实验报告：TCP 实验六

课程名称：计算机网络实践　**年级**：大二　**上机实践成绩**：

指导教师：章玥　**姓名**：邱吉尔

学号：10235101533　**上机实践日期**：**2024/12/23**

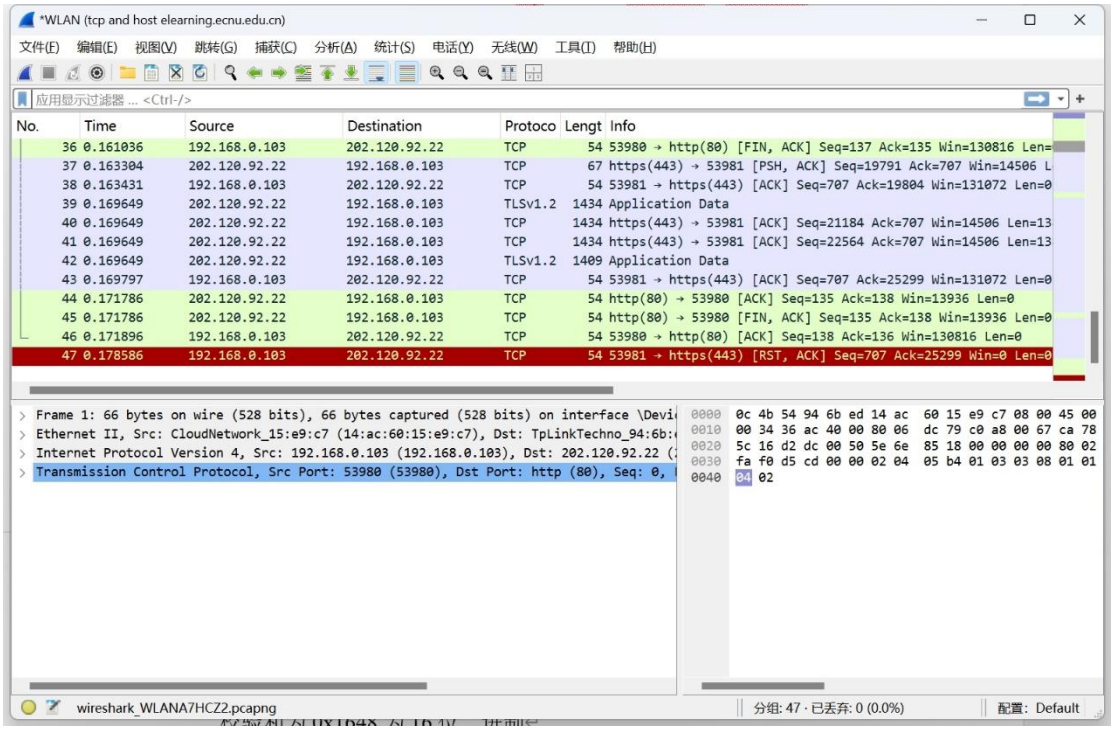
一、 目的

1. 熟悉使用 wireshark 软件进行抓取 TCP 数据包 ；
2. 分析抓取到的 TCP 数据包，掌握 TCP 数据包结构；
3. 掌握 TCP 数据包各字段的含义；
4. 掌握 TCP 连接建立和释放的步骤；
5. 掌握 TCP 数据传输过程；

二、 实验步骤

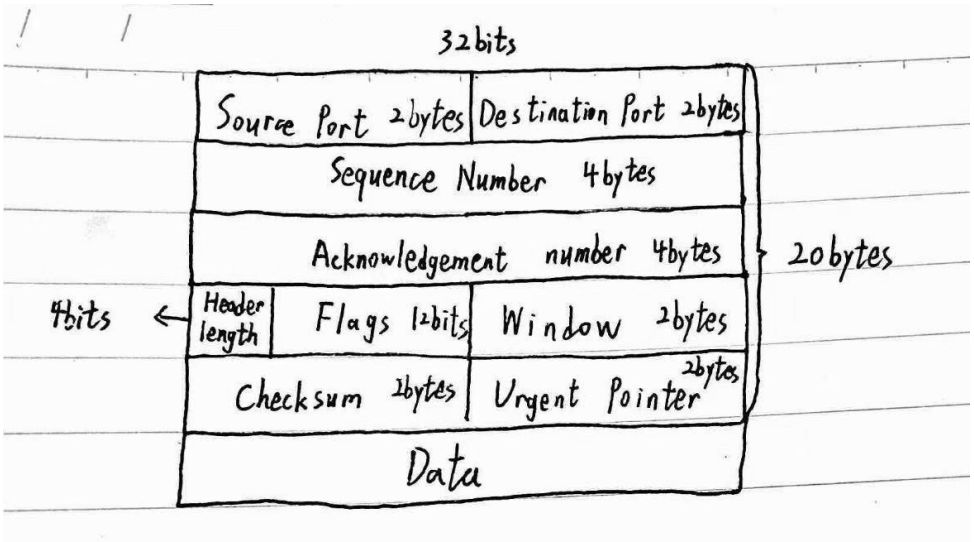
1. 以 http://elearning.ecnu.edu.cn/site/xiaoli/2016.jpg 为例，使用 wget 确认 URL 有效，或者使用你感兴趣的 URL；
2. 启动 Wireshark，在菜单栏的捕获->选项中进行设置，选择已连接的以太网，设置捕获过滤器为 tcp and host xx.xx.xx，xx.xx.xx 是要从中获取内容的服务器名称，如上述例子中的 elearning.ecnu.edu.cn；
3. 捕获开始后，重复第一步，重新发送请求；
4. 命令完成后，停止捕获。

实验抓包截图：



三、 选择一个 TCP 帧，观察其协议层：

1、根据你的理解，绘制 TCP 报文段的结构图（包括头部各字段的位置及大小）。

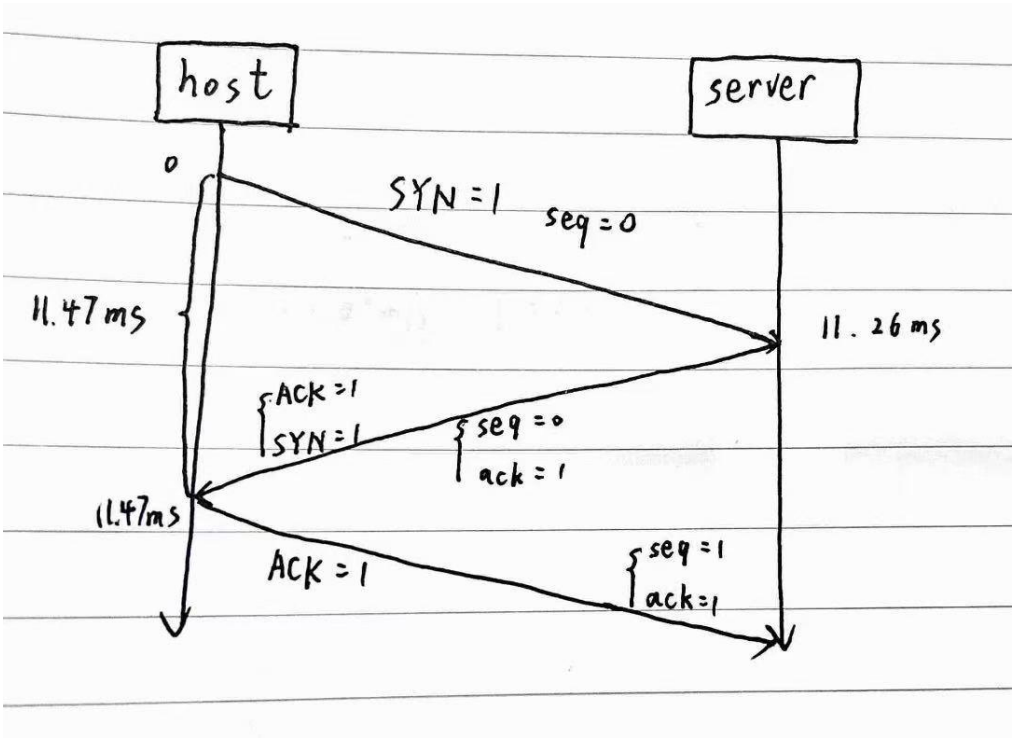


四、 在你捕获到的结果中，找到设置了 SYN 标志的 TCP 段及其后的数据包，完成以下问题：

1、 绘制三次握手的时序图，直到并包括建立连接后计算机发送的第一个数据包（HTTP GET 请求），包括

- ✧ **每个数据段的序列号和 Ack 标号；**

- ✧ **本地计算机发送或接收每个数据段的时间（以毫秒为单位）；**
- ✧ **本地计算机从发送 SYN 段到接收到 SYN-ACK 段的往返时间；**

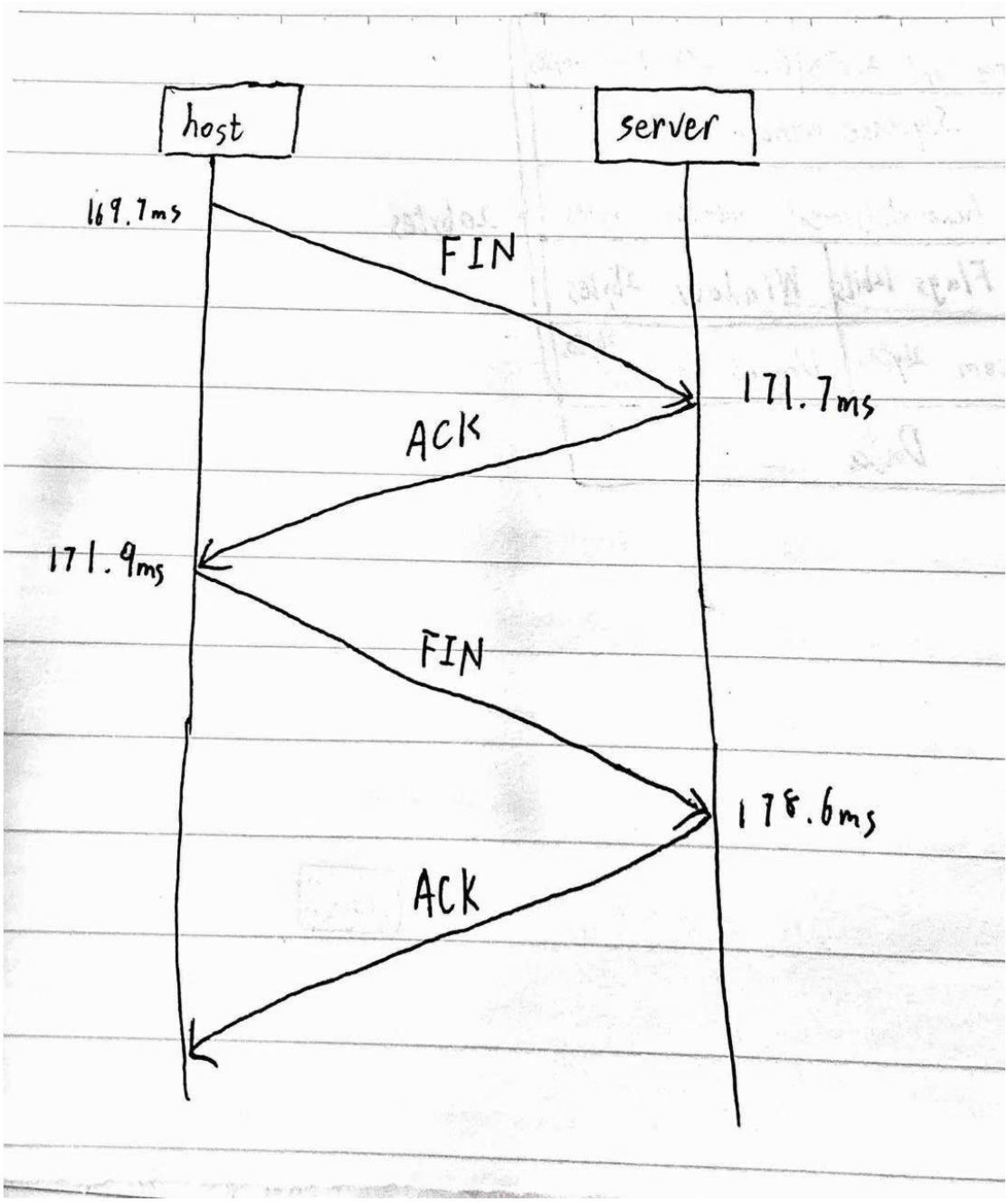


2、SYN 数据包上携带哪些 TCP 选项？

```
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale,
  TCP Option - Maximum segment size: 1460 bytes
  TCP Option - No-Operation (NOP)
  TCP Option - Window scale: 8 (multiply by 256)
  TCP Option - No-Operation (NOP)
  TCP Option - No-Operation (NOP)
  TCP Option - SACK permitted
```

- ➢ **Maximum segment size** 希望接收的最大报文长度 1460bytes
- ➢ **NOP** ： no operation 无意义字段，用于填充
- ➢ **Window Scale** 扩大窗口，实际窗口大小为 2^16 * 2^8 = 2^24 = 16MB
- ➢ **SACK permitted** 当发送序列中某个数据包丢失，可通过 SACK 报文通知发送方这个丢了，发送方于是重新传丢失的包，而不是全部重发

3、传输完成后，TCP 连接会以四次挥手或一端发送 RST 数据包的方式断开，同 1 一样，绘制 TCP 连接释放的时序图（从发出第一个 FIN 或 RST 到连接断开为止）。



五、 在“统计”菜单下，选择“ IO 图表”，以查看数据包速率。

调整过滤器为“tcp.srcport==80”仅查看下载数据包，重新绘图；
调整过滤器为“tcp.dstport==80”仅查看上传数据包，重新绘图；
通过你对数据传输的理解，回答以下问题：

1、实验中下载的大概速率为多少？（以 packets/s 和 bits/s 为单位）

约为 200packet/s，2Mbps

2、下载内容（即 TCP 有效负载）占下载率的百分比是多少？

134（payload 有效载荷） / 154（总长度） =87.01%

3、实验中上传的大概速率为多少？（以 packets/s 和 bits/s 为单位）

约为 120packet/s ，60Kbps

4、如果最近从服务器收到的 TCP 数据段的序列号是 X，那么下一个发送 TCP 报文中的 Ack 号是多少？

Ack = X + 这个 TCP 报文的长度 segment length

六、 在完成本实验后继续探索 TCP 协议：

1. 探索 TCP 的拥塞控制和经典 AIMD 策略。

- **TCP 拥塞控制：**拥塞窗口 cwnd(congestion window)：发送方维护一个状态变量(即拥塞窗口)，大小取决于网络的拥塞程度且动态变化。 发送方自己的发送窗口=拥塞窗口； 如果要考虑接收方的接受能力，发送窗口可能<拥塞窗口。

  原则：只要网络没有出现拥塞(发送方没有按时收到 ACK 确认报文)，就增大窗口； 若出现拥塞减少拥塞窗口。

- **经典 AIMD 策略：**“和式增加、积式减少” “成瘾性增加，乘法减少”；

  这样能使得双方速率逐渐成为均分的形态。

  AIMD 策略的效果——收敛、公平。

2. 更深入地探索 TCP 的可靠性机制。捕获包括段丢失的 TCP 连接，查看什么触发重新传输以及何时触发，另外查看往返时间估算工具。

1) 超时重传机制：超过时间还未收到 ACK 则重新发送。
2) 校验和：校验和错误时重传。
3) 序列号：序列号能够确认缺少了哪个位置的数据，保证按序到达，同时筛除重复数据。
4) 确认应答机制：ACK 标志位=1 时，检查 Ack=U+1，说明前 U 个数据包都正确接收。

3. 查看包括 SACK 在内的选项的使用以了解详细信息。

TCP 的一个选项，允许 TCP 单独确认非连续片段，用于告知真正丢失的包，只重传丢失的片段。 options 字段中：SACKpermitted 当发送序列中某个数据包丢失，可通过 SACK 报文通知发送方这个丢了，发送方于是重新传丢失的包，而不是全部重发。