

《计算机网络》课内实验 实验指导手册

授课教师:

实验指导教师:

教学对象:

开课时间： 第五学期

北京邮电大学软件学院

2017 年 9 月

实验一：数据链路层实验（2 学时）

1、 实验目的

通过本实验使学生理解协议数据单元（PDU）概念、掌握以太网帧结构字段定义和功能。

2、 实验任务

搭建实验环境，使用网络抓包软件（如 Wireshark 软件等）抓取访问互联网所产生的数据包，分析其中的以太网帧结构字段组成，掌握以太网帧结构字段的功能。

3、 实验内容

- 1) 在可以访问互联网的主机上下载并安装网络抓包软件 Wireshark。
- 2) 运行 Wireshark 软件，启动 Wireshark 软件的抓包功能抓取本主机访问互联网中某网站过程中发送和接收的数据包。
- 3) 对所抓取的数据包进行分析，分析所发送和接收的数据包的以太网帧结

构中的源 MAC 地址、目的 MAC 地址和类型（type）字段的使用方法；理解各字段的含义和功能。

- 4) 选做部分：分析所抓取的数据包中的 DNS（Domain Name System）消息、TCP 报文、IP 分组、HTTP 协议消息的字段组成及作用。

4、 实验环境

- 1) Windows 系统主机或 Linux 系统主机；
- 2) Wireshark 软件，软件下载网址：<https://www.wireshark.org/>

5、 实验进度安排

本实验 2 学时。在实验前，根据所使用的主机操作系统下载正确的 Wireshark 软件版本、下载 Wireshark 软件使用手册。

6、 实验方法与步骤

- 1) 在主机上安装 Wireshark 软件。在 Linux 操作中，也可以使用在线安装方式安装 Wireshark 软件。
- 2) 检查主机是否能够正常访问互联网。如果不能，检查出错原因并排除故障。
- 3) 运行 Wireshark 软件，选择主机访问互联网所使用的网卡（NIC），开启在这个网卡上的抓包功能。若主机配置有多个网卡，也可以开启对所有网卡的抓包。
- 4) 启动浏览器程序，在浏览器的地址栏输入需要访问的 Web 网站的 URL 地址。应该可以看到在浏览器上呈现网页内容过程的同时，在 Wireshark 软件主窗口中也会陆续新增与访问该网页过程中主机发送和接收的数据包对应的记录。鼠标双击主窗口中与一个数据包对应的记录后，弹出此数据包的解析窗口。

图 1 为 Wireshark 主窗口示例。图 2 为 Wireshark 数据包解析窗口示例。

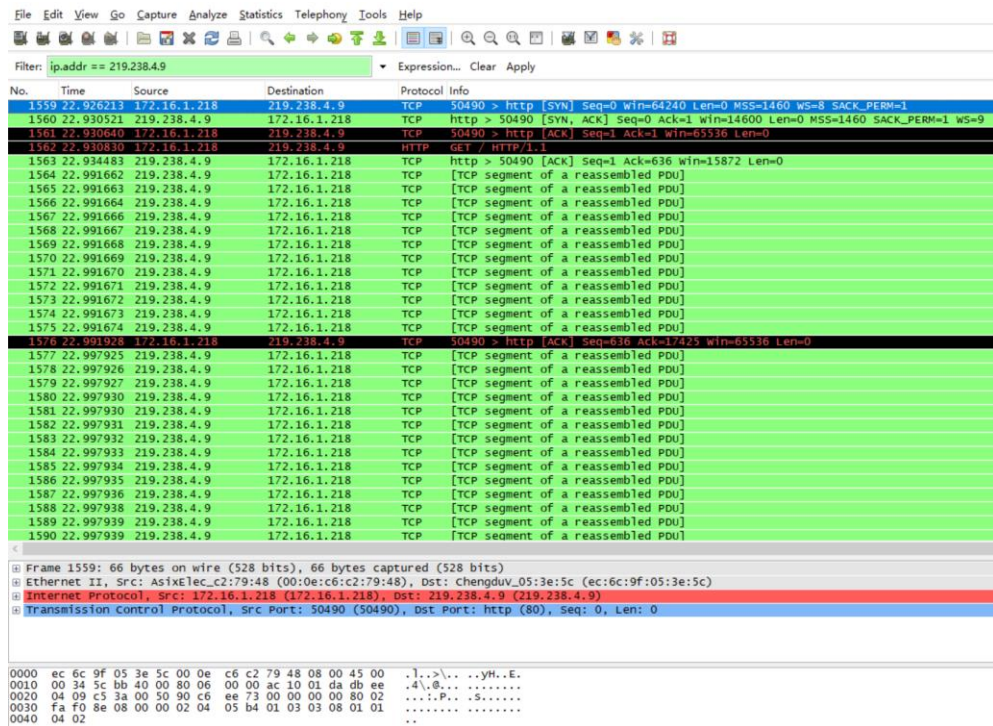


图 1. Wireshark 主窗口示例

- 5) 查看主机网卡的 IP 地址和 MAC 地址：在 Windows 系统中，使用 ipconfig 命令查看网卡的 IP 地址和 MAC 地址；在 Linux 系统中，使用 ifconfig 命令查看网卡的 IP 地址和 MAC 地址。
- 6) 分析所抓取的数据包是本主机发送的数据包还是接收的数据包。分析所抓取数据包中以太网帧结构：源 MAC 地址字段、目的 MAC 地址字段、类型（type）字段的取值、帧中负荷（payload）字段的协议消息类型。
- 7) 选做部分：分析访问互联网网站的协议过程，包括 DNS 域名解析过程、TCP 连接建立过程、HTTP 协议过程；分析 DNS 协议消息、IP 协议消息、TCP 协议消息、HTTP 协议消息格式。
- 8) 停止 Wireshark 抓包过程，退出 Wireshark 软件前将所抓取的数据包存入文件，以备后用。

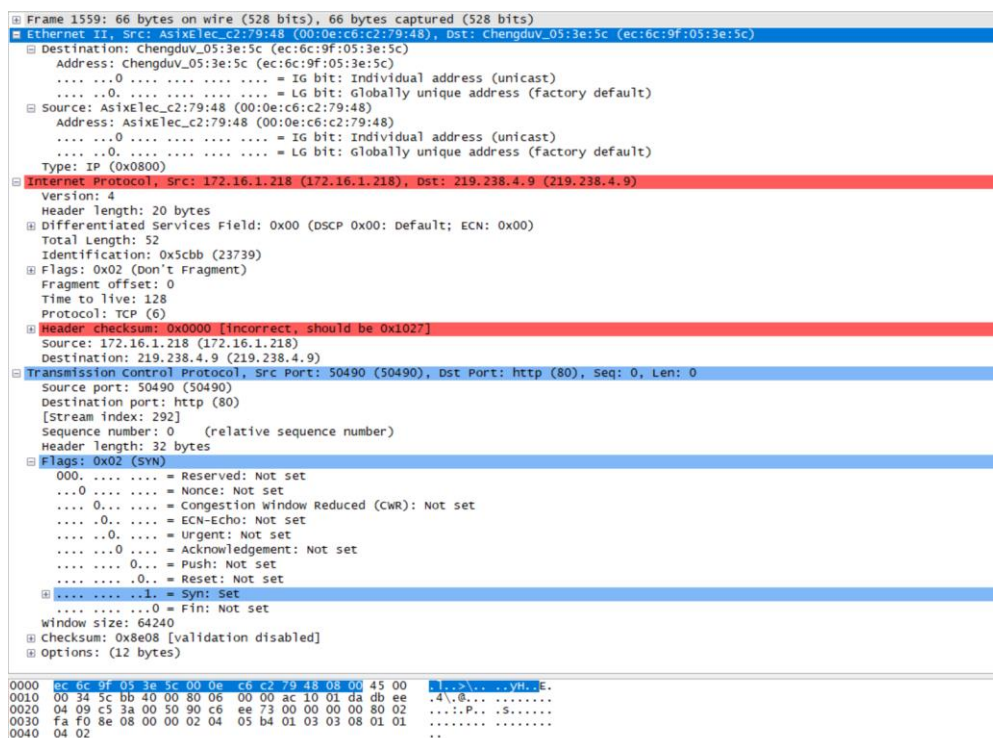


图 2. Wireshark 数据包解析窗口示例

7、实验管理规则及实验验收

实验完成后填写并提交实验报告。实验报告需记录实验过程、所抓取的数据包界面截图，分析所抓取的数据包，并阐述以太网帧结构各字段的含义、作用、使用方法。

实验二：网络层实验（2 学时）

1、实验目的

通过本实验使学生理解网络层协议功能、理解并掌握网络层的转发（Forwarding）和路由（Routing）概念、掌握 Linux 系统网络基本配置。

2、实验任务

基于虚拟机平台（Oracle VirtualBox 或 VMware WorkStation）和 Linux 操作系统搭建实验用网络拓扑环境；规划并配置该网络拓扑环境中网络设备 IP 地址；配置该网络拓扑环境中的路由；实现该网络拓扑环境中设备之间网络层连通。

3、实验内容

1) 规划实验用网络拓扑：在该网络拓扑中，共有两台路由器（R1、R2）和两台主机（H1、H2）；路由器 R1、R2 各配置有两个网卡，主机 H1、H2

各配置一个网卡。路由器 R1、R2、和主机 H1、H2 的拓扑连接如图 1 所示。

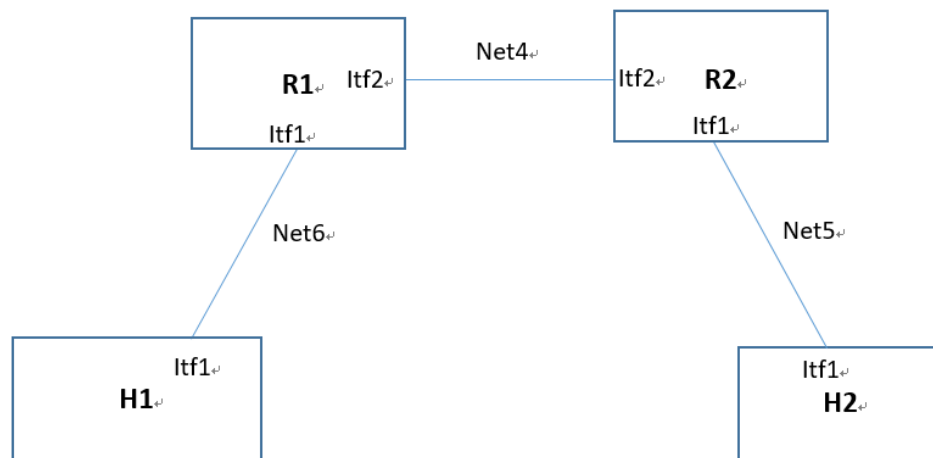


图 1. 实验用网络拓扑

- 2) 在虚拟机平台（Oracle VirtualBox 或 VMware WorkStation）上创建四台 Linux 虚拟机（本实验以 Ubuntu 操作系统为例）；其中两台 Linux 虚拟机配置两个网卡，对这两台 Ubuntu 虚拟机进行配置，启用其转发 IP 分组功能，使这两台虚拟机具有路由器功能，记这两台 Linux 虚拟机为路由器 R1 和 R2；另外两台 Linux 虚拟机各配置一个网卡，记为主机 H1 和 H2。这四台 Linux 虚拟机网卡的类型都设置为自定义类型（VMware Workstation 平台环境中）、或内部网络类型（Oracle VirtualBox 平台环境中）。
- 3) 利用虚拟机平台提供的虚拟网络功能，将 Linux 虚拟机 R1 和 H1 配置在同一虚拟以太网中（如图 1，在本实验中记此虚拟以太网为 Net6）；将 Linux 虚拟机 R1、R2 配置到同一虚拟以太网中（如图 1，在本实验中记此虚拟以太网为 Net4）；将 Linux 虚拟机 R2 和 H2 配置在同一虚拟以太网中（如图 1，在本实验中记此虚拟以太网为 Net5）。
- 4) 规划虚拟以太网 Net4、Net5 和 Net6 的网络 ID、子网掩码。为虚拟机 R1、R2、H1 和 H2 的网卡配置 IP 地址。
- 5) 在主机 H1 中配置其缺省路由器 IP 地址为路由器 R1 的网卡 ltf1（如图 1 所示）的 IP 地址；在主机 H2 上配置其缺省路由器 IP 地址为路由器 R2

的网卡 `ltf1`（如图 1 所示）的 IP 地址。采用静态路由配置方法，在路由器 R1 中配置到网络 Net5 的路由，在路由器 R2 中配置到网络 Net6 的路由。

- 6) 使用 `ping` 命令，测试主机 H1 和 H2 网络层的连通性。如果网络层不通，利用 Wireshark 软件在 H1、H2、R1 和 R2 上抓取 ICMP 协议数据包，分析原因，修改虚拟机的路由配置问题或其它网络配置问题，直至主机 H1 和 H2 之间网络层连通。

4、 实验环境

- 1) Windows 系统主机或 Linux 系统主机；
- 2) Wireshark 软件：软件下载网址：<https://www.wireshark.org/>
- 3) Oracle VirtualBox 软件，软件下载网址：<https://www.virtualbox.org/>；
或 VMware Workstations 软件。

5、 实验进度安排

本实验 2 学时。在实验前，下载实验用虚拟机平台 VirtualBox；在 Windows 系统宿主机中安装虚拟机平台软件 VirtualBox；在虚拟机平台中创建四台 Ubuntu 系统虚拟机，在虚拟机中安装 Wireshark 软件。

6、 实验方法与步骤

- 1) 将所创建的 Ubuntu 虚拟机中的两台各配置两个网卡，这两台虚拟机记为 R1 和 R2；另两台 Ubuntu 虚拟机各配置一个网卡，记这两个 Ubuntu 虚拟机为 H1 和 H2。
- 2) 启动虚拟机 R1，编辑 `/etc/sysctl.conf` 文件，去掉文件中的 `net.ipv4.ip_forward=1` 语句前面的注释符号（#），从而开启 Linux 系统的转发 IPv4 分组的功能。

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

图 2. `/etc/sysctl.conf` 中的 `ip_forward` 配置项

对虚拟机 R2 做同样配置修改，开启 R2 的 IPv4 分组转发功能。

- 3) 规划虚拟以太网 Net4、Net5 和 Net6 的网络 ID、子网掩码。为虚拟机 R1、R2、H1 和 H2 的网卡配置 IP 地址。

在 Ubuntu 系统中，可使用 `lshw -class network` 命令查看系统配置的网卡信息，此命令示例如图 3。可使用 `ifconfig` 命令查看网卡 IP 地址配置信息，此命令示例如图 4。

通过编辑 `/etc/network/interfaces` 文件为网卡配置 IP 地址。在图 5 所示 `/etc/network/interfaces` 文件中，`eth0`、`eth1`、`eth2` 为网卡的名称；网卡 `eth0` 采用动态 IP 地址分配方式（DHCP）方式获得 IP 地址；网卡 `eth1`、`eth2` 采用静态 IP 地址分配方式获得 IP 地址。也可使用 Linux 系统的 `ip` 命令为网卡配置临时 IP 地址。在 Linux 系统中可使用 `man` 命令查看 Linux 命令使用方法。如在 Linux Shell 窗口中输入 `man ip` 命令可以查看 `ip` 命令的使用方法。

```
abc@abc-VirtualBox:~/Downloads$ sudo lshw -class network
*-network:0
  description: Ethernet interface
  product: 82540EM Gigabit Ethernet Controller
  vendor: Intel Corporation
  physical id: 3
  bus info: pci@0000:00:03.0
  logical name: eth0
  version: 02
  serial: 08:00:27:4f:b8:9e
  size: 1Gbit/s
  capacity: 1Gbit/s
  width: 32 bits
  clock: 66MHz
  capabilities: pm pci_x bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-f
  configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=7.3.21-k8-NAPI
  resources: irq:19 memory:f0000000-f001ffff ioport:d010(size=8)
*-network:1
  description: Ethernet interface
  product: 82540EM Gigabit Ethernet Controller
  vendor: Intel Corporation
```

图 3. `lshw -class network` 命令示例

```
abc@abc-VirtualBox:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:4f:b8:9e
          inet addr:192.168.0.12 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4f:b89e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43819 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26337 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41431013 (41.4 MB)  TX bytes:4972231 (4.9 MB)
```

图 4. `ifconfig` 命令示例

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.103.101
    netmask 255.255.255.0

auto eth2
iface eth2 inet static
    address 192.168.106.101
    netmask 255.255.255.0
```


图 5. /etc/network/interfaces 文件内容示例

- 4) 在主机 H1 中采用 `route` 命令为主机 H1 配置缺省路由，将主机 H1 的缺省路由设置为图 1 中路由器 R1 的 `Itf1` 网卡的 IP 地址。

图 6 展示了使用 `route` 命令设置缺省路由，以及使用 `route` 命令显示路由表中的路由配置信息。

对主机 H2 做类似配置，使得主机 H2 的缺省路由网关为 R2 的 `Itf1` 的 IP 地址。

```
user01@ubuntu:~$ sudo route add default gw 192.168.169.131
user01@ubuntu:~$ route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          192.168.169.131 0.0.0.0          UG    0      0      0 eth0
192.168.169.0    *                255.255.255.0    U      0      0      0 eth0
user01@ubuntu:~$
```

图 6. 缺省路由配置命令示例及显示路由表命令示例

- 5) 在虚拟机 R1 中使用 `route` 命令配置从路由器 R1 到网络 Net5（如图 1）的路由。图 7 中，使用 `route` 命令在路由表中添加了一条到网络 192.168.169.0/24 的路由，此路由通过网卡 `eth1` 转发 IP 分组，下一跳（Next-Hop）IP 地址是 192.168.25.131。

```
abc@ubuntu:/etc/network$ sudo route add -net 192.168.169.0 netmask 255.255.255.0 gw 192.168.25.131 dev eth1
abc@ubuntu:/etc/network$ route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
link-local       *                255.255.0.0      U      1000   0      0 eth0
192.168.25.0     *                255.255.255.0    U      0      0      0 eth1
192.168.140.0    *                255.255.255.0    U      0      0      0 eth2
192.168.150.0    *                255.255.255.0    U      0      0      0 eth0
192.168.169.0    192.168.25.131  255.255.255.0    UG     0      0      0 eth1
abc@ubuntu:/etc/network$
```

图 7. 使用 `route` 命令配置路由项示例及路由表信息显示示例

- 对路由器 R2 做类似配置，使得路由器 R2 中有一条到网络 Net6 的路由。
- 6) 在主机 H1 上使用 `ping` 命令测试主机 H1 与主机 H2 在网络层的连通性。
- 如果在网络层不通，检查路由表、IP 地址等是否配置正确。可使用 Wireshark 软件分别在主机 H1、路由器 R1、路由器 R2 和主机 H2 抓取 ICMP 协议消息，分析 `ping` 命令产生的 ICMP 消息的转发情况，定位故障原因，排除配置错误，直至主机 H1 和主机 H2 在网络层连通。

7、 实验管理规则及实验验收

实验完成后填写并提交实验报告。实验报告需要记录所规划的网络地址、IP 地址配置、虚拟机 H1、H2、路由器 R1 和 R2 中配置的路由表、`ping` 命令执

行结果界面截图。

实验三：传输层实验（4 学时）

1、 实验目的

通过本实验使学生理解并掌握 TCP 连接，掌握 TCP 报文（TCP Segment）字段组成、字段的作用和使用方法。

2、 实验任务

编写基于 TCP 套接字（Socket）的 TCP 服务器应用程序和 TCP 客户端应用程序，TCP 服务器应用程序支持与多个 TCP 客户端应用程序同时建立 TCP 连接；将 TCP 客户端应用程序和服务器应用程序分别部署到本课程实验二（“网络层实验”）搭建的网络环境中的主机 H1 和 H2 上。分别观测下列情况下 TCP 协议过程及 TCP 报文字段取值：

- 1) TCP 服务器应用程序未启动情况下，TCP 客户端应用程序向服务器发出 TCP 连接建立请求；
- 2) TCP 服务器应用程序已启动，但主机 H1 和 H2 之间的通路上的路由器丢弃主机 H1 发出的 IP 包的情况下，TCP 客户端应用程序向服务器发出 TCP 连接建立请求；
- 3) 多个 TCP 客户端应用程序与 TCP 服务器应用程序建立 TCP 连接后，TCP 客户端与 TCP 服务器间传递数据；并模拟应用程序一次发送数据量大于链路数据包最大长度情况下的 TCP 报文的发送/接收。
- 4) 关闭 TCP 连接。

3、 实验内容

- 1) 恢复本课程实验二（“网络层实验”）中的网络环境，并使主机 H1 和主机 H2 在网络层连通。
- 2) 编写基于 TCP 套接字的 TCP 客户端应用程序和 TCP 服务器应用程序。
TCP 客户端程序和 TCP 服务器程序可以采用不同编程语言开发；TCP 服务器应用程序支持与多个 TCP 客户端应用程序同时建立 TCP 连接；TCP

连接建立后，可以在 TCP 连接上传递结构化消息，消息中的字段取值长度为可变长度。

- 3) 将 TCP 客户端应用程序和 TCP 服务器应用程序分别部署到主机 H1 和主机 H2 上。
- 4) 在未启动 TCP 服务器应用程序的情况下，在主机 H1 上启动 TCP 客户端应用程序，观测 TCP 客户端应用程序在 TCP 连接建立过程中的收发的 TCP 报文、TCP 连接建立结果。
- 5) 在启动 TCP 服务器应用程序的情况下，在路由器 R1 上设置丢弃主机 H1 发出的 IP 分组的防火墙规则，使得 TCP 客户端应用程序发出的 TCP 连接建立请求无法到达主机 H2，然后在主机 H1 上启动 TCP 客户端程序，观测 TCP 客户端应用程序在 TCP 连接建立过程中的收发的 TCP 报文、TCP 连接建立结果。
- 6) 删除在路由器 R1 中设置的丢弃主机 H1 发出的 IP 分组的防火墙规则，使得主机 H1 发出 IP 分组能够到达主机 H2。在主机 H2 启动 TCP 服务器应用程序，在主机 H1 上分别多次运行 TCP 客户端应用程序，在 TCP 服务器和 TCP 客户端间建立多条 TCP 连接。观测 TCP 连接建立的三次握手过程、观测 TCP 连接建立后 TCP 连接上 TCP 报文传递过程。
- 7) TCP 连接建立后，在客户端与 TCP 服务器间传递数据，观测应用程序一次发送数据量大于链路数据包最大长度情况下的 TCP 报文的发送/接收。
- 8) 分别先后关闭 TCP 客户端程序和 TCP 服务器程序，观测 TCP 连接拆除过程。

4、实验环境

- 1) Linux 系统主机。
- 2) Oracle VirtualBox 软件，软件下载网址：<https://www.virtualbox.org/>；
或 VMware Workstations 软件。
- 3) Wireshark 软件，软件下载网址：<https://www.wireshark.org/>

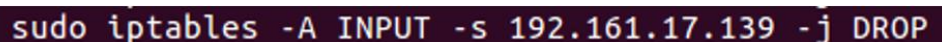
- 4) Linux 环境下 Socket 网络编程开发和运行环境：Java 程序开发环境 Eclipse、C/C++语言开发环境或所选编程语言开发环境。

5、 实验进度安排

本实验 4 学时。可安排 2 学时用于设计、编码实现实验用 TCP 客户端应用程序和 TCP 服务器应用程序；安排 2 学时完成实验任务中要求的四种情况下的 TCP 协议过程及 TCP 报文字段取值的分析。

6、 实验方法与步骤

- 1) 恢复本课程实验二（“网络层实验”）中的网络环境，并使主机 H1 和主机 H2 在网络层连通。
- 2) 设计并实现 TCP 客户端应用程序和 TCP 服务器应用程序，并将该客户端应用程序和服务器应用程序分别部署到主机 H1 和主机 H2 上。
- 3) 在未启动 TCP 服务器应用程序的情况下，在主机 H1、主机 H2 上分别启动 Wireshark 软件抓取主机 H1、H2 收发的 IP 分组；然后在主机 H1 上启动 TCP 客户端应用程序，观察 TCP 客户端应用程序的执行情况，分析 Wireshark 软件抓取的 TCP 连接建立请求消息和响应消息、以及这些 TCP 报文中的 SYN 比特位、RST 比特位的取值情况。
- 4) 在启动 TCP 服务器应用程序的情况下，在路由器 R1 上使用 iptables 命令防火墙规则，设置丢弃从主机 H1 发出的 IP 分组，使得 TCP 客户端应用程序发出的 TCP 连接请求消息无法到达主机 H2。图 1 为使用 iptables 命令设置过滤规则的示例。



```
sudo iptables -A INPUT -s 192.161.17.139 -j DROP
```

图 1. 使用 iptables 命令设置过滤规则示例

如果在路由器 R1 没有安装 iptables 命令程序，可以在关闭路由器 R1 的情况下，为路由器 R1 再配置一个类型为 NAT 或桥接类型的网卡，使得路由器可访问互联网，然后在启动路由器 R1 后，使用 `sudo apt-get install iptables` 命令在线安装 iptables 命令程序。

然后在主机 H1 上启动 TCP 客户端程序，观察 TCP 客户端应用程序的执行情况，分析 Wireshark 软件抓取的 TCP 连接建立请求消息、这些 TCP

报文中的 SYN 比特位、ACK 比特位、序列号字段的取值情况、以及重发的 TCP 连接请求分组的发送时间间隔。

图 2 所示的 TCP 连接建立中，从报文 5、6、10、13、14、15 为 TCP 客户端发出的连接请求报文；在发出报文 14 的同时，由于在路由器 R1 上使用 iptables 命令删除了丢弃 H1 发出的 IP 分组这一过滤规则，使得 TCP 客户端重发的连接请求报文（图 2 中的报文 15）能够到达 TCP 服务器，从而 TCP 服务器能够回复 TCP 连接响应消息（图 2 中的报文 16），从而接下来 TCP 客户端回复 TCP 连接响应消息（图 2 中的报文 17），这样完成了 TCP 连接建立的三次握手过程。

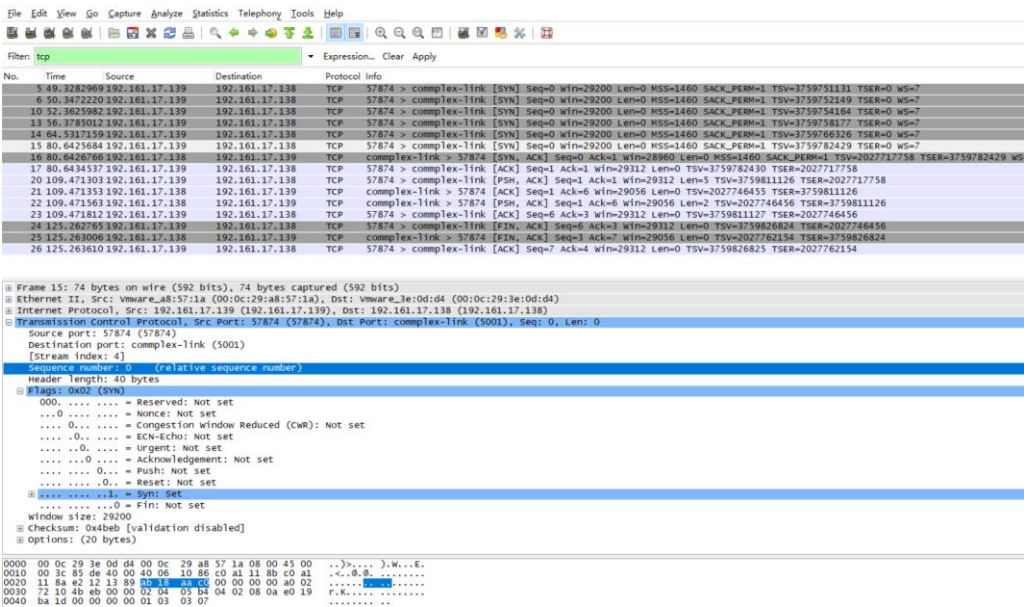


图 2. TCP 连接建立过程抓包示例

图 3 为使用 iptables 命令删除前述的过滤规则的命令示例。

```
sudo iptables -D INPUT -s 192.161.17.139 -j DROP
```

图 3. 使用 iptables 命令删除过滤规则示例

- 5) 观察 TCP 连接建立的三次握手过程中的 TCP 报文交互过程：分析携带 TCP 报文的 IP 分组的源 IP 地址、目的 IP 地址、协议（protocol）字段、IP 分组头长度（IHL）字段、IP 分组长度（Total Length）字段取值；分析 TCP 连接建立过程中的每个 TCP 报文的源端口号、目的端口号、SYN 比特位、ACK 比特位、FIN 比特位、序列号（Sequence Number）字段、

确认号（Acknowledgement Number）字段的取值。

- 6) 在主机 H1 上再启动一个 TCP 客户端应用程序建立与 TCP 服务器的 TCP 连接，观察新建立的 TCP 连接的三次握手过程中的 TCP 报文的字段取值与本实验步骤 5) 中的 TCP 连接建立的三次握手过程中的 TCP 报文字段的取值有哪些字段的取值是相同的。
- 7) TCP 连接建立后，触发 TCP 应用程序一次发送数据的数据量大于链路数据包最大长度，通过 Wireshark 抓包分析与此对应的 TCP 连接上 TCP 报文的收发，判断 TCP 协议是否对发送的数据进行了分段处理，分析这些 TCP 报文的序列号（Sequence Number）字段、确认号（Acknowledgement Number）字段的取值。
- 8) 关闭 TCP 客户端程序和 TCP 服务器应用程序，通过 Wireshark 抓包分析 TCP 连接拆除过程、TCP 连接拆除过程中交互的 TCP 报文、每个报文的 SYN 比特位、ACK 比特位、FIN 比特位、序列号（Sequence Number）字段、确认号（Acknowledgement Number）字段的取值。

7、实验管理规则及实验验收

实验完成后填写并提交实验报告。实验报告需给出 TCP 客户端应用程序和 TCP 服务器应用程序的设计和源程序，记录每种实验场景下的 TCP 客户端和 TCP 服务器之间交互的 TCP 报文过程、报文的字段的取值及分析。