

****

信息学院软件工程系

《计算机网络》实验报告

**题　　目 实验4观察 TCP 报文段并 侦听析 分析 FTP 协议**

**班　　级 软件工程2018级4班**

**姓　　名 余嘉炜**

**学　　号 21620182203533**

**实验时间 2020年3月25日**

**2020 年 3 月 25 日**

# 实验目的

用 PCAP 库侦听并解析 FTP 口令

# 实验环境

WINDOWS10, C++

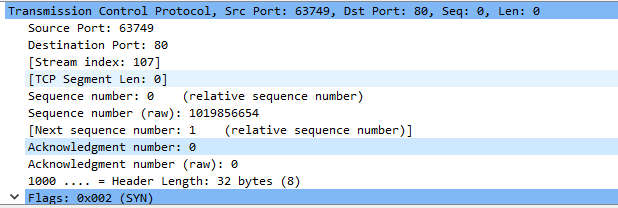
# 实验结果

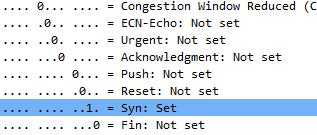
1. 用wireshark监听TCP包

三次握手过程



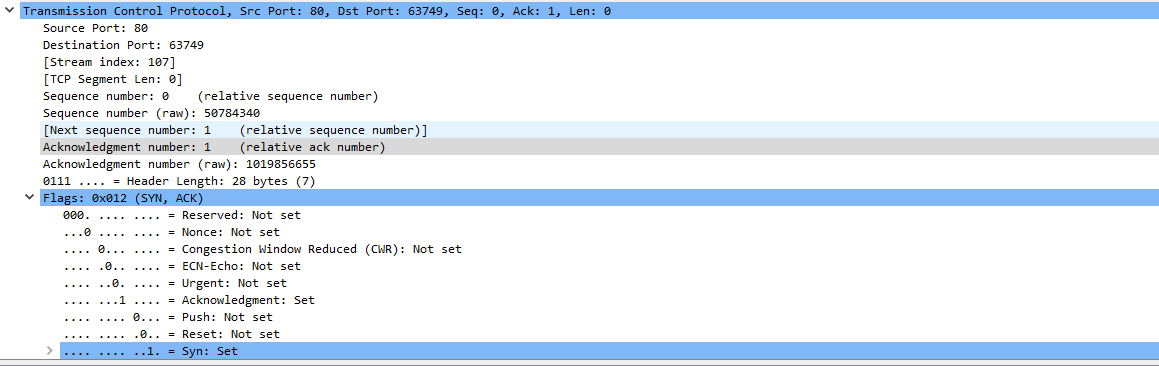
第一次握手： 源端口63749向目的端口80发送连接申请，指定序列号为S=1019856654，窗口大小为64240，此时SYN为1





第二次握手：

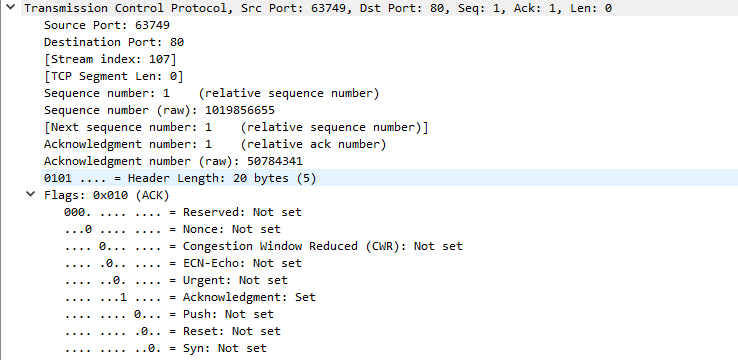
源端口80向目的端口63749发送确认，指定序列号为S=50784340，acknumber为1019856655，窗口大小为14600，此时ACK和SYN均为1





第三次握手：

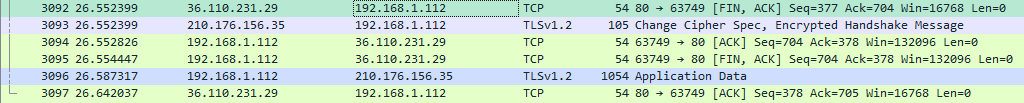
源端口63749向目的端口80发送确认信号，指定序列号为S=1019856655，窗口大小为517，此时ACK为1



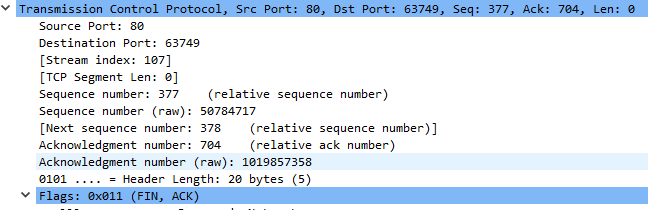


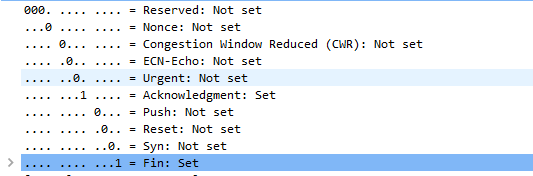
自此完成三次握手过程。

四次握手断开连接：

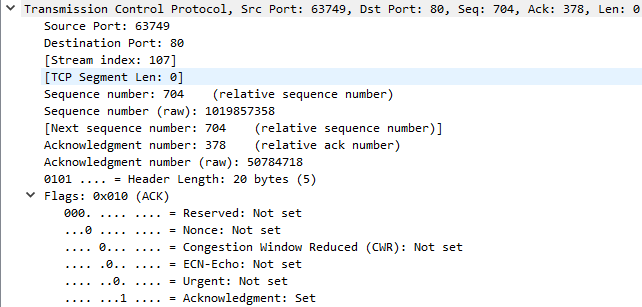


1. 服务器先断开ACK=1,FIN=1

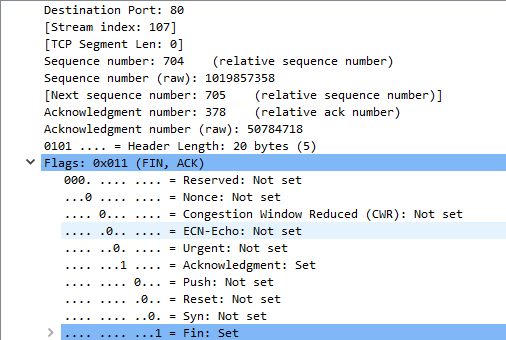




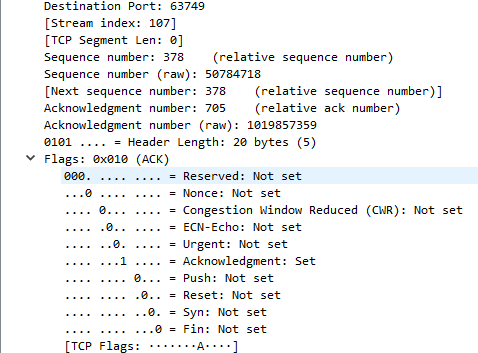
1. 客户端确认，ACK=1



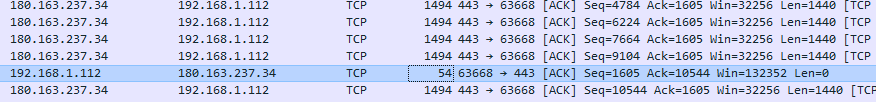
（4）客户端向服务端发送断开连接：ACK=1,FIN=1



(4)服务器端发送确认 ACK=1

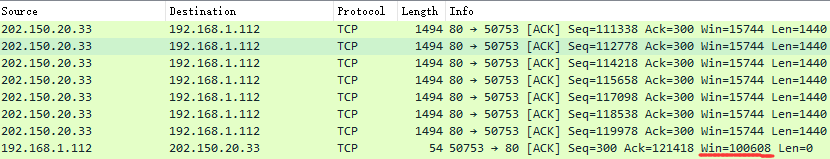


TCP窗口机制：

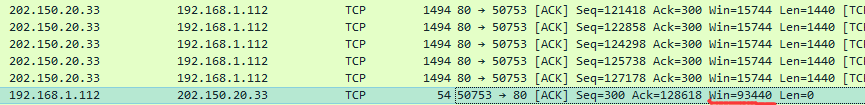


如下截图可以看到ack num = seq num + len

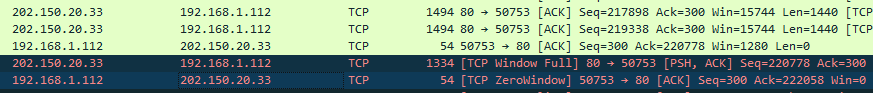




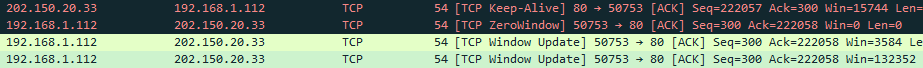
再连续发送大量数据后，窗口减小



降为0



重新启动：



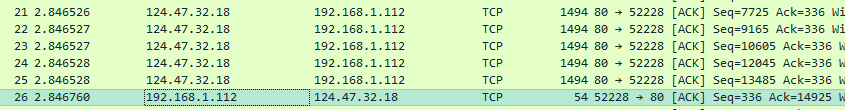
报文丢失情况下普通重传机制：第二行略过

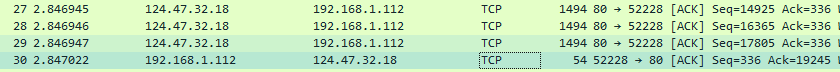


拥塞控制：

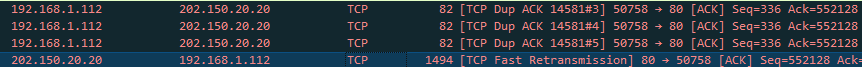
无法展示发送窗口的大小。

慢开始：发送方发送数据量加大

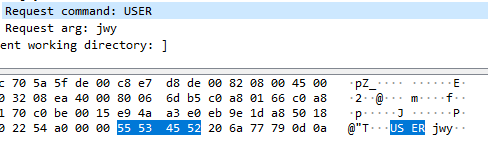




快速重传：上面的ACK等于下面的Seq



2.分析FTP包，user所在位置，前面共54byte数据



用户名所在位置，遇到0d即13则停止



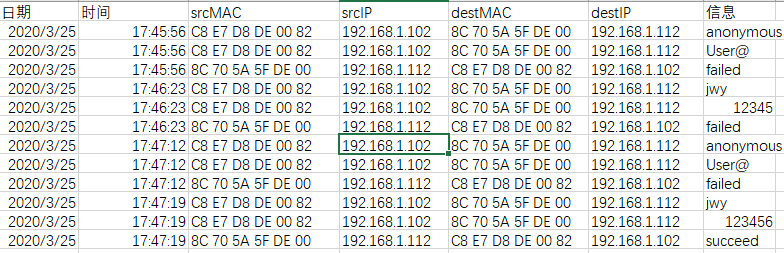
注：实际账号为：jwy 密码：123456

3. 第一次登陆输入错误的密码，日志如下

第二次登陆连接成功



全截图：



# 实验总结

wireshark与有道翻译会冲突，使用wireshark时需关闭有道翻译。在使用wireshark抓包时，发送端的发送窗口大小无法展示，一般需要通过接收端的窗口值去推测。TCP拥塞控制机制需要合理抓包，此处使用下载文件来表现拥塞。