

**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 | 任永鹏 | | 院系 | 计算机学院 | | |
| 班级 | 1737102 | | 学号 | 1173710107 | | |
| 任课教师 | 张英涛 | | 指导教师 | 张英涛 | | |
| 实验地点 | 格物楼213 | | 实验时间 | 周六12节 | | |
| 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： 7. 利用Wireshark分析DNS协议 8. 利用Wireshark分析UDP协议 9. 利用Wireshark分析ARP协议 |
| 实验过程： |
| 1. wireshark的使用   首先下载安装wireshark软件,然后打开后看到如下的首页    在capture选项中选择想要捕捉的interface(网卡),然后我选择WLAN选项,然后就进入了工作界面,之后打开浏览器,访问http://www.hit.edu.cn,在顶部的筛选框中输入http,然后按enter键,之后出现如下界面:    在这个界面,上部的框展示了http协议的关键信息,如源目的地址,http的status,  中部的框展示了具体的数据帧内容。下部的框展示了数据帧16进制内容,右面是对应的ascii码。   1. 利用wireshark分析HTTP协议   打开wireshark的分组嗅探器,然后使用浏览器访问<http://202.118.253.94:8080/eportal/success.jsp?userIndex=34656561326364323636353634663235633965343166323965643462396138335f3137322e32302e3130312e3134395f31313733373130313037>  ,然后停止wireshark分组嗅探器。  得到的截图:    思考问题:  浏览器发送请求使用的是http 1.1协议  访问的服务器使用的协议也是 http 1.1协议  浏览器接收的语言是 zh-CN,zh 表示的是中文  我的计算机的ip地址是172.20.101.149  访问的服务器ip地址是 202.118.253.94  服务器返回的状态码是 200  (2)条件get/response 交互    首先清空浏览器的缓存,然后打开wireshark的分组俘获,用浏览器访问<http://202.118.253.94:8080/eportal/success.jsp?userIndex=34656561326364323636353634663235633965343166323965643462396138335f3137322e32302e3130312e3134395f31313733373130313037>  得到如下的http访问分组    经过对第一个http get请求的分析    没有发现IF-MODIFIED-SINCE 这一行  服务器返回的报文中明确返回了文件的内容,因为有line-based text-data这一项内容:    因此可以断定服务器返回了文件的所有内容。  对较晚的http Get请求中出现了IF-MODIFIED-SINCE    字段,在该字段后面跟着的是缓存文件最后更改的时间,用于询问服务器该文件在这个时间之后是否发生了修改,如果没有发生修改,浏览器就直接使用缓存,如果发生了修改,则服务器返回更改后的新document  对于这个请求,服务器的http响应报文中出现了304    没有http的内容部分,表明服务器没有返回文件的内容,让浏览器直接使用缓存的文件内容。  http协议抓包总结:  分析了几个报文,整个http协议使用明文传输,其结构由header和body两部分组成,header和body中间用一个空行隔开,header中有若干键值对,用于记录http请求和响应的一些基本信息,body中是http协议搭载的文件信息。   1. TCP分析   首先在浏览器中打开了网站:http://gaia.cs.umass.edu/wireshark-labs/alice.txt,将网页的内容保存成一个本地文件,文件名是alice,然后打开网站http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html,选择好上面保存的文件。  之后打开wireshark的分组俘获开关,在页面点击upload alice.txt file    得到了网站的如下反馈:    之后在wireshark中输入筛选条件tcp,得到如下的结果    问题回答:  向gaia.cs.umass.edu服务器传送文件的客户端主机的IP地址是172.20.101.149    端口是63321  Gaia.cs.umass.edu服务器的IP地址是128.119.245.12  端口是80  客户端与服务器三次握手时SYN报文段是:    其序列号是0,通过把 SYN=1来标识这是SYN报文段  以下是服务器的SYNACK报文段的TCP表头:    报文段序号是0,Acknowledgement是1,服务器因为要对客户端发送的数据报进行ACK,所以设置Acknowledgement为1,表明这个帧是对SYN的ACK  通过设置Acknowledgement为1,且SYN为1,唯一标识这个帧是SYNACK  第三次握手的TCP报文如下:    以上是TCP连接建立的三次握手过程  以下是包含POST请求的TCP报文:    可以看出其seq是1  第六个TCP报文段的截图如下:    其序号是6497,发送的时间是5.859217  对该报文的ACK的时间是6.104255  前6个TCP报文的长度分别是: 656B,1460B,1460B,1460B,1460B,1460B    在双方的TCP连接中,最小的缓存大小是152977B,在限制发送后,接收端的缓存够用。  在发送过程中,有重传的报文段,判断的依据是如果有相同seq的报文段被发送,表示重传该报文段    throughput的计算过程:  在第一个包含post的数据报被传输时,时间是5.859217,其seq是1,一秒过后,找到这个时候的一个数据帧:其seq是152978,  可以知道在1s内,发送的数据长度大约是150000B,其吞吐量大约是150Kb/s   1. IP分析   下载安装pingplotter软件,打开界面后,打开Edit->Options->Packet,设置    Packet size 大小为56bytes  之后打开wireshark的数据报捕获开关,开始捕获数据包。  之后在trace输入框中输入www.hit.edu.cn ,点击开始trace,得到如下运行结果:    之后分别修改packet size大小为2000和3500,得到如下的运行结果:      在wireshark中的捕获结果如图所示:    本机的ip地址是172.20.101.49  ip数据包头中,上层协议字段的值是 ICMP(1)  Ip数据报的头部有20bytes长  这个字段表示的就是ip数据报的头部的长度  该ip数据包的净载大小为60-20=40bytes  这个数据报没有分片,因为其MF标志位是0,且fragment offset的值是0,表示这个ip分组是最后一个分组,且offset为0,所以一定是没有分组的ipv4数据报。  (2)ip分组中TTL 字段 header checksum字段 和 identification 字段的值总是在发生改变。  Version, protocol 字段必须保持常量,而 identification 和 header checksum,以及TTL字段必须改变。  version和protocol对于ICMP协议都是固定不变的,而每次的identification 因为是在发送时随机选取,所以一定会不一样,而TTL,因为是要trace router,所以每次发送的ICMP报文的TTL都依次+1,所以一定不一样,而checksum的值与所有字段的 值有关,所以只要有字段的值发生改变,check sum的值一定会发生改变。  identification字段的格式是四个16进制的数字,大小为2字节  (3)找到由最近的路由器（第一跳）返回主机的ICMP Time-to-live exceeded消息    其identification 是60533,TTL 是63  与发出去时其ICMP报文中的TTL相比,少了65  因为ICMP time-to-live-exceeded报文返回了超时的报文段的信息,而到达超时结点的时候,这个报文段的TTL已经变成63了,所以返回的ICMP中,TTL就是63  (4)单击Time列按钮，对捕获的数据包按时间排序。找到在将包大小改为2000字节后你的主机发送的第一个ICMP Echo Request消息。    这个是找到的第一个ICMP Echo Request消息,其长度为520,并不是设定的2000,所以发生了分组,将2000的长度分为了4部分,每个分组长度为500bytes  Ip协议分析总结:  ip协议的头部中version,header length ,flags reserved bit这些数值很少发生变化,而identification,flags MF,FO,DF,TTL,checksum ,source,destination这些字段的值经常发生改变。   1. 抓取ARP数据报     通过运行arp -a命令,可以查看到主机上arp 缓存的内容  在wireshark中开启分组捕获  在命令行中输入ping 172.20.101.77,之后在wireshark中查看捕获结果,使用arp的关键词筛选分组      由9部分组成  Hardware type 表示使用的链路层硬件类型 2bytes  Protocol 表示协议的类型 2bytes  Hardware size 为6 与 mac地址的大小6字节对应 1bytes  op用来标识是request 类型的arp还是 response类型的arp 2bytes  之后是source mac address 6bytes 和source IP address 4bytes  以及 destination mac address 6bytes 和 destination IP address 4bytes    在OP字段中,0×0001 时是请求，为0×0002 时是应答请求。  Request 使用的是广播地址,是因为它不知道ip对应的mac地址是多少,所以只能通过广播的形式发出,但是如果对应ip的主机收到了这个广播帧,则必须针对性的告知request主机自己的mac是多少,没有必要再广播地址。   1. 抓取UDP数据包   启动Wireshark，开始分组捕获,发送QQ消息给好友停止Wireshark组捕获；  得到如下的分组捕获结果:    QQ的通讯是基于UDP协议的  这是一个示例的qq udp 数据报    主机的ip地址是 172.20.101.149 端口是59144  目的主机的ip地址是 183.57.37.180 目的端口是 8200  数据报的格式是:  Source port 源端口号 占2bytes  Destination port 目的端口号 占2bytes  Udp长度 指udp数据报的整个长度 占2bytes  Udp校验和 占2bytes  之后是数据字段  在客户端发送了一个UDP给QQ服务器之后,QQ也会给客户端发送一个UDP数据报,表示服务器已经收到消息  因为UDP是不可靠数据传输,但是QQ作为一个及时通讯软件,所以必须要在UDP的基础上(在应用层)自己实现一个可靠的数据传输,所以出现了服务器会立即给客户端发送UDP确认数据报的情况  通过与TCP协议的对比,可以发现UDP是没有连接的建立过程的,可以直接从源IP地址的源端口发送数据报给目的IP的目的端口,且如果没有数据发送,可以直接停止传输;而TCP如果想要发送数据,则必须要先通过三次握手建立连接,双方才可以互相发送数据报,同时,如果TCP链接想要中断,则必须要通过四次数据交换才能中断连接,由此可以看出,UDP协议是无连接协议,而TCP是有连接的协议。   1. 利用WireShark进行DNS协议分析   操作过程:  先打开wireshark的分组捕获开关  然后打开浏览器输入 [www.baidu.com](http://www.baidu.com)  按下回车访问该网站  在wireshark中得到了如下的DNS数据报的截图:    如下是DNS协议的格式:    DNS报文格式分为五大部分。分别为： 报文头Header， 问题区段（Question），回答区段（Answer），权威区段（Authority）， 额外信息区段（Additional）。但是不是五个段必须存在，只有Header必须存在，别的段在不同情况下不存在。  DNS ID号（DNS ID Number）： 用来对应DNS查询和DNS响应 |
| 实验结果 |
| 在实验过程一栏中都有 |
| 问题讨论： |
| 遇到的问题:  在分析TCP数据报的过程中,遇到以下的问题:  找不到post  解决办法：问同学 |
| 心得体会： |
| wireshark这个工具真的太强大了，我对HTTP,TCP,ipv4,ARP,DNS等不同协议格式有了更深入的了解，中间遇到了各种各样的困难，最后都通过问同学的到了解决。写完很有成就感！ |