天 津 商 业 大 学 学 生 实 验 报 告

开课实验室：现代信息交流中心 403　　开课时间：2014 年 12 月 2 日　　实验报告：2014 年 12 月 2 日

学院名称	信息工程学院	年级、专业、班	软件工程 1201	学号	20125041	姓名	王靖伟	同组姓名	无
课程名称	Computer Networks and Internets	实验项目名称	传输层协议实验			指导教师	尉斌		
实验类型	验证 ■	综合 □	设计 □	创新 □				成绩	
教师评语	教师签名： 年　月　日								

实验报告内容一般包括以下几个内容：1、目的要求　2、仪器用具及材料（仪器名称及主要规格、用具名称）　3、实验内容及原理（简单但要抓住要点，写出依据原理）　4、操作方法与实验步骤　5、数据图表格（照片）　6、实验过程原始记录　7 数据处理及结果（按实验要求处理数据、结论）　8、作业题　9、讨论（对实验中存在的问题、进一步的想法等进行讨论）

实验报告内容：

1）实验目的：①熟悉传输层协议的内容和功能；

②掌握传输层协议的使用方法

2）实验要求：①掌握 TCP 建立和重传的功能和使用方法

②掌握与该协议相关的工具操作方法

3）实验设备：协议服务器（利用已有的网络服务器）；协议客户端软件

4）实验过程：

① TCP 的功能

② 客户端软件的配置与操作

③ 回答课后问题（需详细阐述，注明题号和问题）

5）实验心得

注 1.每个实验项目一份实验报告。2.实验报告第一页学生必须使用规定的实验报告纸书写，附页用实验报告附页纸或 A4 纸书写，字迹工整，曲线要画在坐标纸上，线路图要整齐、清楚（不得徒手画）。3. 实验教师必须对每份实验报告进行批改，用红笔指出实验报告中的错、漏之处，并给出评语、成绩，签全名、注明日期。 4.待实验课程结束以后，要求学生把实验报告整理好，交给实验指导教师，加上实验课学生考勤及成绩登记表（见附件 2）、目录和学院统一的封面（见附件 3）后，统一装订成册存档。

天津商业大学学生实验报告附页

开课实验室：现代信息交流中心 403　　开课时间：2014 年 12 月 2 日　　实验报告：2014 年 12 月 2 日

根据 tcp_pcattcp_n1.cap 和 tcp_ssh.cap 回答下列问题。

（1）在 tcp_ssh.cap 中，哪些分组包含三次握手？

在 tcp_ssh.cap 中，分组 1 到 3 显示的就是三次握手，如图 1 所示。

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.101	128.153.4.131	TCP	62	stgxfws > ssh [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2	0.049886	128.153.4.131	192.168.0.101	TCP	60	ssh > stgxfws [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460
3	0.049935	192.168.0.101	128.153.4.131	TCP	54	stgxfws > ssh [ACK] Seq=1 Ack=1 Win=64240 Len=0

图 1. 三次握手分组

（2）在 tcp_ssh.cap 中，每个方向中真实的初始序号是多少？你如何得知？

观察分组 1，Wireshark 显示的序号是 0。然而，如果选择分组首部的序号字段，原始框中“c6 a1 4e 5a”被突出显示。Wireshark 所显示的是逻辑序号，而真正的初始序号不是 0，如图 2 所示。

```
⊞ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: Intel_53:87:d9 (00:07:e9:53:87:d9), Dst: LinksysG_8d:be:1d (00:06:25:8d:be:1d)
⊞ Internet Protocol Version 4, Src: 192.168.0.101 (192.168.0.101), Dst: 128.153.4.131 (128.153.4.131)
⊟ Transmission Control Protocol, Src Port: stgxfws (1226), Dst Port: ssh (22), Seq: 0, Len: 0
    Source port: stgxfws (1226)
    Destination port: ssh (22)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
  ⊞ Checksum: 0x2829 [validation disabled]
```

```
0000  00 06 25 8d be 1d 00 07  e9 53 87 d9 08 00 45 00   ..%..... .S....E.
0010  00 30 12 7b 40 00 80 06  a2 23 c0 a8 00 65 80 99   .0.{@... .#...e..
0020  04 83 04 ca 00 16 c6 a1  4e 5a 00 00 00 00 70 02   ........ NZ....p.
0030  fa f0 28 29 00 00 02 04  05 b4 01 01 04 02         ..()....  ......
```

图 2. 分组 1 的报文中显示的初始序号

观察分组 2，Wireshark 显示的序号是 0。然而，如果选择分组首部的序号字段，原始框中“50 dc 25 5f”被突出显示。Wireshark 所显示的是逻辑序号，而真正的初始序号不是 0，如图 3 所示。

```
⊞ Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: Intel_53:87:d9 (00:07:e9:53:87:d9)
⊞ Internet Protocol Version 4, Src: 128.153.4.131 (128.153.4.131), Dst: 192.168.0.101 (192.168.0.101)
⊟ Transmission Control Protocol, Src Port: ssh (22), Dst Port: stgxfws (1226), Seq: 0, Ack: 1, Len: 0
    Source port: ssh (22)
    Destination port: stgxfws (1226)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 24 bytes
  ⊞ Flags: 0x012 (SYN, ACK)
    Window size value: 1460
    [Calculated window size: 1460]
```

```
0000  00 07 e9 53 87 d9 00 06  25 8d be 1d 08 00 45 00   ...S.... %.....E.
0010  00 2c 58 d9 40 00 ea 06  f1 c8 80 99 04 83 c0 a8   .,X.@... ........
0020  00 65 00 16 04 ca 50 dc  25 5f c6 a1 4e 5b 60 12   .e....P. %_..N[`.
0030  05 b4 bc 20 00 00 02 04  05 b4 67 4b               ... .... ..gK
```

图 3. 分组 2 的报文中显示的初始序号

（3）在这两个跟踪文件中，使用 SYN 分组和 SYNACK 分组来计算每个连接的往返时间。往返于本地网的机器的时间和到远程服务器的往返时间相比怎样？

往返于本地网的机器的时间的 SYN 分组和 SYNACK 分组如图 4 所示。往返于远程服务器的时间的 SYN 分组和 SYNACK 分组如图 5 所示。

3 0.002936	192.168.0.100	192.168.0.102	TCP	62
4 0.005476	192.168.0.102	192.168.0.100	TCP	62

图 4. 往返于本地网的机器的时间的 SYN 分组和 SYNACK 分组

1 0.000000	192.168.0.101	128.153.4.131	TCP	62
2 0.049886	128.153.4.131	192.168.0.101	TCP	60

图 5. 往返于远程服务器的时间的 SYN 分组和 SYNACK 分组

从上面两个图中可以看到，往返于本地网的机器的时间为 0.005476-0.002936=0.002540s，而往返于远程服务器的时间为 0.049886s，可以看出，往返于远程服务器的时间明显大于往返于本地网的机器的时间，大约是其 19.6 倍。

（4）在 tcp_pcattcp_n1.cap 中，哪些分组只包含首部而没有数据？写一个显示过滤器筛选出这些分组。在 tcp_ssh.cap 中有多少分组能与这个过滤器匹配呢？

如图 6 所示 tcp_pcattcp_n1.cap 的分组只包含首部而没有数据。

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Intel_53:87:d9	Broadcast	ARP	42	who has 192.168.0.102? Tell 192.168.0.100
2	0.002925	Agere_44:e5:07	Intel_53:87:d9	ARP	60	192.168.0.102 is at 00:02:2d:44:e5:07
3	0.002936	192.168.0.100	192.168.0.102	TCP	62	spearway > 5001 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
4	0.005476	192.168.0.102	192.168.0.100	TCP	62	5001 > spearway [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460 SACK_PERM=1
5	0.005500	192.168.0.100	192.168.0.102	TCP	54	spearway > 5001 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.014136	192.168.0.102	192.168.0.100	TCP	60	5001 > spearway [ACK] Seq=1 Ack=2921 Win=17520 Len=0
12	0.023425	192.168.0.102	192.168.0.100	TCP	60	5001 > spearway [ACK] Seq=1 Ack=5841 Win=17520 Len=0
14	0.028116	192.168.0.102	192.168.0.100	TCP	60	5001 > spearway [ACK] Seq=1 Ack=8194 Win=17520 Len=0
15	0.029205	192.168.0.102	192.168.0.100	TCP	60	5001 > spearway [FIN, ACK] Seq=1 Ack=8194 Win=17520 Len=0
16	0.029216	192.168.0.100	192.168.0.102	TCP	54	spearway > 5001 [ACK] Seq=8194 Ack=2 Win=64240 Len=0

图 6. tcp_pcattcp_n1.cap 中只包含首部而没有数据的分组

我所写的过滤器如图 7 所示。

图 7. 过滤器筛选出只包含首部而没有数据的分组

在 tcp_ssh.cap 中有 59 个分组能与这个过滤器匹配，如图 8 所示过滤结果。

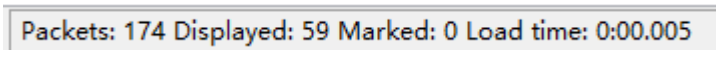

图 8. tcp_ssh.cap 过滤结果

（5）在 TCP 流中仅有的没有设置 ACK 标志的报文段是什么？为什么？

在 TCP 流中仅有的没有设置 ACK 标志的报文段是[SYN]报文，这条报文是没有数据的 TCP 报文段，并且将首部的 SYN 位设置成 1。因此，第一条报文常常被称为 SYN 分组。这个报文段里的序号可以被设置成任何值。不管设置成什么值，它都表示客户端为后续报文设定的起始编号。SYN 分组通常是从客户端发送到服务器端。这个报文段请求建立连接。因为一旦成功建立了连接，服务器进程必须已经在监听 SYN 分组所指示的 IP 地址和端口号。如果没有建立连接，SYN 分组将不会应答。如果第一个分组丢失了，客户端通常会发送若干个 SYN 分组，要不然客户端将会停止并报告一个错误给应用程序。

（6）在 tcp_pcattcp_n1.cap 中，所有的从服务器到客户端的分组的确认号都是相同的吗？为什么？所有的从客户端到服务器的分组的确认号也都是相同的吗？（注意：在这种情况下，服务器是接收端，因为它监听接入的连接）

比如说分组 4，从服务器到客户端的分组的确认号是 94 f2 2e bf,如图 9 所示。分组 8，从服务器到客户端的分组的确认号是 94 f2 3a 27,如图 10 所示。如此可见是不同的。

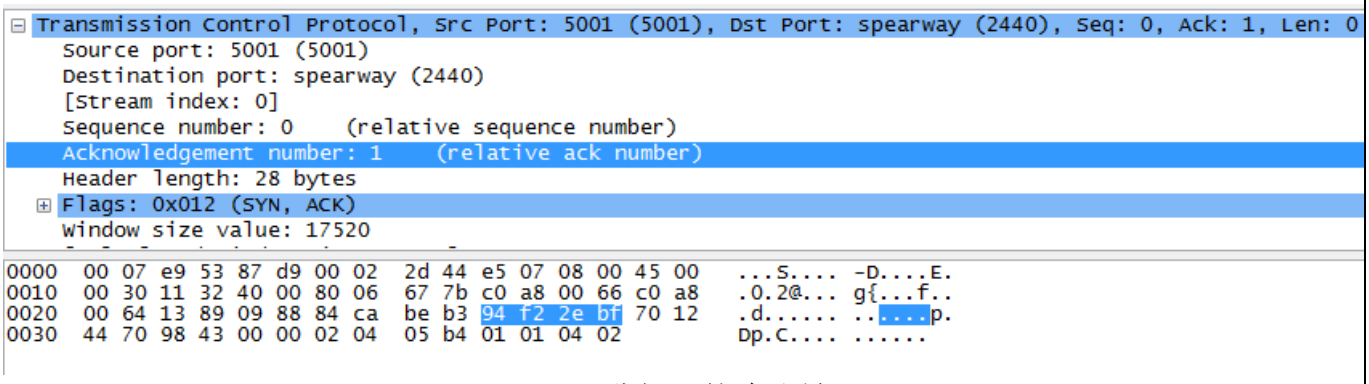

图 9. 分组 4 的确认号

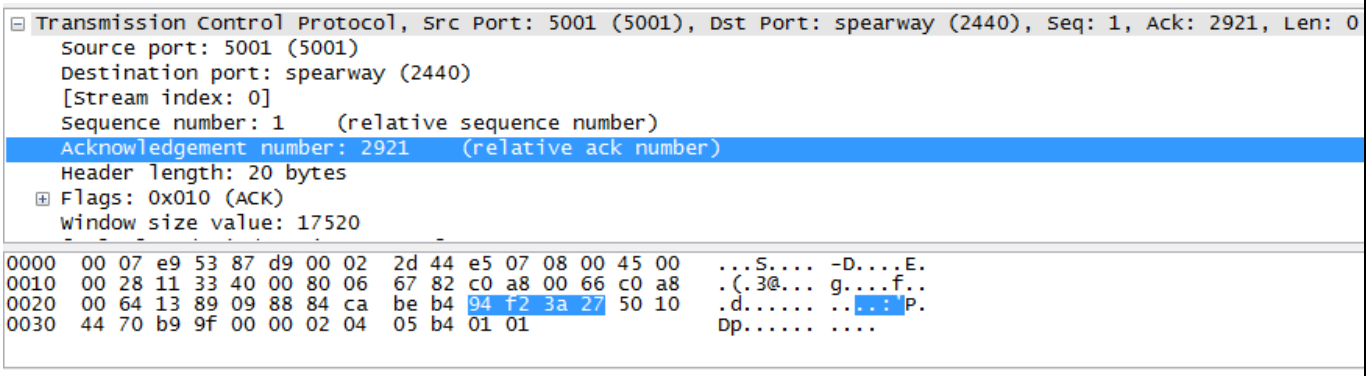

图 10. 分组 8 的确认号

所有的从客户端到服务器的分组的确认号也都是不相同的。

（7）在 tcp_pcattcp_n1.cap 中，有多少分组是分组 12 确认的呢？分组 14 确认的分组又有多少呢？

分组 12 确认的是分组 13，分组 14 的确认是分组 16。

实验心得

通过这次实验使我学会了很多实战上的知识。在课堂上学习理论，在实验课上实践，使我对于传输层协议的基础知识更加了解。通过实验，使我更加能熟练使用 wireshark 工具抓取网络报文、查看已有报文信息并能对报文加以分析。使我掌握了基本的 wireshark 过滤器的写法和用法。

通过三次握手实验，使我更加理解三次握手的过程。通过查看报文，也对 TCP 报文段的首部格式有了一个更深的理解和记忆。通过这次实验，也使我更加熟练使用命令行命令来为 ttcp 流运行接收端，然后监听 TCP 的某端口。在实验过程中，我详细阅读每行报文，掌握了 TCP 数据报的格式以及内容，在以后的学习中还应该加强对于报文的阅读理解的能力。