作业四：使用 Wireshark 分析数据段

1711342 李纪

2019 年 11 月 17 日

摘要

这是我的**作业四**的实验报告，请老师查阅，谢谢。

关键字：HTTP、Wireshark、TCP

目录

1 作业简介 **3**

2 测试环境简介 **3**

3 TCP 数据段整体分析 **3**

4 特定 TCP 数据段分析 **4**

4.1 第 23 号数据段：建立连接的三次握手——第一次握手 4

4.2 第 25 号数据段：建立连接的三次握手——第二次握手 6

4.3 第 32 号数据段：数据传输——服务器的 HTTP 应答 8

4.4 第 571 号数据段：数据传输——TCP Keep-Alive 10

4.5 第 781 号数据段：连接关闭——服务器关闭连接 12

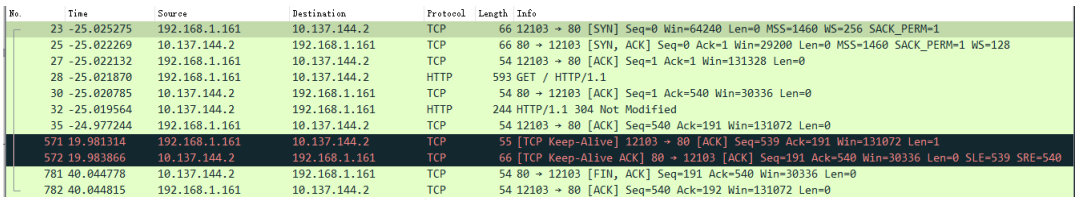
1 作业简介

通过 HTTP 访问某个网页，使用 Wireshark 对整个过程中的数据段进行捕获，分析 TCP 连接建立、数据传输、连接关闭的全过程，至少对其中 5 个典型的 TCP 数据段进行详细分析，给出界面截图，并同时提交捕获文件。

2 测试环境简介

- Wireshark 版本：3.0.6
- 用于测试的网页：http://oslab.mobisys.cc/
- 本机测试网卡 IPv4 地址：192.168.1.161
- 测试网页对应的 IPv4 地址：10.137.144.2
- **备注：这个网页是我们操作系统课的官方网站，属于宫晓利老师，服务器在校园网内。仅限校园网内访问。**

建立连接到连接关闭的整个 TCP 流参考图 1 。



No.	Time	Source	Destination	Protocol	Length	Info
23	-25.025275	192.168.1.161	10.137.144.2	TCP	66	12103 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	-25.022269	10.137.144.2	192.168.1.161	TCP	66	80 → 12103 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
27	-25.022132	192.168.1.161	10.137.144.2	TCP	54	12103 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
28	-25.021870	192.168.1.161	10.137.144.2	HTTP	593	GET / HTTP/1.1
30	-25.020785	10.137.144.2	192.168.1.161	TCP	54	80 → 12103 [ACK] Seq=1 Ack=540 Win=30336 Len=0
32	-25.019564	10.137.144.2	192.168.1.161	HTTP	244	HTTP/1.1 304 Not Modified
35	-24.977244	192.168.1.161	10.137.144.2	TCP	54	12103 → 80 [ACK] Seq=540 Ack=191 Win=131072 Len=0
571	19.981314	192.168.1.161	10.137.144.2	TCP	55	[TCP Keep-Alive] 12103 → 80 [ACK] Seq=539 Ack=191 Win=131072 Len=1
572	19.983866	10.137.144.2	192.168.1.161	TCP	66	[TCP Keep-Alive ACK] 80 → 12103 [ACK] Seq=191 Ack=540 Win=30336 Len=0 SLE=539 SRE=540
781	40.044778	10.137.144.2	192.168.1.161	TCP	54	80 → 12103 [FIN, ACK] Seq=191 Ack=540 Win=30336 Len=0
782	40.044815	192.168.1.161	10.137.144.2	TCP	54	12103 → 80 [ACK] Seq=540 Ack=192 Win=131072 Len=0

图 1: http://oslab.mobisys.cc/

3 TCP 数据段整体分析

我们可以将 TCP 流的全过程分为三个部分：连接建立、数据传输、连接关闭。由图 1 中可看出各部分含有的数据段序号如下：

- 连接建立：23，25
- 数据传输：27，28，30，32，35，571，572
- 连接关闭：781，782

4 特定 TCP 数据段分析

4.1 第 23 号数据段：建立连接的三次握手——第一次握手

第 23 号数据段的详细信息参考图 2 。

这个数据段的源端口号为 12103，这是一个由客户端自行决定的端口。目的端口号为 80[1]，这是一个 HTTP 使用的默认端口。序号为 0，确认号也是 0，因为这是整个 TCP 流中的第一个 TCP 数据段。首部长度为 32 字节，而不是 20 字节，说明这个 TCP 数据段中包含 12 字节的选项。在标志位中，仅有 SYN 位为 1，其余标志位为 0，SYN 位为 1 是建立连接请求的特征之一。接收窗口为 64240，代表客户端愿意接受的字节数量为 64240 字节。校验和为 0xe500，用于校验。紧急数据指针为 0。

选项部分：

第一个选项为最大报文段长度，用于客户端与服务器协商最大报文长度（MSS），由此可知客户端现在通知服务器自己的最大报文长度（MSS）为 1460 字节。第二个选项为无操作。第三个选项为 TCP 窗口比例，用于增加允许的接收窗口大小，使其超过其以前的最大值 65535 字节。第四个选项为无操作。第五个选项为无操作。第六个选项为允许选择确认。用于结合选择确认机制以缓存乱序段，表明客户端支持 SACK。[1]

[1]TCP 的 80 端口是一个由 IANA 官方指定的提供超文本传输协议（HTTP 协议）使用的端口

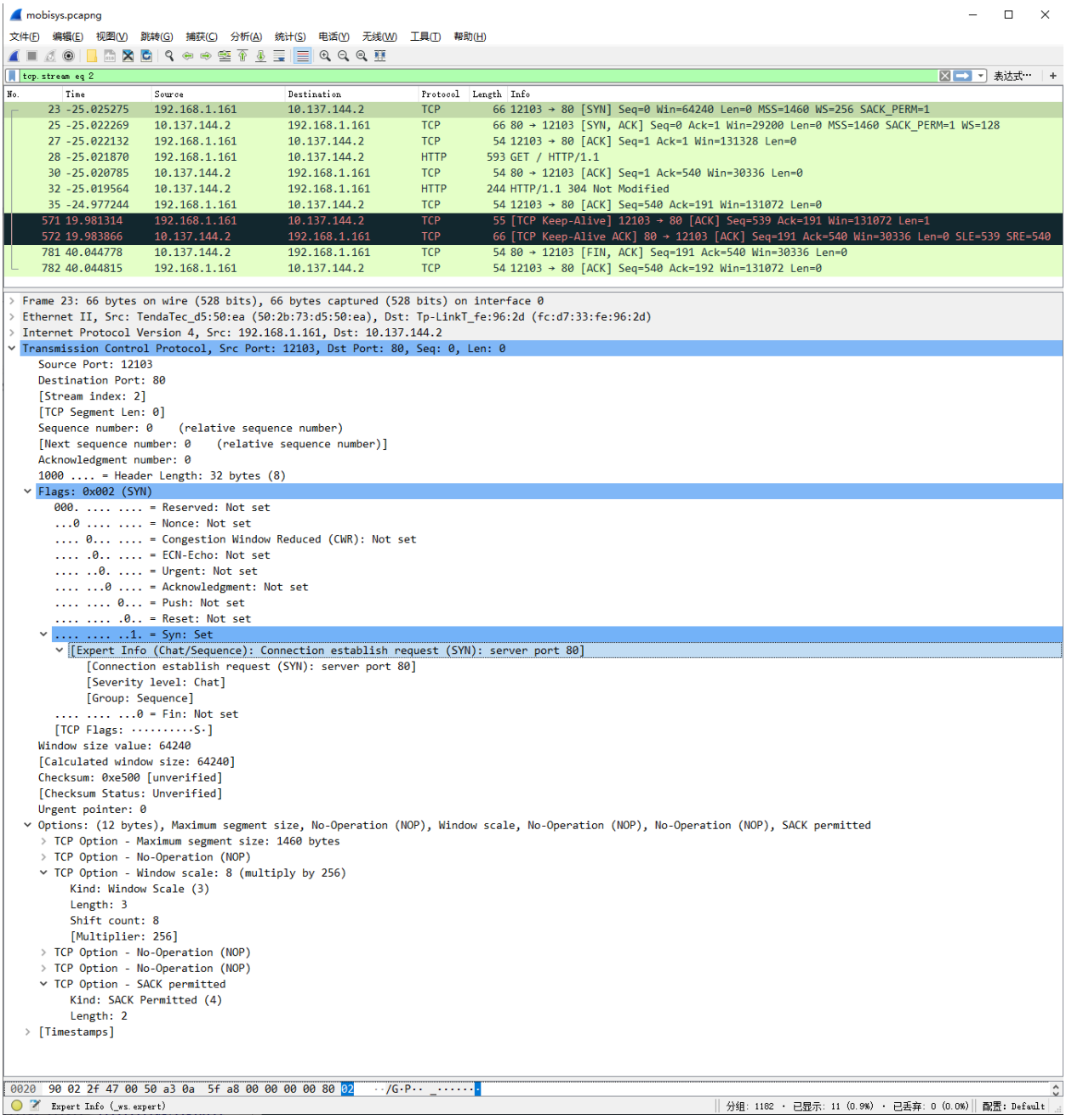


图 2: 第 23 号数据段：建立连接的三次握手——第一次握手

4.2 第 25 号数据段：建立连接的三次握手——第二次握手

第 25 号数据段的详细信息参考图 3 。

这个数据段的源端口号为 80，这是一个 HTTP 使用的默认端口。目的端口号为 12103，为客户端指定的端口。序号为 0，确认号是 1，因为第 23 号数据段的序号为 0，服务器收到了 23 号数据段。首部长度为 32 字节，而不是 20 字节，说明这个 TCP 数据段中包含 12 字节的选项。在标志位中，仅有 ACK 位以及 SYN 位为 1，其余标志位为 0，ACK 位为 1 表明确认字段有效，SYN 位为 1 是建立连接请求的特征之一。接收窗口为 29200，代表服务器愿意接受的字节数量为 29200 字节。校验和为 0x9b5f，用于校验。紧急数据指针为 0。

选项部分：

第一个选项为最大报文段长度，用于客户端与服务器协商最大报文长度（MSS），由此可知服务器现在通知客户端自己的最大报文长度（MSS）为 1460 字节。第二个选项为无操作。第三个选项为 TCP 窗口比例，用于增加允许的接收窗口大小，使其超过其以前的最大值 65535 字节。第四个选项为无操作。第五个选项为无操作。第六个选项为允许选择确认。用于结合选择确认机制以缓存乱序段，表明服务器支持 SACK。

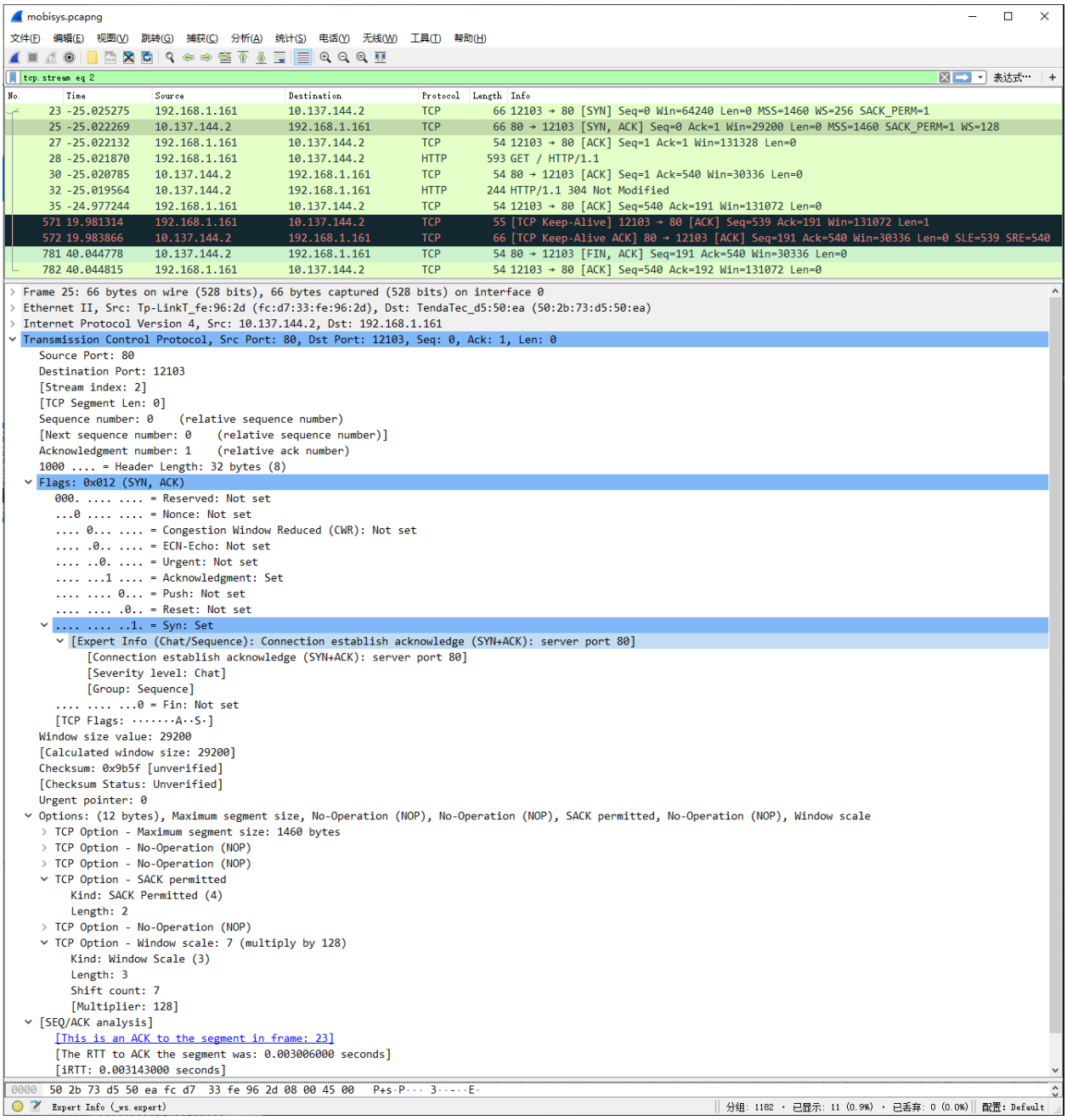


图 3: 第 25 号数据段：建立连接的三次握手——第二次握手

4.3 第 32 号数据段：数据传输——服务器的 HTTP 应答

第 32 号数据段的详细信息参考图 4 。

这个数据段的源端口号为 80，这是一个 HTTP 使用的默认端口。目的端口号为 12103，为客户端指定的端口。序号为 1，确认号是 540，因为第 28 号数据段的序号为 1，长度为 539，服务器收到了 28 号数据段。首部长度为 20 字节，说明这个 TCP 数据段中选项长度为 0。在标志位中，仅有 ACK 位以及 PSH 位为 1，其余标志位为 0，ACK 位为 1 表明确认字段有效，PSH 位为 1 说明要求将缓冲的数据推送到接收应用程序。接收窗口为 237，代表服务器愿意接受的字节数量为 237 字节。校验和为 0xbef3，用于校验。紧急数据指针为 0。

选项部分：

这个数据段不存在选项部分。

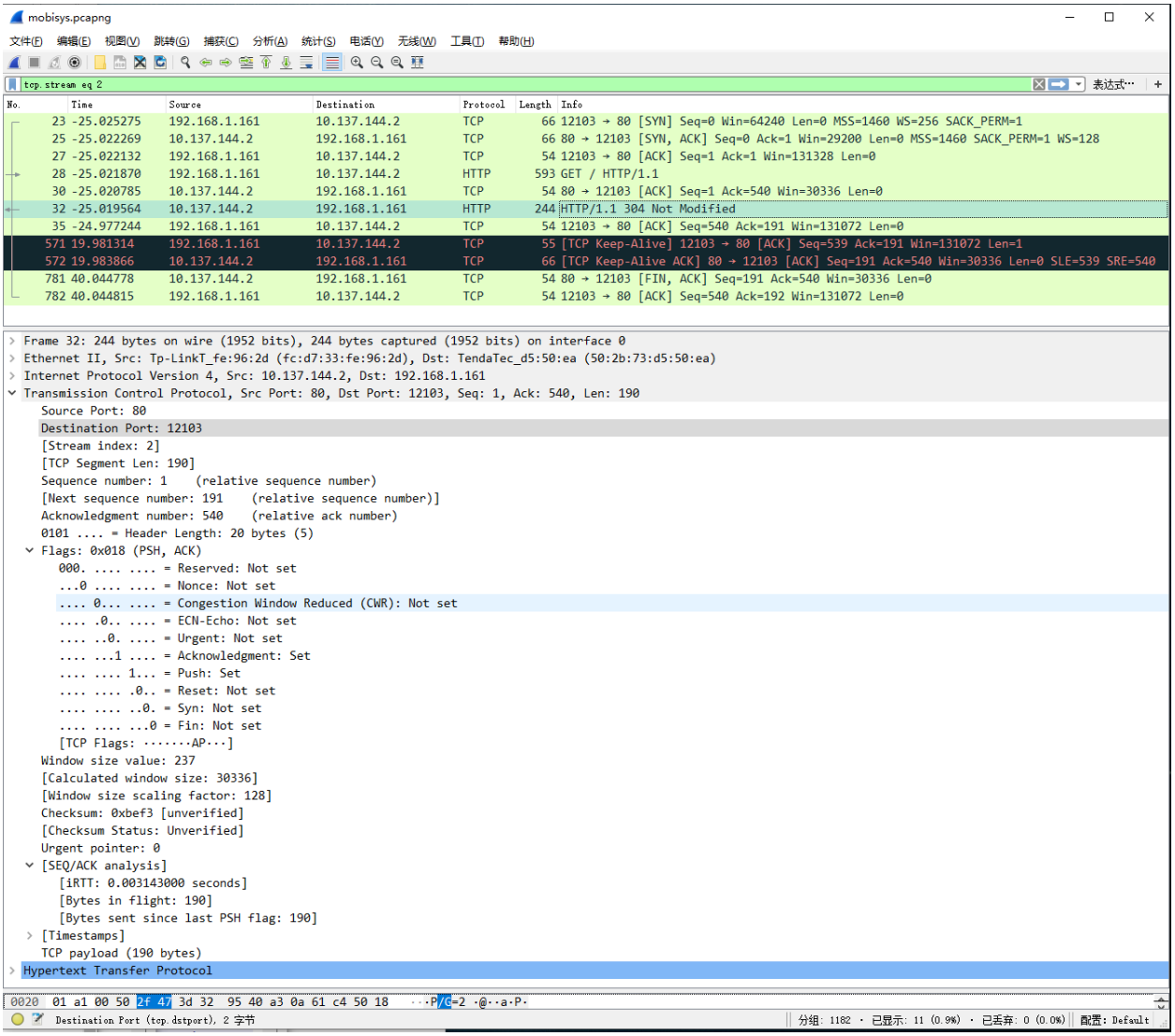


图 4: 第 32 号数据段：数据传输——服务器的 HTTP 应答

4.4 第 571 号数据段：数据传输——TCP Keep-Alive

第 571 号数据段的详细信息参考图 5 。

TCP Keep-Alive 消息用于确认连接是否仍在运行中，也可用于防止连接断裂。[2]

通常，TCP Keep-Alive 在空闲的 TCP 连接上每 45 或 60 秒发送一次。我们可以发现在这个 TCP 流中，571 号数据段与它的上一个数据段（即 35 号数据段）相隔为 45 秒左右。有关 TCP Keep-Alive 的具体详情可参考 Keepalive 。

这个数据段的源端口号为 12013，为客户端指定的端口。目的端口号为 80，这是一个 HTTP 使用的默认端口。序号为 539，确认号是 191，与第 35 号数据段一致。首部长度为 20 字节，说明这个 TCP 数据段中选项长度为 0。在标志位中，仅有 ACK 位为 1，其余标志位为 0，ACK 位为 1 表明确认字段有效。接收窗口为 512，代表服务器愿意接受的字节数量为 512 字节。校验和为 0x4969，用于校验。紧急数据指针为 0。数据大小为 1 个字节。

选项部分：

这个数据段不存在选项部分。

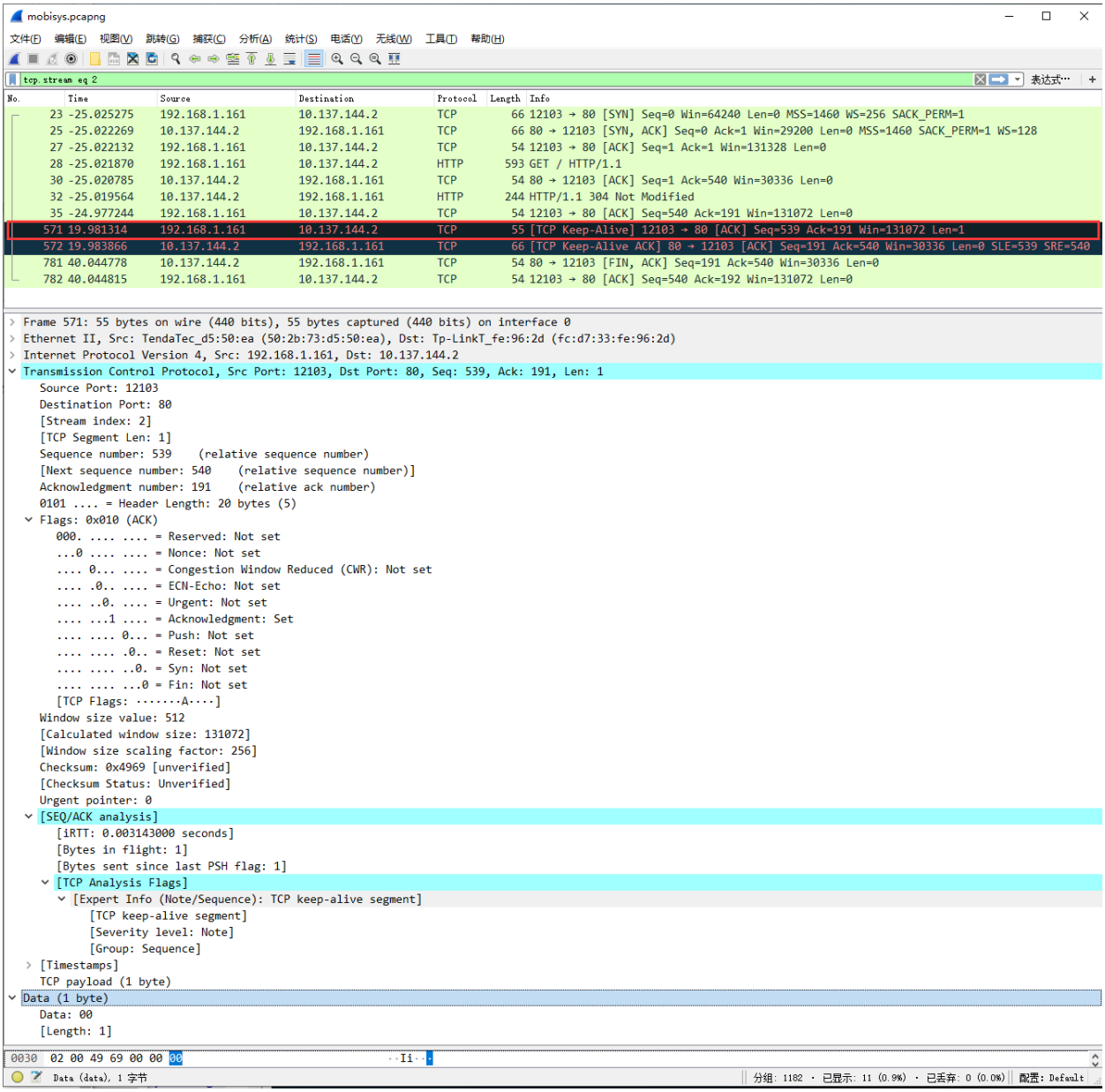


图 5: 第 571 号数据段：数据传输——TCP Keep-Alive

4.5 第 781 号数据段：连接关闭——服务器关闭连接

Linux 默认的 TCP 终止时延为 20s。服务器已经有 20s 没有收到任何请求，所以终止了这个 TCP 连接。

第 781 号数据段的详细信息参考图 6 。

这个数据段的源端口号为 80，这是一个 HTTP 使用的默认端口。目的端口号为 12103，为客户端指定的端口。序号为 191，确认号是 540，因为第 572 号数据段的序号为 191，在这个 TCP 流中，572 号数据段与 781 号数据段都是服务器发出的数据段，且它们之间不存在任何数据段。首部长度为 20 字节，说明这个 TCP 数据段不包含选项。在标志位中，仅有 ACK 位以及 FIN 位为 1，其余标志位为 0，ACK 位为 1 表明确认字段有效，FIN 位为 1 表示这是服务器终止了这个连接。接收窗口为 237，代表服务器愿意接受的字节数量为 237 字节。校验和为 0x4a7b，用于校验。紧急数据指针为 0。

选项部分：

这个数据段不存在选项。

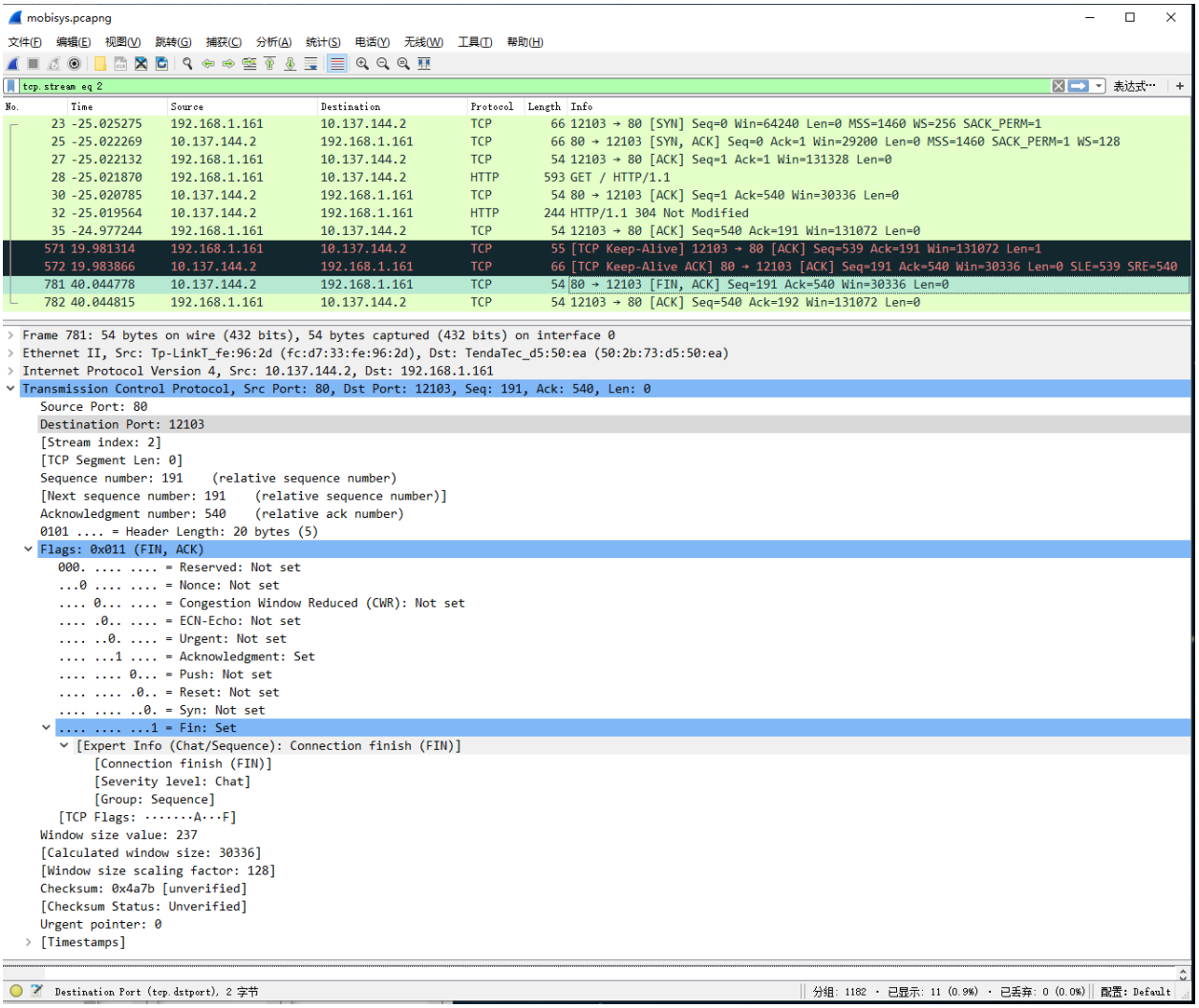


图 6: 第 781 号数据段：连接关闭——服务器关闭连接

References

[1] Keith W. Ross James F. Kurose. *Computer Network - A Top-down Approach*. Pearson, 2018.

[2] Wikipedia. *Keepalive*. URL: https://en.wikipedia.org/wiki/Keepalive. (accessed: 7.12.2012).