**实验五 传输层协议（UDP/TCP）**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机科学与技术学院 | | 年级专业 | | 21计科 | | 姓名 | 陈晨 | 学号 | 2127405030 |
| 课程名称 | | 计算机网络实验 | | | | | | | 成绩 |  |
| 指导教师 | | 钱龙华 | | 同组实验者 | | 王睿语、檀佳玟、张歆、孙启兰、黄启鹏 | | 实验日期 | 2023.12.16 | |

**实验目的**

1. 掌握UDP协议的报文格式

2. 掌握UDP协议校验和的计算方法

3. 理解UDP协议的优缺点

4. 理解协议栈对UDP协议的处理方法

5. 理解UDP上层接口应满足的条件

6. 掌握TCP协议的报文格式

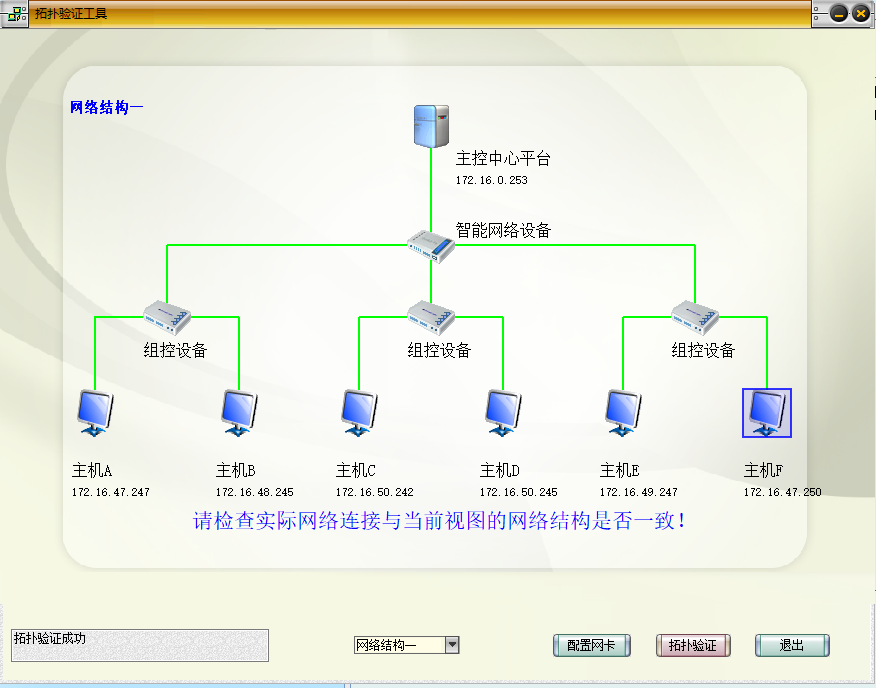
7. 掌握TCP连接的建立和释放过程

8. 掌握TCP数据传输中编号与确认的过程

**实验步骤**

**练习1：编辑并发送UDP数据报**

各主机打开协议分析器，进入相应的网络结构并验证网络拓扑的正确性，如果通过拓扑验证，关闭协议分析器继续进行实验，如果没有通过拓扑验证，请检查网络连接。本练习将主机A和B作为一组，主机C和D作为一组，主机E和F作为一组。现仅以主机A、B所在组为例，其它组的操作参考主机A、B所在组的操作。



1. 主机A打开协议编辑器，编辑发送给主机B的UDP数据报。

MAC层：

目的MAC地址：接收方MAC地址

源MAC地址：发送方MAC地址

协议类型或数据长度：0800，即IP协议

IP层：

总长度：包括IP层、UDP层和数据长度

高层协议类型：17，即UDP协议

首部校验和：其它所有字段填充完毕后填充此字段

源IP地址：发送方IP地址

目的IP地址：接收方IP地址

UDP层：

源端口：1030

目的端口：大于1024的端口号

有效负载长度：UDP层及其上层协议长度

其它字段默认，计算校验和。

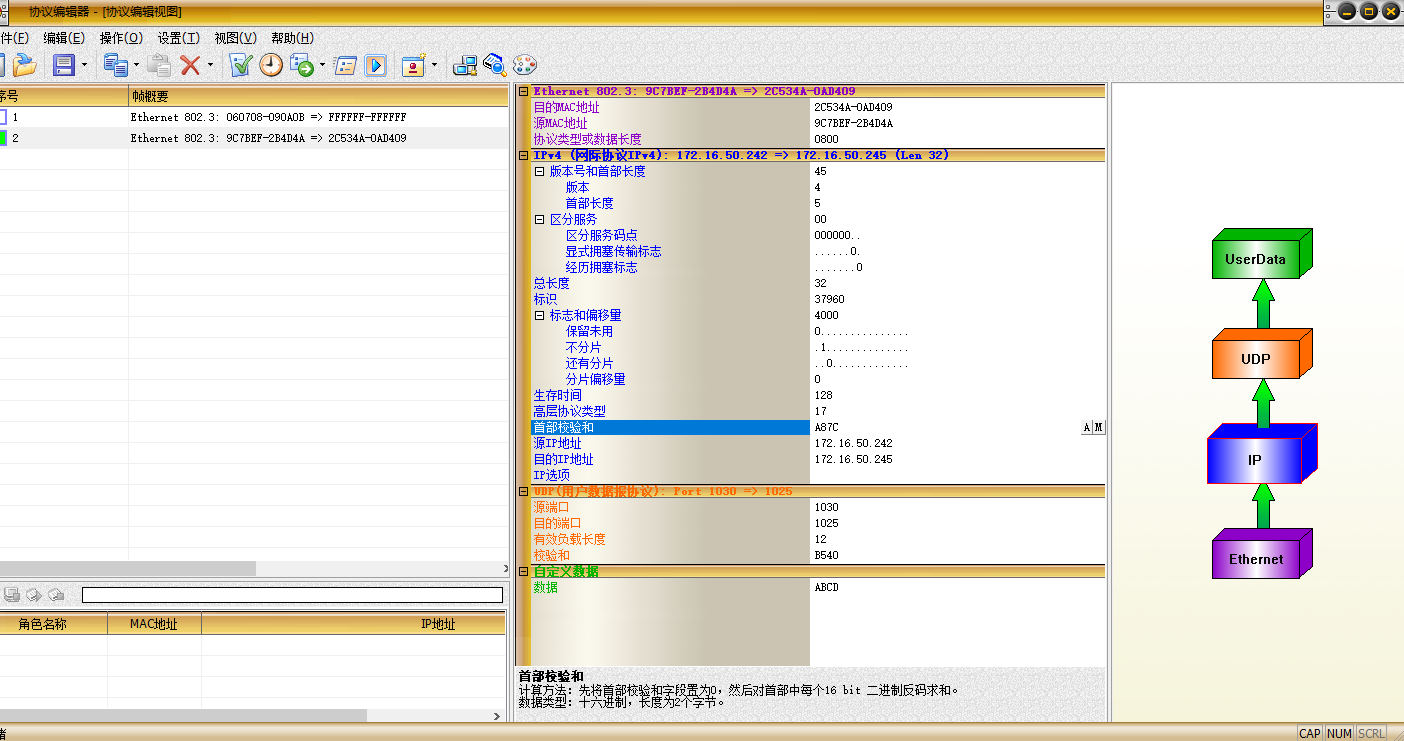
● UDP在计算校验和时包括哪些内容？

**答：伪首部、UDP首部和UDP数据**

1. 在主机B上启动协议分析器捕获数据，并设置过滤条件（提取UDP协议）。

3. 主机A发送已编辑好的数据报。

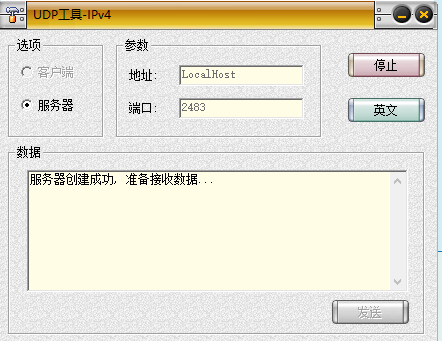
4. 主机B停止捕获数据，在捕获到的数据中查找主机A所发送的数据报。



**练习2：UDP单播通信**

本练习将主机A、B、C、D、E、F作为一组进行实验。

1. 主机B、C、D、E、F上启动“实验平台工具栏中的UDP工具”，作为服务器端，监听端口设置为2483，“创建”成功。



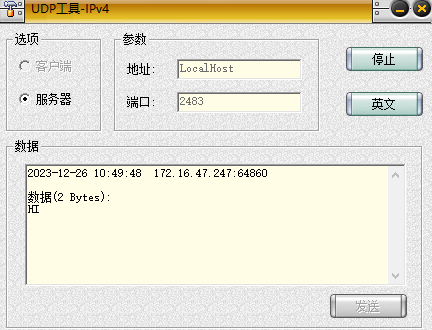
1. 主机C、E上启动协议分析器开始捕获数据，并设置过滤条件（提取UDP协议）。



3. 主机A上启动“实验平台工具栏中的UDP工具”，作为客户端，以主机C的IP为目的IP地址，以2483为端口，填写数据并发送。

4. 查看主机B、C、D、E、F上的“UDP工具”接收的信息。

● 哪台主机上的“UDP工具”能够接收到主机A发送的UDP报文？

****

**答：主机C**

5. 查看主机C协议分析器上的UDP报文，并回答以下问题：

● UDP是基于连接的协议吗？阐述此特性的优缺点。

**答：不是。优点：相对于面向连接的服务，UDP传送数据较快速，系统开销也少。**

**缺点：但它不能防止报文的丢失、重复和乱序。由于它的每个报文必须包括完整的源 地址的目的地址，因此开销很大每个请求必须足够小，使其能够装入到用户数据报中。**

● UDP报文交互中含有确认报文吗？阐述此特性的优缺点。

**答：没有。**

**UDP报文交互中没有确认报文的特性，既有优点也有缺点。**

**优点：**

**效率高：没有确认报文意味着UDP数据报文的发送不需要等待接收方的回应，因此具 有更高的传输效率，适用于对实时性要求较高的应用场景，如流媒体、在线游戏等。**

**简单易用：UDP协议相对简单，实现起来较为容易，对于一些简单的传输需求，使用 UDP可以降低开发成本。**

**缺点：**

**不可靠：由于没有确认报文，UDP数据报文可能会丢失、重复或者乱序到达，因此UDP 协议在传输过程中不能保证数据的可靠性和顺序性。**

**不适合大数据传输：由于UDP协议没有确认机制，如果传输大量数据，可能会出现数 据丢失的情况，因此不适合用于大数据的传输。**

6. 主机A上使用协议编辑器向主机E发送UDP报文，其中：

目的MAC地址：E的MAC地址

目的IP地址：主机E的IP地址

目的端口：2483

校验和：0

发送此报文，并回答以下问题：

● 主机E上的UDP通信程序是否接收到此数据包？UDP是否可以使用0作为校验和进行通信？

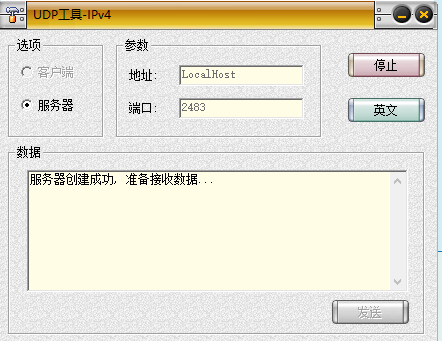
**答：主机E可以收到数据包。UDP可以使用0作为校验和进行通信。**

1. 主机B、C、D、E、F关闭服务端，主机A关闭客户端。

**练习3：UDP广播通信**

本练习将主机A、B、C、D、E、F作为一组进行实验。

1. 主机B、C、D、E、F上启动“实验平台工具栏中的UDP工具”，作为服务器端，监听端口设为2483。

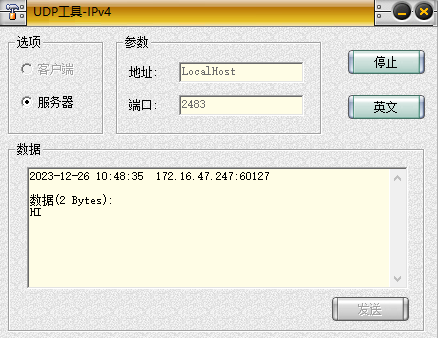


1. 主机B、C、D、E、F启动协议分析器捕获数据，并设置过滤条件（提取UDP协议）。



3. 主机A上启动“实验平台工具栏中的UDP工具”，作为客户端，以255.255.255.255为目的地址，以2483为端口，填写数据并发送。

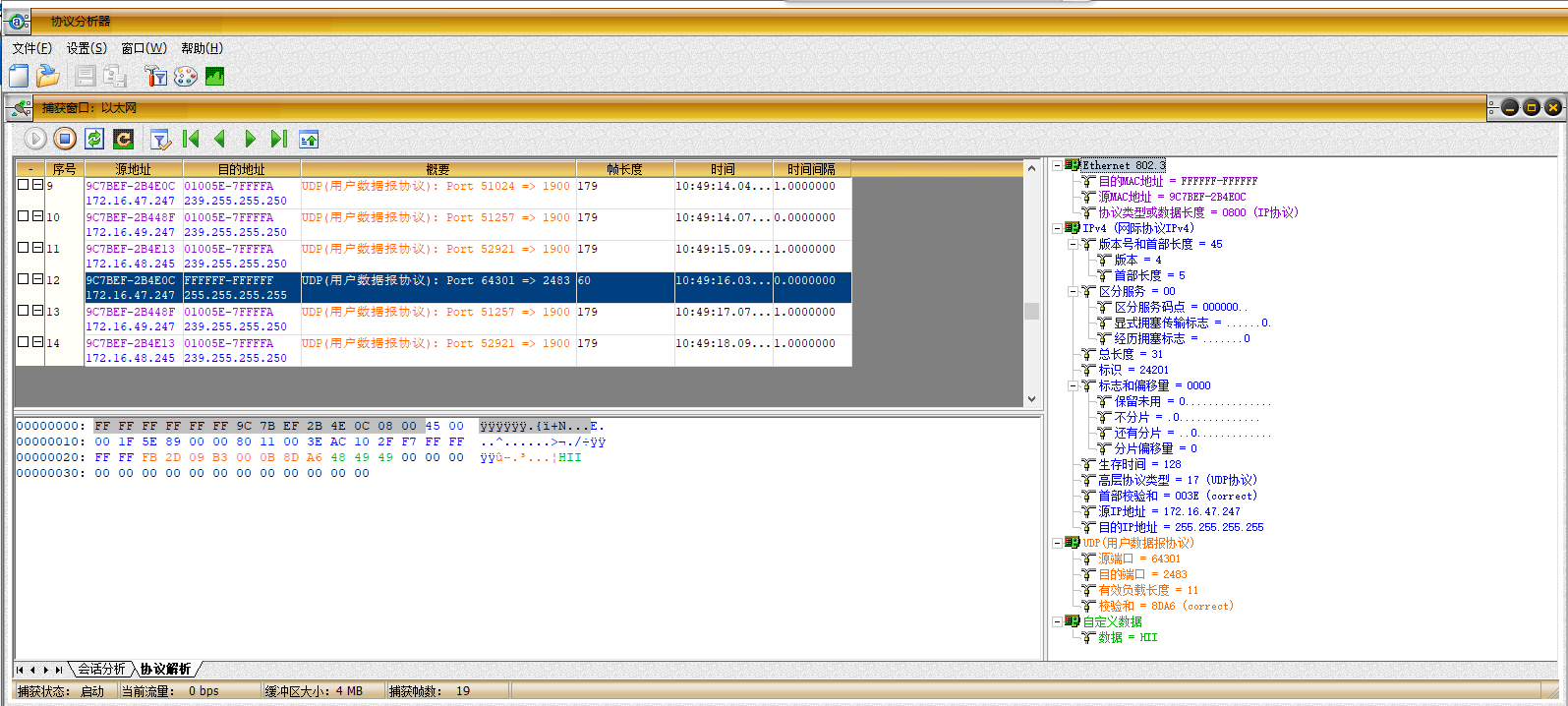
4. 查看主机B、C、D、E、F上的“UDP工具”接收的信息。



● 哪台主机能够接收到主机A发送的UDP报文？

**答：主机BCDEF**

1. 查看协议分析器上捕获的UDP报文，并回答以下问题：



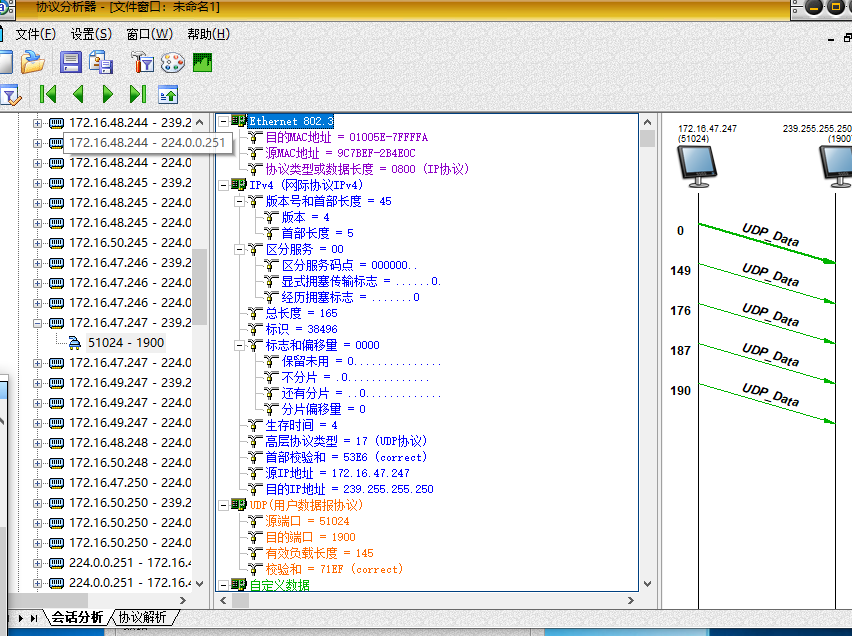
● 主机A发送的报文的目的MAC地址和目的IP地址的含义是什么？

**答：MAC地址是广播的物理地址，IP地址是广播的IP地址**

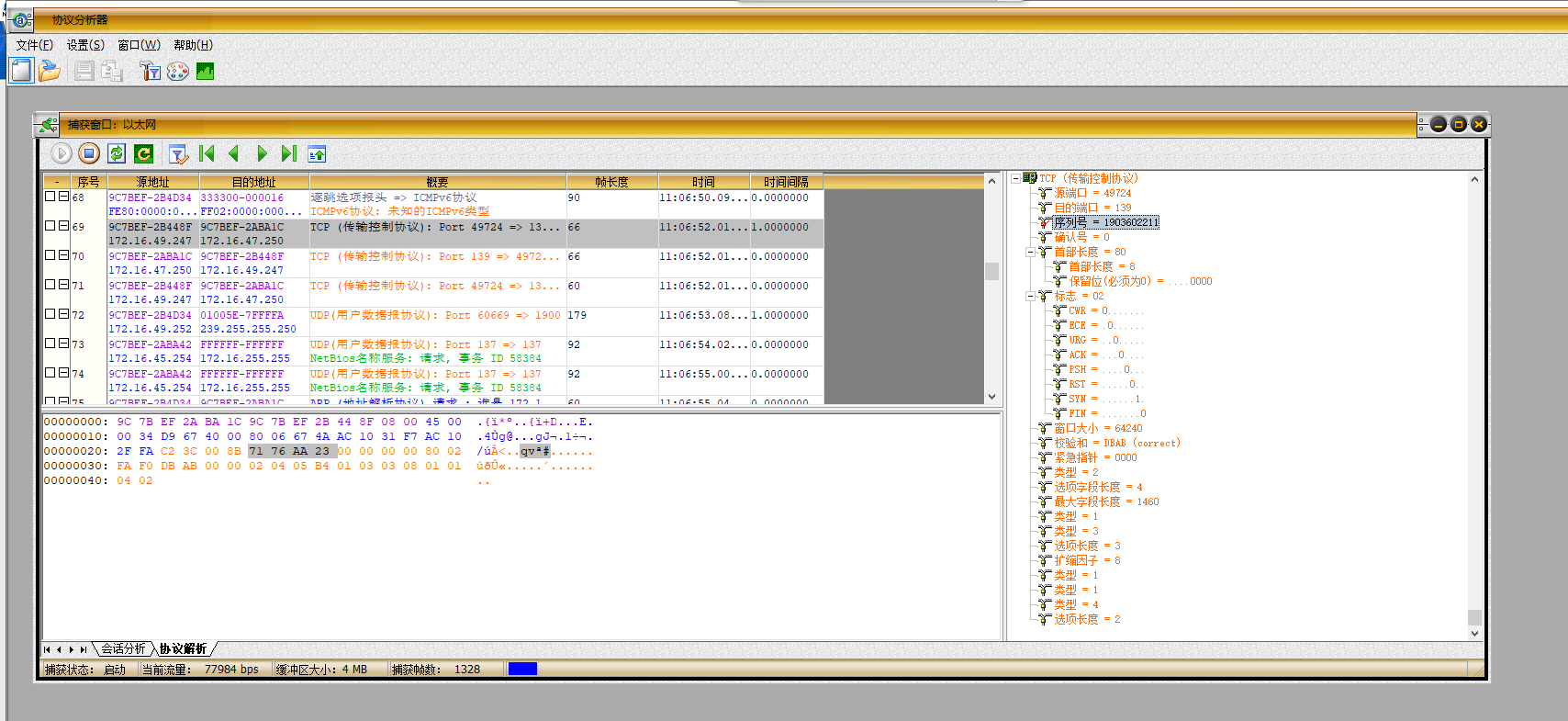
**练习4：查看TCP连接的建立和释放**

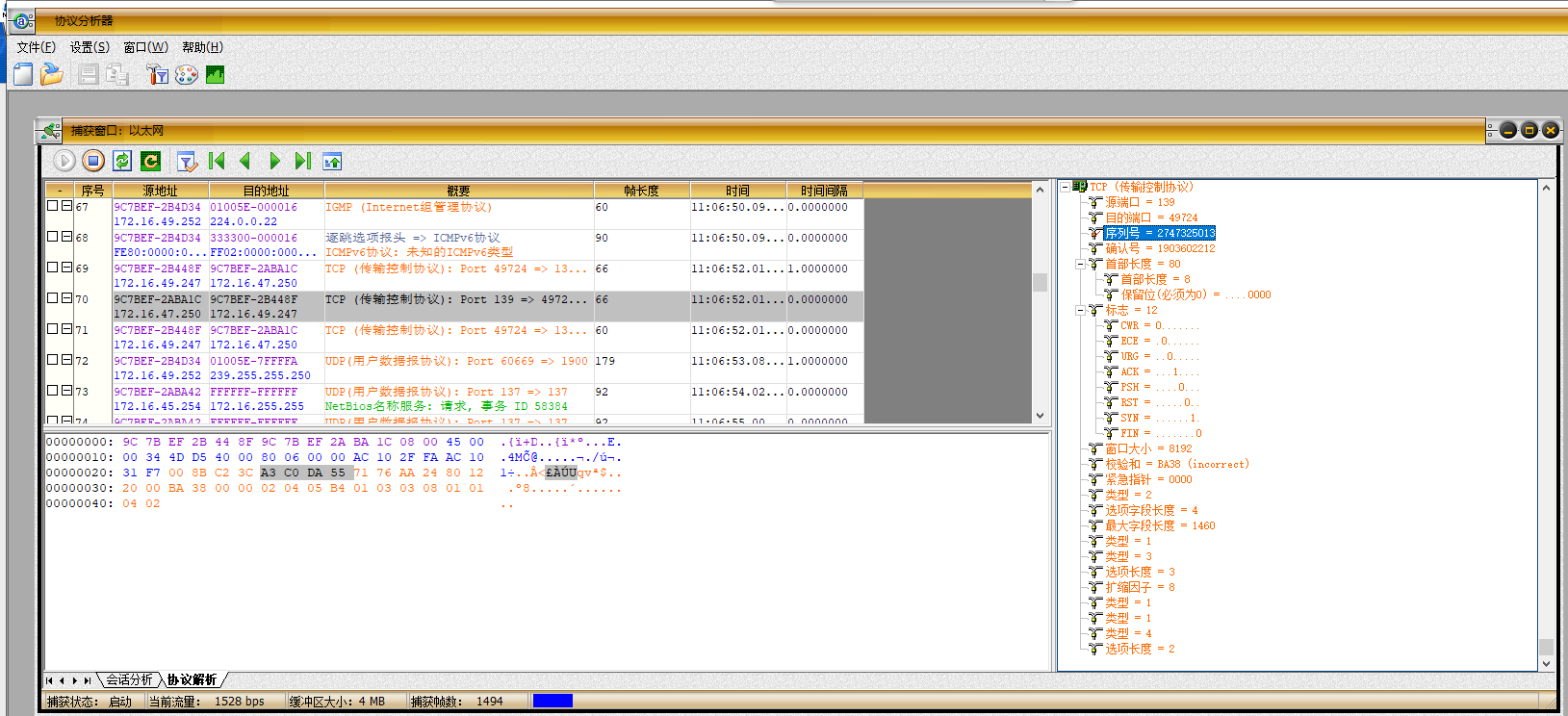
各主机打开工具区的“拓扑验证工具”，选择相应的网络结构，配置网卡后，进行拓扑验证，如果通过拓扑验证，关闭工具继续进行实验，如果没有通过，请检查网络连接。  
     本练习将主机A和B作为一组，主机C和D作为一组，主机E和F作为一组。现仅以主机A、B为例，其它组的操作参考主机A、B的操作。

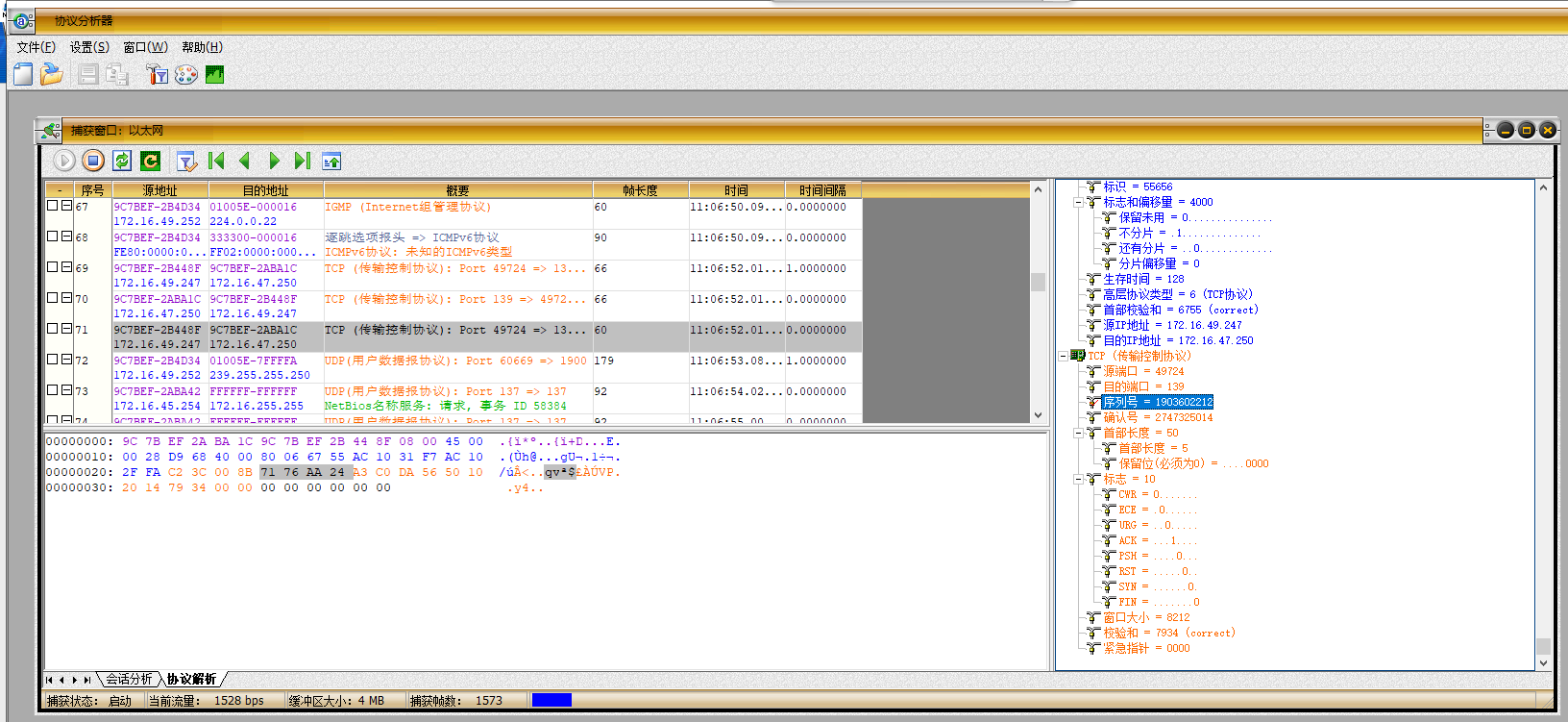
1. 主机B启动协议分析器捕获数据，并设置过滤条件（提取TCP协议）。  
        主机B在命令行下输入：netstat -a -n命令来查看主机B的TCP端口号。



1. 主机A启动TCP工具连接主机B。  
        主机A启动实验平台工具栏中的“TCP工具”。选中“客户端”单选框，在“地址”文本框中填入主机B的IP地址，在“端口”文本框中填入主机B的一个TCP端口，点击[连接]按钮进行连接。
2. 察看主机B捕获的数据，填写下表。







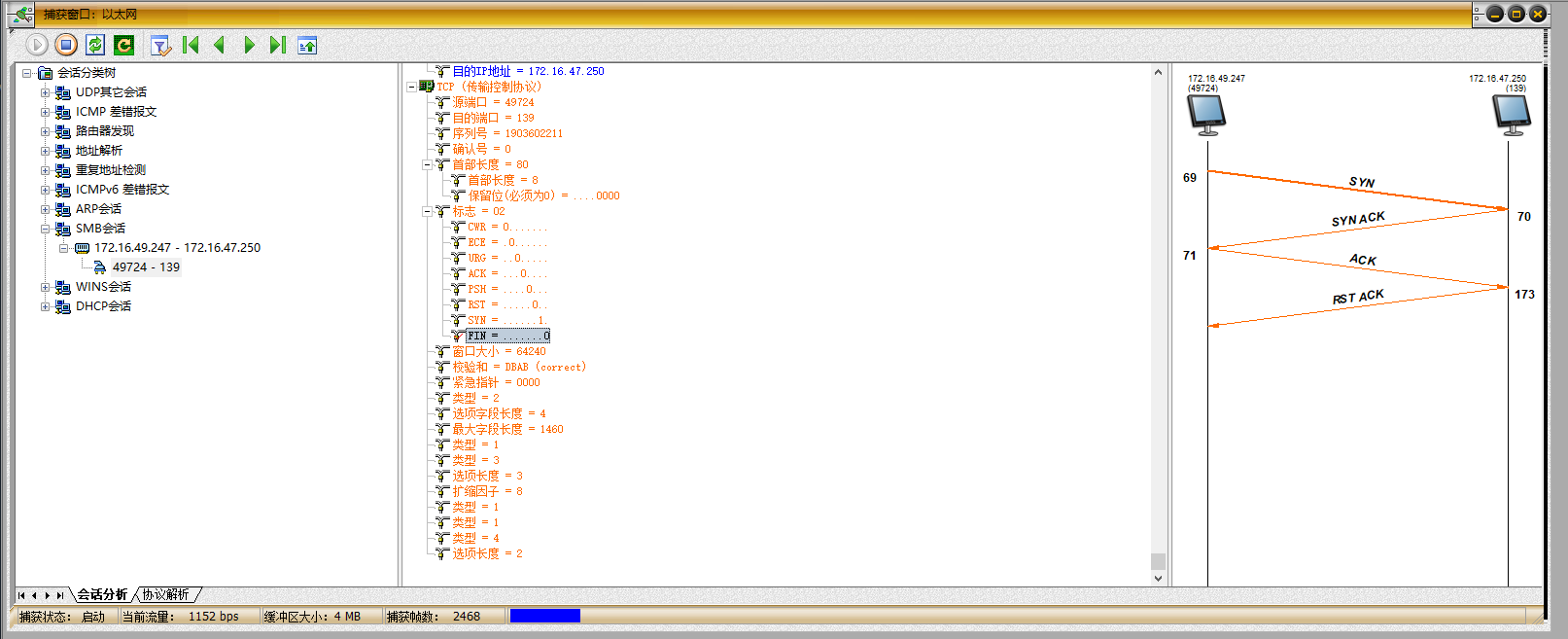
|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 报文1 | 报文2 | 报文3 |
| 序列号 | 1903602211 | 2747325013 | 1903602212 |
| 确认号 | 0 | 1903602212 | 2747325014 |
| ACK | ...0.... | ...1.... | ...1.... |
| SYN | ......1. | ......1. | ......0. |

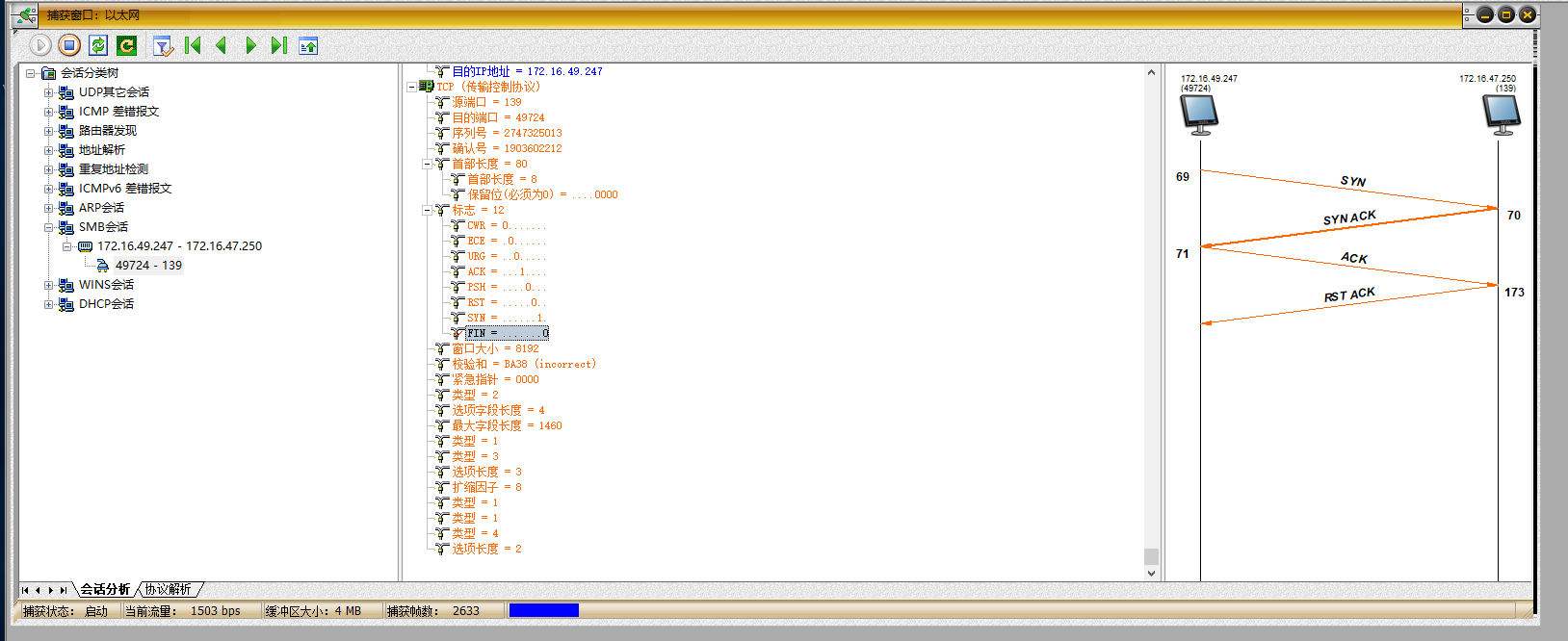
● TCP连接建立时，前两个报文的首部都有一个“最大段长度”字段，它的值是多少？作用是什么？结合IEEE802.3协议规定的以太网最大帧长度分析此数据是怎样得出的。

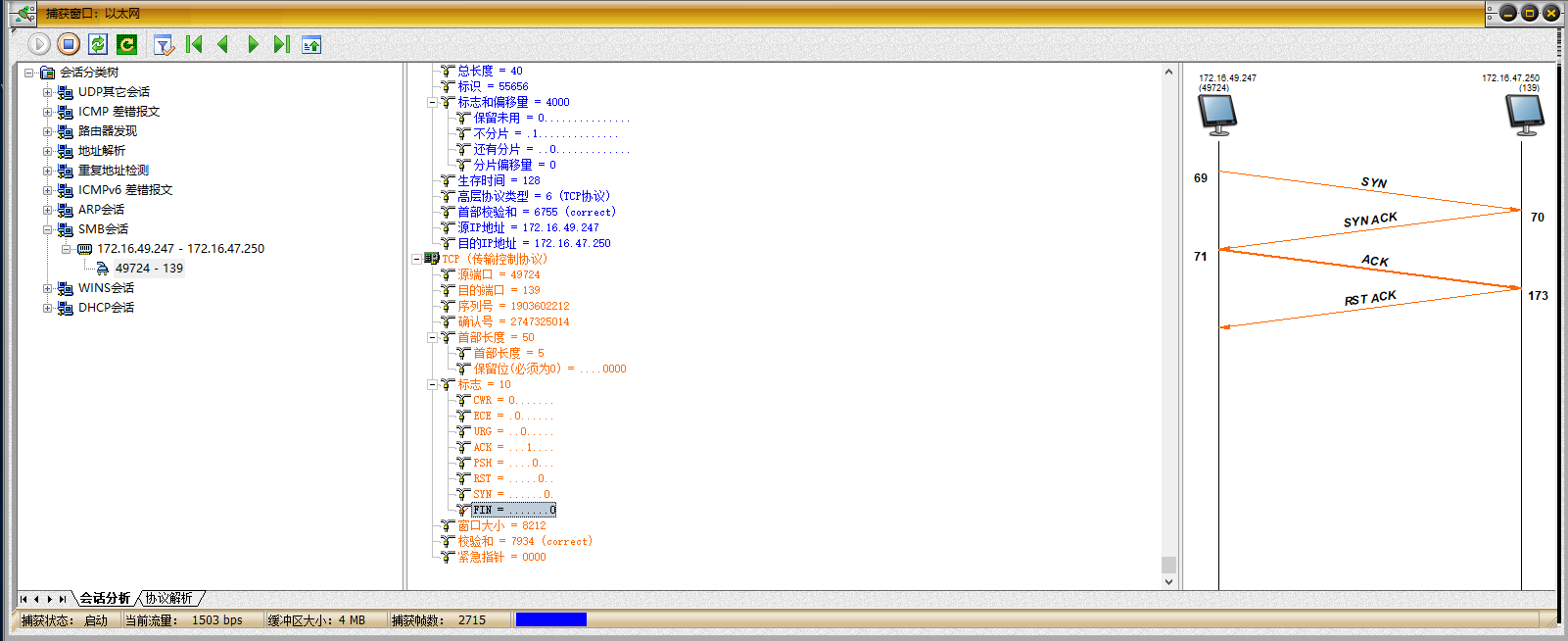
**答：值为1460；  
 它是由发送端指定，表示能在网络上传输的最大的段尺寸  
 最大帧长度= MTU-IP首部长度20- TCP首部长度20**

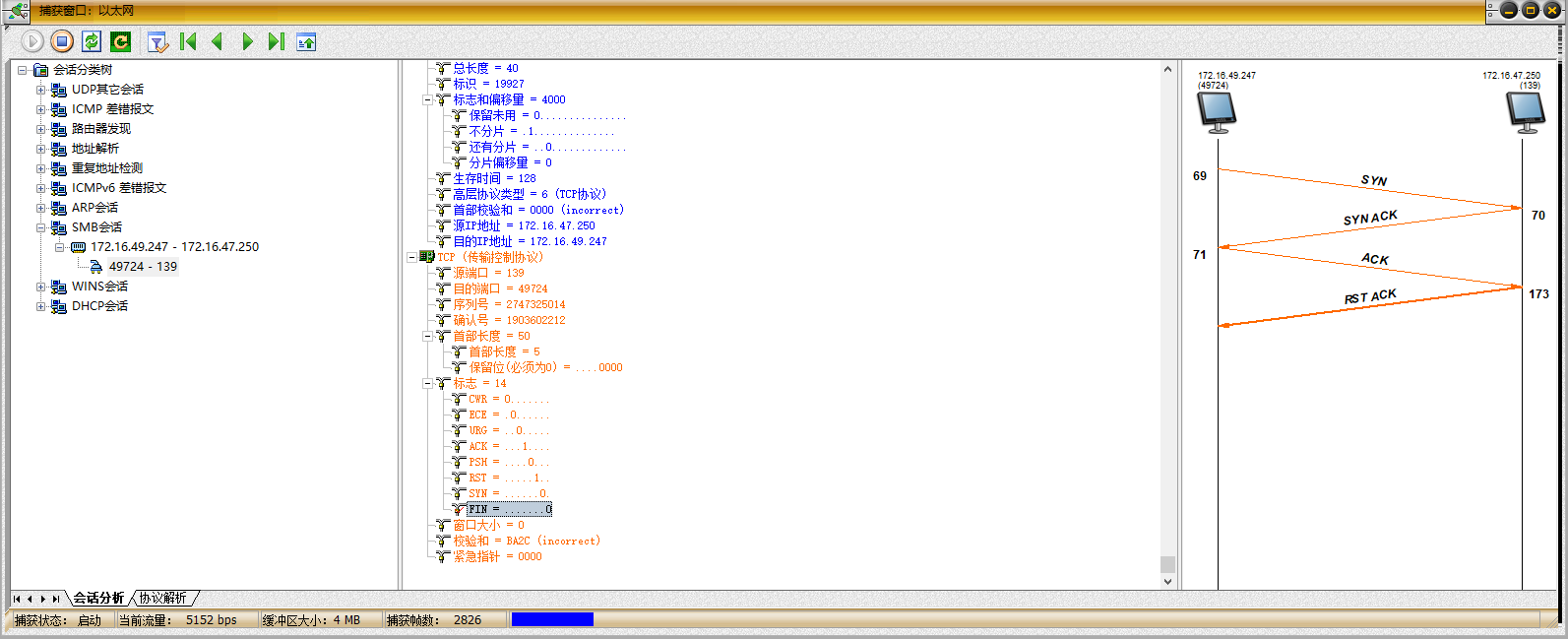
4. 主机A断开与主机B的TCP连接。

5. 查看主机B捕获的数据，填写下表。









|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 报文4 | 报文5 | 报文6 | 报文7 |
| 序列号 | 1903602211 | 2747325013 | 1903602212 | 2747325014 |
| 确认号 | 0 | 1903602211 | 2747325014 | 1903602212 |
| ACK | ...0.... | ...1.... | ...1.... | ...1.... |
| FIN | .......0 | .......0 | .......0 | .......0 |

● 结合步骤3、5所填的表，理解TCP的三次握手建立连接和四次握手的释放连接过程，理解序号、确认号等字段在TCP可靠连接中所起的作用。

**答：三次握手：**

**第一次握手：建立链接客户端发送：SYN包[seq=x]到服务器，并进入SYN\_SEND状态，等待服务器确定；**

**第二次握手：服务器收到SYN包，必须确定客户的SYN[ack=x+1] 同时自己也发送一个SYN包（seq=y）,即SYN+ACK包，此时服务器进入SYN\_RECV状态。**

**第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确定包ACK(ack=y+1),此包发送完毕，客户端和服务器进入ESTABLSHED状态，完成三次握手。**

**四次握手：**

**第一次挥手：客户端发送一个FIN,用来关闭客户端到服务端的数据传输，客户端进入FIN\_WAIT\_1状态**

**第二次挥手：服务器收到FIN后，发送一个ACK给客户端，确定序号为手到序号+1（与SYN相同，一个ACK给客户端，确定序列号为收到序列号+1，同样与SYN相同，一个FIN占用一个序号）服务器端进入CLOSE\_WAIT状态。**

**第三次挥手：服务器发送一个FIN,用来关闭服务器到客户端的数据传输，服务器进入LAST\_ACK状态；**

**第四次挥手：客户端收到FIN后，客户端进入TIME\_WAIT状态接着发送一个ACK给服务器，确定序号为收到序号+1，服务器进入CLOSED状态，完成四次握手链接断开。**