

Lab1 实验报告

一、实验目的

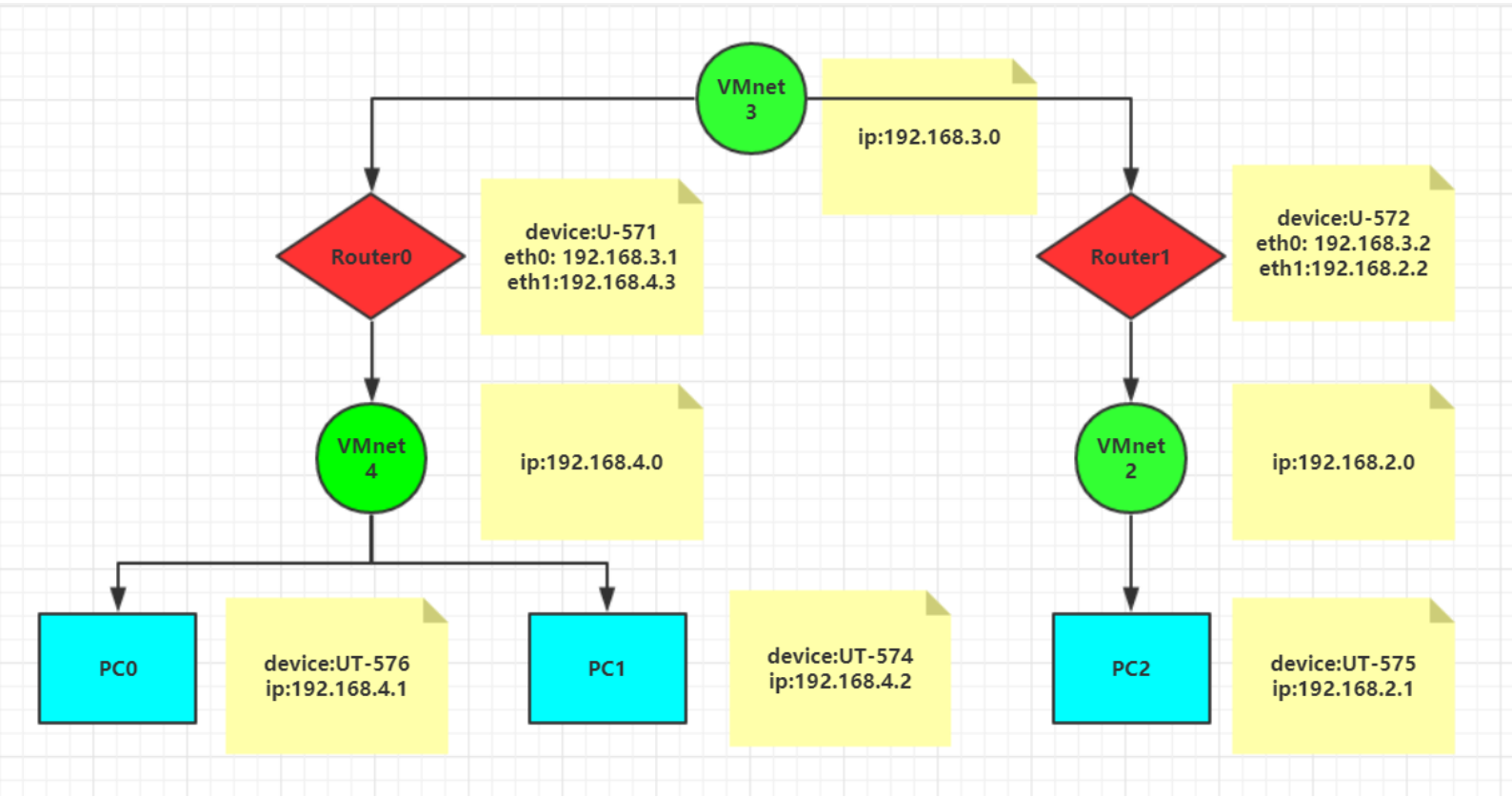
- 1.学会在 Linux 下用虚拟机搭建一个小型的可以通信的网络拓扑。
- 2.熟悉系统的常用网络工具集合（指令：ifconfig/ping/route 软件：wireshark）。
- 3.学会利用 wireshark 抓包，熟悉观察和分析协议数据单元 PDU 的含义。

二、网络拓扑配置

表：

节点名	虚拟设备名	ip	netmask
Router0	U-571	eth0: 192.168.3.1	255.255.255.0
		eth1: 192.168.4.3	255.255.255.0
Router1	U-572	eth0: 192.168.3.2	255.255.255.0
		eth1: 192.168.2.2	255.255.255.0
PC0	UT-576	192.168.4.1	255.255.255.0
PC1	UT-574	192.168.4.2	255.255.255.0
PC2	UT-575	192.168.2.1	255.255.255.0

图：

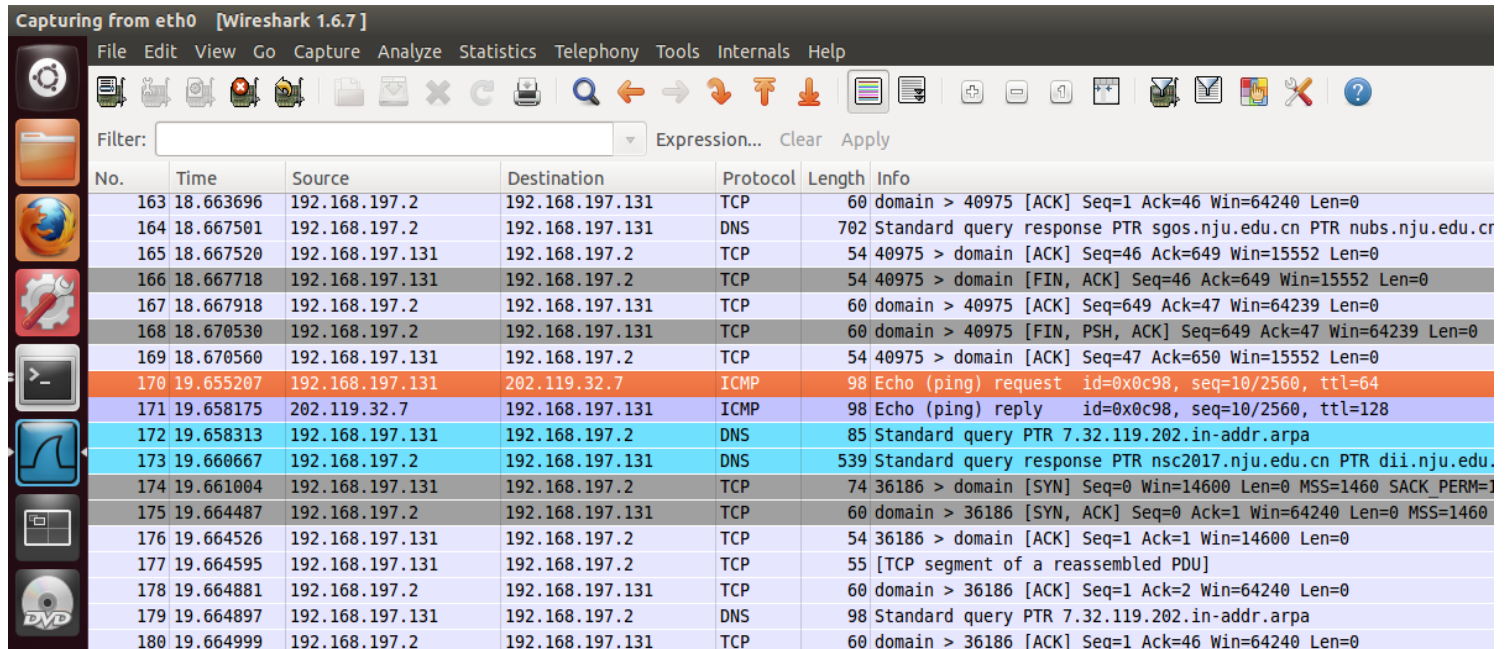


三、 路由规则配置：

Router0	
设计	发往子网 VMnet2(192.168.2.0)的数据包, 将通过 Router1 的 eth0(192.168.3.2) 发送出去。 发往子网 VMnet3(192.168.3.0)的数据包, 将通过 Router0 的 eth0(192.168.3.1) 发送出去。 发往子网 VMnet4(192.168.4.0)的数据包, 将通过 Router0 的 eth1(192.168.4.3) 发送出去。
命令	ifconfig eth0 192.168.3.1 netmask 255.255.255.0 ifconfig eth1 192.168.4.3 netmask 255.255.255.0 ip route add 192.168.2.0/24 via 192.168.3.2 ip route add 192.168.3.0/24 via 192.168.3.1 ip route add 192.168.4.0/24 via 192.168.4.3 echo 1 > /proc/sys/net/ipv4/ip_forward
Router1	
设计	发往子网 VMnet2(192.168.2.0)的数据包, 将通过 Router1 的 eth1(192.168.2.2) 发送出去。 发往子网 VMnet3(192.168.3.0)的数据包, 将通过 Router1 的 eth0(192.168.3.2) 发送出去。 发往子网 VMnet4(192.168.4.0)的数据包, 将通过 Router0 的 eth0(192.168.3.1) 发送出去。
命令	ifconfig eth0 192.168.3.2 netmask 255.255.255.0 ifconfig eth1 192.168.2.2 netmask 255.255.255.0 ip route add 192.168.2.0/24 via 192.168.2.2 ip route add 192.168.3.0/24 via 192.168.3.2 ip route add 192.168.4.0/24 via 192.168.3.1 echo 1 > /proc/sys/net/ipv4/ip_forward
PC0	
设计	将 数 据 包 发 往 子 网 VMnet4(192.168.4.0) 中 的 路 由 器 Router0 的 eth1(192.168.4.3)
命令	ifconfig eth0 192.168.4.1 netmask 255.255.255.0 route add default gw 192.168.4.3
PC1	
设计	将 数 据 包 发 往 子 网 VMnet4(192.168.4.0) 中 的 路 由 器 Router0 的 eth1(192.168.4.3)
命令	ifconfig eth0 192.168.4.2 netmask 255.255.255.0 route add default gw 192.168.4.3
PC2	
设计	将 数 据 包 发 往 子 网 VMnet2(192.168.2.0) 中 的 路 由 器 Router1 的 eth1(192.168.2.2)
命令	ifconfig eth0 192.168.2.1 netmask 255.255.255.0 route add default gw 192.168.2.2

四、数据包截图及协议报文分析

①ping 系主页 cs.nju.edu.cn

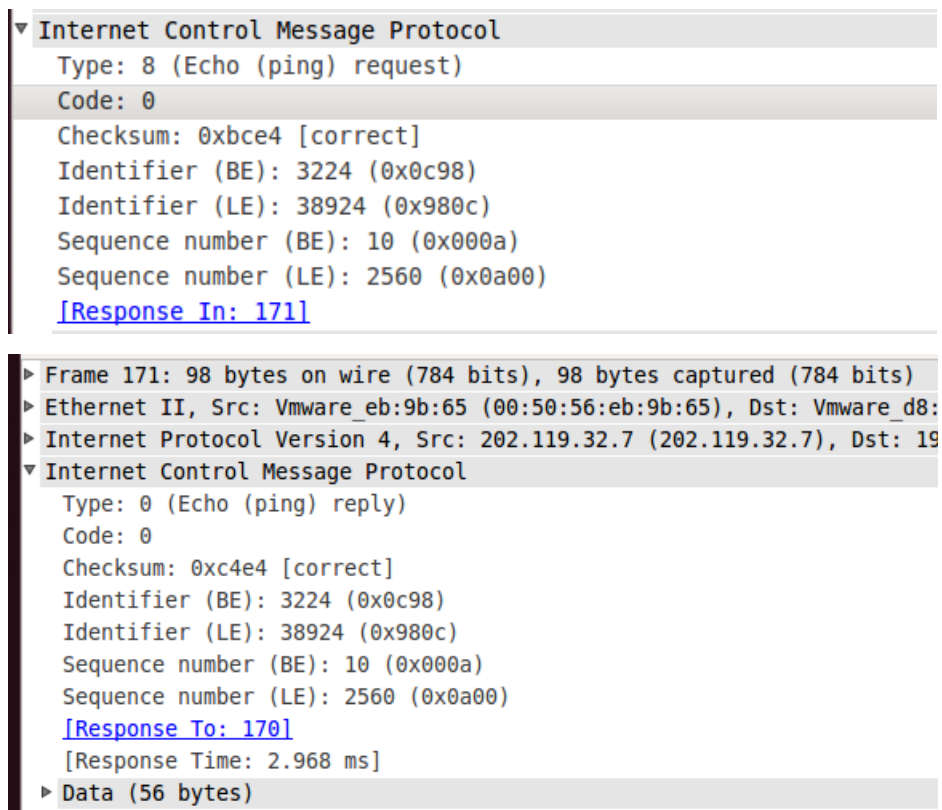


No.	Time	Source	Destination	Protocol	Length	Info
163	18.663696	192.168.197.2	192.168.197.131	TCP	60	domain > 40975 [ACK] Seq=1 Ack=46 Win=64240 Len=0
164	18.667501	192.168.197.2	192.168.197.131	DNS	702	Standard query response PTR sgos.nju.edu.cn PTR nubs.nju.edu.cn
165	18.667520	192.168.197.131	192.168.197.2	TCP	54	40975 > domain [ACK] Seq=46 Ack=649 Win=15552 Len=0
166	18.667718	192.168.197.131	192.168.197.2	TCP	54	40975 > domain [FIN, ACK] Seq=46 Ack=649 Win=15552 Len=0
167	18.667918	192.168.197.2	192.168.197.131	TCP	60	domain > 40975 [ACK] Seq=649 Ack=47 Win=64239 Len=0
168	18.670530	192.168.197.2	192.168.197.131	TCP	60	domain > 40975 [FIN, PSH, ACK] Seq=649 Ack=47 Win=64239 Len=0
169	18.670560	192.168.197.131	192.168.197.2	TCP	54	40975 > domain [ACK] Seq=47 Ack=650 Win=15552 Len=0
170	19.655207	192.168.197.131	202.119.32.7	ICMP	98	Echo (ping) request id=0x0c98, seq=10/2560, ttl=64
171	19.658175	202.119.32.7	192.168.197.131	ICMP	98	Echo (ping) reply id=0x0c98, seq=10/2560, ttl=128
172	19.658313	192.168.197.131	192.168.197.2	DNS	85	Standard query PTR 7.32.119.202.in-addr.arpa
173	19.660667	192.168.197.2	192.168.197.131	DNS	539	Standard query response PTR nsc2017.nju.edu.cn PTR dii.nju.edu.cn
174	19.661004	192.168.197.131	192.168.197.2	TCP	74	36186 > domain [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1
175	19.664487	192.168.197.2	192.168.197.131	TCP	60	domain > 36186 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
176	19.664526	192.168.197.131	192.168.197.2	TCP	54	36186 > domain [ACK] Seq=1 Ack=1 Win=14600 Len=0
177	19.664595	192.168.197.131	192.168.197.2	TCP	55	[TCP segment of a reassembled PDU]
178	19.664881	192.168.197.2	192.168.197.131	TCP	60	domain > 36186 [ACK] Seq=1 Ack=2 Win=64240 Len=0
179	19.664897	192.168.197.131	192.168.197.2	DNS	98	Standard query PTR 7.32.119.202.in-addr.arpa
180	19.664999	192.168.197.2	192.168.197.131	TCP	60	domain > 36186 [ACK] Seq=1 Ack=46 Win=64240 Len=0

选取 170 和 171 这两个数据包分析。

170	19.655207	192.168.197.131	202.119.32.7	ICMP	98	Echo (ping) request	id=0x0c98, seq=10/2560, ttl=64
171	19.658175	202.119.32.7	192.168.197.131	ICMP	98	Echo (ping) reply	id=0x0c98, seq=10/2560, ttl=128

从图中可以看出 ping 系主页使用的协议是 ICMP，发出 request 请求的时候，是源地址 192.168.197.131 到目的地址 202.119.32.7，信息的长度是 98。受到 reply 回复的时候，是源地址 202.119.32.7 到目的地址 192.168.197.131，信息的长度也是 98。后面的 id 是该数据包的标识,seq 是发送的频率。



▼ Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xbce4 [correct]
Identifier (BE): 3224 (0x0c98)
Identifier (LE): 38924 (0x980c)
Sequence number (BE): 10 (0x000a)
Sequence number (LE): 2560 (0x0a00)
[Response In: 171]

▶ Frame 171: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▶ Ethernet II, Src: Vmware_eb:9b:65 (00:50:56:eb:9b:65), Dst: Vmware_d8:1b:15:1d (00:50:56:00:00:00)
▶ Internet Protocol Version 4, Src: 202.119.32.7 (202.119.32.7), Dst: 192.168.197.131
▼ Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0xc4e4 [correct]
Identifier (BE): 3224 (0x0c98)
Identifier (LE): 38924 (0x980c)
Sequence number (BE): 10 (0x000a)
Sequence number (LE): 2560 (0x0a00)
[Response To: 170]
[Response Time: 2.968 ms]
▶ Data (56 bytes)

Type 的值指示了是 request 还是 reply, request 是 8, 而 reply 是 0。下面的 response in 是指向请求目的地的数据包标号 171, 相应地 response to 是指向回复目的地的数据包标号 170, 他们之间应该是成对的关系。

②浏览器打开 www.nju.edu.cn

No.	Time	Source	Destination	Protocol	Length	Info
1620	24.491699	182.61.248.48	192.168.197.132	TCP	60	http > 49367 [ACK] Seq=1 Ack=405 Win=64240 Len=0
1621	24.496280	182.61.248.48	192.168.197.132	TCP	60	http > 49368 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1622	24.496319	192.168.197.132	182.61.248.48	TCP	54	49368 > http [ACK] Seq=1 Ack=1 Win=14600 Len=0
1623	24.496342	182.61.248.48	192.168.197.132	TCP	60	http > 49369 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1624	24.496375	192.168.197.132	182.61.248.48	TCP	54	49369 > http [ACK] Seq=1 Ack=1 Win=14600 Len=0
1625	24.496596	192.168.197.132	182.61.248.48	HTTP	472	GET /static/api/js/view/select_view.js?v=14bb0f0f.js HTTP/1.1
1626	24.496836	182.61.248.48	192.168.197.132	TCP	60	http > 49368 [ACK] Seq=1 Ack=419 Win=64240 Len=0
1627	24.529210	182.61.248.48	192.168.197.132	HTTP	420	HTTP/1.1 304 Not Modified
1628	24.530532	182.61.248.48	192.168.197.132	HTTP	420	HTTP/1.1 304 Not Modified
1629	24.530541	192.168.197.132	182.61.248.48	TCP	54	49367 > http [ACK] Seq=405 Ack=367 Win=15544 Len=0
1630	24.538366	182.61.248.48	192.168.197.132	HTTP	420	HTTP/1.1 304 Not Modified
1631	24.538381	192.168.197.132	182.61.248.48	TCP	54	49368 > http [ACK] Seq=419 Ack=367 Win=15544 Len=0
1632	24.568064	192.168.197.132	182.61.248.48	TCP	54	49363 > http [ACK] Seq=1181 Ack=7310 Win=35040 Len=0
1633	24.670750	192.168.197.132	182.61.248.48	HTTP	467	GET /static/api/js/base/tangram.js?v=37768233.js HTTP/1.1
1634	24.670884	192.168.197.132	182.61.248.48	HTTP	456	GET /static/api/js/share/api_base.js HTTP/1.1
1635	24.671034	182.61.248.48	192.168.197.132	TCP	60	http > 49363 [ACK] Seq=7310 Ack=1594 Win=64240 Len=0
1636	24.671231	182.61.248.48	192.168.197.132	TCP	60	http > 49369 [ACK] Seq=1 Ack=403 Win=64240 Len=0
1637	24.680554	192.168.197.132	182.61.248.48	HTTP	456	GET /static/api/js/view/view_base.js HTTP/1.1
1638	24.681562	182.61.248.48	192.168.197.132	TCP	60	http > 49367 [ACK] Seq=367 Ack=807 Win=64240 Len=0
1639	24.709216	182.61.248.48	192.168.197.132	HTTP	418	HTTP/1.1 304 Not Modified

可以看出, 在浏览器中打开网页, 使用的全部是 HTTP 和 TCP 协议。和 ping 系主页不同的是, 这里的互相的通信并不是成对出现, 两边的通信是不对等的, 而且可以看出使用浏览器, 数据包的发送量会增加很多。

这里选取 HTTP 协议进行分析。

```
▶ Frame 1625: 472 bytes on wire (3776 bits), 472 bytes captured (3776 bits)
▶ Ethernet II, Src: Vmware_d8:7c:a6 (00:0c:29:d8:7c:a6), Dst: Vmware_eb:9b:65 (00:50:56:eb:9b:65)
▶ Internet Protocol Version 4, Src: 192.168.197.132 (192.168.197.132), Dst: 182.61.248.48 (182.61.248.48)
▶ Transmission Control Protocol, Src Port: 49368 (49368), Dst Port: http (80), Seq: 1, Ack: 1, Len: 418
▼ Hypertext Transfer Protocol
  ▶ GET /static/api/js/view/select_view.js?v=14bb0f0f.js HTTP/1.1\r\n
    Host: bdimg.share.baidu.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:11.0) Gecko/20100101 Firefox/11.0\r\n
    Accept: */*\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Cookie: BAIDUID=9BE2E234BD3AAC79A3A9CD339E915C23:FG=1\r\n
    If-Modified-Since: Mon, 28 Sep 2015 08:06:42 GMT\r\n
    If-None-Match: "3775481591"\r\n
  \r\n
  [Full request URI: http://bdimg.share.baidu.com/static/api/js/view/select_view.js?v=14bb0f0f.js]
```

从协议的内容来看, HTTP 协议主要用来获取网页的 URL 中的终端 (Host), 代理 (User-Agent), 可以使用的语言和编码 (Accept-Language 和 Accept-Encoding) 以及其他与网页配置有关的信息。