

**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 | 张瑞 | | 院系 | 计算机科学与技术 | | |
| 班级 | 1903104 | | 学号 | 1190201421 | | |
| 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 | 2021年11月21日 | | |
| 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： |
| **（一）Wireshark的使用**    打开Wireshark，选择网络接口，此处选择WLAN。    进入用户界面，能看到命令菜单、俘获分组列表、分组头部明细、分组内容和筛选规则等5个重要组成部分。  **（二）HTTP分析**  **1.HTTP GET/response交互**  访问<http://hitgs.hit.edu.cn/news>并俘获HTTP GET请求报文及其对应的响应报文，如下：      可以看到浏览器和所访问的服务器运行的均是HTTP1.1，浏览器向服务器指出它能接收的语言版本为简体中文（zh-CN），实验所用计算机的IP地址为2001:250：fe01:130:e160:84fa:6bad:3f7d，服务器的IP地址为2001:da8:b800:253::dbd9:e219，服务器返回给浏览器的状态码为200。  **2. HTTP条件GET/response交互**  清空浏览器缓存后，访问<http://www.4399.com>，待网页加载成功后，再次刷新该网页，分别俘获两次的HTTP报文，如下：      从图中可以看出，当浏览器发出第一个HTTP GET请求报文时，没有IF-MODIFIED-SINCE这一行，服务器明确返回了文件内容，因为返回的状态码为200。当浏览器第二次发出HTTP GET请求时，请求报文中出现IF-MODIFIED-SINCE这一行，后面跟着的信息为缓存最后更新的时间（Sat, 27 Nov 2021 01:15:31 GMT），响应报文中的状态码为304，服务器不会明确返回文件，因为服务器判断的结果为Not Modified，浏览器可以使用本地尚未过期的缓存文件。  **（三）TCP分析**  向gaia.cs.umass.edu发送alice.txt，俘获大量的TCP分组，如下：    从上图可以看到，客户端主机的IP地址为172.20.44.236，TCP端口号为59732；服务器的IP地址为128.119.245.12，对这一连接，其用来发送和接收TCP报文的端口号为80。    进一步查看客户端发送给服务器的报文内容，可以看到初始化TCP连接的TCP SYN报文段的序号是 0，通过将Flags标志位中的SYN位置为1，来表示该报文段是SYN报文段。    再查看服务器向客户端发送的报文内容，可以看到该SYN ACK报文段的序号是0，Acknowledgment字段的值为1，该字段的值应为下次希望收到的报文段序号（此处为1），同时，该报文段将Flags标志位中的SYN和ACK位均置为1,来表示该报文是SYN ACK报文段。    仔细查看此处，可以看到TCP三次握手的过程（忽略灰色部分）：客户端先向服务器发送连接请求，然后服务器返回响应，客户端再进行ACK确认，成功建立连接。    如果查看该过程中的HTTP报文，能看到包含HTTP POST命令的TCP报文段的序号为152191。    如果将包含HTTP POST命令的TCP报文段看作是TCP连接上的第一个报文段，可以看到，该TCP连接上的第六个报文段的序号是1，是在客户端的HTTP POST后发送的（6.684109s时），其对应的ACK是在服务器端的HTTP响应报文之后，TCP连接关闭之前接收的（7.001271s时），如下：    通过观察TCP报文信息，或者HTTP POST报文信息，均能发现前6个TCP报文段长度分别为654、1360、1360、1360、1360、1360，如下两图所示：      在整个跟踪过程中，接收端公示的最小的可用缓存空间是29200：    本次实验中未出现限制发送端传输的情况，但如果出现这种情况，接收端的缓存仍然可能不够用，这可能是由于已发送但尚未到达的报文引起的。  本次实验中也没有出现重传的报文段，因为没有出现过重复的序列号或者重复ACK号。    考察TCP连接建立的过程，传输数据bits，用时s，吞吐率为bits/s。  **（四）IP分析**  启动PingPlotter访问jwes.hit.edu.cn，依次设置Packet Size为56、2000和3500，俘获数据包如下：    从中可以看到，主机IP地址为172.20.44.236，在IP数据包头中，上层协议（upper layer）字段的值为0x01，IP头有20字节，净荷载等于IP分组长度（56字节）-IP首部长度（20字节），即36字节。且该IP数据包未分片，因为Flags字段为全零（其中的More fragments位为0表示无后续分片，片偏移量也为0，说明是从数据包的第一个字节开始的）。  观察主机发出的一系列ICMP消息，发现IP数据报中的TTL、Identification和Header Checksum字段总是改变，因为不同的数据报有不同的ID（不分片的话），每经过一跳TTL就会减1，校验和也需要重新计算；但是IP版本必须保持常量，因为在传输过程中通常需要固定一个版本。另外，也可以注意到Identification字段为16位的加一递增的值。  查看最近的路由器返回给主机的ICMP Time-to-live exceeded消息，发现Identification字段和TTL字段值分别为0x0000和0xff，如下图所示：    这些值不会改变，因为作为第一跳路由器返回的消息，不用经过任何其它路由器就能到达主机，不会有信息的修改。  接下来观察将Packet Size改为2000字节后主机发送的第一个ICMP Echo Request消息，如下图。其中片偏移量为1480，说明IP数据包被拆开了。    如下图，再观察第一个IP分片，IP头部的Flags中的More Fragments为1说明数据包被分片了；偏移量0说明该片为第一个分片，是从原数据的第一个字节开始封装的；用总长度1500字节减去头部长度20字节得到分片长度为1480字节。    最后观察将Packet Size改为3500字节后主机发送的第一个ICMP Echo Request消息，如下图。    因为一个IP数据报最多封装1480字节的数据（前面也已经看到），这里发现片偏移量为2960，说明加上当前这片共有3片，从IPv4被俘获的情况和详细信息也可以得到验证。再对比这些分片，可以发现IP数据报头部的总长度、标志位（Flags中的More Fragments和Fragment Offset）、校验和是不一样的。  **（五）ARP分析**    如上图所示，查看主机上ARP缓存的内容，可以看到有三列内容，从左到右分别是：IP地址、物理地址和类型。  捕获一个ARP数据包，如下图：    可见ARP数据包由9个部分组成：硬件类型（2字节），协议类型（2字节），硬件地址长度（1字节），协议地址长度（1字节），OP（2字节)，发送端MAC地址（6字节），发送端IP地址（4字节），目标MAC地址（6字节），目标IP地址（4字节）。格式如下图：    通过检测OP字段可以判断ARP数据是请求包还是应答包：当OP为0x0001时，是请求包；OP为0x0002时，是应答包。如下两图所示：      另外，由于ARP查询不知道目的IP地址所对应的MAC地址，所以需要在广播帧中传送，确保能被对方收到；而ARP响应已经从查询报文中知晓目的IP地址对应的MAC地址，且局域网中其他主机不需要该查询结果，所以在一个有着明确目的局域网地址的帧中传送。  **（六）UDP分析**  用QQ给好友发信息后，俘获分组，如下图：    从上图可以看到，QQ消息是基于UDP的，实验所使用的主机和目的主机的IP地址分别是2001:250:fe01:130:5458:bd27:5ad5:2d40和2402:4e00:1900:1038:0:9085:4fc:949f，两者的端口号分别是4021和8001。  UDP数据报包括首部和数据，其中首部有源端口号（2字节）、目的端口号（2字节）、数据报长度（2字节）以及校验和（2字节），数据长度不定。具体格式如下：    主机每发送一个ICQ数据包，服务器都会返回一个ICQ数据包，这是因为ICQ数据包是在UDP数据包的基础上实现的，所以它也和UDP一样是不可靠传输，服务器能对接收的数据返回ACK，但不能保证数据没有丢失。与前面的TCP对比，可以看到UDP没有三次握手的过程，也不用提供接收和发送缓存、拥塞控制参数以及序号与确认号等参数，是无连接的。  **（七）DNS分析**  打开[www.baidu.com](http://www.baidu.com)并俘获分组如下：    可以看到DNS协议架设在UDP之上，域名解析服务器在53号端口接收查询信息。 |
| 问题讨论： |
| 1.在对一些网站（例如http://news.hit.edu.cn）进行二次访问时，HTTP请求报文中也不会有IF-MODIFIED-SINCE这一行，是因为这些网站在第一次返回时，就不提供缓存时间信息，导致二次访问时无法查询缓存的有效性，必须重新获取网站信息。  2.WireShark会俘获大量的分组，想要高效率地查找数据包时，必须灵活使用filter来筛选。 |
| 心得体会： |
| 本次实验让我掌握了WireShark的使用方法，在一次次的抓包分析中更深入地理解了各种网络协议及其具体格式，也对协议栈有了更深刻的印象。 |