**南京航空航天大学**

计算机科学与技术学院/软件学院

**实验设计报告**

|  |  |
| --- | --- |
| 课程名称： | 计算机网络 |
| 学 号： | 162230217 |
| 姓 名： | 陈梓鹏 |
| 专 业： | 软件工程 |

二〇二四年十月

目录

[一、 实验内容 3](#_Toc181102125)

[二、 实验分析和实验设计 3](#_Toc181102126)

[可能应用的场景 3](#_Toc181102127)

[三、 实验实现及实验运行结果 4](#_Toc181102128)

[1.无线局域网 4](#_Toc181102129)

[1.1以本机为源的无线局域网单播帧、以本机为目的地的无线局域网单播帧 4](#_Toc181102130)

[2.IP分组 6](#_Toc181102131)

[2.1以本机为源的IP分组 6](#_Toc181102132)

[2.2以本机为目的地的IP分组 6](#_Toc181102133)

[3.Ping数据 7](#_Toc181102134)

[3.1以本机为源的Ping数据 8](#_Toc181102135)

[3.2以本机为目的地的Ping数据 8](#_Toc181102136)

[4.TCP报文 8](#_Toc181102137)

[4.1以本机为源的TCP报文 9](#_Toc181102138)

[4.2以本机为目的地的TCP报文 9](#_Toc181102139)

[5.UDP报文 10](#_Toc181102140)

[5.1以本机为源的UDP报文 10](#_Toc181102141)

[5.2以本机为目的地的UDP报文 11](#_Toc181102142)

[6.HTTP 11](#_Toc181102143)

[6.1以本机为源的HTTP请求 11](#_Toc181102144)

[6.1以本机为目的地的HTTP应答 12](#_Toc181102145)

[四、 实验总结和心得 13](#_Toc181102146)

# 实验内容

WireShark实验，利用WireShark按不同要求捕获相应数据。包括以下要求：

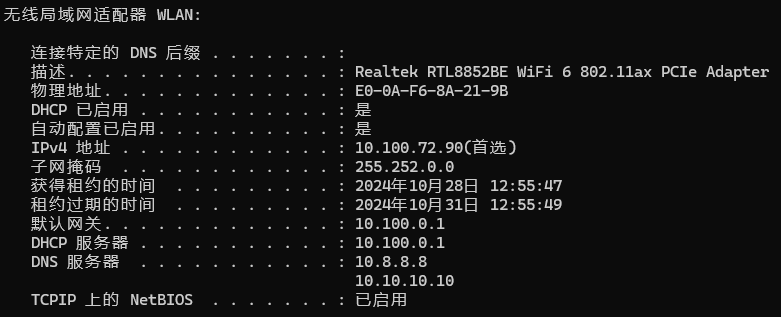
1. 以本机为源的无线局域网单播帧；
2. 以本机为目的地的无线局域网单播帧；
3. 以本机为源的IP分组；
4. 以本机为目的地的IP分组；
5. 以本机为源的Ping数据；
6. 以本机为目的的Ping的应答。
7. 以本机为源的TCP报文；
8. 以本机为目的地的TCP报文；
9. 以本机为源的UDP报文；
10. 以本机为目的地的UDP报文；
11. 以本机为源的HTTP请求；
12. 以本机为目的的HTTP应答。

# 实验分析和实验设计

## 可能应用的场景

1. **网络故障排查**：可以实时监控网络流量，分析延迟、丢包率等，帮助识别网络瓶颈或配置问题，特别适用于检测网络连接不稳定的情况。
2. **协议分析**：Wireshark支持多种网络协议解析，可以深入分析特定协议（如HTTP、TCP、DNS等）的数据流，定位协议层面的问题，比如HTTP 404错误的具体原因、DNS解析延迟等。
3. **安全漏洞检测**：通过监控网络数据包，可以发现异常流量，帮助识别潜在的网络攻击行为，如DDoS攻击、ARP欺骗等，适合网络安全工程师用于威胁检测。
4. **数据包追踪**：Wireshark可以帮助追踪数据在网络中的传输路径，尤其适用于调试多台设备间的数据通信，适用于网络管理员检查数据是否沿正确路径传输。

# 实验实现及实验运行结果

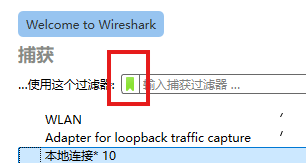


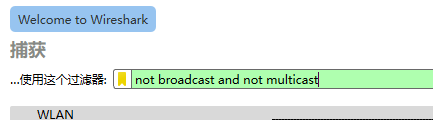
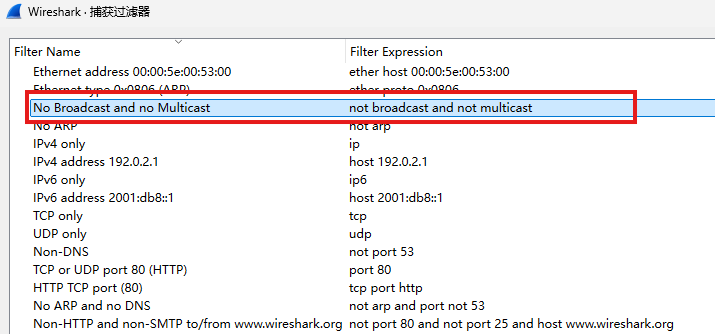
本机IP地址、MAC地址

## 1.无线局域网

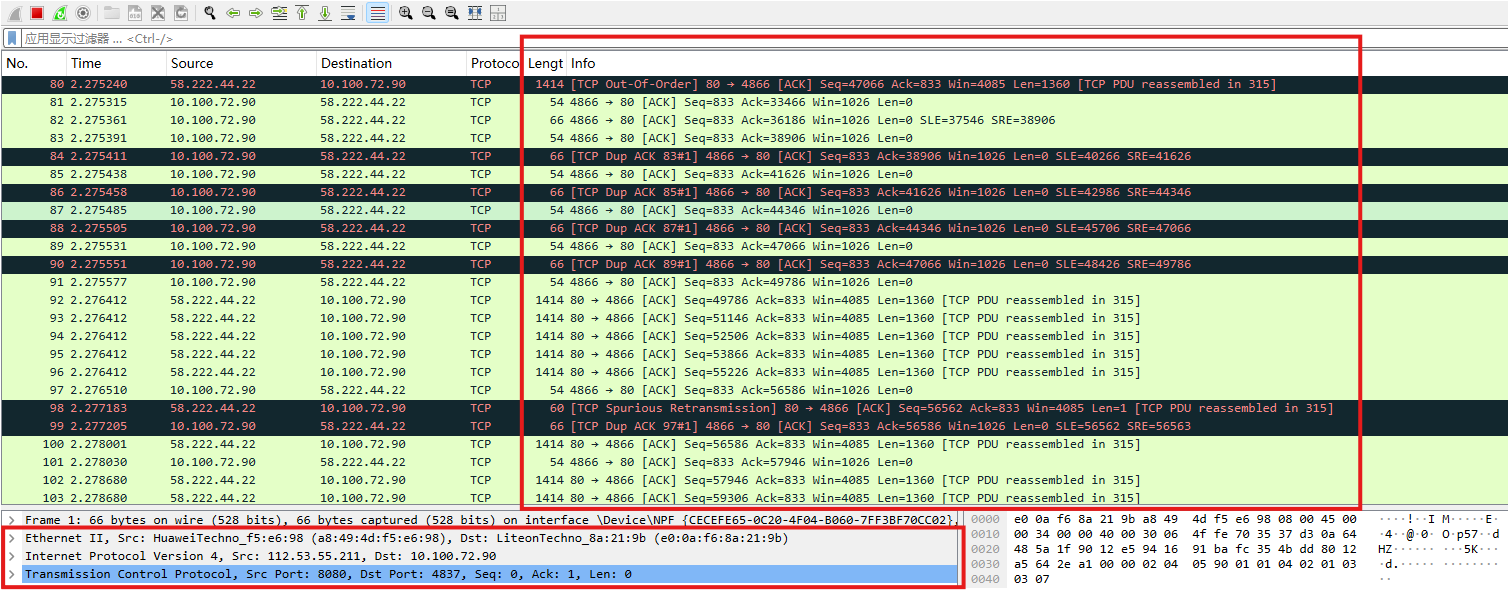
### 1.1以本机为源的无线局域网单播帧、以本机为目的地的无线局域网单播帧

打开wireshark，首先需要选择捕获过滤器，同时选择WLAN进行捕获，操作步骤如下所示

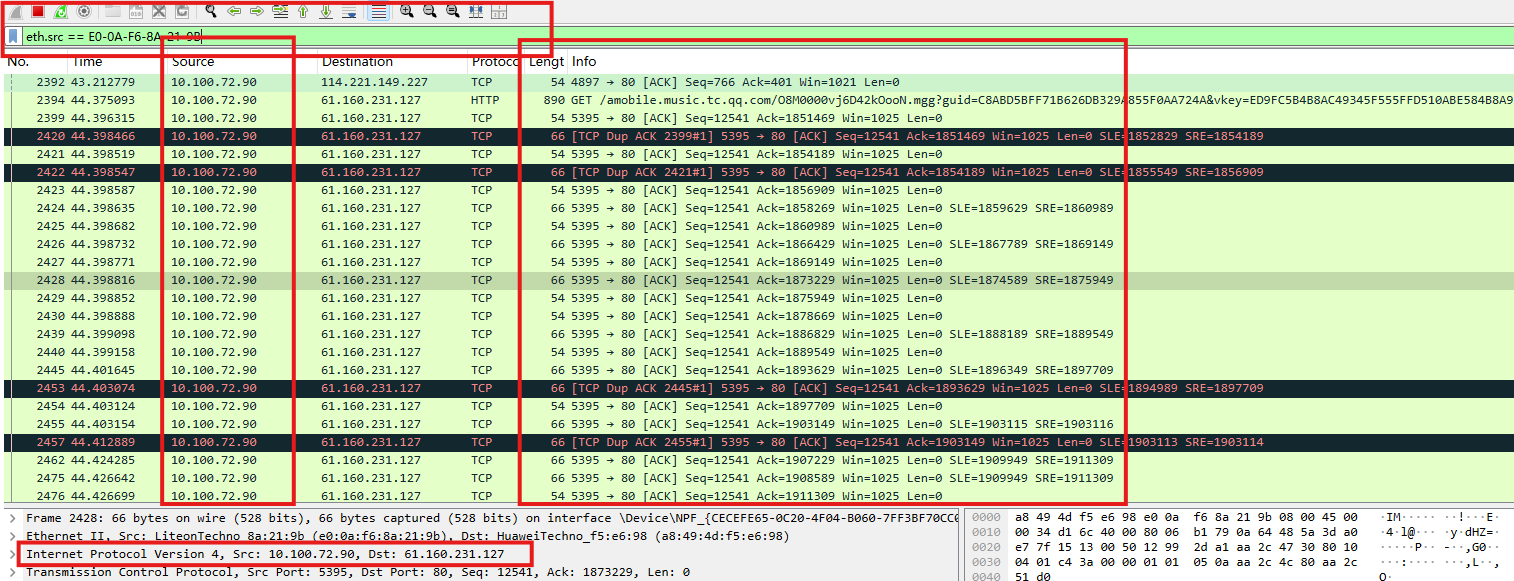




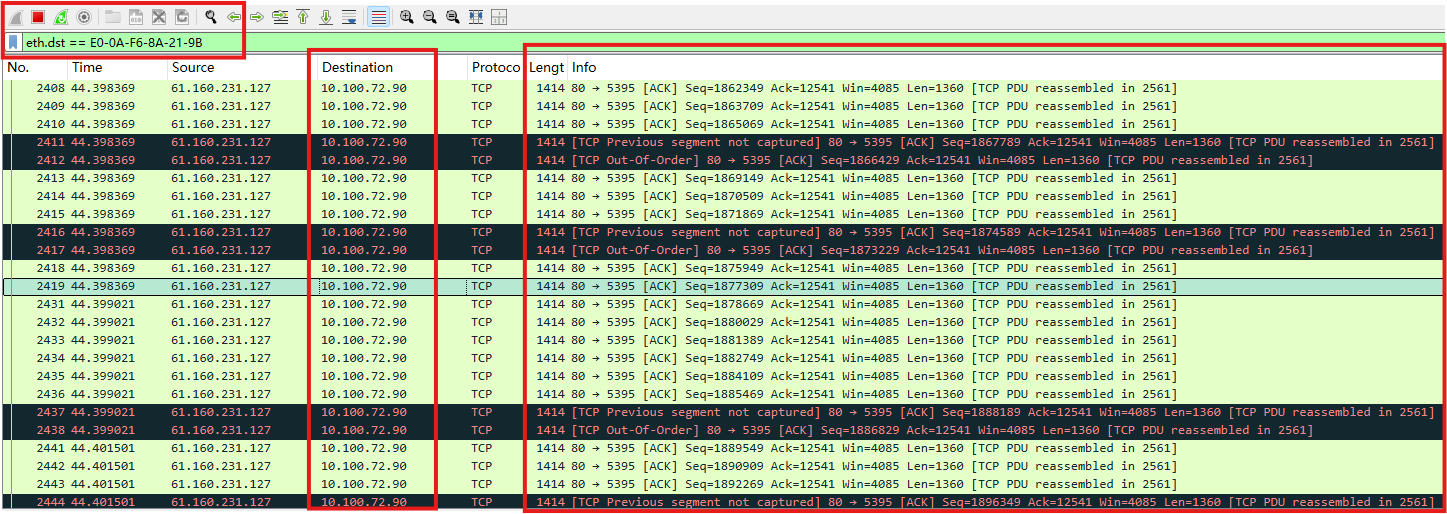
我们可以观察到捕获到的即为无线局域网单播帧，接下来我们即可以获得相应的单播帧。



在抓包过滤框中直接输入eth.src == <本机MAC地址>即可获得以本机为源的无线局域网单播帧。

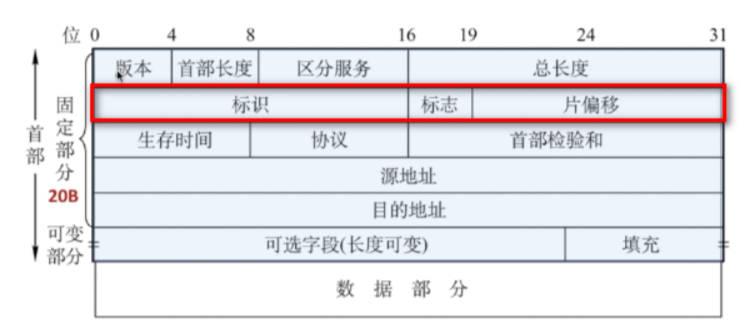


在抓包过滤框中直接输入eth.src == <本机MAC地址>即可获得以本机为源的无线局域网单播帧。



## 2.IP分组

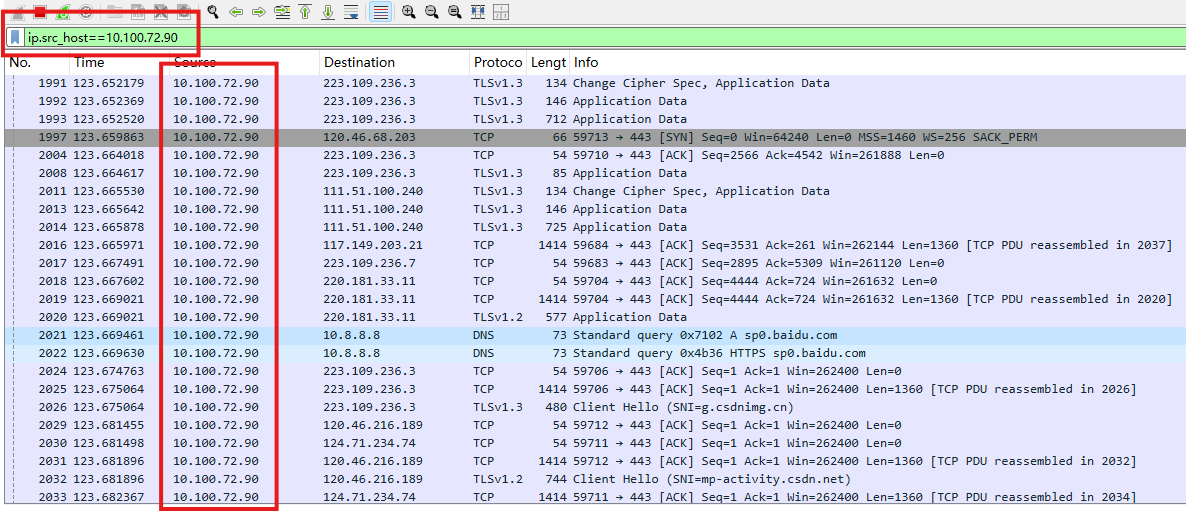
IP（Internet Protocol，网际协议）分组，也叫“数据报”，是在Internet网络中传输数据的基本单位。它是IP协议中传输数据的基本单位，包括IP头部和IP数据。



IP分组示意图

### 2.1以本机为源的IP分组

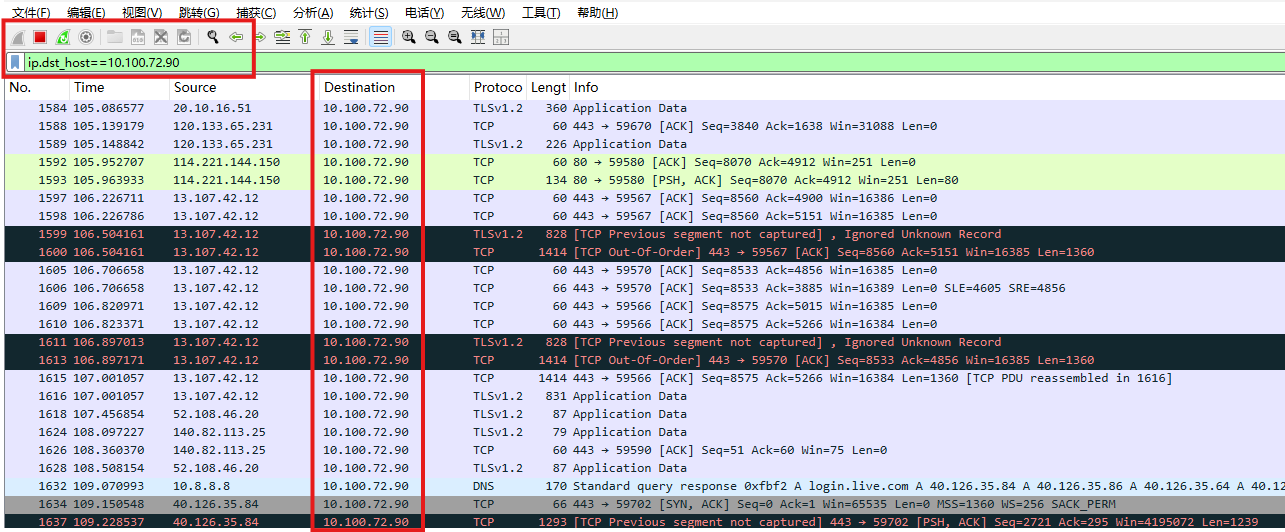
在抓包过滤框中直接输入ip.src\_host == <本机IP地址>即可。



以本机为源的IP分组

### 2.2以本机为目的地的IP分组

在抓包过滤框中直接输入ip.dst\_host== <本机IP地址>即可。



以本机为目的的IP分组

## 3.Ping数据

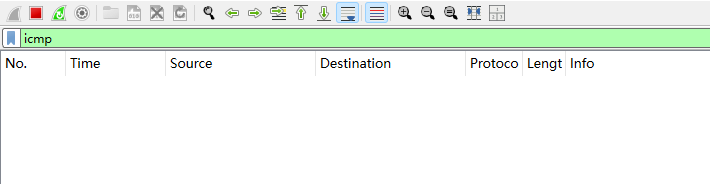
Ping是一句DOS 命令，一般用于检测网络通与不通 ，也叫时延，其值越大，速度越慢 PING (Packet Internet Grope)，因特网包探索器，用于测试网络连接量的程序。Ping 发送一个 ICMP（将在下面详细介绍） 回声请求消息给目的地并报告是否收到所希望的 ICMP 回声应答。

它是用来检查网络是否通畅或者网络连接速度的命令。作为一个生活在网络上的管理员或 者黑客来说，ping 命令是第一个必须掌握的 DOS 命令，它所利用的原理是这样的：网络上 的机器都有唯一确定的 IP 地址，我们给目标 IP 地址发送一个数据包，对方就要返回一个同 样大小的数据包，根据返回的数据包我们可以确定目标主机的存在，可以初步判断目标主机 的操作系统等。

Ping是Windows 系列自带的一个可执行命令。利用它可以检查网络是否能够连通，用 好它可以很好地帮助我们分析判定网络故障。应用格式：Ping IP 地址。该命令还可以加许多参数使用，具体是键入 Ping 按回车即可看到详细说明。

ICMP (Internet Control Message Protocol) Internet控制报文协议，它是TCP/IP协议族的一个子协议，用于在IP主机、路由器之间传递控制消息(即提供TCP/IP网络上设备、服务协议及路由器可用性的信息)，ICMP允许主机或路由器报告差错情况和提供有关异常情况的报告，目的是为了更有效地转发IP数据报和提高交付成功的机会。

在Ping之前，我们可以看到无法获取到任何ICMP，我们需要手动使用Ping命令。



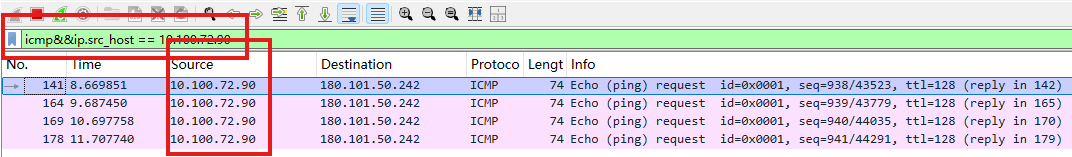
打开cmd，ping [www.baidu.com](http://www.baidu.com),即可获得ICMP报文



cmd操作指令

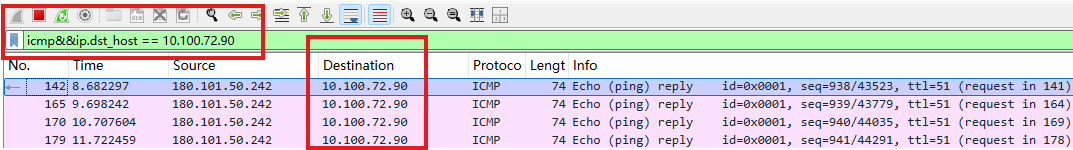
### 3.1以本机为源的Ping数据

在抓包过滤框中直接输入icmp&&ip.src\_host == <本机IP地址>即可。



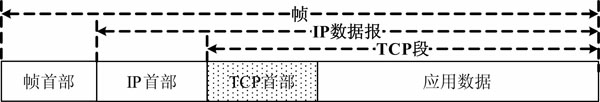
### 3.2以本机为目的地的Ping数据

在抓包过滤框中直接输入icmp&&ip.dst\_host == <本机IP地址>即可。

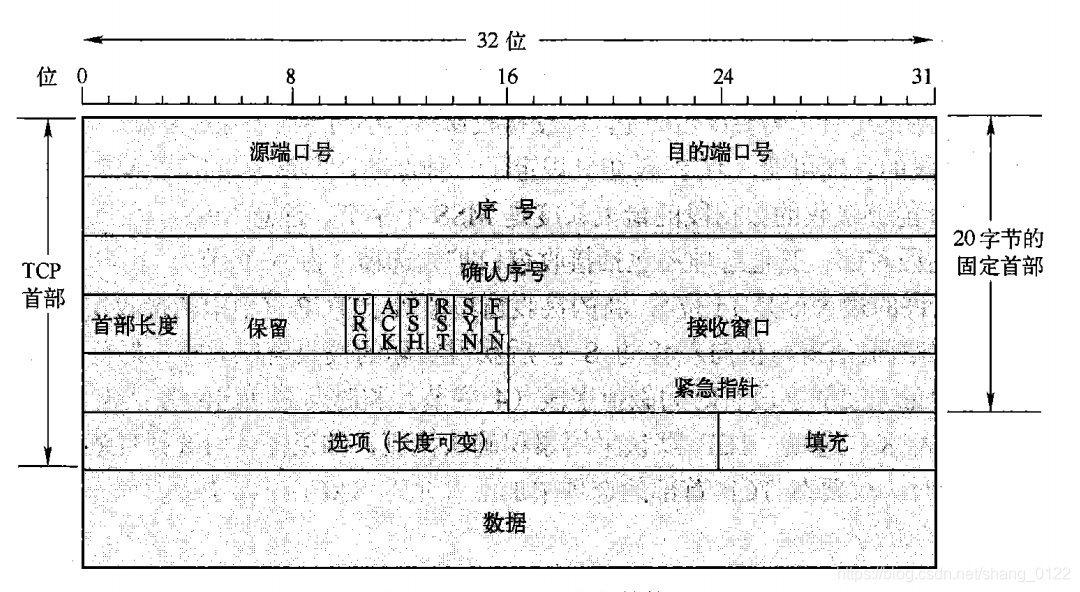


## 4.TCP报文

位于传输层的TCP数据分组称为段（Segment），又译为报文段、数据段或分段。TCP将来自应用层的数据分块并封装成TCP段进行发送。TCP段封装在IP数据报中，然后再封装成数据链路层中的帧，如下图所示:

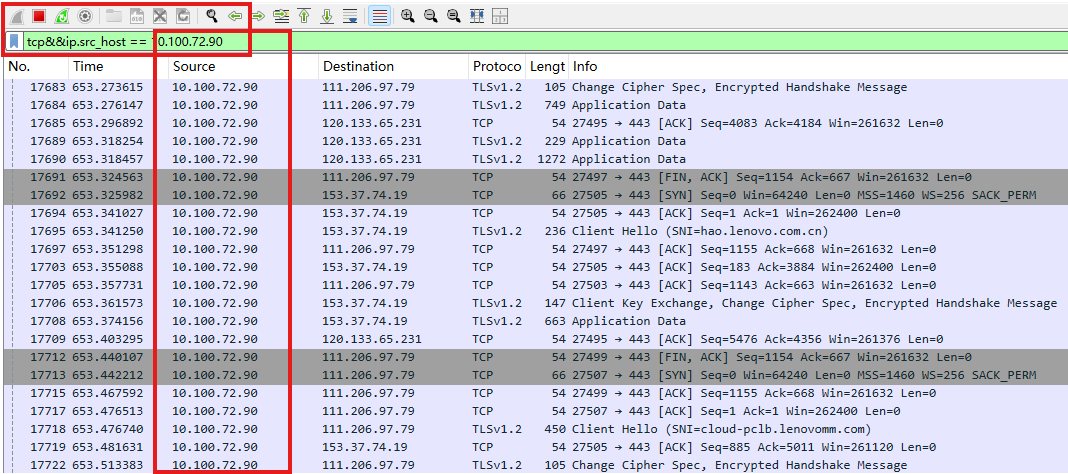


**TCP段=TCP首部+应用数据**



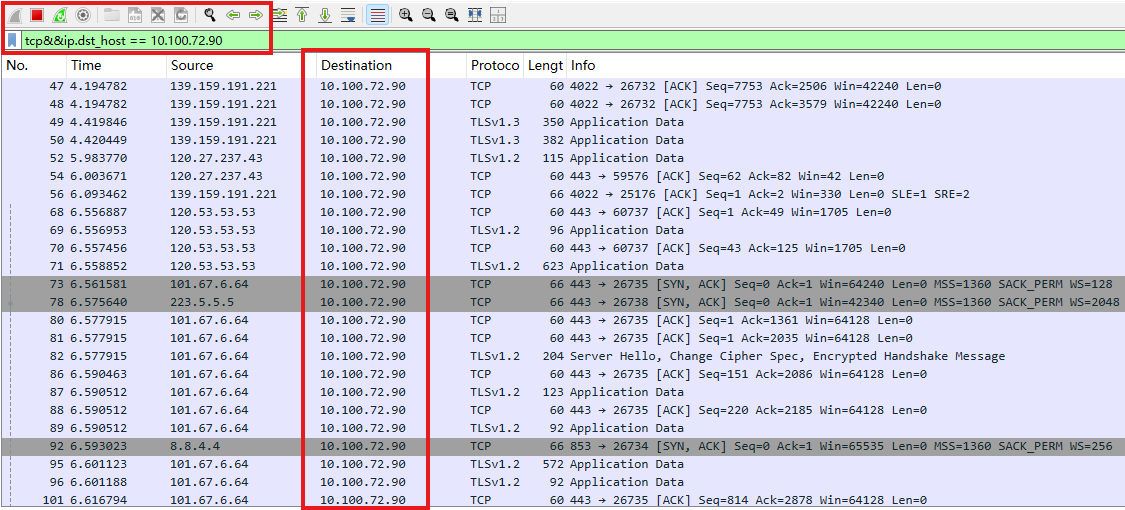
### 4.1以本机为源的TCP报文

直接在Filter框中直接输入协议名即可，同时加上ip.src\_host ==<本机ip地址>。**注意：协议名称需要输入小写。**



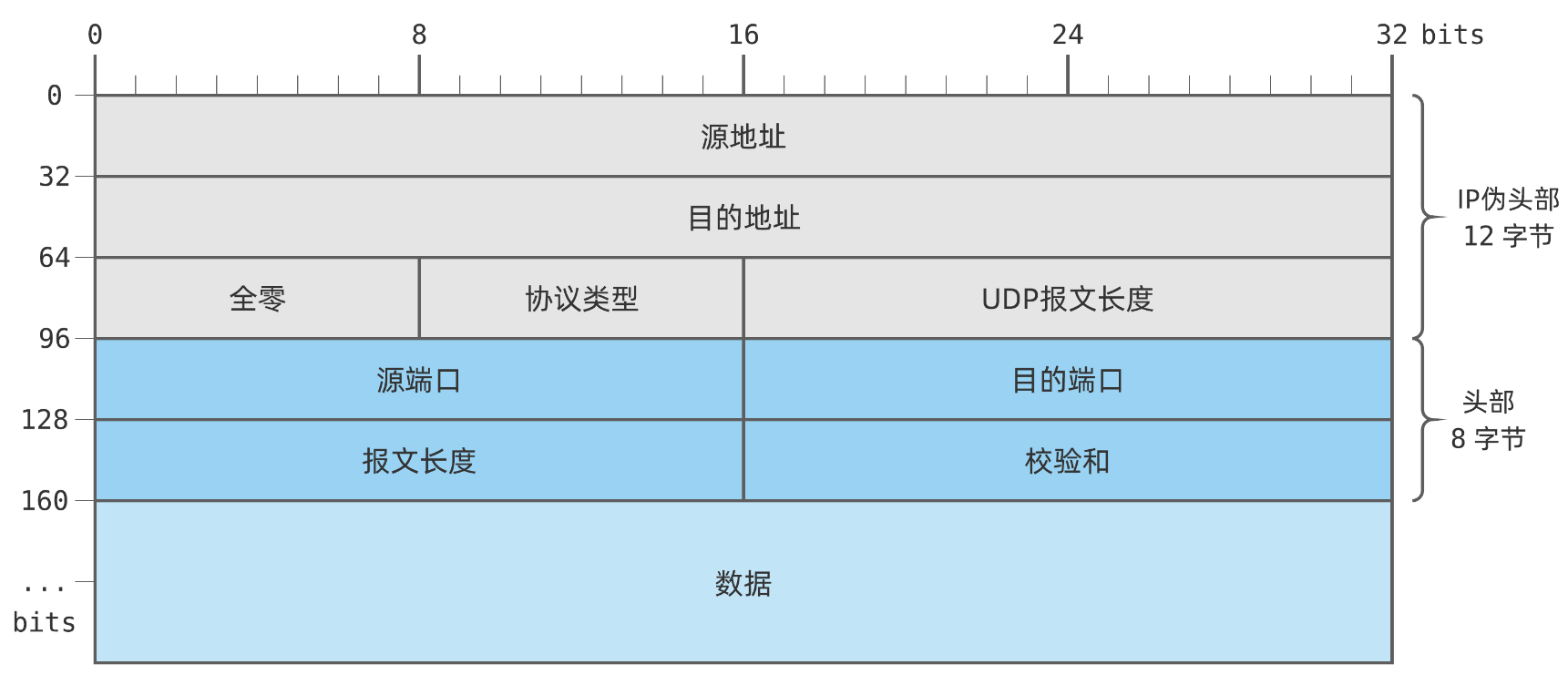
### 4.2以本机为目的地的TCP报文

直接在Filter框中直接输入协议名即可，同时加上ip.dst\_host ==<本机ip地址>。**注意：协议名称需要输入小写。**



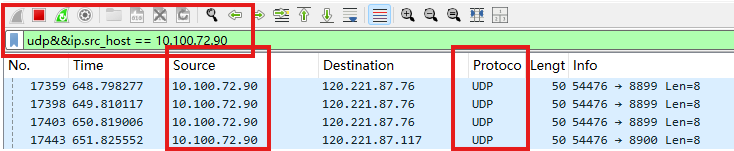
## 5.UDP报文

UDP 是 用户数据包协议 （ *user datagram protocol* ）的简称，它是一种简单的数据报式传输层协议。UDP 数据报结构非常简单，头部只包含端口号等若干个字段。由于 UDP 位于传输层，因而报文有时也称为 UDP 段或 UDP 分组，UDP 报文也分为头部和数据两个部分，如下图所示：



### 5.1以本机为源的UDP报文

直接在Filter框中直接输入udp&&ip.src\_host ==<本机ip地址>即可。



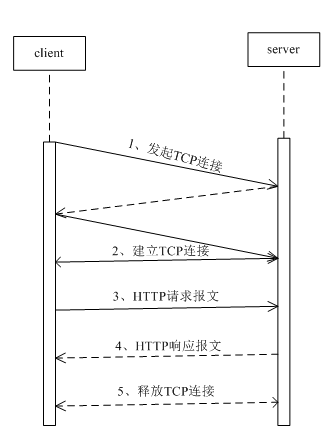
### 5.2以本机为目的地的UDP报文

直接在Filter框中直接输入udp&&ip.dst\_host ==<本机ip地址>即可。



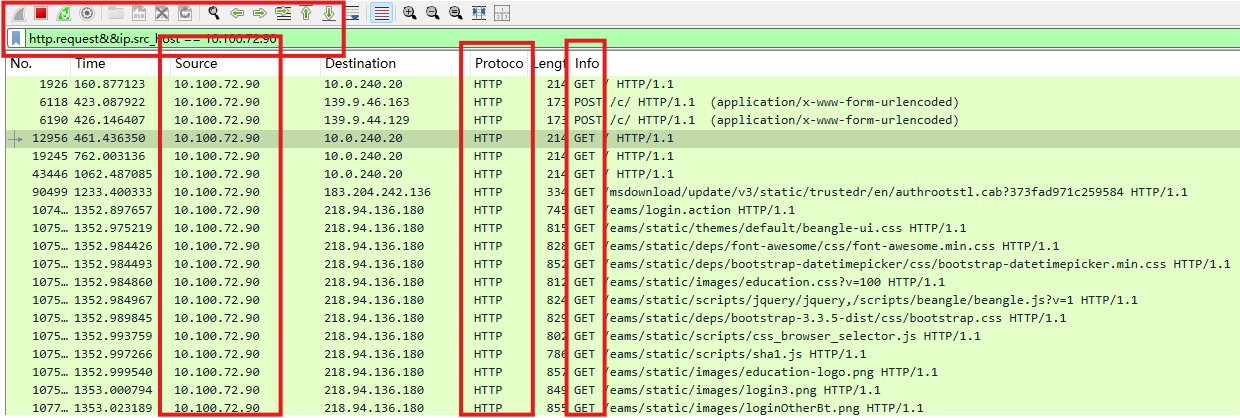
## 6.HTTP

HTTP是一个**无状态**的协议。所谓的无状态指的是客户端**（Web浏览器）**和服务器之间不需要建立持久的连接。这也就意味着当一个客户端向服务器发出请求，然后服务器返回响应之后，连接也就关闭了。服务器并不会保留连接的相关信息，HTTP遵循的是**请求（Request）/应答（Response）**模型。客户端（Web浏览器）向服务器发送请求，服务器处理请求并返回适当的应答。所有的HTTP连接都被构造成一套请求和应答。在这个过程中要经历4个阶段，包括建立连接、发送请求信息、发送响应信息和关闭连接，如下图所示：



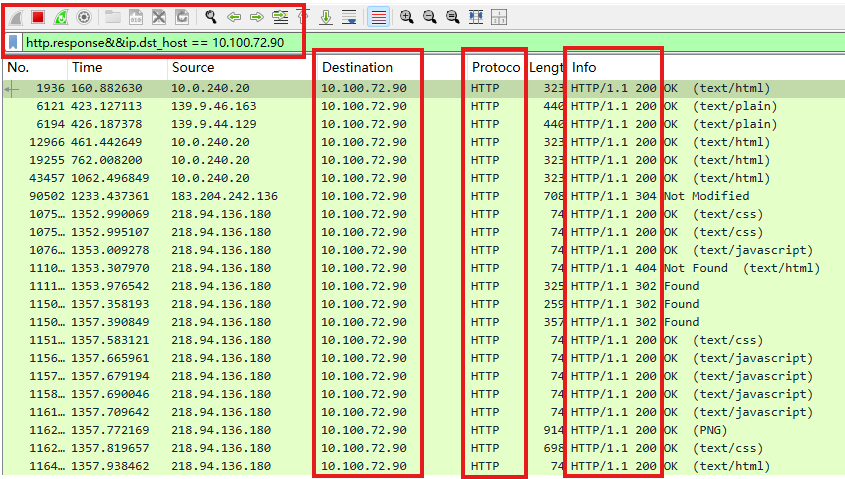
### 6.1以本机为源的HTTP请求

直接在Filter框中直接输入http.request&&ip.src\_host ==<本机ip地址>即可。



### 6.1以本机为目的地的HTTP应答

直接在Filter框中直接输入http.request&&ip.dst\_host ==<本机ip地址>即可。



# 实验总结和心得

这次WireShark实验真是让我大开眼界！在捕获不同类型的数据包时，几乎是一步步打开了网络的“黑箱”，亲眼见证了数据在无线局域网中的流动，确实让人感到十分神奇。

实验里需要捕获各式各样的数据包，从简单的单播帧、IP分组，到Ping包、TCP和UDP数据包，再到应用层的HTTP请求和应答，一开始看着挺复杂的，但上手操作后，我发现每一种数据包其实各有特色。通过抓取这些不同的数据包，我能更清楚地理解它们的特点和用处，比如TCP协议中三次握手的过程、UDP的快速传输方式、还有HTTP请求与应答之间的对话流等等。

在ARP请求的实验中，花了一整天的时间去做，结果一直无法成功，最后查到了资料只有一个设备是无法完成的，最后也是反馈给了老师，虽然没成功做出来，但也是收获非常多，对ARP有了更深的认识。

在使用WireShark的过程中，我也学会了用各种过滤器来锁定需要的数据包，这极大地提高了效率。WireShark真是个厉害的工具，操作起来就像在网络世界里“显微镜”一样，放大了各种平时看不到的细节，让我对网络通信的流程有了更深的理解。

总的来说，这次实验不仅是学到了技能，更让我对网络安全有了更强的意识。能看到数据流的来龙去脉，也让我意识到未经授权的网络监控会带来安全风险，必须在合法合规的范围内使用这些工具。