TCP数据分析实验

同济大学软件学院

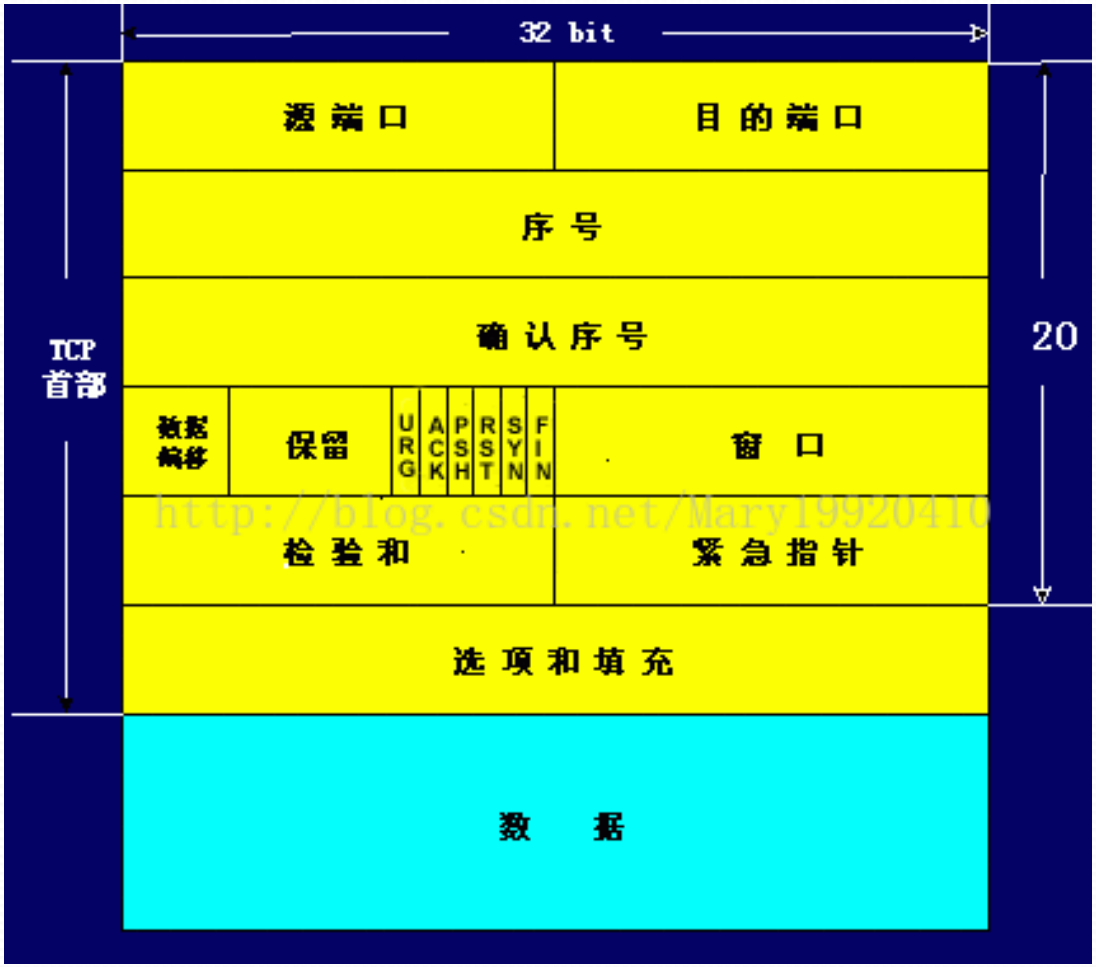
TCP数据

1. TCP的概述

TCP是**传输层**的协议，功能即为在IP的数据报服务之上增加了最基本的服务：**复用**和**分用**以及**差错检测**。TCP 是一个基于连接的四层协议，提供全双工地，可靠地传输系统。它能够保证数据被远程主机接收。并且能够为高层协议提供flow-controlled 服务。空间上，TCP需要在端系统中**维护连接**状态，需要一定的开销。此连接装入包括接收和发送缓存，拥塞控制参数和序号与确认号的参数。UDP不维护连接状态，也不跟踪这些参数，开销小。空间和时间上都具有优势。

2. TCP的报文格式

TCP报文是TCP层传输的数据单元，也叫报文段。



TCP数据

3. TCP报文字段

A、端口号：用来标识同一台计算机的不同的应用进程。

1）源端口：源端口和IP地址的作用是标识报文的返回地址。

2）目的端口：端口指明接收方计算机上的应用程序接口。

TCP报头中的源端口号和目的端口号同IP数据报中的源IP与目的IP唯一确定一条TCP连接。

TCP数据

3. TCP报文字段

B、序号和确认号：是TCP可靠传输的关键部分。**序号**是本报文段发送的数据组的第一个字节的序号。在TCP传送的流中，每一个字节一个序号。例如：一个报文段的序号为300，此报文段数据部分共有100字节，则下一个报文段的序号为400。所以序号确保了TCP传输的有序性。确认号，即ACK，指明下一个期待收到的字节序号，表明该序号之前的所有数据已经正确无误的收到。确认号只有当ACK标志为1时才有效。比如建立连接时，SYN报文的ACK标志位为0。

3. TCP报文字段

C、数据偏移／首部长度：4bits。由于首部可能含有可选项内容，因此TCP报头的长度是不确定的，报头不包含任何任选字段则长度为20字节，4位首部长度字段所能表示的最大值为1111，转化为10进制为15，15*32/8 = 60，故报头最大长度为60字节。首部长度也叫数据偏移，是因为首部长度实际上指示了数据区在报文段中的起始偏移值。

D、保留：为将来定义新的用途保留，现在一般置0。

3. TCP报文字段

E、控制位：URG ACK PSH RST SYN FIN，共6个，每一个标志位表示一个控制功能。

1）URG：紧急指针标志，为1时表示紧急指针有效，为0则忽略紧急指针。

2）ACK：确认序号标志，为1时表示确认号有效，为0表示报文中不含确认信息，忽略确认号字段。

3）PSH：push标志，为1表示是带有push标志的数据，指示接收方在接收到该报文段以后，应尽快将这个报文段交给应用程序，而不是在缓冲区排队。

3. TCP报文字段

E、控制位：URG ACK PSH RST SYN FIN

4）RST：重置连接标志，用于重置由于主机崩溃或其他原因而出现错误的连接。或者用于拒绝非法的报文段和拒绝连接请求。

5）SYN：同步序号，用于建立连接过程，在连接请求中，SYN=1和ACK=0表示该数据段没有使用捎带的确认域，而连接应答捎带一个确认，即SYN=1和ACK=1。

6）FIN：finish标志，用于释放连接，为1时表示发送方已经没有数据发送了，即关闭本方数据流。

TCP数据

3. TCP报文字段

F、窗口：滑动窗口大小，用来告知发送端接受端的缓存大小，以此控制发送端发送数据的速率，从而达到流量控制。窗口大小时一个16bit字段，因而窗口大小最大为65535。

G、校验和：奇偶校验，此校验和是对整个的 TCP 报文段，包括 TCP 头部和 TCP 数据，以 16 位字进行计算所得。由发送端计算和存储，并由接收端进行验证。

3. TCP报文字段

H、紧急指针：只有当*URG*标志置*1*时紧急指针才有效。紧急指针是一个正的偏移量，和顺序号字段中的值相加表示紧急数据最后一个字节的序号。*TCP*的紧急方式是发送端向另一端发送紧急数据的一种方式。

I、选项和填充：最常见的可选字段是最长报文大小，又称为MSS（Maximum Segment Size），每个连接方通常都在通信的第一个报文段（为建立连接而设置SYN标志为1的那个段）中指明这个选项，它表示本端所能接受的最大报文段的长度。

TCP数据

3. TCP报文字段

I、选项和填充：选项长度不一定是32位的整数倍，所以要加填充位，即在这个字段中加入额外的零，以保证TCP头是32的整数倍。

J、数据部分：*TCP* 报文段中的数据部分是可选的。在一个连接建立和一个连接终止时，双方交换的报文段仅有 *TCP* 首部。如果一方没有数据要发送，也使用没有任何数据的首部来确认收到的数据。在处理超时的许多情况中，也会发送不带任何数据的报文段。

TCP数据

3. TCP报文字段

I、选项和填充：选项长度不一定是32位的整数倍，所以要加填充位，即在这个字段中加入额外的零，以保证TCP头是32的整数倍。

J、数据部分：*TCP* 报文段中的数据部分是可选的。在一个连接建立和一个连接终止时，双方交换的报文段仅有 *TCP* 首部。如果一方没有数据要发送，也使用没有任何数据的首部来确认收到的数据。在处理超时的许多情况中，也会发送不带任何数据的报文段。

4. TCP连接过程

相对于SOCKET开发者,TCP创建过程和链接折除过程是由TCP/IP协议栈自动创建的。因此开发者并不需要控制这个过程。但是对于理解TCP底层运作机制，相当有帮助。TCP连接过程简单一句话概括:“三次握手四次挥手”。

A. TCP三次握手

所谓三次握手(Three-way Handshake)，是指建立一个TCP连接时，需要客户端和服务器总共发送3个包。三次握手的目的是连接服务器指定端口，建立TCP连接,并同步连接双方的序列号和确认号并交换 TCP 窗口大小信息.在socket编程中，客户端执行connect()时。将触发三次握手。

4. TCP连接过程

第一次握手:

客户端发送一个TCP的SYN标志位置1的包指明客户打算连接的服务器的端口，以及初始序号X,保存在包头的序列号(Sequence Number)字段里。

<table>
<tr><td colspan="5">源端口</td><td colspan="3">目标端口</td></tr>
<tr><td colspan="8">X</td></tr>
<tr><td colspan="8">接收顺序号</td></tr>
<tr><td>偏置值</td><td>保留</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>1</td><td>FIN</td><td>窗口</td></tr>
<tr><td colspan="5">检查和</td><td colspan="3">紧急指针</td></tr>
<tr><td colspan="8">任选项+补丁</td></tr>
<tr><td colspan="8">用户数据</td></tr>
</table>

4. TCP连接过程

第二次握手:

服务器发回确认包(ACK)应答。即SYN标志位和ACK标志位均为1同时，将确认序号(Acknowledgement Number)设置为客户的ISN加以1.即X+1。

<table>
<tr><td colspan="4">源端口</td><td colspan="4">目标端口</td></tr>
<tr><td colspan="8">Y</td></tr>
<tr><td colspan="8">X+1</td></tr>
<tr><td>偏置值</td><td>保留</td><td>URG</td><td>1</td><td>PSH</td><td>RST</td><td>1</td><td>FIN</td><td colspan="2">窗口</td></tr>
<tr><td colspan="4">检查和</td><td colspan="4">紧急指针</td></tr>
<tr><td colspan="8">任选项+补丁</td></tr>
<tr><td colspan="8">用户数据</td></tr>
</table>

TCP数据

4. TCP连接过程

第三次握手.

客户端再次发送确认包(ACK) SYN标志位为0,ACK标志位为1.并且把服务器发来ACK的序号字段+1,放在确定字段中发送给对方.并且在数据段放写ISN的+1.

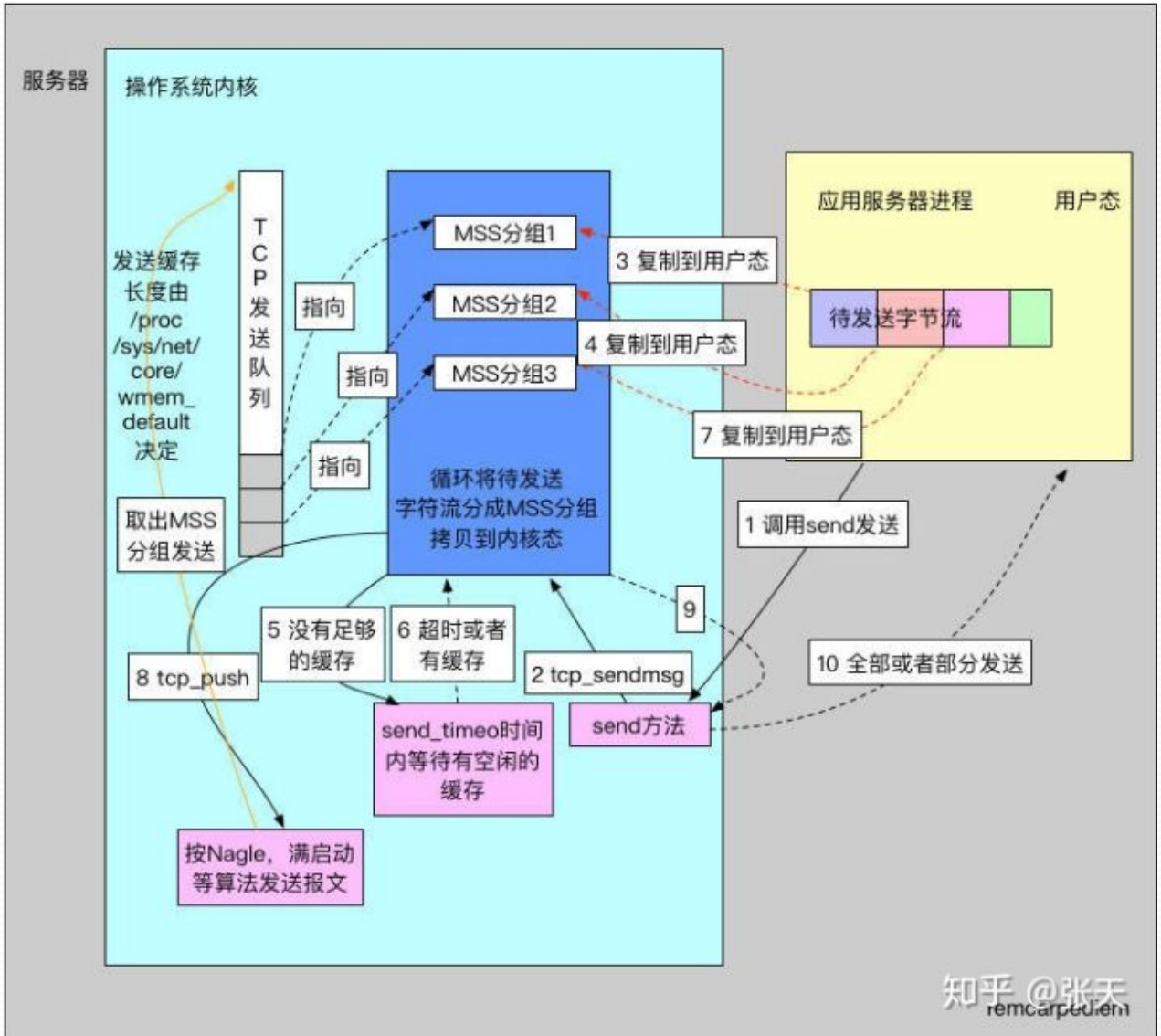
源端口									目标端口
发送顺序号									
Y+1									
偏置值	保留		URG	1	PSH	RST	SYN	FIN	窗口
检查和									紧急指针
任选项+补丁									
DATA（X+1）									

4. TCP连接过程

Tip:**SYN攻击** 在三次握手过程中，服务器发送SYN-ACK之后，收到客户端的ACK之前的TCP连接称为半连接(half-open connect).此时服务器处于Syn_RECV状态.当收到ACK后，服务器转入ESTABLISHED状态。攻击客户端在短时间内伪造大量不存在的IP地址，向服务器不断地发送syn包，服务器回复确认包，并等待客户的确认，由于源地址是不存在的，服务器需要不断的重发直至超时，这些伪造的SYN包将长时间占用未连接队列，正常的SYN请求被丢弃，目标系统运行缓慢，严重者引起网络堵塞甚至系统瘫痪。

4. TCP连接过程

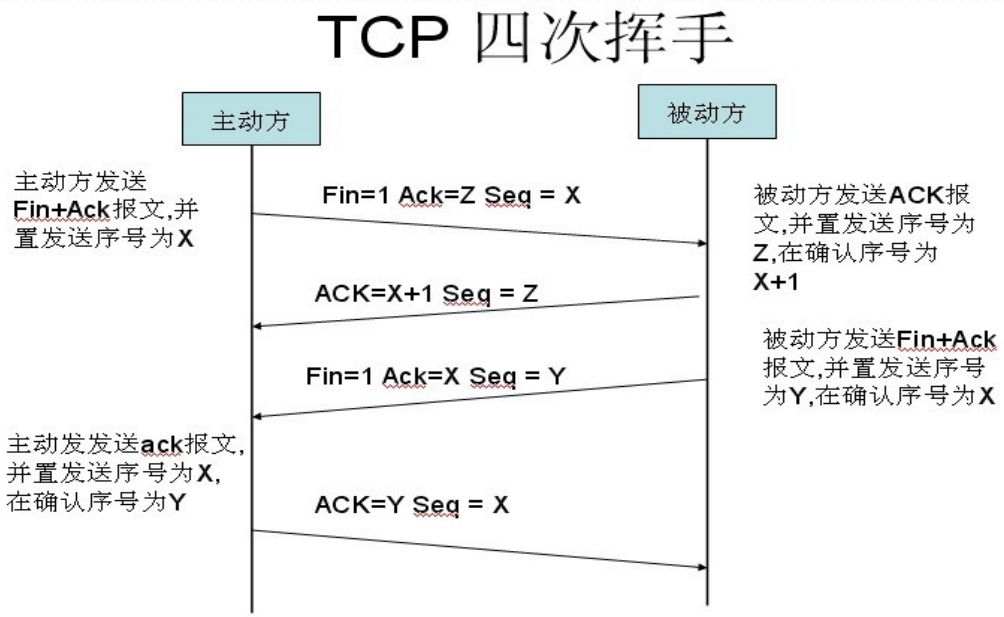
Tip2:TCP发送报文，操作系统内核与用户态



4. TCP连接过程

B.TCP四次挥手

TCP的连接的拆除需要发送四个包，因此称为四次挥手(four-way handshake)。客户端或服务器均可主动发起挥手动作，在socket编程中，任何一方执行close()操作即可产生挥手操作。



http://bluedrum.cublog.cn

4. TCP连接过程

C. TCP三次握手会涉及TCP的状态转换图如下

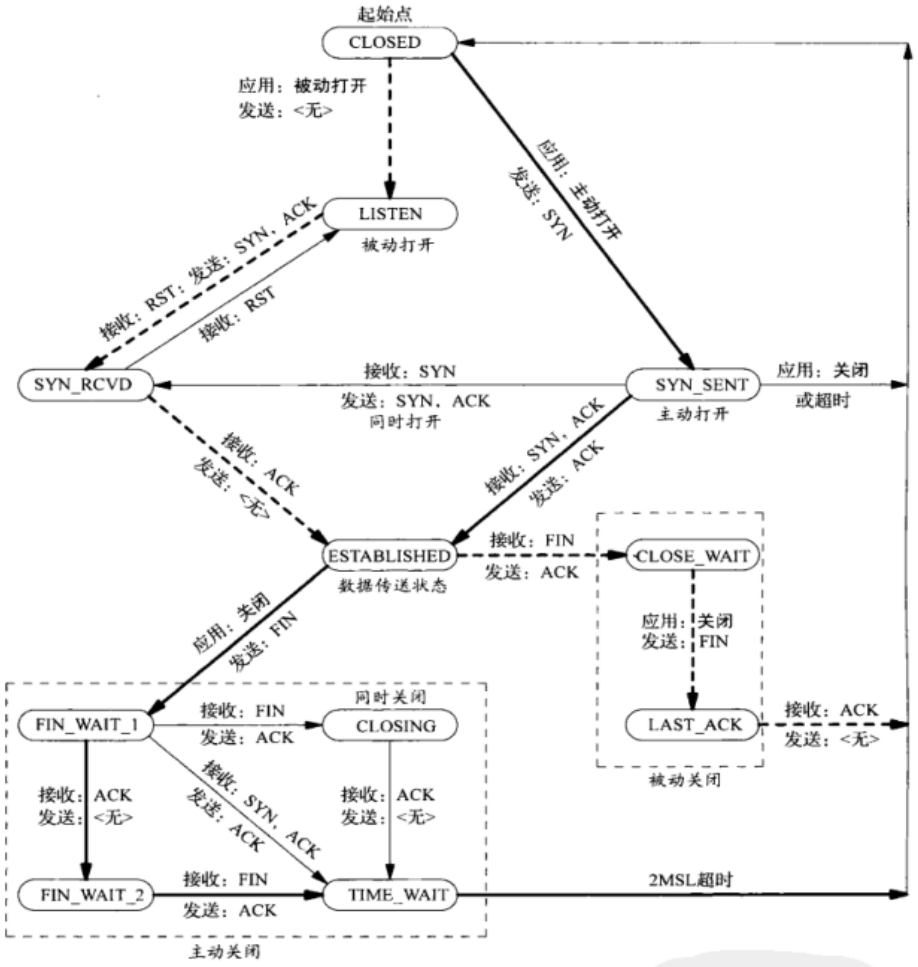


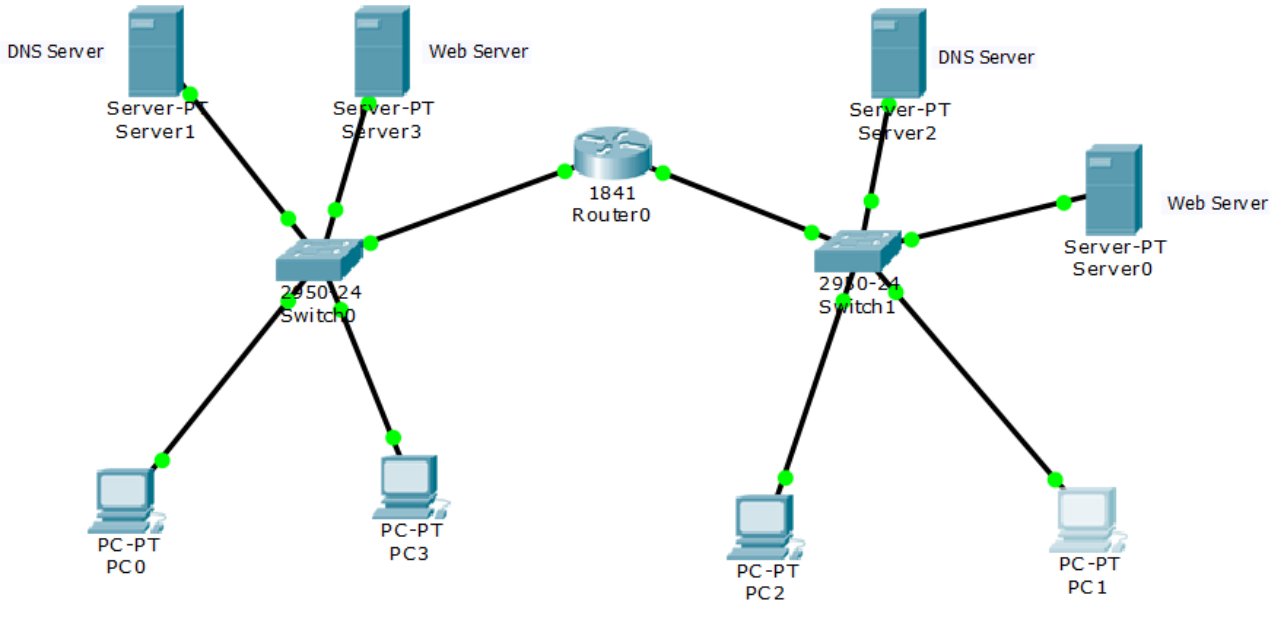
图2-4 TCP状态转换图

TCP数据报文

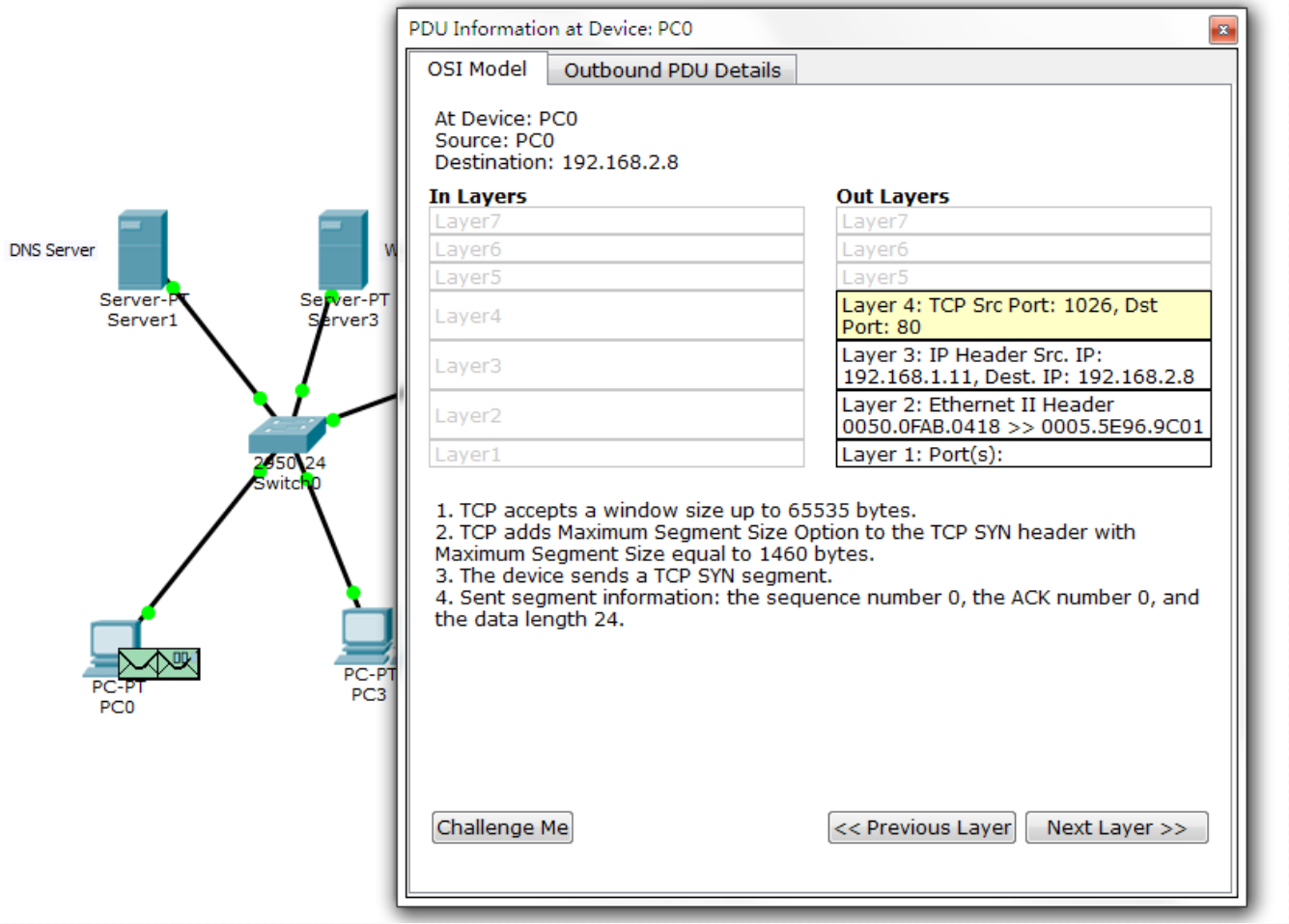
5.Packet Tracer 分析TCP报文

网络结构图

1）设置WEB服务器和简单的DNS服务器；2)打开PC0浏览器，输入配置Web服务器的Web地址，如www.tongji.edu.cn,产生TCP数据报文。

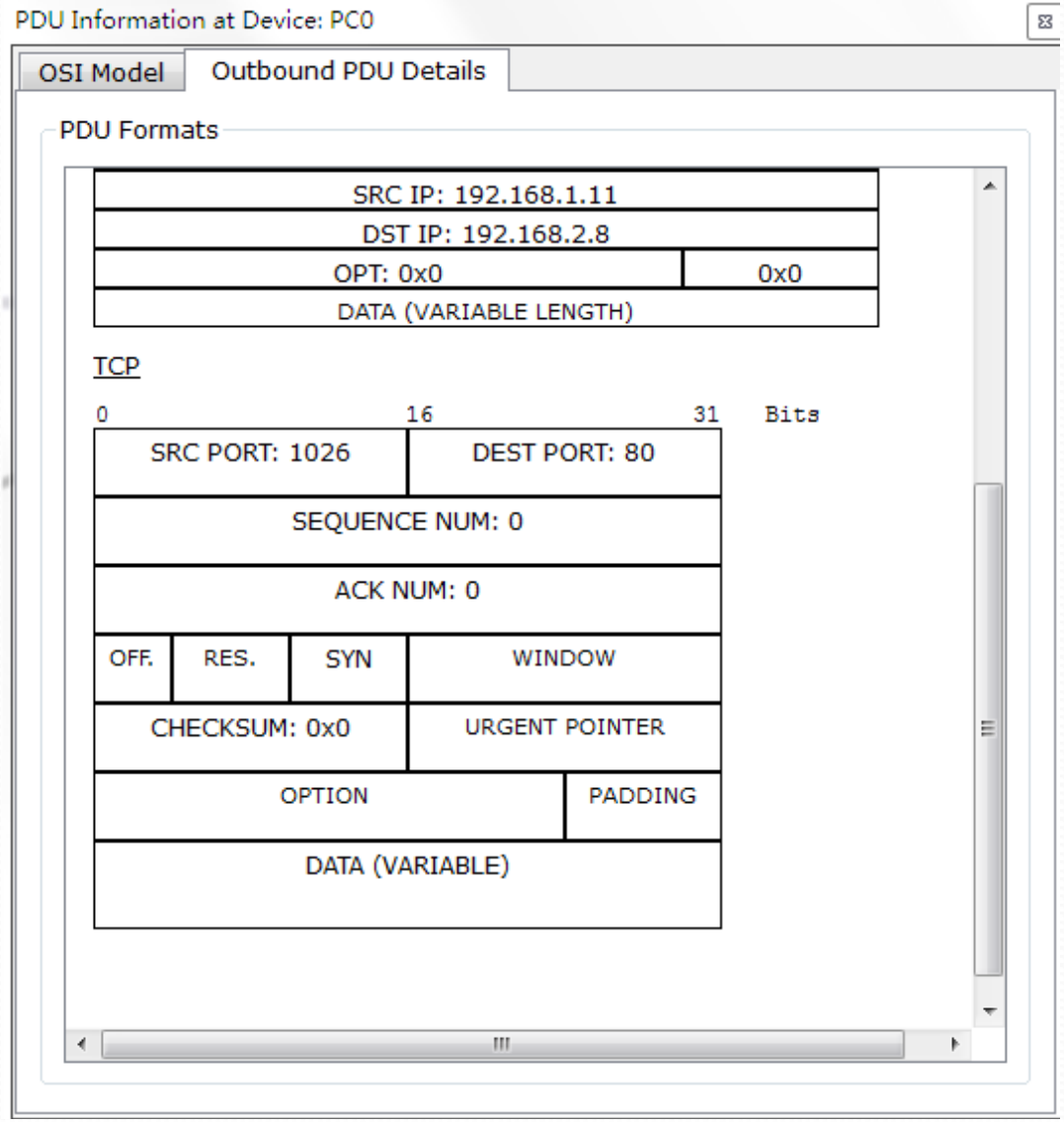


5 .Packet Tracer 分析TCP报文



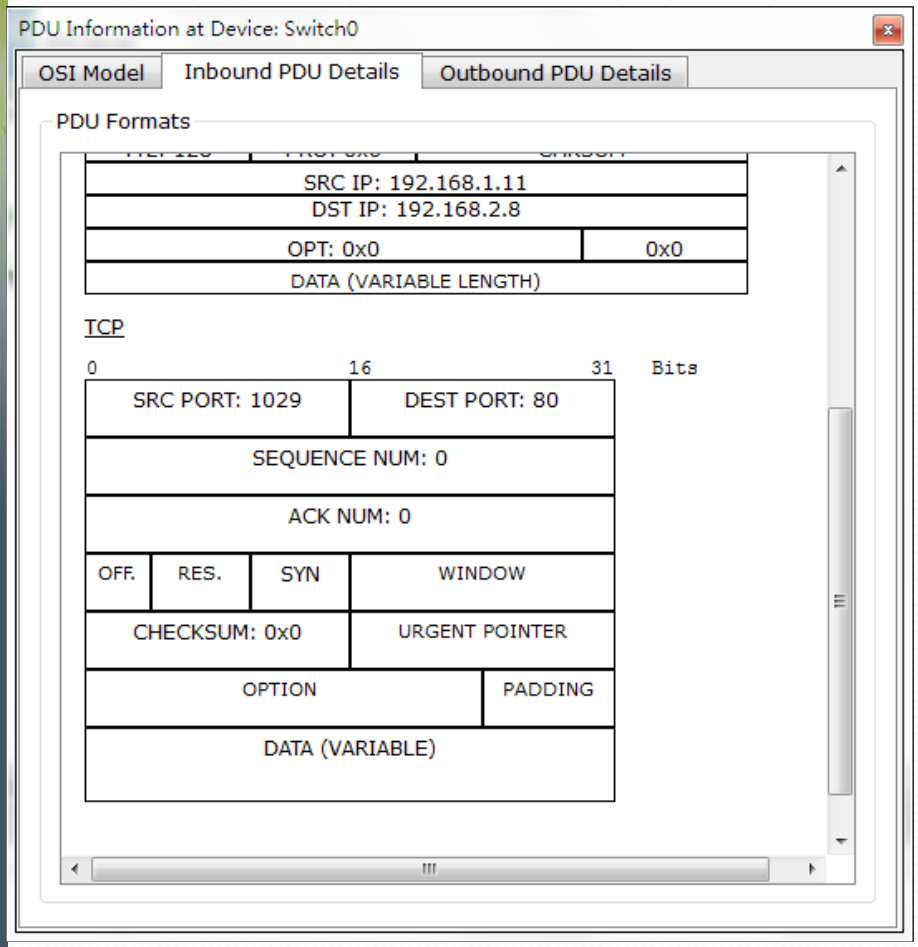
TCP数据报文

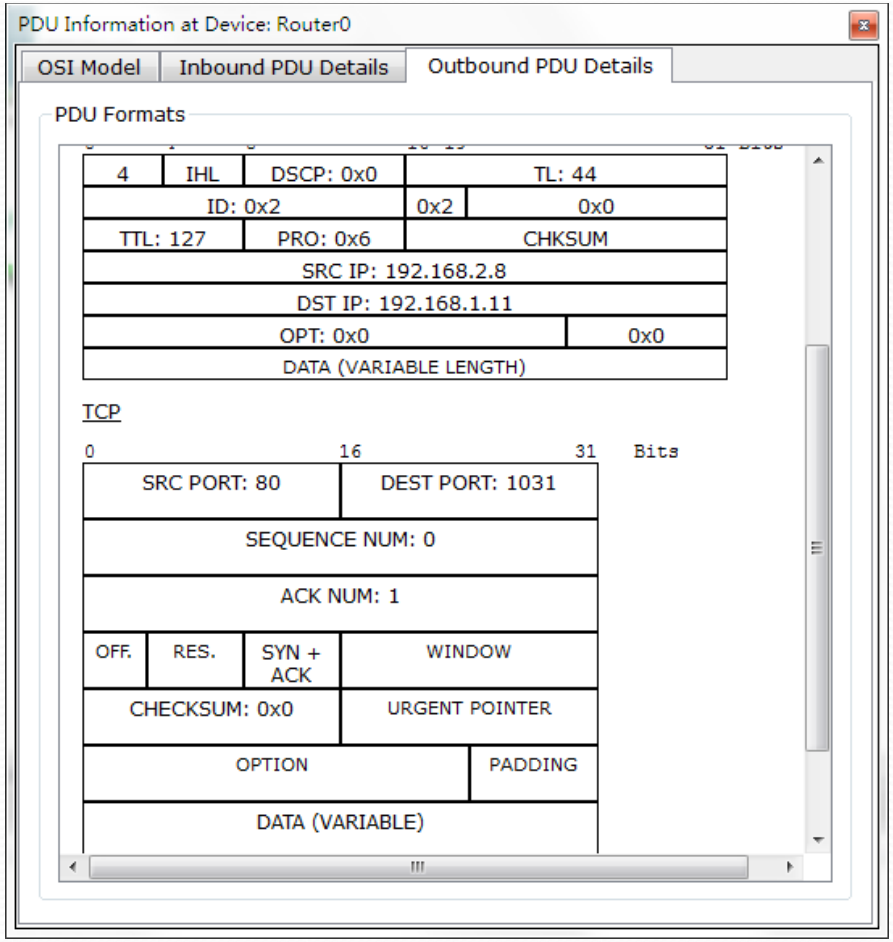
5 .Packet Tracer 分析TCP报文



TCP数据报文

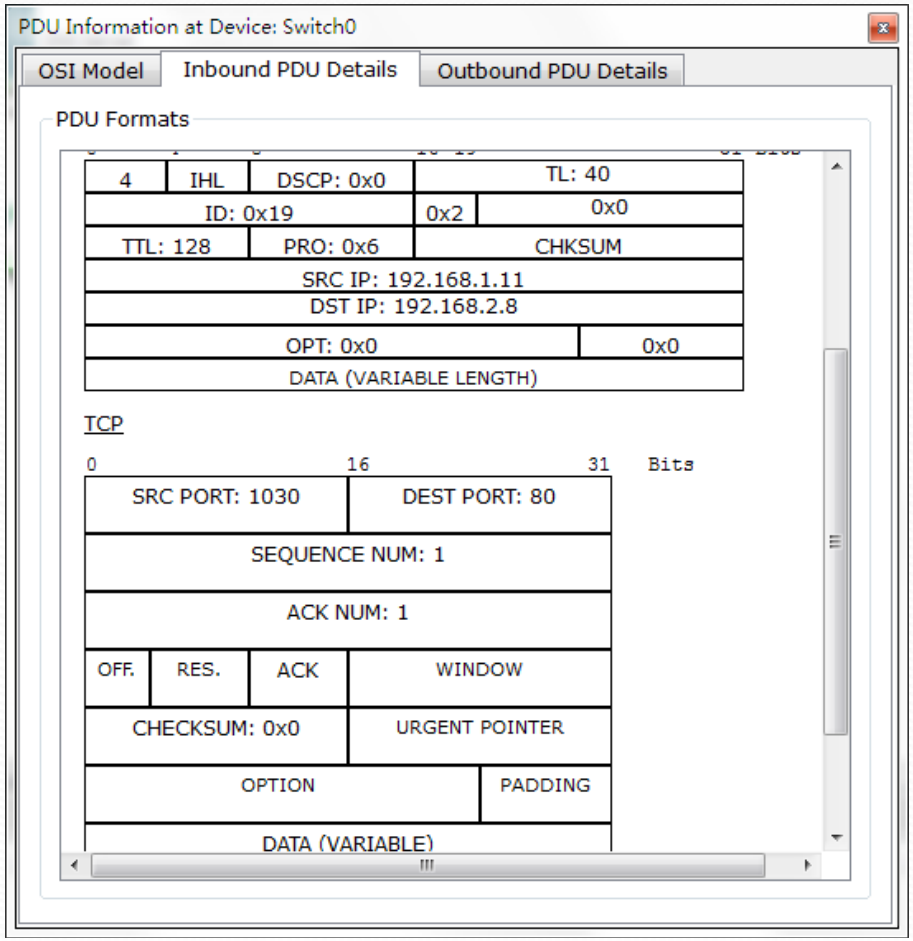
5 .Packet Tracer 分析TCP报文

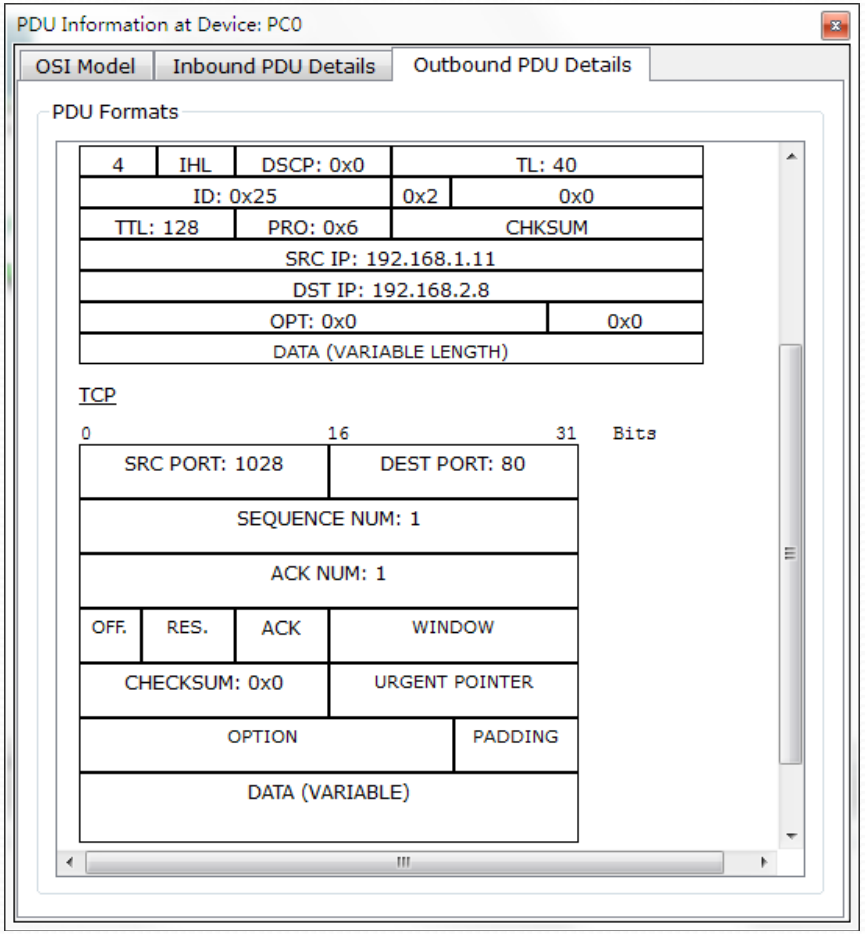




TCP数据报文

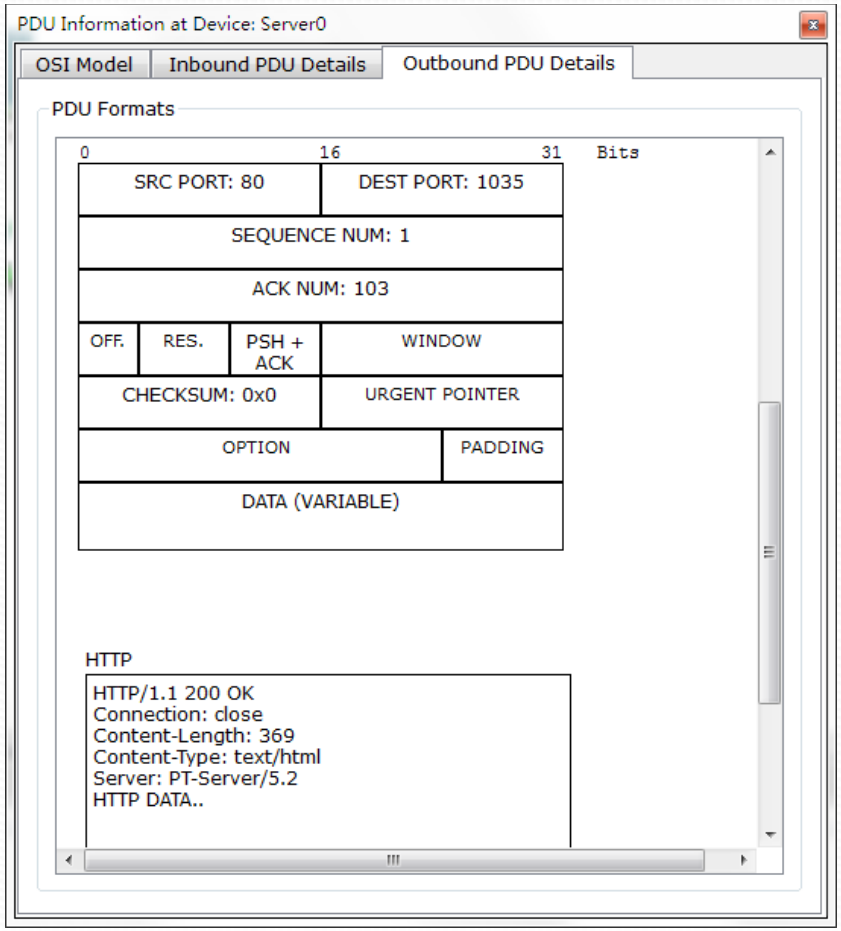
5 .Packet Tracer 分析TCP报文

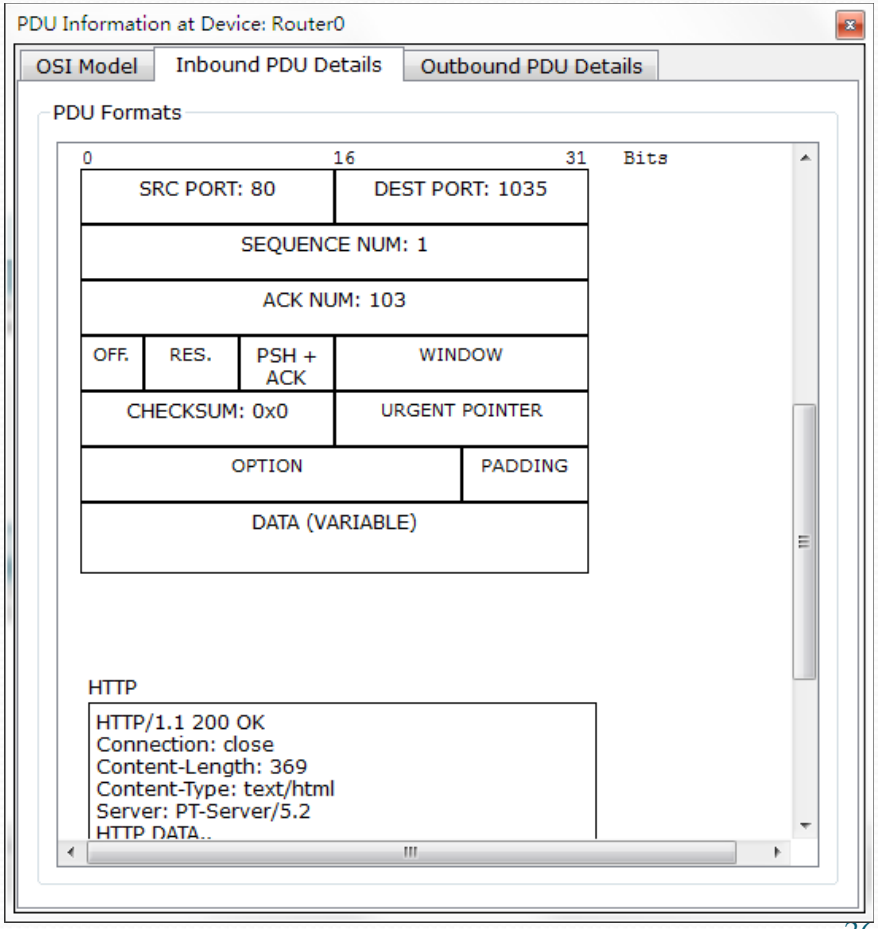




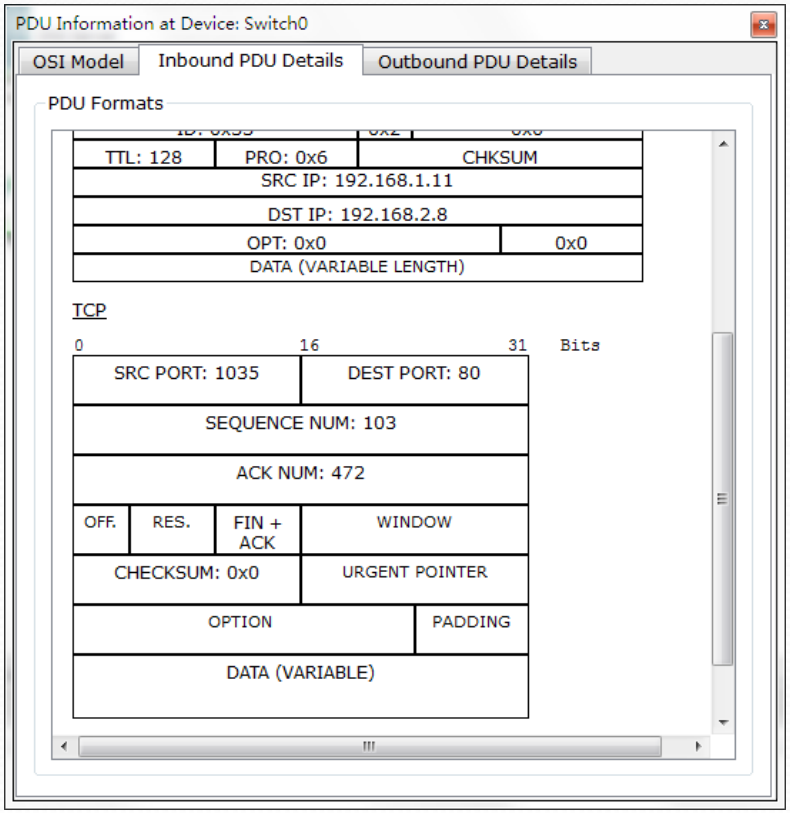
TCP数据报文

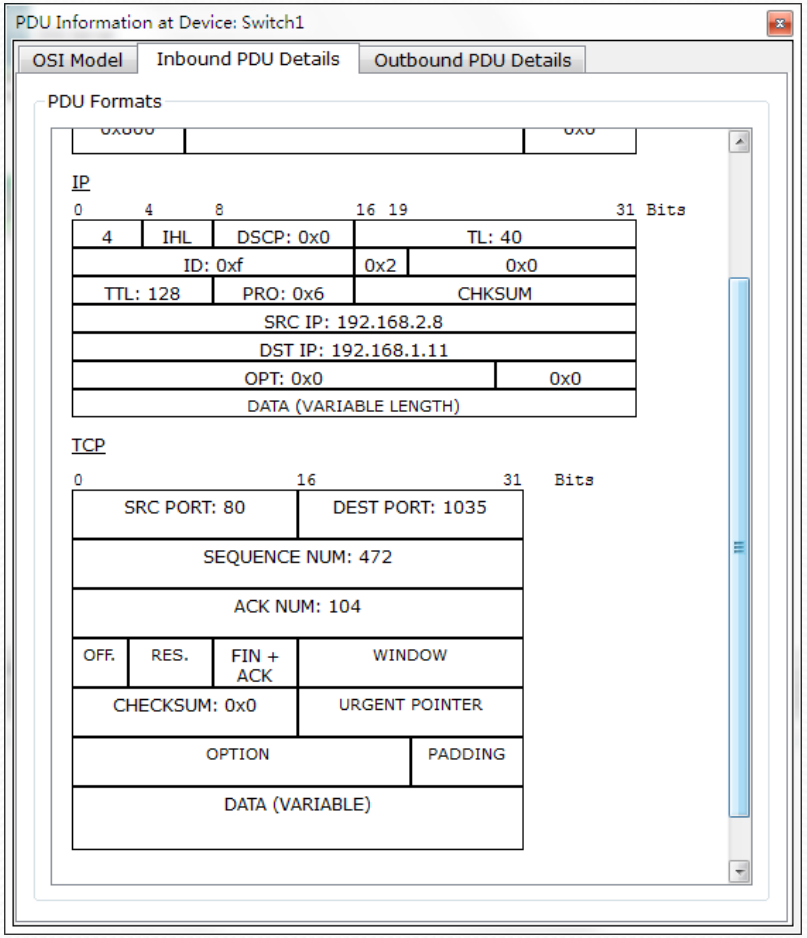
5 .Packet Tracer 分析TCP报文



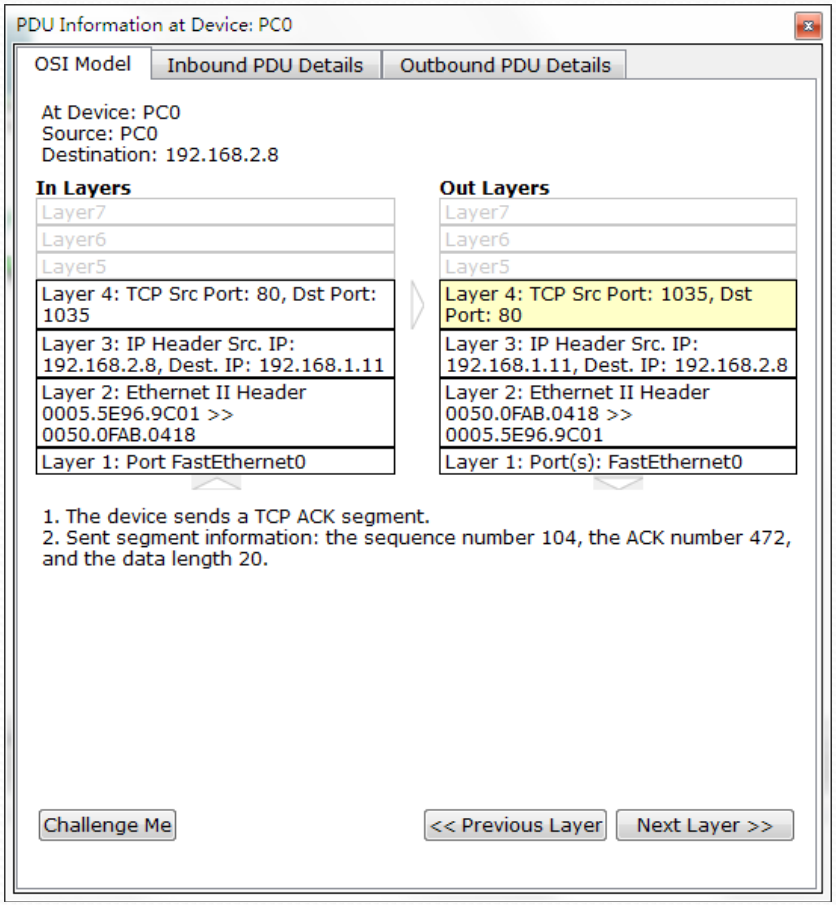


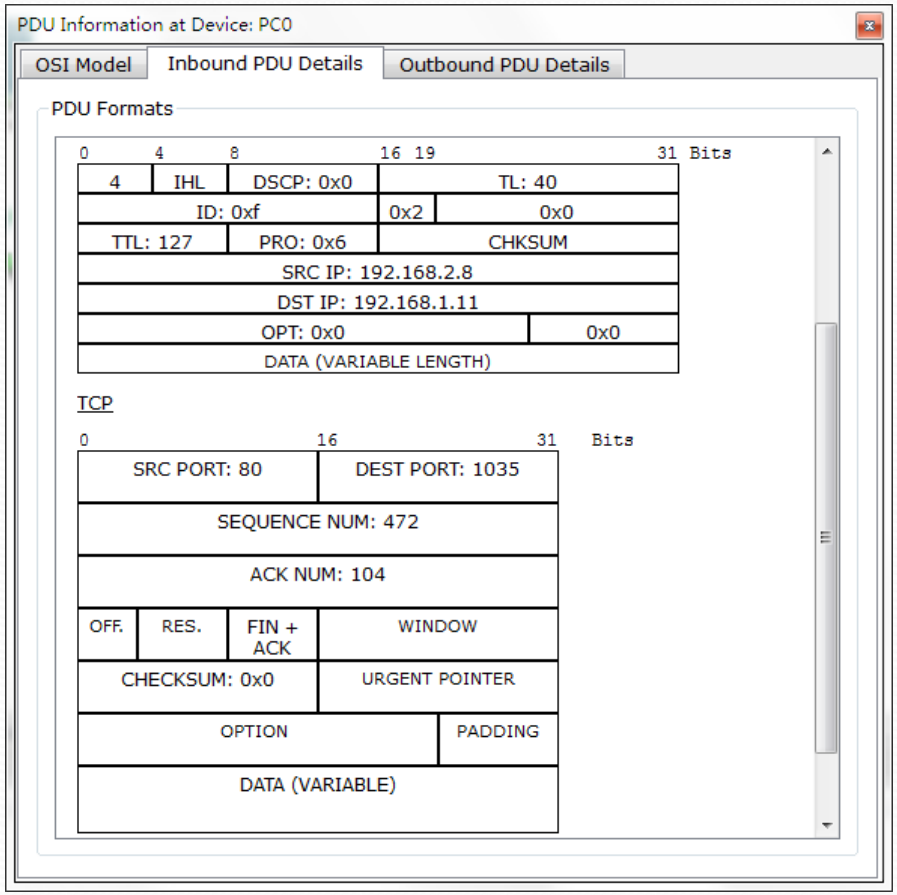
5 .Packet Tracer 分析TCP报文





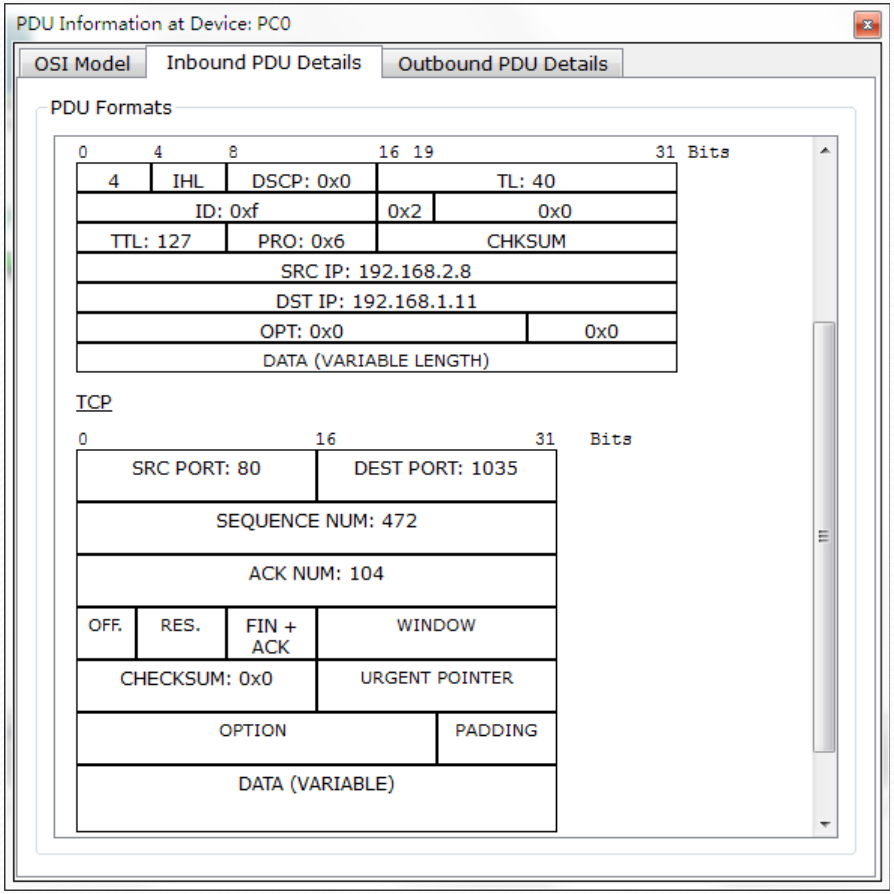
5 .Packet Tracer 分析TCP报文

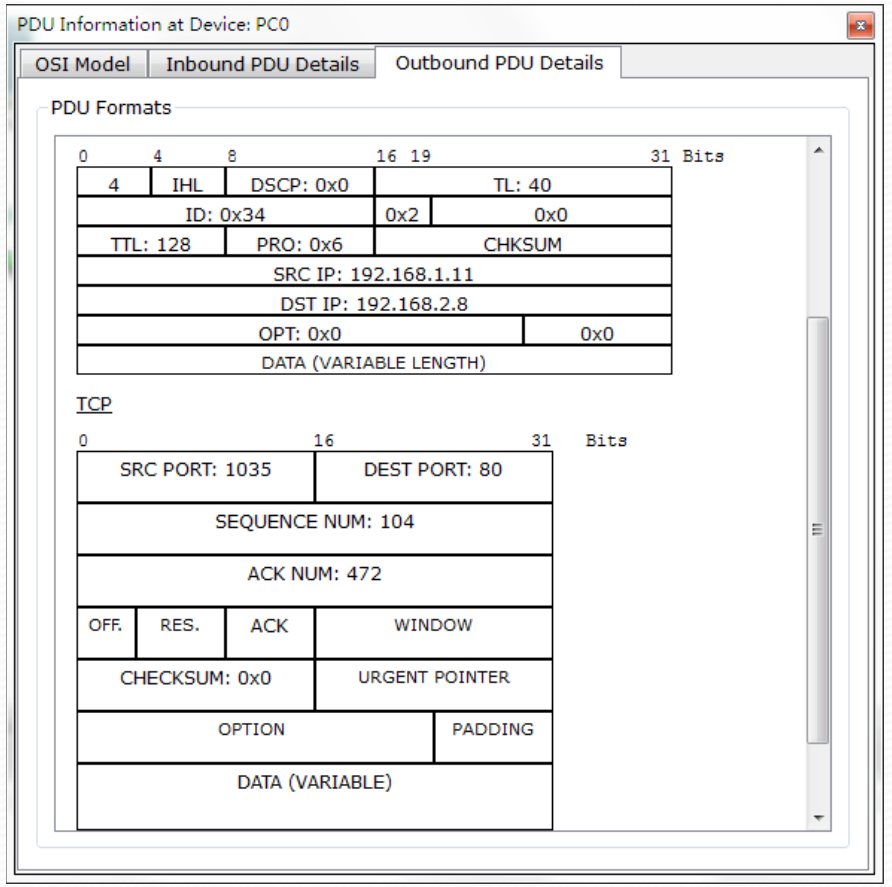




TCP数据报文

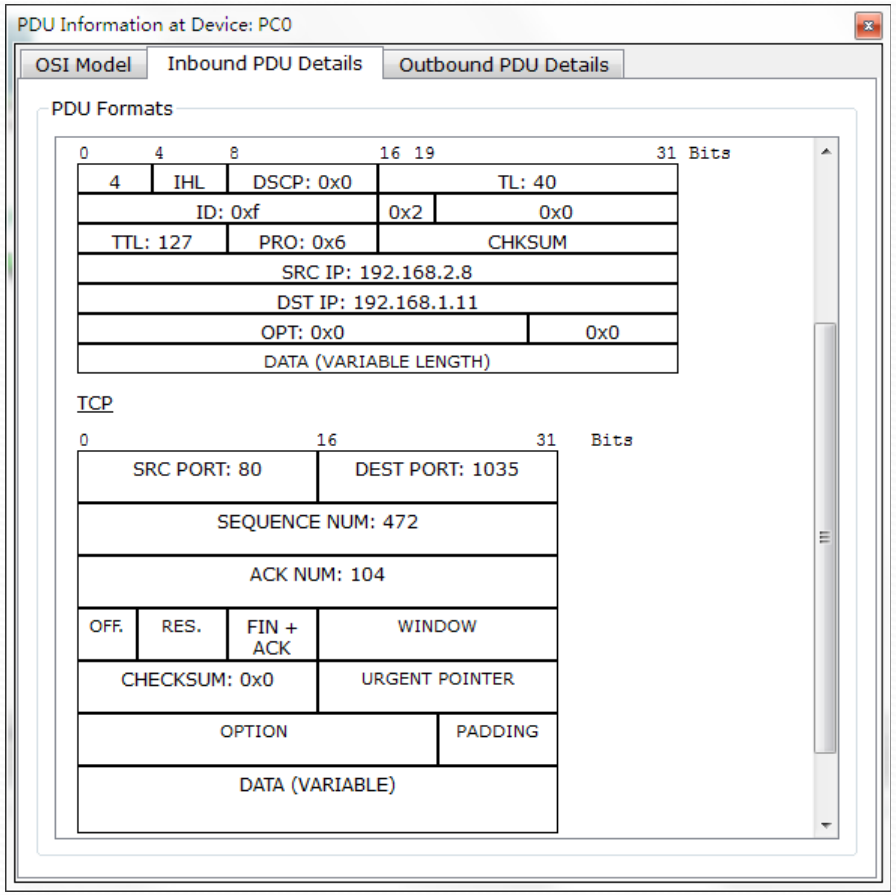
5 .Packet Tracer 分析TCP报文

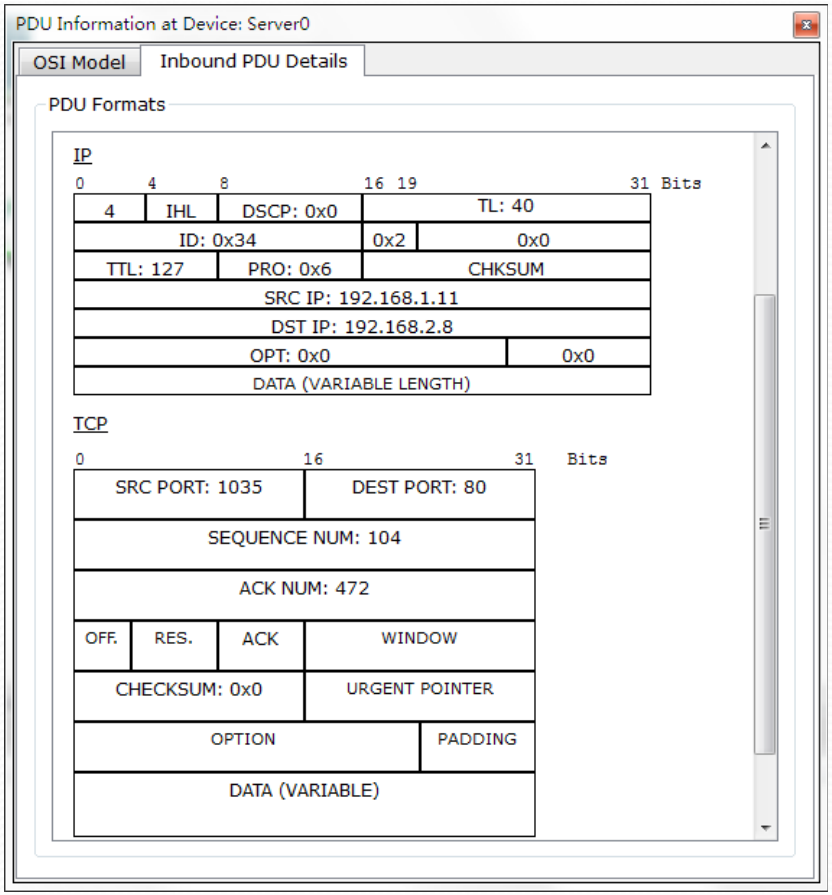




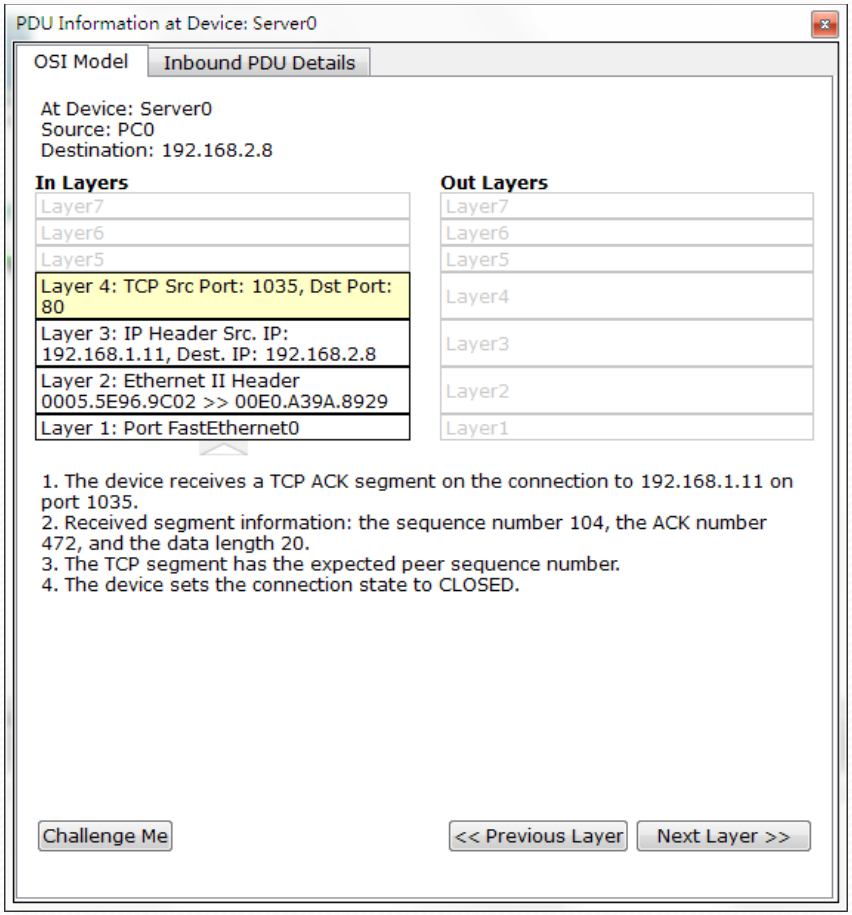
TCP数据报文

5 .Packet Tracer 分析TCP报文





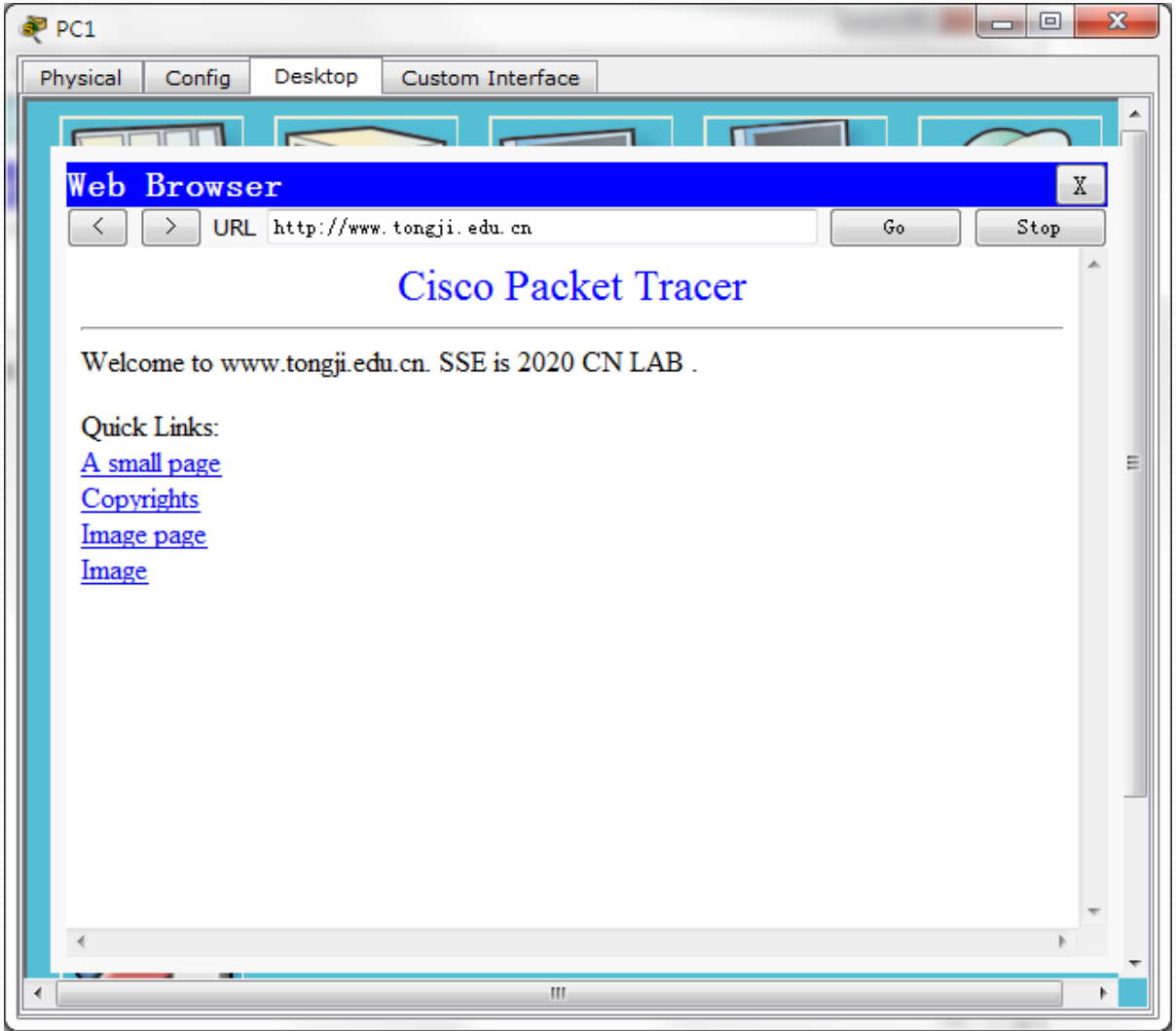
5 .Packet Tracer 分析TCP报文



TCP数据报文

4. Packet Tracer 分析报文

PC1 WEB Browser



5. WireShark TCP报文抓取分析

Microsoft - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

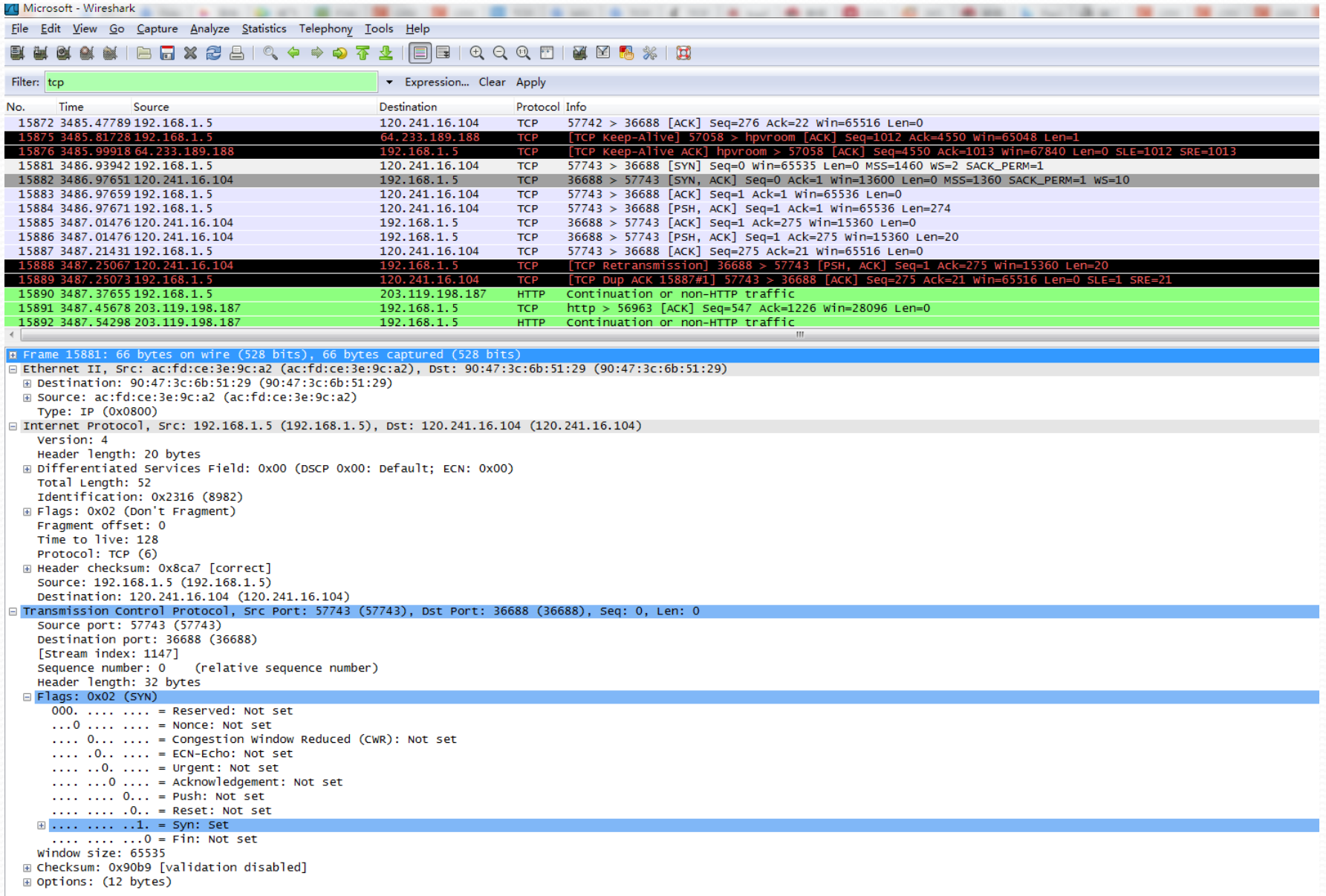
Filter: tcp Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
15872	3485.47789	192.168.1.5	120.241.16.104	TCP	57742 > 36688 [ACK] Seq=276 Ack=22 Win=65516 Len=0
15875	3485.81728	192.168.1.5	64.233.189.188	TCP	[TCP Keep-Alive] 57058 > hpvroom [ACK] Seq=1012 Ack=4550 Win=65048 Len=1
15876	3485.99918	64.233.189.188	192.168.1.5	TCP	[TCP Keep-Alive ACK] hpvroom > 57058 [ACK] Seq=4550 Ack=1013 Win=67840 Len=0 SLE=1012 SRE=1013
15881	3486.93942	192.168.1.5	120.241.16.104	TCP	57743 > 36688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
15882	3486.97651	120.241.16.104	192.168.1.5	TCP	36688 > 57743 [SYN, ACK] Seq=0 Ack=1 Win=13600 Len=0 MSS=1360 SACK_PERM=1 WS=10
15883	3486.97659	192.168.1.5	120.241.16.104	TCP	57743 > 36688 [ACK] Seq=1 Ack=1 Win=65536 Len=0
15884	3486.97671	192.168.1.5	120.241.16.104	TCP	57743 > 36688 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=274
15885	3487.01476	120.241.16.104	192.168.1.5	TCP	36688 > 57743 [ACK] Seq=1 Ack=275 Win=15360 Len=0
15886	3487.01476	120.241.16.104	192.168.1.5	TCP	36688 > 57743 [PSH, ACK] Seq=1 Ack=275 Win=15360 Len=20
15887	3487.21431	192.168.1.5	120.241.16.104	TCP	57743 > 36688 [ACK] Seq=275 Ack=21 Win=65516 Len=0
15888	3487.25067	120.241.16.104	192.168.1.5	TCP	[TCP Retransmission] 36688 > 57743 [PSH, ACK] Seq=1 Ack=275 Win=15360 Len=20
15889	3487.25073	192.168.1.5	120.241.16.104	TCP	[TCP Dup ACK 15887#1] 57743 > 36688 [ACK] Seq=275 Ack=21 Win=65516 Len=0 SLE=1 SRE=21
15890	3487.37655	192.168.1.5	203.119.198.187	HTTP	Continuation or non-HTTP traffic
15891	3487.45678	203.119.198.187	192.168.1.5	TCP	http > 56963 [ACK] Seq=547 Ack=1226 Win=28096 Len=0
15892	3487.54298	203.119.198.187	192.168.1.5	HTTP	Continuation or non-HTTP traffic

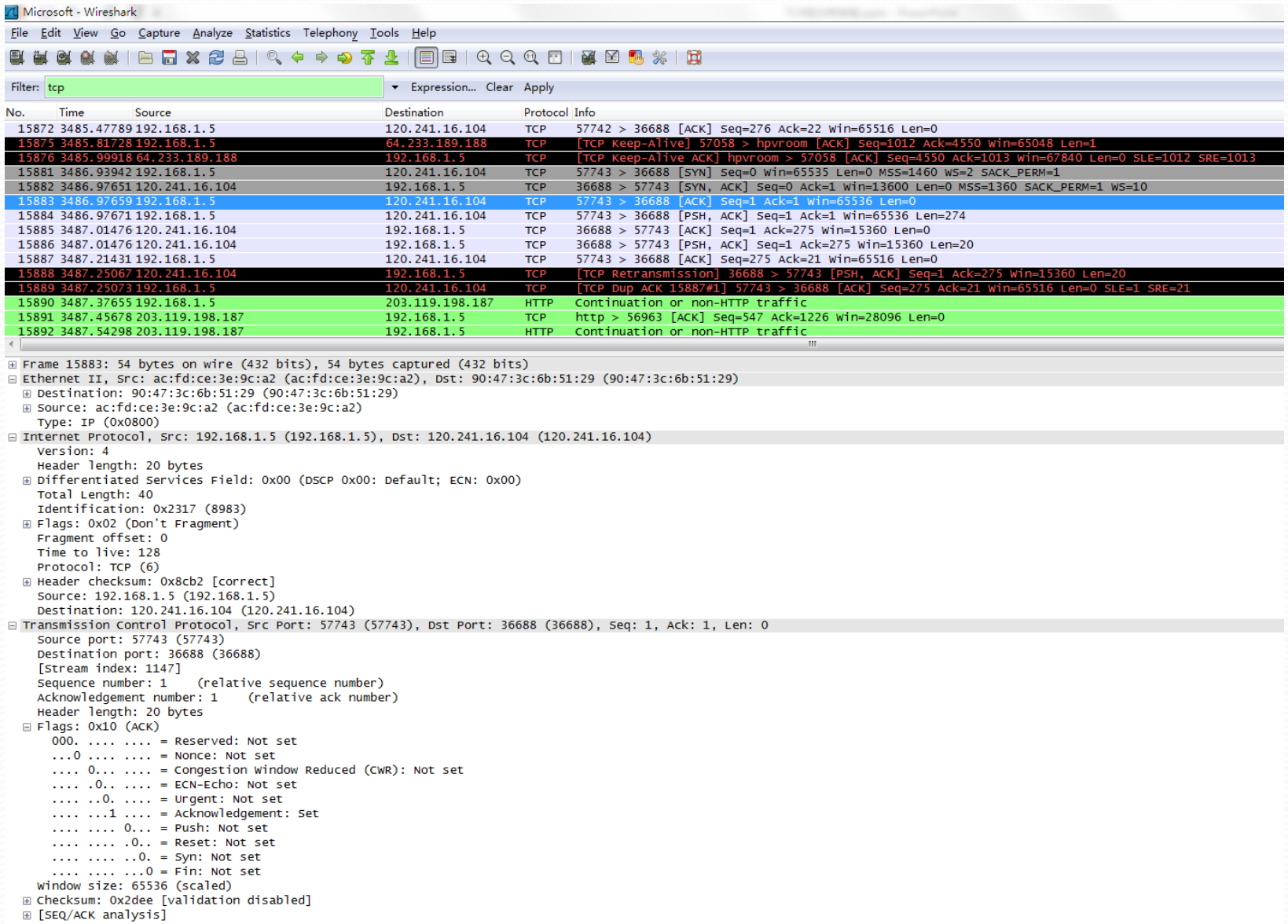
```
⊞ Frame 15881: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
⊟ Ethernet II, Src: ac:fd:ce:3e:9c:a2 (ac:fd:ce:3e:9c:a2), Dst: 90:47:3c:6b:51:29 (90:47:3c:6b:51:29)
  ⊞ Destination: 90:47:3c:6b:51:29 (90:47:3c:6b:51:29)
  ⊞ Source: ac:fd:ce:3e:9c:a2 (ac:fd:ce:3e:9c:a2)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.1.5 (192.168.1.5), Dst: 120.241.16.104 (120.241.16.104)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 52
    Identification: 0x2316 (8982)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊞ Header checksum: 0x8ca7 [correct]
    Source: 192.168.1.5 (192.168.1.5)
    Destination: 120.241.16.104 (120.241.16.104)
⊟ Transmission Control Protocol, Src Port: 57743 (57743), Dst Port: 36688 (36688), Seq: 0, Len: 0
    Source port: 57743 (57743)
    Destination port: 36688 (36688)
    [Stream index: 1147]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊟ Flags: 0x02 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgement: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
    Window size: 65535
  ⊞ Checksum: 0x90b9 [validation disabled]
  ⊞ Options: (12 bytes)
```

School of Software Engineering, Tongji University

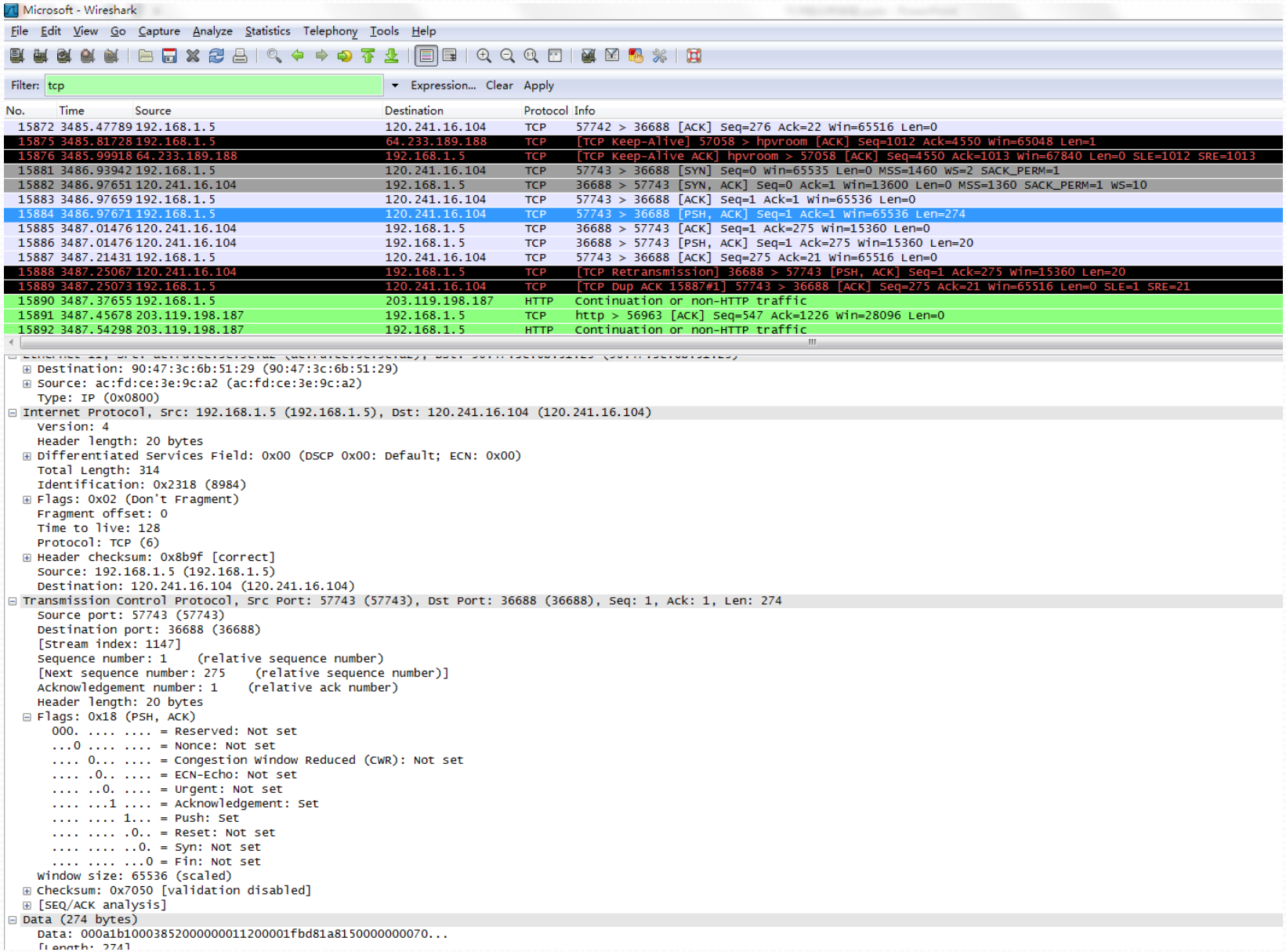
5. WireShark TCP报文抓取分析



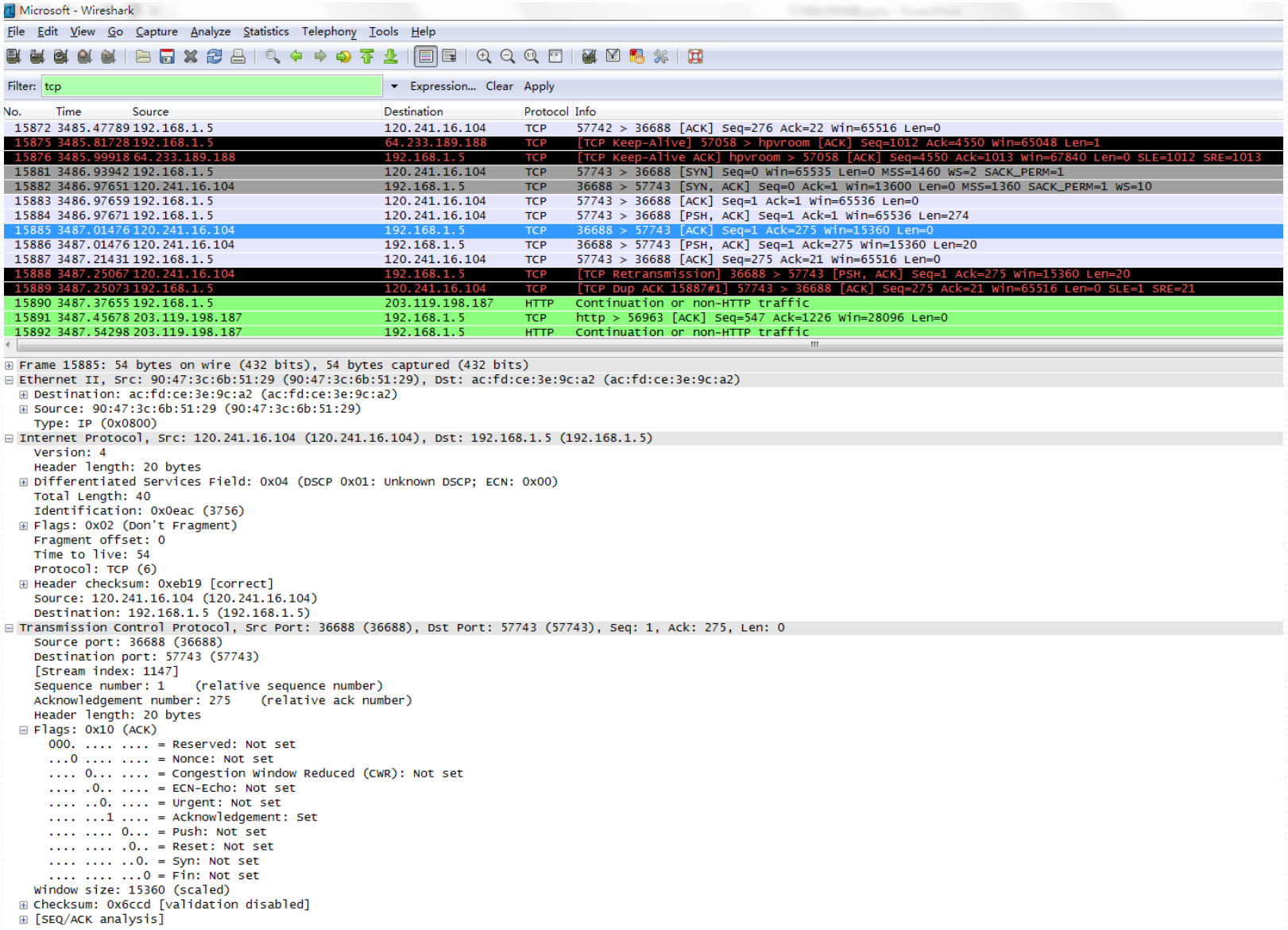
5. WireShark TCP报文抓取分析



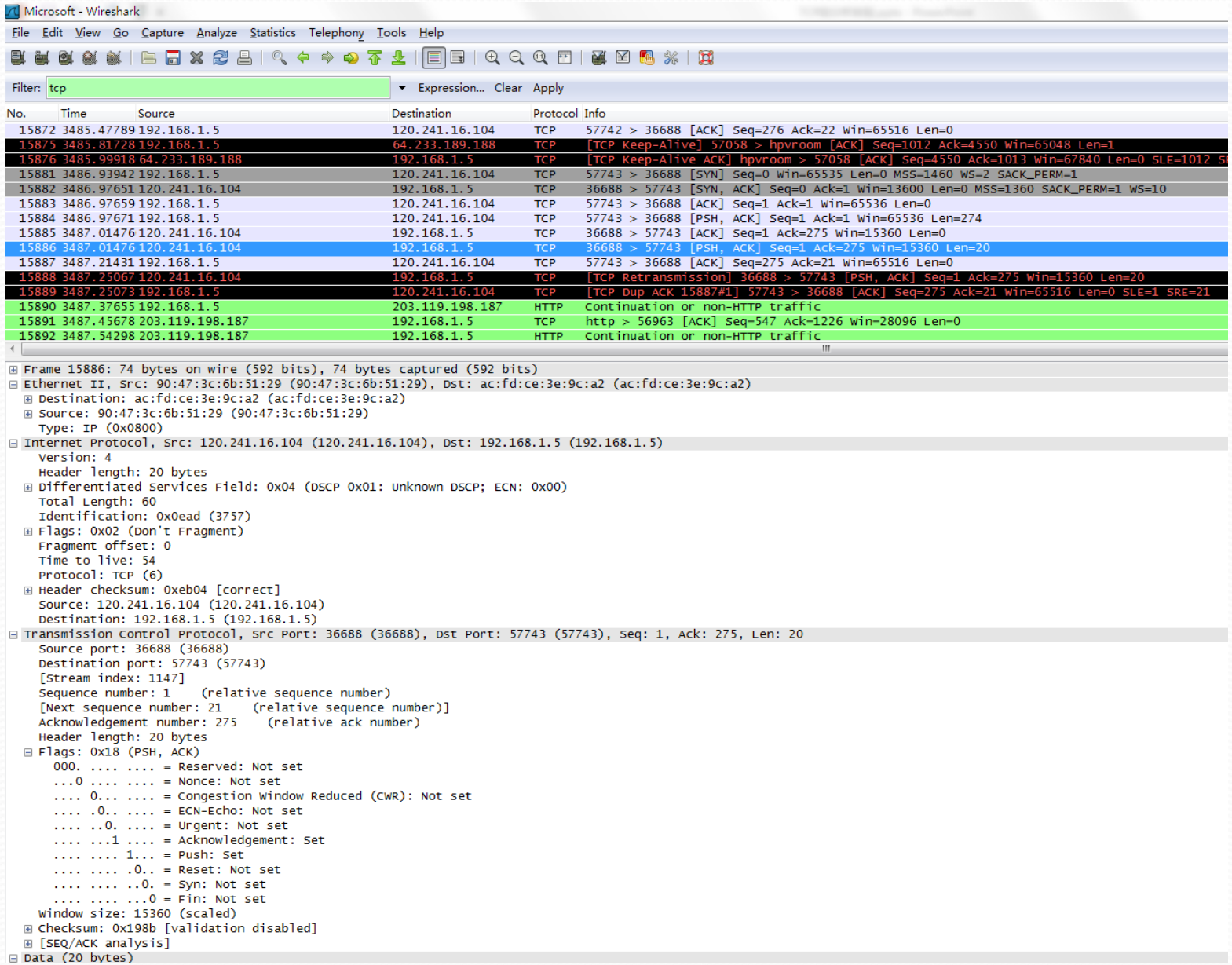
5. WireShark TCP报文抓取分析



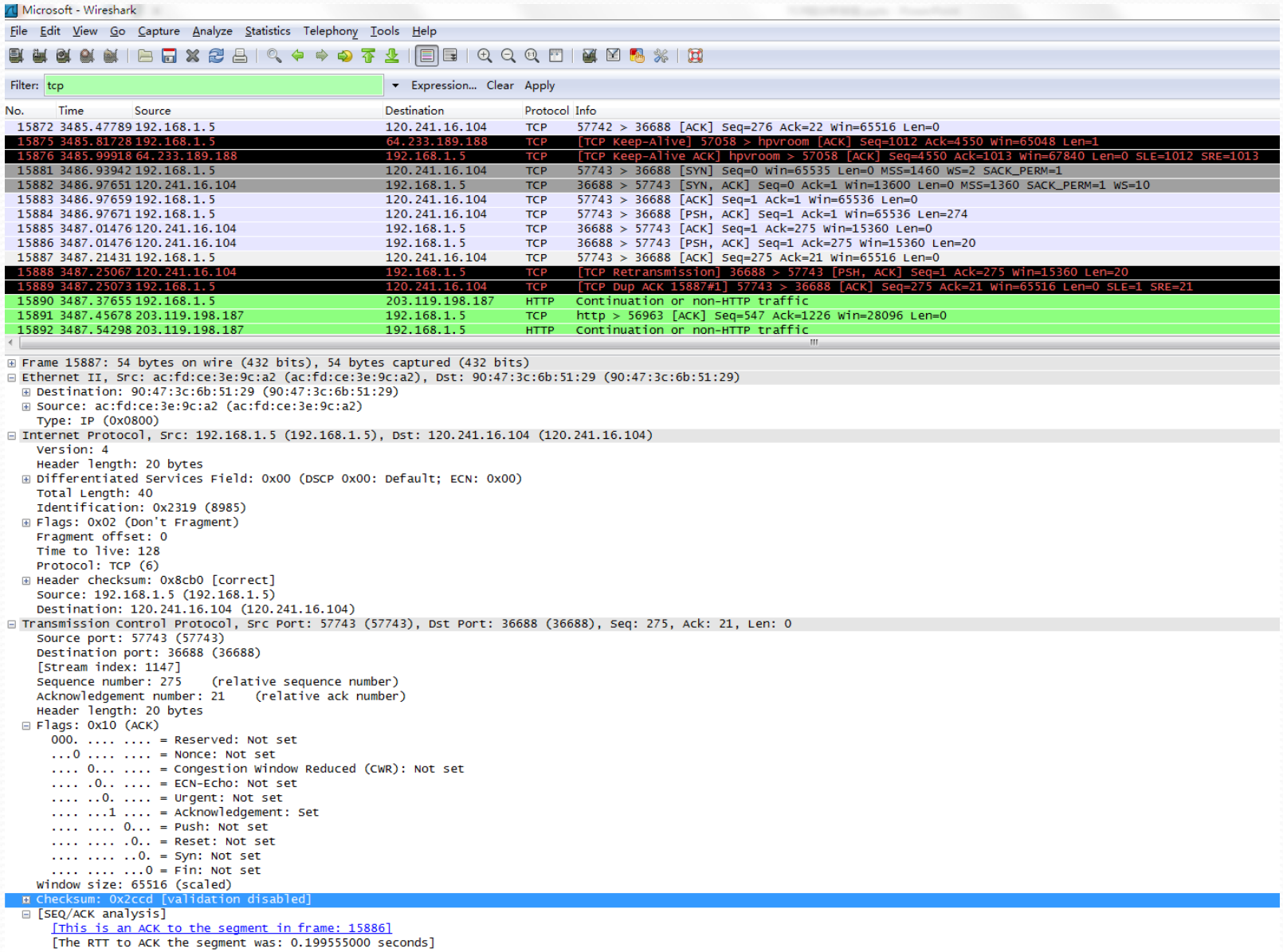
5. WireShark TCP报文抓取分析



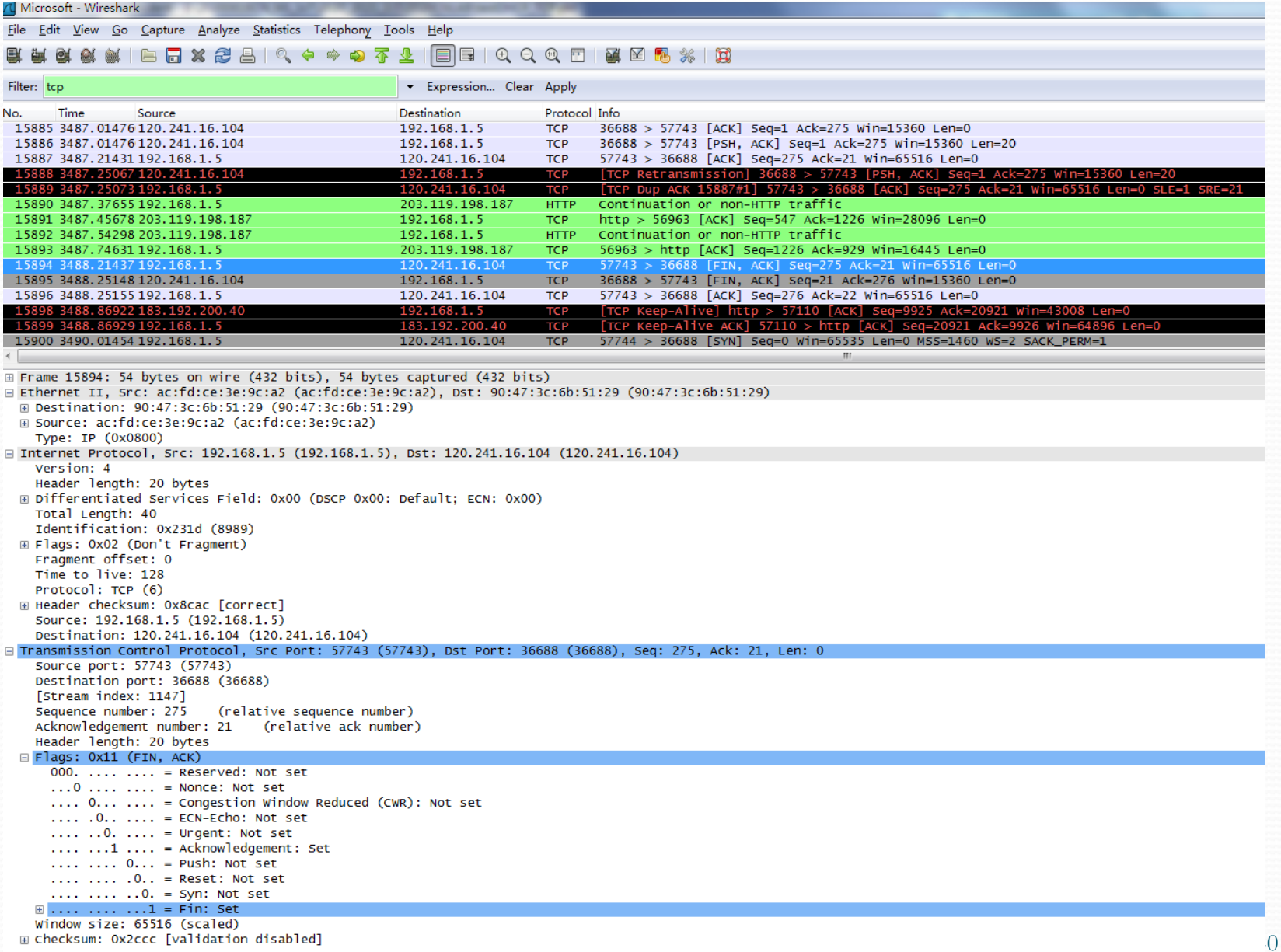
5. WireShark TCP报文抓取分析



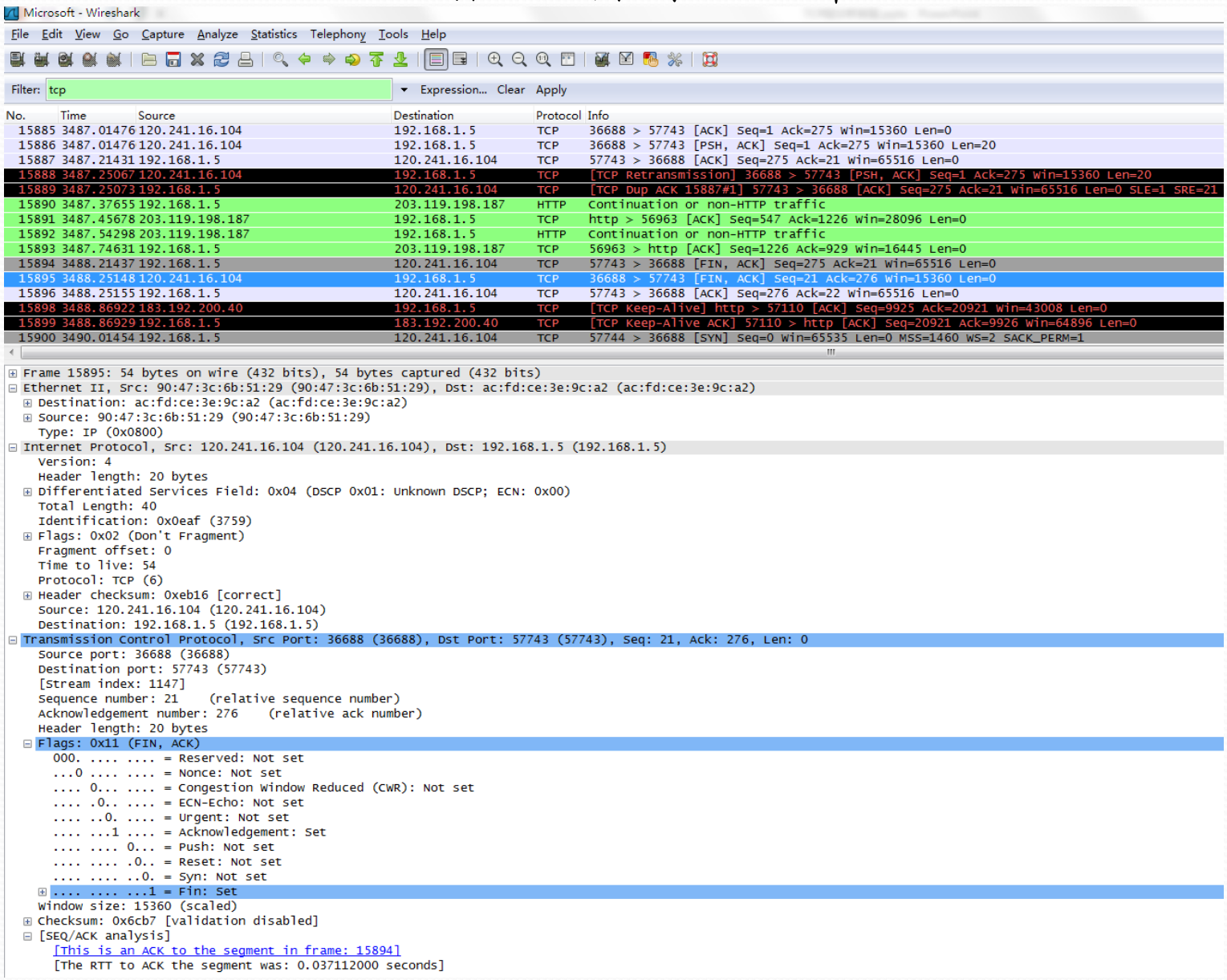
5. WireShark TCP报文抓取分析



5. WireShark TCP报文抓取分析



5. WireShark TCP报文抓取分析



实验主要分析内容

1.分析在Packet tracer中TCP报文情况；
2.用WireShark抓取TCP数据包；
3.查看TCP报文字段内容，并解读；
4.仔细研读TCP连接建立过程数据报文；
5.仔细研读TCP拆链过程数据报文；