**中国海洋大学**

**计算机网络 课程实验报告**

实验题目: WireShark抓包

|  |
| --- |
| **实验概述** |
| 【实验目的及要求】     1. 了解抓包的基本原理:Wireshark可以嗅探网络中的数据包,并解析各类网络协议。 2. 学习使用Wireshark的基本操作: 启动抓包、设置网络接口、应用过滤规则等。 3. 抓取不同种类的数据包。 4. 训练和考察英文文献阅读的能力。     【实验环境】实验用的软硬件环境（配置）  依据实验手册给出的[www.wireshark.org](http://www.wireshark.org)下载并安装wireshark       下载安装后的界面如下，由于实验已做，因此有抓包记录： |
| **实验内容** |
| 【实验方案设计】（思路、步骤和方法，）  **实验任务1：**Capture Traces（捕获跟踪）   1.1 思路/步骤/方法  首先在捕获选项中混杂项勾选本地WLAN接口，仅对该接口进行捕获，按照实验手册，捕获过滤器我们选择tcp port 80，然后我们开始捕获。    紧接着我们访问wireshark官网给出的测试网址：https://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html    成功访问！  1.2 结果及分析  返回捕获器发现如下结果：    我们立即点击左上角红色“停止捕获”按钮，得到的背景色为绿色的即为有效捕获信息，此时说明我们抓包成功！  **实验任务2：**Inspect the Trace（检查跟踪）   2.1 思路/步骤/方法  我们先将上一步捕获到的数据包保存下来，命名为success\_catch:    我们选择HTTP GET数据包，No.4包即为GET数据包，是从我们的计算机发送到服务器上的，选择GET包后wireshark会给我们一个配对反馈，我们会发现对应No.6包表示服务器对我们计算机的响应，回应200 OK，说明我们通过GET发送的请求成功。  在选择HTTP GET数据包后，下方的面板会展示协议块，每个块可以查看详细信息。  Frame块不是协议，而是指网络数据包的基本单位，其中包含了一些元数据和实际的数据。  Ethernet为以太网、 Internet Protocol为IP协议、 Transmission Control Protocol为TCP协议、 Hypertext Transfer Protocol为HTTP协议。下图已标出：    仔细查看捕获的数据包：    2.2 结果及分析  我们正在获取一个网页，所以我们知道正在使用的协议层如实验手册中给出的：    HTTP是用于获取URLs的应用层web协议，运行在TCP/IP传输和网络层协议上。      **实验任务3：**Packet Structure（数据包结构）   3.1 思路/步骤/方法  我们下面逐一排查以太网、IP协议、TCP协议、HTTP协议：  在下方面板中逐个点击对应协议块，wireshark会在下面的面板中数据包中凸显它对应的字节，并在窗口底部展示长度。  Ethernet（以太网）：    IP协议：    TCP协议：    HTTP协议：    3.2 结果及分析  数据包结构：（单位：byte）   |  |  |  |  | | --- | --- | --- | --- | | Ethernet | IP | TCP | HTTP | | 14 | 20 | 20 | 111 |   在GET数据包的IP协议层找到source address即为本地IP    本机地址：192.168.0.5  应用层的协议名称为HTTP    正在访问www.msftconnecttest.com  **实验任务4：**Capture Packets from Other students（捕获别的同学的包）  4.1 思路/步骤/方法  首先在捕获选项中选择以混杂模式捕获数据包，我们得到如下的捕获结果：      在其中找到了一个不是本机的数据包（No.45数据包），即舍友使用的应用程序的数据包；  舍友计算机的IP地址为图片内左下红框 10.169.1.70；  舍友正在使用百度云管家。 |
| **小结** |
| 本次实验了解抓包的基本原理:Wireshark可以嗅探网络中的数据包,并解析各类网络协议。学习使用Wireshark的基本操作: 启动抓包、设置网络接口、应用过滤规则等。抓取不同种类的数据包并且训练和考察英文文献阅读的能力。对数据在各个层次之间的结构与状态以及传递过程有了更进一步的了解。锻炼了在Wireshark中查看本地计算机IP、查看服务器IP、查询客户使用程序的能力。同时对客户进程向服务器进程发出请求，请求建立连接，服务器进程接收客户进程发来的请求并作出回应这一过程有了更深层次与具象化的理解。 |