# 实验二 TCP协议分析（2课时）

## — 实验目的

TCP（Transmission Control Protocol 传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议。本实验通过运用 Wireshark 对网络活动进行分析，观察TCP 协议报文，分析通信时序，理解 TCP 的工作过程，掌握 TCP 工作原理与实现；学会运用 Wireshark 分析 TCP 连接管理、流量控制和拥塞控制的过程，发现 TCP 的性能问题。

## 二 实验内容

观察 TCP 三次握手与四次挥手报文，注意报文收发过程中，双方 TCP 状态的变化。以本次捕获的报文为依据，分别画出本次 TCP 连接三次握手与四次挥手的时序图，结合 TCP 状态机，在双方各阶段标出对应的TCP 状态。选择其中一个TCP 报文，配合Wireshark 截图，分析该报文 TCP 首部各字段的定义、值及其含义。

1. TCP 数据流的追踪

TCP 数据流在 Internet 流量中占据了很大一部分。在这么多的 TCP 流量里，如何追踪数据流的蛛丝马迹呢？

Wireshark 分析功能中最实用的功能就是数据流的追踪了。数据流追踪，也就是说它能将各种流重组成容易阅读的格式。Wireshark 提供了TCP、UDP、SSL、HTTP 四种最常见数据流的追踪功能。

以一个简单的 TCP 交互为例，在捕获的流量数据里，鼠标点击任何一个 TCP 数据包（找到一个 TCP 数据包是非常容易的，协议字段已经表明各个数据包的类型），右键菜单中就会出现“追踪流”功能，再选择 TCP，Wireshark 就会显示这个 TCP 会话所有的数据包，并且列表在一个新的窗口中显示。

1. TCP 连接的建立

追踪任何一个 TCP 数据流，这个数据流开始的三个数据包都是其连接建立过程的三次握手。也可以使用 FLAGS 标志位进行检索，例如三次握手的第二个数据包非常特殊，SYN ACK 同时置位，可以利用这个特点发现一个三次握手过程。

实例：Wireshark 过滤显示SYN ACK 置位数据包

tcp.flags.syn == 1 && tcp.flags.ack == 1 ～flags 表示 TCP 标志字段

TCP 连接建立过程的三次握手：

No Time Source Destination Protocol Length Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 7.0908 192.168.1.103 | 223.119.144.197 | TCP | 66 | 54168 → 80 [SYN] Seq=0 |
| 8 | 7.1840 223.119.144.197 | 192.168.1.103 | TCP | 66 | 80→54168 [SYN, ACK] Seq=0 Ack=1 |
| 9 | 7.1841 192.168.1.103 | 223.119.144.197 | TCP | 54 | 54168 → 80 [ACK] Seq=1 Ack=1 |

1. TCP 连接的终止

在每个正常结束的 TCP 数据流，其尾部都是 TCP 连接终止时在客户机和服务器间的数据包交互。TCP 通信的双方都有权力发起TCP 连接的终止。也可以使用 FLAGS 标志位进行检索， 例如发起TCP 连接终止的数据包非常特殊，FIN 置位，可以利用这个特点发现一次终止过程。

实例：Wireshark 过滤显示 FIN 置位数据包

tcp.flags.fin == 1

下表中的数据包列表显示了一个由服务器方发起的终止。

|  |  |  |  |
| --- | --- | --- | --- |
| TCP 连接的终止： |  | | |
| No. Time Source | Destination | Protocol | Length Info |
| 1986 501.25 13.107.4.52 | 192.168.1.103 | TCP | 54 80→54168 [FIN, ACK] Seq=804 |
| 1987 501.25 192.168.1.103 | 13.107.4.52 | TCP | 54 54168→80 [ACK] Seq=155 |
| 1988 501.25 192.168.1.103 | 13.107.4.52 | TCP | 54 54168→80 [FIN, ACK] Seq=155 |
| 1989 501.34 13.107.4.52 | 192.168.1.103 | TCP | 54 80→54168 [ACK] Seq=805 |

服务器发送一个 FIN 标志置位的报文段，表示希望结束这次通信，很快，客户机给出了ACK 确认。在客户机的 ACK 包中，确认号为刚收到的 FIN 数据包加一。这时，服务器和客户机之间的连接进入半关闭状态。由于客户机和服务器都有各自的发送、接收缓冲区，半关闭状态的通信模式相信是容易理解的。这种状态下，客户机和服务器之间的通信只能单向进行了。

1. TCP 连接的重置

在理想的情况下，TCP 连接都是正常关闭的。但在现实中，TCP 连接会突然断掉。例如网络的瞬时拥塞、潜在的攻击者等等。在这些情况下，就可以使用 RST 标志置位的数据包， 指出连接被异常终止或拒绝连接请求。

实例：Wireshark 过滤显示RST 置位数据包

tcp.flags.reset == 1

一个使用RST 置位数据包的典型场景是访问一个不存在的网络服务。例如可以使用你的浏览器访问一个你熟悉的网站。在正常情况下，是可以浏览这个网站的网页的，但是只要修改这个网址的访问端口，例如原来访问的是 [www.mit.edu](http://www.mit.edu/)，现在使用 www.mit.edu:8080，访问就会失败，因为主机没有在 8080 端口监听连接请求。因此服务器返回了一个 RST 置位的数据包，通知客户机此次连接无效，也就是拒绝了你的连接请求。RST 除了 RST、ACK 标志置位以外，没有其他任何信息。

## 三 实验原理

作为TCP/IP 协议簇中的骨干，TCP 协议基于“尽力而为”的网络层为应用层提供可靠的进程间通信服务，具体地说是可靠的全双工的端对端字节流传输服务。在 TCP 的协议传输单元中(TCP 报文段，TCP Segment)，收发双方以字节为单位使用序号(Sequence Number) 明确收发的数据，使用 ACK 反馈 (Acknowledgment) 机制，实现端对端的可靠传输控制。

## 四 扩展实验（选做，有一定难度）

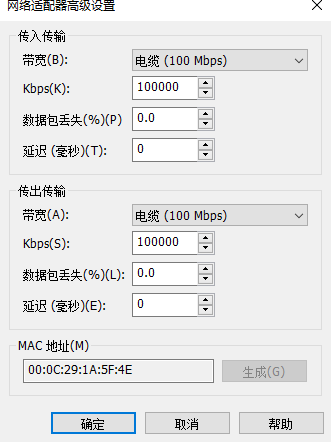
两台实验机本地相互连接，在实验机中仿真不同的网络条件，以便观察 TCP 的各种控制现象。使用 VMware Player 运行两台虚拟机，并通过“虚拟机设置-> 硬件-> 网络适配器-> 高级”（如图1）设置虚拟机的网卡传入/传出带宽、数据包丢失率、延迟等。

图1 虚拟机网卡设置

**软硬件配置：**

硬件：处于同一局域网的两台 PC 机 (可使用虚拟机也可使用物理机)。

软件：Ubuntu 系统(18.04 版)，预装wireshark、curl、vsftp、netwox、telnet、nmap 和iperf3。

**环境准备：**分别以PC1、PC2 作为TCP 的客户端与服务端；启动两台实验机后，可使用ping 进行连接性测试，也可使用 nmap 扫一下对方打开的端口，确保实验环境正常。

* + 1. 在 PC2 上启动一个简易的 web 服务器，创建index.html 文件为测试站首页，在 80

|  |  |  |
| --- | --- | --- |
| 命令 | 作用 | 参考 |
| KHEQPHKI | 配置网络 | JVVRU:／／OCP NKPWZFG PGV／KHEQPHKI |
| POCR | 网络扫描 | JVVRU:／／POCR QTI／OCP／\J／KPFGZ JVON |
| EWTN | 文本浏览器 | JVVRU:／／OCP NKPWZFG PGV／EWTN |
| YIGV | 下载 Web 文件 | YIGV <+2 ／<2CVJ#PF(KNG0COG |
| VE | 流量控制 | tc 命令手册 |
| KRVCDNGU | 防火墙配置 | JVVRU:／／OCP NKPWZFG PGV／KRVCDNGU |
| PGVYQZ | 网络工具 | JVVRU:／／UQWTEGHQTIG PGV／RTQLGEVU／PVYQZ／ |
| UU | Socket 状态 | UU –CVP |
| PGVUVCV | 显示网络状态 | PGVUVCV –CVP |
| YQPFGTUJCRGT | 网卡限速工具 | YQPFGTUJCRGT =网 口? =下载 速 率? =上行 速 率?  YQPFGTUJCRGT ENGCT =网口? |
| KRGTH3 | 网络性能分析 | JVVRU:／／KRGTH HT／ |

* + 1. 端口 启动一个简易web 服务器；打开新终端，键入 UU -VNP 查看当前主机打开的 TCP
    2. 连接，确认 80 端口处理监听状态。
    3. 在 PC1 上打开一个终端，启动抓包软件；再打开一个新终端，键入curl <PC2 的 IP> ；停止抓包，在 wireshark 过滤出 TCP 类型报文。观察首个 TCP 报文头， 并分析各段值代表的意义。如果想要关闭相对序号/确认号，可以选择Wireshark 菜单栏中的Edit*→*Preference*→*protocols*→*TCP，去掉Relative sequence number 勾选项。使用Wireshark 内置的绘制流功能，选择菜单栏中的 Statistics*→*Flow Graph，Flow Type 选择 TCP flows 可以直观地显示 TCP 序号和确认号是如何工作的。
    4. 观察 TCP 三次握手与四次挥手报文，注意报文收发过程中，双方 TCP 状态的变化。以本次捕获的报文为依据，分别画出本次 TCP 连接三次握手与四次挥手的时序图，结合 TCP 状态机，在双方各阶段标出对应的TCP 状态。选择其中一个TCP 报文，配合Wireshark 截图，分析该报文 TCP 首部各字段的定义、值及其含义。