

华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2022

上机实践成绩：

指导教师：张召

姓名：郭夏辉

学号：10211900416

上机实践名称：实验课程简介和准备 1

上机实践日期：

上机实践编号：01

组号：

上机实践时间：2023 年 3 月 3 日

一、实验目的

1. 熟悉 Java 开发环境
2. 熟悉抓包软件 WireShark

二、实验任务

1. 配置 Java 开发环境
2. 编写并运行简单的 Java 程序
3. Wireshark 安装与网络包抓取

三、使用环境

1. Java 开发环境: IntelliJ IDEA 2022.3.2
2. 网络抓包工具 WireShark: Version 4.0.3 (v4.0.3-0-gc552f74cdc23).

四、实验过程

1. 创建一个以自己学号+姓名拼音命名的 java 项目



注意项目名称必须以字母或者“_”开头(类似于编程的变量名规则)

2. 编写 java 代码将自己的学号和姓名打印出来

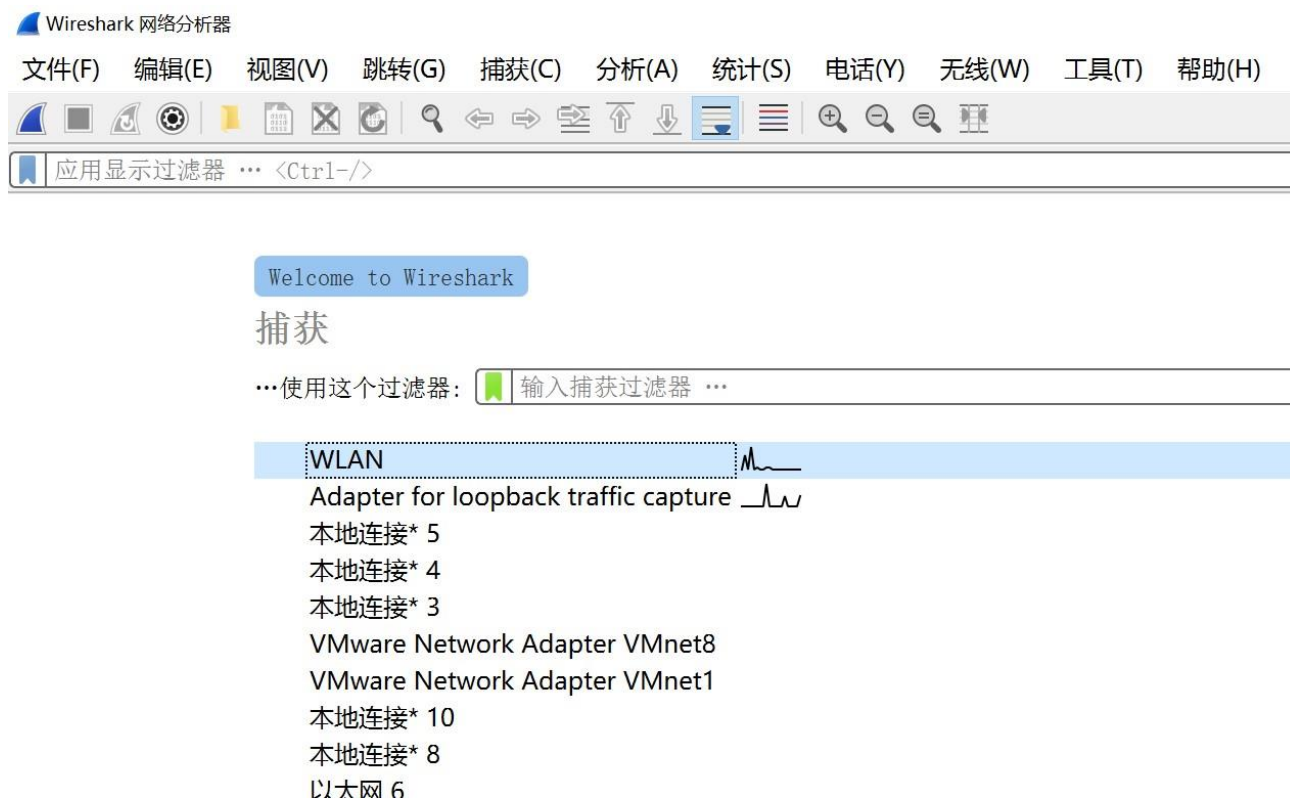
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
```

```
Scanner scanner = new Scanner(System.in);
System.out.print("请输入你的名字: ");
String name = scanner.nextLine();
System.out.print("请输入你的学号: ");
long ID = scanner.nextLong();
System.out.println("你好!" + name + " 你的学号是 " + ID);
}
```

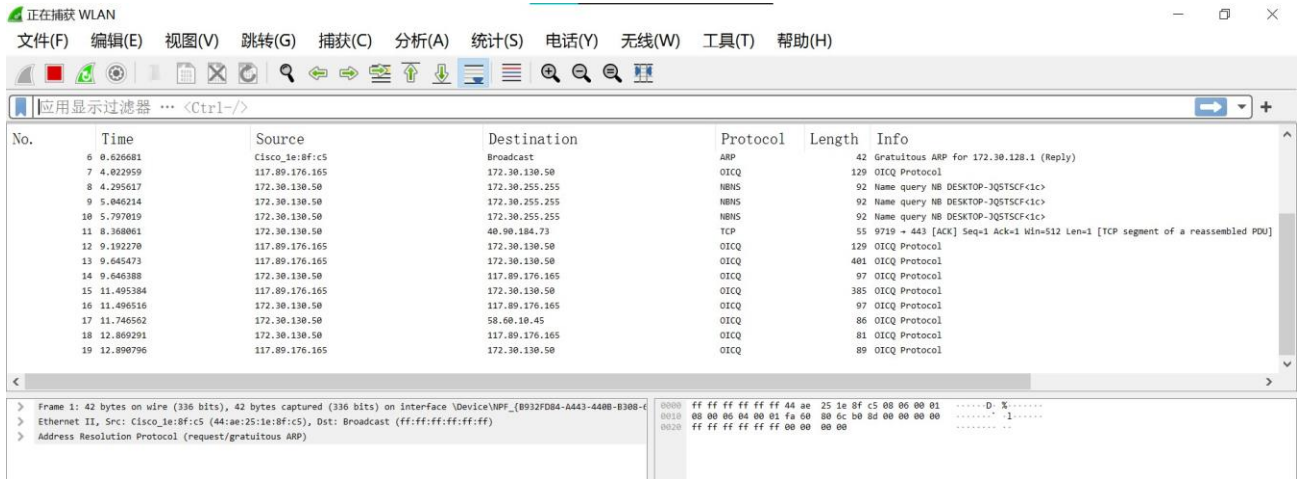
效果如下图所示

```
请输入你的名字: 郭夏辉
请输入你的学号: 10211900416
你好!郭夏辉 你的学号是 10211900416
```

3. 访问我常用的网站，并利用 Wireshark 将访问过程中的网络包抓取下来
首先我打开 Wireshark 软件，找到有流量的网卡，这里是无线网卡 WLAN



结果如下图所示，这个软件会将所有的包都抓取下来，且实时更新：



4. 根据不同的过滤规则删选特定的网络包

首先，我根据链接地址过滤。我以 bing.com 为例，先看一下它的 ip 地址

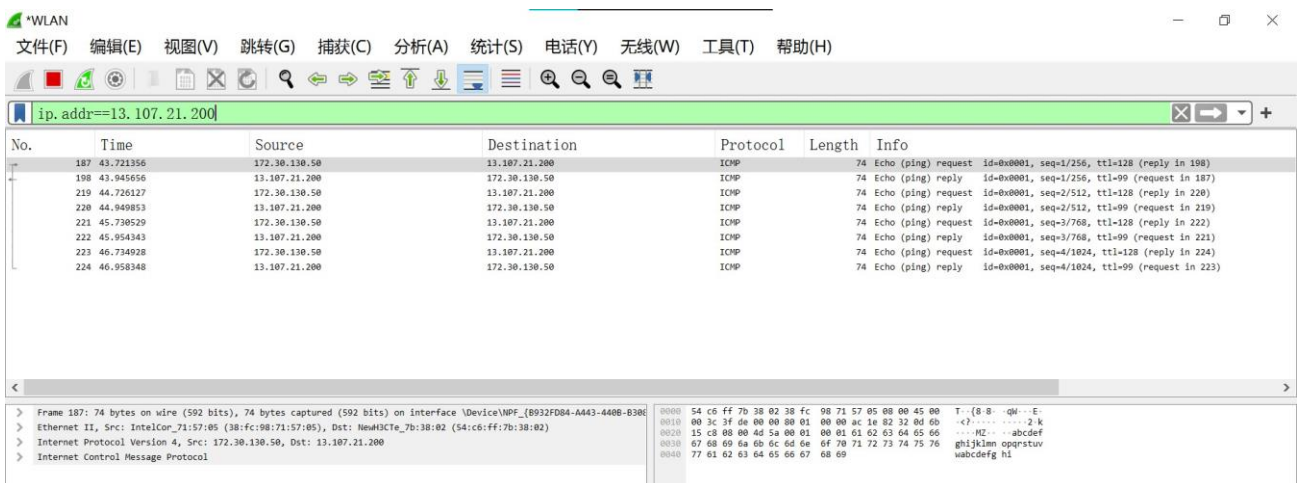
C:\Users\tom>ping bing.com

```
正在 Ping bing.com [13.107.21.200] 具有 32 字节的数据:
来自 13.107.21.200 的回复: 字节=32 时间=224ms TTL=99
来自 13.107.21.200 的回复: 字节=32 时间=223ms TTL=99
来自 13.107.21.200 的回复: 字节=32 时间=223ms TTL=99
来自 13.107.21.200 的回复: 字节=32 时间=223ms TTL=99
```

13.107.21.200 的 Ping 统计信息:

```
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 223ms, 最长 = 224ms, 平均 = 223ms
```

然后再过滤 ip.addr==13.107.21.200



然后我试了一下其他的过滤规则，比如网络包的目的地和源头、使用的协议，下面是操作过程中的截图

The first screenshot shows a Wireshark capture with the filter `ip.src==13.107.21.200`. The packet list shows four ICMP Echo (ping) replies from 13.107.21.200 to 172.30.130.50. The packet details pane for the selected packet (No. 224) shows the Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol layers.

The second screenshot shows a Wireshark capture with the filter `ip.dst==13.107.21.200`. The packet list shows four ICMP Echo (ping) requests from 172.30.130.50 to 13.107.21.200. The packet details pane for the selected packet (No. 224) shows the Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol layers.

The third screenshot shows a Wireshark capture with the filter `icmp and ip.addr==13.107.21.200`. The packet list shows eight ICMP Echo (ping) requests and replies between 172.30.130.50 and 13.107.21.200. The packet details pane for the selected packet (No. 187) shows the Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol layers.

五、总结

通过这次实验，我了解了基础的 Java 编程环境和 Wireshark 抓包软件的使用，感叹网络包的细节距离我们如此接近的同时，也希望自己未来学习计算机网络时能在不断的实践中洞悉网络的奥秘。