

[mirror是什么](#)  
[SPAN场景测试步骤](#)  
[VLAN场景测试步骤](#)  
[RSPAN\(远程端口镜像\)场景测试](#)  
[ERSPAN\(带封装远程端口镜像\)场景测试](#)  
[参考链接](#)

## mirror是什么

---

Mirror就是配置一个bridge，将某些包发给指定的mirrored ports

对于包的选择：

select\_all, 所有的包  
select\_dst\_port  
select\_src\_port  
select\_vlan

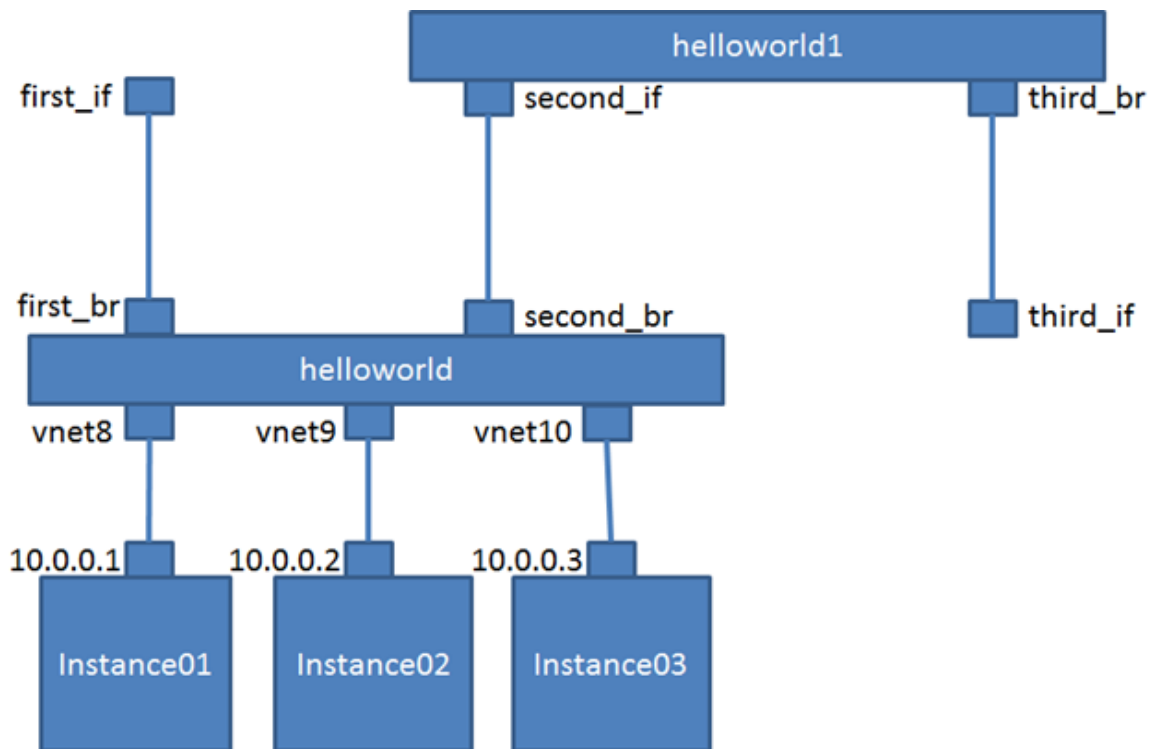
对于指定的目的：

output\_port (SPAN)  
output\_vlan (RSPAN)

## SPAN场景测试步骤

---

测试拓扑：



```
$ sudo ovs-vsctl add-port helloworld first_br
$ sudo ovs-vsctl add-port helloworld second_br -- set Port second_br tag=110
$ sudo ovs-vsctl add-br helloworld1
$ sudo ovs-vsctl add-port helloworld1 second_if -- set Port second_if tag=110
$ sudo ovs-vsctl add-port helloworld1 third_br -- set Port third_br tag=110
$ sudo ovs-vsctl show
```

c24322e6-8453-402a-afaf-64757ef231e9

```
Bridge helloworld
  Controller "tcp:16.158.165.102:6633"
    is_connected: true
  Port "vnet8"
    Interface "vnet8"
  Port first_br
    Interface first_br
  Port second_br
    tag: 110
    Interface second_br
  Port "vnet10"
    Interface "vnet10"
  Port helloworld
    Interface helloworld
      type: internal
  Port "vnet9"
    Interface "vnet9"
Bridge "helloworld1"
  Port second_if
    tag: 110
    Interface second_if
  Port "helloworld1"
    Interface "helloworld1"
      type: internal
  Port third_br
    tag: 110
    Interface third_br
ovs_version: "2.0.1"
```

然后我们在first\_br上面mirror所有进出vnet8的包

```
$ sudo ovs-vsctl -- set bridge helloworld mirrors=@m -- --id=@vnet8 get Port vnet8 -- --
id=@first_br get Port first_br -- --id=@m create Mirror name=mirrorvnet8 select-dst-port=@vnet8
select-src-port=@vnet8 output-port=@first_br
19ed2f51-3245-4d5b-8e6c-67ccbb7c7ebd
```

这时候我们监听first\_if, 并且从instance01里面ping 10.10.10.3, 可以看到下面的效果:

```
1 $ sudo tcpdump -n -i first_if icmp
2 tcpdump: WARNING: first_if: no IPv4 address assigned
3 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
4 listening on first_if, link-type EN10MB (Ethernet), capture size 65535
5 23:58:02.310198 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
  1, length 64
6 23:58:02.312447 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
  1, length 64
7 23:58:02.314314 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
  1, length 64
8 23:58:03.311894 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
  2, length 64
9 23:58:03.312266 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
  2, length 64
```

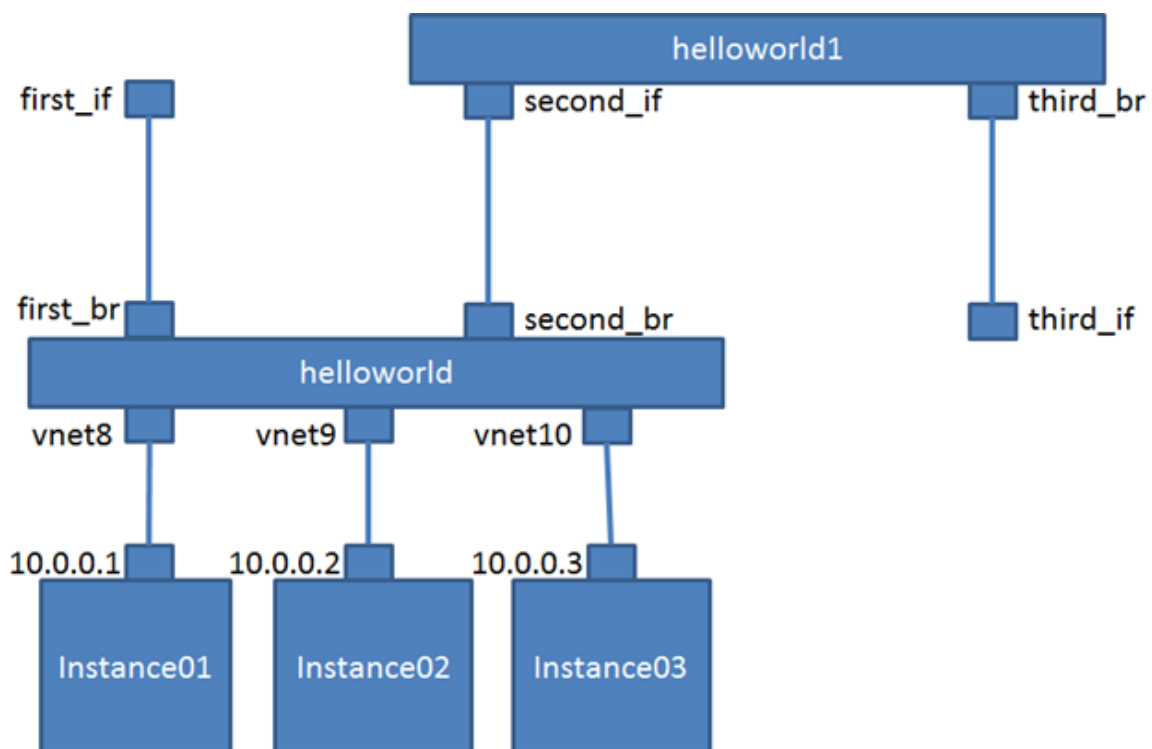
```

10 23:58:04.313522 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
    3, length 64
11 23:58:04.313739 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
    3, length 64
12 23:58:05.314827 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
    4, length 64
13 23:58:05.314979 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
    4, length 64
14 23:58:06.316870 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
    5, length 64
15 23:58:06.317156 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
    5, length 64
16 23:58:07.318242 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
    6, length 64
17 23:58:07.318481 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
    6, length 64
18 23:58:08.319579 IP 10.10.10.1 > 10.10.10.3: ICMP echo request, id 3200, seq
    7, length 64
19 23:58:08.319802 IP 10.10.10.3 > 10.10.10.1: ICMP echo reply, id 3200, seq
    7, length 64

```

## VLAN场景测试步骤

测试拓扑:



我们对进入vnet9的所有进出包，然而output到一个vlan 110

```

$ sudo ovs-vsctl -- set bridge helloworld mirrors=@m -- --id=@vnet9 get Port vnet9 -- --id=@m
create Mirror name=mirrorvnet9 select-dst-port=@vnet9 select-src-port=@vnet9 output-vlan=110
cb361fa2-914d-494b-94ef-c625d194247c

```

在helloworld1中也要配置从110来的，都output到vlan 110

```

$ sudo ovs-vsctl -- set bridge helloworld1 mirrors=@m -- --id=@m create Mirror name=mirrorvlan
select-vlan=110 output-vlan=110
cef13445-c6ea-45e7-bb9d-1a267b24c91c

```

disable mac address learning for vlan 110

```
$ sudo ovs-vsctl set bridge helloworld flood-vlans=110
```

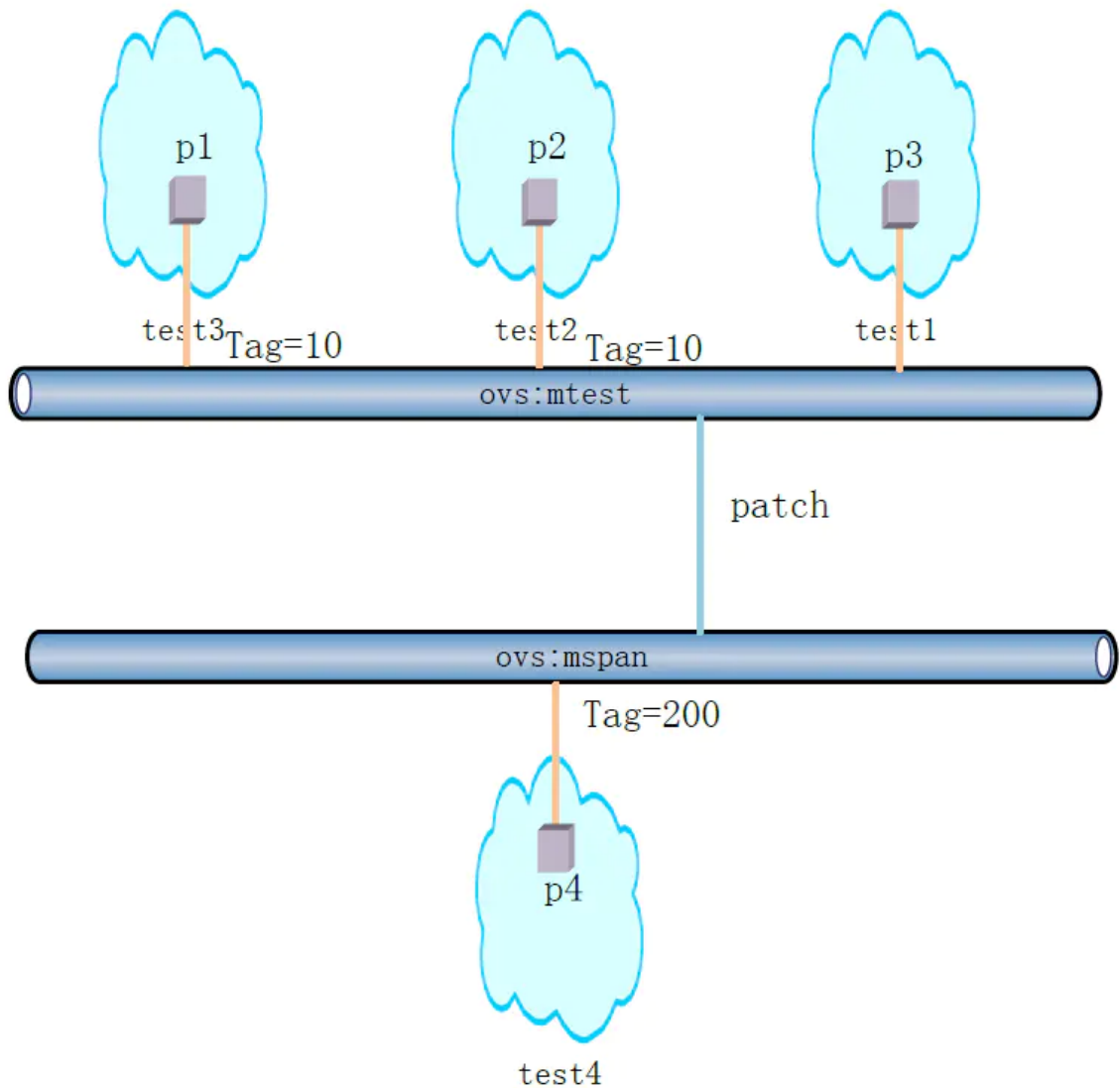
```
$ sudo ovs-vsctl set bridge helloworld1 flood-vlans=110
```

这时候我们监听third\_if, 并且从instance02里面ping 10.10.10.3, 可以看到下面的效果

```
1 $ sudo tcpdump -n -i third_if icmp
2 tcpdump: WARNING: third_if: no IPv4 address assigned
3 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
4 listening on third_if, link-type EN10MB (Ethernet), capture size 65535
   bytes
5 00:24:38.089192 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   1, length 64
6 00:24:38.090844 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   1, length 64
7 00:24:38.093228 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   1, length 64
8 00:24:39.090506 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   2, length 64
9 00:24:39.091024 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   2, length 64
10 00:24:40.091945 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   3, length 64
11 00:24:40.092202 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   3, length 64
12 00:24:41.093826 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   4, length 64
13 00:24:41.093917 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   4, length 64
14 00:24:42.095681 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   5, length 64
15 00:24:42.095915 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   5, length 64
16 00:24:43.097414 IP 10.10.10.2 > 10.10.10.3: ICMP echo request, id 2999, seq
   6, length 64
17 00:24:43.097586 IP 10.10.10.3 > 10.10.10.2: ICMP echo reply, id 2999, seq
   6, length 64
```

## RSPAN(远程端口镜像)场景测试

测试拓扑:



RSPAN场景种需要增加path 链接两个bridge。

添加ovs bridge:

```
1 #ovs-vsctl add-br mspan
2 #ip netns add test4
3 #ovs-vsctl add-port mspan p4 -- set interface p4 type=internal -- set port p4
  tag=200
4 #ip link set dev p4 netns test4
5 #ip netns exec test4 ip link set up p4
```

patch口:

```
1 #ovs-vsctl \
2 -- add-port mtest mpatch0 \
3 -- set interface mpatch0 type=patch options:peer=mpatch1 \
4 -- add-port mspan mpatch1 \
5 -- set interface mpatch1 type=patch options:peer=mpatch0
```

设置mirror:

```
1 需要将p1或p2口的数据，镜像至vlan 20
2 #ovs-vsctl -- --id=@p1 get port p1 \
3 -- --id=@m create mirror name=m0 select-src-port=@p1 output-vlan=200 \
```

```

4  -- set bridge mtest mirrors=@m
5  在mspan上, 将VLAN 22的流量镜像至p4口
6  #ovs-vsctl -- --id=@p4 get port p4 \
7  -- --id=@m create mirror name=m1 select-all=true select-vlan=200 output-
   port=@p4 \
8  -- set bridge mspan mirrors=@m
9
10 # ovs-vsctl list mirror
11 _uuid          : 46ef8c49-a242-4558-8035-fd7aae88f3e3
12 external_ids   : {}
13 name           : "m0"
14 output_port    : []
15 output_vlan    : 200
16 select_all     : false
17 select_dst_port : []
18 select_src_port : [985b0e1d-8cf6-486c-b7ab-3336cffe0e7a]
19 select_vlan    : []
20 statistics     : {tx_bytes=7658, tx_packets=81}
21
22 _uuid          : 3a86586a-8fe2-4409-bbd0-31017d1ff564
23 external_ids   : {}
24 name           : "m1"
25 output_port    : []
26 output_vlan    : []
27 select_all     : true
28 select_dst_port : []
29 select_src_port : []
30 select_vlan    : [200]
31 statistics     : {tx_bytes=0, tx_packets=0}

```

抓包测试:

```

1  # ip netns exec test1 ping 192.168.10.12
2  PING 192.168.10.12 (192.168.10.12) 56(84) bytes of data.
3  64 bytes from 192.168.10.12: icmp_seq=1 ttl=64 time=0.442 ms
4  64 bytes from 192.168.10.12: icmp_seq=2 ttl=64 time=0.051 ms
5
6  # ip netns exec test4 tcpdump -ni p4 -v -e icmp or arp
7  tcpdump: WARNING: p4: no IPv4 address assigned
8  tcpdump: listening on p4, link-type EN10MB (Ethernet), capture size 65535
   bytes
9  16:30:40.649825 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
   (0x0800), length 98: (tos 0x0, ttl 64, id 20855, offset 0, flags [DF],
10  proto ICMP (1), length 84)
   192.168.10.11 > 192.168.10.12: ICMP echo request, id 6512, seq 1,
   length 64
11 16:30:41.649750 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
   (0x0800), length 98: (tos 0x0, ttl 64, id 20856, offset 0, flags [DF],
   proto ICMP (1), length 84)
12 192.168.10.11 > 192.168.10.12: ICMP echo request, id 6512, seq 2,
   length 64
13 16:30:42.649747 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
   (0x0800), length 98: (tos 0x0, ttl 64, id 20857, offset 0, flags [DF],
   proto ICMP (1), length 84)
14 192.168.10.11 > 192.168.10.12: ICMP echo request, id 6512, seq 3,
   length 64
15 16:30:43.649736 06:7c:a0:69:97:f2 > c6:94:5

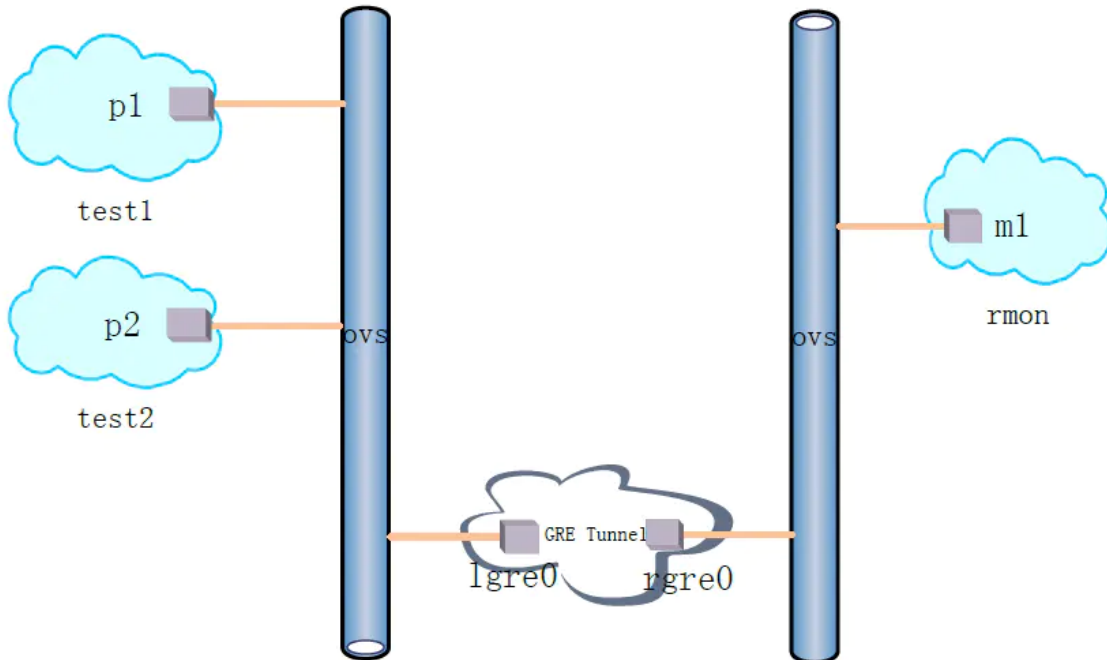
```

# ERSPAN(带封装远程端口镜像)场景测试

封装远程端口镜像，ovs中，利用gre隧道路由镜像流量。

- HOST1: 172.21.12.8
- HOST2: 172.21.12.163

拓扑模型如下：



HOST1:

```
#ovs-vsctl add-port mtest lgre0 -- set interface lgre0 type=gre
options:remote_ip=172.21.12.163 options:key=0x0010
```

HOST2:

```
1 #ovs-vsctl add-br mtest
2 #ovs-vsctl add-port mtest m1 -- set interface m1 type=internal
3 #ip netns add rmon
4 #ip link set m1 netns rmon
5 #ip netns exec rmon ip link set up m1
6 添加vxlan端口:
7 #ovs-vsctl add-port mtest rgre0 -- set interface rgre0 type=gre
options:remote_ip=172.21.12.8 options:key=0x0010
```

HOST1添加mirror:

```
1 #ovs-vsctl -- --id=@p1 get port p1 \
2 -- --id=@lgre0 get port lgre0 \
3 -- --id=@m create mirror name=m0 select-src-port=@p1 output-port=@lgre0 \
4 -- set bridge mtest mirrors=@m
```

HOST2测试:

```
1 # ip netns exec rmon tcpdump -ni m1 -v -e icmp or arp
2 tcpdump: listening on m1, link-type EN10MB (Ethernet), capture size 65535
  bytes
3 15:55:31.571555 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
  (0x0800), length 98: (tos 0x0, ttl 64, id 17776, offset 0, flags [DF],
  proto ICMP (1), length 84)
4     192.168.10.11 > 192.168.10.12: ICMP echo request, id 26594, seq 1,
  length 64
5 15:55:32.571376 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
  (0x0800), length 98: (tos 0x0, ttl 64, id 17777, offset 0, flags [DF],
  proto ICMP (1), length 84)
6     192.168.10.11 > 192.168.10.12: ICMP echo request, id 26594, seq 2,
  length 64
7 15:55:33.571340 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
  (0x0800), length 98: (tos 0x0, ttl 64, id 17778, offset 0, flags [DF],
  proto ICMP (1), length 84)
8     192.168.10.11 > 192.168.10.12: ICMP echo request, id 26594, seq 3,
  length 64
9 15:55:34.571304 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
  (0x0800), length 98: (tos 0x0, ttl 64, id 17779, offset 0, flags [DF],
  proto ICMP (1), length 84)
10    192.168.10.11 > 192.168.10.12: ICMP echo request, id 26594, seq 4,
  length 64
11 15:55:35.571292 06:7c:a0:69:97:f2 > c6:94:59:4b:7a:34, ethertype IPv4
  (0x0800), length 98: (tos 0x0, ttl 64, id 17780, offset 0, flags [DF],
  proto ICMP (1), length 84)
```

## 参考链接

<http://10.10.3.15:8090/pages/viewpage.action?pageId=7906835>

<http://10.10.3.15:8090/pages/viewpage.action?pageId=7918459>

<http://10.10.3.15:8090/display/YHT/11-OVS+MIRROR>