**Lab23\_TCP段分析实验项目**

学生姓名：苏家铭 合作学生：无

实验地点：宿舍 实验时间：2023年12月20日

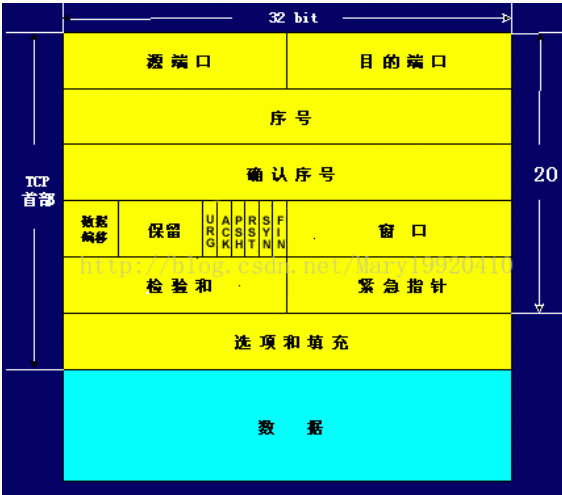
**【实验目的】**

本次实验的实验目的是通过深入研究TCP协议的基本概念、报文结构和连接过程，提供对TCP协议工作原理的深入理解。通过学习TCP的功能和与UDP的区别，了解TCP报文的结构和各个字段的含义，掌握TCP连接的建立和终止过程，以及了解流量控制和校验和等相关内容。这将帮助学生更好地理解TCP协议的底层运作，为网络编程和网络调优等应用提供基础知识和指导。

**【实验原理】**

TCP（Transmission Control Protocol）是传输层的协议，它在IP的数据报服务之上提供了一些基本的服务，包括复用和分用以及差错检测。TCP是一个基于连接的四层协议，它提供了全双工、可靠的传输系统。这意味着TCP能够确保数据被远程主机接收，并且能够为高层协议提供流量控制服务。与此不同，UDP（User Datagram Protocol）不维护连接状态，也不跟踪复杂的参数，因此在空间和时间上具有优势，并且开销较小。TCP在端系统中维护连接状态，这需要一定的开销。连接的状态信息包括接收和发送缓存、拥塞控制参数、序号和确认号等。

TCP报文是TCP层传输的数据单元，也叫报文段。



TCP报文结构

下面是TCP报文中的字段：

A. 端口号：用来标识同一台计算机的不同应用进程。源端口：源端口和IP地址的作用是标识报文的返回地址。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。

B. 序号和确认号：TCP可靠传输的关键部分。序号：是本报文段发送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（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。

C. 数据偏移/首部长度： 4位字段，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，目前一般置为0。

E. 控制位：URG：紧急指针标志，表示紧急指针是否有效。ACK：确认序号标志，表示确认号字段是否有效。PSH：推送标志，指示接收方应尽快交付给应用程序，而不是在缓冲区排队。RST：重置连接标志，用于重置错误的连接或拒绝非法的报文段和连接请求。SYN：同步标志，用于建立连接过程。FIN：结束标志，用于释放连接。

F. 窗口：用于流量控制，指示发送方接收端的缓存大小，从而控制发送端发送数据的速率，从而达到流量控制的目的。窗口大小是一个16位的字段，因而窗口大小最大位65535。

G. 校验和： 对整个TCP报文段（包括报头和数据）进行奇偶校验计算的16位校验和。

H. 紧急指针：仅当URG标志置1时有效，表示紧急数据的最后一个字节的序号。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：可选字段，常见的选项是最大报文段大小（MSS），用于指定本端能接受的最大报文段长度。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。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报文段中的可选数据部分。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。

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”。

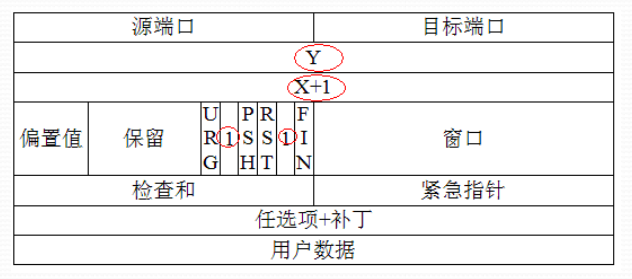
所谓三次握手(Three-way Handshake)，是指建立一个TCP连接时，需要客户端和服务器总共发送3个包。三次握手的目的是连接服务器指定端口，建立TCP连接,并同步连接双方的序列号和确认号并交换 TCP 窗口大小信息.在socket编程中，客户端执行connect()时。将触发三次握手。

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。



第一次握手

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I S N加以1.即X+1。



第二次握手

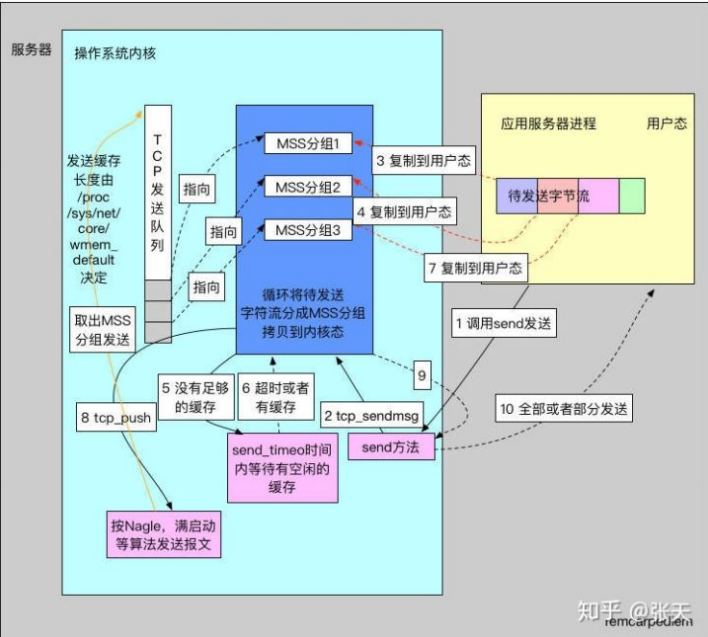
第三次握手：客户端再次发送确认包(ACK) SYN标志位为0,ACK标志位为1.并且把服务器发来ACK的序号字段+1,放在确定字段中发送给对方.并且在数据段放写ISN的+1.



第三次握手

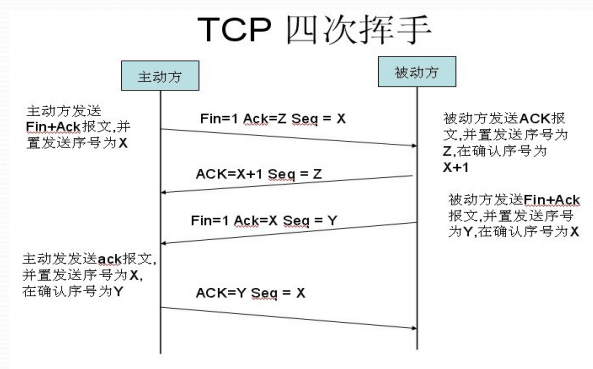
SYN攻击在三次握手过程中，服务器发送SYN-ACK之后，收到客户端的ACK之前的TCP连接称为半连接(half-open connect).此时服务器处于Syn\_RECV状态.当收到ACK后，服务器转入ESTABLISHED状态。攻击客户端在短时间内伪造大量不存在的IP地址，向服务器不断地发送syn包，服务器回复确认包，并等待客户的确认，由于源地址是不存在的，服务器需要不断的重发直至超时，这些伪造的SYN包将长时间占用未连接队列，正常的SYN请求被丢弃，目标系统运行缓慢，严重者引起网络堵塞甚至系统瘫痪。

TCP发送报文，操作系统内核与用户态。

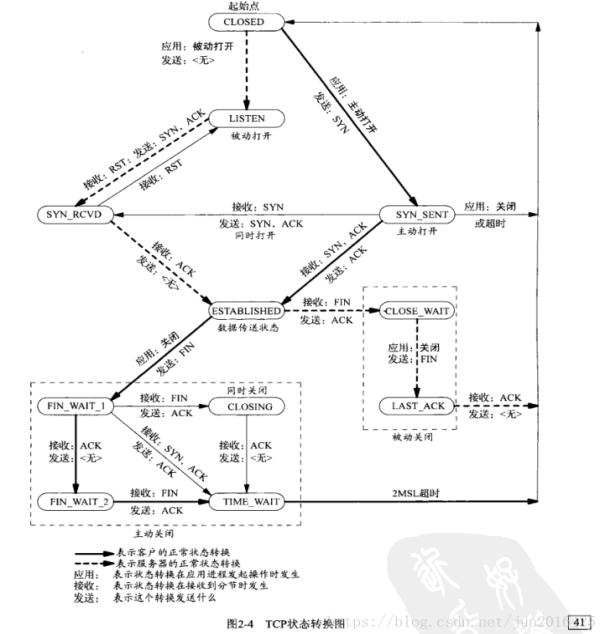


内核与用户态

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。



四次挥手



三次握手状态转换图

**【实验设备】**

PC机一台，并安装有Cisco Packet Tracer软件

**【实验步骤】**

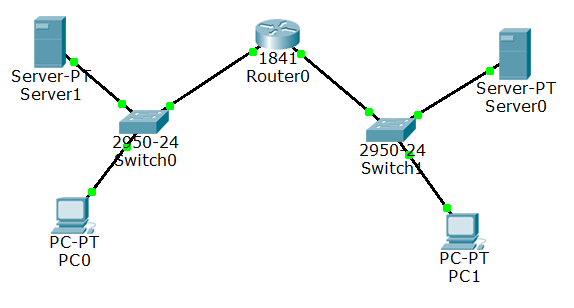
按照PPT提示放置网络设备，并连线

设置仿真模式，设置网络请求

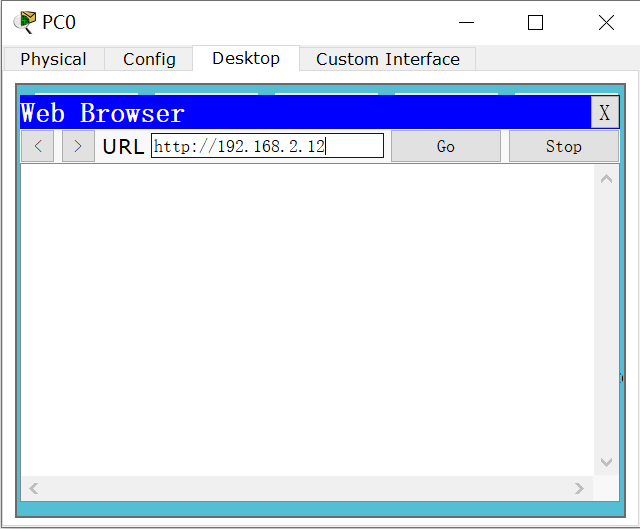
单步调试抓取TCP报文并分析

利用Wireshark抓取TCP报文并分析

**【实验现象】**



网络拓扑图



发出请求

|  |  |
| --- | --- |
|  |  |

PC0发出去

|  |  |
| --- | --- |
|  |  |

server0收到

|  |  |
| --- | --- |
|  |  |

server0发回去

|  |  |
| --- | --- |
|  |  |

PC0收到

|  |  |
| --- | --- |
|  |  |

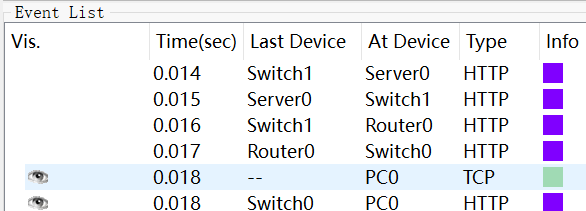
PC0继续发给server0

|  |  |
| --- | --- |
|  |  |

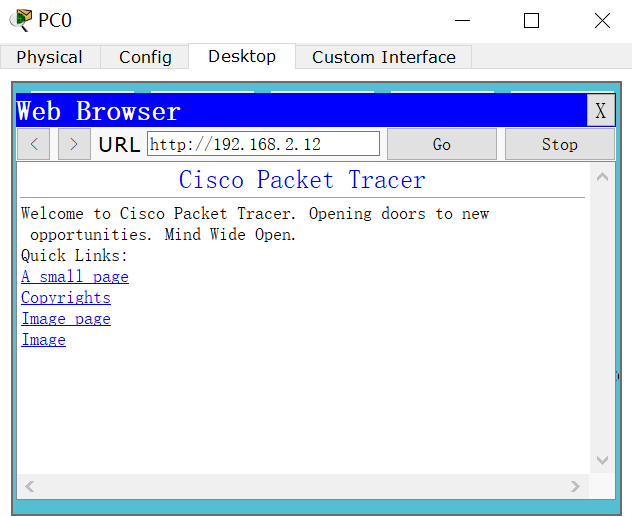
server0收到

|  |  |
| --- | --- |
|  |  |

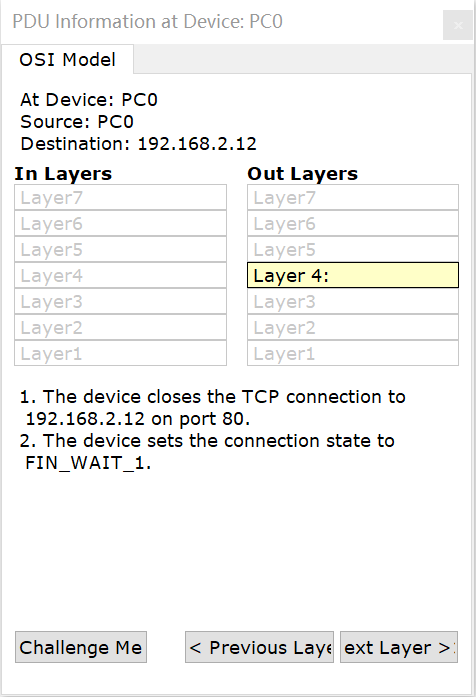
server0发送HTTP给PC0



发送HTTP



PC0收到HTTP



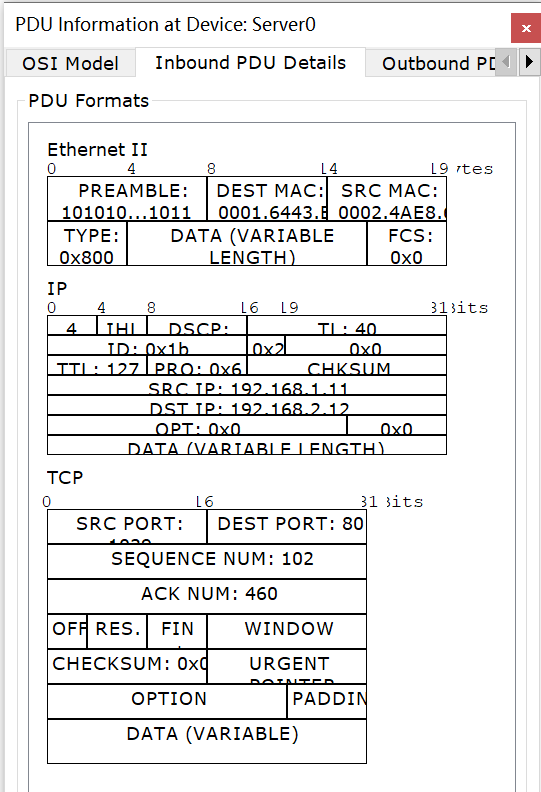
关闭TCP连接

|  |  |
| --- | --- |
|  |  |

server0收到

|  |  |
| --- | --- |
|  |  |

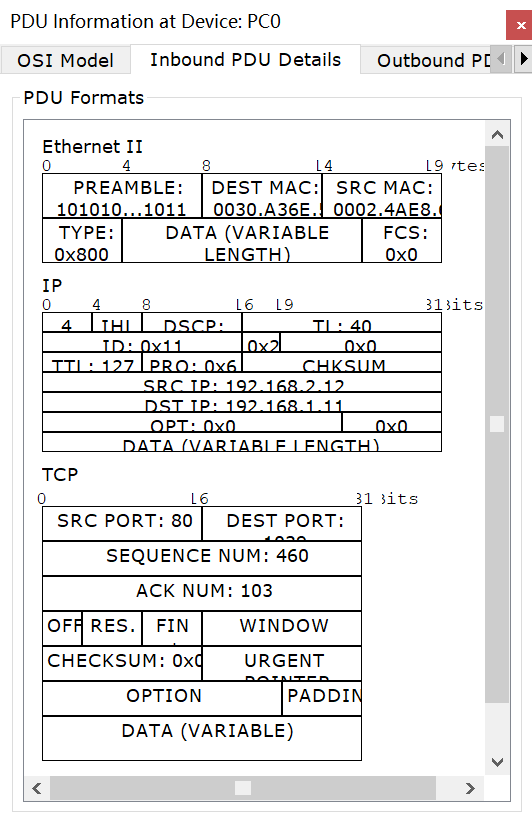
图 21 server发出去



|  |  |
| --- | --- |
|  |  |

PC0收到

|  |  |
| --- | --- |
|  |  |

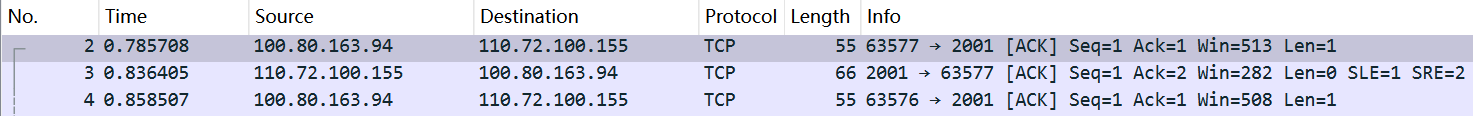


PC0发出去

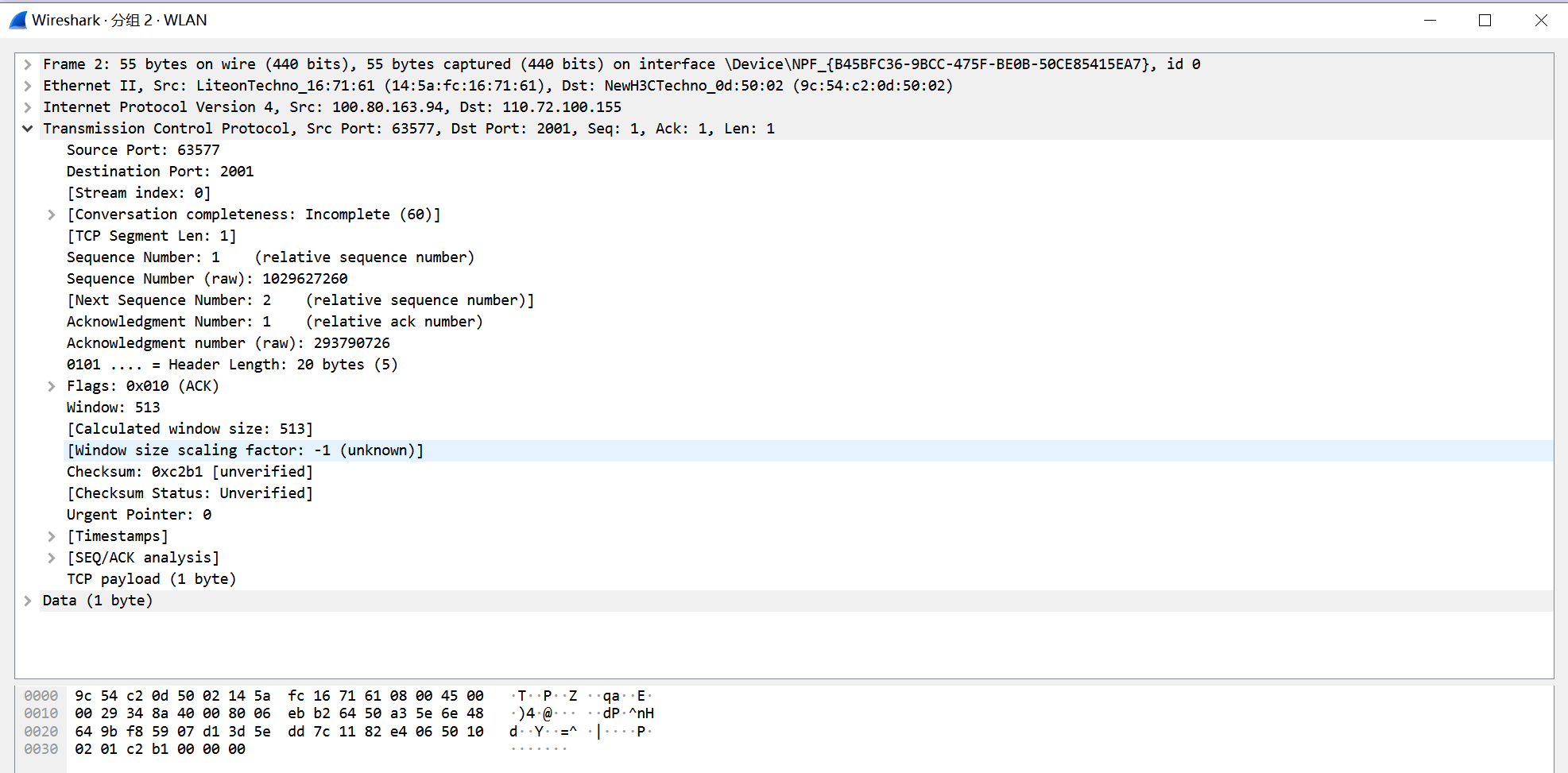
|  |  |
| --- | --- |
|  |  |

server0收到

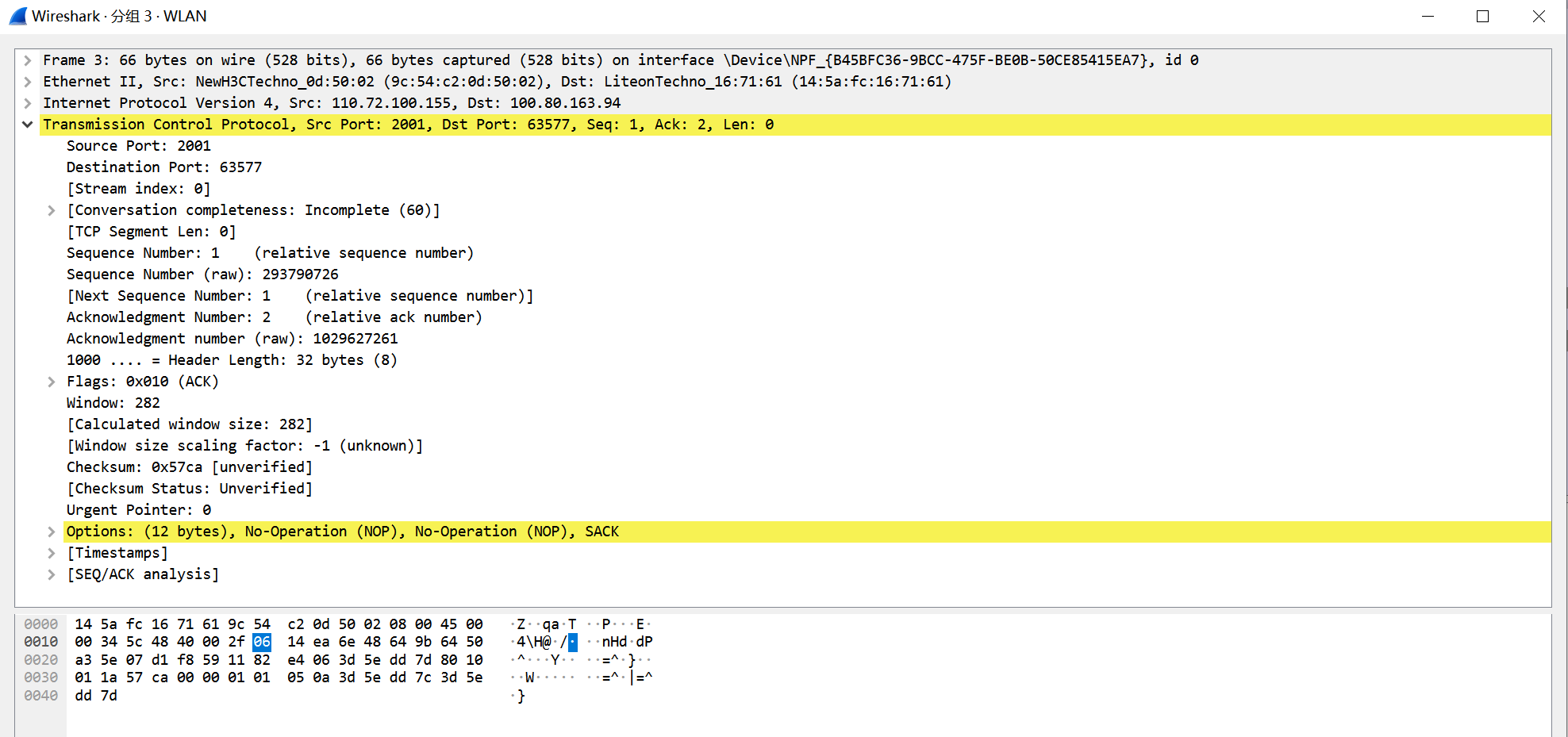
|  |  |
| --- | --- |
|  |  |



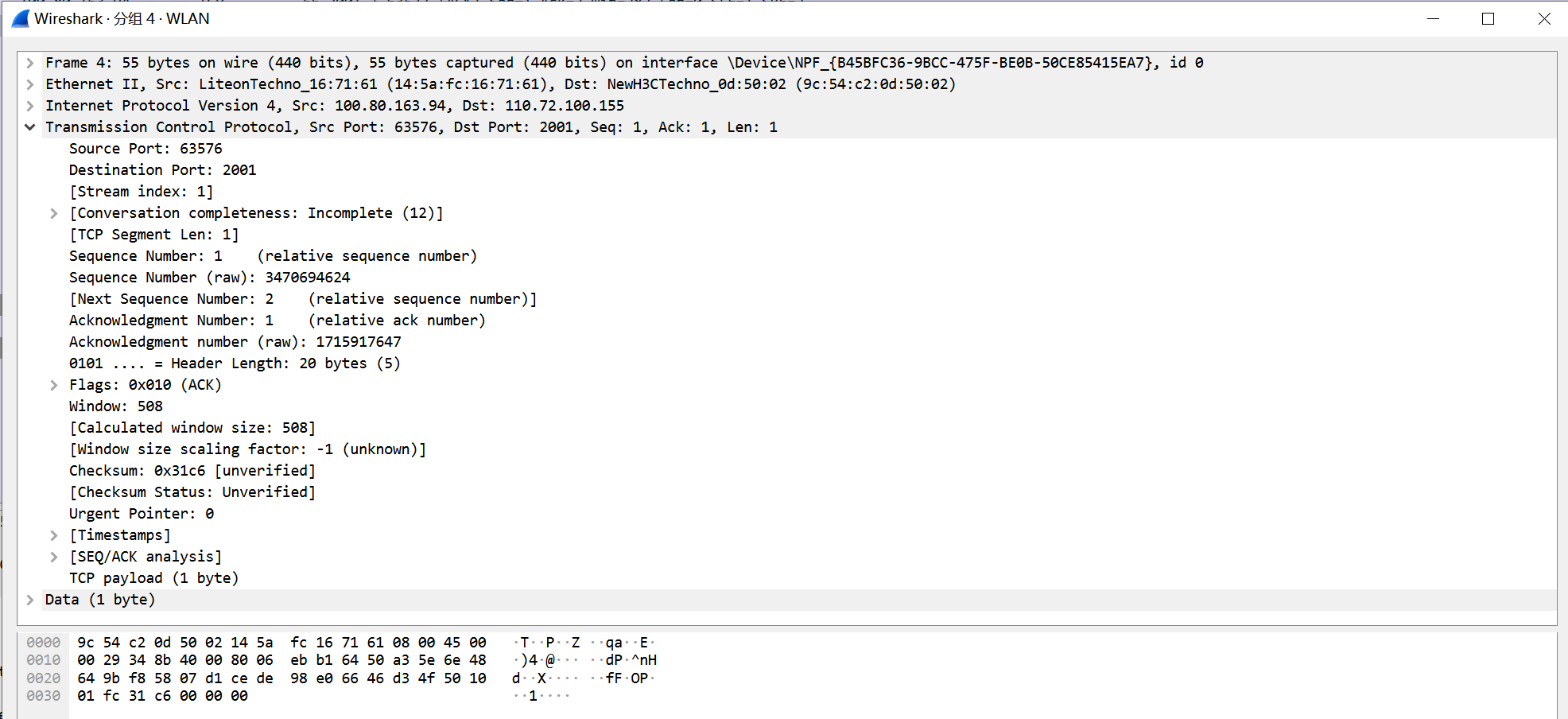
wireShark捕获



1



2



3

**【分析讨论】**

我们以wireshark捕获的报文进行分析

f8 59 07 d1 3d 5e dd 7c 11 82 e4 06 50 10 02 01 c2 b1 00 00

f8 59表示source port为63577

07 d1表示destination port为2001

3d 5e dd 7c表示序号为1029627260

11 82 e4 06表示确认序号为293790726

50 10表示数据偏移、保留和标志位，只有Acknowledgement设置为1

02 01表示窗口，为513

c2 b1表示校验和

00 00表示紧急指针

然后看到发回来的报文

07 d1 f8 59 11 82 e4 06 3d 5e dd 7d 80 10 01 1a 57 ca 00 00 01 01 05 0a 3d 5e dd 7c 3d 5e dd 7d

11 82 e4 06表示序列号为293790726

3d 5e dd 7c表示确认序号为1029627261即为上一个报文的序列号+1

一次类推可以观察到三次握手的情况。

TCP的三次握手和四次挥手是建立和终止TCP连接的过程，对于网络通信的可靠性和稳定性至关重要。在三次握手中，客户端向服务器发送连接请求，服务器确认并回复，最后客户端再次回复确认，以确保双方都同意建立连接。这种机制能够防止已失效的连接请求达到服务器，从而提高了连接的可靠性。

而在四次挥手中，当一方决定终止连接时，它发送一个终止连接请求给对方，对方确认并回复，然后发送自己的终止连接请求，最后另一方再次确认并回复。这个过程保证了双方都完成了数据的传输，避免了数据丢失或不完整的情况。 通过实验和思考，我认识到三次握手和四次挥手的重要性。三次握手确保了双方之间的连接可靠性和一致性，而四次挥手则确保了数据的完整传输和连接的正常终止。理解这些过程不仅在网络编程中有实际应用，也有助于我们更好地理解网络通信的基本原理和机制。因此，深入学习和掌握TCP的三次握手和四次挥手是非常重要的。

【实验名称】：UDP数据包分析实验

学生姓名：杨滕超 合作学生：无

实验地点：济事楼330网络实验室 实验时间：2023年12月4日

【实验目的】

通过使用UDP协议进行数据传输的实验，了解UDP协议的特点和应用场景。通过实验，可以掌握UDP协议的无连接、不可靠传输特性，理解UDP在少量数据传输和实时应用方面的优势。同时，实验还旨在熟悉UDP的报文格式和校验方法，理解UDP在网络通信中的作用和限制，为后续网络编程和应用开发提供基础和参考。

【实验原理】

UDP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。 UDP提供不可靠服务，具有TCP所没有的优 势： UDP无连接，时间上不存在建立连接需要的时 延。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。UCP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。 UDP数据报(用户数据报协议)

DNS如果运行在TCP之上而不是UDP，那么 DNS的速度将会慢很多。HTTP使用TCP而不是 UDP，是因为对于基于文本数据的Web网页来 说，可靠性很重要。同一种专用应用服务器在 支持UDP时，一定能支持更多的活动客户机。 分组首部开销小，TCP首部20字节，UDP首 部8字节。

UDP没有拥塞控制，应用层能够更好的控制 要发送的数据和发送时间，网络中的拥塞控制 也不会影响主机的发送速率。某些实时应用要 求以稳定的速度发送，能容 忍一些数据的丢失， 但是不能允许有较大的时延（比如实时视频， 直播等） UDP提供尽最大努力的交付，不保证可靠交 付。所有维护传输可靠性的工作需要用户在应 用层来完成。没有TCP的确认机制、重传机制。 如果因为网络原因没有传送到对端，UDP也不 会给应用层返回错误信息。

UDP是面向报文的，对应用层交下来的报文， 添加首部后直接乡下交付为IP层，既不合并， 也不拆分，保留这些报文的边界。对IP层交上 来UDP用户数据报，在去除首部后就原封不动 地交付给上层应用进程，报文不可分割，是 UDP数据报处理的最小单位。正是如此UDP显 得不够灵活，不能控制读写数据的次数和数量。 比如我们要发送100个字节的报文，调用一次 sendto函数就会发送100字节，对端也需要用 recvfrom函数一次性接收100字节，不能使用循 环每次获取10个字节，获取十次这样的做法。

UDP常用一次性传输比较少量数据的网络应 用，如DNS,SNMP等，因为对于这些应用，若 是采用TCP，为连接的创建，维护和拆除带来 不小的开销。UDP也常用于多媒体应用（如IP 电话，实时视频会议，流媒体等）数据的可靠 传输对他们而言并不重要，TCP的拥塞控制会 使它们有较大的延迟，也是不可容忍的。 总之，UDP协议提供不可靠无连接的数据 报传输服务。

UDP的首部格式 UDP数据报分为首部和用户数据部分，整个 UDP数据报作为IP数据报的数据部分封装在IP数 据报中，UDP数据报文结构如图所示：



图 29 UDP数据报文结构

UDP首部有8个字节，由4个字段构成，每个字 段都是两个字节，1).源端口： 源端口号，需要 对方回信时选用，不需要时全部置0. 2).目的端口：目的端口号，在终点交付报文的 时候需要用到。 3).长度：UDP的数据报的长度（包括首部和数 据）其最小值为8（只有首部）

校验和：检测UDP数据报在传输中是否有错， 有错则丢弃。该字段是可选的，当源主机不想 计算校验和，则直接令该字段全为0. 当传输层从IP层收到UDP数据报时，就根据首 部中的目的端口，把UDP数据报通过相应的端 口，上交给应用进程。 如果接收方UDP发现收到的报文中的目的端口 号不正确（不存在对应端口号的应用进程0,）， 就丢弃该报文，并由ICMP发送“端口不可达” 差错报文给对方。

在计算校验和的时候，需要在UDP数据报之前 增加12字节的伪首部，伪首部并不是UDP真正 的首部。只是在计算校验和，临时添加在UDP 数据报的前面，得到一个临时的UDP数据报。 校验和就是按照这个临时的UDP数据报计算的。 伪首部既不向下传送也不向上递交，而仅仅是 为了计算校验和。这样的校验和，既检查了 UDP数据报，又对IP数据报的源IP地址和目的IP 地址进行了检验。

UDP校验和的计算方法和IP数据报首部校验和 的计算方法相似，都使用二进制反码运算求和 再取反，但不同的是：IP数据报的校验和只检 验IP数据报的首部，但UDP的校验和是把首部 和数据部分一起校验。

发送方，首先是把全零放入校验和字段并且添 加伪首部，然后把UDP数据报看成是由许多16 位的子串连接起来，若UDP数据报的数据部分 不是偶数个字节，则要在数据部分末尾增加一 个全零字节（此字节不发送），接下来就按照 二进制反码计算出这些16位字的和。将此和的 二进制反码写入校验和字段。在接收方，把收 到得UDP数据报加上伪首部（如果不为偶数个 字节，还需要补上全零字节）后，按二进制反 码计算出这些16位字的和。

当无差错时其结果全为1,。否则就表明有差错出 现，接收方应该丢弃这个UDP数据报。注意：1). 校验时，若UDP数据报部分的长度不是偶数个字 节，则需要填入一个全0字节，但是此字节和伪 首部一样，是不发送的。2).如果UDP校验和校验 出UDP数据报是错误的，可以丢弃，也可以交付 上层，但是要附上错误报告，告诉上层这是错误 的数据报。3).通过伪首部，不仅可以检查源端口 号，目的端口号和UDP用户数据报的数据部分， 还可以检查IP数据报的源IP地址和目的地址。这 种差错检验的检错能力不强，但是简单，速度快

【实验设备】

PC机一台，并安装有Cisco Packet Tracer软件

【实验步骤】

按照PPT提示放置网络设备，并连线

配置DNS服务器

设置仿真模式，设置网络请求

单步调试抓取UDP报文并分析

利用Wireshark抓取UDP报文并分析

【实验现象】

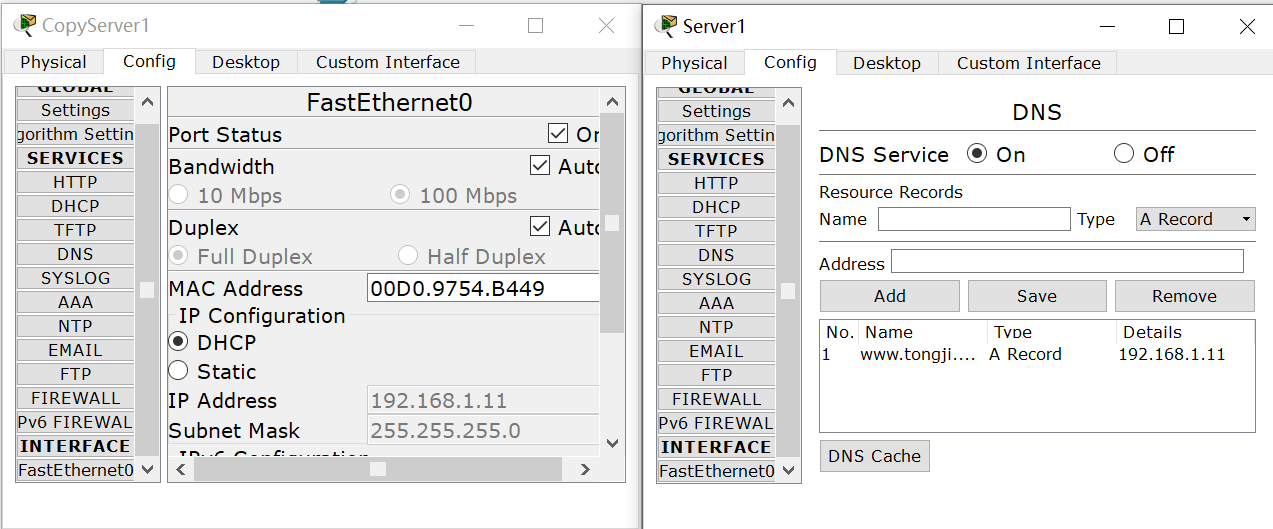


图 30 配置DNS

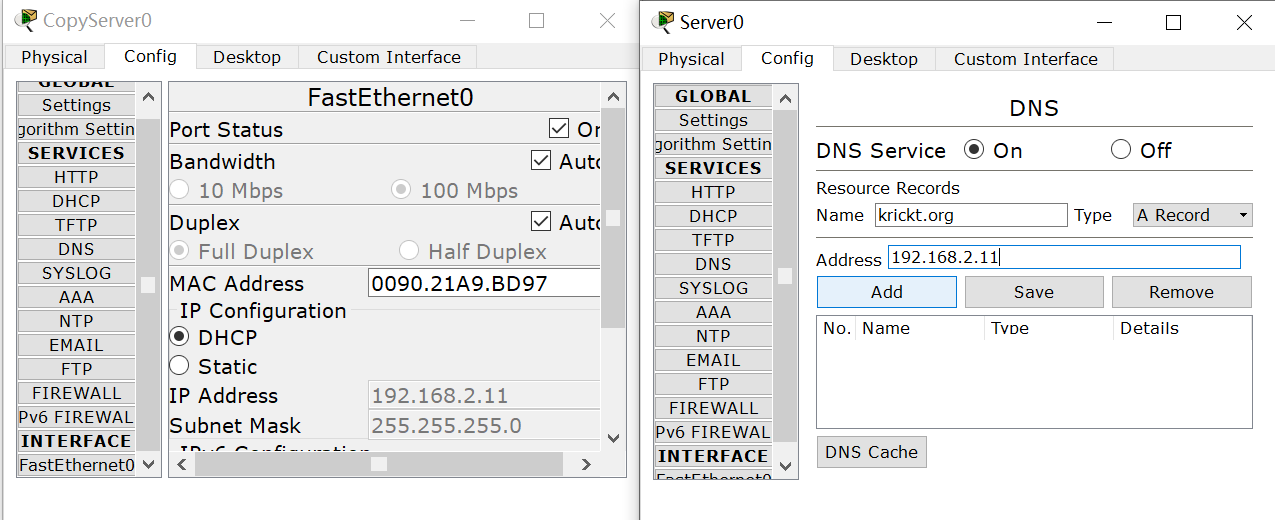


图 31 配置DNS

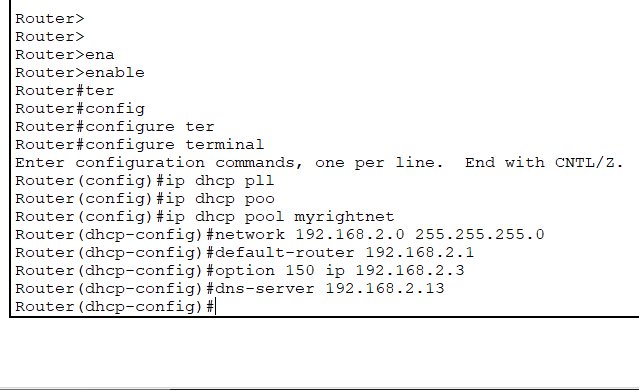


图 32 配置路由器DNS

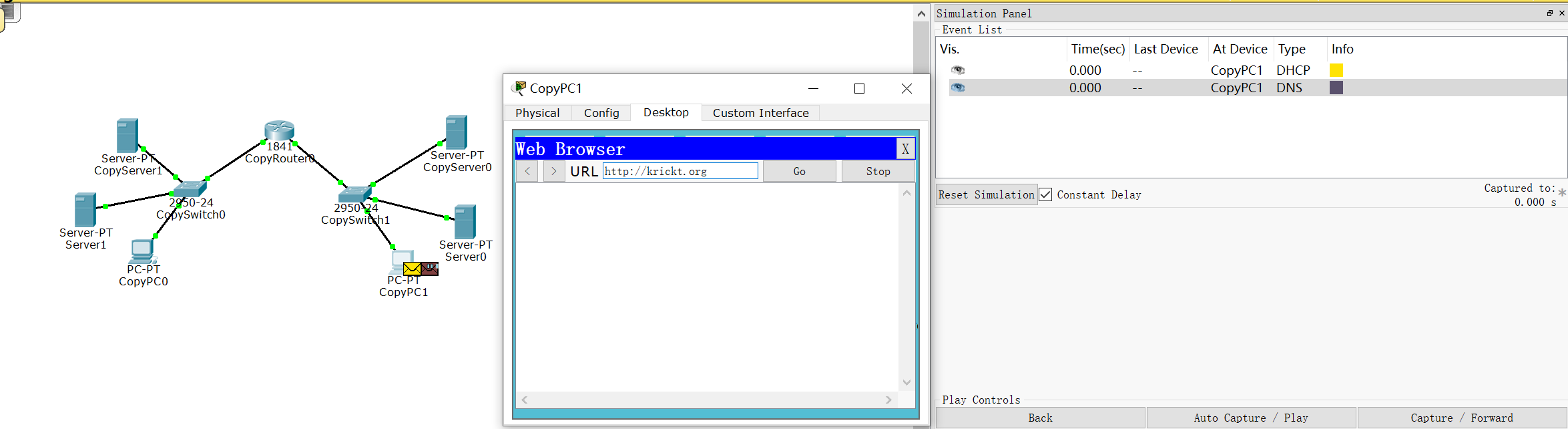


图 33 网络拓扑图与CopyPC1发出去

|  |  |
| --- | --- |
|  |  |

图 34 UPD报文结构

|  |  |
| --- | --- |
|  |  |

图 35 server0发出去

|  |  |
| --- | --- |
|  |  |

图 36 CopyPC1收到

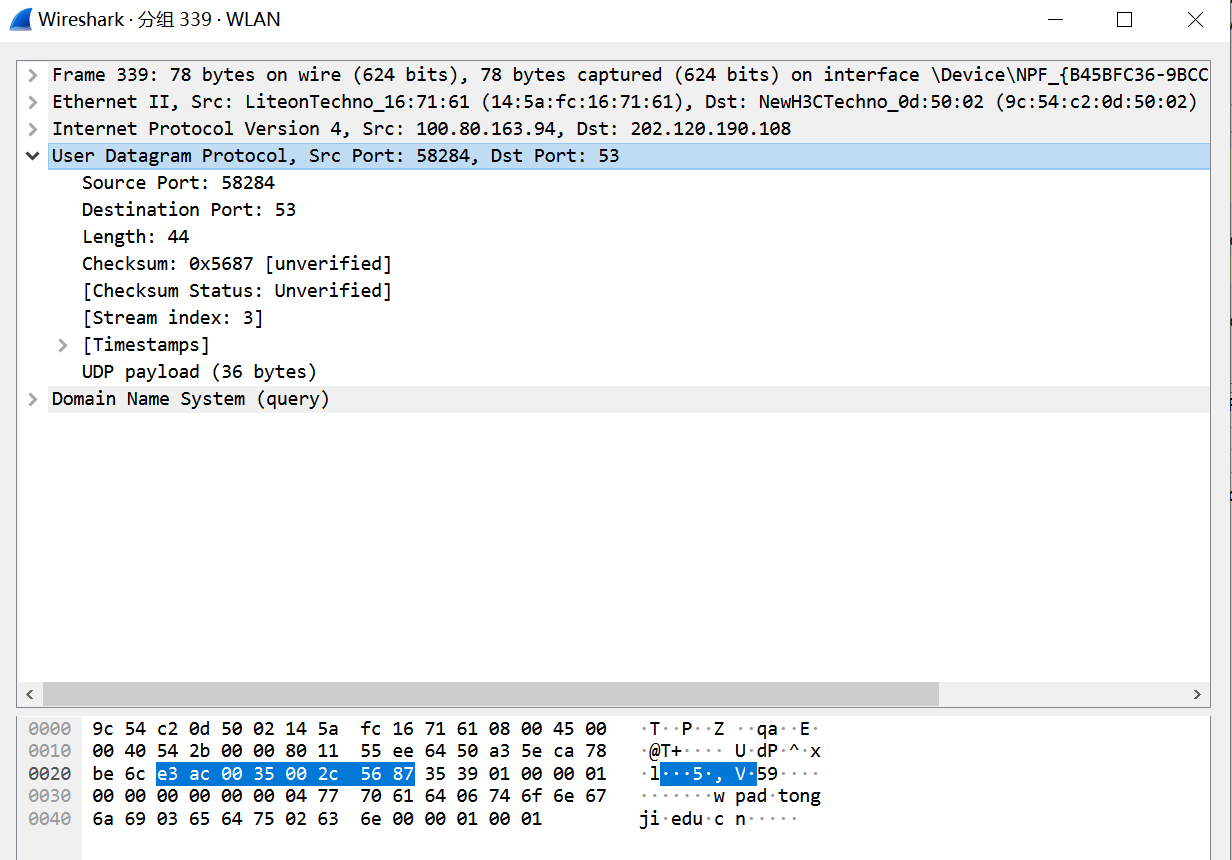


图 37 wireshark捕捉UDP数据报文

【分析讨论】

这里分析wireshark捕捉的数据报文

e3 ac 00 35 00 2c 56 87

e3 ac表示src port为58284

00 35表示destination port为53

00 2c表示长度为44

56 87表示检验和

通过比较发现，UDP的头部比TCP少很多东西。者决定了UDP牺牲了可靠性，节省了额外的开销，提供了更高的性能和效率。

无连接性：UDP是一种无连接协议，不需要在数据传输之前进行建立连接的握手过程，从而减少了延迟和开销。相比之下，TCP需要建立连接、维护状态和执行可靠的数据传输，但提供了可靠的、有序的数据传输。

无可靠性：UDP报文不提供可靠性保证，它不保证数据传输的完整性、顺序性和可靠性。UDP报文没有确认机制，一旦发送出去就无法得知是否被正确接收。而TCP通过确认机制、重传机制和流量控制来确保数据的可靠传输。

简单轻量：UDP报文相对简单，它只包含了必要的头部信息，没有复杂的控制字段和拥塞控制机制。这使得UDP报文的开销小，传输效率高，适合在性能敏感的实时应用中使用。

4. 适用场景：UDP由于其低延迟和高效性，适合用于实时通信、音频/视频传输、游戏、DNS查询等应用。在这些场景下，速度和实时性更加重要，而数据可靠性可以通过应用层的机制来处理。