

**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 | 冯运 | | 院系 | 计算机学院 | | |
| 班级 | 1603108 | | 学号 | 1160300524 | | |
| 任课教师 | 聂兰顺 | | 指导教师 | 聂兰顺 | | |
| 实验地点 | 格物楼207 | | 实验时间 | 2018.11.23 | | |
| 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://hitgs.hit.edu.cn/news/main.psp,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192.168.199.184  访问的服务器ip地址是 219.217.226.25  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://http://hitgs.hit.edu.cn/news/main.psp 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    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    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'S ADVENTURES IN WONDERLAND,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192.168.1.139      端口是52075  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6502,发送的时间是5.379198  对该报文的ACK的时间是5.636489  前6个TCP报文的长度分别是: 715B,1514B,1514B,1514B,1514B,1514B  在双方的TCP连接中,最小的缓存大小是19200B,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5.379198,其seq是1,一秒过后,找到这个时候的一个数据帧:其seq是150105,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  遇到的问题:  在分析TCP数据报的过程中,遇到以下的问题:    对其中的window size scaling factor不太懂,于是查找资料获得:  由于这个rwcd字段的位数是16bit,则其能表示的最大数是65535,如果遇到更大的值将不能表示,于是引入了window size scaling factor这个变量,表示在返回的window size字段基础上\* scaling value, 从而扩大了表示的范围。  这个scaling value的值是在三次握手的过程中通过options 约定的,于是我在SYNACK中看到如下字段:     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3000,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192.168.199.184  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56-20=46bytes    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27980,TTL 是1  与发出去时其ICMP报文中的TTL相比,少了1  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1了,所以返回的ICMP中,TTL就是1  (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    其fragment offset的值不为0,表明其确实发生了分组,MF为0表明这个分组是最后一个分组。该分片的长度为500bytes  C、找到在将包大小改为3500字节后主机发送的第一个ICMP Echo Request消息    数据包被分成了15份,每份20bytes,  头部中total length的值由520变成了40,而flags中fragment offset的值由185变成了370,表明分组的长度发生了改变,同时checksum也发生了改变。  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. 以太网帧     以下是一个捕获的以太网帧,其头部由三部分组成:destination, source, type  其destination 是一个mac地址,是链路层的接口地址,这里是我们寝室的wifi的mac地址  source是传输该帧到局域网上的适配器的mac地址  type表明这个以太网帧上载有的是一个ipv4协议的报文  之后的是数据字段,这部分用来装载ipv4的数据报。  以下是以太网帧的结构:     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 192.168.1.82,之后在wireshark中查看捕获结果,使用arp的关键词筛选分组  发现没有捕获到任何的arp 分组,于是我开始分析:  ping使用的是ICMP协议,是一个网络层的协议,而ARP协议是链路层的协议,ARP是用来询问某个ip对应的mac地址的,为何ping不会触发ARP报文呢？  最后明白了,因为这个ip不在局域网中,访问它不需要知道他的mac地址,所以通过运行arp -d \*指令删除所有的arp缓存,然后再次执行ping 192.168.1.82时,就有了acp数据报的捕获。    捕获的arp数据报如下图所示    问题回答:  1、ARP缓存中有几个接口,表示的是计算机中有几个不同的网卡,而在每一个接口下,又有几条不同的记录,每一条记录的第一个值是一个ip地址,第二个值是这个ip地址对应的mac地址,之后的类型表示的是这个ip对应的mac地址是静态(长期不变)的还是动态的(有一定时效的)  2、以下是捕获到的arp request的数据包:    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 192.168.199.184 端口是4001  目的主机的ip地址是 111.30.159.65 目的端口是 8000  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    查询/响应（Query/Response， QR）： 用来指明这个报文是DNS查询还是响应，占1个比特位。为1代表响应，0代表查询    操作代码（OpCode）：用来定义消息中请求的类型    权威应答（Authoritative Answer， AA）：这个比特位在响应的时候才有意义。则说明这个响应是由域内权威域名服务器发出的  截断（Truncation， TC）：用来指出报文比允许的长度还要长，导致被截断    期望递归（Recursion Desired, RD）：如果设置了RD，就建议域名服务器进行递归解析，递归查询的支持是可选的。    在这个DNS数据报中使用了递归查询的方式  保留（Z）： 未使用，用0表示    问题计数（Question Count）： 问题区段中的问题记录数    回答计数（Answer Count）：回答区段中的回答记录数    域名服务计数（Name Server Count）：权威区段中的记录数    额外记录数（Additional  Records Count）：在额外信息区段中的记录数      可以看出其查询的是sp0.baidu.com这个域名 |
| 实验结果 |
| 在实验过程一栏中已结合实验过程同步给出,详见实验过程一栏 |
| 问题讨论： |
| 在实验过程一栏中已结合实验过程同步给出,详见实验过程一栏 |
| 心得体会： |
| 通过自己使用wireshark这个分组嗅探工具对计算机网络不同层次的不同协议进行捕获和分析,加深了对HTTP,TCP,ipv4,ARP,DNS等不同协议格式的了解,以及各种协议的特点,以及传输数据的原理的理解,特别是对TCP复杂的连接和传输过程有了实际而直观的体会。通过这个实验,我得以对课堂上学过的多种多样的协议有了深入的理解和实际的操作,特别是一些非常底层的协议,平时很少接触,在这次实验中可以实际的获得和体验。 |