

目录

X55 的回环功能介绍 ..... 1

用 QPST 工具设置 X55 的 NV 使能 loopback 功能 ..... 1

路由器/CPE 上设置 WAN 和端口转发..... 5

USB 和 PCIE 驱动 LOG ..... 6

CM 拨号 LOG ..... 7

CPU 负载平衡调节 ..... 8

iperf 和 mpstat 测试..... 9

Iperf 命令详解..... 10

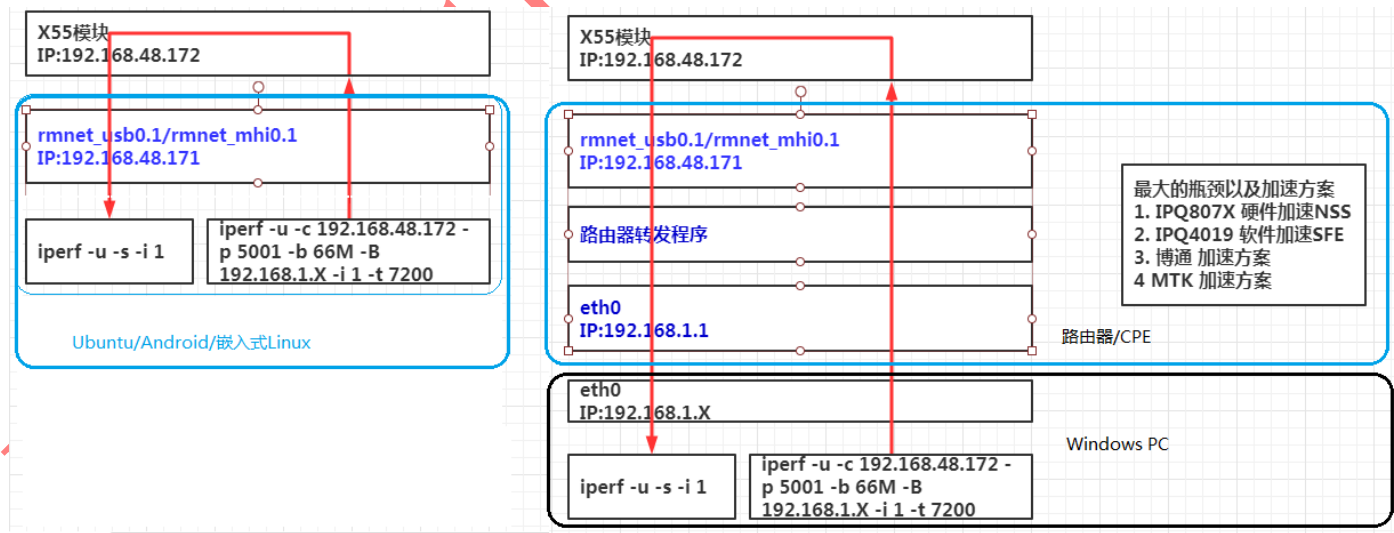
TCP 测速注意事项 ..... 11

X55 的回环功能介绍

X55 支持 rmnet/qmi 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。

5G 的速率非常高，客户的 AP 必须正确的配置之后(比如 CPU 的负载平衡，软件/硬件加速方案等等)，才能测到最高的速率。Loopback 测试不需要插 SIM 卡(也就是不需要实网)，也不需要仪器，非常方便测试/调试 AP 的性能。建议客户在做实网和仪器 TPUT 之前，都先做 loopback 测试。

如下是框架图：



用 QPST 工具设置 X55 的 NV 使能 loopback 功能

- 1. QPST 工具打开模块的 DM 口



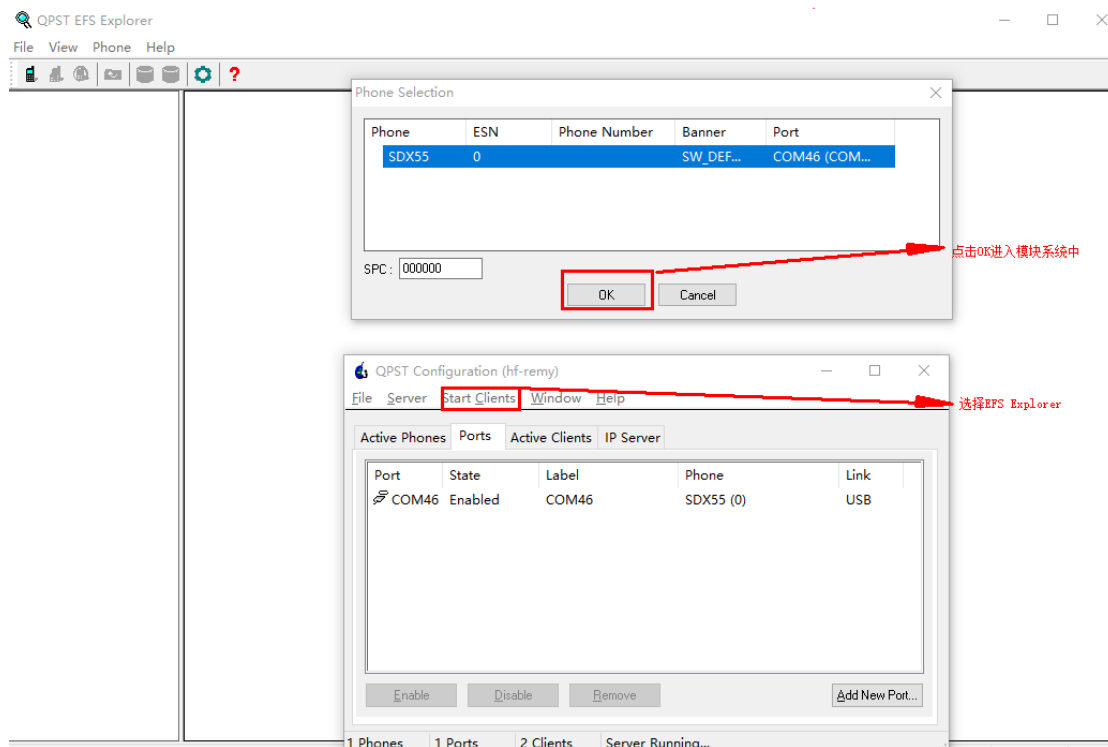
```

root@OpenWrt:/# busybox microcom /dev/ttyUSB2
at+qcfg="data_interface"
+QCFG: "data_interface",1,1

OK
root@OpenWrt:/# QLog -s 9000 -p /dev/mhi_DIAG
[000.000] QLog Version: Quectel_QLog_Linux&Android_V1.4.8
[000.001] will save log into dir: 9000
[000.001] will use device: /dev/mhi_DIAG
[000.001] will use filter file: default filter
[000.022] open /dev/mhi_DIAG ttyfd = 3
[000.022] Press CTRL+C to stop catch log.
[000.022] catch log via tty port
[000.022] Starting the TCP server(9000)...
[000.023] bind OK!
[000.023] listen OK!
Waiting the TCP Client...
[016.529] TCP Client 192.168.1.153:4547 connect

```

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



### 2.1 在如下文件夹下面创建文件/nv/item\_files/modem/data/3gpp/ps/loopback\_config.txt

(此处/data/3gpp/ps/文件夹需要自己创建)

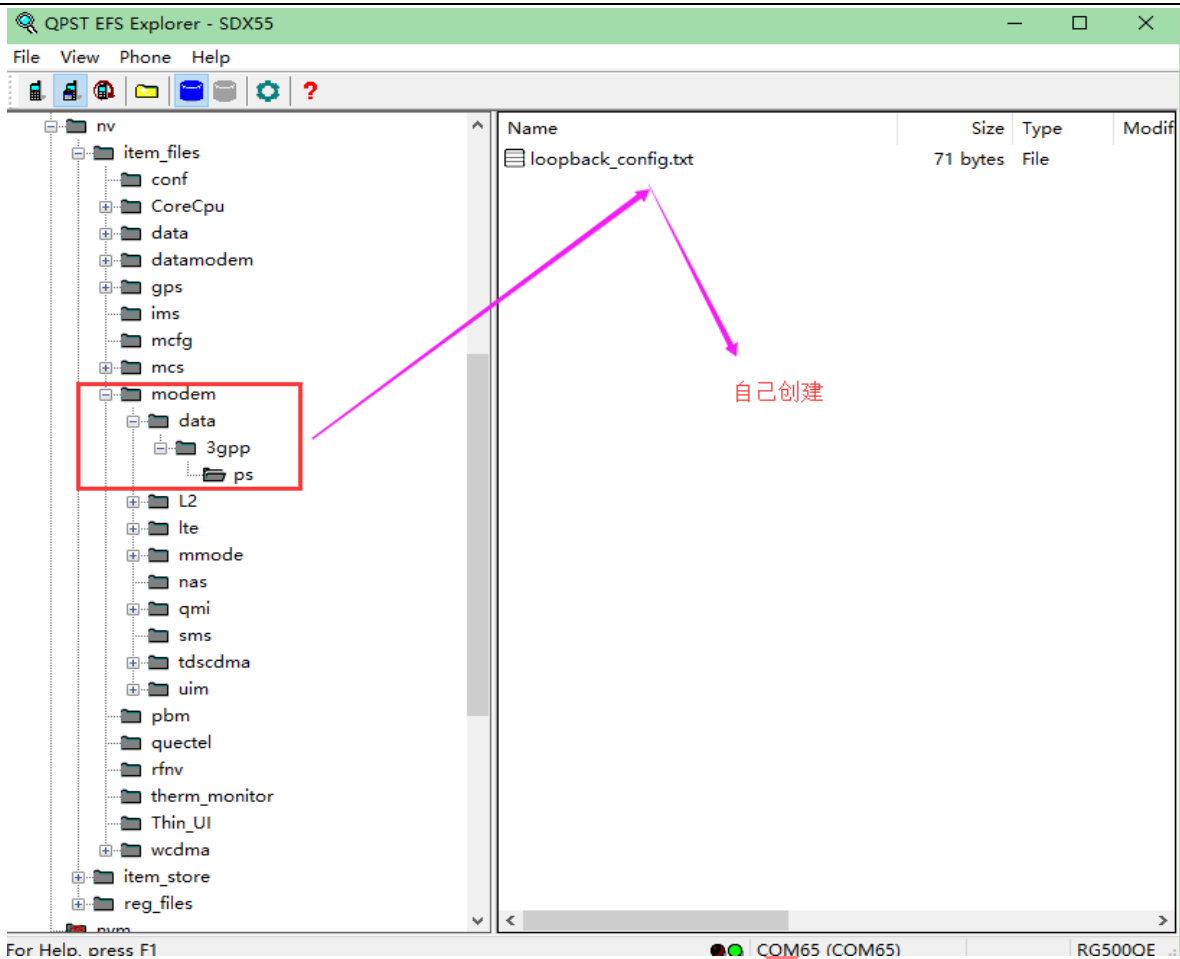
loopback\_config.txt 文档内容如下:

```

LOOPBACK_ENABLED:1;
L2_LOOPBACK_ENABLED:0;
SYS MODE:9;
REP_FACTOR:14;

```

REP\_FACTOR 是回环倍数，可自定义回环倍数。



2.2 设置完，重启模块生效， 重启之后，通过下面的命令，再次确认下是否正确。

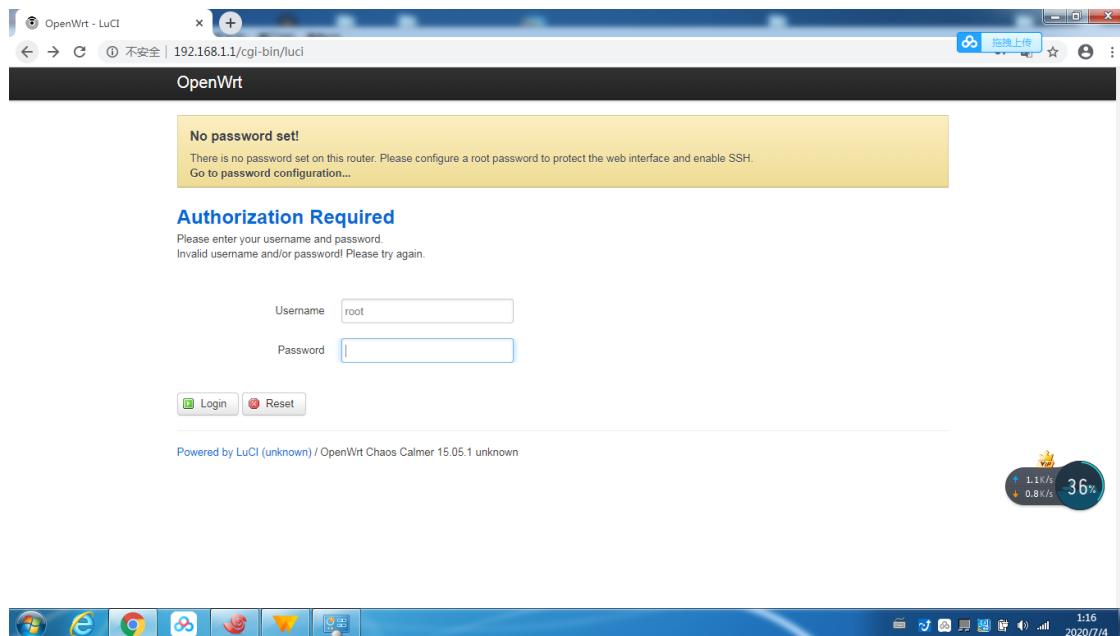
```
root@OpenWrt:~# busyboxmicrocom /dev/ttyUSB2
AT+QNVFR="/nv/item_files/modem/data/3gpp/ps/loopback_config.txt"
+QNVFR:
4C4F4F504241434B5F454E41424C45443A313B0D0A4C325F4C4F4F504241434B5F454E41424C45443A303B0D0A535953204D4F44453A393B0D0A
5245505F464143544F523A31343B
```

OK

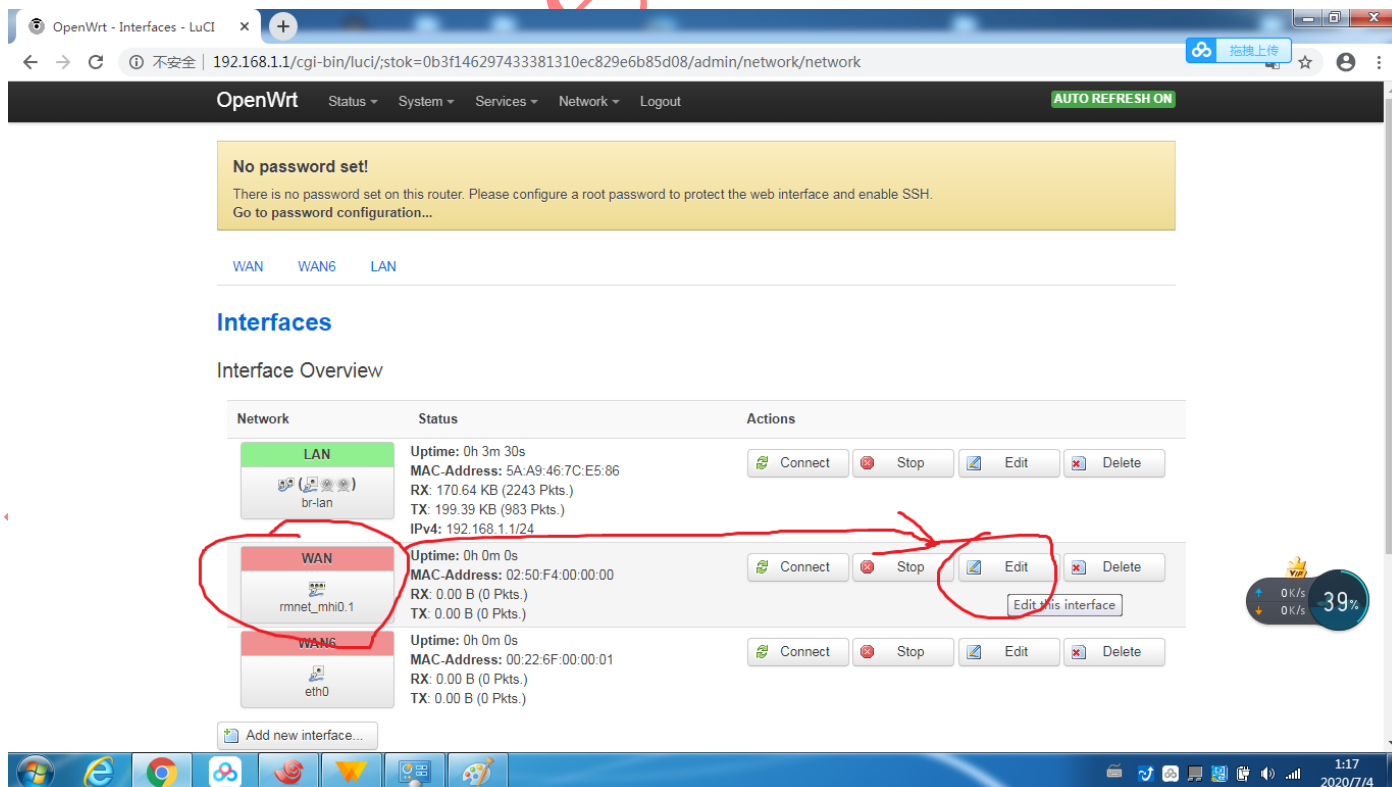
## 路由器/CPE 上设置 WAN 和端口转发

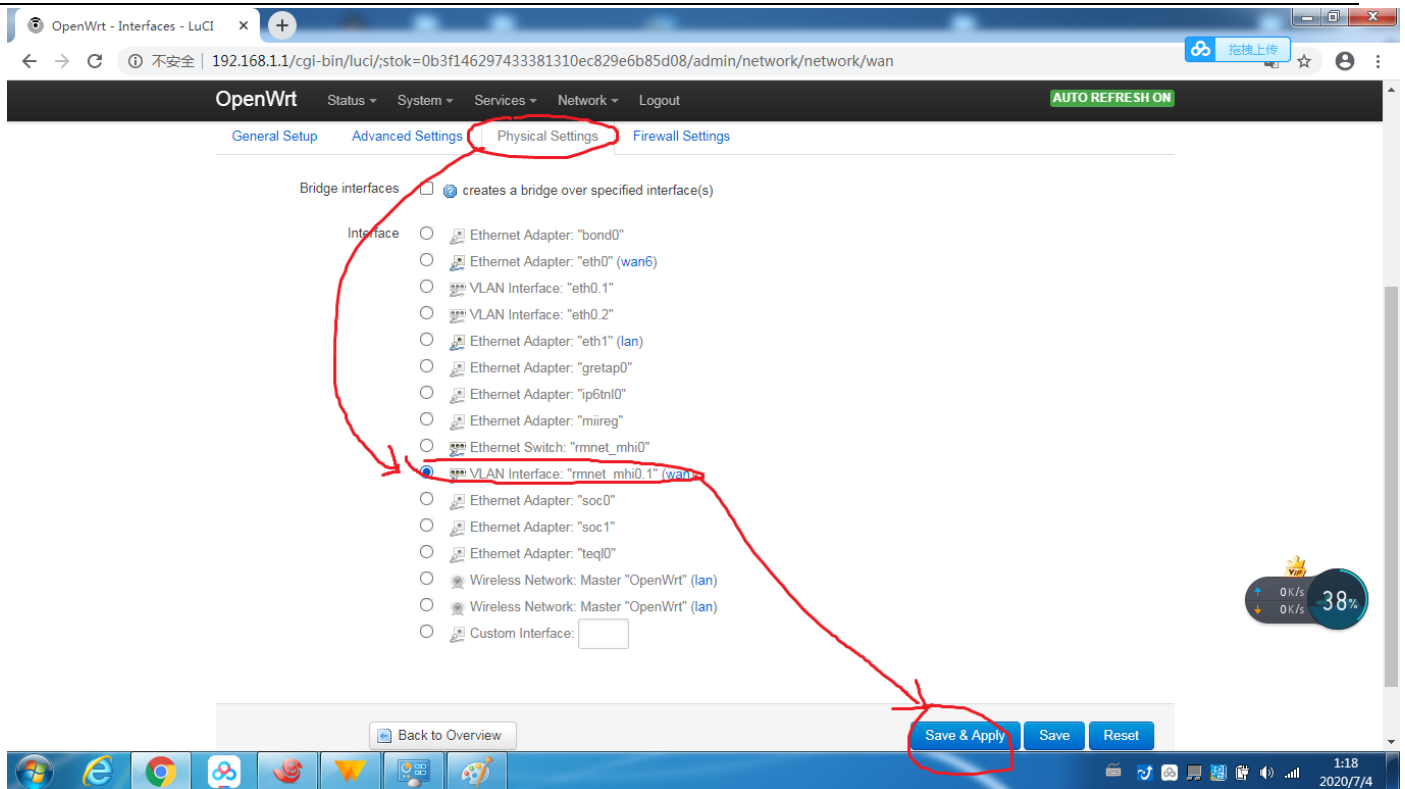
需要设置我们模块的网卡 (rmnet\_mhi0.1/rmnet\_usb0.1) 为 WAN，且设置端口转发。

- 1.1 Windows PC 通过网线连接路由器/CPE, 通过浏览器登录路由器的管理界面。网址一般是 192.168.1.1，用户名密码一般是 root/root 或者 admin/admin



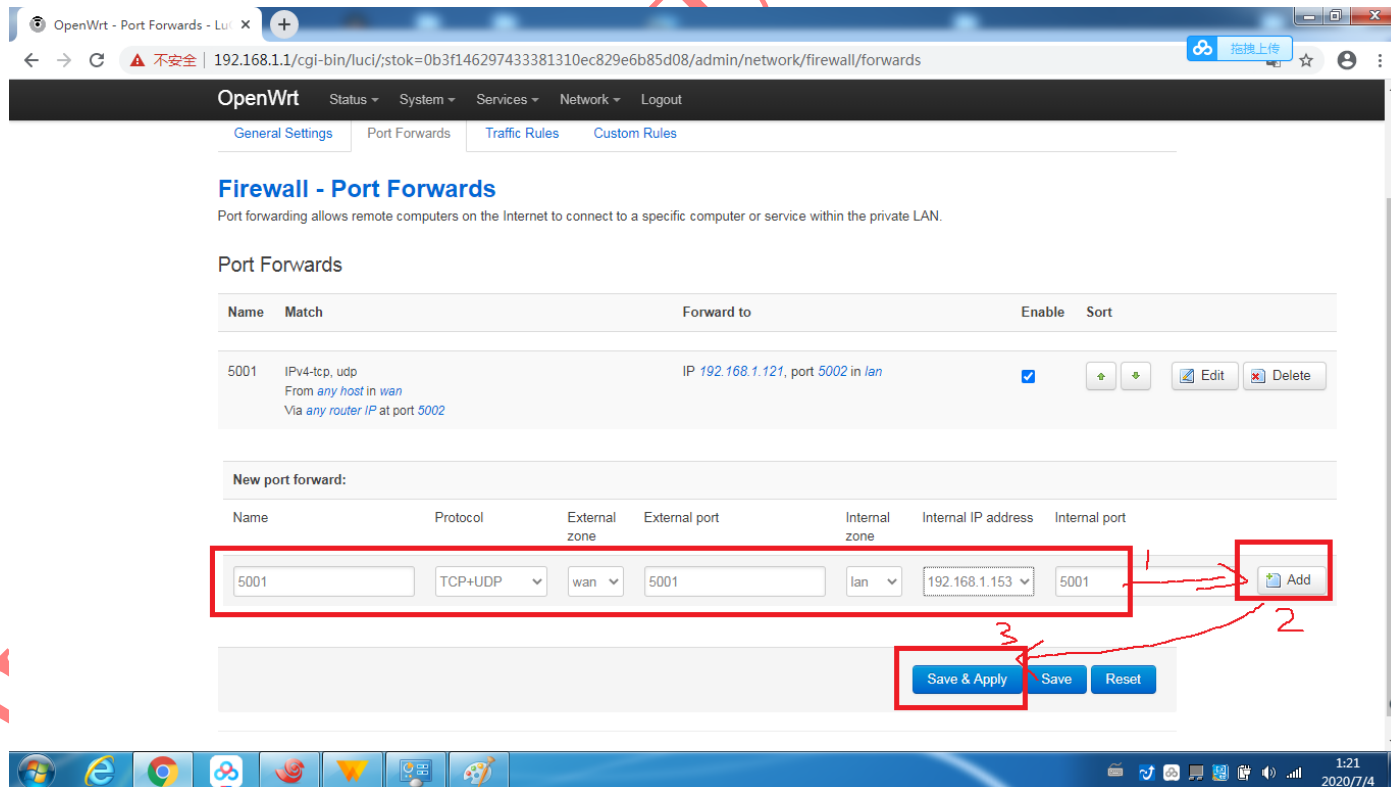
- 1.2 进入菜单 network -> interface -> WAN -> Edit -> Physical Settings  
选择我们模块的网卡为 WAN 卡，并点击 Save & Apply，流程如下图：





### 1.3 进入菜单 network -> Firewall -> Port Forwards

如下设置是端口转发 UDP 5001 到 IP 为 192.168.1.153 的 Windows PC



## USB 和 PCIE 驱动 LOG

驱动会生成 2 个网卡:

Qmi\_wwan\_q 驱动生成 rmnet\_usb0 和 rmnet\_usb0.1, rmnet\_usb0.1 用于上网。

Pcie\_mhi 驱动生成 rmnet\_mhi0 和 rmnet\_mhi0.1, rmnet\_mhi0.1 用于上网。

```
[ 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
```

## CM 拨号 LOG

#quectel-CM-1 14

-l 参数表示回环倍数. 注意实际的回环倍数是由前面的 NV 决定的, 这里只是告诉 CM 工具, 在做回环测速。

```
[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] QmiThreadSendQMIMessageThreadCondTimeoutNP 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] requestGetSIMStatusSIMStatus: 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 udhcpd 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] busyboxudhcpd -f -n -q -t 5 -i rmnet_usb0.1
udhcpd: started, v1.28.3
udhcpd: sending discover
udhcpd: sending select for 192.168.48.181
udhcpd: lease of 192.168.48.181 obtained, lease time 7200
[04-02_04:17:08:642] udhcpd: ifconfig rmnet_usb0.1 192.168.48.181 netmask 255.255.255.252 broadcast +
[04-02_04:17:08:651] udhcpd: setting default routers: 192.168.48.182
```

## CPU 负载均衡调节

网卡的工作，可以分为 3 个部分：

1. 收 QMAP 包
2. 拆分 QMAP 包成单个 IP 包
3. 提交 IP 包到内核协议栈。如果是路由器，就是交给 NAT 转发模块，由 NAT 转发数据包到对应的 LAN 网卡。

其中第三部分的 CPU 消耗是最高的。

如果不做任何设置，则这 3 部分都是跑在 CPU0 上，很容易把 CPU0 跑满，从而导致性能瓶颈。可以用 mpstat 工具查看每个核的负载。

可以参考如下的配置，把 3 部分工作分摊到 3 个 CPU 上。

### 1.1 USB/PCIE 中断默认跑在 CPU0 上。

收 QMAP 包跑在 CPU0 上

### 1.2 把 2 个网卡的负载分别调到 CPU1 和 CPU2 上

拆分 QMAP 包成单个 IP 包 跑在 CPU1 上。

提交 IP 包到内核协议栈 跑在 CPU2 上。

Qmi\_wwan\_q 驱动

```
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
```

pcie\_mhi 驱动

```
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
```



这里 2/4 这些数字表示哪个 CPU，比如 CPUX 用 BIT(X) 表示，实例如下：

```
Cpu0 - bit(0) - 1
Cpu1 - bit(1) - 2
Cpu2 - bit(2) - 4
Cpu3 - bit(3) - 8
Cpu4 - bit(4) - 10
```

注意如果客户的 AP 是大小核组合，则推荐把第三部分工作设置到大核 CPU 上去。比如 RK3399 是 4 小核+2 大核组合，cpu4/5 是大核。且推荐设置如下：

```
echo 10 > /sys/class/net/rmnet_usb0.1/queues/rx-0/rps_cpus
```

1.3 增大网卡待处理 skb 队列数量大小，防止网卡驱动短时间内收到 skb 过多，系统来不及处理。

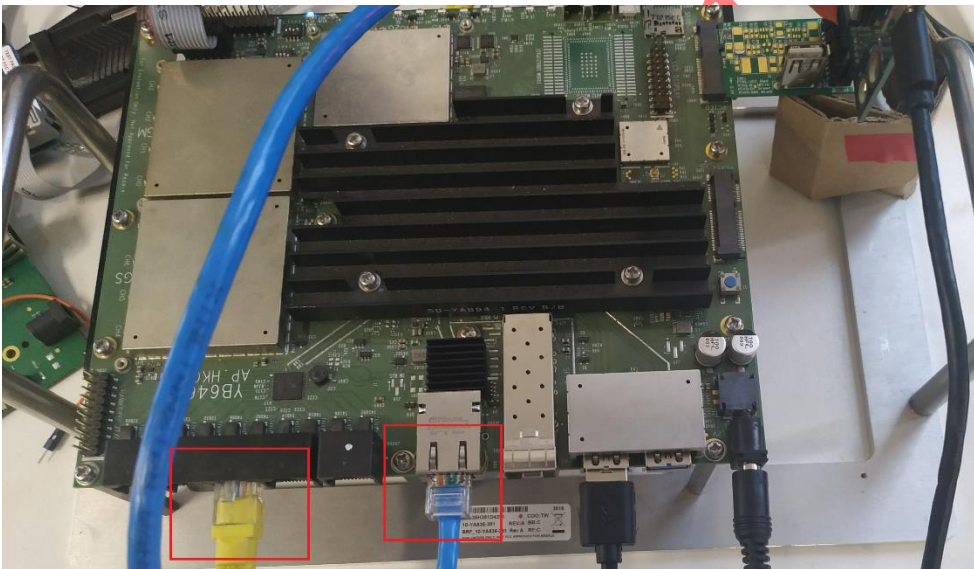
```
echo 2000 > /proc/sys/net/core/netdev_max_backlog
```

## iperf 和 mpstat 测试

1 准备 2 台 PC，其中一个台 PC 有 2.5G/10G 网卡，一台 PC 有 1G 网卡即可。

这里用 2 台 PC 的原因是：这个 2.5G 网卡在 Window 7 系统上，如果同时收发，性能不稳定。  
如果有线网卡的性能没问题，也可以只使用一台电脑，自发自收即可。

如下图，蓝色网线是 IPQ 的 10G 网卡插口，黄色位置是 1G 网卡插口



2 接蓝色网线的 PC(有 2.5G/10G 网卡的)装一个网速监控工具来监控网速。

Windows 系统可以用 TrafficMonitor, Ubuntu 可以用 nload

3 接黄色网线的 PC(有 1G 网卡的)，运行下面的 iperf 命令，上传数据到 X55。

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

4 测速的时候，需要监控 IPQ 的 CPU 负载，看看是否平衡。

```
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
```



-u	Server & Client	加这个表示 UDP 测速，不加表示 TCP 测速
-s	Server	表示 Server 模式
-c Server IP	Client	表示 Client 模式, -c 后面跟需要连接的 Server 的 IP 地址
-b UDP 带宽	Client	UDP 测速的时候，表示灌包的带宽。 比如 loopback 测试的时候，我们用 1G 的网卡来测试，则最好灌包 66M，这样回环的数据就是 14*66M，上下行加起来是 990M，刚好不超过 1G 网卡的理论上限。
-w TCP 窗口大小	Server & Client	TCP 测速的时候，表示 TCP 窗口的大小。比如 -w 1M 实际协商出来的窗口是 Server & Client 中的最小值。
-B Client IP	Server & Client	5G 模块的理论下行速度是 2G 多。如果用有线网卡来测试。 要么用 1 个 2.5G/10G 的网卡来测试,要么用多个 1G 网卡来测试。 如果多个 1G 网卡都是在同一台电脑上。 则可以用 -B IP 地址的方式来指定用哪个网卡。
-i 秒数	Server & Client	每隔多少秒显示一次测速的结果。 UDP 测速，需要看 Server 显示的速率以及丢包率，丢包率高说明 Client 灌包的带宽太大了。 TCP 测速，Server 和 Client 会显示同样的速率
-t 秒数	Client	测试多久，默认是 10 秒
-P 线程数	Client	同时灌几条 UDP/TCP 流。 UDP 测速一般一条流就可以跑满带宽。

## TCP 测速注意事项

TCP 是有 ACK 流控机制的，所以需要设置合适的窗口大小，窗口太小会导致速率上不去，窗口太大的话，万一需要重传，重传的数据太多，反而降低网速。

窗口大小的理论计算方式 带宽 X 延时 X 2。

实际中可以直接测出来一个合适的窗口，比如先只单线程测速，且设置窗口为 1M，看下测出来的速度是多少。

然后把窗口设置为 2M、3M、4M，看加到那个值之后，结果没有变化。

如果 iperf 是直接跑在 Linux 系统里，比如 Ubuntu、Android 之类的非路由器/CPE 设备。

Linux 系统本身有窗口大小限制的，比如下面的，iperf 设置窗口为 1M，但实际是 416KB

```
carl@carl-OptiPlex-7050:~$ iperf -s -w 1M
```

```
-----
Server listening on TCP port 5001
```

```
TCP window size: 416 KByte (WARNING: requested 1.00 MByte)
```

可以使用如下命令，调整下系统的 TCP 窗口限制

```
echo 4194304 > /proc/sys/net/core/rmem_max
```

```
echo 4194304 > /proc/sys/net/core/wmem_max
```

```
echo 2097152 > /proc/sys/net/core/rmem_default
```

```
echo 2097152 > /proc/sys/net/core/wmem_default
```

```
echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_rmem
```

```
echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_wmem
```

```
echo 1 > /proc/sys/net/ipv4/tcp_window_scaling
```