西安电子科技大学网络与信息安全学院

《组网与运维》

TCP/IP 报文分析

实 验 报告

班级：1918039

姓名：王申奥

学号：19180300017

日期：2021.11.20

TCP/IP报文分析

一、实验目的

1. 掌握 H3C 设备 Ping 和 Tracert 命令的使用。
2. 掌握 H3C 设备的系统调试功能。
3. 掌握 ICMP 报文在 Ping 操作下的工作原理。
4. 掌握 H3C 设备 TCP 参数的设置。
5. 在 H3C 设备上进行 TCP 报文分析。
6. 在 H3C 设备上进行 UDP 报文分析。
7. 进一步熟悉 debug 命令的使用。

二、实验要求

1. 3 台具有 24 个以太网接口的交换机；
2. 2 台装有 Windows 系列操作系统的 PC（台式机或笔记本）；
3. 2 条双绞跳线（交叉线）；

三、实验步骤

1. 按实验 1 要求配置 H3C 路由器基本参数

（截取你自己的配置界面，并配以简单文字解释重要命令的含义。）

```
[H3C-R]sysname H3C-R1
[H3C-R1]inte
[H3C-R1]interface g
[H3C-R1]interface GigabitEthernet 0/0
[H3C-R1-GigabitEthernet0/0]ip a
[H3C-R1-GigabitEthernet0/0]ip address 192.168.1.1 24
[H3C-R1-GigabitEthernet0/0]%Nov 20 14:48:36:281 2021 H3C-R1 IFNET/3/PHY_UPDOWN: Physical state on the interface Gigabit
%Nov 20 14:48:36:282 2021 H3C-R1 IFNET/5/LINK_UPDOWN: Line protocol state on the interface GigabitEthernet0/0 changed t
%Nov 20 14:48:36:282 2021 H3C-R1 IFNET/3/PHY_UPDOWN: Physical state on the interface GigabitEthernet0/0.1 changed to up
%Nov 20 14:48:36:282 2021 H3C-R1 IFNET/3/PHY_UPDOWN: Physical state on the interface GigabitEthernet0/0.2 changed to up
%Nov 20 14:48:36:283 2021 H3C-R1 IFNET/5/LINK_UPDOWN: Line protocol state on the interface GigabitEthernet0/0.1 changed
%Nov 20 14:48:36:283 2021 H3C-R1 IFNET/5/LINK_UPDOWN: Line protocol state on the interface GigabitEthernet0/0.2 changed

[H3C-R1-GigabitEthernet0/0]quit
[H3C-R1]ip ro
[H3C-R1]ip route-static 192.168.2.0 255.255.255.0 192.168.1.2
```

```
[zfg]sysname H3C-R3
[H3C-R3]inter
[H3C-R3]interface g
[H3C-R3]interface GigabitEthernet 0/1
[H3C-R3-GigabitEthernet0/1]ip ad
[H3C-R3-GigabitEthernet0/1]ip address 192.168.2.2 24
[H3C-R3-GigabitEthernet0/1]quit
[H3C-R3]%Mar 25 10:44:37:174 2025 H3C-R3 IFNET/3/PHY_UPDOWN: Physical state on the interface Tunnel1 changed to down.
%Mar 25 10:44:37:175 2025 H3C-R3 IFNET/5/LINK_UPDOWN: Line protocol state on the interface Tunnel1 changed to down.

[H3C-R3]ip r
[H3C-R3]ip ro
[H3C-R3]ip route-static 192.168.1.0 255.255.255.0 192.168.2.1
```

1) Interface GigabitEthernet 0/0: 进入 Gigabitethernet 0/0 接口进行配置。
2) Ip address 192.168.1.1.24:ip 地址为 192.168.1.1.24。

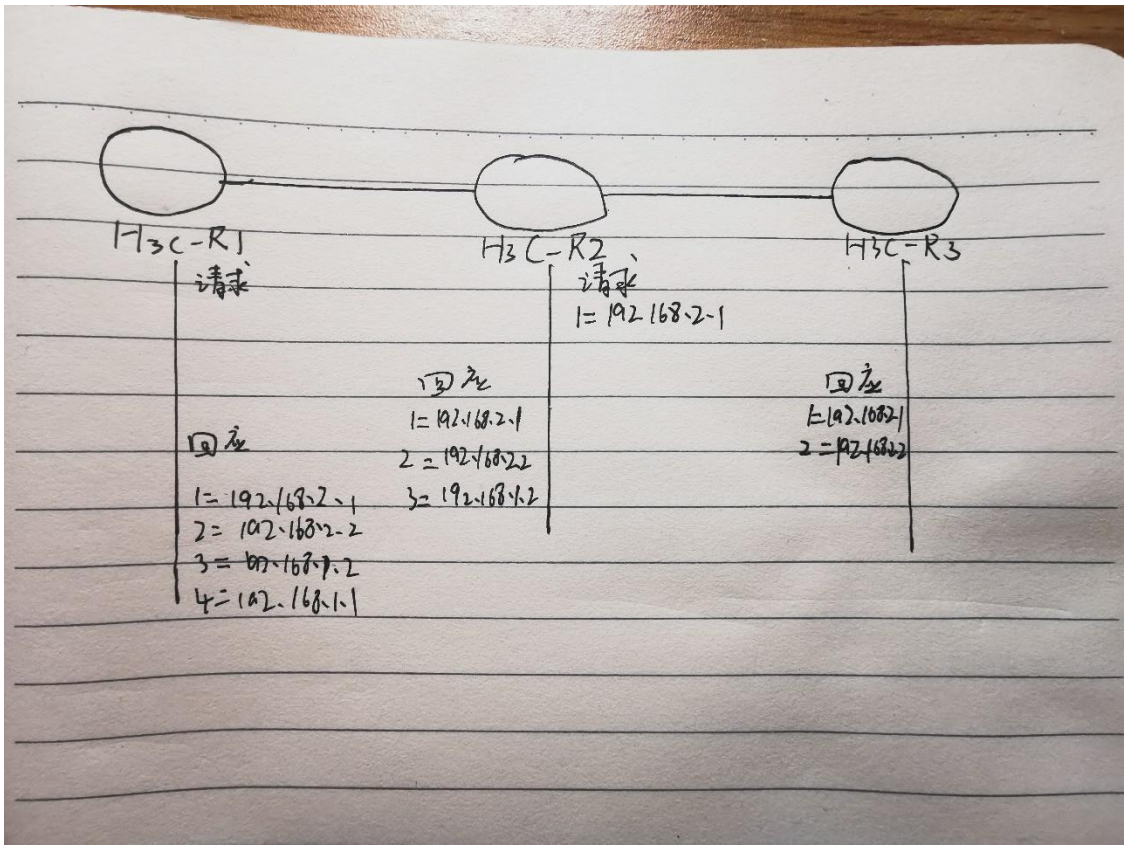
3） Ip route-static:配置静态路由。

2. 掌握 Ping 调试工具

（请截取你从 H3C-R1 上 ping 测试 H3C-R2 的 IP 地址 192.168.1.2 是否可达的图片，并参考图 7-7 自己绘图并配简单文字分析 ping -r 的原理。）

```
<H3C-R1>ping 192.168.1.2
Ping 192.168.1.2 (192.168.1.2): 56 data bytes, press CTRL_C to break
56 bytes from 192.168.1.2: icmp_seq=0 ttl=255 time=0.584 ms
56 bytes from 192.168.1.2: icmp_seq=1 ttl=255 time=0.319 ms
56 bytes from 192.168.1.2: icmp_seq=2 ttl=255 time=0.297 ms
56 bytes from 192.168.1.2: icmp_seq=3 ttl=255 time=0.316 ms
56 bytes from 192.168.1.2: icmp_seq=4 ttl=255 time=0.289 ms

--- Ping statistics for 192.168.1.2 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.289/0.361/0.584/0.112 ms
<H3C-R1>%Nov 20 14:55:50:077 2021 H3C-R1 PING/6/PING_STATISTICS: Ping statistics for 192
eceived, 0.0% packet loss, round-trip min/avg/max/std-dev = 0.289/0.361/0.584/0.112 ms.
```



原理：h3c-r1 发送请求给目的端设备 h3c-r3，中间设备 h3c-r2 将自己出接口的 ip 添加到 ICMP 请求报文中，并将报文转发。目的端收到报文后，发送 ICMP 响应报文，此报文会拷贝响应报文的内容，并将自己出接口的 ip 添加进去。中间设备 h3c-r2 将自己的出接口 IP 添加进去，并转发报文。最后源端 h3c-r1 收到响应报文，将自己的 ip 地址添加进去。最后得到具体经过的路由。

3. 掌握 Tracert 调试工具

（在 H3C-R1 上使用 tracert 命令查看报文从源端到目的端（IP 地址为 192.168.2.2）所经过的路径，在此处截图。然后解释如何解决路由器超时现象并截图。参考图 7-8 自己绘图并配简单文字解释 tracert 的原理。）使用 tracert 命令查看报文从源端到目的端（IP 地址为 192.168.2.2）所经过的路径：

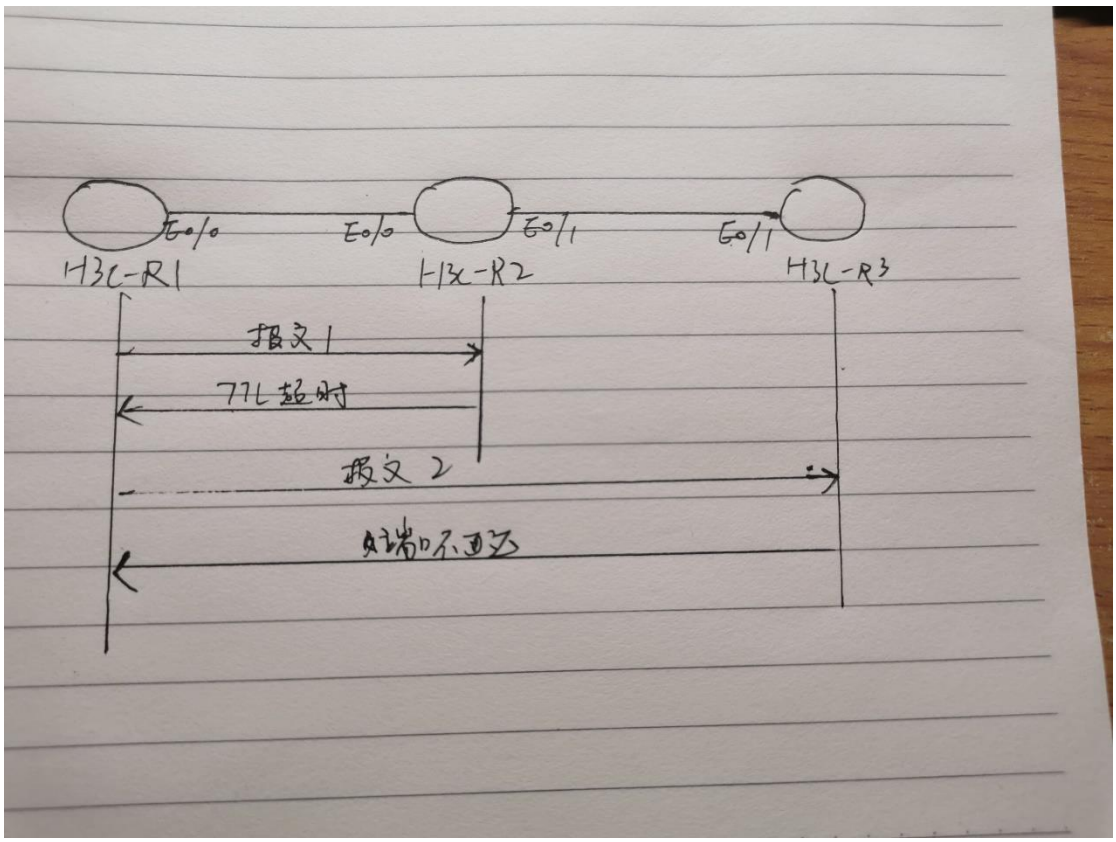
```
[H3C-R3]ip unreachables e
[H3C-R3]ip unreachables enable
[H3C-R3]
<H3C-R1>t
<H3C-R1>tr
<H3C-R1>tracert 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops at most, 52 bytes each packet, press CTRL_C to break
 1  192.168.1.2 (192.168.1.2)  0.486 ms  0.266 ms  0.267 ms
 2  192.168.2.2 (192.168.2.2)  0.522 ms  0.397 ms  0.407 ms
```

解决路由器超时现象：

```
<H3C-R1>tracert 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops at most, 52 bytes each packet, press CTRL_C to break
 1  * * *
 2  * * *
 3  * * *
 4
```

```
[H3C-R3]ip unreachables e
[H3C-R3]ip unreachables enable
[H3C-R3]
```

路由器超时是因为默认情况下，H3C 设备的 ICMP 超时报文发送功能是关闭的，将此命令开启即可，命令如图所示。



原理：

源端 H3C—R1 发送一个目的地址为 192.168.22,TTI 值为 1 的护数据报文到达 H3C—R2 时,第一跳 H3C—R2 发现报文的目的地不是本地且报文的 TTL 字段是 1，获得网关地址的下一跳为 192.168.12.并发送 TTL 超时差错报文。这样源端就得到了第一个三层设备的地址。

H3C—R1 根据差错报文获得下一跳的 ip 地址为 192.168.1.2。

H3C—R1 会重新向 H3C—R3 发送一个 IP 数据报文,其目的地址为 192.168.2.2,TTL 值为 2,H3C—R3 收到一个 UDP 的本机报文,根据报文的目的端口,无法找到对应的进程,则向报文源端发送一个 ICMP 不可达报文，H3C—R1 根据不可达报文获知到达了目的设备 H 3 C-R3。从而得到数据报文从 H3C-R1 到 H3C-R3 经历的路径。

4. 配置系统调试功能——Ping

（对“在 H3C-R1 上使用 Ping 命令向 H3C-R3 的 IP 地址 192.168.2.2 发送一个 Ping 报文，并在 H3C-R1 上打开 ICMP 报文信息的开关来观察 ICMP 报文输出”进行截图并加以文字简单说明。）

```
<H3C-R1>ping -c 1 -r 192.168.2.2
Ping 192.168.2.2 (192.168.2.2): 56 data bytes, press CTRL_C to break
56 bytes from 192.168.2.2: icmp_seq=0 ttl=254 time=0.742 ms
RR:     192.168.2.1
        192.168.2.2
        192.168.1.2
        192.168.1.1

--- Ping statistics for 192.168.2.2 ---
1 packet(s) transmitted, 1 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.742/0.742/0.742/0.000 ms
```

5. 配置系统调试功能——tracert

（截取“从 tracert 命令的 debug 调试信息可以看到 H3C-R1 接收的 ICMP 信息”，并配以简单文字解释。）

```
<H3C-R1>undo debugging all
All possible debugging has been turned off.
<H3C-R1>debu
<H3C-R1>debugging ip icmp
<H3C-R1>ping -c 1 192.168.2.2
Ping 192.168.2.2 (192.168.2.2): 56 data bytes, press CTRL_C to break
56 bytes from 192.168.2.2: icmp_seq=0 ttl=254 time=0.610 ms

--- Ping statistics for 192.168.2.2 ---
1 packet(s) transmitted, 1 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.610/0.610/0.610/0.000 ms
<H3C-R1>*Nov 20 15:05:31:711 2021 H3C-R1 SOCKET/7/ICMP:
ICMP Output:
 ICMP Packet: src = 192.168.1.1, dst = 192.168.2.2
              type = 8, code = 0 (echo)

*Nov 20 15:05:31:711 2021 H3C-R1 SOCKET/7/ICMP:
ICMP Input:
 ICMP Packet: src = 192.168.2.2, dst = 192.168.1.1
              type = 0, code = 0 (echo-reply)

%Nov 20 15:05:31:713 2021 H3C-R1 PING/6/PING_STATISTICS: Ping statistics for 192.168.2.2
 0.0% packet loss, round-trip min/avg/max/std-dev = 0.610/0.610/0.610/0.000 ms.
```

```
<H3C-R1>tracert -q 1 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops at most, 52 bytes each packet, press CTRL_C to break
 1  192.168.1.2 (192.168.1.2)  0.548 ms
 2  192.168.2.2 (192.168.2.2)  0.640 ms
<H3C-R1>*Nov 20 15:08:34:710 2021 H3C-R1 SOCKET/7/ICMP:
ICMP Input:
 ICMP Packet: src = 192.168.1.2, dst = 192.168.1.1
              type = 11, code = 0 (ttl-exceeded)
 Original IP: src = 192.168.1.1, dst = 192.168.2.2
              proto = 17, first 8 bytes = 82B2829A 00200000

*Nov 20 15:08:34:712 2021 H3C-R1 SOCKET/7/ICMP:
ICMP Input:
 ICMP Packet: src = 192.168.2.2, dst = 192.168.1.1
              type = 3, code = 3 (port-unreachable)
 Original IP: src = 192.168.1.1, dst = 192.168.2.2
              proto = 17, first 8 bytes = 82B2829B 00200000
```

说明：从 Tracert 命令的 debug 调试信息中可以看到 H3C—R1 接收的 ICMP 信息。

H3C—R1 发送一个目的地址为 192.168.22,TTI 值为 1 的护数据报文到达 H3C—R2 时,H3C—R2 发现报文的目的地不是本地且报文的 TTL 字段是 1，获得网关地址的下一跳为 192.168.12.则发送 TTL 超时差错报文。

H3C—R1 根据差错报文获得下一跳的 ip 地址为 192.168.1.2

H3C—R1 会重新向 H3C—R3 发送一个 IP 数据报文,其目的地址为 192.168.2.2,TTL 值为 2,H3C—R3 收到一个 UDP 的本机报文,根据报文的目的端口,无法找到对应的进程,则向报文源端发送一个 ICMP 不可达报文，H3C—R1 根据不可达报文获知到达了目的设备 H３C-R3。

6. 按实验 2 配置 H3C 路由器基本参数

（截取你配置 H3C 路由器基本参数的界面，以及配置完后，测试 H3C-R1 和 H3C-R2 连通性的界面）

```
[H3C-R1]interface GigabitEthernet 0/0
[H3C-R1-GigabitEthernet0/0]ip a
[H3C-R1-GigabitEthernet0/0]ip address 192.168.1.1 24
[H3C-R1-GigabitEthernet0/0]quit
[H3C-R1]tel
[H3C-R1]telnet s
[H3C-R1]telnet server e
[H3C-R1]telnet server enable
[H3C-R1]loc
[H3C-R1]local-s
[H3C-R1]local-u
[H3C-R1]local-user h3c
New local user added.
[H3C-R1-luser-manage-h3c]pa
[H3C-R1-luser-manage-h3c]password s
[H3C-R1-luser-manage-h3c]password simple h3c
[H3C-R1-luser-manage-h3c]ser
[H3C-R1-luser-manage-h3c]service-type t
[H3C-R1-luser-manage-h3c]service-type tel
[H3C-R1-luser-manage-h3c]service-type telnet
[H3C-R1-luser-manage-h3c]quit
[H3C-R1]us
[H3C-R1]use
[H3C-R1]user-i
[H3C-R1]user-interface vty 0 4
[H3C-R1-line-vty0-4]a
[H3C-R1-line-vty0-4]au
[H3C-R1-line-vty0-4]aut
[H3C-R1-line-vty0-4]authentication-mode s
[H3C-R1-line-vty0-4]authentication-mode scheme
[H3C-R1-line-vty0-4]quit
```

```
<H3C-R1>ping 192.168.1.2
Ping 192.168.1.2 (192.168.1.2): 56 data bytes, press CTRL_C to break
56 bytes from 192.168.1.2: icmp_seq=0 ttl=255 time=0.521 ms
56 bytes from 192.168.1.2: icmp_seq=1 ttl=255 time=0.235 ms
56 bytes from 192.168.1.2: icmp_seq=2 ttl=255 time=0.202 ms
56 bytes from 192.168.1.2: icmp_seq=3 ttl=255 time=0.203 ms
56 bytes from 192.168.1.2: icmp_seq=4 ttl=255 time=0.195 ms

--- Ping statistics for 192.168.1.2 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.195/0.271/0.521/0.126 ms
<H3C-R1>%Nov 20 15:38:42:036 2021 H3C-R1 PING/6/PING_STATISTICS: Ping statistics for 192
eceived, 0.0% packet loss, round-trip min/avg/max/std-dev = 0.195/0.271/0.521/0.126 ms.
```

7. 测试 TCP 信息

（截取你在 H3C-R2 上使用 telnet 命令访问 H3C-R1 的过程中出现的调试信息输出报文，并根据调试信息输出的报文具体分析 TCP 建立连接三次握手的具体过程，根据调试信息输出的报文具体分析关闭 TCP 连接四次握手的具体过程。）

TCP 建立连接三次握手：

```
TCP Input(vrf = 0, state = LISTEN):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687906, ack = 0, flag =  SYN
             window = 64512, checksum = 0x6115, datalen = 0, headlen = 40

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP Synrespond(vrf = 0, state = SYN_RCVD):
 TCP packet: src = 192.168.1.1/23, dst = 192.168.1.2/29640
             seq = 6088683, ack = 1045687907, flag =  SYN ACK
             window = 4096, checksum = 0x8bdd, datalen = 0, headlen = 40

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP Input(vrf = 0, state = LISTEN):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687907, ack = 6088684, flag =  ACK
             window = 8083, checksum = 0xaa5e, datalen = 0, headlen = 32

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = KEEP) restart, timeout = 75000.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP state change: SYN_RCVD --> ESTABLISHED.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = KEEP) restart, timeout = 7200000.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = REXMT) stop.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.
```

首先,H3C—R2 发送一个 SYN 报文段,指明 H3C—R2 打算连接 H3C—R1 的端口,以及初始顺序号(seq = a)。H3C—R1 发回包含 H3C—R2 的初始顺序

号的 SYN 报文段(seq = b)作为应答。同时,将确认号设置为 H3C-R2 的顺序号加 1,以对 H3C—R2 的 SYN 报文段进行确认(ACK 为 a +1)。H3C—R2 将确认号设置为 H3C—R1 的顺序号加 1,以对 H3C—R1 的 SYN 报文段进行确认(ACK 为 b +1),该报文通知目的主机双方已完成连接建立。

关闭 TCP 连接四次握手：

```
*Nov 20 16:01:01:365 2021 H3C-R1 SOCKET/7/TCP:
TCP Output: TCP send FIN packet.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:01:01:365 2021 H3C-R1 SOCKET/7/TCP:
TCP Output(vrf = 0, state = FIN_WAIT_1):
 TCP packet: src = 192.168.1.1/23, dst = 192.168.1.2/29640
             seq = 6089166, ack = 1045687981, flag =  FIN ACK
             window = 634, checksum = 0x73d7, datalen = 0, headlen = 32

*Nov 20 16:01:01:365 2021 H3C-R1 SOCKET/7/TCP:
TCP Input(vrf = 0, state = FIN_WAIT_1):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687981, ack = 6089167, flag =  ACK
             window = 8083, checksum = 0x56ab, datalen = 0, headlen = 32
```

```
*Nov 20 16:01:01:365 2021 H3C-R1 SOCKET/7/TCP:
TCP state change: FIN_WAIT_1 --> FIN_WAIT_2.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

%Nov 20 16:01:01:367 2021 H3C-R1 SHELL/5/SHELL_LOGOUT: h3c logged out from 192.168.1.2.
*Nov 20 16:01:01:419 2021 H3C-R1 SOCKET/7/TCP:
TCP Input(vrf = 0, state = FIN_WAIT_2):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687981, ack = 6089167, flag =  FIN ACK
             window = 8083, checksum = 0x5674, datalen = 0, headlen = 32

*Nov 20 16:01:01:419 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = KEEP) restart, timeout = 60000.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:01:01:419 2021 H3C-R1 SOCKET/7/TCP:
TCP Input: TCP received FIN packet.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:01:01:419 2021 H3C-R1 SOCKET/7/TCP:
TCP Output(vrf = 0, state = TIME_WAIT):
 TCP packet: src = 192.168.1.1/23, dst = 192.168.1.2/29640
             seq = 6089167, ack = 1045687982, flag =  ACK
             window = 633, checksum = 0x7357, datalen = 0, headlen = 32

*Nov 20 16:02:01:659 2021 H3C-R1 SOCKET/7/TCP:
TCP state change: TIME_WAIT --> CLOSED.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.
```

收到关闭命令后，TCP 在发送完尚未处理的报文段后,发 FIN =1 的报文段给对方,且 TCP 不再受理本方应用进程的数据发送。在 FIN 以前发送的数据字节,包括 FIN ,都需要对方确认,否则要重传。注意 FIN 也占一个顺序号。一旦收到对方对 FIN 的确认以及对方的 FIN 报文段,本方 TCP 就对

该 FIN 进行确认,在等待一段时间后关闭连接。等待是为了防止本方的确认报文丢失,避免对方的重传报文干扰新的连接。

8.配置 TCP 属性

（截取你自己的界面，并配以简单文字解释重要命令的含义。）

```
[H3C-R1]interface GigabitEthernet 0/0
[H3C-R1-GigabitEthernet0/0]tcp mss 1000
[H3C-R1-GigabitEthernet0/0]quit
[H3C-R1]tcp window 10
```

Tcp mss 1000:设置接口的 TCP 最大报文段长度为 1000.

Tcp windows 10:j 将 tcp 连接的收发缓冲区大小设置为 10kb。

五、实验结果及分析

1. 我们第一个实验对应的图 6-7 所示的拓扑图和 132 页下面给的节本参数配置命令之间出现了什么错误？

应将 Ethernet 变为 GigabitEthernet，因为实验室实际是千兆网口。

2. 整个实验过程中遇到什么问题（有截图最好），如何解决的？通过该实验有何收获？

```
TCP Input(vrf = 0, state = LISTEN):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687906, ack = 0, flag =  SYN
             window = 64512, checksum = 0x6115, datalen = 0, headlen = 40

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP Synrespond(vrf = 0, state = SYN_RCVD):
 TCP packet: src = 192.168.1.1/23, dst = 192.168.1.2/29640
             seq = 6088683, ack = 1045687907, flag =  SYN ACK
             window = 4096, checksum = 0x8bdd, datalen = 0, headlen = 40

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP Input(vrf = 0, state = LISTEN):
 TCP packet: src = 192.168.1.2/29640, dst = 192.168.1.1/23
             seq = 1045687907, ack = 6088684, flag =  ACK
             window = 8083, checksum = 0xaa5e, datalen = 0, headlen = 32

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = KEEP) restart, timeout = 75000.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP state change: SYN_RCVD --> ESTABLISHED.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = KEEP) restart, timeout = 7200000.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.

*Nov 20 16:00:50:930 2021 H3C-R1 SOCKET/7/TCP:
TCP timer(type = REXMT) stop.
 Connection info: src = 192.168.1.1:23, dst = 192.168.1.2:29640.
```

开始没有收到数据包的信息，后来听老师讲解，此部分应在实验一 debugging 的基础上去做便会有结果。在输入实验一的 debugging 相关操作后问题解决。