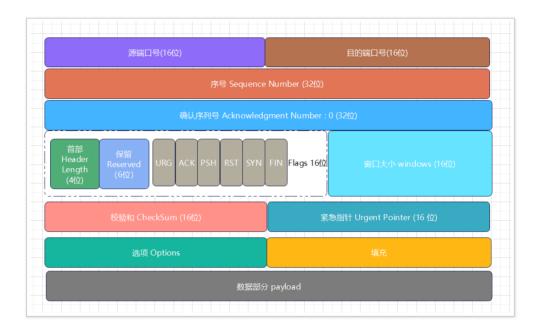
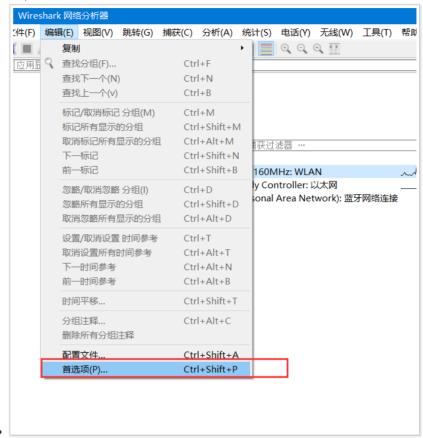
## 5.2 tcp 协议与 wireshark 抓包分析\_物联网 / 嵌入式工程师 - 慕课网

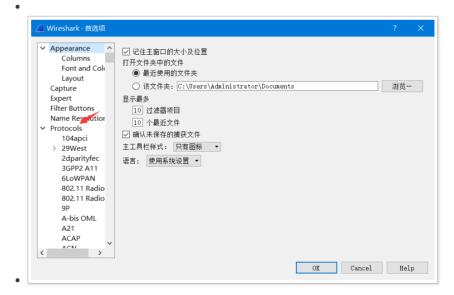
44 慕课网慕课教程 5.2 tcp 协议与 wireshark 抓包分析涵盖海量编程基础技术教程, 以图文图表的形式, 把晦涩难懂的编程专业用语, 以通俗易懂的方式呈现给用户。

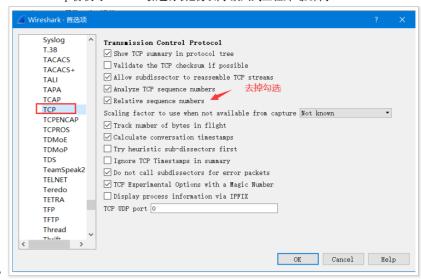


- 源端口号: 发送方使用的端口号
- 目的端口号: 接收方使用的端口号
- 序号: 数据包编号, tcp 协议为每个数据都设置编号, 用于确认是否接收到相应的包
- 确认序列号:使用 tcp 协议接收到数据包,会根据这个接收到这个数据包编号发送一个应答包, 一般为上一次的数据包的编号加上长度,如果是 SYN 或者 FIN,则是上一次的数据包的编号 加 1
- 标志位: 用于标识
  - URG:表示紧急指针是否有效
  - ACK:表示确认号是否有效。称携带 ACK 标志的 tcp 报文段位确认报文段
  - PSH:提示接收端应用程序应该立即从 tcp 接受缓冲区中读走数据,为接受后续数据 腾出空间(如果应用程序不将接收的数据读走,它们就会一直停留在 tcp 缓冲区中)
  - \*\*RST: \*\* 表示要求对方重新建立连接。携带 RST 标志的 tcp 报文段为复位报文 段。
  - \*\*SYN: \*\* 表示请求建立一个连接。携带 SYN 标志的 tcp 报文段为同步报文段。
  - FIN: 表示通知对方本端要关闭连接了。携带 FIN 标志的 tcp 报文段为结束报文段。
- 窗口大小: 用于 tcp 进行流量控制, 这里的窗口用于向发送端说明当前 tcp 接收缓冲区还能存储的数据大小
- 校验和: 用于接收端用于校验接收的数据是否正确, 由发送端进行填充,
  - 计算校验和主要包括 tcp 协议头与数据区

- 校验的方式为 CRC 校验
- tcp 三次握手过程
  - 由客户端给服务器发送 SYN 标志的连接请求包
  - 服务器收到请求包,发送给客户端一个 SYN + ACK 应答包
  - 当客户端收到服务器的应答包,则给服务器发送一个应答包
- 使用 wireshark 抓取 tcp 三次握手数据包
  - step 1: 设置 wireshark 使用绝对数据包编号







•

• step 2:选择网卡,并设置 tcp 端口过滤,开启抓包



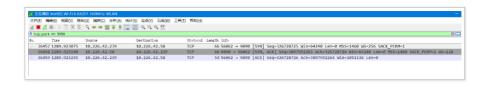
• step 3:启动服务器

```
ben@ubuntu:~/class/week15/codes/part5/A02tcp$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
         inet 10.226.42.58 netmask 255.255.240.0 broadcast 10.226.47.255
        inet6 fe80::f0f9:daec:7c3b:83c3 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:16:35:99 txqueuelen 1000 (Ethernet)
        RX packets 1125384 bytes 1298554622 (1.2 GB)
Terminal RX errors 0 dropped 0 overruns 0 frame 0 TX packets 434521 bytes 39576756 (39.5 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 4079008 bytes 1968708147 (1.9 GB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4079008 bytes 1968708147 (1.9 GB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
ben@ubuntu:~/class/week15/codes/part5/A02tcp$ ./server 10.226.42.58 9090
```

• step 4: 启动网络调试助手, 连接服务器



• step 5: 查看抓包结果



- tcp 三次握手数据包分析
  - 第一次握手: 客户端给服务器发送 SYN 数据包



- Flags 标志 设置为 SYN,数据包编号为 326728725
- 第二次握手:服务器给客户端发送 SYN + ACK 的数据包

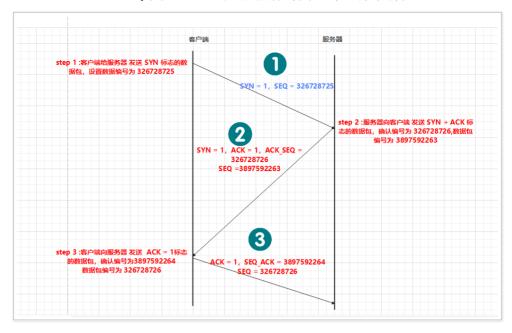


- Flags 标志: 设置为 SYN + ACK
- 数据包应答编号为 326728726, 相当是 SEQ\_SYN + 1
- 数据包编号为 3897592263

•

• 第三次握手:客户端给服务器发送 ACK 数据包

- Flags 标志 : ACK
- 数据包应答编号为 3897592264, 是第二个数据包的编号加 1
- 数据包编号为 326728726
- 总体过程图如下:



- tcp 四次挥手过程 (以客户端主动断开为例)
  - 由客户端向服务器发送 FIN 标志断开请求包
  - 服务器接收到 FIN 数据包,则给客户端发送一个 ACK 应答包
  - 由服务器向客户端发送 FIN 数据包
  - 客户端接收到 FIN 数据包,则给服务端发送一个 ACK 应答包
- wireshark 抓取 tcp 四次挥手
  - step 1: 客户端断开连接, 然后服务器断开



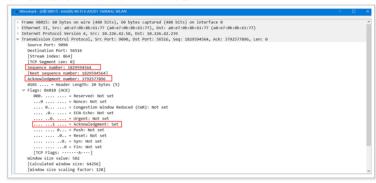
• step 2 : 查看 wireshark

98014 2800.048709 10.226.42.239 10.226.42.58 TCP 54.56516 + 9090 [FIN, ACK] Seq-3792577895 Ack-1829594564 Win-185136 Len-0
98015 2800.096947 10.226.42.239 TCP 60.9090 + 50516 [Ack] Seq-1829594564 Ack-3792577896 Win-64256 Len-0
98042 2806.21715 10.226.42.239 10.226.42.58 TCP 54.56516 + 9090 [ACK] Seq-182959456 Ack-3792577896 Win-64256 Len-0
98043 2806.217196 10.226.42.239 10.226.42.58 TCP 54.56516 + 9090 [ACK] Seq-3792577896 Ack-1829594565 Win-185136 Len-0

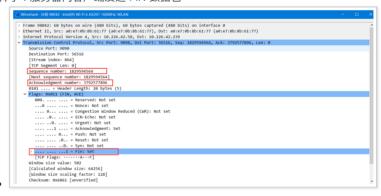
- 四次挥手分析:
  - 第一次挥手:客户端向服务器发送 FIN 数据包

5.2 tcp 协议与wireshark抓包分析\_物联网/嵌入式工程师-慕课网

- Flags 标志: FIN (这里的 ACK 标志不用关注, 为前一个数据包的 ACK)
- 数据包编号: 3792577895
- 第二次挥手:服务器向客户端发送 FIN 的应答包 ACK

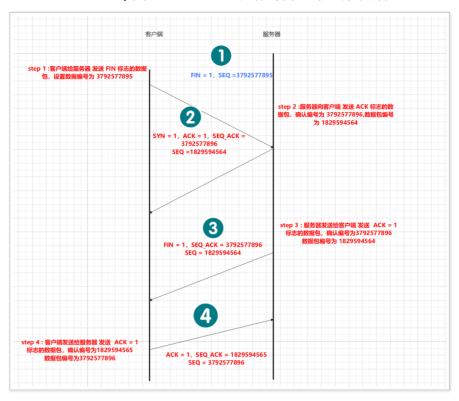


- Flags 标志: ACK
- 数据包编号: 1829594564
- 数据包应答编号: 3792577896
- 第三次挥手:服务器向客户端发送 FIN 数据包



- Flags 标志:FIN
- 数据包编号:1829594564
- 数据包应答编号: 3792577896
- 第四次挥手:客户端向服务器发送 ACK

- Flags 标志: ACK
- 数据包编号:3792577896
- 数据包应答编号: 1829594565



- 1. 理解 tcp 协议格式,并说明每个部分的含义
- 2. 使用 wireshark 抓取 tcp 三次握手与四次挥手的数据包,根据三次握手与四次挥手的步骤来分析数据包

全文完

本文由 简悦 SimpRead 优化,用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta,点击查看详细说明



