计算机网络实验八

传输控制协议 (TCP)

信息学部 朱婉婷

主要内容

- 一、实验原理
- 二、实验步骤
- 三、实验结果及分析

TCP简介

◆传输控制协议 TCP

(Transmission Control Protocol)

DNS

HTTP

SSL

• TCP 是面向连接的传输层协议。

UDP

TCP

• 每一条 TCP 连接只能是点对 点的(一对一)。

IPv4

DHCP

ARP

ICMP

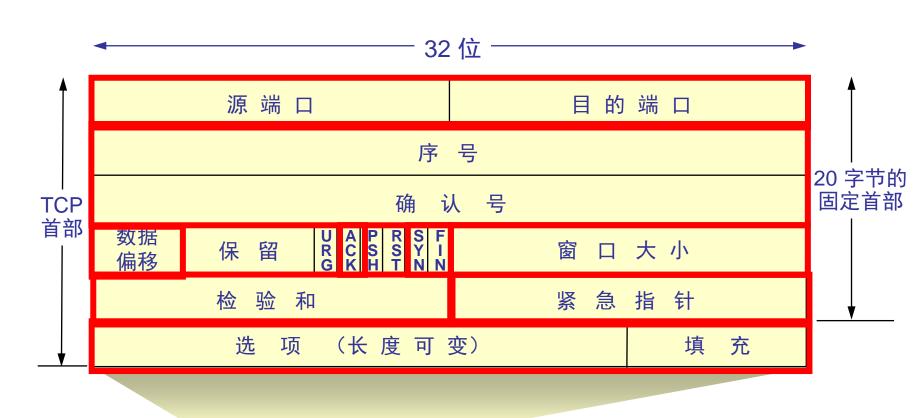
• TCP 提供全双工通信。

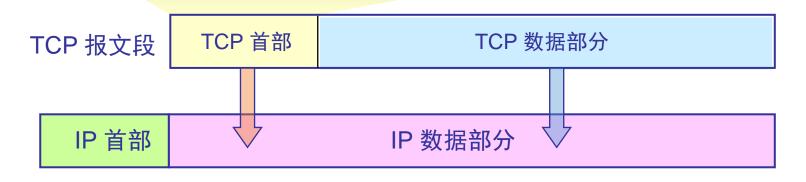
• TCP 是面向字节流。

• TCP 提供可靠交付的服务。

• 滑动窗口和超时重传。

以太网协议 (Ethernet)



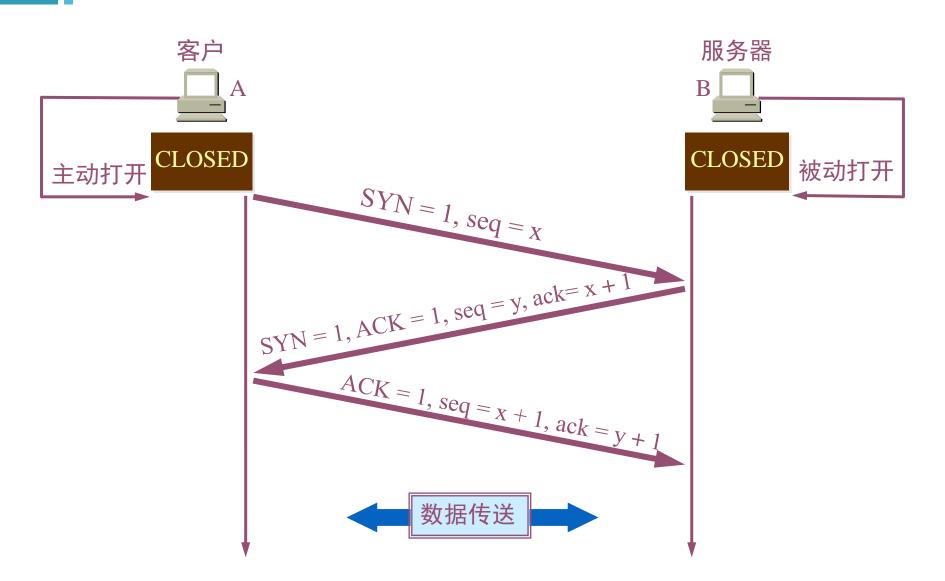


- 源端口和目的端口字段:各占2字节,分别是数据发起者和接收者的端口号。
- 序号字段: 占 4 字节。TCP 连接中传送的数据流中的每一个字节都编上一个序号。序号字段的值则指的是本报文段所发送的数据的第一个字节的序号。
- 确认号字段:占4字节,是期望收到对方的下一个报文段的数据的第一个字节的序号。
- 数据偏移(即首部长度): 占 4 位,它指出 TCP 报文段的数据起始处距离 TCP 报文段的起始处有多远。以 4 字节为计算单位。
- 保留字段:占6位,保留为今后使用,但目前应置为0。

- 紧急 URG: 当 URG = 1 时,表明紧急指针字段有效。它告诉系 统此报文段中有紧急数据,应尽快传送(相当于高优先级的数据)。
- 确认 ACK: 只有当 ACK = 1 时确认号字段才有效,代表这是一个确认(ACK)包。当 ACK=0 时,确认号无效。
- 推送 PSH:接收到 PSH = 1 的报文段,接收方应尽快地将报文交付给应用层,不做队列处理。
- 复位 RST: 当 RST = 1 时,表明 TCP 连接中出现严重差错(如由于主机崩溃或其他原因),必须释放连接,然后再重新建立运输连接。
- 同步 SYN: 同步 SYN = 1 表示这是一个连接请求或连接响应报文。
- 终止 FIN: 用来释放一个连接。FIN = 1 表明此报文段的发送端的数据已发送完毕,并要求释放运输连接。

- 窗口字段: 占 2 字节,用来让对方设置发送窗口的依据,表示准备收到的每个TCP数据的大小,单位为字节。
- 检验和:占2字节。检验和字段检验的范围包括首部和数据 这两部分。在计算检验和时,要在TCP报文段的前面加上 12字节的伪首部。
- 紧急指针字段:占16位,指出在本报文段中紧急数据共有多少个字节(紧急数据放在本报文段数据的最前面),当URG位取值为1时有效。
- 选项字段: 长度可变。
- 填充字段: 这是为了使整个首部长度是 4 字节的整数倍。

TCP 的连接建立(三次握手)

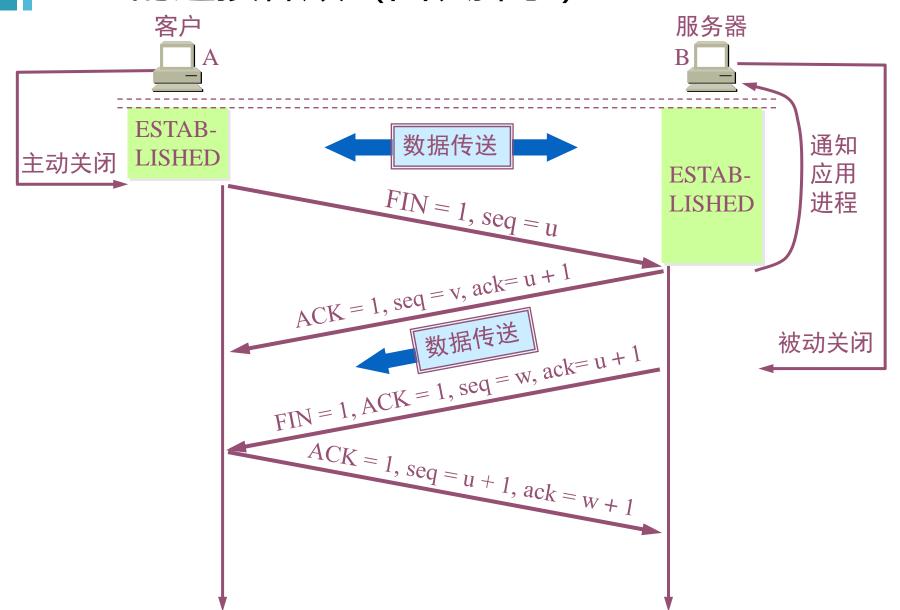


TCP 的连接建立

◆三次握手过程

- (1) A 的 TCP 向 B 发出连接请求报文段,其首部中的同步 位 SYN = 1,并选择序号 seq = x,表明传送数据时的第一个 数据字节的序号是 x。
- (2) B 的 TCP 收到连接请求报文段后,如同意,则发回确认。B 在确认报文段中应使 SYN = 1,使 ACK = 1,其确认号ack = x +1,自己选择的序号 seq = y。
- (3) A 收到此报文段后向 B 给出确认,其 ACK = 1,确认号 ack = y +1。A 的 TCP 通知上层应用进程,连接已经建立。B 的 TCP 收到主机 A 的确认后,也通知其上层应用进程,TCP 连接已经建立。

TCP 的连接释放(四次挥手)



TCP 的连接释放

◆四次挥手过程

- (1)数据传输结束后,通信的双方都可释放连接。现在 A 的应用 进程先向其 TCP 发出连接释放报文段,并停止再发送数据,主动 关闭 TCP连接。 A 把连接释放报文段首部的 FIN = 1,其序号seq = u,等待 B 的确认。
- (2) B 发出确认,确认号 ack = u + 1,而这个报文段自己的序号 seq = v。TCP 服务器进程通知高层应用进程。从A到B这个方向 的连接就释放了,TCP 连接处于半关闭状态。B 若发送数据,A 仍 要接收。
- (3) 若 B 已经没有要向 A 发送的数据, 其应用进程就通知 TCP 释放连接, 发送一个FIN报文(序号seq = w, 确认号 ack = u + 1) 给客户端。
- (4) A 收到连接释放报文段后,必须发出确认。在确认报文段中 ACK = 1,确认号 ack = w + 1,序号 seq = u + 1。A 在发送完最后 一个 ACK 报文段后,再经过时间 2MSL,才真正释放掉TCP连接。

主要内容

- 一、实验原理
- 二、实验步骤
- 三、实验结果及分析

实验环境搭建

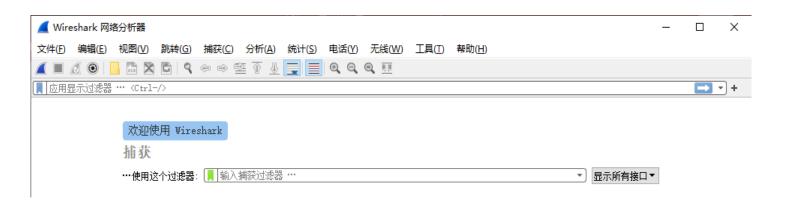
列出本次实验所使用的平台和相关软件,以下为例:

(打开cmd指令窗口,输入指令 "ipconfig /all"查看)

- 1、主机: 联想笔记本 (Win10系统); 主机IP地址:
- 192.168.1.106; 子网掩码: 255.255.255.0; 主机网卡
- MAC地址: 34-F6-4B-C0-90-40。
- 2、网络连接方式:无线连接;默认网关地址:
- 192.168.1.1.
- 3、抓包工具: Wireshark (v3.6.2)。

当主机打开某个网页时,主机和Web服务器之间建立TCP连接,随后进行HTTP服务,并在最后释放TCP连接。

1、打开Wireshark软件,双击本次实验正在使用的网络接口, 开始进行抓包。



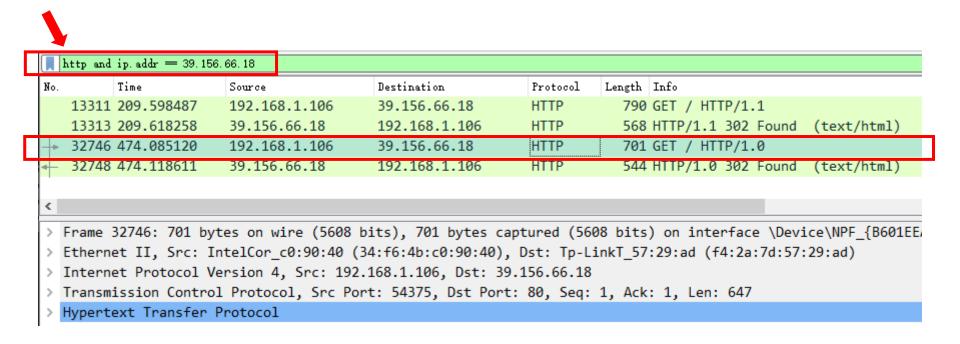
- 2、然后打开浏览器,在网页地址栏中输入网址,例如访问北工大官网并浏览新闻,或者访问百度官网等。
- 3、关闭浏览器,结束访问。
- 4、通过ping命令获取网站的IP地址39.156.66.14。



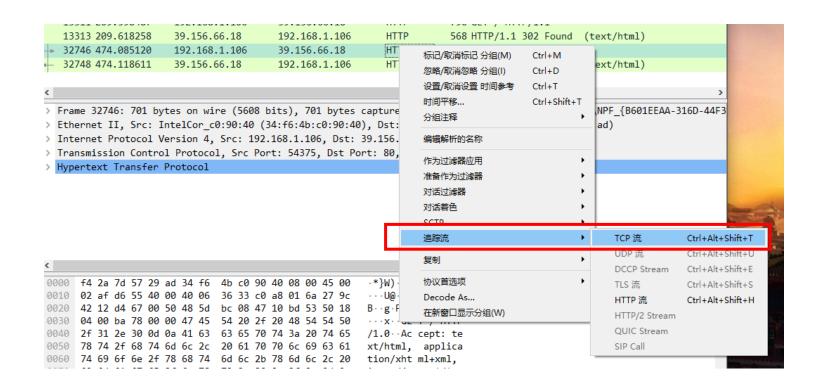
```
C:\Users\zwt717>ping www.baidu.com

正在 Ping www.a.shifen.com. [39.156.66.18] 具有 32 字节的数据:
来自 39.156.66.18 的回复: 字节=32 时间=119ms TTL=53
来自 39.156.66.18 的回复: 字节=32 时间=6ms TTL=52
来自 39.156.66.18 的回复: 字节=32 时间=7ms TTL=53
来自 39.156.66.18 的回复: 字节=32 时间=6ms TTL=53
39.156.66.18 的回复: 字节=32 时间=6ms TTL=53
在 39.156.66.18 的 Ping 统计信息:
数据包: 已发送 = 4,已接收 = 4,丢失 = 0(0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 6ms,最长 = 119ms,平均 = 34ms
```

5、在过滤栏内输入 "http and ip.addr == **39.156.66.18**" 进行包过滤。并找到HTTP协议包info项为 "GET" 形式的数据包。



6、点击选中HTTP协议包Info项为"GET"形式的数据包,随后点击Wireshark导航栏内的"分析->追踪流->TCP流",或右键选择"追踪流->TCP流"进行选中数据包的TCP流追踪。



6、点击选中HTTP协议包Info项为"GET"形式的数据包,随后点击Wireshark导航栏内的"分析->追踪流->TCP流",或右键选择"追踪流->TCP流"进行选中数据包的TCP流追踪。

TCP连接建立:三次握手

	A	top.stream eq 323				₩ 📑 🔻 +
·	J.	Time	Source	Destination	Protocol	Length Info
	г	32740 474.077663	192.168.1.106	39.156.66.18	TCP	66 54375 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=140
		32742 474.084398	39.156.66.18	192.168.1.106	TCP	66 80 → 54375 [SYN, ACK] Seq=0 Ack=1 Win=8192 Ler
		32743 474.084508	192.168.1.106	39.156.66.18	TCP	54 54375 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
-	Þ	32746 474.085120	192.168.1.106	39.156.66.18	HTTP	701 GET / HTTP/1.0
		32747 474.098031	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [ACK] Seq=1 Ack=648 Win=79744 Len=
4	+	32748 474.118611	39.156.66.18	192.168.1.106	HTTP	544 HTTP/1.0 302 Found (text/html)
П		32749 474.118611	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [FIN, ACK] Seq=491 Ack=648 Win=7974
		32750 474.118723	192.168.1.106	39.156.66.18	TCP	54 54375 → 80 [ACK] Seq=648 Ack=492 Win=261632 L€
		32753 474.122485	192.168.1.106	39.156.66.18	TCP	54 54375 → 80 [FIN, ACK] Seq=648 Ack=492 Win=2610
		32755 474.129639	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [ACK] Seq=492 Ack=649 Win=79744 Ler
٦	L	36404 477.128783	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [RST] Seq=492 Win=0 Len=0
	<					>

TCP连接释放: 四次挥手

7、从中选取任一TCP报文段进行TCP首部格式分析。

```
Time
                     Source
                                        Destination
                                                            Protocol
                                                                     Length Info
                                        39.156.66.18
                                                            TCP
 32740 474.077663
                     192.168.1.106
                                                                        66 54375 → 80 [SYN] Seg=0 Win=65535 Len=0
Transmission Control Protocol, Src Port: 54375, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 54375
  Destination Port: 80
  [Stream index: 323]
  [Conversation completeness: Complete, WITH DATA (63)]
  [TCP Segment Len: 0]
  Sequence Number: 0
                         (relative sequence number)
  Sequence Number (raw): 1214102535
  [Next Sequence Number: 1
                               (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
> Flags: 0x002 (SYN)
  Window: 65535
  [Calculated window size: 65535]
  Checksum: 0x6a33 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
> Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (N
> [Timestamps]
```

> [Timestamps]

8、从中选取一组TCP连接建立的三次握手进行观察、分析。

```
Destination
                                                         Protocol Length Info
     Time
32740 474.077663
                                                                      66 54375 → 80 [SYN] Seq=0 Win=65535 Len=0 N
                   192.168.1.106
                                      39.156.66.18
                                                         TCP
                                      192.168.1.106
                                                                      66 80 → 54375 [SYN, ACK] Seg=0 Ack=1 Win=81
32742 474.084398
                   39.156.66.18
                                                         TCP
32743 474.084508
                                                                      54 54375 → 80 [ACK] Seg=1 Ack=1 Win=262144
                   192.168.1.106
                                      39.156.66.18
                                                         TCP
```

```
Transmission Control Protocol, Src Port: 54375, Dst Port: 80, Seq: 0, Len: 0
   Source Port: 54375
   Destination Port: 80
   [Stream index: 323]
   [Conversation completeness: Complete, WITH DATA (63)]
  [TCP Segment Len: 0]
  Sequence Number: 0
                        (relative sequence number)
  Sequence Number (raw): 1214102535
   [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
   Window: 65535
   [Calculated window size: 65535]
  Checksum: 0x6a33 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
> Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (N
```

9、从中选取一组TCP连接释放的四次挥手进行观察、分析。

No.	Time	Source	Destination	Protocol	Length Info
	32749 474.118611	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [FIN, ACK] Seq=491 Ack=648 W:
	32750 474.118723	192.168.1.106	39.156.66.18	TCP	54 54375 → 80 [ACK] Seq=648 Ack=492 Win=26:
	32753 474.122485	192.168.1.106	39.156.66.18	TCP	54 54375 → 80 [FIN, ACK] Seq=648 Ack=492 W:
	32755 474.129639	39.156.66.18	192.168.1.106	TCP	54 80 → 54375 [ACK] Seq=492 Ack=649 Win=79
					>

```
Transmission Control Protocol, Src Port: 80, Dst Port: 54375, Seq: 491, Ack: 648, Len: 0
```

Source Port: 80

Destination Port: 54375

[Stream index: 323]

[Conversation completeness: Complete, WITH_DATA (63)]

[TCP Segment Len: 0]

Sequence Number: 491 (relative sequence number)

Sequence Number (raw): 1192279869

[Next Sequence Number: 492 (relative sequence number)]

Acknowledgment Number: 648 (relative ack number)

Acknowledgment number (raw): 1214103183 0101 = Header Length: 20 bytes (5)

> Flags: 0x011 (FIN, ACK)

Window: 2492

[Calculated window size: 79744] [Window size scaling factor: 32] Checksum: 0x9864 [unverified] [Checksum Status: Unverified]

Urgent Pointer: 0

三次挥手也可以

- ◆四次挥手变三次挥手
- 第二次和第三次被合并了。
- TCP延迟确认机制。

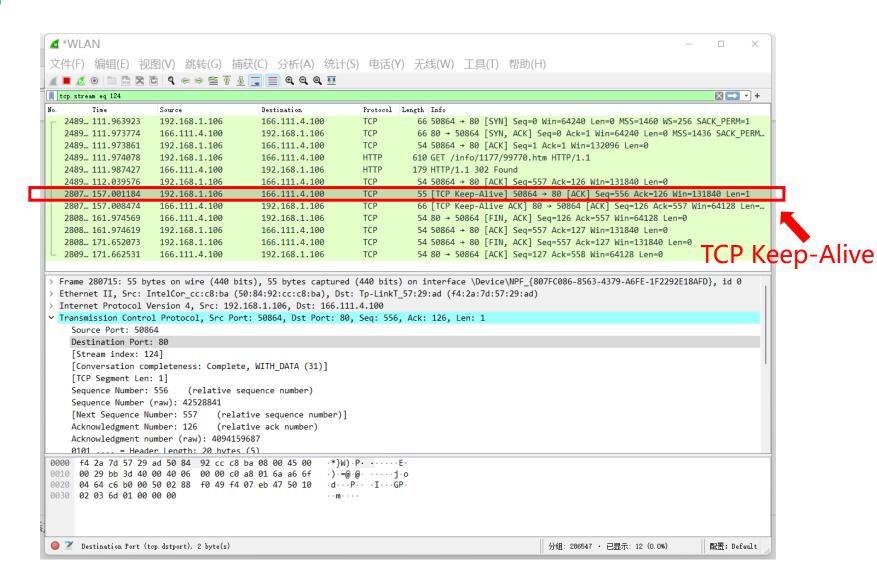
- ◆短连接和长连接(持续连接)
- 在http1.0中默认是短连接:每次与服务器交互,都需要新开一个连接。耗费资源,已经很少使用了。
- 在http1.1中默认使用<mark>持续连接(keep-alive)来解决:建立一次连接,多次请求均由这个连接完成。现在大部分用的是http1.1,并向http2.0过渡。</mark>

三次挥手也可以

◆示例:三次挥手时,可能出现的抓包结果如下图所示。



TCP Keep-Alive



实验要求

- ◆ 本次实验以下两种情况都可以:
 - □ 分析任意TCP首部格式+三次握手+四次挥手;
 - □ 分析任意TCP首部格式+三次握手+三次挥手。

主要内容

- 一、实验原理
- 二、实验步骤
- 三、实验结果及分析

TCP首部格式分析

实验结果:

```
Length Info
No.
         Time
                        Source
                                           Destination
                                                               Protocol
   32740 474.077663
                                           39.156.66.18
                                                                           66 54375 → 80 [SYN] Seq=0 Win=65535 Len=0 N
                        192.168.1.106
                                                               TCP
  Transmission Control Protocol, Src Port: 54375, Dst Port: 80, Seq: 0, Len: 0
     Source Port: 54375
     Destination Port: 80
     [Stream index: 323]
     [Conversation completeness: Complete, WITH DATA (63)]
     [TCP Segment Len: 0]
                           (relative sequence number)
     Sequence Number: 0
     Sequence Number (raw): 1214102535
     [Next Sequence Number: 1
                                  (relative sequence number)]
     Acknowledgment Number: 0
     Acknowledgment number (raw): 0
     1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
     Window: 65535
     [Calculated window size: 65535]
     Checksum: 0x6a33 [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP)
  > [Timestamps]
```

实验分析:

- 源端口号 Source Port: 54375, 定义了发送这个报文段的主机中的应用程序的端口号。
- 目的端口号Destination Port: 80, 定义了接收这个报文段的主机中的应用程序的端口号, 因为主机要向服务器请求HTTP服务, 所以此处为HTTP协议的端口80。
- 序号Sequence number: 0, 定义了指派给本报文段第一个数据字节的编号。为了保证连接性, 要发送的每一个字节都要编上号。序号可以告诉终点, 报文段中的第一个字节是这个序列中的哪一个字节。
- 确认号 Acknowledgment number: 0, 定义了报文段的接收方期望从对方接收的字节编号。如果报文段的接收方成功地接收了对方发来的编号为X的字节,那么它就返回所期望接收的字节序列作为确认号。
- 首部长度 1000 = Header Length: 32 bytes (8)

- 标志位Flags: 0x002 (SYN)
- 000. = Reserved: Not set #保留位
- ...0 = Nonce: Not set #Noce位
- 0... = Congestion Window Reduced (CWR): Not set #CER位
-0.. = ECN-Echo: Not set #ECN位
-0. = Urgent: Not set #紧急指针有效位
-0 = Acknowledgment: Not set #确认有效位
- 0... = Push: Not set #请求推送位
-0.. = Reset: Not set #连接复位位
-1. = Syn: Set #同步序号位
-0 = Fin: Not set #终止连接位

- 窗口大小Window size value: 65535,并由接受方来决定。
- 检验和 Checksum: 0x6a33 [unverified]
- 紧急指针Urgent pointer: 0,只有当紧急标志置位时,这个16位的字段 才有效。
- 选项Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

绘制TCP首部字段格式 (SYN):

	So	our	се	Por	t: 5	43	75				Destination Port: 80
							S	equ	number: 0		
						A	ckn	ow	ent number: 0		
Header Length: 32 bytes	0 0 0	0	0	0	0	0	0	0	1	Window size value: 65535	
	С	he	cks	um	:0x	6a3	33				Urgent pointer: 0
I -	-		-	-					_		nt size, No-Operation (NOP), Window Operation (NOP), SACK permitted

实验结果:

```
No.
         Time
                        Source
                                            Destination
                                                                Protocol
                                                                          Length Info
                        192.168.1.106
                                            39.156.66.18
   32740 474.077663
                                                                TCP
                                                                              66 54375 → 80 [SYN] Seg=0 Win=65535 Len=0 M
   32742 474.084398
                        39.156.66.18
                                            192,168,1,106
                                                                              66 80 → 54375 [SYN, ACK] Seq=0 Ack=1 Win=81
                                                                TCP
                                                                              54 54375 → 80 [ACK] Seg=1 Ack=1 Win=262144
                        192.168.1.106
                                            39.156.66.18
   32743 474.084508
                                                                TCP
```

```
Sequence Number: 0 (relation equence number)
 Sequence Number (raw): 1214102535
  [Next Sequence Number: 1
                       (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
    000. .... = Reserved: Not set
                                                         第一次握手:客户端向服务器
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
                                                         发送连接请求报文段,标志位
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
                                                         SYN置为1,序列号为0。
    .... ...0 .... = Acknowledgment: Not set
    .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
    Window: 65535
  [Calculated window size: 65535]
```

绘制第一次握手TCP报文首部:

	So	our	се	Por	t: 5	43	75				Destination Port: 80
							S	equ	number: 0		
						Ad	kn	ow	led	ent number: 0	
Header Length: 32 bytes	0 0 0	0	0	0	0	0	0	0	1	0	Window size value: 65535
	С	he	cks	um	:0x	6a3	33				Urgent pointer: 0
I -	-		-	-					_		nt size, No-Operation (NOP), Window Operation (NOP), SACK permitted

实验结果:

```
Length Info
                                        Destination
                                                             Protocol
      Time
                     Source
32740 474.077663
                    192.168.1.106
                                        39, 156, 66, 18
                                                                          66 54375 → 80 [SYN] Seq=0 Win=65535 Len=0 M
                                                             TCP
32742 474.084398
                     39.156.66.18
                                        192.168.1.106
                                                                          66 80 → 54375 [SYN, ACK] Seq=0 Ack=1 Win=81
                                                             TCP
                    192.168.1.106
                                        39.156.66.18
                                                                          54 54375 → 80 [ACK] Seg=1 Ack=1 Win=262144
32743 474.084508
                                                             TCP
```

```
Sequence Number: 0 (relate sequence number)
 Sequence Number (raw): 1192279378
  [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment number (raw): 1214102536
  1000 .... = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
    .... Not set
    [TCP Flags: ······A··S·]
 Window: 8192
  [Calculated window size: 8192]
```

第二次握手:服务器端收到客户端发过来的请求连接报文段,由SYN=1知道客户端要求建立连接。随后服务器端向客户端发送一个SYN和ACK都置为1的TCP报文段,并且在这个报文段当中,设置初始序列号0,将确认号设置为客户端的序列号加1。

绘制第二次握手TCP报文首部:

		So	urc	e P	ort	: 80)				Destination Port: 54375
							S	equ	uen	number: 0	
						A	ckn	ow	led	ent number: 1	
Header Length: 32 bytes	0 0 0	0	0	0	0	1	0	0	1	0	Window size value: 8192
	C	he	cksı	um	:0x	45c	da				Urgent pointer: 0
	•								•	nt size, No-Operation (NOP), Window Operation (NOP), SACK permitted	

实验结果:

```
Length Info
                     Source
                                         Destination
      Time
                                                              Protocol
32740 474.077663
                     192,168,1,106
                                         39, 156, 66, 18
                                                              TCP
                                                                           66 54375 → 80 [SYN] Seq=0 Win=65535 Len=0 M
32742 474.084398
                     39.156.66.18
                                         192.168.1.106
                                                              TCP
                                                                           66 80 → 54375 [SYN, ACK] Seg=0 Ack=1 Win=81
32743 474.084508
                     192.168.1.106
                                         39, 156, 66, 18
                                                              TCP
                                                                           54 54375 → 80 [ACK] Seg=1 Ack=1 Win=262144
```

```
Sequence Number: 1 (relate equence number)
  Sequence Number (raw): 1214102536
                          (relative sequence number)]
  [Next Sequence Number: 1
  Acknowledgment Number: 1
                            tive ack number)
  Acknowledgment number (raw): 1192279379
  0101 .... = Header Length: 20 bytes (5)

▼ Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... 1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... Not set
    [TCP Flags: ······A····]
  Window: 1024
  [Calculated window size: 262144]
```

第三次握手:客户端收到服务器发来的SYN+ACK报文段之后,客户端发送ACK报文段,ACK标志位为1,SYN标志位为0。确认号是服务器端来的SYN+ACK报文段中的序号加1,发送序号为SYN+ACK报文段中的确认号。服务器端收到客户端发来的ACK报文段,随后连接建成进行数据传输。

绘制<mark>第三次握手TCP</mark>报文首部:

	So	our	ce l	Por	t: 5	43	75			Destination Port: 80	
							S	eqı	uen	number: 1	
						Ad	ckn	ow	led	ent number: 1	
Header Length: 20 bytes	Header 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0							0	0	0	Window size value: 1024
	C	he	cks	um	:0x	a29	92				Urgent pointer: 0

实验结果:本次TCP连接释放是HTTP服务端首先发起的。

No.	Time	Source	Destination	Protocol	Length	Info			
	32749 474.118611	39.156.66.18	192.168.1.106	TCP	54	80 → 54375	[FIN,	ACK] Seq=	491 Ack=648 W
	32750 474.118723	192.168.1.106	39.156.66.18	TCP	54	54375 → 80	[ACK]	Seq=648 A	ck=492 Win=26
	32753 474.122485	192.168.1.106	39.156.66.18	TCP	54	54375 → 80	[FIN,	ACK] Seq=	648 Ack=492 W
	32755 474.129639	39.156.66.18	192.168.1.106	TCP	54	80 → 54375	[ACK]	Seq=492 A	ck=649 Win=79
<									>
	Sequence Number:	491 (rela	equence number)						
	Sequence Number (•	/c/c •			エロロス	> ㅁㅁᅩᄔ니 <i>나</i> ᅩ
	[Next Sequence Nu	ımber: 492 (rela	tive sequence numb	er)]	第一	次挥于:	HI	IP服务	P 器端发
		ımber: 648 📫 a		<u>.</u>	:坐	Λ EINI	子户	肥久里	器端到客
	Acknowledgment nu	ımber (raw): 121410	3183	•		L'EHN,	大 四	リルスプライ	谷纳扎合
		er Length: 20 bytes	(5)		一类	的数据信	丰详	(昭冬	哭啱不
	▼ Flags: 0x011 (FIN				الاللال ا		△1 ∇	(カスプン	百百四四二
		= Reserved: Not se	t	-	再发:	送报文约	合客戶	分端 。	伯可接
	0	= Nonce: Not set							
		= Congestion Windo		Not set	受客!	户端报文	之) 。	FIN	设立的
	0	-							
		= Urgent: Not set		$HN_{\overline{2}}$	江设置为] 1。:	Seg序	号为	
		= Acknowledgment:	•						
	0	= Push: Not set	491,	ACK号	∵刃64	48°			

Window: 2492

.... .0.. = Reset: Not set0. = Syn: Not set

>1 = Fin: Set > [TCP Flags:A...F]

绘制第一次挥手TCP报文首部:

		So	urc	e P	ort	: 80)			Destination Port: 54375	
							Se	qu	enc	umber: 491	
						Ack	no	wle	edg	me	nt number: <mark>648</mark>
Header Length: 20 bytes	Length: 0 20 0								0	1	Window size value: 2492
	C	hed	cks	um	:0x	986	64				Urgent pointer: 0

实验结果:

```
Time
                                            Destination
No.
                        Source
                                                                Protocol
                                                                          Length Info
                                                                             54 80 → 54375 [FIN, ACK] Seq=491 Ack=648 Wi
   32749 474.118611
                        39.156.66.18
                                            192,168,1,106
                                                                TCP
                                                                             54 54375 → 80 [ACK] Seq=648 Ack=492 Win=261
   32750 474.118723
                        192,168,1,106
                                            39.156.66.18
                                                                TCP
                                                                             54 54375 → 80 [FIN, ACK] Seg=648 Ack=492 Wi
   32753 474.122485
                        192.168.1.106
                                            39.156.66.18
                                                                TCP
   32755 474.129639
                        39.156.66.18
                                            192.168.1.106
                                                                             54 80 → 54375 [ACK] Seg=492 Ack=649 Win=797
                                                                TCP
```

```
Sequence Number: 648 (rela sequence number)
 Sequence Number (raw): 1214103183
  [Next Sequence Number: 648 (relative sequence number)]
  Acknowledgment number (raw): 1192279870
  0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ······A····]
  Window: 1022
```

第二次挥手:客户端收到这个FIN之后发回一个ACK报文,它的Seq序号是收到的服务器端发送的FIN报文的Ack号648,它的Ack号是收到的服务器端发送的FIN报文的Seq序号加1,即491+1为492。

绘制第二次挥手TCP报文首部:

	So	our	ce	Por	t: 5	43	75			Destination Port: 80	
							Se	qu	enc	umber: <mark>648</mark>	
						Ack	no	wle	edg	me	nt number: 492
Header Length: 20 bytes	Length: 0 20 0								0	0	Window size value: 1022
	C	he	cks	um	:0x	9e2	22				Urgent pointer: 0

实验结果:

```
No.
       Time
                                 Destination
                                                Protocol
                  Source
                                                        Length Info
  32749 474.118611
                                                          54 80 → 54375 [FIN, ACK] Seq=491 Ack=648 Wi
                  39.156.66.18
                                 192,168,1,106
                                                TCP
                                                          54 54375 → 80 [ACK] Seq=648 Ack=492 Win=261
  32750 474.118723
                  192.168.1.106
                                 39.156.66.18
                                                TCP
                                 39.156.66.18
                                                          54 54375 → 80 [FIN, ACK] Seq=648 Ack=492 Wi
  32753 474.122485
                  192.168.1.106
                                                TCP
                                                          54 80 → 54375 [ACK] Seg=492 Ack=649 Win=797
  32755 474.129639
                  39.156.66.18
                                 192.168.1.106
                                                TCP
   Sequence Number: 648 (related sequence number)
   Sequence Number (raw): 1214103183
                                                      第三次挥手: 当服务器端关闭
    [Next Sequence Number: 649 (relative sequence number)]
   TCP连接之后,接下来进行的
   Acknowledgment number (raw): 1192279870
   0101 .... = Header Length: 20 bytes (5)
                                                      是客户端关闭TCP连接。客户
  Flags: 0x011 (FIN, ACK)
     000. .... = Reserved: Not set
                                                      端发送一个FIN,关闭客户端到
      ...0 .... = Nonce: Not set
                                                      HTTP服务器端的数据传送。
      .... 0... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
                                                      FIN数据报的FIN位设置为1,
      .... ..0. .... = Urgent: Not set
      .... = Acknowledgment: Set
                                                      ACK位设置为1。
      .... Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
    > .... Set
    > [TCP Flags: ·····A···F]
    Window: 1022
```

绘制第三次挥手TCP报文首部:

	So	our	ce l	Por	t: 5	43	75			Destination Port: 80	
							Se	qu	enc	umber: 648	
						Ack	no	wle	edg	nt number: 492	
Header Length: 20 bytes	0 0 0	0	0	0	0	1	0	0	0	1	Window size value: 1022
	С	hed	cks	um	:0x	9e2	21				Urgent pointer: 0

实验结果:

```
Time
                        Source
                                            Destination
                                                                Protocol
                                                                          Length Info
No.
   32749 474.118611
                        39.156.66.18
                                            192.168.1.106
                                                                TCP
                                                                             54 80 → 54375 [FIN, ACK] Seq=491 Ack=648 Wi
   32750 474.118723
                                            39.156.66.18
                                                                             54 54375 → 80 [ACK] Seq=648 Ack=492 Win=261
                        192.168.1.106
                                                                TCP
                                           39.156.66.18
                                                                             54 54375 → 80 [FIN, ACK] Seg=648 Ack=492 Wi
   32753 474.122485
                        192.168.1.106
                                                                TCP
                                                                             54 80 → 54375 [ACK] Seq=492 Ack=649 Win=797
   32755 474.129639
                        39.156.66.18
                                            192.168.1.106
                                                                TCP
```

```
Sequence Number: 492 (relaters sequence number)
  Sequence Number (raw): 1192279870
  [Next Sequence Number: 492
                              (relative sequence number)]
  Acknowledgment Number: 649
                             Analative ack number)
  Acknowledgment number (raw): 1214103184
  0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... 1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... Not set
    [TCP Flags: ······A····]
  Window: 2492
```

第四次挥手: HTTP服务器端收到上一个FIN+ACK之后发回一个ACK报文, ACK标志位置1。它的Seq序号是收到的客户端发送的FIN+ACK报文的Ack号492,它的Ack号是收到的客户端发送的FIN+ACK数据报的Seq序号加1,即648+1为649。

绘制第四次挥手TCP报文首部:

		So	urc	e P	ort	: 80)			Destination Port: 54375	
							Se	qu	enc	umber: 492	
						Ack	no	wle	edg	nt number: <mark>649</mark>	
Header Length: 20 bytes	Header 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0								0	0	Window size value: 2492
	С	hed	cks	um	:0x	986	63				Urgent pointer: 0

思考题

思考题: 抓取并观察TCP Keep-Alive报文, 分析其使用意图。

