计算机网络实验五

动态路由协议 RIP，OSPF，和 BGP 观察

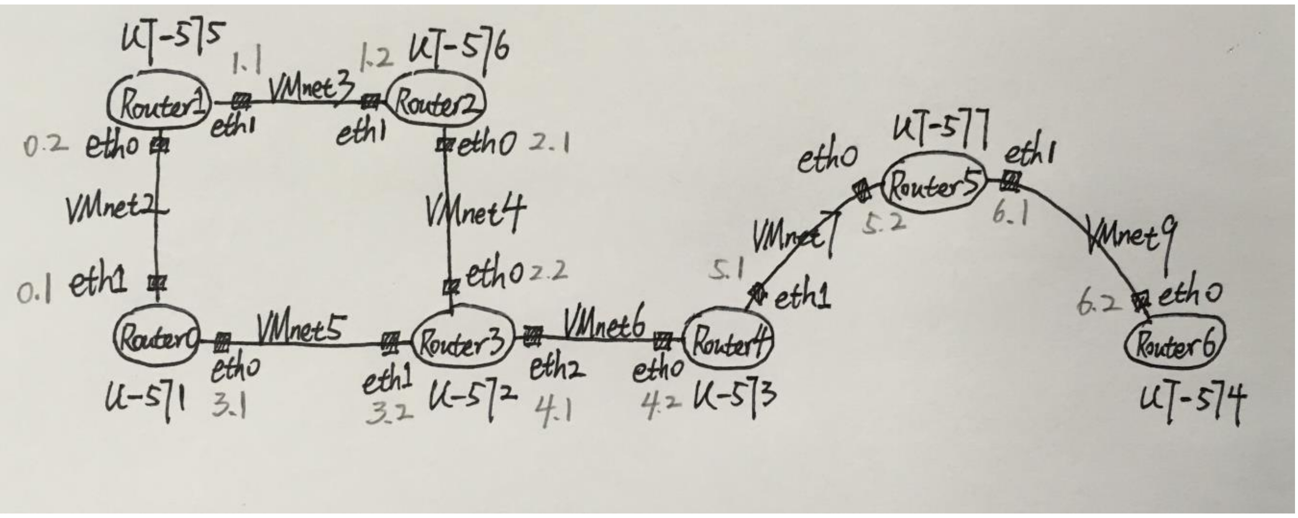
邱梓豪

141130077

1.实验目的

理解自治系统（AS），观察 RIP，OSPF 以及 BGP 动态路由协议的实际运行过程。在网络拓扑结构变更的情况下观察路由表的动态变更，通过实验理解动态路由选择算法。

2.网络拓扑结构



节点名	虚拟设备名	IP	Netmask
Router0	U-571	eth0: 192.168.3.1	255.255.255.0
		eth1: 192.168.0.1	255.255.255.0
Router1	UT-575	eth0: 192.168.0.2	255.255.255.0
		eth1: 192.168.1.1	255.255.255.0
Router2	UT-576	eth0: 192.168.2.1	255.255.255.0
		eth1: 192.168.1.2	255.255.255.0
Router3	U-572	eth0: 192.168.2.2	255.255.255.0
		eth1: 192.168.3.2	255.255.255.0
		eth2: 192.168.4.1	255.255.255.0
Router4	U-573	eth0: 192.168.4.2	255.255.255.0
		eth1: 192.168.5.1	255.255.255.0
Router5	UT-577	eth0: 192.168.5.2	255.255.255.0
		eth1: 192.168.6.1	255.255.255.0
Router6	UT-574	eth0: 192.168.6.2	255.255.255.0

3.IP 及路由器路由规则设置

为了使路由器彼此之间可以 ping 通，要为每块网卡设置 ip，并且要为路由器设置路由转发规则。

对左半边的 AS，设置如下：

（1）设置 ip
（2）Echo 1 > /proc/sys/net/ipv4/ip_forward
（3）设置路由规则：

```
Router0(571): sudo ip route add 192.168.2.0/24 via 192.168.0.2
              sudo ip route add 192.168.6.0/24 via 192.168.0.2
Router1(575): sudo ip route add 192.168.2.0/24 via 192.168.1.2
              sudo ip route add 192.168.6.0/24 via 192.168.1.2
              sudo ip route add 192.168.1.0/24 via 192.168.1.2
              sudo ip route add 192.168.0.0/24 via 192.168.0.1
Router2(576) sudo ip route add 192.168.2.0/24 via 192.168.2.2
              sudo ip route add 192.168.6.0/24 via 192.168.2.2
              sudo ip route add 192.168.0.0/24 via 192.168.1.1
Router3(572): sudo ip route add 192.168.0.0/24 via 192.168.2.1
              sudo ip route add 192.168.6.0/24 via 192.168.4.2
```

对右半边的 AS，设置如下：

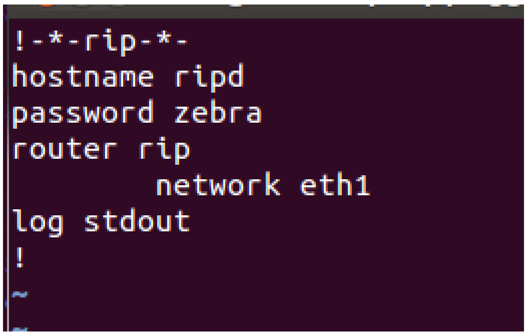
（1）设置 ip
（2）Echo 1 > /proc/sys/net/ipv4/ip_forward
（3）设置路由规则

```
Router4(573): sudo ip route add 192.168.6.0/24 via 192.168.5.2
             sudo ip route add 192.168.0.0/24 via 192.168.4.1
Router5(577): sudo ip route add 192.168.6.0/24 via 192.168.6.2
              sudo ip route add 192.168.0.0/24 via 192.168.5.1
             sudo ip route add 192.168.5.0/24 via 192.168.5.1
Router6(574): sudo ip route add 192.168.0.0/24 via 192.168.6.1
             sudo ip route add 192.168.5.0/24 via 192.168.6.1
```

4.配置文件的设置及数据包的抓取

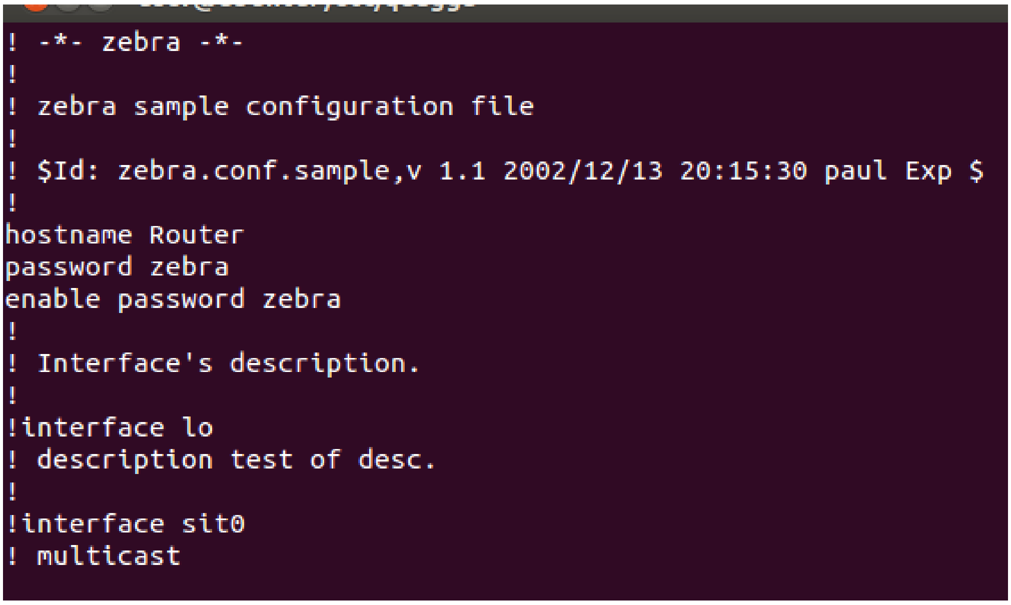
（1）RIP 协议观察

在 router0 和 router3 上配置 ripd.conf，内容如下：



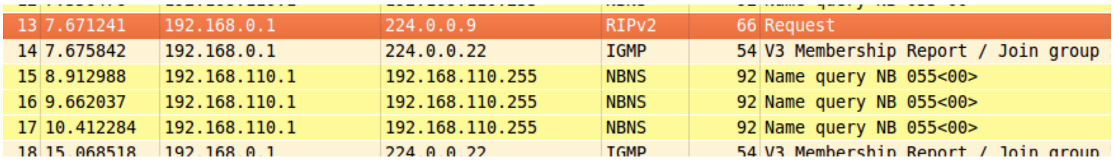
```
!-*-rip-*-
hostname ripd
password zebra
router rip
        network eth1
log stdout
!
~
```

在 router0 和 router3 上配置 zebra.conf，内容如下：



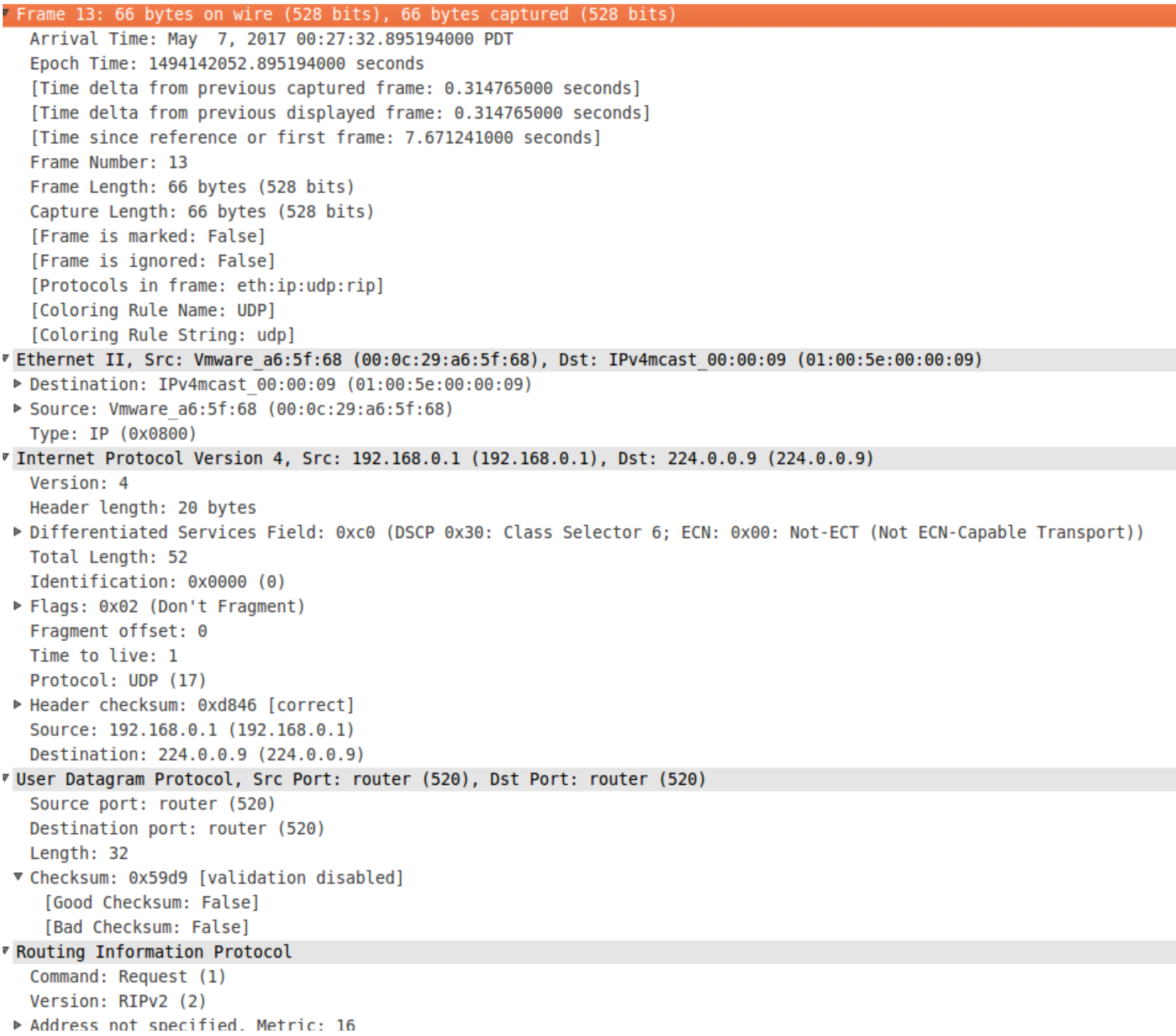
```
! -*- zebra -*-
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
!interface lo
! description test of desc.
!
!interface sit0
! multicast
```

配置完成后，启动 zebra 和 ripd 两个进程，通过 wireshark 抓取到的报文如下：



```
13 7.671241   192.168.0.1     224.0.0.9        RIPv2   66 Request
14 7.675842   192.168.0.1     224.0.0.22       IGMP    54 V3 Membership Report / Join group
15 8.912988   192.168.110.1   192.168.110.255  NBNS    92 Name query NB 055<00>
16 9.662037   192.168.110.1   192.168.110.255  NBNS    92 Name query NB 055<00>
17 10.412284  192.168.110.1   192.168.110.255  NBNS    92 Name query NB 055<00>
18 15.068518  192.168.0.1     224.0.0.22       IGMP    54 V3 Membership Report / Join group
```

该 RIP 报文的具体内容如下：

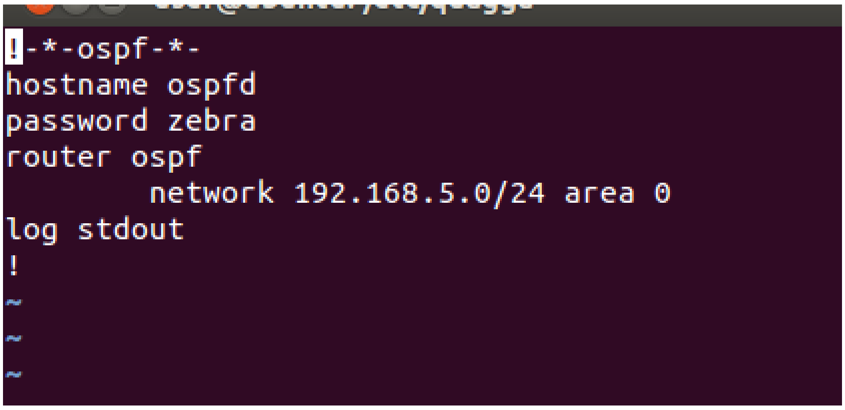


```
▾ Frame 13: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
    Arrival Time: May  7, 2017 00:27:32.895194000 PDT
    Epoch Time: 1494142052.895194000 seconds
    [Time delta from previous captured frame: 0.314765000 seconds]
    [Time delta from previous displayed frame: 0.314765000 seconds]
    [Time since reference or first frame: 7.671241000 seconds]
    Frame Number: 13
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:rip]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
▾ Ethernet II, Src: Vmware_a6:5f:68 (00:0c:29:a6:5f:68), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)
  ▸ Destination: IPv4mcast_00:00:09 (01:00:5e:00:00:09)
  ▸ Source: Vmware_a6:5f:68 (00:0c:29:a6:5f:68)
    Type: IP (0x0800)
▾ Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1), Dst: 224.0.0.9 (224.0.0.9)
    Version: 4
    Header length: 20 bytes
  ▸ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 52
    Identification: 0x0000 (0)
  ▸ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 1
    Protocol: UDP (17)
  ▸ Header checksum: 0xd846 [correct]
    Source: 192.168.0.1 (192.168.0.1)
    Destination: 224.0.0.9 (224.0.0.9)
▾ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
    Source port: router (520)
    Destination port: router (520)
    Length: 32
  ▾ Checksum: 0x59d9 [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
▾ Routing Information Protocol
    Command: Request (1)
    Version: RIPv2 (2)
  ▸ Address not specified, Metric: 16
```

从报文中可以看出 RIP 报文是包装在 UDP 数据报中的，UDP 的源端口和目的端口都是 router；目的 IP 是 224.0.0.9，这正是 RIPv2 的组播 IP。

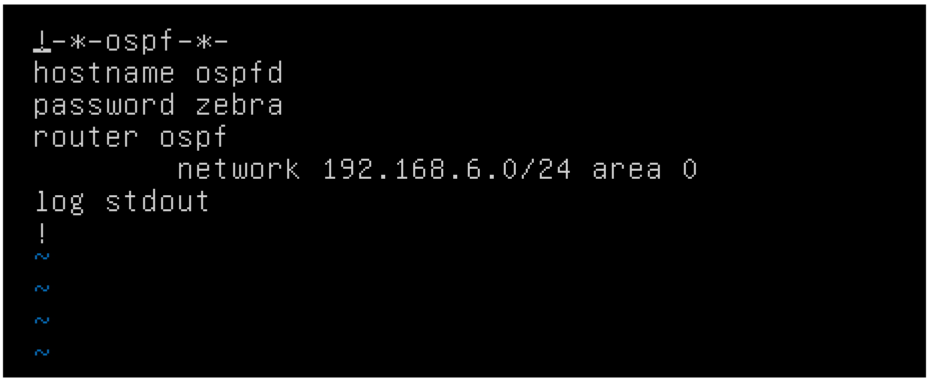
（2）OSPF 协议观察

在 router4 中配置 ospfd.conf 文件如下：



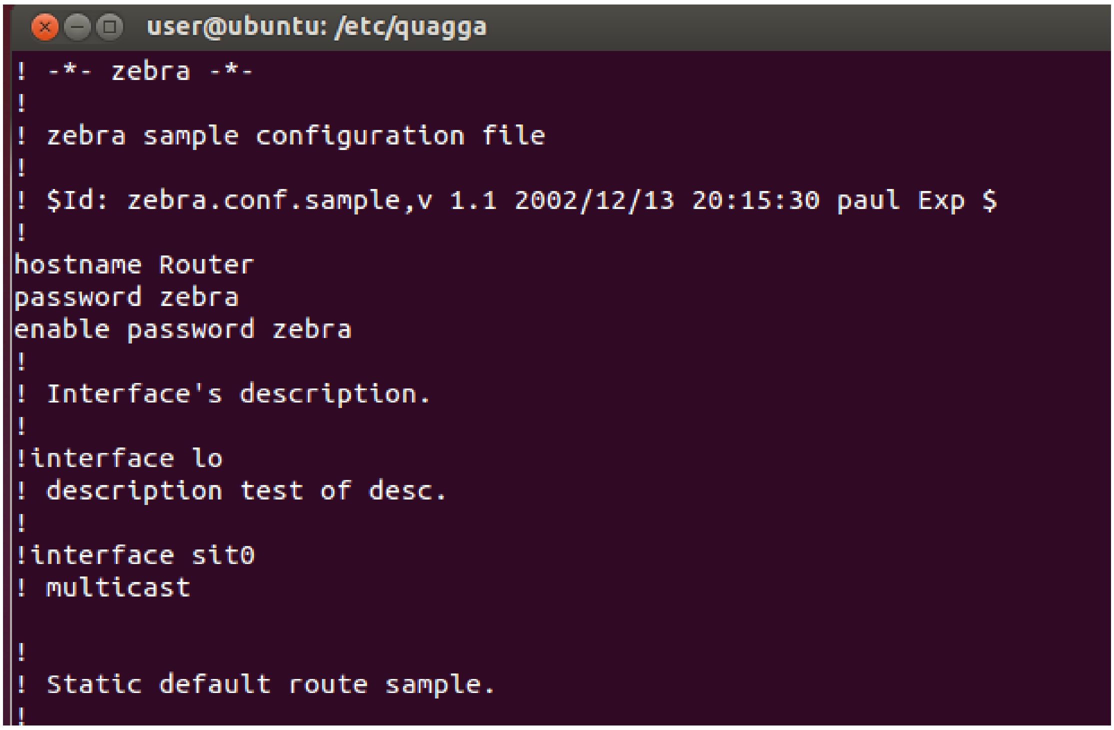
```
!-*-ospf-*-
hostname ospfd
password zebra
router ospf
        network 192.168.5.0/24 area 0
log stdout
!
~
~
~
```

在 router6 中配置 ospfd.conf 文件如下：



```
!-*-ospf-*-
hostname ospfd
password zebra
router ospf
        network 192.168.6.0/24 area 0
log stdout
!
~
~
~
~
```

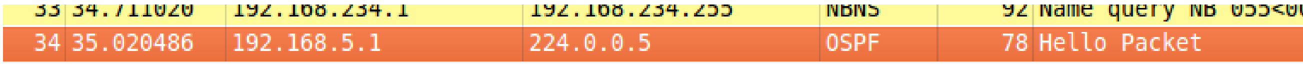
在 router4 和 router6 上配置 zebra.conf，内容如下：



user@ubuntu: /etc/quagga

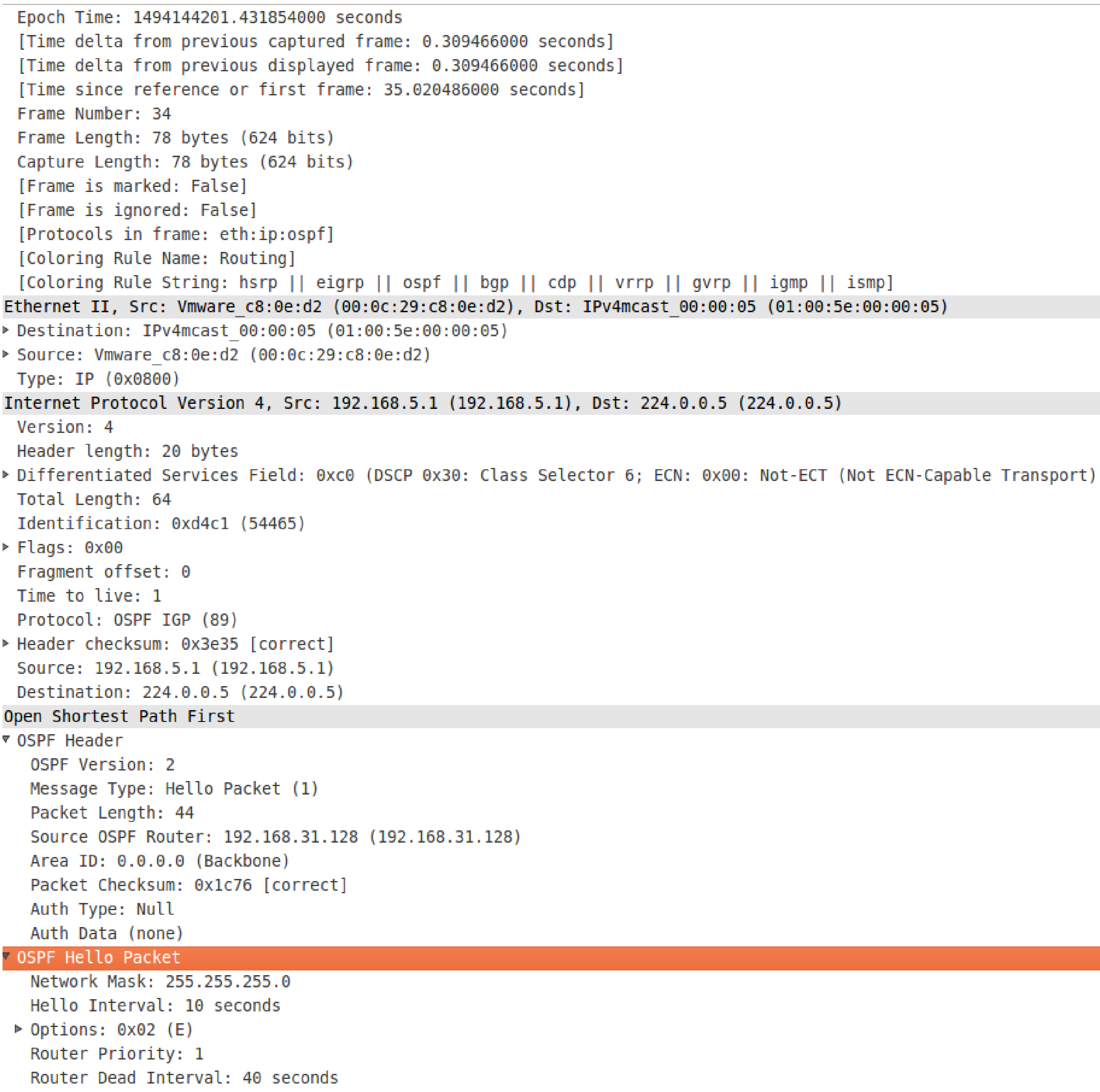
```
! -*- zebra -*-
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
!interface lo
! description test of desc.
!
!interface sit0
! multicast

!
! Static default route sample.
!
```

配置完成后，启动 zebra 和 ospfd 两个进程，通过 wireshark 抓取到的报文如下：



```
33 34.711020  192.168.234.1   192.168.234.255  NBNS   92 Name query NB 055<00
34 35.020486  192.168.5.1     224.0.0.5        OSPF   78 Hello Packet
```

具体的报文如下：



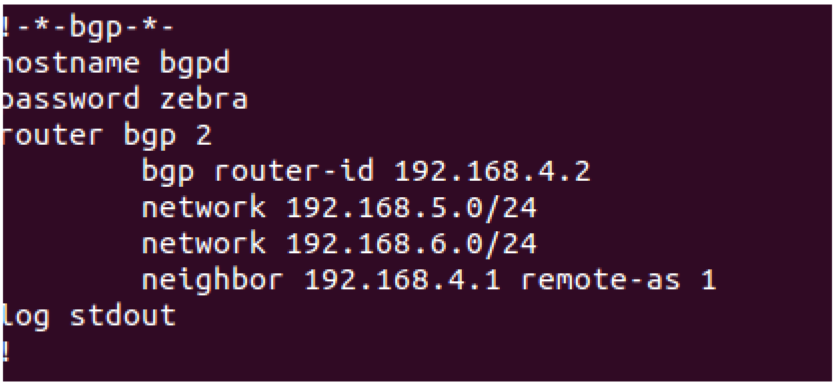
```
  Epoch Time: 1494144201.431854000 seconds
  [Time delta from previous captured frame: 0.309466000 seconds]
  [Time delta from previous displayed frame: 0.309466000 seconds]
  [Time since reference or first frame: 35.020486000 seconds]
  Frame Number: 34
  Frame Length: 78 bytes (624 bits)
  Capture Length: 78 bytes (624 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:ospf]
  [Coloring Rule Name: Routing]
  [Coloring Rule String: hsrp || eigrp || ospf || bgp || cdp || vrrp || gvrp || igmp || ismp]
Ethernet II, Src: Vmware_c8:0e:d2 (00:0c:29:c8:0e:d2), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)
▸ Destination: IPv4mcast_00:00:05 (01:00:5e:00:00:05)
▸ Source: Vmware_c8:0e:d2 (00:0c:29:c8:0e:d2)
  Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.5.1 (192.168.5.1), Dst: 224.0.0.5 (224.0.0.5)
  Version: 4
  Header length: 20 bytes
▸ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)
  Total Length: 64
  Identification: 0xd4c1 (54465)
▸ Flags: 0x00
  Fragment offset: 0
  Time to live: 1
  Protocol: OSPF IGP (89)
▸ Header checksum: 0x3e35 [correct]
  Source: 192.168.5.1 (192.168.5.1)
  Destination: 224.0.0.5 (224.0.0.5)
Open Shortest Path First
▾ OSPF Header
    OSPF Version: 2
    Message Type: Hello Packet (1)
    Packet Length: 44
    Source OSPF Router: 192.168.31.128 (192.168.31.128)
    Area ID: 0.0.0.0 (Backbone)
    Packet Checksum: 0x1c76 [correct]
    Auth Type: Null
    Auth Data (none)
▾ OSPF Hello Packet
    Network Mask: 255.255.255.0
    Hello Interval: 10 seconds
  ▸ Options: 0x02 (E)
    Router Priority: 1
    Router Dead Interval: 40 seconds
```

从报文中可以看出 RIP 报文是包装在 OSPF 数据报中的，目的 IP 是 224.0.0.5，224.0.0.5 指代在任意网络中所有运行 OSPF 进程的接口都属于该组，于是接收所有 224.0.0.5 的组播数据包。Hello Interval 是 10s，表示每 10s

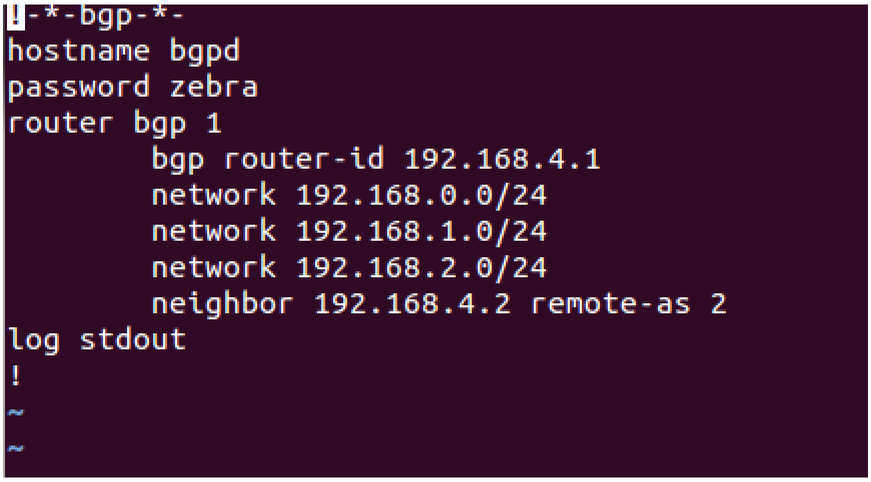
发送一次 Hello Packet；Router Dead Interval 是 40s，表示如果 40s 没有收到某路由器的响应，表示其已断开连接。

（3）BGP 协议观察

在 router3 和 router4 中配置 bgpd.conf 配置文件如下：

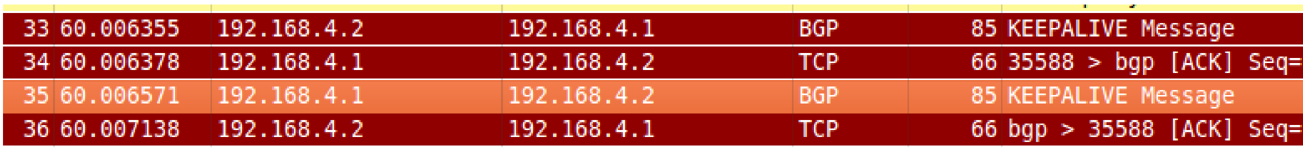


```
!-*-bgp-*-
hostname bgpd
password zebra
router bgp 2
        bgp router-id 192.168.4.2
        network 192.168.5.0/24
        network 192.168.6.0/24
        neighbor 192.168.4.1 remote-as 1
log stdout
!
```



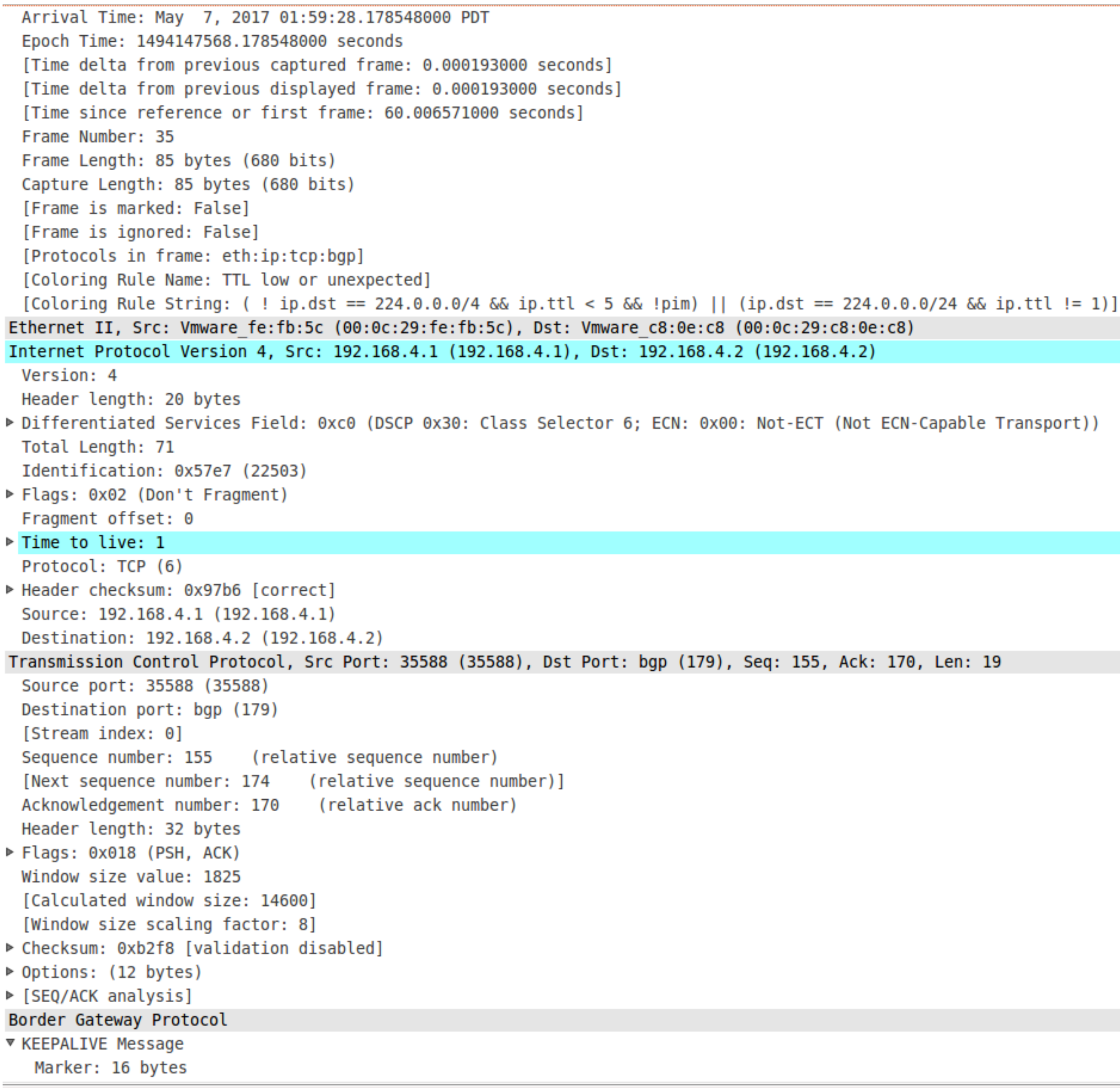
```
!-*-bgp-*-
hostname bgpd
password zebra
router bgp 1
        bgp router-id 192.168.4.1
        network 192.168.0.0/24
        network 192.168.1.0/24
        network 192.168.2.0/24
        neighbor 192.168.4.2 remote-as 2
log stdout
!
~
~
```

启动 bgpd 进程，可以抓取到 BGP 报文：



```
33 60.006355  192.168.4.2   192.168.4.1   BGP   85 KEEPALIVE Message
34 60.006378  192.168.4.1   192.168.4.2   TCP   66 35588 > bgp [ACK] Seq=
35 60.006571  192.168.4.1   192.168.4.2   BGP   85 KEEPALIVE Message
36 60.007138  192.168.4.2   192.168.4.1   TCP   66 bgp > 35588 [ACK] Seq=
```

BGP 报文如下

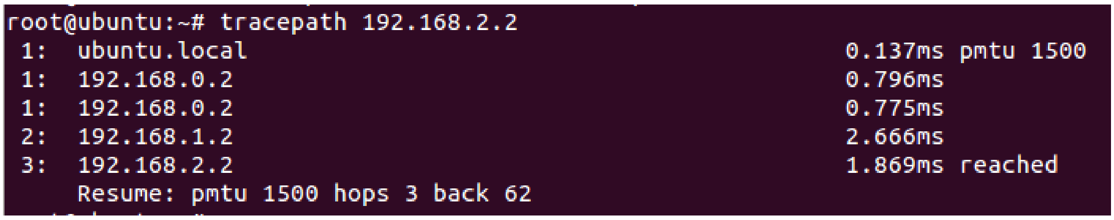


```
  Arrival Time: May  7, 2017 01:59:28.178548000 PDT
  Epoch Time: 1494147568.178548000 seconds
  [Time delta from previous captured frame: 0.000193000 seconds]
  [Time delta from previous displayed frame: 0.000193000 seconds]
  [Time since reference or first frame: 60.006571000 seconds]
  Frame Number: 35
  Frame Length: 85 bytes (680 bits)
  Capture Length: 85 bytes (680 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:tcp:bgp]
  [Coloring Rule Name: TTL low or unexpected]
  [Coloring Rule String: ( ! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim) || (ip.dst == 224.0.0.0/24 && ip.ttl != 1)]
Ethernet II, Src: Vmware_fe:fb:5c (00:0c:29:fe:fb:5c), Dst: Vmware_c8:0e:c8 (00:0c:29:c8:0e:c8)
Internet Protocol Version 4, Src: 192.168.4.1 (192.168.4.1), Dst: 192.168.4.2 (192.168.4.2)
  Version: 4
  Header length: 20 bytes
▶ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 71
  Identification: 0x57e7 (22503)
▶ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
▶ Time to live: 1
  Protocol: TCP (6)
▶ Header checksum: 0x97b6 [correct]
  Source: 192.168.4.1 (192.168.4.1)
  Destination: 192.168.4.2 (192.168.4.2)
Transmission Control Protocol, Src Port: 35588 (35588), Dst Port: bgp (179), Seq: 155, Ack: 170, Len: 19
  Source port: 35588 (35588)
  Destination port: bgp (179)
  [Stream index: 0]
  Sequence number: 155    (relative sequence number)
  [Next sequence number: 174    (relative sequence number)]
  Acknowledgement number: 170    (relative ack number)
  Header length: 32 bytes
▶ Flags: 0x018 (PSH, ACK)
  Window size value: 1825
  [Calculated window size: 14600]
  [Window size scaling factor: 8]
▶ Checksum: 0xb2f8 [validation disabled]
▶ Options: (12 bytes)
▶ [SEQ/ACK analysis]
Border Gateway Protocol
▼ KEEPALIVE Message
    Marker: 16 bytes
```

从报文中可以看出BGP报文是包装在TCP数据报中的，目的端口都是179（BGP），源IP和目的IP是边界网关路由器的IP（192.168.4.1和192.168.4.2）。

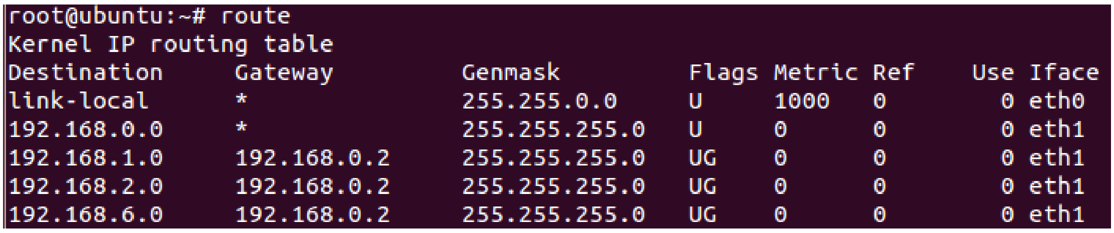
（4）观察路由表动态变更

首先追踪router0到router3的包传输路径：



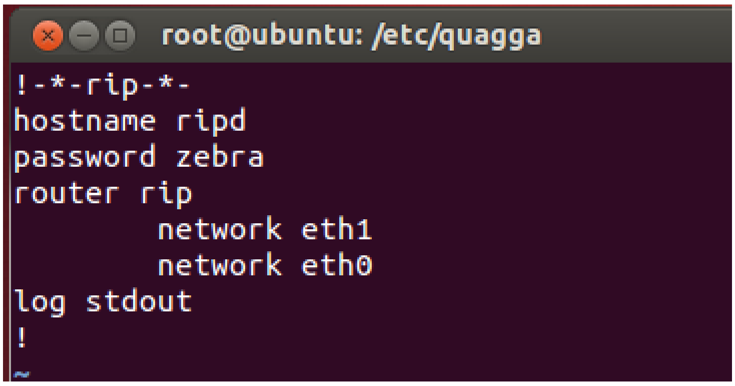
```
root@ubuntu:~# tracepath 192.168.2.2
 1:  ubuntu.local                                          0.137ms pmtu 1500
 1:  192.168.0.2                                           0.796ms
 1:  192.168.0.2                                           0.775ms
 2:  192.168.1.2                                           2.666ms
 3:  192.168.2.2                                           1.869ms reached
     Resume: pmtu 1500 hops 3 back 62
```

此时利用 route 命令，察看路由表情况如下：

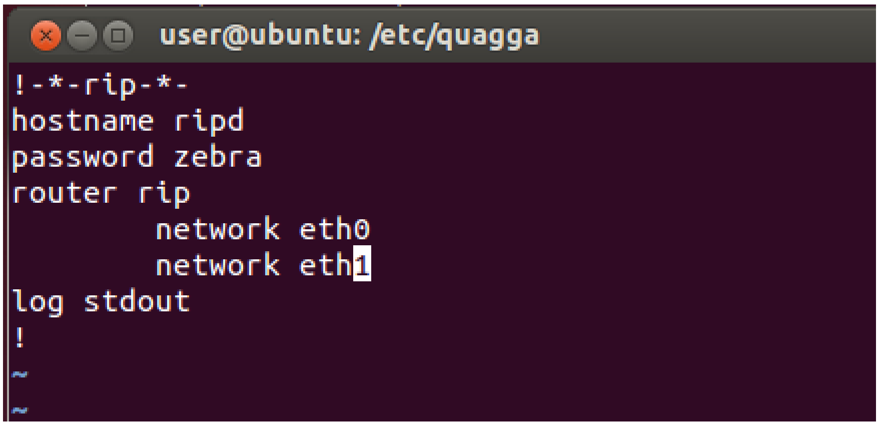


```
root@ubuntu:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
link-local      *               255.255.0.0     U     1000   0        0 eth0
192.168.0.0     *               255.255.255.0   U     0      0        0 eth1
192.168.1.0     192.168.0.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     192.168.0.2     255.255.255.0   UG    0      0        0 eth1
192.168.6.0     192.168.0.2     255.255.255.0   UG    0      0        0 eth1
```

在 router0 和 router3 之间添加一条连接，为新添加的网卡设置 ip 及相应路由规则，重新配置 router0 和 router3 的 ripd.conf 文件如下：

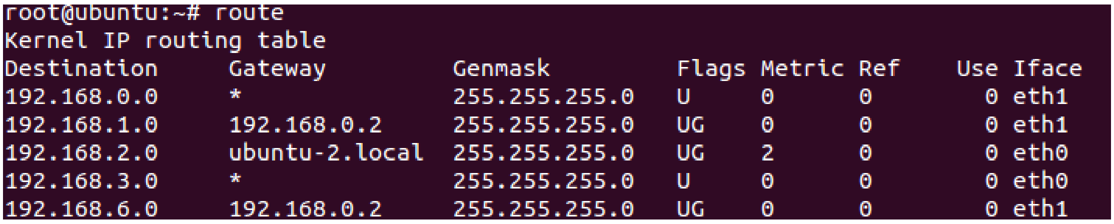


```
!-*-rip-*-
hostname ripd
password zebra
router rip
        network eth1
        network eth0
log stdout
!
~
```



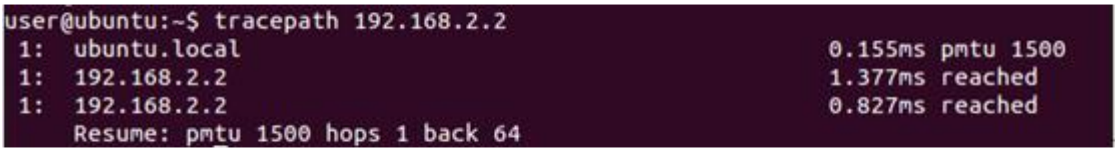
```
!-*-rip-*-
hostname ripd
password zebra
router rip
        network eth0
        network eth1
log stdout
!
~
~
```

之后重启 quagga，观察 router0 的路由表如下：



```
root@ubuntu:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     *               255.255.255.0   U     0      0        0 eth1
192.168.1.0     192.168.0.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     ubuntu-2.local  255.255.255.0   UG    2      0        0 eth0
192.168.3.0     *               255.255.255.0   U     0      0        0 eth0
192.168.6.0     192.168.0.2     255.255.255.0   UG    0      0        0 eth1
```

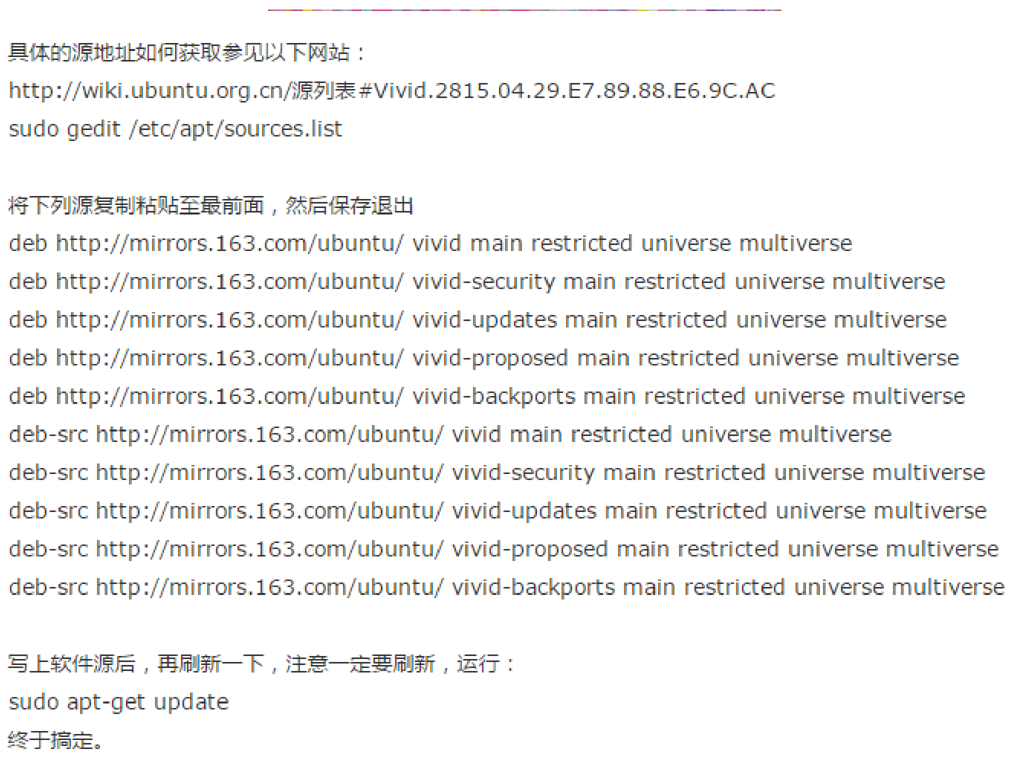
再次追踪 router0 到 router3 的包传输路径，如下：



```
user@ubuntu:~$ tracepath 192.168.2.2
 1:  ubuntu.local                                          0.155ms pmtu 1500
 1:  192.168.2.2                                           1.377ms reached
 1:  192.168.2.2                                           0.827ms reached
     Resume: pmtu 1500 hops 1 back 64
```

5.实验中遇到的问题

实验开始时我发现 quagga 包无法正常下载，百度之后我发现可以通过更新 apt 的软件源来解决这个问题。



具体的源地址如何获取参见以下网站：
http://wiki.ubuntu.org.cn/源列表#Vivid.2815.04.29.E7.89.88.E6.9C.AC
sudo gedit /etc/apt/sources.list

将下列源复制粘贴至最前面，然后保存退出

```
deb http://mirrors.163.com/ubuntu/ vivid main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ vivid-security main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ vivid-updates main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ vivid-proposed main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ vivid-backports main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ vivid main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ vivid-security main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ vivid-updates main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ vivid-proposed main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ vivid-backports main restricted universe multiverse
```

写上软件源后，再刷新一下，注意一定要刷新，运行：
sudo apt-get update
终于搞定。