暨南大学本科实验报告专用纸

课程名称 计算机网络实验 成绩评定_____

实验项目名称 传输控制协议（TCP） 指导老师 某某某

实验项目编号 6 实验项目类型 综合类 实验地点 N117

学生姓名 某某 学号 xxxxxxxx

学院 网络空间安全学院 系____ 专业 网络空间安全

实验时间 2020 年 11 月 18 日 晚 上～ 2020 年 11 月 18 日 晚 上

（一）实验目的

1. 掌握 TCP 协议的报文格式
2. 掌握 TCP 连接的建立和释放过程
3. 掌握 TCP 数据传输中编号与确认的过程
4. 掌握 TCP 协议校验和的计算方法
5. 理解 TCP 重传机制

（二）实验环境

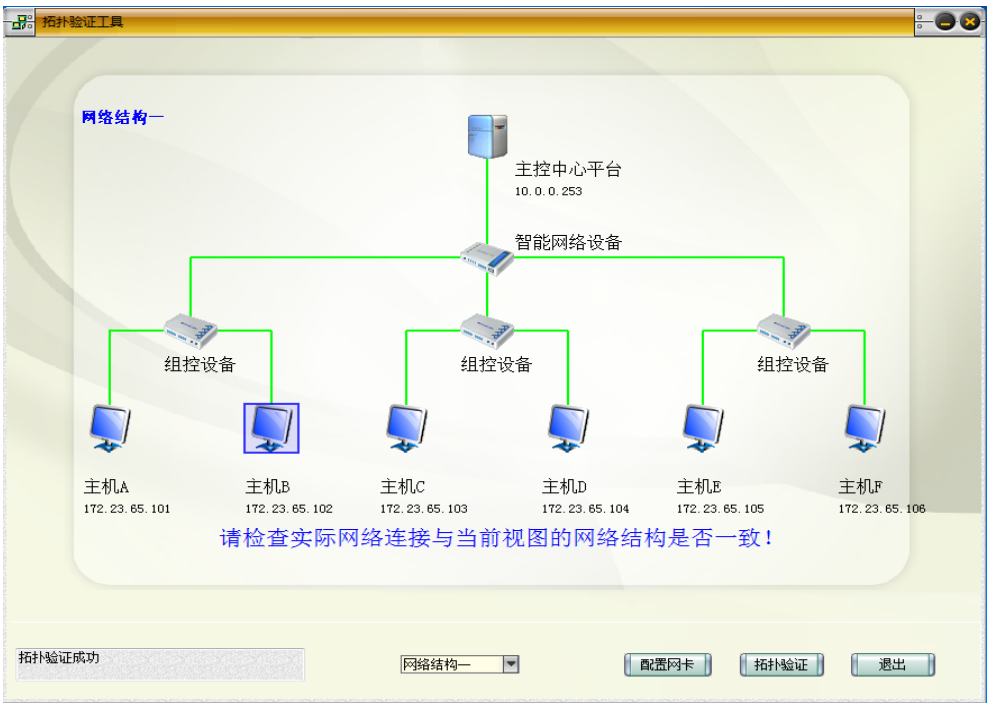


图 1：实验环境（该实验采用网络结构一，本机为主机 B）

（三）实验原理

1. TCP 协议简介

TCP（传输控制协议）协议是 TCP/IP 协议族中的面向连接的、可靠的传输层协议。TCP 与 UDP 不同，它允许发送和接收字节流形式的数据。为了使服务器和客户端以不同的速度发送和接收数据，TCP 提供了发送和接收两个缓冲区。TCP 提供全双工服务，数据同时能双向流动。通信的每一方都有发送和接收两个缓冲区，可以双向发送数据。TCP 在报文中加上一个递增的确认序列号来告诉发送端，接收端期望收到的下一个报文，如果在规定时间内，没有收到关于这个包的确认响应，则重新发送此包，这保证了 TCP 是一种可靠的传输层协议。

2. TCP 报文格式

TCP 报文的格式如下图所示：

<table>
<tr><td colspan="10">源端口（16 位）</td><td colspan="6">目的端口（16 位）</td></tr>
<tr><td colspan="16">序列号（32 位）</td></tr>
<tr><td colspan="16">确认号（32 位）</td></tr>
<tr><td colspan="2">首部长度
（4 位）</td><td colspan="2">保留
（4 位）</td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="6">窗口大小（16 位）</td></tr>
<tr><td colspan="10">校验和（16 位）</td><td colspan="6">紧急指针（16 位）</td></tr>
<tr><td colspan="16">选项和填充</td></tr>
</table>

图 2: TCP 报文格式

TCP 报文包括 20～60 字节的首部，接着是应用程序的数据部分。首部在没有选项时是 20 字节，而当有选项时长度会增加，但是最大不会超过 60 字节。

- 源端口：该字段定义了在主机中发送这个报文的应用程序端口号。
- 目的端口：该字段定义了数据报发往的主机中接收这个报文的应用程序的端口号。
- 序列号：该字段定义了指派给本报文第一个数据字节的一个序号。TCP 是流式传输协议，为了保证连通性，要在发送的每一个字节上编号。序号指定了这个序列中的哪一个字节是报文的第一个字节。在连接建立时，双方使用随机数产生器产生初始序号，通常每一方的初始序号都是不同的。
- 确认号：该字段定义了报文的接收端期望从对方接收的序号。如果报文的接收端成功地接收了对方发来的序号为 x 的报文，它就把确认号定义为 x+1。确认可以和数据一起发送。
- 首部长度：该字段指定 TCP 首部的长度，以 4 字节为单位。首部长度可以在 20～60 字节之间。因此，这个字段的值可以在 5 至 15 之间。
- 保留：这是 6 位字段，保留为今后使用。
- 控制：这个字段定义了 8 种不同的标志。如下图所示。在同一时间可设置一位或多位标志。
- 窗口大小：该字段定义对方必须维持的窗口值（以字节为单位）。这个字段的长度是 16 位，因此窗口值的最大长度是 65535 字节。这个值通常是作为接收窗口，并

由接收端来确定。这时，发送端必须服从接收端的决定。

- 校验和：该字段的校验范围包括伪首部、TCP 首部和 TCP 数据部分。
- 紧急指针：只有当紧急标志置位时，这个 16 位字段才有效，这时的报文中包括紧急数据。
- 选项：在 TCP 首部中可以有多达 40 字节的可选信息。

3. TCP 连接建立与释放

a. 建立连接

TCP 以全双工方式传送数据。当两个进程建立了 TCP 连接后，它们能够同时向对方发送数据。在传送数据之前，双方都要对通信进行初始化，得到对方的认可。

b. 三次握手

TCP 的连接建立过程叫做三次握手。服务器程序首先准备好接受 TCP 连接，这个过程叫做被动打开请求。这时，服务器的 TCP 就已准备好接受任何一台主机的 TCP 连接了。

客户程序发出 TCP 连接请求的过程叫做主动打开。然后服务器与客户端就开始三次握手过程，如下图所示（在图中客户端与服务器端各使用一条时间线，并给出每个阶段的几个重要字段，包括序号、确认号、控制标志以及非零的窗口值）。这个过程有以下 3 个步骤。

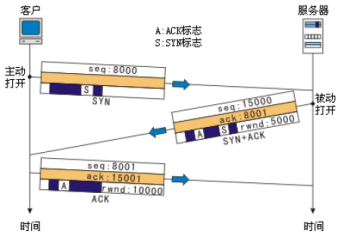


图 3: 使用三次握手建立连接

(1) 客户发送第一个报文，这是一个 SYN 报文，在这个报文中只有 SYN 标志置为 1。这个报文的作用是使序号同步。

(2) 服务器发送第二个报文，即 SYN+ACK 报文，其中 SYN 和 ACK 标志被置为 1。这个报文有两个目的。首先，它是一个用来和对方进行通信的 SYN 报文。服务器使用这个报文同步初始序号，以便从服务器向客户发送字节。服务器还使用 ACK 标志确认已从客户端收到了 SYN 报文，同时给出期望从客户端收到的下一个序号。另外，服务器还定义了客户端要使用的接收窗口的大小。

(3) 客户发送第三个报文。这仅仅是一个 ACK 报文。它使用 ACK 标志和确认号字段来确认收到了第二个报文。

c. 连接终止

通信双方中的任何一方都可以关闭连接。当一方的连接被终止时，另一方还可继续向对方发送数据。TCP 的连接终止有两种方式：三次握手和具有半关闭的四次握手。

d. **三次握手方式终止连接**

使用三次握手的 TCP 终止过程如下图所示：

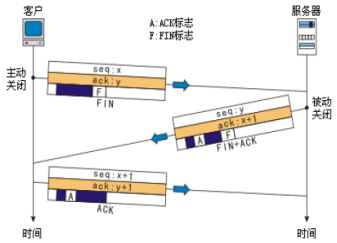


图 4: 使用三次握手的连接终止

(1) 当客户端想关闭 TCP 连接时，它发送一个 TCP 报文，把 FIN 标志位设置为 1。

(2) 服务器端在收到这个 TCP 报文后，把 TCP 连接即将关闭的消息发送给相应的进程，并发送第二个报文——FIN+ACK 报文，以证实从客户端收到了 FIN 报文，同时也说明，另一个方向的连接也关闭了。

(3) 客户端发送最后一个报文以证实从 TCP 服务器收到了 FIN 报文。这个报文包括确认号，它等于从服务器收到的 FIN 报文的序号加 1。

e. **半关闭的四次握手方式终止连接**

在 TCP 连接中，一方可以终止发送数据，但仍然保持接收数据，这就叫做半关闭。半关闭通常是由客户端发起的。图 7-7 描绘了半关闭的过程。客户发送 FIN 报文，半关闭了这个连接。服务器发送 ACK 报文接受这个半关闭。但是，服务器仍然可以发送数据。当服务器已经把所有处理的数据都发送完毕时，就发送 FIN 报文，客户端发送 ACK 报文给予确认。

在半关闭一条连接后，客户端仍然可以接收服务器发送的数据，而服务器也可以接收客户端发送的确认。但是，客户端不能传送数据给服务器。

（四）实验步骤

练习1 察看 TCP 连接的建立和释放

各主机打开工具区的“拓扑验证工具”，选择相应的网络结构，配置网卡后，进行拓扑验证，如果通过拓扑验证，关闭工具继续进行实验，如果没有通过，请检查网络连接。本练习将主机 A 和 B 作为一组，主机 C 和 D 作为一组，主机 E 和 F 作为一组。现仅以主机 A、B 为例，其它组的操作参考主机 A、B 的操作。

1. 主机 B 启动协议分析器捕获数据，并设置过滤条件（提取 TCP 协议）。主机 B 在命令行下输入：netstat -a -n 命令来查看主机 B 的 TCP 端口号。

2. 主机 A 启动 TCP 工具连接主机 B。

   主机 A 启动实验平台工具栏中的“TCP 工具”。选中“客户端”单选框，在“地址”文本框中填入主机 B 的 IP 地址，在“端口”文本框中填入主机 B 的一个 TCP 端口，点击［连接］按钮进行连接。

3. 察看主机 B 捕获的数据，填写下表。

字段名称	报文 1	报文 2	报文 3
序列号	1011123451	1374152147	1011123452
确认号	0	1011123452	1374152148
ACK	0	1	1
SYN	1	1	0

TCP 连接建立时，前两个报文的首部都有一个“最大字段长度”字段，它的值是多少？作用是什么？结合 IEEE802.3 协议规定的以太网最大帧长度分析此数据是怎样得出的。

1460； 由发送端指定，表明了能在网络上传输的最大的段尺寸； maximum segment size = MTU - 20（IP 首部）-20（TCP 首部）。

4. 主机 A 断开与主机 B 的 TCP 连接。

5. 察看主机 B 捕获的数据，填写下表。

字段名称	报文 4	报文 5	报文 6	报文 7
序列号	1051246324	1352416854	1352416854	1051246325
确认号	1352416854	1051246325	1051246325	1352416855
ACK	1	1	1	1
FIN	1	0	1	0

结合步骤 3、5 所填的表，理解 TCP 的三次握手建立连接和四次握手的释放连接过程，理解序号、确认号等字段在 TCP 可靠连接中所起的作用。

(1) 建立连接协议（三次握手）

* 客户端发送一个带 SYN 标志的 TCP 报文到服务器。这是三次握手过程中的报文 1。
* 服务器端回应客户端的，这是三次握手中的第 2 个报文，这个报文同时带 ACK 标志和 SYN 标志。因此它表示对刚才客户端 SYN 报文的回应；同时又标志 SYN 给客户端，询问客户端是否准备好进行数据通讯。
* 客户必须再次回应服务段一个 ACK 报文，这是报文段 3。

(2) 连接终止协议（四次挥手）。由于 TCP 连接是全双工的，因此每个方向都必须单独进行关闭。这原则是当一方完成它的数据发送任务后就能发送一个 FIN 来终止这个方向的连接。收到一个 FIN 只意味着这一方向上没有数据流动，一个 TCP 连接在收到一个 FIN 后仍能发送数据。首先进行关闭的一方将执行主动关闭，而另一方执行被动关闭。

* TCP 客户端发送一个 FIN，用来关闭客户到服务器的数据传送（报文段 4）。
* 服务器收到这个 FIN，它发回一个 ACK，确认序号为收到的序号加（报文段 5）。和 SYN 一样，一个 FIN 将占用一个序号。

* 服务器关闭客户端的连接，发送一个 FIN 给客户端（报文段 6）。
* 客户段发回 ACK 报文确认，并将确认序号设置为收到序号加 1(报文段 7)。

(3) 其他概念

* 序号：本报文段所发送的数据的第一个字节的序号。
* 确认号：期待收到对方下一个报文段的第一个数据字节的序号。
* 确认 ACK：占 1 位，仅当 ACK=1 时，确认号字段才有效。ACK=0 时，确认号无效。
* 同步 SYN：连接建立时用于同步序号。当 SYN=1，ACK=0 时表示：这是一个连接请求报文段。若同意连接，则在响应报文段中使得 SYN=1，ACK=1。因此，SYN=1 表示这是一个连接请求，或连接接受报文。
* 终止 FIN：用来释放一个连接。FIN=1 表示：此报文段的发送方的数据已经发送完毕，并要求释放运输连接。

思考问题

1. 为什么在 TCP 连接过程中要使用三次握手？如不这样做可能会出现什么情况。

(1) 防止已失效的连接请求报文段突然又传到服务器端，因而产生错误。

(2) 已失效的连接请求报文段突然又传到服务器端，因而产生错误。总之，三次握手完成两个重要的功能：既要双方做好发送数据的准备工作，双方都知道彼此已准备好，也要允许双方就初始序列号进行协商。这个序列号在握手过程中被发送和确认。如不这样做可能会出现死锁。

2. 解释 TCP 协议的释放过程？

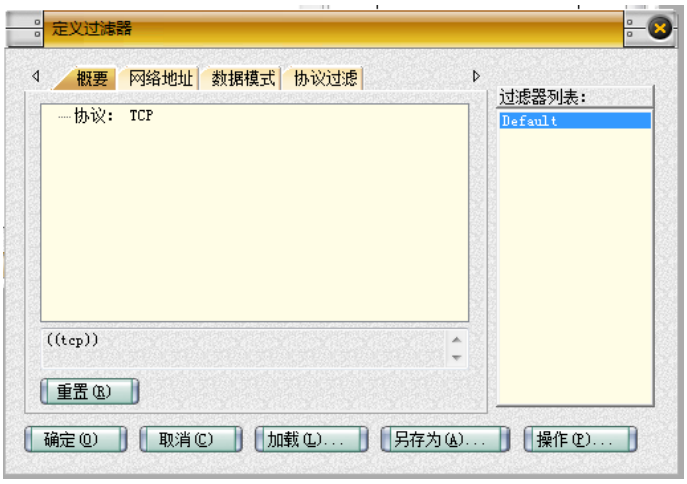
(1) 当客户端想关闭 TCP 连接时，它发送一个 TCP 报文，把 FIN 标志位设置为 1。

(2) 服务器端在收到这个 TCP 报文后，把 TCP 连接即将关闭的消息发送给相应的进程，并发送第二个报文——FIN+ACK 报文，以证实从客户端收到了 FIN 报文，同时也说明，另一个方向的连接也关闭了。

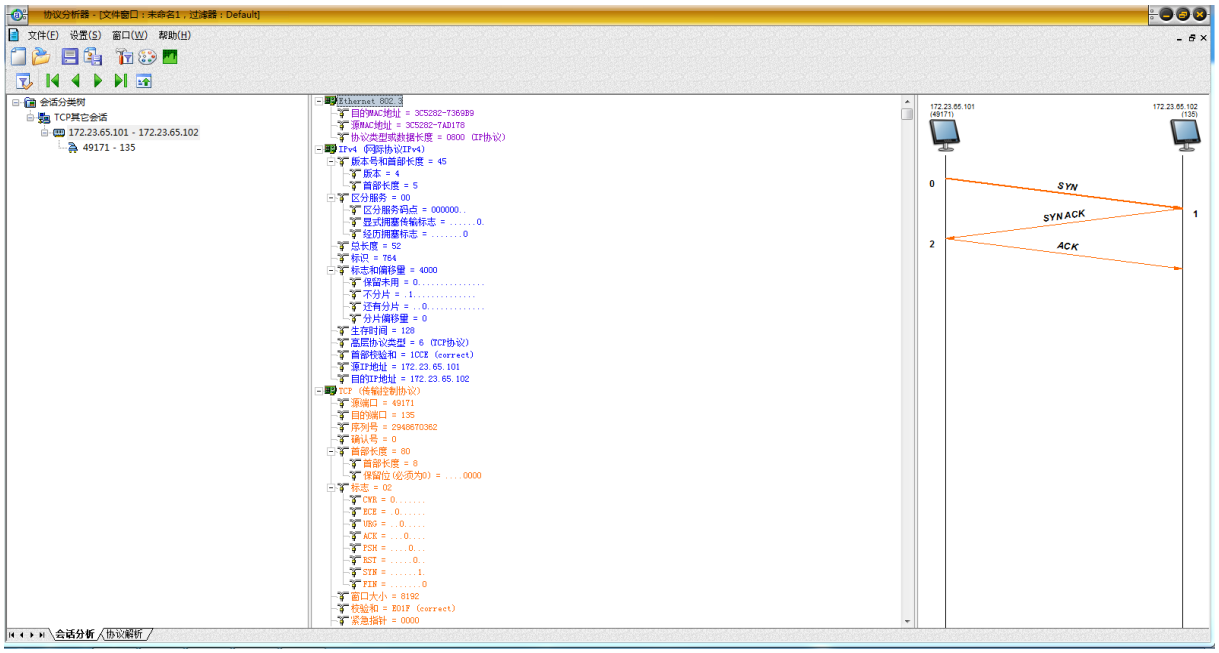
(3) 客户端发送最后一个报文以证实从 TCP 服务器收到了 FIN 报文。这个报文包括确认号，它等于从服务器收到的 FIN 报文的序号加 1。

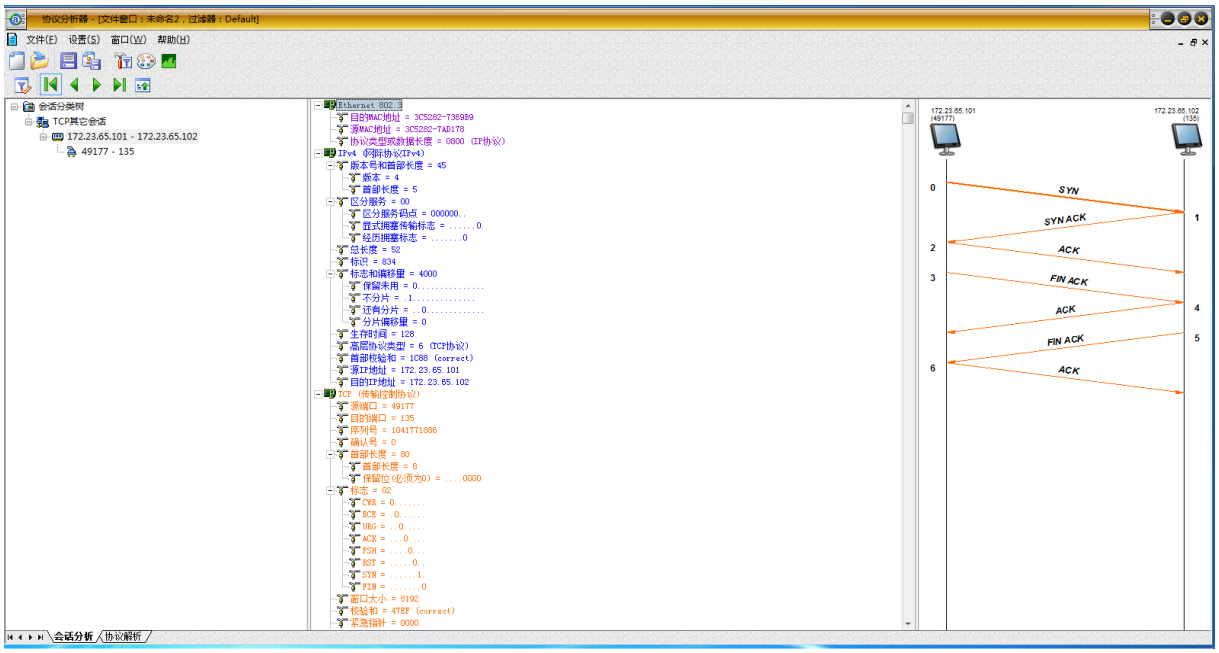
（五）实验结果与分析

练习1 察看 TCP 连接的建立和释放



分析：B 主机（本机）查看本机监听的 TCP 端口号并告诉主机 A，同时启动协议捕捉器捕捉 TCP 协议数据。





分析：B 主机（本机）查看捕捉到的数据，加深对 TCP 三次握手和四次挥手的理解。

（六）附录

由于对网络编程感兴趣，所以写了一个非常轻量的服务器，它能同时提供 TCP 服务和 UDP 服务。使用 epoll 多路 IO 机制，对于面向连接的 TCP 协议，可以每次客户端往套接字写数据时便创建一个线程去处理，或者维护一个线程池提高事件处理效率，这里为了简化逻辑不采用多线程方式，直接处理 epoll_wait 返回的事件。

简要说明

1. 小程序仍然是把数据中小写转化为大写
2. wrap.h 中已经包含了需要使用的所有头文件