Supernova 测试仪
IPOE 测试配置手册

网测科技

2021/01/26

目录

1. 文档说明

本文档主要介绍 IPOE 的配置和测试过程。随着需求的不断改变，可能会对用例配置进行修改和升级，从而改变配置过程，所以有任何问题，请联系我们的售前或售后支持人员。

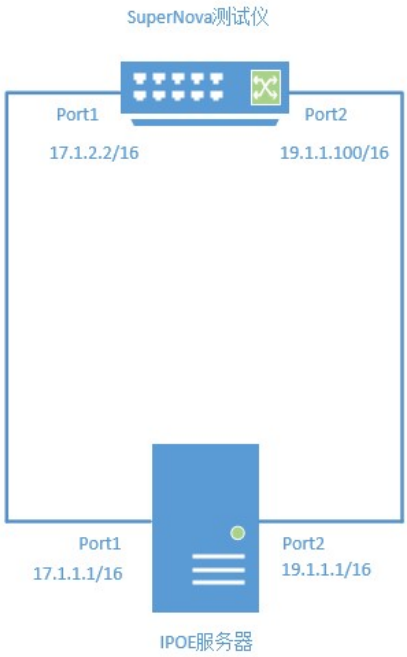
2. IPOE 测试说明

模拟 IPOE 工作原理，每个虚拟用户，在客户端接口上，虚拟出一个子接口，发送 DHCP 请求获取 IP 地址后，再广播 ARP 报文获取网关 MAC 地址，然后每个子接口都发送 UDP 报文，并在服务器端口上接收 UDP 报文。

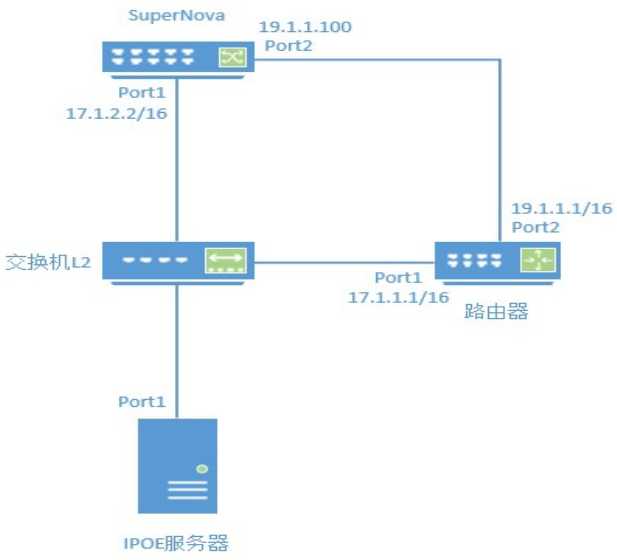
2.1 测试环境拓扑图

需要装一个 CentOS 做 IPOE 服务器，并打开内核转发功能，其实就是一台机器既当 IPOE 服务器，又当路由器转包。 所以上面至少有三个接口，一个是管理接口，一个是 IPOE 服务端口，一个路由转包接口。

拓扑 1 图注：Supernova 测试仪的 port1(DHCP/UDP 客户端，17.1.2.2/16)，IPOE 服务器和 Router 端口(17.1.1.1/16)路由转包接口(19.1.1.1/16)port2 (UDP 服务器端口，19.1.1.100/16)。



拓扑图 2 注： Supernova 测试仪的 port1(DHCP/UDP 客户端，17.1.2.2/16)IPOE 服务器获取地址。Supernova 测试仪的 port1(DHCP/UDP 客户端，17.1.2.2/16) Router port1(17.1.1.1/16)port2 (19.1.1.1/16)UDP 服务器端口 port2 (19.1.1.100/16)。



3.配置 CentOS

3.1 为 port1 和 port2 添加 ip 地址

```
[root@localhost ~]# ifconfig port1 17.1.1.1 netmask 255.255.0.0
[root@localhost ~]# ifconfig port2 19.1.1.1 netmask 255.255.0.0
[root@localhost ~]# ifconfig
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 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

mgmt1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.16.105  netmask 255.255.255.0  broadcast 192.168.16.255
        inet6 fe80::6e01:c147:b53d:d49a  prefixlen 64  scopeid 0x20<link>
        ether 04:92:26:c3:b2:9f  txqueuelen 1000  (Ethernet)
        RX packets 25837  bytes 26553838 (25.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14711  bytes 3374132 (3.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa2200000-a2220000

port1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 17.1.1.1  netmask 255.255.0.0  broadcast 17.1.255.255
        ether 6c:b3:11:1b:25:3c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 2311 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa2020000-a203ffff

port2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 19.1.1.1  netmask 255.255.0.0  broadcast 19.1.255.255
        ether 6c:b3:11:1b:25:3d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 1958 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa2000000-a201ffff

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:d9:35:9b  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

[root@localhost ~]#
```

3.2 下载 accel-ppp-1.12.0

wget https://jaist.dl.sourceforge.net/project/accel-ppp/accel-ppp-1.12.0.tar.bz2

解压：tar jxvf accel-ppp-1.12.0.tar.bz2

3.3 编译安装 IPOE 服务器

[root@localhost accel-ppp-1.12.0]# make deps

```
[root@localhost ~]# cd accel-ppp-1.12.0/
accel-cmd/  accel-pppd/ cmake/      contrib/    crypto/     drivers/    rfc/        .svn/
[root@localhost ~]# cd accel-ppp-1.12.0/
[root@localhost accel-ppp-1.12.0]# ls
accel-cmd  accel-ppp.conf  accel-pppd  cmake  CMakeLists.txt  config.h.in  contrib  COPYING  crypto  drivers  Makefile  README  rfc
[root@localhost accel-ppp-1.12.0]# make deps
```

[root@localhost accel-ppp-1.12.0]# make

```
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
[ 93%] Built target ipv6_dhcp
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
[ 94%] Built target ipv6_nd
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
[ 98%] Built target shaper
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
[ 98%] Built target accel-cmd
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
make[3]: 进入目录“/root/accel-ppp-1.12.0/build”
[ 99%] Generating driver/ipoe.ko
make: 进入一个未知的目录
make: *** /usr/src/kernels/3.10.0-957.el7.x86_64: 没有那个文件或目录。 停止。
make: 离开一个未知的目录
make[3]: *** [drivers/ipoe/driver/ipoe.ko] 错误 2
make[3]: 离开目录“/root/accel-ppp-1.12.0/build”
make[2]: *** [drivers/ipoe/CMakeFiles/ipoe_drv.dir/all] 错误 2
make[2]: 离开目录“/root/accel-ppp-1.12.0/build”
make[1]: *** [all] 错误 2
make[1]: 离开目录“/root/accel-ppp-1.12.0/build”
make: *** [all] 错误 2
```

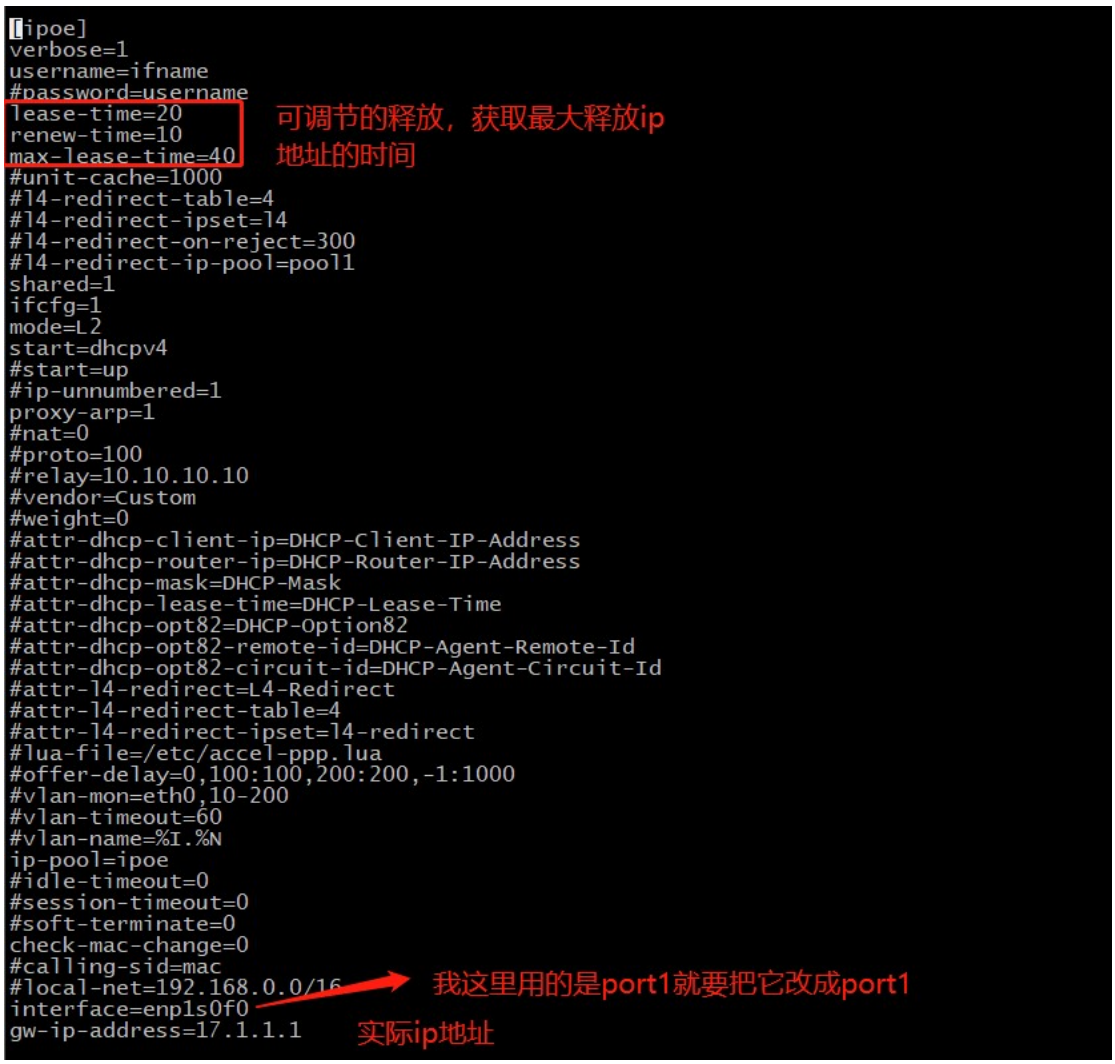
在 make 时如果有报错则需要 update 后再 make　　即 yum -y update

systemctl restart accel-ppp　　重启 IPOE 服务器

systemctl status accel-ppp　　查看 IPOE 服务器状态

3.4 修改 IPOE 服务器配置

vi /etc/accel-ppp.conf



配置方法请参考:

https://buildmedia.readthedocs.org/media/pdf/accel-ppp-demo/latest/accel-ppp-demo.pdf

3.5 打开内核转发功能

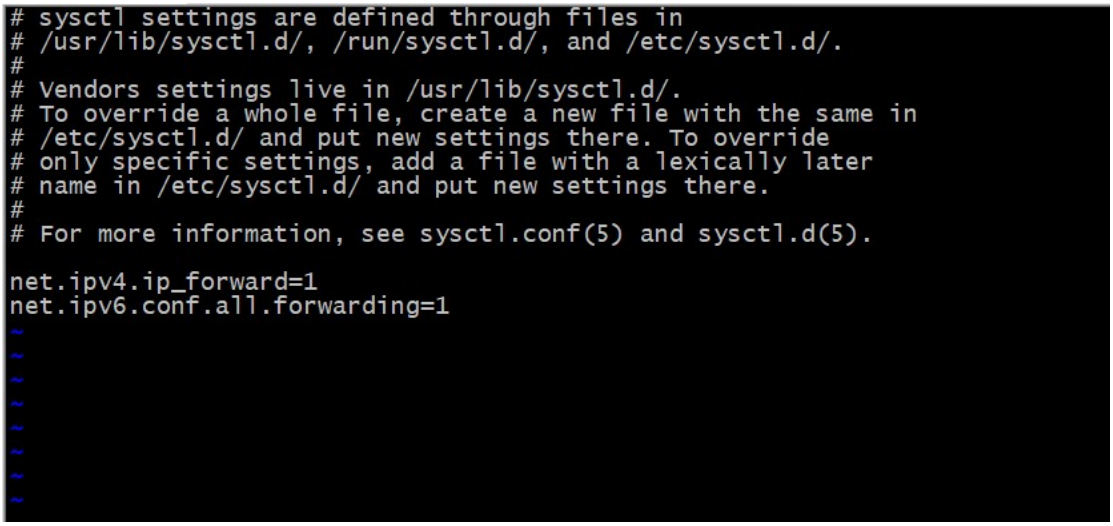
（拓扑图 2 不要执行此步）

[root@localhost accel-ppp-1.12.0]# vi /etc/sysctl.conf

将这两行加入这个配置文件

net.ipv4.ip_forward=1

net.ipv6.conf.all.forwarding=1



Reboot 重启服务器

3.6 关闭防火墙

systemctl disable firewalld

systemctl stop firewalld

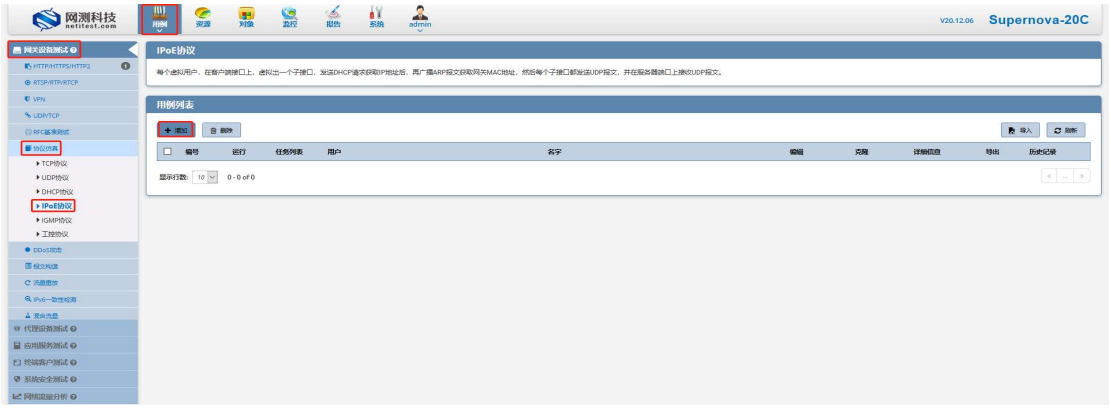
查看服务器上的接口情况，刚刚重启后的端口是没有什么数据的。

```
[root@localhost ~]# ifconfig
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 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

mgmt1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.16.105  netmask 255.255.255.0  broadcast 192.168.16.255
        inet6 fe80::6e01:c147:b53d:d49a  prefixlen 64  scopeid 0x20<link>
        ether 04:92:26:c3:b2:9f  txqueuelen 1000  (Ethernet)
        RX packets 321  bytes 35461 (34.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 225  bytes 33584 (32.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa2200000-a2220000

port1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 17.1.1.1  netmask 255.255.0.0  broadcast 17.1.255.255
        ether 6c:b3:11:1b:25:3c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 2207 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa2020000-a203ffff

port2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 19.1.1.1  netmask 255.255.0.0  broadcast 19.1.255.255
        ether 6c:b3:11:1b:25:3d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 2207 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa2000000-a201ffff
```

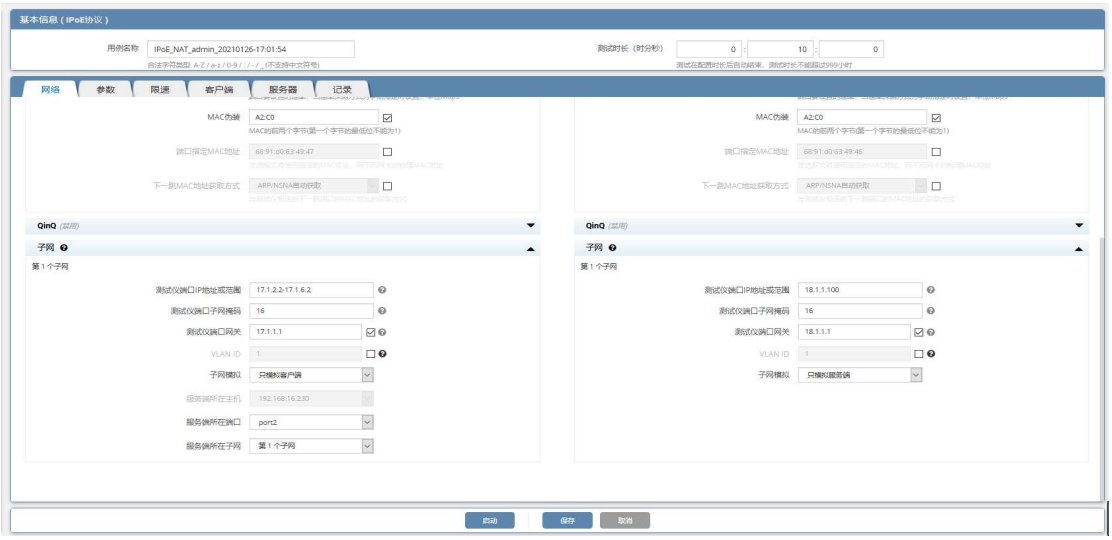
4. 运行 Supernova 测试仪 IPOE 用例

4.1 新建 IPoE 用例

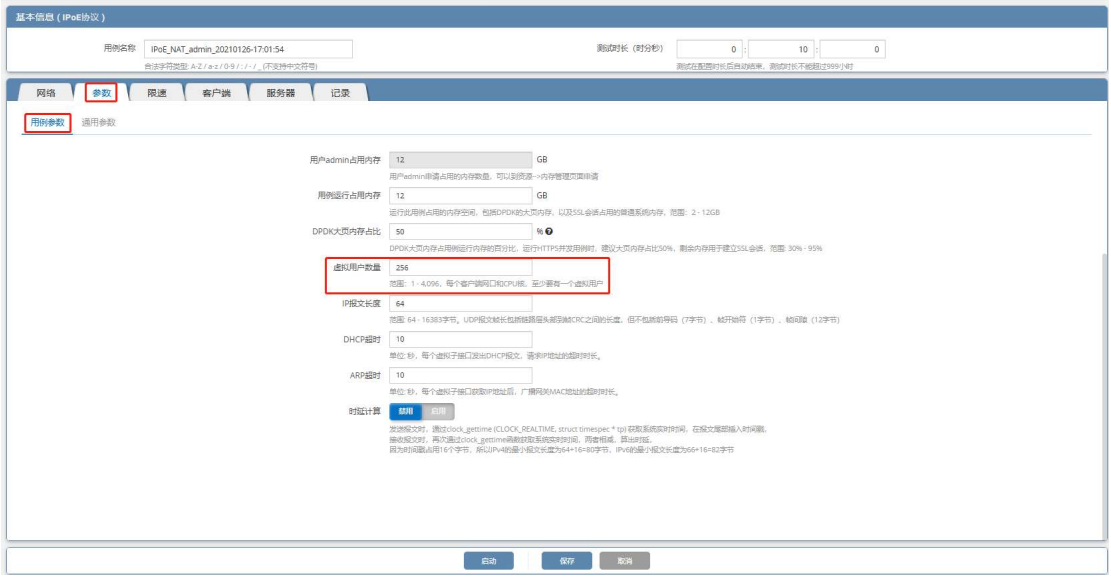
1）点击网关设备测试→协议仿真→IPoE 协议→增加，创建测试用例。



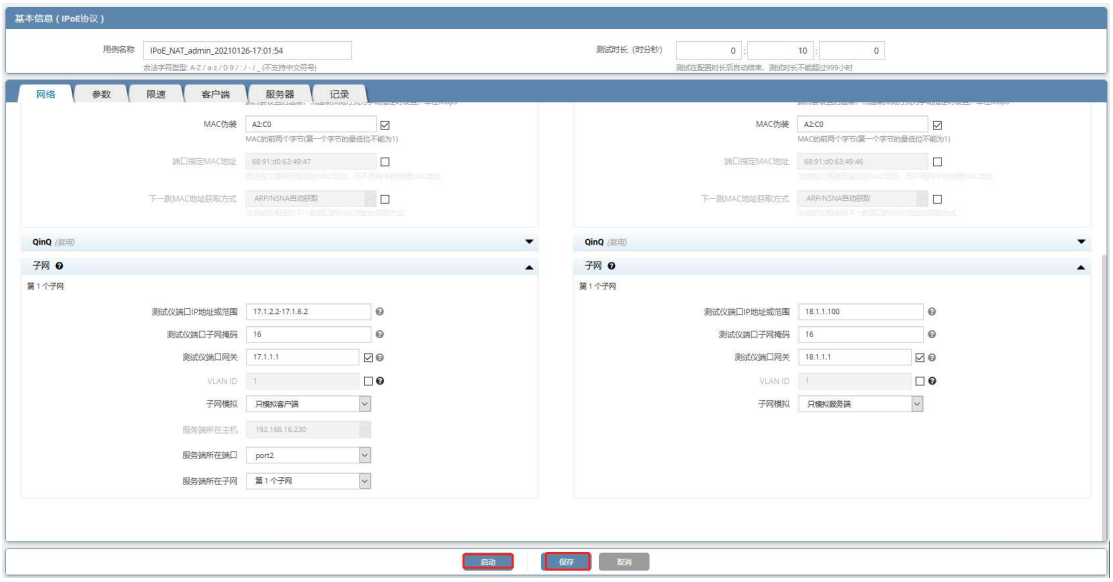
2）配置用例网络



3）配置虚拟用户以及服务器端口信息。



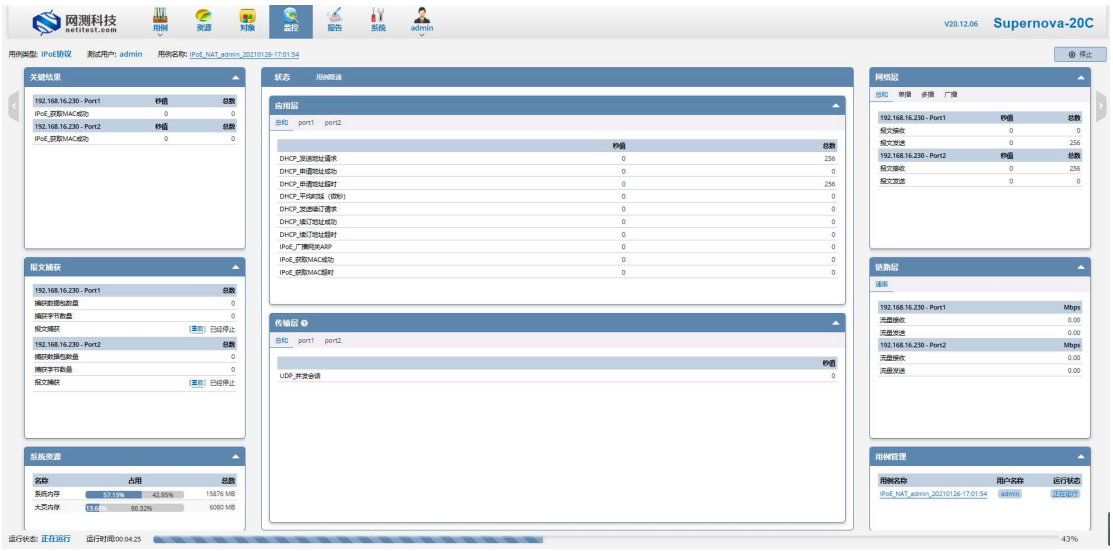
4）用例配置完成后，可以保存测试用例，也可以直接启动测试用例。



4.2 启动用例

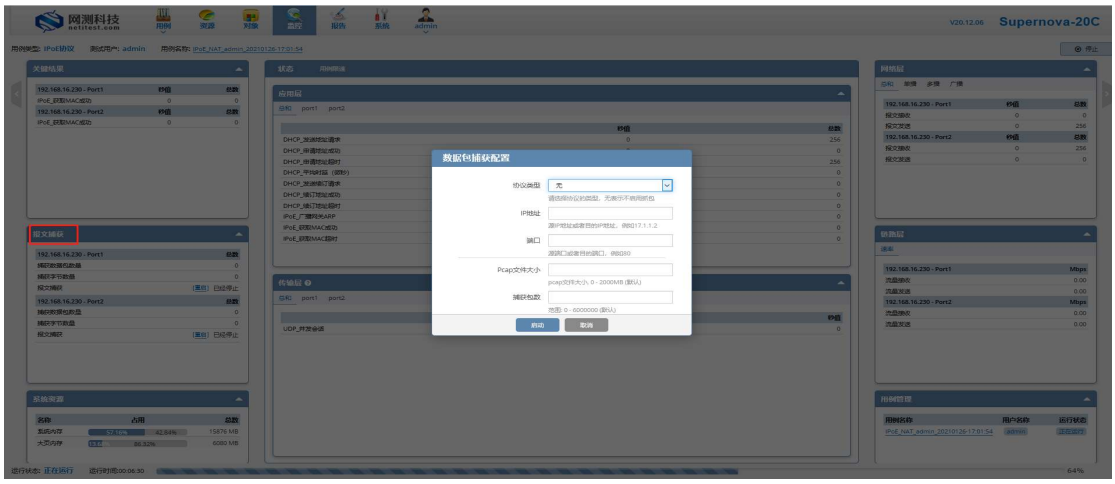
1）用例保存后自动返回主页面，点击运行配置保存的 IPoE 测试用例。

2）IPoE 测试用例启动后进入运行状态，显示详细运行数据。



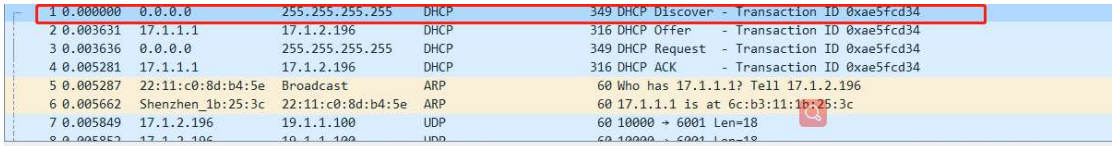
4.3 数据包分析

1）用例启动或者运行过程中，可以配置报文捕获，查看运行交互报文数据。



2）下载报文捕获的运行报文，可以通过 wireshark 打开查看报文数据。先拿出来一个虚拟用户做分析。

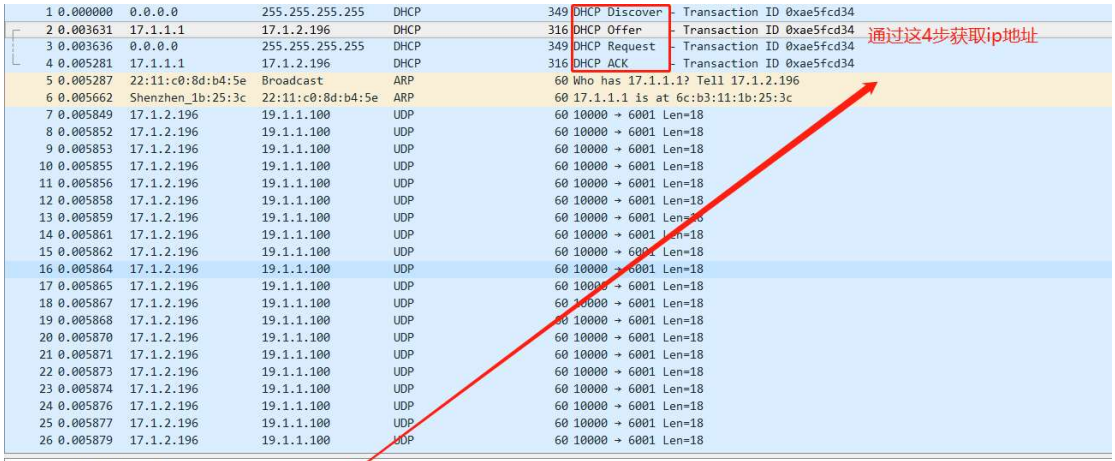
Client 端交互报文：



1	0.000000	0.0.0.0	255.255.255.255	DHCP	349 DHCP Discover - Transaction ID 0xae5fcd34
2	0.003631	17.1.1.1	17.1.2.196	DHCP	316 DHCP Offer - Transaction ID 0xae5fcd34
3	0.003636	0.0.0.0	255.255.255.255	DHCP	349 DHCP Request - Transaction ID 0xae5fcd34
4	0.005281	17.1.1.1	17.1.2.196	DHCP	316 DHCP ACK - Transaction ID 0xae5fcd34
5	0.005287	22:11:c0:8d:b4:5e	Broadcast	ARP	60 Who has 17.1.1.1? Tell 17.1.2.196
6	0.005662	Shenzhen_1b:25:3c	22:11:c0:8d:b4:5e	ARP	60 17.1.1.1 is at 6c:b3:11:1b:25:3c
7	0.005849	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18

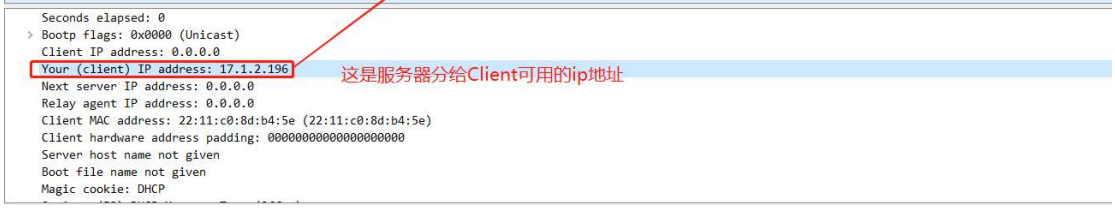
```
Fragment offset: 0
Time to live: 64
Protocol: UDP (17)
Header checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source: 0.0.0.0
Destination: 255.255.255.255
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 68, Dst Port: 67
Bootstrap Protocol (Discover)
```

源目端口号



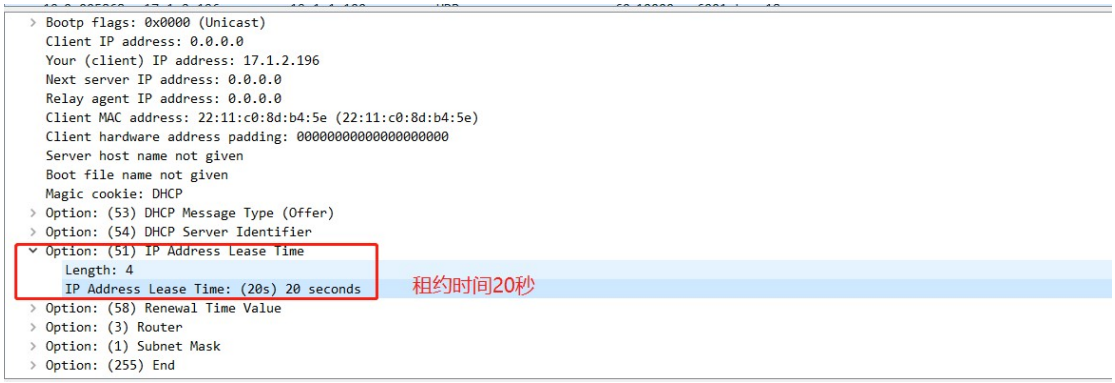
1	0.000000	0.0.0.0	255.255.255.255	DHCP	349 DHCP Discover - Transaction ID 0xae5fcd34
2	0.003631	17.1.1.1	17.1.2.196	DHCP	316 DHCP Offer - Transaction ID 0xae5fcd34
3	0.003636	0.0.0.0	255.255.255.255	DHCP	349 DHCP Request - Transaction ID 0xae5fcd34
4	0.005281	17.1.1.1	17.1.2.196	DHCP	316 DHCP ACK - Transaction ID 0xae5fcd34
5	0.005287	22:11:c0:8d:b4:5e	Broadcast	ARP	60 Who has 17.1.1.1? Tell 17.1.2.196
6	0.005662	Shenzhen_1b:25:3c	22:11:c0:8d:b4:5e	ARP	60 17.1.1.1 is at 6c:b3:11:1b:25:3c
7	0.005849	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
8	0.005852	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
9	0.005853	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
10	0.005855	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
11	0.005856	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
12	0.005858	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
13	0.005859	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
14	0.005861	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
15	0.005862	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
16	0.005864	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
17	0.005865	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
18	0.005867	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
19	0.005868	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
20	0.005870	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
21	0.005871	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
22	0.005873	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
23	0.005874	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
24	0.005876	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
25	0.005877	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
26	0.005879	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18

通过这4步获取ip地址



```
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 17.1.2.196
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: 22:11:c0:8d:b4:5e (22:11:c0:8d:b4:5e)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
```

这是服务器分给Client可用的ip地址



```
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 17.1.2.196
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: 22:11:c0:8d:b4:5e (22:11:c0:8d:b4:5e)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Offer)
Option: (54) DHCP Server Identifier
Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (20s) 20 seconds
Option: (58) Renewal Time Value
Option: (3) Router
Option: (1) Subnet Mask
Option: (255) End
```

租约时间20秒

1	0.000000	0.0.0.0	255.255.255.255	DHCP	349 DHCP Discover - Transaction ID 0xae5fcd34
2	0.003631	17.1.1.1	17.1.2.196	DHCP	316 DHCP Offer - Transaction ID 0xae5fcd34
3	0.003636	0.0.0.0	255.255.255.255	DHCP	349 DHCP Request - Transaction ID 0xae5fcd34
4	0.005281	17.1.1.1	17.1.2.196	DHCP	316 DHCP ACK - Transaction ID 0xae5fcd34
5	0.005287	22:11:c0:8d:b4:5e	Broadcast	ARP	60 Who has 17.1.1.1? Tell 17.1.2.196
6	0.005662	Shenzhen_1b:25:3c	22:11:c0:8d:b4:5e	ARP	60 17.1.1.1 is at 6c:b3:11:1b:25:3c
7	0.005849	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
8	0.005852	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
9	0.005853	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
10	0.005855	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
11	0.005856	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
12	0.005858	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
13	0.005859	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
14	0.005861	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
15	0.005862	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
16	0.005864	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
17	0.005865	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
18	0.005867	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
19	0.005868	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
20	0.005870	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18

获取到ip地址后发送目的ip为网关的广播
网关给出自己的mac地址

7	0.005849	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
8	0.005852	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
9	0.005853	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
10	0.005855	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
11	0.005856	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
12	0.005858	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
13	0.005859	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
14	0.005861	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
15	0.005862	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
16	0.005864	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
17	0.005865	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
18	0.005867	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
19	0.005868	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
20	0.005870	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
21	0.005871	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
22	0.005873	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
23	0.005874	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
24	0.005876	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
25	0.005877	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
26	0.005879	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18

Client向UDP服务器的6001端口发送udp报文

```
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 17.1.2.196
    Destination: 19.1.1.100
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
> User Datagram Protocol, Src Port: 10000, Dst Port: 6001
> Data (18 bytes)
```

Server 端交互报文:

1	0.000000	Shenzhen_1b:25:3d	Broadcast	ARP	60 Who has 19.1.1.100? Tell 19.1.1.1
2	0.000018	22:aa:13:01:01:64	Shenzhen_1b:25:3d	ARP	60 19.1.1.100 is at 22:aa:13:01:01:64
3	0.000084	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
4	0.000087	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
5	0.000091	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
6	0.000095	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
7	0.000097	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
8	0.000099	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
9	0.000101	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
10	0.000104	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
11	0.000106	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
12	0.000108	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
13	0.000109	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
14	0.000112	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
15	0.000113	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
16	0.000115	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
17	0.000118	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
18	0.000125	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
19	0.000135	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
20	0.000143	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
21	0.000151	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
22	0.000159	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
23	0.000167	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
24	0.000175	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
25	0.000182	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18
26	0.000190	17.1.2.196	19.1.1.100	UDP	60 10000 → 6001 Len=18

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: Shenzhen_1b:25:3d (6c:b3:11:1b:25:3d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

4.4 查看服务器接口信息