

**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 | 靳贺霖 | | 院系 | 计算机学院 | | |
| 班级 | 1603109 | | 学号 | 1160300312 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物213 | | 实验时间 | 2018/11/17 | | |
| 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**： |
| （注：实验报告模板中的各项内容仅供参考，可依照实际实验情况进行修改。）  本次实验的主要目的。  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**： |
| **概述本次实验的主要内容，包含的实验项等。** 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，实验结果以及问题讨论**： |
| 以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。 **Wireshark的使用：**  首先启动web浏览器，然后启动Wireshark，可以通过“capture”中的菜单中的“Capture Options”选项中设置分组俘获的选项。还可以通过“Interface List”菜单中选择计算机使用的网络接口，然后点击“Start”开始进行分组俘获的过程就会接受由选定网卡发送和接收的分组。点击Start后，在浏览器地址栏收到某网页的URL，比如<http://www.hit.edu.cn>，当页面加载完成以后，单机stop按钮，停止俘获，就可以看到这个时间段内俘获的所有分组。可以在显示筛选规则中输入http，点击回车，就只显示HTTP协议报文，如下所示  然后，也可以选择分组窗口中的某个http报文，这样，就会在下面窗口中显示这个报文的详细信息，如下图所示：  以上就是Wireshark的基本使用。 **HTTP分析： （1）HTTP GET/response交互**  访问<http://hitgs.hit.edu.cn/>，然后俘获获得的http报文，对其进行相关分析。首先分析发送方的GET报文，如下图所示  可以看出，我的浏览器运行的是HTTP1.1，我的浏览器指出它能接受的语言是简体中文（zh-cn），我的计算机的IP地址是172.20.5.147，服务器<http://hitgs.hit.edu.cn/>的IP地址是219.217.226.25。然后分析服务器返回的一个相应报文，如下所示  可以看出，服务器所运行的也是HTTP1.1，服务器返回的状态码是200 OK，表示请求成功，然后对于整个交互的过程，如下图所示  可以看出除了200 OK，服务器还相应有404 Not Found，说明请求的一部分组建是没有得到相应的。 **（2）HTTP条件 GET/response交互**  清空浏览器的缓存，然后再次访问<http://hitgs.hit.edu.cn/>，然后得到的第一个GET报文如下所示：  可以看出其中是没有If-Modified-Since行的，然后对应的第一个服务器相应报文如下所示：  可以看到，时有Data字段的，所以第一次服务器响应明确返回了文件的内容。然后发呢系稍晚发送的其中的一个HTTP GET请求如下图所示  可以看出其中是有If-Modified-Since行的，后面紧接的是一个日期，对应的是缓存拥有的该组建所对应的日期。再看其对应的服务器的相应报文，如下所示  可以看出服务器相应的代码为304 Not Modified，而且没有数据，所以服务器并没有明确返回文件的内容。 **TCP分析： A.俘获大量的由本地主机到远程服务器的TCP分组**  按照实验指导书的步骤完成即可。 **B.浏览追踪信息**  在显示筛选规则中输入tcp，可以看到如下的结果  可以看出，进行SYN报文的三次握手。还可以看出客户端发送方端口为54998，IP地址为192.168.43.254；Gaia.cs.umass.edu服务器的IP地址为111.13.65.52，它用来发送和接收TCP报文的端口号为80。 **C.TCP基础**  首先来看客户端向服务器发送的第一个SYN报文，如下图所示：  可以看出，初始化的Sequence number为0。在该报文段中，它是用Flags字段为0x002来标识它是SYN报文段的。接下来来看服务器向客户端发送的SYN ACK报文，如下图所示  可以看出，服务器向客户端发送的SYN ACK报文段序号为0；在该报文段中，Acknowledgment字段为1，它是通过要发送的确认ACK数来确定的。在该报文段动，是通过Flags字段为0x012来标示该报文是SYN ACK报文段的。然后来分析TCP建立的过程。最开始收发的三个报文段如下图所示：  可以看出，首先客户端向服务器发送SYN报文，然后服务器向服务器发送SYN ACK，表明自己收到这个连接请求并相应，接下来，客户端再向服务器相应一个ACK，说明自己收到服务器的相应。这三次收发体现了TCP连接建立过程的三次握手过程。接下来来看包含HTTP POST命令的TCP报文段，如下图所示：  可以看出，包含HTTP POST命令的TCP报文段的序号为1。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，那么TCP连接上的第六个报文段如下图所示：  可以看出，第六个报文段的序号为4597，对应的ACK是第六个ACK，如下图所示：  这个ACK是在第六个报文段发送间隔一个窗口长度发送的，如下图所示：  然后由下图分析  可以看出，前六个报文长度分别为， 515，1360，1360，1360，1360。然后接受端控制的最小可用缓存空间如下所示：  为Window size value = 16660，而且分析可以看出，这个窗口是可以增大的所以接收端的缓存是够用的。在跟踪的报文中，没有发现有重传的报文，这个也比较容易解释，因为发送过程比较稳定没有丢包的发生，客户端不发送重复的SEQ。最后来计算这个TCP连接的吞吐率throughtput，相关数据如下图所示：  可以看出，整个连接传输了152869B，然后整个数据传输从3.02214s到5.864819s，计算得throughput = 152869B/(5.864819-3.02214)s = 53.789 kBps **IP分析： A.通过执行tranceroute执行捕获数据包**  按照实验指导书进行设置即可。 **B.对捕获的数据包进行分析**  可以在捕获窗口中看到主机发送的一系列ICMP Echo Request包和中间路由器返回的一系列ICMP TTL-exceeded消息。如下图所示  然后观察其中一个ICMP Echo Request包的Internet Protocol部分，如下图所示   可以看出，主机的IP地址为2001:250:fe01:130:f9ce:e1b6:c9c1:4e61，上层协议字段的值为58（ICMPv6）；IP头的字节数为44，该IP数据包净载1960-44=1912字节。而且可以看出Offset=181，所以这个IP数据报是分片了的。  然后将捕获的数据包按源IP地址排序。可以看出，主机发出的一系列ICMP消息的IP数据报中，identification，checksum，TTL是一直在变化的。因为要区分数据报并按序组合，所以ID必须是变化的；TTL也是一直在变化的，每一跳要减一；checksum也要重新计算，所以也是变化的。其他值一般都需要保持常量。IP数据报的identification字段如下所示：  可以看出其是32位的，并且对于发送的顺序是按序增加的。然后，考虑最近的路由器返回给主机中的ICMP报文如下所示：  可以看出，TTL（Hop Limit）的值为255，identification字段为0x000000f0；最近的路由器返回给主机的ICMP中，返回给主机的TTL exceeded消息中这些值是不变的，因为标识是用来标识这个数据报，是不会改变的，因为是第一跳返回的数据报，所以TTL是最大值255。  将捕获的数据包按时间排序，包大小为2000字节的第一个ICMP Echo Request消息如下所示：  可以看出，该消息被分成了两个IP数据报。IP头部的More Fragments = yes表明数据报是第一个而不是最后一个，这个数据报的长度为1448bytes。 **抓取ARP数据包：**  用MS-DOS命令：arp –a来看主机中ARP缓存的内容，如下图所示：  其中，第一列为IP地址，第二列为MAC地址，第三列为类型（静态是不变的，动态在一定时间后会被删除）。然后通过命令行模式下输入ping 192.168.1.82来抓取ping命令时的数据包。选择第一个发送的ARP数据包，如下所示：  可以看出，ARP数据包由九个部分构成，分别为：两个字节的硬件类型，两个字节的协议类型，一个字节的硬件地址长度，一个字节的协议地址长度，两个字节的Opcode，六个字节的发送方MAC地址，四个字节的发送端IP地址，六个字节的目标MAC地址，四个字节的目标IP地址。可以通过Opcode字段来判断ARP数据的类型，0x0001时是请求，0x0002时是应答。而且因为查询时没有对应的目的MAC地址，所以要采用广播的方式来获取；应答时，主机可以通过ARP报文获知源主机的MAC地址和IP地址，因此可以向明确目的局域网地址的帧中传送。 **抓取UDP数据包：**  启动Wireshark，发送QQ消息给好友，捕获相应的数据报。如下图所示：  可以看出，消息是基于UDP和OICQ的。截取其中一个发送报文的信息，如下图所示：  可以看出，我的主机IP地址为172.20.5.147，目的主机ip地址为183.232.127.124。发送QQ消息的端口号为4028，QQ服务器的端口号为8000。数据报包含源端口，目的端口，长度和校验和，均占两个字节。发送一个ICQ数据包后，服务器会返回给主机一个ICQ数据包来确认。但是只有这一次返回基本的ACK，并没有保证数据一定送达。而且可以看出，UDP数据报中没有TCP数据报中的大多数元素，比如说序列号，说明UDP可能是按序到达的。而且并没有握手确认连接，说明UDP是无连接的。 **利用WireShark进行DNS协议分析：**  用浏览器输入[www.baidu.com](http://www.baidu.com)，启动Wireshark抓取DNS报文，得到如下的结果：  截取其中一个DNS报文，如下所示：  可以看出DNS也是基于UDP协议的。 |
| **心得体会**： |
| 结合实验过程和结果给出实验的体会和收获。   1. 学会了Wireshark的使用，了解如何通过抓取某个网卡接收到的数据包来对协议等进行分析； 2. 通过使用Wireshark对HTTP协议进行分析，加深了对HTTP报文格式的理解，以及对HTTP协议的GET/response交互有了深刻的认识； 3. 通过使用Wireshark对TCP协议进行分析，直观得感受了使用SYN报文段进行三次握手的过程，以及对实际报文进行解读，加深了理解； 4. 通过使用Wireshark对IP协议进行分析，了解了如何使用traceroute工具，而且对IP数据包头部信息进行比较详尽的解读，尤其是对IP分片有了更深刻的理解； 5. 通过使用Wireshark对ARP数据包进行分析，了解了ARP解析的过程以及Ethernet II的封装格式，加深了对ARP解析的理解； 6. 通过使用Wireshark对UDP数据包和DNS进行分析，了解了UDP数据报的简单结构已经无连接的特性。 |