912040 1 [Nov 17 2019 19:00:
00]
[03-05-11:01:20:671] qmap_settings.rx_urb_size = 16384
[03-05-11:01:20:681] requestSetLoopBackState(loopback_state=1, replication_factor=7)
[03-05-11:01:20:687] requestGetSIMStatus SIMStatus: SIM_ABSENT
[03-05-11:01:20:690] requestGetProfile[1] cnet//j/o
[03-05-11:01:20:692] requestRegistrationState2 MCC: 0, MNC: 0, PS: Detached, DataCap: UNKNOWN
[03-05-11:01:20:696] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[03-05-11:01:20:696] ifconfig pcie_mhi0 down
[03-05-11:01:20:719] ifconfig pcie_mhi0 0.0.0.0
[03-05-11:01:20:724] SetLoopBackInd: loopback_state=1, replication_factor=7
[03-05-11:01:20:735] requestSetupDataCall WdsConnectionIPv4Handle: 0x52365640
[03-05-11:01:20:746] ifconfig pcie_mhi0 up
[03-05-11:01:20:751] Fail to access /usr/share/udhccp/default.script, errno: 2 (No such file or direct
ory)
[03-05-11:01:20:751] busybox udhccp -f -n -q -t 5 -l pcie_mhi0
[03-05-11:01:20:798] udhccp (v1.22.1) started
[03-05-11:01:20:814] Setting IP address 0.0.0.0 on pcie_mhi0
[03-05-11:01:20:848] Sending discover...
[03-05-11:01:20:892] Sending select for 192.168.48.171...
[03-05-11:01:20:940] Lease of 192.168.48.171 obtained, lease time 7200
[03-05-11:01:20:941] Setting IP address 192.168.48.171 on pcie_mhi0
++ 主机名搜索失败
[03-05-11:01:20:949] Deleting routers
[03-05-11:01:20:949] SIOCDLR: 没有那个进程
[03-05-11:01:20:949] Adding router 192.168.48.172
[03-05-11:01:20:966] Recreating /etc/resolv.conf

```

```

qquefda@QUEAFKL:~$
64 bytes from 192.168.48.172: icmp_seq=1 ttl=64 time=7.07 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.91 ms
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.94 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.94 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.94 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.94 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.95 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=2 ttl=64 time=5.95 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.61 ms
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.63 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.64 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.64 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.64 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.64 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=3 ttl=64 time=4.65 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.70 ms
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.73 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.74 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.74 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.74 ms (DUP!)
64 bytes from 192.168.48.172: icmp_seq=4 ttl=64 time=3.74 ms (DUP!)
AC
--- 192.168.48.172 ping statistics ---
4 packets transmitted, 4 received, +24 duplicates, 0% packet loss, time 3000ms
rtt min/avg/max/ndev = 1.590/5.280/7.677/1.585 ms
qquefda@QUEAFKL:~$ lperf -u -s &
[1] 2251
qquefda@QUEAFKL:~$
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 Kbyte (default)
lperf -u -c 192.168.48.172 -b 100m
Client connecting to 192.168.48.172, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 Kbyte (default)
[ 3] local 192.168.48.171 port 42203 connected with 192.168.48.172 port 5001
[ 3] local 192.168.48.171 port 5001 connected with 192.168.48.172 port 42203
[ 10] Interval Transfer Bandwidth
[ 3] 0.0-10.0 sec 120 Mbytes 101 Mbits/sec
[ 3] Sent 85471 datagrams
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 3] 0.0-10.0 sec 294 Mbytes 246 Mbits/sec 0.613 ms 26466/85469 (31%)
[ 3] 0.0-10.0 sec 150461 datagrams received out-of-order
[ 3] Server Report:
[ 3] 0.0-10.0 sec 294 Mbytes 246 Mbits/sec 0.613 ms 26466/85469 (31%)
[ 3] 0.0-10.0 sec 150461 datagrams received out-of-order
qquefda@QUEAFKL:~$ ifconfig pcie_mhi0
pcie_mhi0 Link encap:以太网 硬件地址: 82:50:f4:00:00:00
inet 地址: 192.168.48.171 点対点: 192.168.48.171 掩码: 255.255.255.248
UP POINTOPOINT RUNNING NOARP MTU:1500 跃点数:1
接收数据包: 598385 错误: 0 丢弃: 0 过载: 0 帧数: 0
发送数据包: 85487 错误: 0 丢弃: 0 过载: 0 载波: 0
接收字节: 897533062 (897.5 MB) 发送字节: 128562896 (128.5 MB)
qquefda@QUEAFKL:~$

```

6.常见问题

6.1 IPQ 上需要对导入的 pcie 驱动进行源码修改

pcie_mhi pcie 驱动下进行源码的修改

pcie_mhi/devices/mhi_netdev_quectel.c 文件


```

QUEAFKL: /home/qquefda/Remy/loopback/pcie_mhi/devices
#include <net/lp6.h>
#include <net/tcp.h>
#include <linux/usb/cdc.h>
#include "../core/mhi.h"

#ifdef CONFIG_ARCH_IPQ807x
#define CONFIG_QCA_NSS_DRV
#endif

#ifdef CONFIG_QCA_NSS_DRV
/* ***** Callback Hooks ***** */
static struct rmnet_nss_cb *rmnet_nss_callbacks __rcu __read_mostly;
/*
    depend on: qsdk/qca/feeds/nss-host/qca-nss-drv
    EXTRA_CFLAGS="-I$(STAGING_DIR)/usr/include/qca-nss-drv $(EXTRA_CFLAGS)"
    qsdk/qca/src/data-kernel/drivers/rmnet-nss/rmnet_nss.c
*/
#include "rmnet_nss.c"

static uint qca_nss_enabled = 1;
module_param( qca_nss_enabled, uint, S_IRUGO);
#endif

#ifdef CONFIG_MHI_NETDEV_MBIM
#define CONFIG_MHI_NETDEV_RMNET_ETH
#define QUECTEL_MHI_DATA_AGG
#define SDX55_LOOPBACK_BUG_FIX
#endif

#ifdef CONFIG_MHI_NETDEV_RMNET_ETH
static const unsigned char node_id[ETH_ALEN] = {0x02, 0x50, 0xf4, 0x00, 0x00, 0x00};
static const unsigned char default_noden_addr[ETH_ALEN] = {0x02, 0x50, 0xf3, 0x00, 0x00, 0x00};
#endif

#if defined(CONFIG_BRIDGE) || defined(CONFIG_BRIDGE_MODULE)
//define QUECTEL_BRIDGE_MODE
#endif

#ifdef QUECTEL_BRIDGE_MODE
static uint __read_mostly bridge_mode = BIT(0)*|BIT(1)*|;
module_param( bridge_mode, uint, S_IRUGO );
#endif

struct qmap_hdr {
    u8 cd_rsvd_pad;
    u8 mux_id;
    u16 pkt_len;
} __packed;
#define QUECTEL_QMAP_MUX_ID 0x81

#if 0
static void qmap_hex_dump(const char *tag, unsigned char *data, unsigned len) {
    uint i;
    uint *d = (uint *)data;

    printk(KERN_DEBUG "%s data=%p, len=%x\n", tag, data, len);
    len = (len+3)/4;
    for (i = 0; i < len; i+=4) {
        printk(KERN_DEBUG "%08x %08x %08x %08x\n", i*4, d[i+0], d[i+1], d[i+2], d[i+3]);
    }
}

```

●

6.2 IPQ 上测试 loopback 功能 iperf 打流需要指定网卡和端口

1. 前置条件

1.1 SPF11 版本

1.2 支持 IPQ 硬件加速的 qmi_wwan_q 驱动

1.3 支持 loopback 的 X55 模块，且模块配成 14 倍回环

2. 配置

2.1 由于我们现在只有 1G 网卡，所以 PC 必须连接 2 个网线到 IPQ8074。

如下 2 个网卡. 2 个 IP

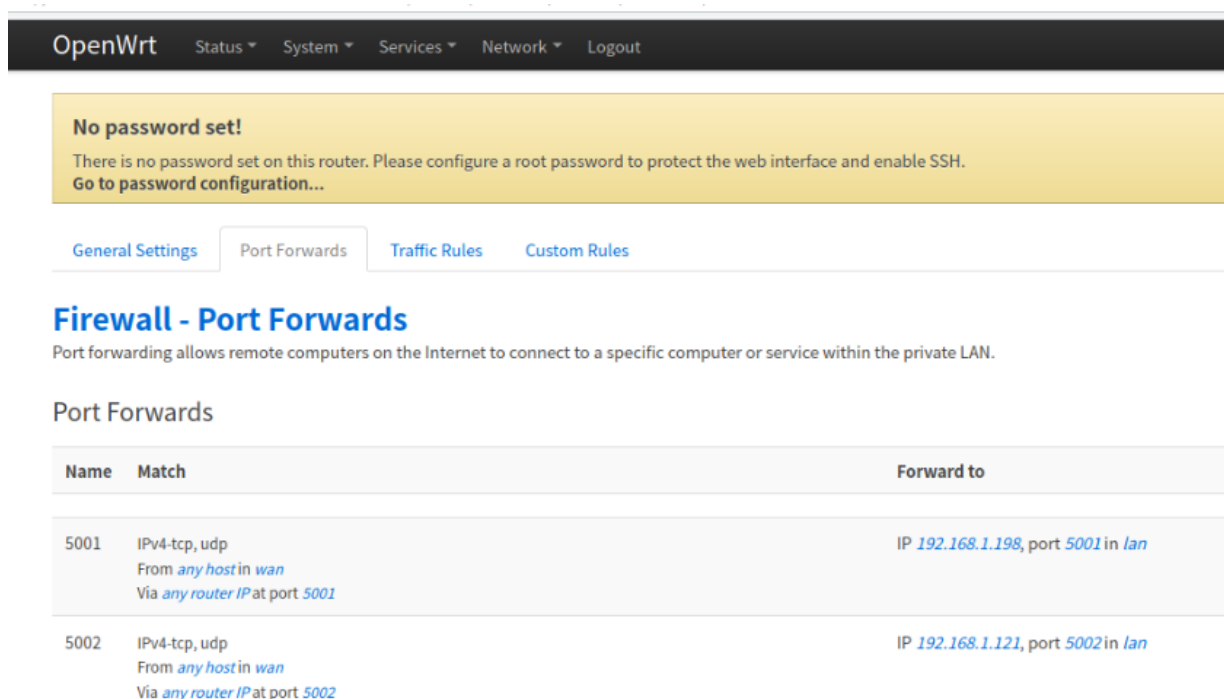
```
marco@marco:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.198  netmask 255.255.255.0  broadcast 192.168.1.255

marco@marco:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.121  netmask 255.255.255.0  broadcast 192.168.1.255
```

2.2 Ipq8074 配置端口转发.

分别转发 5001 和 5002 到电脑上的 2 个不同的网卡。

这里是 5001 到 192.168.1.198， 5002 到 192.168.1.121



2.3 lpg8074 配置负载均衡.

lpg8074 有 4 个核，但是每个核的处理能力都不强，所以必须把事情合理的分摊的每个核上。

2.3.1 我们模块的 USB 中断绑定到 CPU3

```
root@OpenWrt:~# cat /proc/interrupts | grep usb1
229:      22134329          0          0    39778536          GIC 172 Edge
xhci-hcd:usb1
```

```
root@OpenWrt:~# echo 8 > /sys/class/net/wwan0/queues/rx-0/rps_cpus
```

```
root@OpenWrt:~# cat /proc/irq/229/smp_affinity
```

```
8
```

2.3.2 我们模块网卡的 rps 绑定到所有 CPU 上，即我们网卡驱动收到的包，交给所有的 CPU 去处理。

```
root@OpenWrt:~# echo f > /sys/class/net/wwan0/queues/rx-0/rps_cpus
```

```
root@OpenWrt:~# cat /sys/class/net/wwan0/queues/rx-0/rps_cpus
```

```
f
```

注意上述的负载均衡方法，不是一成不变的。基本原来是保证每个核都有事做，不能一核忙死，其他核围观。

测速的时候，可以通过如下命令，查看每个核的负载情况

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

3. iperf 测试

PC 上运行 iperf，如下：

```
iperf -u -s -p 5001 -i 2 -B 192.168.1.198
```

```
iperf -u -c 192.168.48.172 -p 5001 -b 65M -t 7200 -B 192.168.1.198
```

```
iperf -u -s -p 5002 -i 2 -B 192.168.1.121
```

```
iperf -u -c 192.168.48.172 -p 5002 -B 192.168.1.121 -b 50M -t 7200
```

注意，这里-p port 和-B ip 的对应关系，必须和前面设置的端口转发一致。

4. 结果

4.1 2 个网卡加起来 2 个 1.2G

4.2 CPU3 忙死了，瓶颈应该是在我们的网卡驱动 (现在 QMAP 用的是 16KB，换成 31KB，应该能提升)

4.3 现在的覆盖平衡方法，应该不是最优的，应该 CPU0 和 1 的空闲还是很大的。

```

[ 3] 432.0-434.0 sec 108 MBytes 455 Mbits/sec 2.476 ms
[ 3] 432.00-434.00 sec 73292 datagrams received out-of-order
[ 3] 434.0-436.0 sec 106 MBytes 446 Mbits/sec 0.609 ms
[ 3] 434.00-436.00 sec 71763 datagrams received out-of-order
[ 3] 436.0-438.0 sec 104 MBytes 435 Mbits/sec 0.947 ms
[ 3] 436.00-438.00 sec 69850 datagrams received out-of-order
[ 3] 438.0-440.0 sec 103 MBytes 433 Mbits/sec 0.741 ms
[ 3] 438.00-440.00 sec 69683 datagrams received out-of-order
[ 3] 440.0-442.0 sec 100 MBytes 453 Mbits/sec 0.274 ms
[ 3] 440.00-442.00 sec 71000 datagrams received out-of-order
[ 3] 442.0-444.0 sec 100 MBytes 453 Mbits/sec 0.274 ms
[ 3] 442.00-444.00 sec 71000 datagrams received out-of-order
[ 3] 444.0-446.0 sec 100 MBytes 453 Mbits/sec 0.274 ms
[ 3] 444.00-446.00 sec 71000 datagrams received out-of-order
CPU %usr %nice %sys
all 0.00 0.00 0.00
[ 3] 0.0-37.7 sec 10:06:35
[ 3] Sent 77269 datagram 10:06:35
[ 3] Server Report: 10:06:37
[ 3] 0.0-37.7 sec 10:06:37
[ 3] 0.00-37.66 sec 10:06:37
marco@marco:~$ ^C
marco@marco:~$ iperf
00
----- 10:06:37
Client connecting to 192.168.1.198:5001
Binding to local address 192.168.1.198:5001
Sending 1470 byte data: 10:06:39
UDP buffer size: 208 10:06:39
----- 10:06:39
[ 3] local 192.168.1.198:5001
[ 3] 664.0-666.0 sec 121818 datagrams received out-of-order
[ 3] 666.0-668.0 sec 179 MBytes 752 Mbits/sec 0.346 ms 0/ 9384 (0%)
[ 3] 666.00-668.00 sec 121375 datagrams received out-of-order
[ 3] 668.0-670.0 sec 180 MBytes 754 Mbits/sec 0.497 ms 0/ 9490 (0%)
[ 3] 668.00-670.00 sec 121639 datagrams received out-of-order
[ 3] 670.0-672.0 sec 175 MBytes 735 Mbits/sec 0.305 ms 0/ 9041 (0%)
[ 3] 670.00-672.00 sec 118663 datagrams received out-of-order
[ 3] 672.0-674.0 sec 180 MBytes 756 Mbits/sec 0.413 ms 0/ 9455 (0%)
[ 3] 672.00-674.00 sec 121914 datagrams received out-of-order
[ 3] 674.0-676.0 sec 184 MBytes 772 Mbits/sec 0.899 ms 0/ 9757 (0%)
[ 3] 674.00-676.00 sec 124534 datagrams received out-of-order
[ 3] 676.0-678.0 sec 185 MBytes 777 Mbits/sec 0.455 ms 0/ 9933 (0%)
[ 3] 676.00-678.00 sec 125223 datagrams received out-of-order
[ 3] 678.0-680.0 sec 184 MBytes 772 Mbits/sec 0.769 ms 0/ 9841 (0%)
[ 3] 678.00-680.00 sec 124460 datagrams received out-of-order
[ 3] 680.0-682.0 sec 187 MBytes 785 Mbits/sec 0.449 ms 0/ 9897 (0%)
[ 3] 680.00-682.00 sec 126784 datagrams received out-of-order
[ 3] 682.0-684.0 sec 186 MBytes 781 Mbits/sec 0.386 ms 0/ 9890 (0%)
[ 3] 682.00-684.00 sec 125915 datagrams received out-of-order
[ 3] 684.0-686.0 sec 185 MBytes 777 Mbits/sec 0.714 ms 0/ 9884 (0%)
[ 3] 684.00-686.00 sec 125303 datagrams received out-of-order
[ 3] 686.0-688.0 sec 185 MBytes 777 Mbits/sec 0.345 ms 0/ 9932 (0%)
[ 3] 686.00-688.00 sec 125384 datagrams received out-of-order
0.00 2.02 14.05 0.00 0.00 0.00 81.82
0.00 0.00 99.50 0.00 0.00 0.00 0.50
0.00 16.92 82.59 0.00 0.00 0.00 0.50
----- 10:06:37
CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
all 0.25 0.00 0.76 0.00 6.19 64.22 0.00 0.00 0.00 28.57
0 0.00 0.00 1.02 0.00 5.61 58.67 0.00 0.00 0.00 34.69
1 1.02 0.00 2.04 0.00 2.55 14.80 0.00 0.00 0.00 79.59
2 0.00 0.00 0.00 0.00 0.00 98.99 0.00 0.00 0.00 1.01
3 0.00 0.00 0.00 0.00 16.50 83.50 0.00 0.00 0.00 0.00

```