

**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 | 郭子阳 | | 院系 | 计算机学院 | | |
| 班级 | 1703101 | | 学号 | 1170300520 | | |
| 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 | 2019.11.9 | | |
| 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  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 |
| 实验过程与结果： |
| 1. 学习Wireshark的使用  选择网卡后即可自动开始抓取数据  图片包含 屏幕截图  描述已自动生成  **2. 利用Wireshark分析HTTP协议**  **1）HTTP GET/response 交互**  访问<http://hitgs.hit.edu.cn/news>  结果如下：  图片包含 屏幕截图  描述已自动生成  HTTP协议请求头如下  图片包含 屏幕截图  描述已自动生成  HTTP响应头如下：  图片包含 屏幕截图  描述已自动生成  可以看出，浏览器运行的HTTP版本为HTTP 1.1，访问的服务器所运行的HTTP协议版本为HTTP 1.1  请求头中有Accept-Language字段，表明可接受en-US版本的对象  服务器返回的状态码为302  IP协议报文段如下：  图片包含 屏幕截图  描述已自动生成  表明本机的IP为2001:250:fe01:130:7536:656a:77cd:df71，服务器的IP地址为2001:da8:b800:253::dbd9:e219，都使用IPv6  **2）HTTP 条件 GET/response 交互**  由于<http://hitgs.hit.edu.cn/news>已无内容，改为访问<http://www.people.com.cn>  情况浏览器缓存后，首次访问网页时，请求头如下：  图片包含 屏幕截图  描述已自动生成  请求头中并没有IF-MODIFIED-SINCE字段  服务器响应如下：  图片包含 屏幕截图  描述已自动生成  响应的状态码为200，响应头中包含了LAST-MODIFIED字段，值为Thu, 07 Nov 2019 12:57:04 GMT，响应包含了文件的所有内容，File Data为163953字节。  当再次刷新网页时，浏览器向服务器发送的请求头如下：  图片包含 屏幕截图  描述已自动生成  其中包含了If-Modified-Since字段，值和上一次返回的Last-Modified的值相同，都是Thu, 07 Nov 2019 12:57:04 GMT。  服务器响应头为：  图片包含 屏幕截图  描述已自动生成  服务器响应的状态码为304，并没有包含文件信息，因为浏览器缓存的内容并未过期，服务器返回的字段里有ETag字段，指示缓存编号，浏览器可以根据ETag直接在缓存中查找文件，该ETag与第一次访问时返回的ETag字段内容一致。  **3. TCP分析**  上传Alice.txt后，wireshark获取到的第一个TCP包如下：  图片包含 屏幕截图  描述已自动生成  可以看出，客户端的IP地址为172.20.67.203，TCP端口号为64990，服务器的IP地址为52.17.172.5，端口号为80  用于初始化TCP SYN报文段的序号（Seq）为0，在Flags中，Syn位被置为1，表明该报文段为SYN报文段。  服务器向客户端返回的SYN ACK报文段如下，该报文段序号为0，Acknowledgment序号为1，因为客户端向服务器发送的报文段序号为0，所以服务器期望获得的后续报文段序号为1。Flags中将Acknowledgment和Syn字段设置为1，表明该报文段为SYNACK报文段。  图片包含 屏幕截图  描述已自动生成  TCP三次握手的报文段如下（13、14、15号）    包含HTTP POST命令的报文段如下，Seq为152730    三次握手后，客户端开始传送TCP报文段，共分为107个报文段：    第六个报文段如下：  图片包含 屏幕截图  描述已自动生成  序号为6482，是在POST发送之前发送的，对应的ACK是服务器返回的第六个ACK。  前六个TCP报文段信息如下：    大小分别为689字节、1448字节、1448字节、1448字节、1448字节、1448字节。  接收端公示的最小可用缓存空间为第二个ACK报文显示的窗口大小，为111：  图片包含 屏幕截图  描述已自动生成  后续接收的时候，接收端会不断增加窗口大小，故接收端缓存够用  整个发送期间没有重传现象发生，因为所有发送的报文段的序号Seq都不相同  TCP连接的throughput：  寻找len为0的报文，可获取头部长度为66 bytes：  图片包含 屏幕截图  描述已自动生成  共分为107个段传送，TCP报文段总头部长度为66 \* 107 = 7062 bytes  又因为：  图片包含 文字  描述已自动生成  于是TCP总传输数据大小为153010 + 7062 = 160072 bytes  最后一个段中，wireshark提供了Timestamps字段，记录了从第一个段到最后一个段所花费的时间：  图片包含 屏幕截图  描述已自动生成  共花费2.040246秒  于是throughput为160072 bytes / 2.040246 s = 78457.2057 bytes/s，约为78.5 KB/s  **4. IP分析**  使用pingplotter，首先在设置中将packet size设置为56 bytes    主机的IP地址如下，为172.20.77.246：    在IP数据包头中，上层协议号为1，如下：    由上图可以看出，IP头有20个字节，IP数据包大小为56字节，于是净载为36字节  展开flags字段，可以看到分段信息。可以看到该段偏移为0，More fragments为0，所以没有分片    通过分析多个ICMP包，总是在变的字段有Identification、Time to live和Header checksum  Identification用于鉴别不同的数据包，Time to live用来测试路由信息，Header checksum为校验和，这三个字段必须改变，其他字段保持常量  Identification为16位二进制数，按1递增  第一跳返回的TTL exceeded消息中，Identification为0，TTL为254    不变，因为相同的Identification是为了分段之后组装时为同一段，给同一个主机返回的ICMP，标识不代表序号，所以Identification不变，因为是第一跳路由器发送的数据报，所以TTL为最大值减一，总是为254。  当包大小改变为2000字节时，第一个Echo request被分片发送，第一个IP分片信息如下：    段偏移为0，More fragments字段设为1，表示该段为第一段，后续还有段，分片长度为1500字节  当包大小改为3000字节时，如下：    分成了三个段发送，前两个分片More fragments字段为1，后两个分片的offset分别为1480和2960  **5. 抓取ARP数据包**  在CMD中输入arp -a命令，结果如下：    其中，第一列为IP地址，第二列MAC地址，最后一列为类型，动态说明一定时间后记录会被删除  arp -d清除arp缓存后，ping 172.20.77.246后可过滤到arp数据包，如下：    Arp数据包格式如下：    接收方MAC       6字节  发送方MAC 6字节  Ethertype          2字节  硬件类型 hdtyp              2字节  上层协议类型protyp       2字节  MAC地址长度hdsize       1字节  IP地址长度 prosize         1字节  操作码  op              2字节  发送方MAC smac[6]        6字节  发送方IP sip[4]              4字节  接收方MAC  dmac[6]     6字节  接收方IP          dip     4字节  填充数据              18字节  判断arp包是请求包还是应答包可以根据opcode，opcode为1的是请求包，2为应答包  由于请求时，源主机不知道目的主机的mac地址，故无法在链路层封装该IP的mac帧，于是采用广播的模式，而当应答时，主机可以通过arp帧获取到源主机的mac地址，可以对特定主机应答。  **6. 抓取UDP数据包**  从QQ中发送消息后，捕获到的数据包如下：    该消息基于UDP，本机IP为172.20.77.246，目的IP为125.39.132.99，本机QQ端口为4024，目的服务器的端口为8000  消息包含字段：源端口，目的端口，长度，校验和，各占16个字节  没发送一个ICQ数据包，服务器就会返回一个ICQ数据包，返回的是接受结果，UDP是不可靠的数据传输，仅仅返回一个简单的接收状态，无重传等机制，UDP数据包是没有序列号的，于是数据是乱序的无连接的。  **7. 利用DNS进行DNS协议分析**  请求：    响应： |
| 心得体会： |
| 学会了使用WireShark进行协议分析，深入理解了各个协议的实现 |