

**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 | 余涛 | | 院系 | 计算机科学与技术学院 | | |
| 班级 | 1803202 | | 学号 | 1180300829 | | |
| 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 | 2020.11.21 | | |
| 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 | | | | | | |
|  | | | | | | |

**计算学部**

|  |
| --- |
| 实验目的： |
| 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： |
| 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。  在实验开始前，首先需要知道我的Windows 10主机的IPv4地址：  Windows10主机的IPv4地址：172.20.69.192（每次连接校园网产生的ip地址不同）    **(1) Wireshark的使用**  对于捕获工具Wireshark的使用，我选择了Windows 10主机的无线网卡来进行分组捕获，然后在进行分组捕获的同时，在浏览器输入网页url为<http://www.hit.edu.cn>进行访问。在完整的页面加载完成后，结束分组捕获。并且在显示筛选规则中输入“http”进行筛选，然后可以发现，Wireshark捕获了本机所有利用本机的无线网卡与其他网络实体进行交换的报文，然后选择分组列表窗口中的第一条http报文，这一条就是计算机向www.hit.edu.cn服务器发送的HTTP GET报文，然后就可以看到该报文的以太网帧、IP 数据报、TCP 报文段、以及HTTP 报文首部信息。将所有分组保存在文件wireshark\_use.pcapng中。  **(2) HTTP分析**  **1. HTTP GET/response交互**  首先在显示筛选规则中输入“http”进行筛选，然后使用Wireshark开始捕获，打开一个web，访问http://today.hit.edu.cn，然后停止分组捕获。将所有分组保存在http\_get\_response\_1.pcapng中。  **2. HTTP 条件GET/response交互**  首先启动浏览器，清除浏览器的缓存，然后启动Wireshark，在显示筛选规则中输入“http”进行筛选，然后使用Wireshark开始捕获，访问http://today.hit.edu.cn，然后刷新该网页，然后停止分组捕获。将所有分组保存在http\_get\_response\_2.pcapng中。  **(3) TCP分析**  首先访问<http://gaia.cs.umass.edu/wireshark-labs/alice.txt>，获得alice.txt文件。然后打开<http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>，选择好本地alice.txt文件的位置。然后启动Wireshark开始分组捕获，在浏览器中点击”Upload alice.txt file”按钮上传文件，在文件上传完毕后停止Wireshark分组捕获。在筛选规则中选择“tcp”部分，进行分析即可，将所有分组保存在文件tcp.pcapng中。  **(4) IP分析**  使用pingplotter进行实验，待发送IP分组的网站为www.baidu.com，启动Wireshark开始分组捕获，首先发送一系列56字节的包；再发送一系列2000字节的包，；再发送一系列3500字节的包，然后停止Wireshark捕获。将所有分组保存在ip.pcapng中（实验室对于3500字节的包出了点错误，将其单独实验分组结果保存在ip\_3500bytes.pcapng中）。  **(5) 抓取ARP数据包**  在命令行输入 arp –a 命令，查看主机上ARP缓存的内容。在命令行模式下输入：ping 192.168.1.82。然后启动Wireshark进行捕获。将所有分组保存在arp.pcapng中。  **(6) 抓取UDP数据包**  首先启动Wireshark分组捕获，然后用QQ给好友发送消息，消息发送结束后，停止分组捕获。  将所有分组保存在udp.pcapng中。  **(7) 利用Wireshark进行DNS协议分析**  首先清空dns缓存，在浏览器中访问<http://www.google.com.hk/>，进行Wireshark抓包。将所有分组保存在dns.pcapng中。 |
| 实验结果： |
| 采用演示截图、文字说明等方式，给出本次实验的实验结果。  **(1) Wireshark的使用**  捕获工具Wireshark的安装过程略。对本Windows 10的无线网卡进行捕获，在输入“http”进行分组过滤后，访问<http://www.hit.edu.cn>，然后点击第一条HTTP报文，可以看到如下的信息：    **(2) HTTP分析**  **1. HTTP GET/response交互**  对本Windows 10的无线网卡进行捕获，在输入“http”进行分组过滤后，访问http://today.hit.edu.cn，然后点击第一条HTTP报文，可以看到如下的信息：    思考问题：  问题一：浏览器运行的协议为HTTP/1.1；访问的服务器运行的HTTP协议版本号同为HTTP/1.1。  问题二：浏览器向服务器指出的接收的语言版本对象为：  Accept: text/html, application/xhtml+xml, image/jxr, \*/\*\r\n  问题三：本机ip地址为172.20.69.192，服务器http://today.hit.edu.cn的ip地址为202.118.254.117  问题四：服务器向我的浏览器返回的状态代码为200  **2. HTTP 条件GET/response交互**  对本Windows 10的无线网卡进行捕获。首先打开浏览器，清除浏览器的缓存，然后打开Wireshark，在输入“http”进行分组过滤后，访问http://today.hit.edu.cn，然后刷新页面，点击第一条HTTP报文，可以看到如下的信息：    思考问题：  问题一：第一个HTTP GET请求没有IF-MODIFIED-SINCE头部。  问题二：服务器在第一个GET中明确了返回文件的内容。如下所示      能够看出服务器明确回复了文件的内容：用来构成主页HTML的其他元素。    问题三：对于浏览器向服务器发出的较晚的HTTP GET 请求，报文中有一行IF-MODIFIED-SINCE。再起后面跟着的信息是缓存文件上次修改的时间。    问题四：服务器对于较晚的HTTP GET 请求的响应中的HTTP状态代码为304。根据同一文件前后请求的差别，在上图中可以看出，最近请求服务器回复消息的长度远远小于之前的长度，说明服务器并未明确返回文件的内容，而是浏览器从缓存中读取内容。  **(3) TCP分析**  启动Wireshark开始捕获，然后上传alice.txt后如下：    在Wireshark筛选TCP报文后如下：    问题一：向gaia.cs.umass.edu 服务器传送文件的客户端主机的IP 地址和TCP 端口号分别为172.20.69.192和49592。因为此时客户端主机向服务器发送大量TCP报文使用的是49592。  问题二：Gaia.cs.umass.edu 服务器的IP 地址为128.119.245.12,。对这一连接，它用来发送和接收TCP 报文的端口号为80。  问题三：客户服务器之间用于初始化TCP 连接的TCP SYN 报文段的序号是0。在该报文段中将SYN置为1，表示该报文段用于tcp建立连接。如下图：    问题四：服务器向客户端发送的SYNACK 报文段序号为0。该报文段中Acknowledgement 字段的值是1。Gaia.cs.umass.edu服务器通过SYN请求报文段的seq序号加1来决定此值。在该报文段中，是使用flag部分的ack以及SYN标记为1来标示该报文段是SYNACK报文段的。如下图：      问题五：分析tcp三次握手过程：  上面两张图展示了tcp三次握手中的前两次握手，包括客户端向服务器发送SYN请求报文，然后服务器向客户端回复SYNACK报文，然后第三次握手就是客户端向服务器回复ack报文段，此时回复的ack报文段中，ack的内容为为1（为SYNACK报文段序号加1），说明是第三次握手。如下图：    问题六：包含HTTP POST 命令的TCP报文段的序号为1。如下图：      问题七：此种条件下的TCP连接上的第六个报文段为如下图所示：    是在第一帧发送后的3.512325秒后发送的该报文段。  该报文段对应的ACK报文接收如下：    可以看出接收时间为第一帧发送后的3.790654秒。  问题八：前六个TCP报文段的长度分别如下：    问题九：接收端公示的最小的可用缓存空间为29200字节。在整个过程中接收端并没有对发送端的传输进行限制。    问题十：得到序列号随时间的变化，可以看出整个过程中序列号随时间一直增长，而若有重传的报文段会出现序列号变小的情况。说明没有发生重传。  问题十一：吞吐率的计算，如下所示：  此次共传输的字节数为149238-1=149237字节，所用时间为4.647357-3.233182=1.414175秒，  吞吐量throughput=149237字节/1.414175秒=1.05MB/S    **(4) IP分析**  使用pingplotter向www.baidu.com发送一系列大小为56字节，2000字节和3500字节的IP分组，然后用Wireshark进行捕获结果如下：      问题一：我的主机的IP地址为172.20.0.172  问题二：对我主机第一个发送的ICMP报文进行查看，IP数据包头中，上层协议为ICMP(1)  问题三：IP头为20字节，该IP数据包的净载为36字节（IP数据包总大小为56字节，头部有20字节，则净载为56-20=36字节）  问题四：没有分片，观察flag区如下：    可以看到没有其余的帧且帧的偏移为0，MF=0，则说明该IP数据包没有分片。  问题五：通过比较三个分组可以发现，这些IP数据包的Identification、TTL和checknum字段总是发生改变。如下图：        问题六：必须保持常量的字段有：版本号、上层协议、源IP地址和目的IP地址。原因为要使该IP数据包成功发送到目的地址，这些字段必须保持该值；必须改变的字段有：Identification，TTL以及checknum，Identification就是IP数据包的序号，每个包的序号都不同，且根据traceroute的工作原理，每次主机发送的IP数据包的TTL都加一，而校验和为头部数据求和得出，这两者的变化都会使校验和发生改变。  问题七：Identification字段为两个字节，第一个IP数据包的Identification是随机产生的值，之后的IP数据包的Identification每次增加1。  问题八：对所有报文按照时间排序：    可以发现第三个报文就是最近的路由器返回给我主机的ICMP Time-to-lice exceeded消息，然后查看该报文如下：    可以发现Identification字段为0x97bd，TTL字段为64  问题九：  最近的路由器返回的这些消息的Identification字段的值会改变，但是TTL不变，都是64。这是因为Identification字段用来区分不同的IP数据包，而TTL字段为默认设置的64。  问题十：  可以发现包大小改为2000字节后我的主机发送的第一个ICMP Echo Request消息被分成了两片。如下图：    问题十一：观察第一个报文段，此时DF=0，MF=1，说明了该数据包进行了分片，并且不是最后一个分片。并且该数据包的片偏移为0，说明该包是第一个数据包。该分片的长度为1500字节。    问题十二：（在发送3500字节的IP数据包时出了点小错误，所以新建了一个捕获，分组保存在ip\_3500bytes.pcapng中）  可以发现包大小改为3500字节后我的主机发送的第一个ICMP Echo Request消息被分成了三片。如下图：    问题十三：  这三个IP分片的数据头部 Total length，片偏移量，标志位，checksum字段发生了变化。  **(5) 抓取ARP数据包**      问题一：ARP缓存中第一列为为接口的IP地址，第二列为接口的MAC地址，第三列为地址的类型，其动态表示该条目是通过arp协议学来，而静态表示该条目是由网卡自己生成或者是通过手工配置的。  问题二：清除主机上ARP缓存的内容,抓取ping 192.168.1.82命令时的数据包如下：    ARP数据包的格式如下：    其由这些部分构成：  硬件类型：2字节  协议类型：2字节  硬件地址（MAC地址）长度：1字节  协议地址长度：1字节  OP：2字节  发送方MAC地址：6字节  发送方IP地址：4字节  接收方MAC地址：6字节  接收方IP地址：4字节  问题三：根据这个ARP数据包的OP字段判断，若OP=1则为请求包，OP=2则为应答包  。  问题四：原因为：查询MAC时主机不知道目的IP的MAC地址是多少，所以需要在自己的局域网中广播该查询，提高了找到该设备的可能性。而ARP响应只需要发给提出查询的主机即可，故ARP响应要在一个有着明确目的局域网地址的帧中传送。  **(6) 抓取UDP数据包**  启动Wireshark进行分组捕获，发送QQ消息给好友，停止捕获，筛选出UDP数据包如下：    问题一：消息基于UDP。  问题二：我的主机ip地址为172.20.0.132，目的主机ip地址为125.39.132.147  问题三：主机发送QQ消息的端口号是4011，QQ服务器的端口号是8000。    问题四：  数据包的格式如上图所示，且其字段分别为：  源端口号：2字节  目的端口号：2字节  报文长度：2字节  校验和：2字节  问题五：由于UDP是不可靠数据传输，所以每次发送一个ICQ数据包后服务器会返回一个ICQ数据包表示确认。而且UDP一个与TCP不同的地方就是UDP发送数据之前没有进行握手，从这里可以看出UDP是无连接的。  **(7) 利用Wireshark进行DNS协议分析**  访问www.baidu.com的抓包结果如下： |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。  所有的思考问题都在上面的实验结果部分进行了回答。 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  通过Wireshark分析各种网络协议的执行，可以加深对于各种协议交互过程的理解。并且对于协议中各个字段的作用有了更深的了解。通过使用Wireshark对各个协议的分析，加深了我对不同协议之间的层级关系以及TCP、UDP、DNS、HTTP等协议的功能的理解。 |