



警示

1. 实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2. 当次小组成员成绩只计学号、姓名登录在下表中的。
3. 在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4. 实验报告文件以 PDF 格式提交。

| | | | | | |
|------|--|----------|----------|-----------------------------|-----|
| 院系 | 计算机学院 | 班 级 | 软工1班 | 组长 | 崔子潇 |
| 学号 | 19308024 | 19335040 | 19335286 | | |
| 学生 | 崔子潇 | 丁维力 | 郑有为 | | |
| 实验分工 | | | | | |
| 崔子潇 | 配置 HTTP 服务器, 管理服务器(.15.5), 实验抓包 | | 丁维力 | 配置管理两台路由器, 搭建实验拓扑环境, 管理主机 A | |
| 郑有为 | 配置 HTTP 服务器, 管理服务器(.15.6), 实验抓包, 写实验报告 | | | | |

【实验题目】配置 TCP 负载分配。

【实验目的】

配置网络地址变换，使用一个单地址实现两台 WEB 服务器负载平衡。

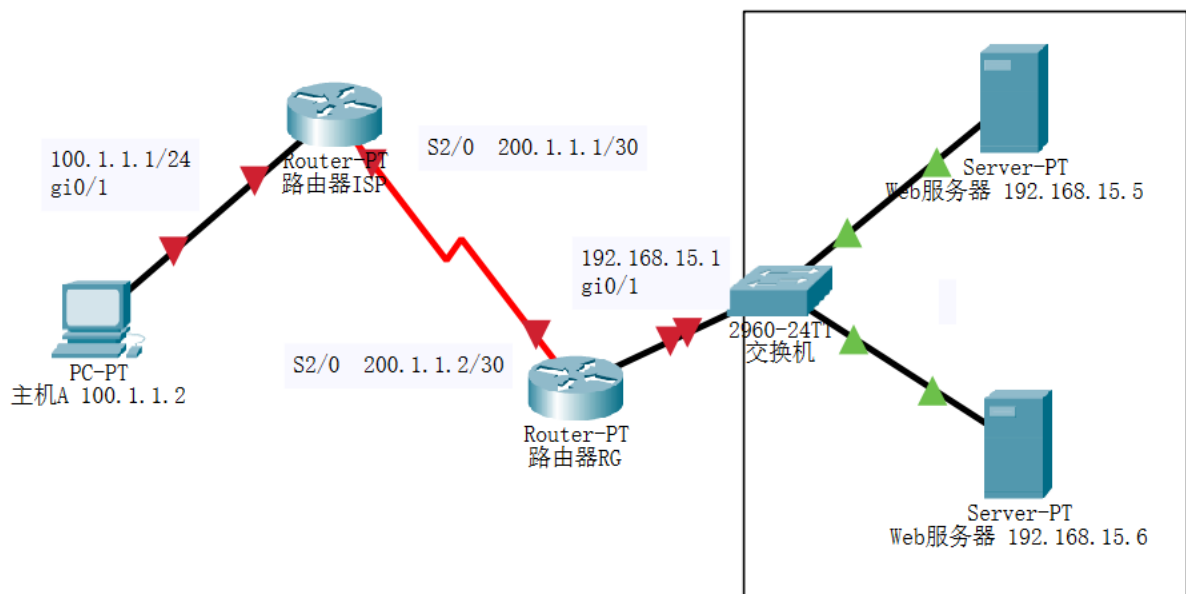
【实验内容】

1. 完成实验实 9-4 (P314)，注意步骤 0 和步骤 6。
2. 在进行验证时如果不用 Web，而改用 Telnet 或远程桌面连接，同样能验证吗？
3. 请回答 P317 的实验思考。

【实验要求】

重要信息需给出截图，注意实验步骤的前后对比。

【实验记录】(如有实验拓扑请自行画出)



(TCP 负载均衡实验拓扑图)

步骤 1:

(1) 搭建 Web 服务器

首先，下载 Apache 并解压，下载网站 <https://apachelounge.com/download>
然后，修改.conf 文件配置和 index.html，修改 SRVROOT 和增加 ServerName



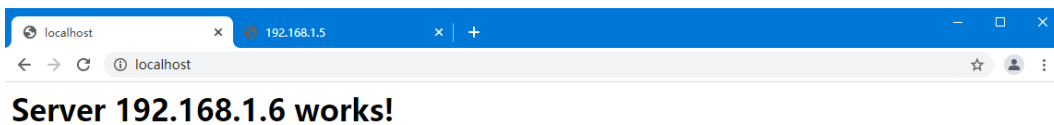
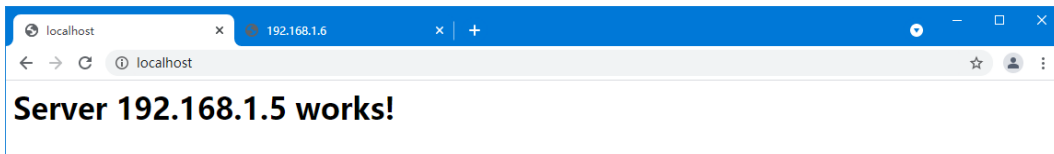
```
Define SRVROOT "C:\Users\Administrator\Desktop\httpd-2.4.48-win64-VS16\Apache24"  
ServerRoot "${SRVROOT}"
```

```
#ServerName www.example.com:80  
ServerName localhost:80
```

最后，安装 Apache 并启动 Apache，命令分别是：httpd -k install 和 httpd.exe -w -n "Apache2.4" -k start

```
C:\Users\Administrator\Desktop\httpd-2.4.48-win64-VS16\Apache24\bin>httpd -k install  
[Fri Jun 11 09:16:11.812759 2021] [mpm_winnt:error] [pid 6948:tid 384] AH00433: Apache2.4: Service is already installed.  
C:\Users\Administrator\Desktop\httpd-2.4.48-win64-VS16\Apache24\bin>httpd.exe -w -n "Apache2.4" -k start
```

服务器运行结果：



备注：出于设备原因，我们将服务器网段改为了 192.168.15.0，后续实验中我们修改了网页服务器的显示内容，以上截图是修改前的。

(2) 在完成步骤 2 后测试连通性，客户机 PC 可以 ping 通两台服务器

```
C:\Users\Administrator>ping 192.168.15.5  
  
正在 Ping 192.168.15.5 具有 32 字节的数据:  
来自 192.168.15.5 的回复: 字节=32 时间=525ms TTL=62  
来自 192.168.15.5 的回复: 字节=32 时间=48ms TTL=62  
来自 192.168.15.5 的回复: 字节=32 时间=38ms TTL=62  
来自 192.168.15.5 的回复: 字节=32 时间=69ms TTL=62  
  
192.168.15.5 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 38ms, 最长 = 525ms, 平均 = 170ms  
  
C:\Users\Administrator>ping 192.168.15.6  
  
正在 Ping 192.168.15.6 具有 32 字节的数据:  
来自 192.168.15.6 的回复: 字节=32 时间=160ms TTL=62  
来自 192.168.15.6 的回复: 字节=32 时间=150ms TTL=62  
来自 192.168.15.6 的回复: 字节=32 时间=132ms TTL=62  
来自 192.168.15.6 的回复: 字节=32 时间=142ms TTL=62  
  
192.168.15.6 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 132ms, 最长 = 160ms, 平均 = 146ms
```

(3) 查看 NAT 表：

Show ip nat translations 查看的是 nat 转换槽位，此时为空，表项包括：协议、内部全局地址：端口、内部本地地址：端口、外部本地地址：端口、外部全局地址：端口。

```
RG(config)#show ip nat translations  
Pro inside global      inside local      outside local      outside global  
RG(config)#
```



步骤 2: 在路由器上配置 IP 地址和路由, 以下是配置截图:

路由器 RG 的配置:

```
RG(config)#in serial 2/0
RG(config-if-Serial 2/0)#ip address 200.1.1.2 255.255.255.252
RG(config-if-Serial 2/0)#in gi 0/1
RG(config-if-GigabitEthernet 0/1)#ip address 192.168.15.1 255.255.255.0
RG(config-if-GigabitEthernet 0/1)#ip route 0.0.0.0 0.0.0.0 serial 2/0
RG(config)#
```

路由器 ISP 的配置:

```
ISP(config-if-Serial 2/0)#exit
ISP(config)#in serial 2/0
ISP(config-if-Serial 2/0)#ip address 200.1.1.1 255.255.255.252
ISP(config-if-Serial 2/0)#in gi 0/1
ISP(config-if-GigabitEthernet 0/1)#ip address 100.1.1.1 255.255.255.0
ISP(config-if-GigabitEthernet 0/1)#ip route 0.0.0.0 0.0.0.0 serial 2/0
ISP(config)#
```

步骤 3: 通过一个虚拟主机许可生命定义一个扩展的 IP 访问列表

```
RG(config)#access-list 150 permit ip any host 50.1.1.10
RG(config)#
```

步骤 4: 为真实主机定义一个 IP NAT 池, 确保其为旋转池

```
RG(config)#$2.168.15.5 192.168.15.6 prefix-length 24 type rotary
RG(config)#
```

步骤 5: 定义访问列表与主机之间的映射

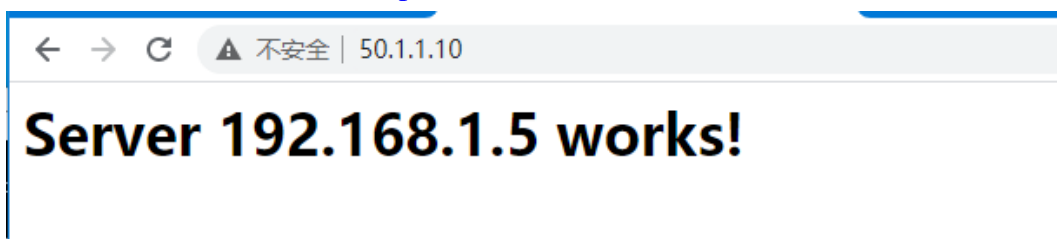
```
RG(config)#ip nat inside destination list 150 pool webserver
RG(config)#
```

步骤 6: 指定一个内部端口和一个外部端口

```
RG(config)#in serial 2/0
RG(config-if-Serial 2/0)#ip nat outside
RG(config-if-Serial 2/0)#in gi 0/1
RG(config-if-GigabitEthernet 0/1)#ip nat inside
RG(config-if-GigabitEthernet 0/1)#
```

步骤 7: 验证测试

(1) 在主机 A 上用浏览器打开 <http://50.1.1.10>



(2) 查看地址翻译过程

实验截图即步骤 7- (3) 的截图, 在输入该指令后, 并没有什么输出返回, 我们怀疑是访问服务器没有与 debug 同时进行的缘故。

(3) 查看 NAT 表, 说明表中端口号有什么作用

```
tcp 100.1.1.2:3433 100.1.1.2:3433 50.1.1.10:80 192.168.15.5:80
RG#debug ip nat
RG#
RG#show ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 100.1.1.2:3431 100.1.1.2:3431 50.1.1.10:80 192.168.15.5:80
tcp 100.1.1.2:3432 100.1.1.2:3432 50.1.1.10:80 192.168.15.5:80
tcp 100.1.1.2:3433 100.1.1.2:3433 50.1.1.10:80 192.168.15.5:80
RG#show ip nat translations
```

我们选第一行进行分析: 协议为 TCP、内部全局地址和内部本地地址都为 100.1.1.2:3421、而外部本地地址为 50.1.1.10:80、外部全局地址为 192.168.15.5:80, 这显然与我们所设想的不一樣。在我们的拓扑结构中, 192.168.15.0 才是内部网络, 而 50.1.1.0



为外部网络，内部本地地址应该是 192.168.15.5:80，内部全局地址应该是 50.1.1.10:80，而实验结果与我们的设想恰恰相反，产生的原因显然是 RG 把主机 A 那一部分当成内网了，具体原因我们再结合后续的实验结果分析。

端口号的作用：原来的 80 端口号是 HTTP 协议固定端口号，而与主机 A 对应的端口号 3421 应该是端口地址转换的结果，NAT 路由器改变了外出数据包的源端口并进行端口转换，转后的端口不会随着 IP 是全局的还是局部的而改变。至于为什么需要改变端口号，原因如下，因为 NAT 下的内部主机共用全局 IP 地址，若同一时刻若干 NAT 内的主机访问同一个外部网络并且以同一个端口会发生矛盾，因此 NAT 需要给他们分配新的端口号，以避免冲突。

(4) 在 Web 服务器上捕获数据包，查看发送过程中报文的 IP 地址转换情况，并解释

如下图所示：服务器 192.168.15.6（第一张图）什么也没检测到，而服务器 192.168.15.5（第二、三张图）可以捕获到来自主机 A（100.1.1.10）的 TCP 和 HTTP 报文，但是没有捕获到虚拟 IP（50.1.1.10）的包，这是由于 NAT 路由器已经将其转换成 IP 地址为 100.1.1.10 的缘故。

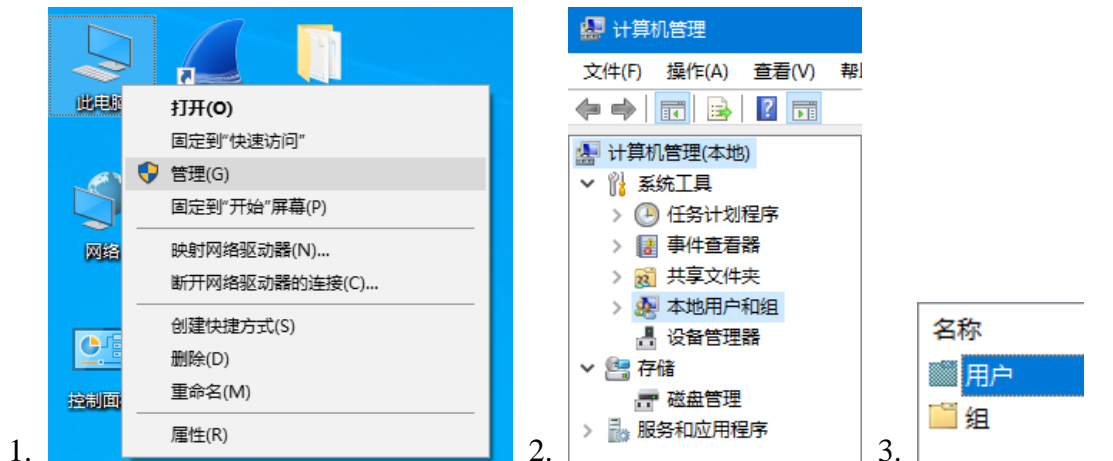
The image shows the Wireshark network protocol analyzer interface. The top pane displays a list of captured packets. The middle pane shows the details of the selected packet (No. 30), which is an HTTP GET request. The bottom pane shows the raw packet data in hexadecimal and ASCII.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|--------------|--------------|----------|--------|---|
| 5 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 6 | 0.000000 | 192.168.15.6 | 192.168.15.5 | TCP | 60 | 80 → 2219 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 7 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 8 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 9 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 10 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 11 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 12 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 13 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 14 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 15 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 16 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 17 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 18 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 19 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 20 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 21 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 22 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 23 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 24 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 25 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 26 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 27 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 28 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 29 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 30 | 0.000000 | 192.168.15.5 | 192.168.15.6 | HTTP | 566 | GET / HTTP/1.1 |
| 31 | 0.000000 | 192.168.15.5 | 192.168.15.6 | HTTP | 566 | 200 OK (text/html) |
| 32 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 33 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 34 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 35 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 36 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 37 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 38 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 39 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 40 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 41 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 42 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 43 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 44 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 45 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 46 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 47 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 48 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 49 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 50 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 51 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 52 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 53 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 54 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 55 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 56 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 57 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 58 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 59 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 60 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 61 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 62 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 63 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 64 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 65 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 66 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 67 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 68 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 69 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 70 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 71 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 72 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 73 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 74 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 75 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 76 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 77 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 78 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 79 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 80 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 81 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 82 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 83 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 84 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 85 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 86 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 87 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 88 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 89 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 90 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 91 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 92 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 93 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 94 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 95 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 96 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 97 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 98 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 99 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 100 | 0.000000 | 192.168.15.5 | 192.168.15.6 | TCP | 60 | 2219 → 80 [ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1 |

Frame 30: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits) on interface \Device\NPF_{EBC28BE2-6340-4CBB-87CD-F4EB4A236F0F}, id 0
Ethernet II, Src: 00:08:89:00:13:4a (00:08:89:00:13:4a), Dst: RuijieNe_47:2c:1d (00:05:88:47:2c:1d)
Internet Protocol Version 4, Src: 192.168.15.5, Dst: 100.1.1.2
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 287
Identification: 0x0755 (1877)
> Flags: 0x40, Don't fragment
Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.15.5
Destination Address: 100.1.1.2
> Transmission Control Protocol, Src Port: 80, Dst Port: 2219, Seq: 1, Ack: 513, Len: 247
> Hypertext Transfer Protocol

(5) 在服务机上建立用户名和口令，采用 Telnet 和远程桌面连接以替代 (1)，重做 (2~4) 内容。由于无法获取实验室电脑远程桌面用户权限，我们仅通过 Telnet 来测试负载均衡现象。

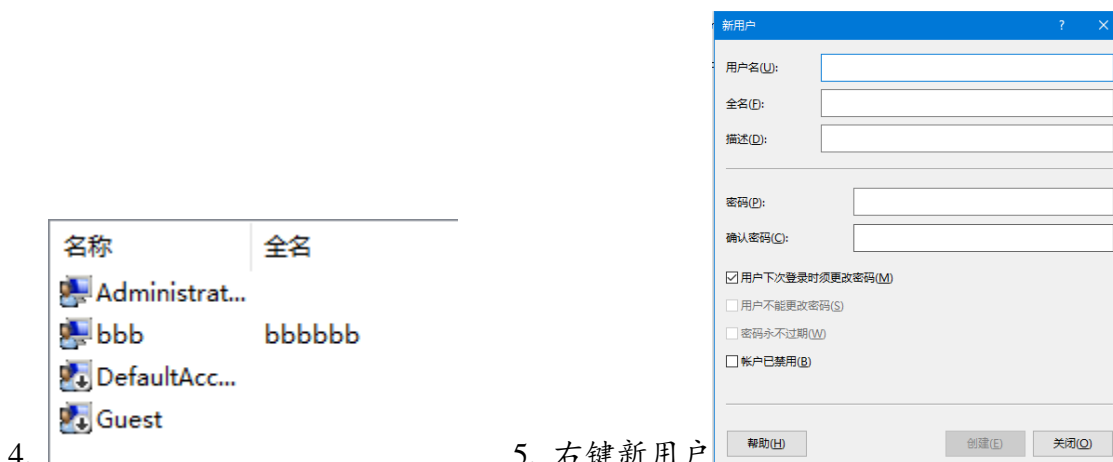
(a) 远程桌面用户建立过程：



1.

2.

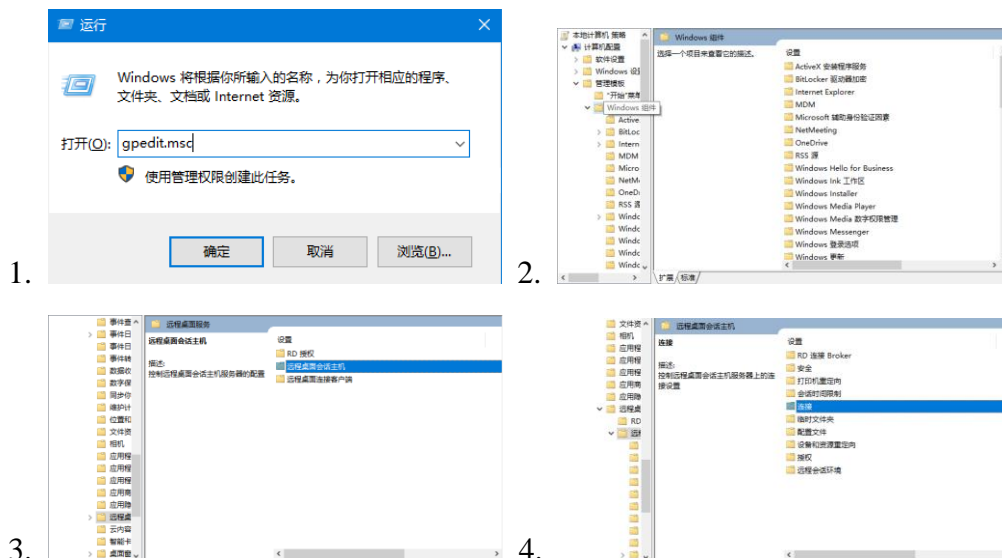
3.



4.

5. 右键新用户

(b) 勾选远程桌面选项的方法:

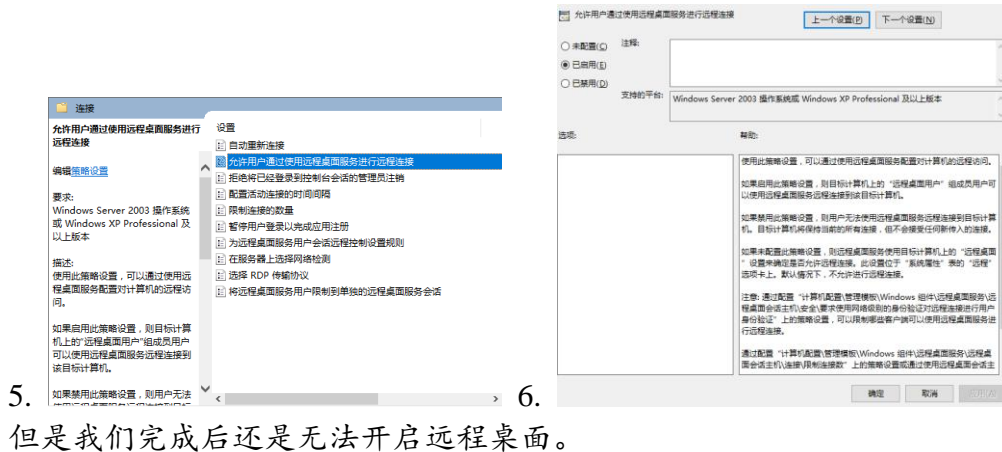


1.

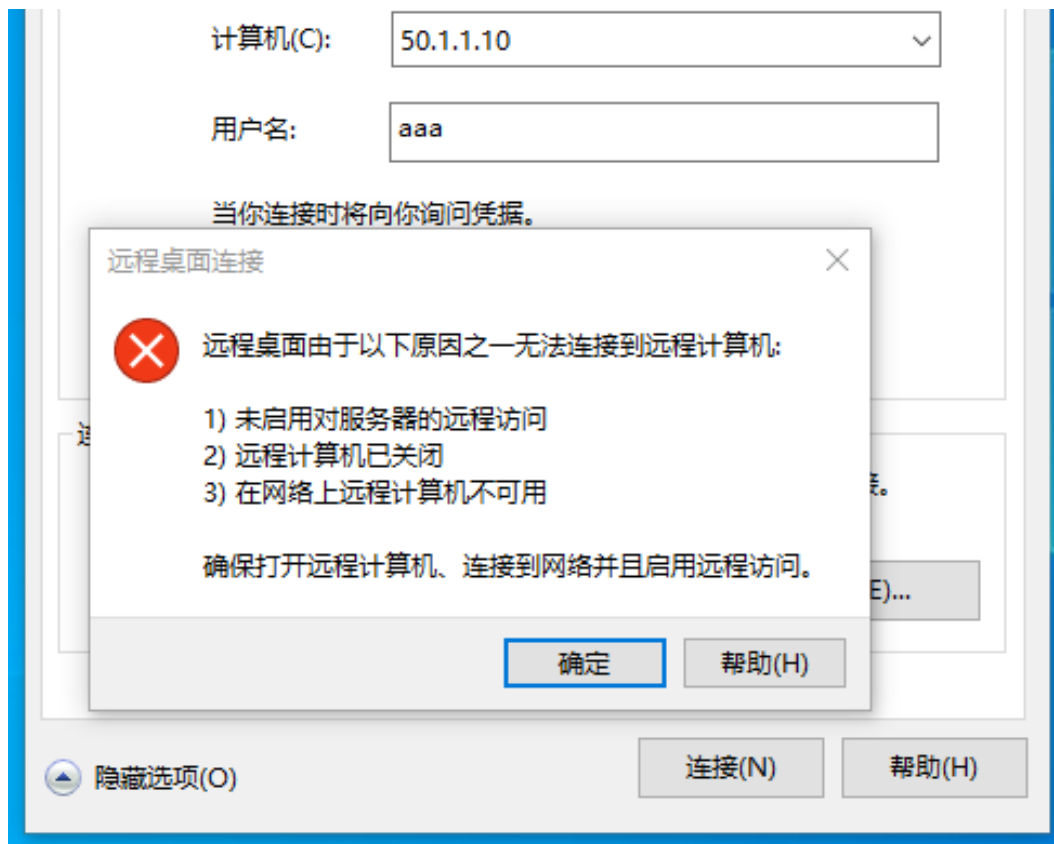
2.

3.

4.



但是我们完成后还是无法开启远程桌面。



(c) 改为使用软件 MobaXterm 开启了 Telnet 并进入，但是也没有要求输入用户和密码

```
11/06/2021 10:16.17 /home/mobaxterm telnet 50.1.1.10 22
Trying 50.1.1.10...
Connected to 50.1.1.10.
Escape character is '^]'.
SSH-2.0-OpenSSH_for_Windows_8.1

telnet> open
?Already connected to 50.1.1.10
telnet> send ayt
```

此时重做 (2~4)，查看地址翻译、NAT 表、抓包并解释，可以看到 debug ip nat 还是什么都没有，而 show ip nat translations 不上一次结果不一样了，这次的 inside 和 outside 符合我们的实验拓扑，即 100.1.1.2 是 outside，inside global 是 50.1.1.10:22，且 inside local 是 192.168.15.5:22，Telnet 的端口是 22 号，但我们依旧没有看到负载均衡，即另一个服务器 (.15.6) 没有接收到任何信息。



```
RG#debug ip nat
RG#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
tcp 50.1.1.10:22        192.168.15.5:22   100.1.1.2:5706     100.1.1.2:5706
RG#
```

下面这两张是 Wireshark 捕获截图，可以看到服务器 (.16.5) 可以捕获到 TCP 信息，但另一个服务器依然没有接收到服务请求。

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|--------------|--------------|----------|--------|--|
| 101 | 6.929868 | 100.1.1.2 | 192.168.15.5 | TCP | 66 | 5547 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 102 | 6.929990 | 192.168.15.5 | 100.1.1.2 | TCP | 66 | 80 → 5547 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 147 | 9.910883 | 100.1.1.2 | 192.168.15.5 | TCP | 66 | [TCP Retransmission] 5547 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 149 | 9.930355 | 192.168.15.5 | 100.1.1.2 | TCP | 66 | [TCP Retransmission] 80 → 5547 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 231 | 15.080240 | 100.1.1.2 | 192.168.15.5 | TCP | 62 | [TCP Retransmission] 5547 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1 |
| 233 | 15.931170 | 192.168.15.5 | 100.1.1.2 | TCP | 62 | [TCP Retransmission] 80 → 5547 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1 |
| 239 | 16.269943 | 100.1.1.2 | 192.168.15.5 | TCP | 60 | 5547 → 80 [ACK] Seq=1 Ack=1 Win=16445440 Len=0 |
| 285 | 19.006219 | 100.1.1.2 | 192.168.15.5 | TCP | 66 | [TCP Dup ACK 239#1] 5547 → 80 [ACK] Seq=1 Ack=1 Win=16445440 Len=0 SLE=0 SRE=1 |
| 378 | 24.922875 | 100.1.1.2 | 192.168.15.5 | TCP | 66 | [TCP Dup ACK 239#2] 5547 → 80 [ACK] Seq=1 Ack=1 Win=16445440 Len=0 SLE=0 SRE=1 |
| 1142 | 76.261713 | 192.168.15.5 | 100.1.1.2 | TCP | 54 | 80 → 5547 [FIN, ACK] Seq=1 Ack=1 Win=16776960 Len=0 |
| 1191 | 79.264032 | 192.168.15.5 | 100.1.1.2 | TCP | 54 | [TCP Retransmission] 80 → 5547 [FIN, ACK] Seq=1 Ack=1 Win=16776960 Len=0 |
| 1269 | 84.003095 | 100.1.1.2 | 192.168.15.5 | TCP | 60 | 5547 → 80 [ACK] Seq=1 Ack=2 Win=16445440 Len=0 |
| 1270 | 84.009908 | 100.1.1.2 | 192.168.15.5 | TCP | 60 | 5547 → 80 [FIN, ACK] Seq=1 Ack=2 Win=16445440 Len=0 |
| 1271 | 84.009983 | 192.168.15.5 | 100.1.1.2 | TCP | 54 | 80 → 5547 [ACK] Seq=2 Ack=2 Win=16776960 Len=0 |
| 1316 | 86.981688 | 100.1.1.2 | 192.168.15.5 | TCP | 60 | [TCP Retransmission] 5547 → 80 [FIN, ACK] Seq=1 Ack=2 Win=16445440 Len=0 |
| 1317 | 86.981752 | 192.168.15.5 | 100.1.1.2 | TCP | 54 | [TCP ZeroWindow] 80 → 5547 [ACK] Seq=2 Ack=2 Win=0 Len=0 |
| 1328 | 87.110202 | 100.1.1.2 | 192.168.15.5 | TCP | 60 | [TCP Dup ACK 1269#1] 5547 → 80 [ACK] Seq=2 Ack=2 Win=16445440 Len=0 |

```
> Frame 101: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{EBC28BE2-6340-4CBB-87CD-F4E84A236F0F}, id 0
  > Ethernet II, Src: RuijieNe_47:2c:1d (80:05:88:47:2c:1d), Dst: 00:88:99:00:13:4a (00:88:99:00:13:4a)
    > Internet Protocol Version 4, Src: 100.1.1.2, Dst: 192.168.15.5
      > Transmission Control Protocol, Src Port: 5547, Dst Port: 80, Seq: 0, Len: 0
        Source Port: 5547
        Destination Port: 80
        [Stream index: 0]
        [TCP Segment Len: 0]
        Sequence Number: 0 (relative sequence number)
        Sequence Number (raw): 3382265644
        [Next Sequence Number: 1 (relative sequence number)]
        Acknowledgment Number: 0
        Acknowledgment number (raw): 0
        1000 .... = Header Length: 32 bytes (8)
      > Flags: 0x002 (SYN)
        Window: 8192
        [Calculated window size: 8192]
```

【实验思考】

(1) 试验时不能采用 ping 进行验证的原因是：ping 发出的是 ICMP 数据包，而我们做的事 TCP 负载均衡，根据 NAT TCP 原理：NAT 负载均衡只适用于 TCP 连接，对于非 TCP 请求，NAT 不会对其进行转换。

(2) 在轮询机制下，TCP 负载均衡是与访问量无关的，除非只有一个访问者，在多访问体系中，两个服务器会因为 NAT 轮询机制而依次相应服务。使得能够观察到负载均衡效果的设计思路是使用多台客户机对服务器地址 50.1.1.10 进行访问，再分别与两台服务器上抓包即可。负载均衡规律性：体现负载均衡需要较多服务请求。

(3) 在 NAT 负载均衡下，多台主机可以同时提供服务并以一个唯一的全局 IP 提供给外网，外网客户机访问时无需记住内网多个服务器分别的 IP 地址，只需通过一个共同 IP 即可访问，并且基于轮询机，各服务器访问相近，不会出现某台服务器独自服务过重的情况，减轻每一台服务器的访问压力，提高了服务质量。

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。（按百分制）

| 学号 | 学生 | 自评分 |
|----------|-----|-----|
| 19358024 | 崔子潇 | 100 |
| 19335040 | 丁维力 | 100 |
| 19335286 | 郑有为 | 100 |
| | | |
| | | |

【交实验报告】

上传实验报告：<ftp://172.18.178.1/>

截止日期（不迟于）：1 周之内

上传包括两个文件：



计算机网络实验报告

(1) 小组实验报告。上传文件名格式：小组号_Ftp 协议分析实验.pdf （由组长负责上传）

例如：文件名“10_Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告

(2) 小组成员实验体会。每个同学单独交一份只填写了实验体会的实验报告。只需填写自己的学号和姓名。

文件名格式：小组号_学号_姓名_Ftp 协议分析实验.pdf （由组员自行上传）

例如：文件名“10_05373092_张三_Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告。

注意：不要打包上传！