

**《计算机网络》课程设计报告**

**报告题目： 课程设计2（实验）-协议及流程分析**

**开课学院： 计算机与人工智能学院**

**学 号： 202112240**

**专 业： 人工智能**

**姓 名： 郭帆**

**二〇二三年十一**

目录

[《计算机网络》课程设计报告 1](#_Toc9432)

[1.网络层次结构及协议 2](#_Toc18067)

[1.1 TCP/IP协议 2](#_Toc29762)

[1.2 TCP/IP的层次结构 2](#_Toc31684)

[(1)应用层 2](#_Toc6146)

[(2)传输层 3](#_Toc15784)

[（3）网络层 3](#_Toc25986)

[（4）链路层 3](#_Toc19393)

[2.TCP/IP协议的报文封装方式 4](#_Toc18194)

[3.主要的TCP/IP协议及其层次结构 5](#_Toc13473)

[3.1 ICMP 5](#_Toc20569)

[3.1.1 ICMP协议及其帧结构 5](#_Toc1846)

[3.1.2 ICMP报文分类 6](#_Toc24323)

[3.1.3 ICMP报文的主要应用 7](#_Toc21278)

[3.2 以太帧 10](#_Toc29236)

[3.2.1 以太帧及其帧结构 10](#_Toc6379)

[3.3 PPoE 11](#_Toc30286)

[3.3.1 PPoE协议及其帧结构 11](#_Toc4419)

[3.4 ARP 12](#_Toc28667)

[3.4.1 ARP协议及其帧结构 12](#_Toc10811)

[3.4.2 ARP协议工作过程 13](#_Toc2223)

[3.5 IP 14](#_Toc21608)

[3.5.1 IP协议及其帧结构 14](#_Toc20465)

[3.5.2 IP报文捕捉 15](#_Toc17771)

[3.6 TCP 16](#_Toc24619)

[3.6.1 TCP协议及其帧结构 16](#_Toc17425)

[3.6.2 TCP协议工作过程 17](#_Toc19164)

[3.7 UDP 20](#_Toc29242)

[3.7.1 UDP协议及其帧结构 20](#_Toc3922)

[4. 典型网络应用的工作原理及其实现流程 21](#_Toc27424)

[4.1PPPoE工作过程 21](#_Toc14437)

[4.2 HTTP工作过程 21](#_Toc185)

[4.3 FTP工作过程 22](#_Toc24770)

[4.4 SMTP工作过程 23](#_Toc31049)

[4.5典型网络应用实现流程 24](#_Toc6849)

[5. 心得体会 27](#_Toc30301)

[6. 参考文献 28](#_Toc24556)

# 1.网络层次结构及协议

## 1.1 TCP/IP协议

TCP/IP是一组用于网络通信的协议，它是互联网的基础。TCP/IP协议族包含多个协议，其中最为重要的两个协议是TCP和IP。这组协议定义了计算机如何在网络上进行通信，以确保数据的可靠传输。

TCP/IP协议族的主要协议：

(1)IP:IP协议负责在网络上传递数据包。它定义了如何将数据分割成小的单元（数据包），并为每个数据包分配源和目标地址。IP地址是标识网络上设备的数字地址。

(2)TCP:TCP是一种面向连接的协议，负责在通信的两端建立可靠的连接。它确保数据的顺序传递和可靠性，通过错误检测和重传机制来保证数据的完整性。TCP还负责流量控制，以防止数据丢失和网络拥塞。

(3)UDP:UDP是一种无连接的协议，相比TCP更轻量级。它不提供像TCP那样的可靠性和顺序传递，但对于某些应用来说，如实时音视频传输，速度更快且更有效。

(4)ICMP：ICMP主要用于在网络上传递控制消息和错误报告。它包含诸如Ping和Traceroute等工具所需的消息类型，以及用于网络故障排除和错误通知的其他消息。

其中TCP、UDP协议主要应用于传输层,实现端对端的数据传输;ICMP、IP协议主要应用于网络层负责网络中数据包的传送等。

## 1.2 TCP/IP的层次结构

TCP/IP协议族采用四层的层次结构，如图2所示分别是：

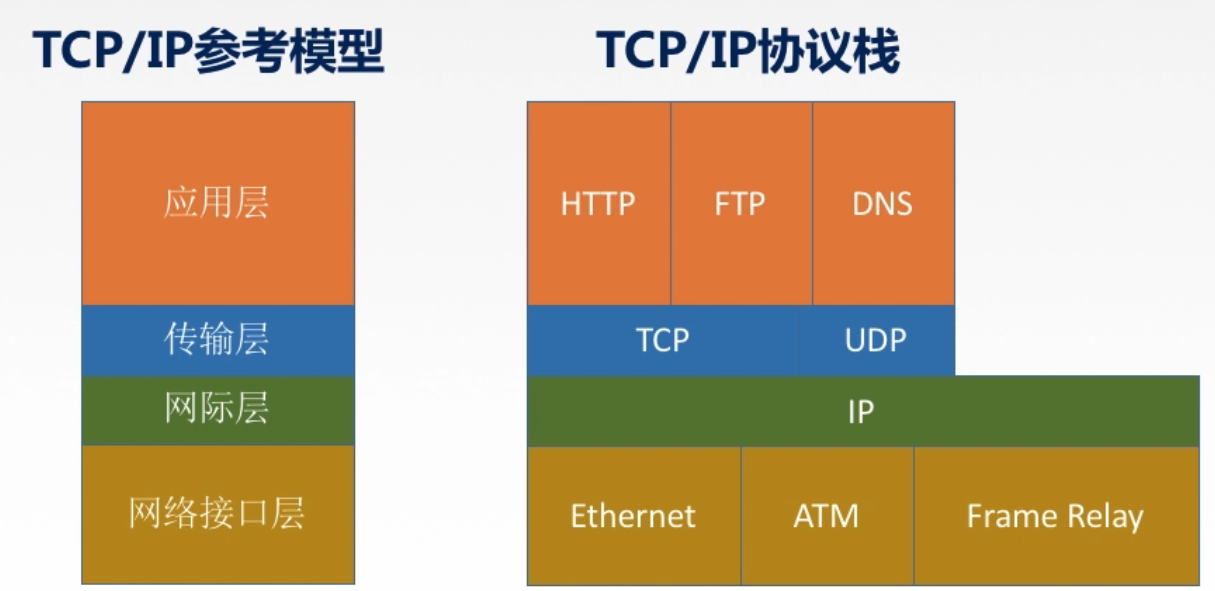


图1 TCP/IP参考模型及TCP/IP协议栈四层层次结构

(1)应用层

应用层包括所有和应用程序协同工作，并利用基础网络交换应用程序的业务数据的协议。一些特定的程序被认为运行在这个层上，该层协议所提供的服务能直接支持用户应用。应用层协议包括HTTP（万维网服务）、FTP（文件传输）、SMTP（电子邮件）、SSH（安全远程登陆）、DNS（域名解析）以及许多其他协议。

(2)传输层

传输层的协议，解决了诸如端到端可靠性问题，能确保数据可靠的到达目的地，甚至能保证数据按照正确的顺序到达目的地。传输层的主要功能大致如下：

①为端到端连接提供传输服务；

②这种传输服务分为可靠和不可靠的，其中TCP是典型的可靠传输，而UDP则是不可靠传输；

③为端到端连接提供流量控制、差错控制、QoS服务质量等管理服务。

传输层主要有两个性质不同的协议：TCP传输控制协议和UDP用户数据报协议。

TCP协议是一个面向连接的、可靠的传输协议，它提供一种可靠的字节流，能保证数据完整、无损并且按顺序到达。TCP尽量连续不断地测试网络的负载并且控制发送数据的速度以避免网络过载。另外，TCP试图将数据按照规定的顺序发送。

UDP协议是一个无连接的数据报协议，是一个“尽力传递”和“不可靠”协议，不会对数据包是否已经到达目的地进行检查，并且不保证数据包按顺序到达。

总体来说，TCP协议传输效率低，但可靠性强；UDP协议传输效率高，但可靠性略低，适用于传输可靠性要求不高、体量小的数据(比如QQ聊天数据)。

（3）网络层

TCP/IP协议网络层的作用是在复杂的网络环境中为要发送的数据报找到一个合适的路径进行传输。简单来说，网络层负责将数据传输到目标地址，目标地址可以是多个网络通过路由器连接而成的某一个地址。另外，网络层负责寻找合适的路径到达对方计算机，并把数据帧传送给对方，网络层还可以实现拥塞控制、网际互连等功能。网络层协议的代表包括：ICMP、IP、IGMP等。

（4）链路层

链路层有时也称作数据链路层或网络接口层，用来处理连接网络的硬件部分。该层既包括操作系统硬件的设备驱动、NIC（网卡）、光纤等物理可见部分，还包括连接器等一切传输媒介。在这一层，数据的传输单位为比特。其主要协议有ARP、RARP等。

这四个层次共同构成了TCP/IP协议栈，每一层都有特定的功能，而且层与层之间通过明确定义的接口进行通信。这种模块化的结构使得TCP/IP协议族成为灵活且可扩展的网络通信体系结构。

TCP/IP模型核OSI参考模型的对应关系入图2所示。

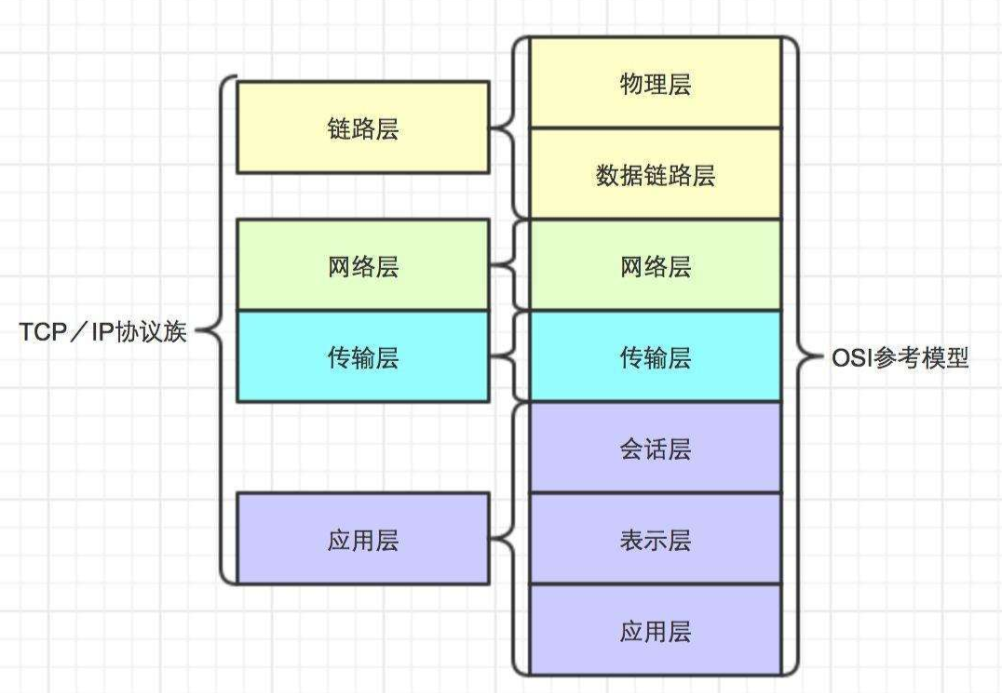


图2 TCP/IP模型与OSI参考模型的对应关系

## 2.TCP/IP协议的报文封装方式

在TCP/IP协议中，上层协议通过对本层数据的封装使用下层协议提供的服务，封装过程如图3所示：

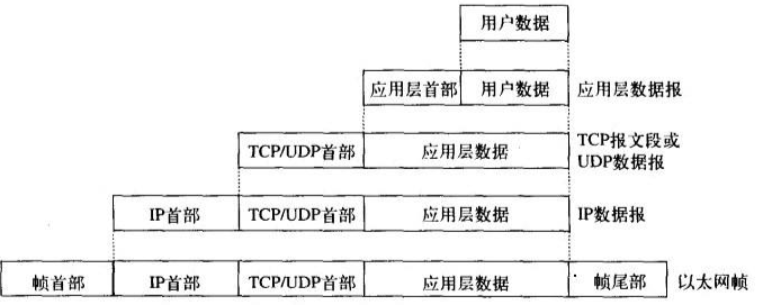


图3 TCP/IP 协议封装过程

TCP/IP协议封装过程：

1. 应用层（Application Layer）数据首先由应用层的应用程序生成，例如，一个Web浏览器或电子邮件客户端。应用层将数据封装为应用层报文。

2. 传输层（Transport Layer）： TCP负责将应用层报文划分为适当的大小，这些部分称为TCP段。 TCP在这一层添加一些控制信息，如源端口号、目标端口号、序列号、确认号等。这一层的工作是提供端到端的可靠数据传输，确保数据的完整性和有序性。UDP数据与TCP数据基本一致，唯一不同的是，UDP传给IP的信息单元称为UDP数据包，并且UDP的首部长为8字节。

3. 网络层（Network Layer）： IP接收传输层的TCP段，将其封装为IP数据包。在这一层，源IP地址和目标IP地址是必不可少的。这一层的主要作用是提供路由功能，将数据包从源主机传输到目标主机。

4.数据链路层（Data Link Layer）： 数据链路层接收来自网络层的IP数据包，将其封装为帧。 在这一层，MAC地址（物理地址）是必需的，用于在局域网中唯一标识设备。数据链路层还包含一些错误检测和纠正的机制，以确保在物理链路上传输的数据的可靠性。

5. 物理层（Physical Layer）：最后，数据帧通过物理层转换为比特流，通过网络媒介传输。物理层处理硬件设备、电缆、光纤等，确保比特流的可靠传输。

总的来说，TCP/IP协议报文封装的过程涉及了多个层次，每个层次都添加了特定的头部信息，以便在网络中正确路由和传输数据。在接收端，数据将被逐层解封，直到达到应用层，然后交给目标应用程序处理。

本节将会介绍常见的计算机网络协议，如：ICMP、以太帧、PPoE、ARP、IP、TCP、UDP以及其报文格式。

# 3.主要的TCP/IP协议及其层次结构

## 3.1 ICMP

### 3.1.1 ICMP协议及其帧结构

ICMP（Internet Control Message Protocol）是一种网络协议，它用于在IP网络上传递控制信息和错误消息。其工作在网络层(OSI模型中的第三层,TCP/IP模型中的第二层)，主要用于在网络设备之间传递控制消息，其允许主机或路由器通过ICMP报文报告差错情况核提供有关异常的报告。ICMP报文携带在IP数据报中进行传送，其报文格式如图1所示，其中IP上层协议号为1。

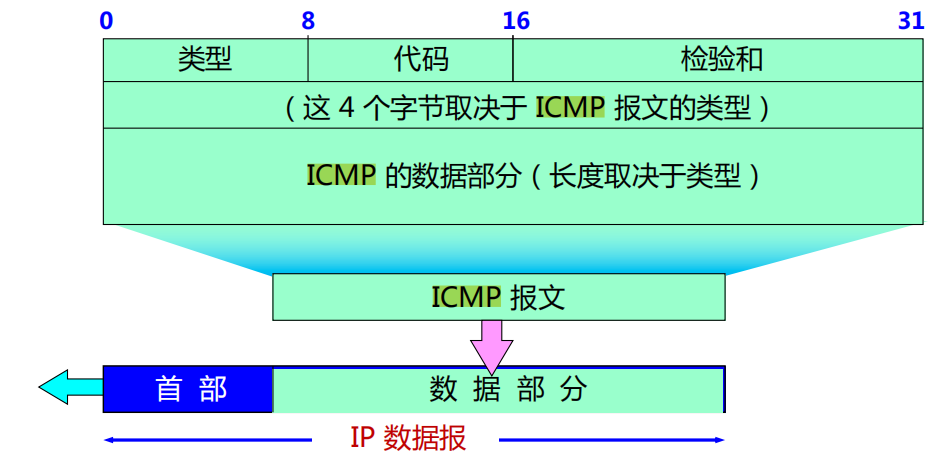


图4 ICMP报文格式

表1 ICMP首部字段说明

|  |  |
| --- | --- |
| ICMP首部字段类型 | 字段说明 |
| 类型 | 占一字节，标识ICMP报文的类型，从类型值来看ICMP报文可以分为两大类。第一类是取值为1~127的差错报文，第2类是取值128以上的信息报文 |
| 代码 | 占一字节，标识对应ICMP报文的代码。它与类型字段一起共同标识了ICMP报文的详细类型 |
| 校验和 | 这是对包括ICMP报文数据部分在内的整个ICMP数据报的校验和，以检验报文在传输过程中是否出现了差错（其计算方法与在我们介绍IP报头中的校验和计算方法是一样的） |

### 3.1.2 ICMP报文分类

常见的ICMP报文可以分为两大类，即差错报文和查询报文。

(1)差错报文

常见的ICMP差错报文有以下几种：目的站不可达、时间超时、改变路由、参数问题[2]。

表2 ICMP差错报文含义及类型值取值

|  |  |
| --- | --- |
| 含义 | 类型值取值 |
| 目的站不可达 | 3 |
| 改变路由 | 5 |
| 时间超时 | 11 |
| 参数问题 | 12 |

①目的站不可达

当路由器不能给数据报找到合适的路由路径，或者主机不能将数据报递交给上层协议时，相应的IP数据报就会被丢弃，然后一个目的站不可达差错控制报文将会被返回给源主机。目的站不可达差错可以由很多因素引起，例如网络不可达、主机不可达、协议不可达、端口不可达等，可以在报文首部中的代码字段指出具体原因。在ICMP目的不可达报文的数据区域，装载了引起目的站不可达的IP数据报首部及数据报数据区的前8个字节，当源主机收到这样一份ICMP报文后，它能根据报文数据区中的数据判断是哪个数据报出现了问题。装载引起差错的数据报数据区的前8个字节是因为这8个字节恰好覆盖了TCP报文或UDP报文中的端口号字段，IP层能够根据这个端口号把ICMP报文传递给具体的上层协议进行处理。

②时间超时

时间超时可以用来防止数据报在网络中被循环的路由，在IP首部中都有一个生存时间（TTL）字段，数据报每被转发一次，TTL的值便会减1，当TTL的值被减为0时，数据报会被网络丢弃，同时一个ICMP数据报超时报文会被返回给源主机。此外在分片重装时，也用到了ICMP超时报文，若某个数据报在重装过程中其重装时间超时，而数据分片还没有全部到达，此时与该数据报所有相关的分片将被删除，同时，一个ICMP超时报文将被返回给源主机。

③ 参数问题：当路由器或目的主机收到的数据报的首部中的字段的值不正确时，就丢弃该数据报，并向源站发送参数问题报文。

④改变路由（重定向）路由器将改变路由报文发送给主机，让主机知道下次应将数据报发送给另外的路由器。

1. 查询报文

常见的ICMP查询报文有:回送(Echo)请求或应答,时间戳(Timestamp)请求或回答

表3 ICMP查询报文含义及其类型值取值

|  |  |
| --- | --- |
| 含义 | 类型值取值 |
| 回送请求或应答 | 8或0 |
| 时间戳请求或回答 | 13或14 |

①回送请求报文:是由主机或路由器向一个特定的目的主机发出的询问。收到此报文的机器必须给源主机发送ICMP回送应答报文。这种询问报文用来测试目的站是否可达以及了解其有关状态。

②ICMP时间戳请求报文:允许系统向另一个系统查询当前的时间。该ICMP报文的好处是它提供了毫秒级的分辨率，而利用其他方法从别的主机获取的时间只能提供秒级的分辨率。请求端填写发起时间，然后发送报文。应答系统收到请求报文时填写接收时间戳，在发送应答时填写发送时间戳。大多数的实现是把后面两个字段都设成相同的值。

### 3.1.3 ICMP报文的主要应用

(1)有测试两台主机之间数据报来回一次的传输时间，即ping操作

原理:传输时，主机填充原始时间戳，接受方收到请求后填充接受时间戳后以类型值字段14的报文格式返回，发送方计算这个时间差。

Ping操作的过程如图2所示，Wirshake抓包结果如图5、6、7所示。

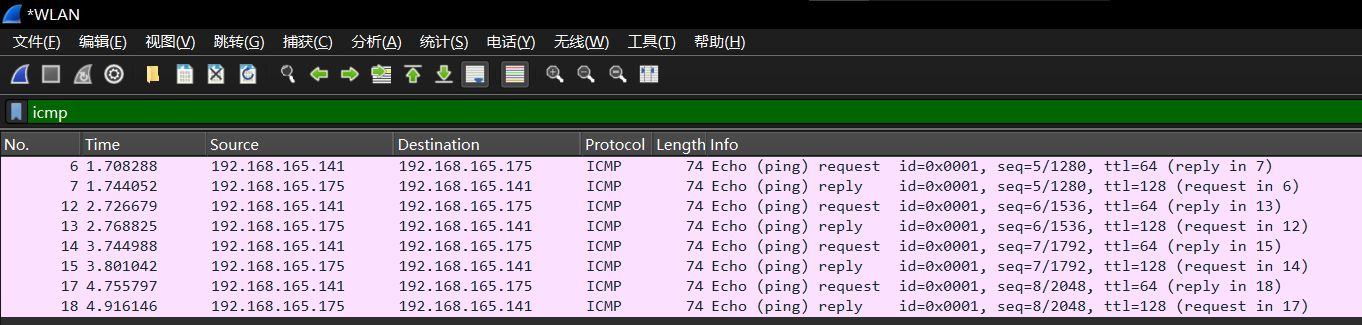


图5 Wirshake抓包结果

从图6可以看到ICMP Request报文格式，ICMP数据报中的Type的值是8，括号中的ping表示是通过ping命令产生的ICMP数据报，后面request就代表请求数据报，通过这些信息可以确定这是一个ICMP请求数据报。

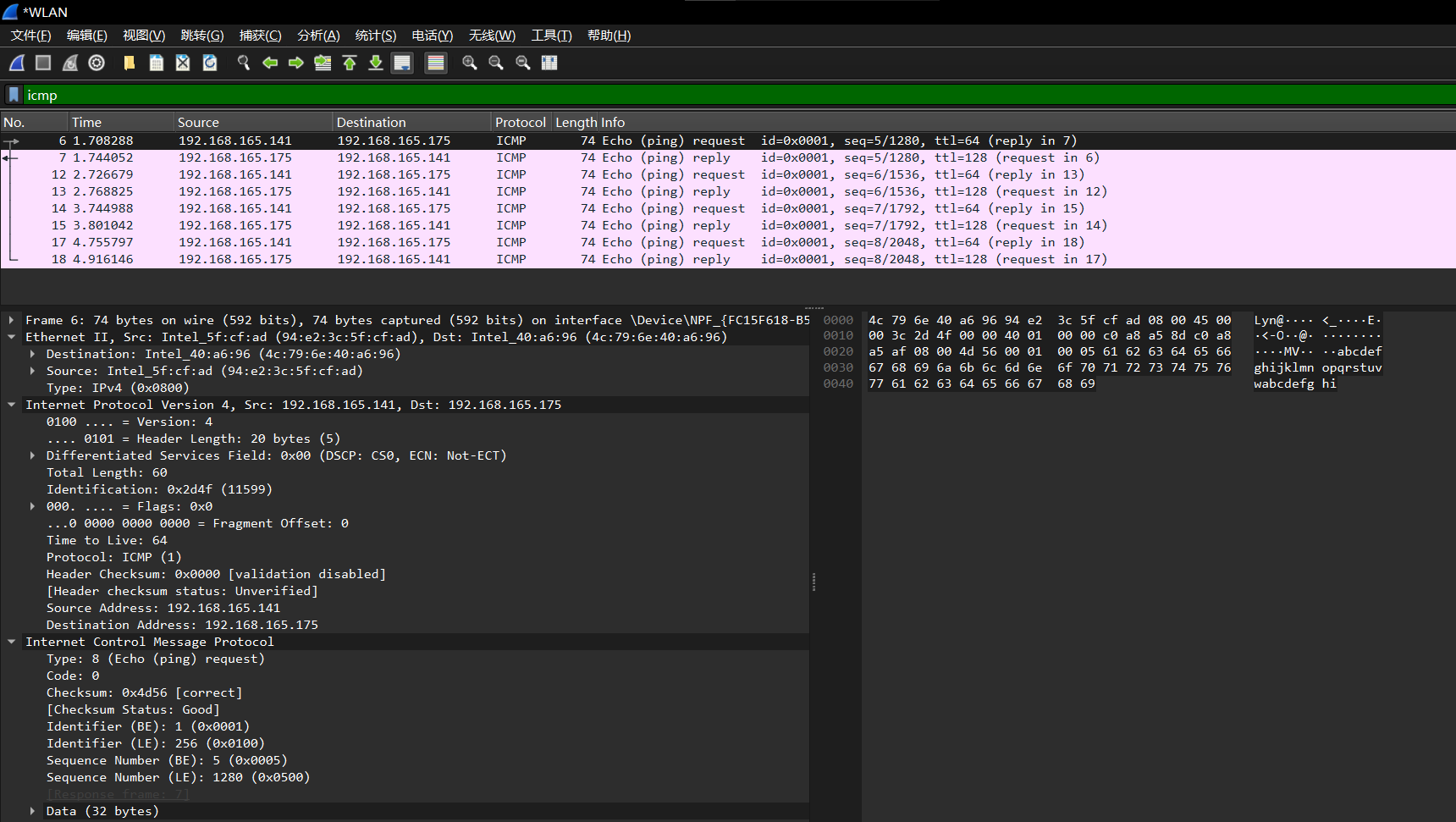


图6 ICMP请求报文结构

从图7中可以看出ICMP Reply报文格式，其中ICMP数据报中的Type的值是0，括号中的ping表示是通过ping命令产生的ICMP数据报，后面reply就代表应答数据报，通过这些信息可以这是一个ICMP应答数据报。

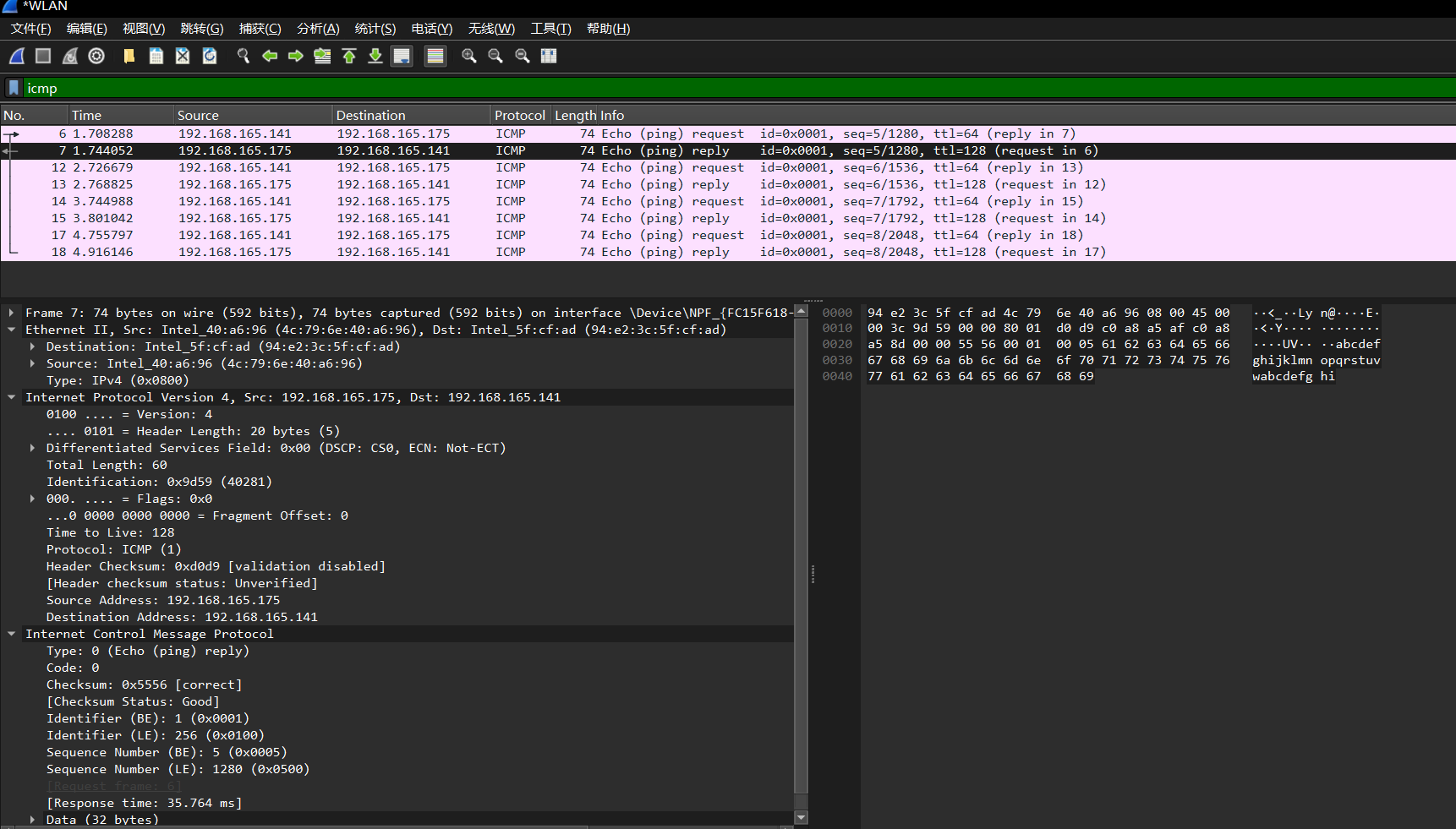


图7 ICMP回应报文结构

1. traceroute基于ICMP协议实现

Traceroute 是一种网络诊断工具，用于确定数据包从源主机到目标主机经过的路由路径。Traceroute 使用 ICMP 协议或 UDP 协议来发送数据包，并通过逐跳的 TTL（Time To Live）值的设置，逐步探测数据包在网络中的路径

原理:当源收到ICMP报文，计算 RTT,Tracert针对同一RTT值执行上述过程3次。停止条件:UDP段最终到达目的地主机,目的地返回ICMP “端口不可达”分组 (类型3, 编码3),当源主机得到该ICMP，停止。

Traceroute执行过程及结果如图8、9所示：

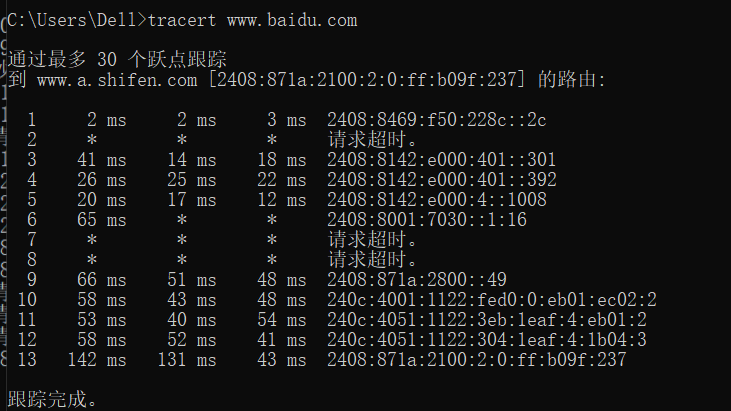


图8 tracert结果和过程

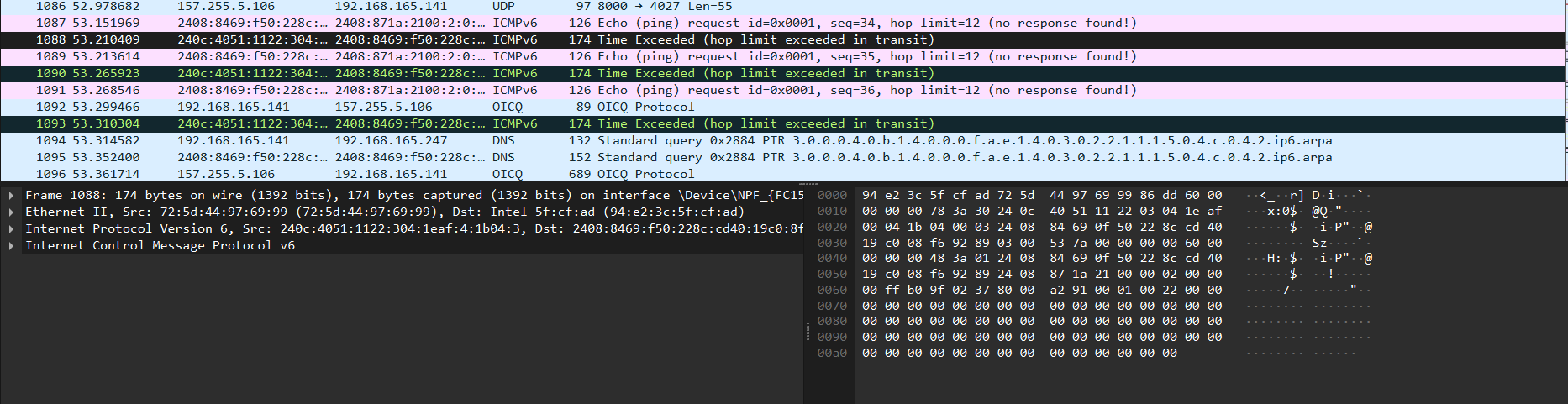


图9 使用wireshark跟踪tracert过程

从图9可以看出，当第n个数据报到达第n个路由器，路由器会向源主机发送时间超时ICMP差错报文。

## 3.2 以太帧

### 3.2.1 以太帧及其帧结构

以太帧有很多种类型。不同类型的帧具有不同的格式和 MTU 值。但在同种物理媒体上都可同时存在。常见的有三种帧格式：Ethernet II 帧是最常见的帧类型，并通常直接被 IP 协议使用；非标准 IEEE 802.3 帧变种；IEEE 802.3帧（后跟逻辑链路控制（LLC) 帧）。

(1)Ethernet II 帧

Ethernet II 帧的帧结构如图10所示.



图10 Ethernet II 帧结构

Ethernet II 类型以太网帧的最小长度为 64 字节（6＋6＋2＋46＋4），最大长度为 1518 字节（6＋6＋2＋1500＋4）。其中： （1）前 12 字节分别标识出发送数据帧的源节点 MAC 地址和接收数据帧的目标节点 MAC 地址； （2）接下来的 2 个字节标识出以太网帧所携带的上层数据类型，如 16 进制数0x0800代表 IP 协议数据，16 进制数0x86dd代表 IPv6 协议数据，16 进制数0x809B代表 AppleTalk 协议数据，16 进制数0x8138代表 Novell 类型协议数据等； （3）在不定长的数据字段（Data）：其长度是 46 至 1500 字节； （4）4 个字节的帧校验序列（Frame. Check Sequence，FCS），采用 32 位 CRC 循环冗余校验对从“目标 MAC 地址”字段到“数据”字段的数据进行校验。

通过WireShark对 Ethernet II帧进行抓包分析，所抓取到的包如图11所示：

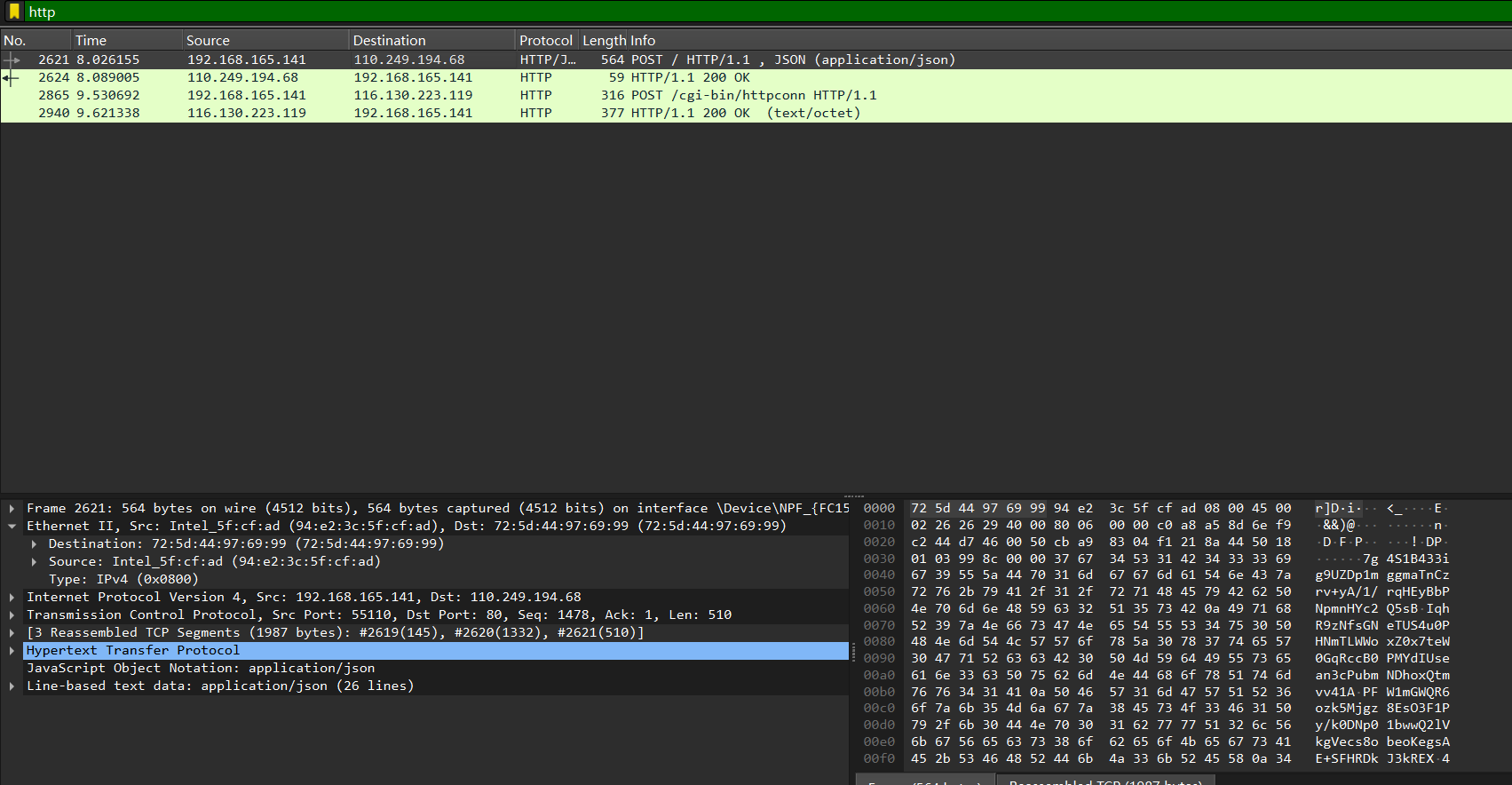


图11 Wireshark抓包Ethernet II

从图11中可以看出该Ethernet II帧所包含的信息有：

①Src: Intel\_5f:cf:ad (94:e2:3c:5f:cf:ad)： 表示源地址，即发送该数据帧的网络接口的物理地址（MAC 地址）。在这个例子中，源地址是 Intel 公司的网络接口卡，其具体的 MAC 地址是 94:e2:3c:5f:cf:ad。

②Dst: 72:5d:44:97:69:99 (72:5d:44:97:69:99)： 表示目标地址，即接收该数据帧的网络接口的物理地址（MAC 地址）。在这个例子中，目标地址是 72:5d:44:97:69:99。

③Type：IPV4，表明数据部分使用的协议是IPv4协议。

## 3.3 PPoE

### 3.3.1 PPoE协议及其帧结构

PPPoE（Point-to-Point Protocol over Ethernet）是一种将点对点协议封装在以太网中的网络协议。表面中文意思是以太网上的PPP。该协议提供了在广播式的网络比如以太网中，多台主机连接到远端的访问集中器（访问集中器也称为宽带接入服务器）上的一种标准。在于以太网并非点到点网络，而是多点到多点的广播网络[1]。这跟之前的PPP的区别就在这里，不是单纯点到点了。PPPoE就整体来看就分成三个阶段，发现阶段，会话阶段，结束阶段。它允许在以太网连接上建立和维护点对点连接，通常用于在宽带接入网络（如DSL和光纤）中实现宽带拨号连接。

其帧的帧格式如图12所示：

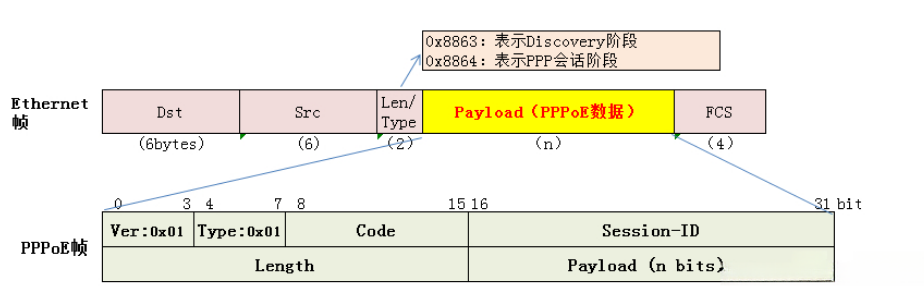


图12 PPoE帧帧格式

PPoE帧中每个字段代表的含义：

1.Ver： 版本号， 4bit， PPPoE规范的本版本为0x01

2.Type： 类型， 4bit， PPPoE规范的本版本应设置为0x01

3.Code： 代码字段、8bit， 根据两阶段中各种数据包的不同功能而值不同

4.Session\_ID： 长度是16比特。是一个网络字节序的无符号值。其值在后面Discovery数据包中定义。

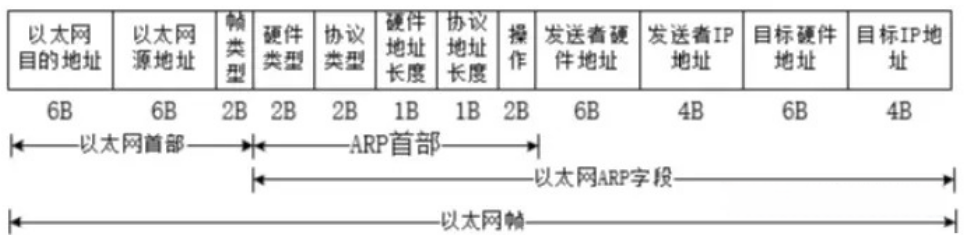
5.Length： 长度是16比特。该值是PPPoE的Payload长度。它不包括以太网头部和PPPoE头部的长度。

6.Payload: PPPoE的Payload，包含0个或多个Tag。

## 3.4 ARP

### 3.4.1 ARP协议及其帧结构

地址解析协议，即ARP（Address Resolution Protocol），是根据IP地址获取物理地址的一个TCP/IP协议。主机发送信息时将包含目标IP地址的ARP请求广播到局域网络上的所有主机，并接收返回消息，以此确定目标的物理地址；收到返回消息后将该IP地址和物理地址存入本机ARP缓存中并保留一定时间，下次请求时直接查询ARP缓存以节约资源。地址解析协议是建立在网络中各个主机互相信任的基础上的，局域网络上的主机可以自主发送ARP应答消息，其他主机收到应答报文时不会检测该报文的真实性就会将其记入本机ARP缓存。ARP协议的帧结构如图13所示：

图13 ARP帧格式

ARP帧每个字段的含义如下：

目的MAC地址（6字节）：指示ARP请求或响应的目标设备的物理MAC地址。在ARP请求中，这个字段通常被设置为全0，表示请求的目标MAC地址未知。在ARP响应中，这个字段包含目标设备的真实MAC地址。

源MAC地址（6字节）：指示ARP请求或响应的发送设备的物理MAC地址。

帧类型（2字节）：指示数据帧中所携带的上层协议类型，对于ARP数据帧，该字段的值为0x0806。

硬件类型（2字节）：指示硬件地址的类型，如以太网的硬件类型为1。

协议类型（2字节）：指示上层协议类型，如IPv4的协议类型为0x0800。

硬件地址长度（1字节）：指示硬件地址的长度，以字节为单位，如以太网的地址长度为6。

协议地址长度（1字节）：指示协议地址的长度，以字节为单位，如IPv4地址的长度为4。

操作码（2字节）：指示ARP请求或响应的类型，如请求为1，响应为2。

发送方硬件地址（6字节）：指示ARP请求或响应的发送设备的物理MAC地址。

发送方协议地址（4字节）：指示ARP请求或响应的发送设备的协议地址，如IPv4地址。

目标硬件地址（6字节）：指示ARP请求或响应的目标设备的物理MAC地址。在ARP请求中，该字段通常被设置为全0，表示请求的目标MAC地址未知。在ARP响应中，该字段包含目标设备的真实MAC地址。

目标协议地址（4字节）：指示ARP请求或响应的目标设备的协议地址，如IPv4地址。

### 3.4.2 ARP协议工作过程

ARP工作过程如下：

假设主机A和B在同一个网段，主机A要向主机B发送信息，具体的地址解析过程如下：

主机A首先查看自己的ARP缓存表，确定其中是否包含有主机B对应的ARP表项。如果找到了对应的MAC地址，则主机A直接利用ARP表中的MAC地址，对IP数据包进行帧封装，并将数据包发送给主机B。

如果主机A在ARP表中找不到对应的MAC地址，则将缓存该数据报文，然后以广播方式发送一个ARP请求报文。ARP请求报文中的发送端IP地址和发送端MAC地址为主机A的IP地址和MAC地址，目标IP地址和目标MAC地址为主机B的IP地址和全0的MAC地址。由于ARP请求报文以广播方式发送，该网段上的所有主机都可以接收到该请求，但只有被请求的主机（即主机B）会对该请求进行处理。

主机B比较自己的IP地址和ARP请求报文中的目标IP地址，当两者相同时进行如下处理：将ARP请求报文中的发送端（即主机A）的IP地址和MAC地址存入自己的ARP表中。之后以单播方式发送ARP响应报文给主机A，其中包含了自己的MAC地址[2]。

主机A收到ARP响应报文后，将主机B的MAC地址加入到自己的ARP表中以用于后续报文的转发，同时将IP数据包进行封装后发送出去。

当主机A和主机B不在同一网段时，主机A就会先向网关发出ARP请求，ARP请求报文中的目标IP地址为网关的IP地址。当主机A从收到的响应报文中获得网关的MAC地址后，将报文封装并发给网关。如果网关没有主机B的ARP表项，网关会广播ARP请求，目标IP地址为主机B的IP地址，当网关从收到的响应报文中获得主机B的MAC地址后，就可以将报文发给主机B；如果网关已经有主机B的ARP表项，网关直接把报文发给主机B。

其工作过程可简化为如图14所示：

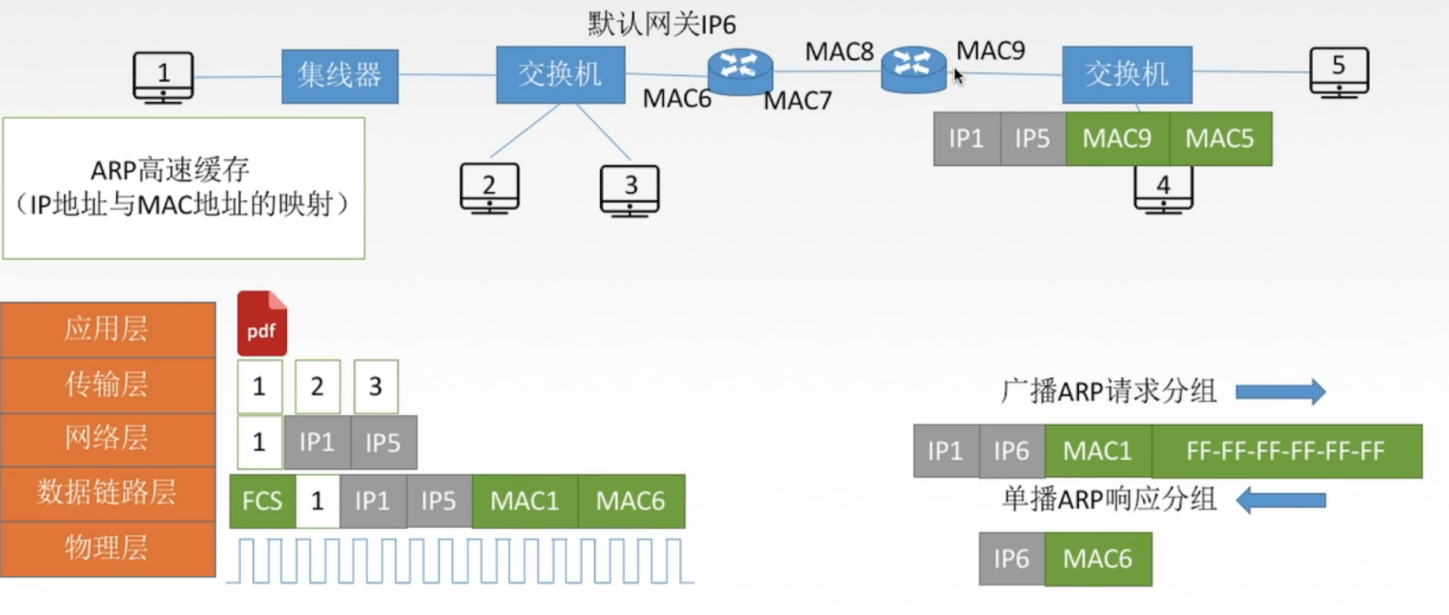


图14 ARP工作过程

使用WireShark抓包ARPj结果如图 15 所示：

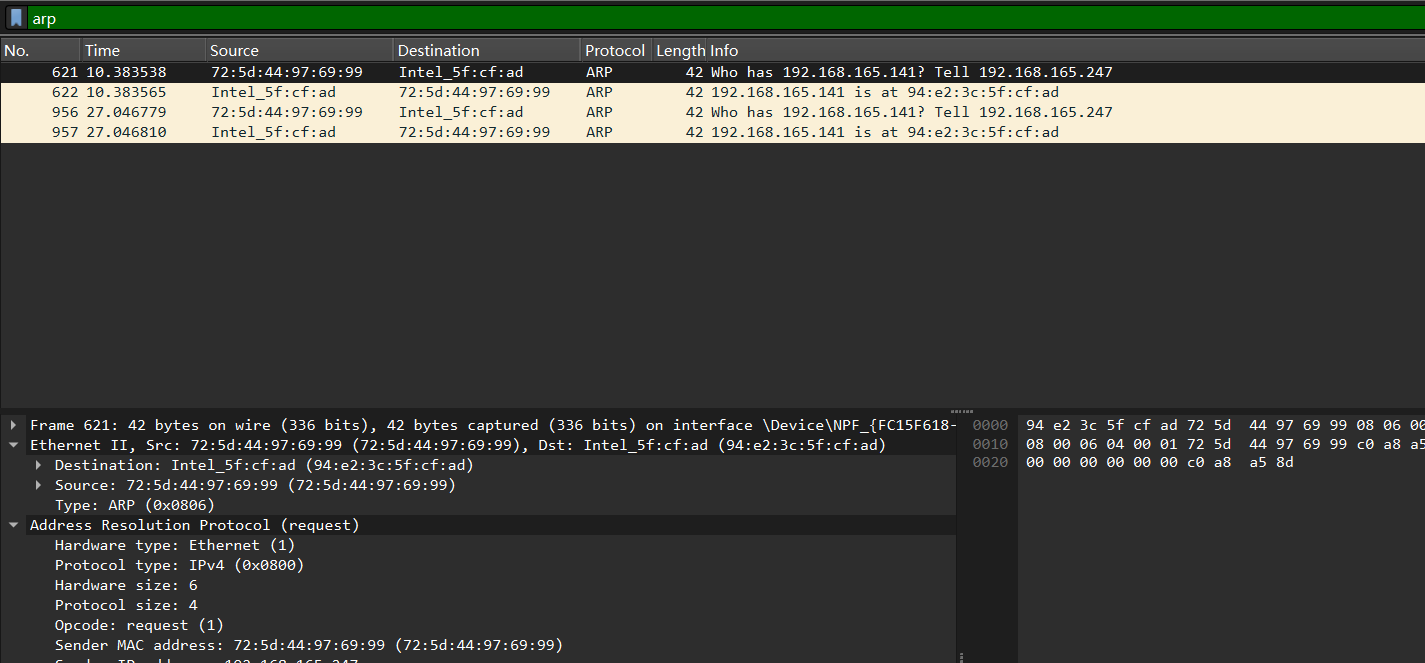


图15 Wireshark抓包ARP帧

通过对Length Info的分析可知第一个ARP帧是在询问ip为192.168.165.141的mac地址是多少，属于ARP请求报文，第二个ARP帧是在回复源主机自己的mac地址是94:e2:3c:5f:cf:ad，是ARP回复报文。

## 3.5 IP

### 3.5.1 IP协议及其帧结构

IP（Internet Protocol）网际协议，是TCP/IP协议的核心部分，其帧结构如图 16所示。虽然IPV4终将会被IPV6替代，但是目前IPV4仍然是IP协议的主流版本，所以我们重点介绍一下IPV4版本。在学习TCP协议和socket套接字编程的时候我们知道，如果想要在互联网这个共享网络中精确定位到一台主机，那么就一定需要该主机的IP地址。主机是拥有IP地址，但是不能进行路由控制（Routing，意思是中转、分组数据包），路由器（Router）这种设备既有IP地址，又可以进行路由控制；我们可以将接入互联网中的主机和路由器都叫做节点。



图16 IP帧结构

IP帧结构每个字段的说明如下：

版本号：4bit，标识目前采用的IP协议版本号，0100表示IPV4，0110表示IPV6。

首部长度：4bit，单位为32位二进制（32bit=4字节）表示IP包头的长度，范围为20–60字节。

优先级与服务类型：8bit，前三位表示优先级，000–111，数值越大优先级越高；中间四位表示服务类型，按顺序分别表示时延、吞吐量、可靠性、成本；0表示普通，1表示较好。

最后一位保留，恒为0。

总长度：16bit，单位为字节，表示整个IP数据段的长度，（应用层+TCP/UDP+IP包头），理论上IP数据段最大可达65535字节，但由于MTU限制，一般无法到达这个值。

MTU: 最大传输单元，一般为1500bit。

标识符：16bit，对较大的上层数据包进行分段操作，由同一个数据包分开的标识符数值相同，方便目的端区分。

标志：3bit，第一位：置为0。第二位：1表示不分片，0表示分片。第三位：1表示后面还有分片，0表示后面无分片即最后一个报文。

段偏移量：13bit，表示分片相对于原始数据的偏移量，用于重组数据。

TTL生存时间：8bit，表示该IP数据包可以经过的最大路由器数量，每经过一个路由器，TTL减一，当减为零时丢弃该包；防止IP数据包在网络中无限传输，占用资源。

（一般TTL < 100 发送端系统为linux， TTL > 100,一般为windows系统 ，255为特例）

协议：8bit，用于区分上层协议。1，6，17 较常见。

首部校验和：16bit，校验IP包头（20–60字节）不包含数据。

源地址和目标地址：均为32bit，源地址IP和目标地址IP。

可选项：以32bit为单位，不足时用0补充，主要用于测试。

### 3.5.2 IP报文捕捉

使用Wireshark对IP报文进行查看，结果如图17所示：

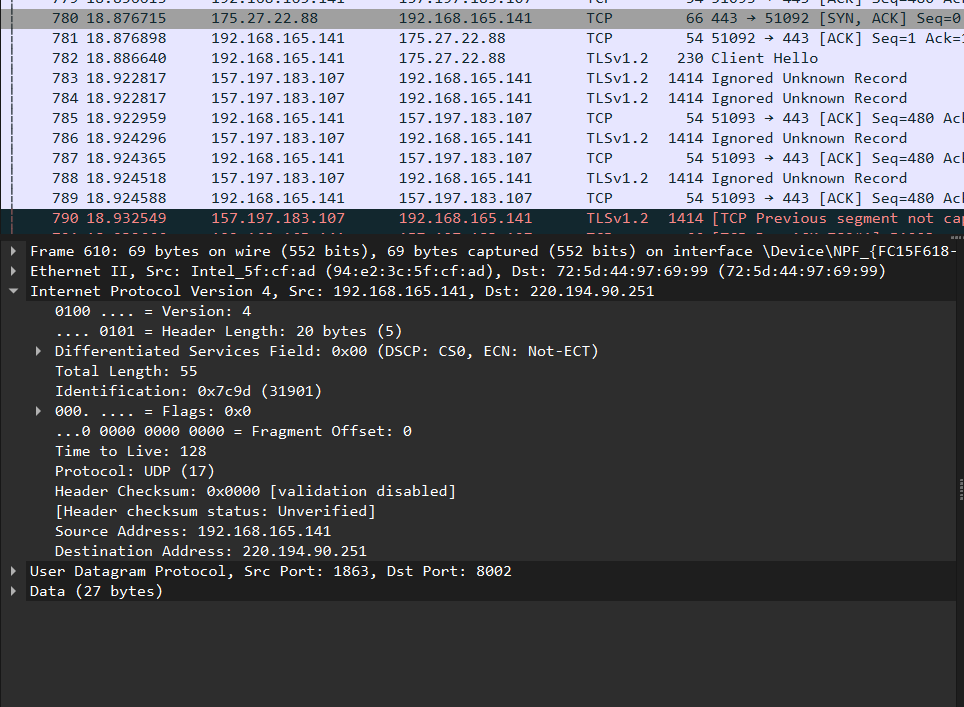


图 17 Wireshark分析IP数据报

从图17中可以得出以下信息：该IP报文所使用的IP版本为IPv4，头部长度为20字节，总长度55字节，表示为331901，片偏移为0，生存时间为128，协议是UDP，首部校验和为0x0000，源地址为192.168.165.141，目的地址为220.194.90.251。

## 3.6 TCP

### 3.6.1 TCP协议及其帧结构

TCP 是面向连接的运输层协议。应用程序在使用 TCP 协议之前，必须先建立 TCP 连接。在传送数据完毕后，必须释放已经建立的 TCP 连接每一条 TCP 连接只能有两个端点，每一条 TCP 连接只能是点对点的(一对一)

TCP 提供可靠交付的服务。通过 TCP 连接传送的数据，无差错、不丢失、不重复，并且按序到达，且TCP 提供全双工通信。TCP 允许通信双方的应用进程在任何时候都能发送数据。TCP 连接的两端都设有发送缓存和接受缓存，用来临时存放双向通信的数据面向字节流[3]。TCP 中的“流”指的是流入到进程或从进程流出的字节序列。TCP协议的帧格式如图18所示：

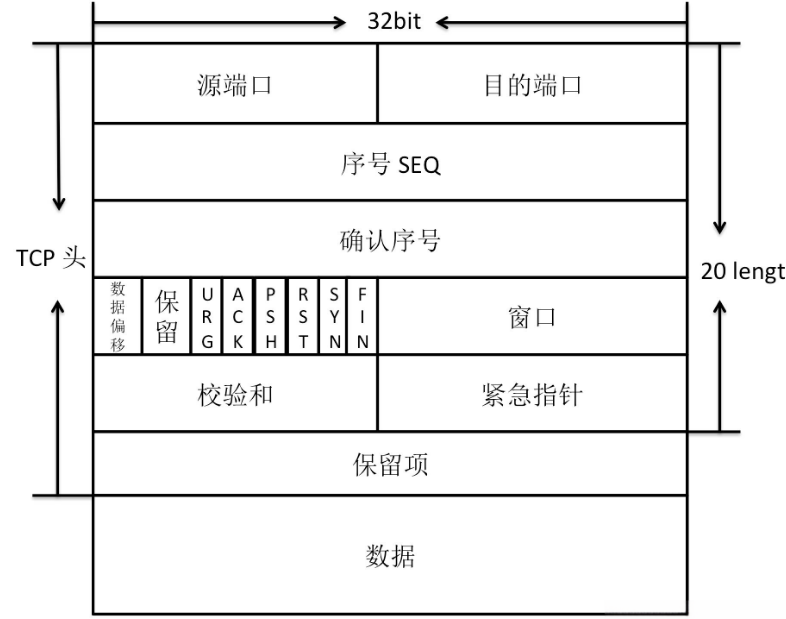


图18 TCP协议帧格式

### 3.6.2 TCP协议工作过程

(1)TCP建立

TCP连接建立需要经历三次握手，其中关键步骤如图19所示：

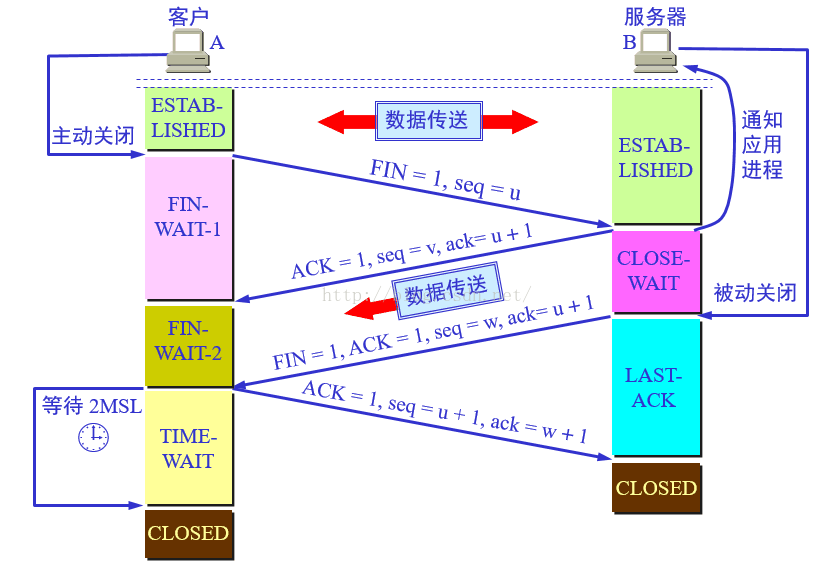


图19 TCP连接建立三次握手过程

如图19所示，TCP建立需要经过三次握手过程:【第一个报文】：客户端会随机初始化序列号（client\_isn），将此序列号置于TCP首部的序列号字段中，同时将SYN标志位置为1，表示该报文为SYN报文。接下来就将第一个SYN报文发送给服务端，表示向服务端发起连接，该报文不包含应用层数据，之后客户端处于 SYN-SENT 状态。【第二个报文】：服务端收到客户端的SYN报文后，首先服务端也随机初始化自己的序列号（server\_isn），将此序号填入TCP首部的序号字段中，其次把TCP首部的确认应答号字段填入 client\_isn + 1，接着把 SYN 和 ACK 标志位置为1。最后把该报文发给客户端，该报文也不包含应用层数据，之后服务端处于 SYN-RCVD 状态。【第三个报文】：客户端收到服务端报文后，还要向服务端回应最后一个报文，首先应答报文 TCP 首部 ACK 标志位置为1，其次确认应答号字段填入 server\_isn + 1，最后把报文发送给服务端，这次报文可以携带客户端到服务器的数据，之后客户端处于 ESTABLISHED 状态[4]。

使用Wireshark对TCP建立过程进行抓包分析可得如图20结果：

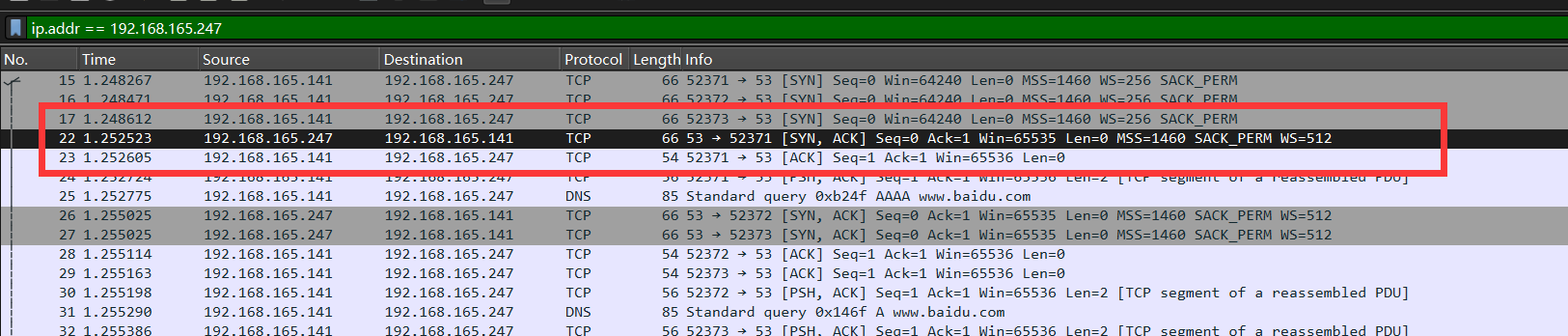


图20 TCP建立过程的三次握手

从图20可得出红框部分满足TCP建立过程的要求，故这部分是在进行客户端和服务器端之间的TCP连接。

(2)TCP释放

TCP连接的释放过程如图21所示

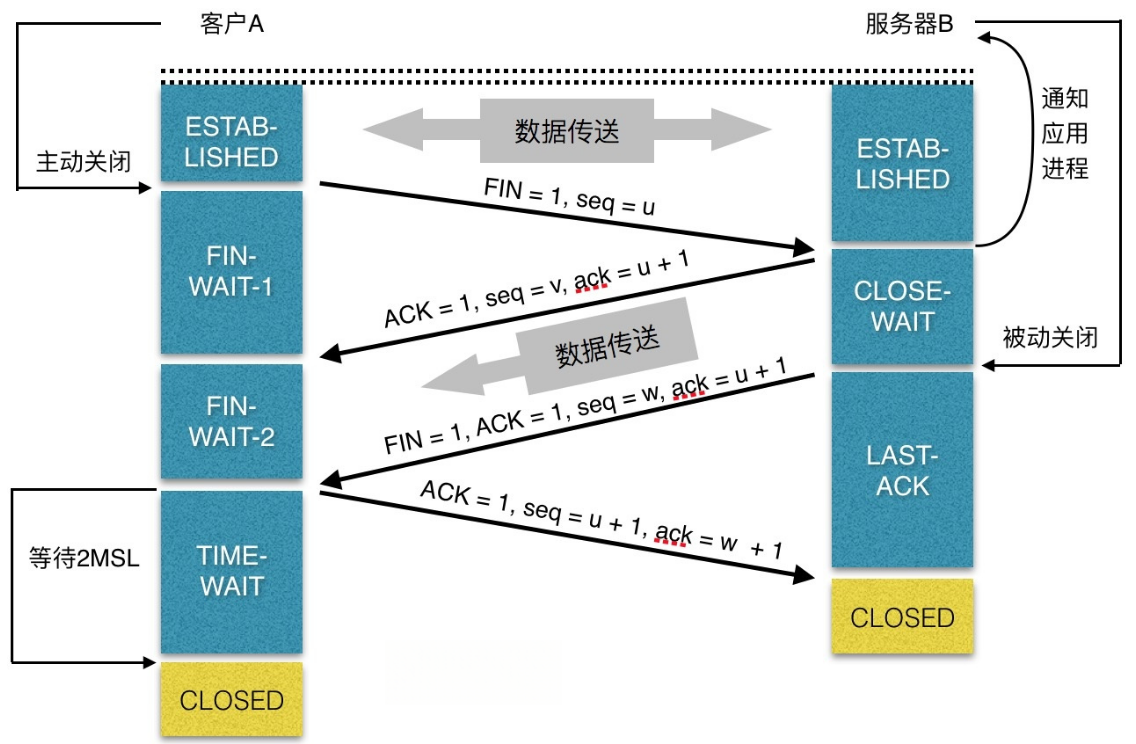


图21 TCP释放连接工作过程

从图21可知，TCP连接释放需要经历四次握手:【第一个报文】：客户端发送一个 FIN 报文，用来关闭客户端到服务端的数据传送，也就是客户端告诉被服务端：我已经不会再给你发数据了(当然，在 FIN 包之前发送出去的数据，如果没有收到对应的 ACK 报文，客户端依旧会重发这些数据)，但此时客户端还可以接受数据；【第二个报文】：服务端收到 FIN 报文后，发送一个 ACK 给对方，确认序号为收到序号 + 1，此时服务端进入 CLOSED\_WAIT 状态。客户端接收到 ACK 报文后，进入 FIN\_WAIT\_2 状态；【第三个报文】：服务端发送一个 FIN 报文，用来关闭被动关闭方到主动关闭方的数据传送，也就是告诉主动关闭方，我的数据也发送完了，不会再给你发数据了，接下来服务端进入 LAST\_ACK 状态；【第四个报文】：客户端收到 FIN 报文之后，发送一个 ACK 给服务端，确认应答号为收到序号 + 1，此时客户端进入 TIME\_WAIT 状态；服务端收到 FIN 报文后，就直接进入了 CLOSED 状态，连接资源将被释放，四次挥手到此结束；客户端在经过 2MSL 时间后，自动进入CLOSED 状态，至此客户端也完成了连接的关闭。

用Wireshark抓包可得到如图22所示的红框部分所示

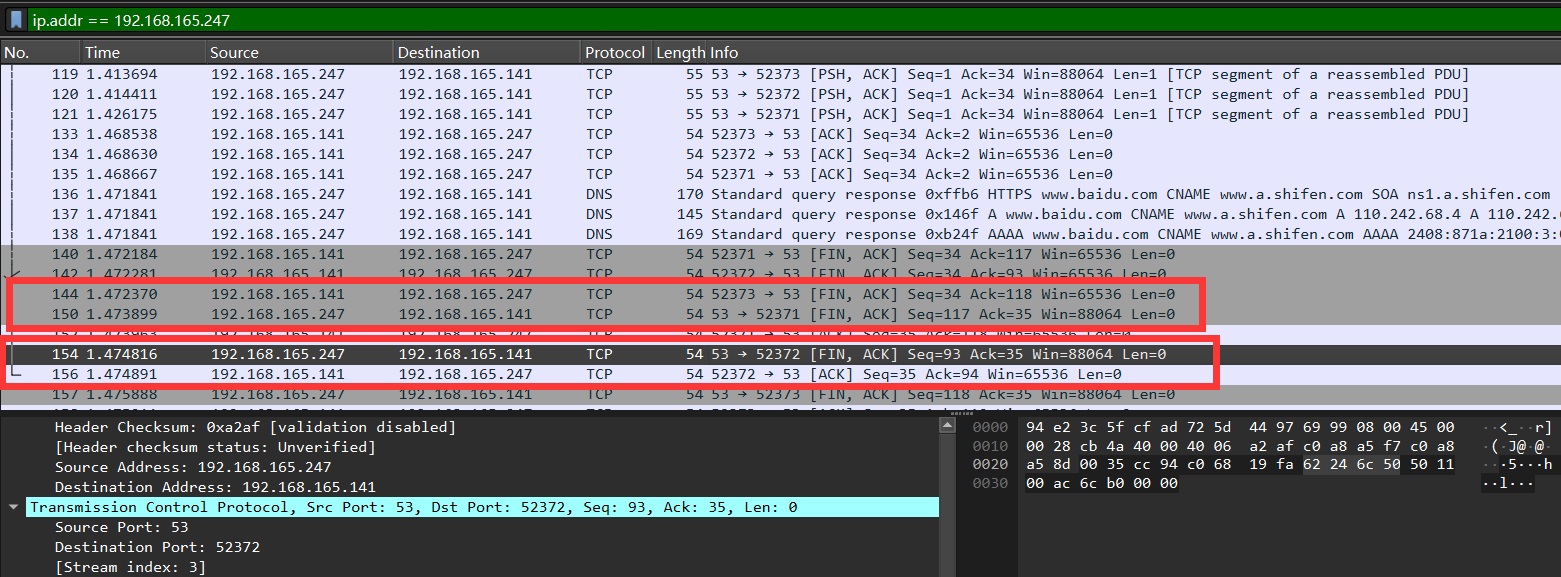


图22 TCP释放连接过程

## 3.7 UDP

### 3.7.1 UDP协议及其帧结构

UDP（UserDatagramProtocol）是一个简单的面向消息的传输层协议，尽管UDP提供标头和有效负载的完整性验证（通过校验和），但它不保证向上层协议提供消息传递，并且UDP层在发送后不会保留UDP 消息的状态[5]。因此，UDP有时被称为不可靠的数据报协议。如果需要传输可靠性，则必须在用户应用程序中实现。UDP使用具有最小协议机制的简单无连接通信模型。UDP提供数据完整性的校验和，以及用于在数据报的源和目标寻址不同函数的端口号。它没有握手对话，因此将用户的程序暴露在底层网络的任何不可靠的方面。如果在网络接口级别需要纠错功能，应用程序可以使用为此目的设计的传输控制协议（TCP）。UDP协议的帧格式如图23所示：



图23 UDP协议帧格式

UDP（用户数据报协议）是一种无连接的、轻量级的传输层协议，用于在计算机网络上通过IP（Internet Protocol）传输数据。UDP帧的结构相对简单，通常由四个字段组成，每个字段都有特定的含义。以下是UDP帧中每个字段的所代表的含义如下：

1.源端口 (Source Port):占用16 bits（2字节），用于 指定发送方的端口号。接收方使用源端口来确定从哪个应用程序或服务发送的数据。

2. 目标端口 (Destination Port):占用16 bits（2字节），用于指定接收方的端口号，帮助接收方将数据传递给正确的应用程序或服务。

3.长度 (Length):占用 16 bits（2字节），用于指定UDP数据报文的长度，包括头部和数据部分，接收方可以使用这个字段来确定接收到的UDP数据报文的实际长度。

4.校验和 (Checksum):占用 16 bits（2字节），用于检测UDP数据报文在传输过程中是否发生了错误，发送方在发送UDP数据报文之前计算校验和，并将其放置在这个字段中。接收方在接收数据时重新计算校验和，如果发现不匹配，就表明数据可能被损坏。

UDP是一种简单的协议，相对于TCP（传输控制协议），它没有连接建立和保持的开销，但也缺乏TCP提供的可靠性和流量控制机制。UDP通常用于实时应用，如音频和视频传输，以及一些不需要可靠性的应用，如DNS（域名系统）查询。

# 典型网络应用的工作原理及其实现流程

在互联网中，典型的网络应用有PPPoE、HTTP、FTP、SMTP等，本部分对这些典型的网络应用的工作原理及其实现流程进行分析。

## 4.1PPPoE工作过程

PPPoE(PPP over Ethernet)是在以太网上建立PPP连接，由于以太网技术十分成熟且使用广泛，而PPP协议在传统的拨号上网应用中显示出良好的可扩展性和优质的管理控制机制，二者结合而成的PPPoE协议得到了宽带接入运营商的认可并广为采用。

PPPoE建立过程可以分为Discovery阶段和PPP会话阶段。Discovery阶段是一个无状态的阶段，该阶段主要是选择接入服务器，确定所要建立的PPP会话标识符Session ID，同时获得对方点到点的连接信息；PPP会话阶段执行标准的PPP过程。

一个典型的Discovery阶段包括以下4个步骤：

(1)主机首先主动发送广播包PADI寻找接入服务器，PADI必须至少包含一个服务名称类型的TAG，以表明主机所要求提供的服务。

(2)接入服务器收到包后如果可以提供主机要求

(3)主机在回应PADO的接入服务器中选择一个合适的，并发送PADR告知接入服务器，PADR中必须声明向接入服务器请求的服务种类。

(4)接入服务器收到PADR包后开始为用户分配一个唯一的会话标识符Session ID，启动PPP状态机以准备开始PPP会话，并发送一个会话确认包PADS。

PPPoE不仅有以太网的快速简便的特点，同时还有PPP的强大功能，任何能被PPP封装的协议都可以通过PPPoE传输。

## 4.2 HTTP工作过程

HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从WWW服务器传输超文本到本地浏览器的传送协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

HTTP由请求和响应构成，是一个标准的客户端服务器模型（B/S）。HTTP协议永远都是客户端发起请求，服务器回送响应,如图24所示

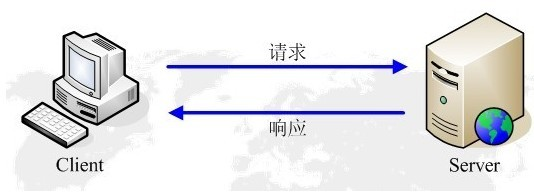


图24 HTTP请求相应模型图

一次HTTP操作称为一个事务，其工作整个过程如下：

1) 地址解析

如用客户端浏览器请求这个页面：http://localhost.com:8080/index.htm

从中分解出协议名、主机名、端口、对象路径等部分，对于我们的这个地址，解析得到的结果如下：协议名：http 主机名：localhost.com 端口：8080 对象路径：/index.htm

在这一步，需要域名系统DNS解析域名localhost.com,得主机的IP地址。

2）封装HTTP请求数据包

把以上部分结合本机自己的信息，封装成一个HTTP请求数据包

3）封装成TCP包，建立TCP连接（TCP的三次握手）

在HTTP工作开始之前，客户机（Web浏览器）首先要通过网络与服务器建立连接，该连接是通过TCP来完成的，该协议与IP协议共同构建Internet，即著名的TCP/IP协议族，因此Internet又被称作是TCP/IP网络。HTTP是比TCP更高层次的应用层协议，根据规则，只有低层协议建立之后才能，才能进行更层协议的连接，因此，首先要建立TCP连接，一般TCP连接的端口号是80。这里是8080端口

4）客户机发送请求命令

建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可内容。

5）服务器响应

服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

实体消息是服务器向浏览器发送头信息后，它会发送一个空白行来表示头信息的发送到此为结束，接着，它就以Content-Type应答头信息所描述的格式发送用户所请求的实际数据

6）服务器关闭TCP连接

一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接，然后如果浏览器或者服务器在其头信息加入了这行代码

Connection:keep-alive

TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。

## 4.3 FTP工作过程

FTP (File Transfer Protocol)在TCP/IP协议族中属于应用层协议，是文件传输标准。主要功能是向用户提供本地和远程主机之间的文件传输，尤其在进行版本升级、日志下载和配置保存筹业务操作时，广泛地使用FTP功能。FTP协议基于相应的文件系统实现。

FTP 工作流程如图25所示。

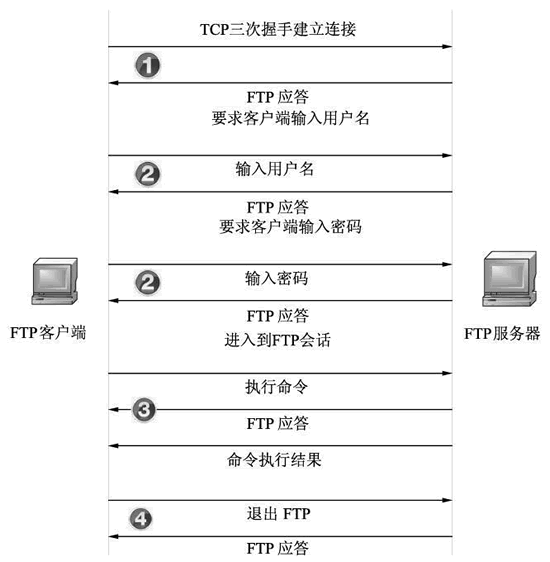


图25 FTP工作过程

## 4.4 SMTP工作过程

SMTP(Simple Mail Transfer Protocol)即简单邮件传输协议，是一种提供可靠且有效电子邮件传输的协议。SMTP是建立在FTP文件传输服务上的一种邮件服务，主要用于传输系统之间的邮件信息并提供与来信有关的通知。

SMTP是工作在两种情况下：一、电子邮件从客户端传输到服务器；二、从某一个服务器传输到另一个服务器。SMTP也是一个请求/响应协议，命令和响应都是基于ASCⅡ文本，并以CR和LF符结束。响应包括一个标识返回状态的三位数字代码。SMTP在TCP协议25号端口监听连续请求。

SMTP连结和发送过程如下：

（1）建立TCP连结。

（2）客户端发送HELO命令以标识发件人自己的身份，然后客户端发送MAIL命令；服务器端以OK作为响应，表明准备接收。

（3）客户端发送RCPT命令，以标识该电子邮件的计划接收人，可以有多个RCPT行；服务器端则表示是否愿意为收件人接收邮件。

（4）协商结束，发送邮件，用命令DATA发送。

（5）以“.”号表示结束输入内容并发送出去，结束此次发送，用QUIT命令退出。

## 4.5典型网络应用实现流程

通过对清华大学主页访问来说明以上典型网络应用的实现流程。

(1)首先使用主机进行拨号上网建立PPP连接,使用Wireshark抓包可得到图26的结果。

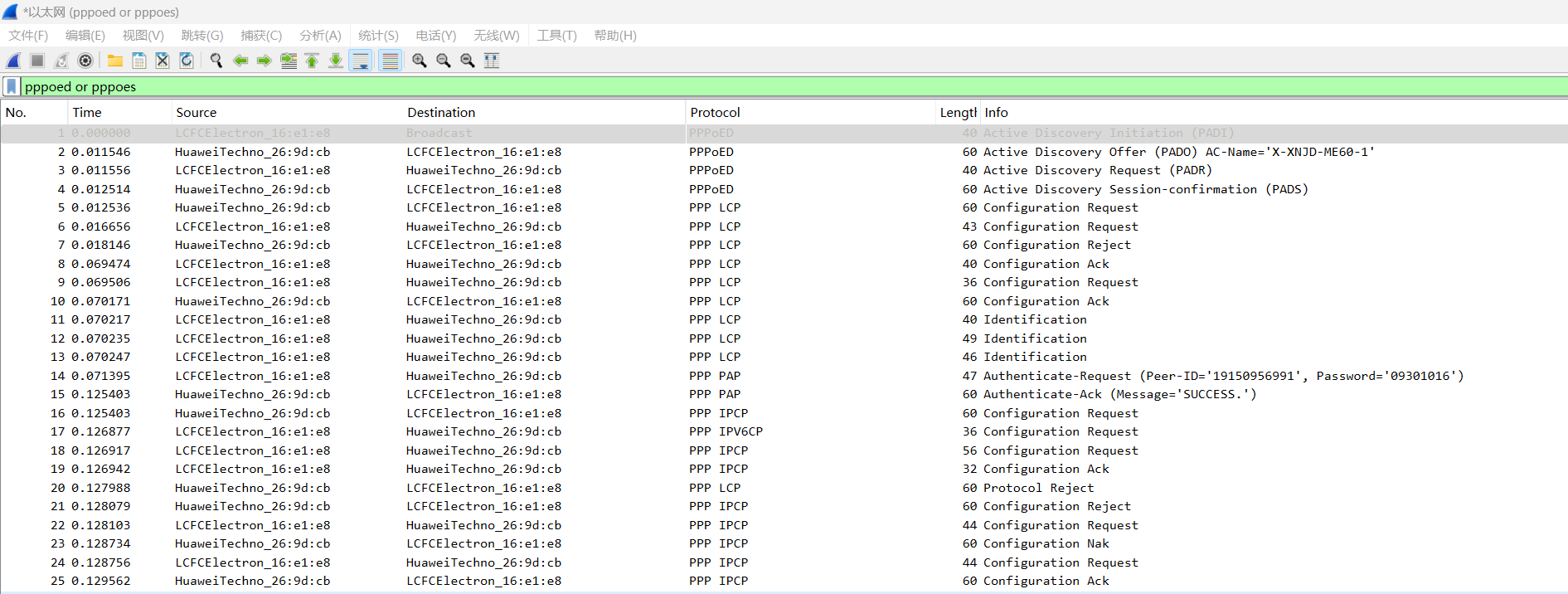


图26 Wireshark抓包PPPoE结果

(2)然后使用浏览器对清华主页进行访问，分五个层次对其进行分析。

应用层：

1.浏览器访问清华主页后，会向DNS服务器发起DNS请求，进行域名解析，获得相应的清华主页IP地址，使用Wireshark进行抓包结果如图27所示

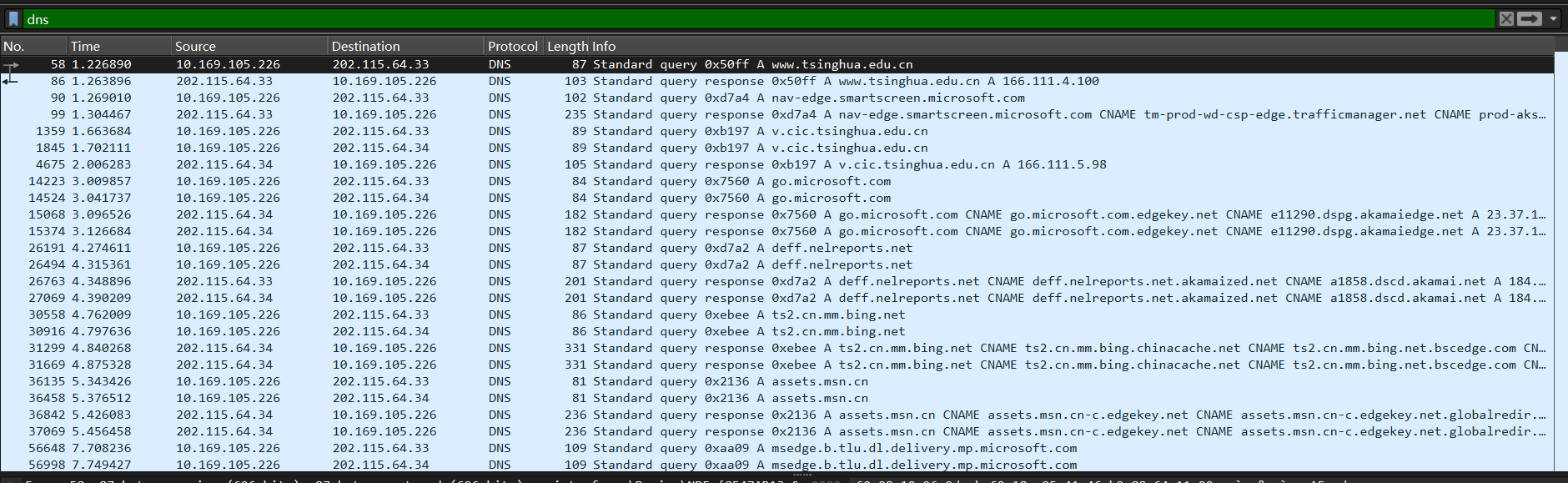


图27 Wireshark DNS抓包

2.根据HTTP协议生成HTTP报文请求

在应用层将要发送的数据内容形成了应用层的报文data,发送到传输层，使用Wireshark进行抓包结果如图28所示

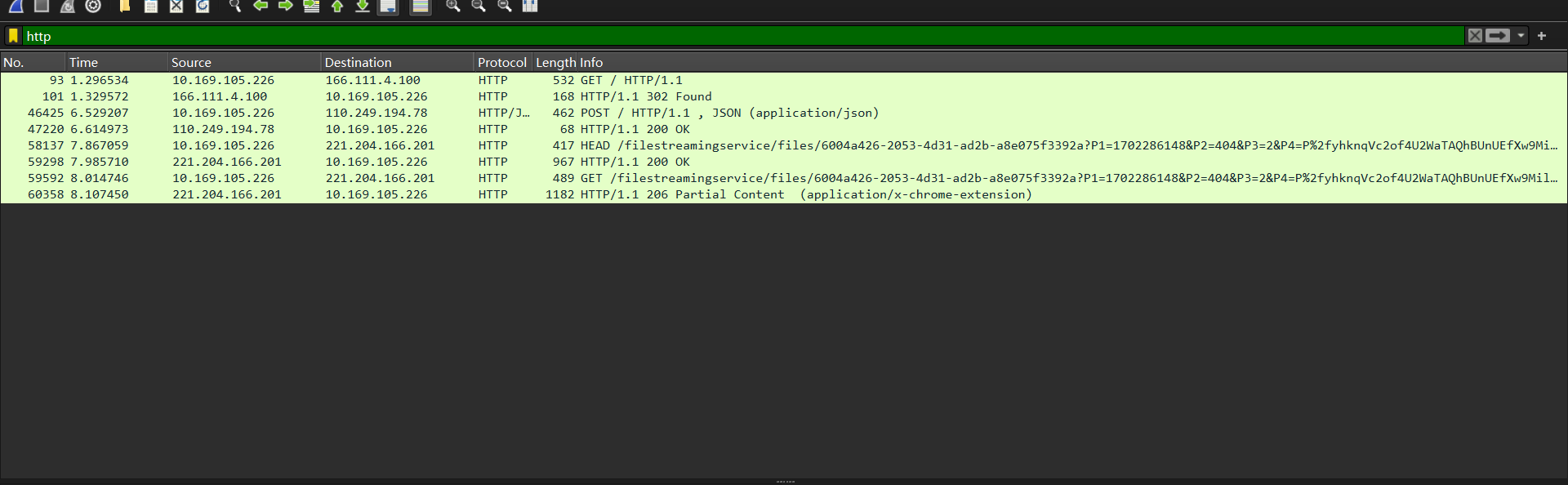


图28 Wireshark HTTP抓包

传输层：

3.TCP的三次握手

浏览器的访问进程会和相应的服务器进程建立TCP连接，其三次握手过程如图29所示

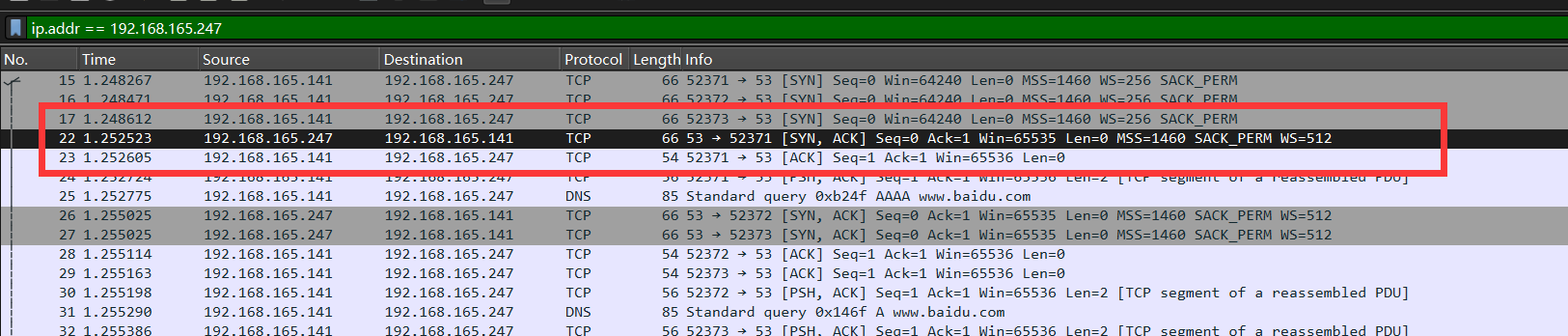


图29 TCP三次握手

然后把处理的数据段发送到网络层

网络层

4.网络层进行IP寻址，利用ARP协议获取MAC地址，利用Wireshark抓包可得图30结果

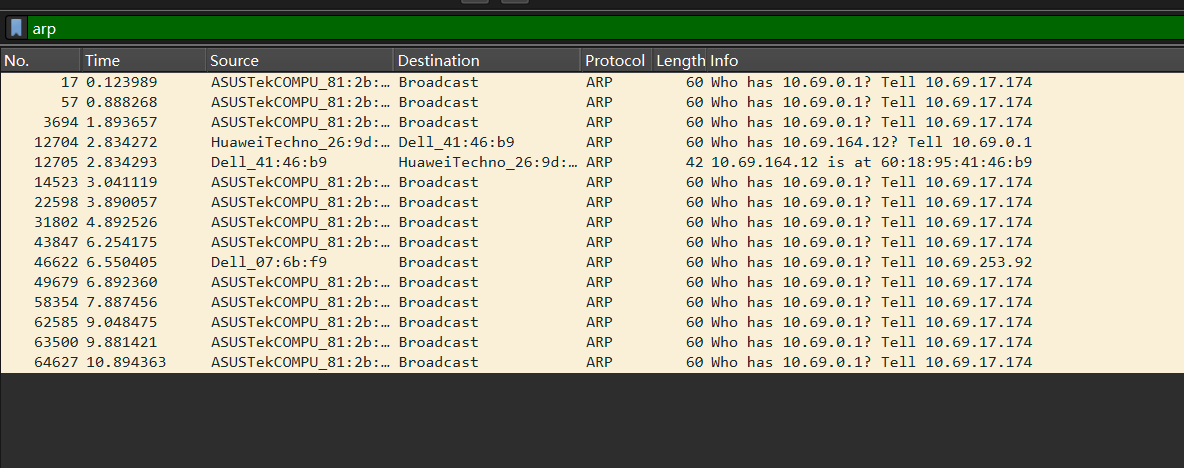


图30 Wireshark抓包ARP协议

数据在网络层被打包，这样封装上了网络层的包头，包头内部含有源及目的的ip地址

数据链路层

1. MAC寻址
2. 服务器接受请求
3. 然后服务端处理请求：服务端处理请求分为：

(1)首页请求:因为输入的url是请求进入网站首页的,不带任何参数请求,而且操作简单,这样下来QPS即每秒查询量是极大的,服务器需要在极短的时间内处理这些流量,这时候会用到CDN系统的缓存服务器将首页的图片迅速分发给用户。在网站和用户之间引入CDN之后，用户不会有任何与原来不同的感觉。使用CDN服务的网站，只需将其域名的解析权交给CDN的负载均衡设备，CDN负载均衡设备将为用户选择一台合适的缓存服务器，用户通过访问这台缓存服务器来获取自己所需的数据。用户可以以最短的路径，最快的速度对网站进行访问。因此，CDN可以加速用户访问速度，减少源站中心负载压力。

(2)其它请求:后台统一处理请求，处理完后响应结果.一般后端都是有统一的验证的，如安全拦截，跨域验证.如果这一步不符合规则，就直接返回了相应的http报文（如拒绝请求等）

然后当验证通过后，才会进入实际的后台代码，此时是程序接收到请求，然后执行（譬如查询数据库，大量计算等等）等程序执行完毕后，就会返回一个http响应包服务端提供响应8.服务端提供响应:服务端处理完请求后,会将所请求的东西响应给客户端,服务器会以同样的顺序同样的方式将响应数据包发送都客户端

9.对访问完成后，TCP建立释放进行四次挥手,使用Wireshark抓包可得图31所得结果。

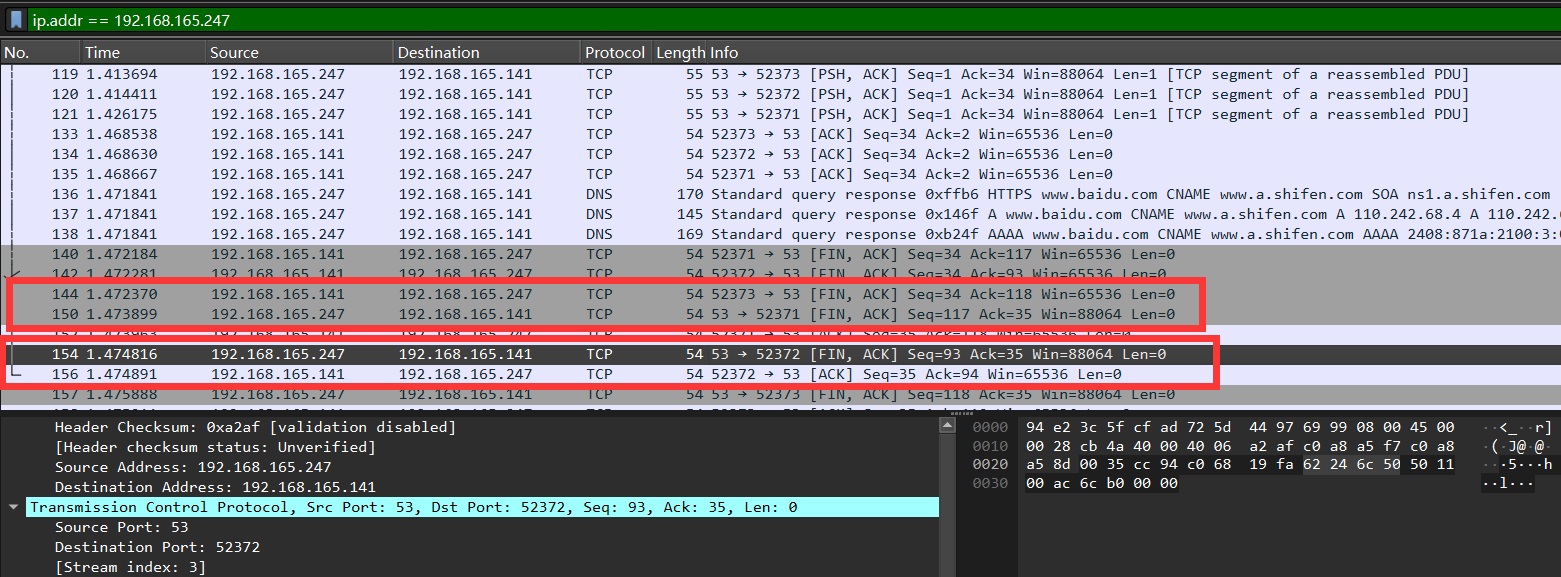


图31 TCP四次挥手

10.浏览器解析和渲染页面，展示给用户浏览器解析和渲染

(1)解析HTML,构建DOM树

HTML文档会被解析成一棵以document为根的DOM树，解析过程中如果遇到JavaScript，则会暂停解析并传输下载相应的文件造成阻塞，故推荐将JavaScript脚本放在HTML文件的后面。

(2)构建CSSSOM树

浏览器根据外部样式，内部样式和内联样式来解析CSS，构建CSSSOM树。

(3)构建渲染树和布局

DOM树和CSSOM树构建完毕后会融合成渲染树，然后浏览器会确认页面各元素的位置。

(4)页面绘制和优化

浏览器根据布局结果进行页面的绘制，并优化页面内容，减小CPU消耗。

12.交互结束

# 心得体会

在本次课程设计中，我深入学习了网络通信中使用的协议和相关的工作流程。这次实验让我更好地理解了网络通信的基本原理，对于网络协议的应用和配合工作流程有了更为清晰的认识。

首先，我对常见的网络协议进行了深入了解，包括但不限于TCP、UDP、HTTP等。了解每种协议的特点、用途和工作方式是理解网络通信的关键。通过Wireshark抓包分析，我能够直观地看到这些协议在数据包中的表现，更深入地了解它们的运作机制。

TCP作为一种面向连接的协议，通过三次握手建立连接，提供可靠的数据传输。UDP则是一种无连接的协议，适用于实时性要求较高的场景。HTTP协议负责超文本传输，在Web应用中起到至关重要的作用。通过这些协议，网络应用能够在不同层次上实现各种功能。

Wireshark是一款功能强大的网络抓包工具，通过它我能够实时捕获和分析网络数据包。在实验中，我使用Wireshark捕获了拨号方式登录清华大学主页的数据包，并进行了深入的分析。通过过滤器，我能够只关注与清华大学主页相关的数据包，更容易地找到并分析与该过程有关的通信。透过Wireshark，我清晰地看到了TCP握手的过程、HTTP请求和响应的内容。这让我对拨号登录服务清华大学主页的网络通信流程有了直观的了解。Wireshark的使用不仅加深了我对协议工作原理的理解，也提升了我的网络分析能力。在实验中，我发现网络通信的工作流程是相对清晰而有序的。首先，通过拨号或其他连接方式建立物理连接，获取IP地址。接着，通过TCP连接，进行HTTP请求和响应。整个过程中，每个环节都有特定的协议负责，各个协议之间紧密配合，确保数据的可靠传输和正确处理。

这次实验让我深刻认识到网络协议的重要性，它们是网络通信的基石，决定着数据如何在网络中传递和交换。了解网络通信的工作流程，不仅有助于问题排查和网络优化，也对网络安全具有重要意义。通过这次网络协议及流程分析的实验，我不仅学到了理论知识，还通过实际操作更深入地理解了网络通信的运作机制。Wireshark的使用为我提供了一个直观的工具，帮助我分析和理解网络数据包。同时，我也意识到网络协议的研究和分析是网络工程和安全领域的基础，对于日常的网络管理和问题排查都具有重要价值。

在未来，我将继续深入学习和研究网络协议，不断提升网络分析和优化的能力。理解网络通信的底层原理，将对我未来的学习和职业发展产生积极的影响。这次实验为我打开了网络通信的一扇新门，让我更加熟悉和自信地面对网络相关的问题和挑战。

# 参考文献

1. 杨奇儒.PPPoE的实现与优化[D].电子科技大学,2006.
2. 黄皓.ARP协议分析及在网络安全中的应用实现[D].电子科技大学,2007.
3. 徐伟. TCP协议的性能建模研究[D].中国科学技术大学,2013.
4. 谢希仁，*计算机网络（第6版）*，电子工业出版社，2013年06月
5. 黄文杰.基于UDP的可靠高效数据传输协议的研究[D].北京邮电大学,2019.