

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 利用 Wireshark 进行协议分析 | | | | | |
| 姓名 |  | | 院系 | 计算学部 | | |
| 班级 |  | | 学号 |  | | |
| 任课教师 | 刘亚维 | | 指导教师 | 刘亚维 | | |
| 实验地点 | 格物207 | | 实验时间 | 2023.11.4 | | |
| 实验课表现 | 出勤、表现得分(10) |  | 实验报告  得分(40) |  | 实验总分 |  |
| 操作结果得分(50) |  |
| 教师评语 | | | | | | |
|  | | | | | | |

****

|  |
| --- |
| 实验目的： |
| 熟悉并掌握 Wireshark 的基本操作，了解网络协议实体间进行交互以及报文交换的情况。 |
| 实验内容： |
| 1) 学习 Wireshark 的使用  2) 利用 Wireshark 分析 HTTP 协议  3) 利用 Wireshark 分析 TCP 协议  4) 利用 Wireshark 分析 IP 协议  5) 利用 Wireshark 分析 Ethernet 数据帧  6）选做内容：  a) 利用 Wireshark 分析 DNS 协议  b) 利用 Wireshark 分析 UDP 协议  c) 利用 Wireshark 分析 ARP 协议 |
| 实验过程： |
| 1. 学习 Wireshark 的使用和HTTP GET/response交互   基本介绍：  Wireshark是一种可以运行在Windows，UNIX，Linux 等操作系统上的分组分析器。Wireshark（前称 Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。使用简介：  （1）启动 Wireshark 软件，打开浏览器，选择网络接口 WLAN；    （2）浏览器输入网址http://hitgs.hit.edu.cn/zhxw/list.htm，Wireshark 抓包，并筛选 HTTP 协议报文；  （3）详细用户界面：命令菜单、俘获分组列表、分组头部明细、分组内容窗口、筛选俘获分组等信息实验指导书已给出，不再展示。   1. HTTP条件GET/response交互   （1）启动浏览器，清空浏览器的缓存（在浏览器中，选择“工具”菜单中的“Internet 选项”命令，在出现的对话框中，选择“删除文件”）。  （2）启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。  （3）在浏览器的地址栏中输入以下 URL: http://hitgs.hit.edu.cn/zhxw/list. htm,在你的浏览器中重新输入相同的 URL 或单击浏览器中的“刷新”按钮。  （4）停止 Wireshark 分组俘获，在显示过滤筛选说明处输入“http”,分组列表子窗口中将只显示所俘获到的 HTTP 报文。   1. TCP分析   A. 俘获大量的由本地主机到远程服务器的 TCP 分组  （1） 启动浏览器，打开http://gaia.cs.umass.edu/wireshark-labs/alice.txt网页，得到ALICE'S ADVENTURES IN WONDERLAND文本，将该文件保存到你的主机上。  （2）打开http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html，如图6-6所示，窗口如下图所示。在Browse按钮旁的文本框中输入保存在你的主机上的文件ALICE'S ADVENTURES IN WONDERLAND的全名（含路径），此时不要按 “Upload alice.txt file”按钮。    （3）启动Wireshark，开始分组俘获。  （4）在浏览器中，单击“Upload alice.txt file”按钮，将文件上传到gaia.cs.umass.edu服务器，一旦文件上传完毕，一个简短的贺词信息将显示在你的浏览器窗口中。    （5）停止俘获。  B. 浏览追踪信息  在显示筛选规则中输入“tcp”,可以看到在本地主机和服务器之间传输的一系列 tcp 和 http 报文，你应该能看到包含 SYN 报文的三次握手。也可以看到有主机向服务器发送的一个 HTTP POST 报文和一系列的“http continuation”报文。   1. IP分析 2. 通过执行 traceroute 执行捕获数据包   （1） 启动 Wireshark 并开始数据包捕获；  （2） 启动 pingplotter 并“Address to Trace Window”域中输入目的地址。   1. 对捕获的数据包进行分析 2. 抓取 ARP 数据包   （1）利用 MS-DOS 命令：arp 或 c:\windows\system32\arp 查看主机上 ARP 缓存的内容。  （2）在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）  （3）启动 Wireshark，开始分组俘获。   1. 抓取 UDP 数据包   （1）启动 Wireshark，开始分组捕获；  （2）发送 QQ 消息给你的好友；  （3）停止 Wireshark 组捕获；  （4）在显示筛选规则中输入“udp”并展开数据包的细节。   1. 利用Wireshark进行DNS协议分析   （1）打开浏览器键入:www.baidu.com  （2）打开 Wireshark,启动抓包.  （3）在控制台回车执行完毕后停止抓包 |
| 实验结果： |
| 1. 学习Wireshark的使用     图 1 Wireshark的抓包界面   1. HTTP分析   浏览器中输入的网址为：<http://hitgs.hit.edu.cn/zhxw/list.htm>，抓包结果如下：    1)HTTP GET/Response交互   1. 浏览器运行的是HTTP1.1，由上图Info中的HTTP/1.1可以看出。     图 2 对应版本号   1. 浏览器向服务器指出可接受的语言版本为zh\_CN，即简体中文。     图 3 accept-language为中文   1. 计算机的IP地址为172.20.19.249；服务器<http://hitgs.hit.edu.cn/zhxw/list.htm>的IP地址为219.217.226.25。      1. 服务器向主机返回的状态代码为200。     图 4 状态码Status Code  2)HTTP条件GET/response交互   1. 在第一个HTTP GET请求报文的内容中，不存在一行IF-MODIFIED-SINCE。     图 5 第一个HTTP GET请求报文的内容   1. 服务器明确返回了文件的内容，可以通过状态码200得知（其已成功发送完整的数据报文）。     图 6 返回报文内容   1. 较晚发送的”HTTP GET”请求中，请求报文中出现了“IF-MODIFIED-SINCE”行，该行后的信息是本地缓存文件中“Last-Modified”的最后缓存时间。     图 7 IF-MODIFIED-SINCE字段   1. 服务器对较晚的 HTTP GET 请求的响应中的 HTTP 状态代码是304，服务器实际上并没有返回文件的内容，因为客户端找到本地缓存之后，在想服务器发送请求后确认缓存是最新的，因此服务器并没有发送数据段，而是让客户端直接使用缓存的数据段。 2. TCP分析 3. 客户端主机的IP地址是175.24.154.66；TCP的端口号是443。 4. 服务器的 IP 地址是172.20.19.249；对这一连接，它用来发送和接收 TCP 报文的端口号是8175。      1. 客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是157747384；用flag的低两位来标注SYN报文段（第二位若为1则为SYN报文段）。     图 8 初始化TCP连接的SYN报文段序号    图 9 标注SYN报文段的flags   1. 服务器向客户端发送的 SYNACK 报文段序号是628135726；该报文段中，Acknowledgement字段的值是1876546121；Gaia.cs.umass.edu 服务器是由SYN报文段序号+1决定此值的；在该报文段中，是用Flags低起第五位和低两位来标示该报文段是SYNACK 报文段的。     图 10 服务器向客户端的SYNACK报文    图 11 SYNACK报文序号    图 12 Acknowledgment Number的值    图 13 SYNACK报文段的Flags   1. 三次握手过程如下：     图 14 三次握手过程   1. 包含HTTP POST命令的TCP报文段序号如下：     图 15 对应TCP报文段序号   1. TCP第六个报文段序号如下：     图 16 第六个报文段序号  发送时间如下：    图 17 对应发送时间  报文段对应的ACK的接收时间：    图 18 ACK接收时间  前六个TCP报文段的长度分别为681,1414,1414,1414,1414,1414。    图 19 前六个TCP报文段长度   1. 最小可用缓存空间为：30720     限制后，接收端缓存逐渐增大，直至最后回复到248064。       1. 跟踪文件中有重传，因为观察捕获的报文，出现了如下的一行。      1. TCP 连接的 throughput (bytes transferred per unit time)：   总长度为：    起止时间分别为：    其Throughput = 153028 / （3.087652-2.259913）= 184874.7 Bps。   1. IP分析   （1）    图 20 WireShark窗口   1. 我主机的IP地址是172.20.171.118；      1. 上层协议的字段值为1，表示ICMP协议；     图 21 协议值   1. IP头有20个字节，净载有36字节，由总长减去IP头得到。     图 22 IP头长度与总长信息   1. 观察标志位，发现其并未分片。     图 23 标志位显示没有分片  （2）    图 24 排序后的ICMP信息   1. 标识ID、TTL、首部校验和还有数据域会经常改变，剩下的数据都会经常保持不变； 2. 经常改变的原因：由于标识ID代表着每个不同的数据包，因此必须互不相同；由于ICMP的ping探测导致了TTL在不断增大；由于ID和TTL一直改变，因此首部校验和也会改变；又因为数据域中含有ICMP报文，ICMP报文首部又一直改变因此数据域也在一直改变。   （3）  1、 Identification是0，TTL是254。    2、 TTL不变，ID改变：第一跳路由器设置TTL字段为RFC指定的值，因此始终保持不变；而ID对每个IP字段是唯一的，所以改变。  （4）  1、 该消息确实被分成了不止一个IP数据报，由其数据报中标志位MF=1，且Fragment Offset不为0可以看出。    图 25 标志位以及offset    图 26 最后一片  2、在图25中，由Offset为0且MF=1可以知道它是首片且不是最后一片；而在图26中，由MF=0且Offset≠0可知它为最后一片。  3、它被分为了三片（如图27），且IP数据报头部中MF和Offset发生了变化（MF=1表示为非尾片，Offset可以知道它在第几片中，如图27中offset=2960就可知其在第三片中，即2960=1480\*2）。    图 27 包大小3500字节的最后一片信息   1. 抓取ARP数据包     图 28 查看arp缓存内容   1. 第一列为IP地址，第二列为MAC地址（即物理地址），第三列为其类型（分为动态和静态）； 2. ARP数据包的格式由９部分组成，一共28B，分别为：硬件类型2B；协议类型2B；硬件地址长度1B；协议地址长度1B；OP 2B；源MAC地址6B；源IP地址4B；目的MAC地址6B；目的IP地址4B。      1. Opcode为1即为请求包，opcode为2即为应答包。     图 30 请求包    图 31 应答包   1. 由于ARP在查询时，不知道IP对应的MAC地址，所以要在广播帧中传送以做到查询；由于在响应时，已经通过帧获得的发送方的源MAC地址，所以只需向该地址发送响应即可。 2. 抓取UDP数据包     图 32 Wireshark界面   1. 消息是基于UDP的； 2. 我的主机ip是172.20.189.41；目的主机的ip是61.149.23.35。      1. 我的主机发送消息的端口号为61650，服务器的端口号为8000。     图 33 主机端口号与服务端的端口号   1. 数据包格式及其大小为：源端口号2B；目的端口号2B；UDP 段长度2B；校验和2B。 2. 服务器返回ICQ用于确认服务器是否收到消息。这是由于UDP提供的是不可靠的无连接的传输服务，客户端并不知道服务器是否已收到信息，因此需要一个ICQ报文表示收到信息；而这个过程就可看出UDP是无连接的，并未采用TCP的三次握手过程。 3. 利用Wireshark进行DNS协议分析     图 34 Wireshark捕获的DNS报文    图 35 DNS报文相关信息 |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。  实验过程中遇到了关于ICMP协议的相关内容，在此对ICMP协议进行一定的展示。ICMP（Internet Control Message Protocol）Internet控制报文协议。它是TCP/IP协议簇的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。  事实上ICMP是IP的一个组成部分，与 IP 协议、ARP 协议、RARP 协议及 IGMP 协议共同构成 TCP/IP 模型中的网络层 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  1.对于计算机网络模型有了更加深入的认识。  2.尤其是对于应用层、传输层、网络层、链路层的一些协议有了更加深入的了解。通过对这些协议报文的抓包分析，对于其结构与工作原理的认识更加深刻。 |