

**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 | cycleke | | 院系 | 计算学部 | | |
| 班级 |  | | 学号 |  | | |
| 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 | | | | | | |
|  | | | | | | |

**计算学部**

|  |
| --- |
| 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:   1. 利用 Wireshark 分析 DNS 协议 2. 利用 Wireshark 分析 UDP 协议 3. 利用 Wireshark 分析 ARP 协议 |
| 实验过程： |
| 1. **Wireshark 的使用**   首先在 Wireshark 官网 <https://www.wireshark.org/download.html> 下载 Wireshark，之后捕获器选择接口进行捕获。由于我本地电脑使用的是无线网卡进行上网，所以我选择捕获 WLAN 接口。    在选定接口后，我可以在界面中看见 Wireshark 的抓包信息。     1. **HTTP 分析** 2. HTTP GET/response 交互   首先启动浏览器，然后启动 Wireshark 分组嗅探器。在窗口的显示过滤说明处输入“http”，分组列表子窗口中将只显示所俘获到的HTTP 报文。之后开始 Wireshark 分组俘获。  在打开的浏览器中访问 <http://today.hit.edu.cn>，捕获HTTP报文，之后停止分组俘获。最后将捕获的报文保存到文件中，用于后续分析。   1. HTTP 条件 GET/response 交互   首先启动浏览器，清空浏览器的缓存，然后启动 Wireshark 分组嗅探器。在窗口的显示过滤说明处输入“http”，分组列表子窗口中将只显示所俘获到的HTTP 报文。之后开始 Wireshark 分组俘获。  在打开的浏览器中刷新 <http://today.hit.edu.cn> 的页面，捕获 HTTP报文，之后停止分组俘获。最后将捕获的报文保存到文件中，用于后续分析。   1. **TCP 分析**   首先下载位于 <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> 的 alice.txt 文本文件，然后打开 <http://gaia.cs.umass.edu/Wireshark-labs/TCP-Wireshark-file1.html> ，选中自己下载的文本文件。  此时打开启动Wireshark，开始分组俘获。在浏览器中，单击“Upload alice.txt file”按钮，将文件上传到 gaia.cs.umass.edu 的服务器。在文件上传完毕，一个简短的贺词信息将显示在浏览器窗口中后，停止 Wireshark 的捕获。最后将捕获的报文保存到文件中，用于分析。   1. **IP 分析**   打开 Wireshark 进行数据包的捕获，之后使用 pingplotter 依次向 hit.edu.cn 发送大小为 56 字节、2000 字节和 3500 字节的 IP 数据包。最后将捕获的报文保存到文件中，用于后续分析。     1. **抓取 ARP 数据包**   首先利用 arp 指令查看本地的 ARP 缓存表。    之后开启 Wireshark 的分组捕获，在命令行中 ping 172.20.90.194，ping 通后停止捕获。   1. **抓取 UDP 数据包**   启动 Wireshark 分组捕获，利用 QQ 给好友发送消息，消息发送结束后，停止分组捕获，之后将这段时间捕获的报文保存到文件中。   1. **利用 Wireshark 进行 DNS 协议分析**   首先打开 Wireshark 进行抓包，在浏览器中访问 [www.baidu.com](http://www.baidu.com) ，完成后停止抓包，保存到文件。 |
| 实验结果： |
| 1. **HTTP 分析** 2. HTTP GET/response 交互   利用 Wireshark，我们可以获得如下的报文，截图如下：     1. 通过截图，可以发现我的浏览器使用的是 HTTP/1.1 协议；访问的服务器使用的也是 HTTP/1.0 协议。      1. 浏览器向服务器指明其可以接收的对象如下，即：   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\n     1. 我的 IP 地址为 172.20.100.197，服务器 today.hit.edu.cn 的 IP 地址为 202.118.254.117。      1. 服务器向本机浏览器返回的状态码为200。      1. HTTP 条件 GET/response 交互 2. 在清除浏览器缓存之后，访问 today.hit.edu.cn，发出的第一个 HTTP GET 请求如下，可以发现不包含 IF-MODIFIED-SINCE 行：      1. 服务器接下来的服务器响应报文返回了明确的内容（若干 json 文件，通过右键-追踪流-HTTP流可以知道），状态码均为 200。      1. 而在较晚的 HTTP GET 请求中，报文头部中添加了 IF-MODIFIED-SINCE 行，后面接着的是缓存文件上次更改的时间。      1. 服务器对于较晚的 HTTP GET 请求，其返回的状态码为 304，而且并返回的内容极短（只有392），并没有包含具体的内容。        1. **TCP 分析** 2. 向 gaia.cs.umass.edu 服务器传送文件的客户端（源主机）主机的 IP 地址与 TCP 端口号为 172.20.100.197 和 2415。 3. gaia.cs.umass.edu 服务器 IP 地址为 120.232.150.121，接收端口号为 3434。      1. 客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号为 0（绝对值为 3301679959）；该报文段将SYN标志位置为 1，表示该报文段为 SYN 段用于 TCP 建立连接。      1. 服务器向客户端发送的 SYNACK 报文段序号是 1（绝对值为 3301679960）；Acknowledgement 字段值是 1（绝对值为 2649404903）。服务器将随机指定一个值以决定此值。在该报文段中，在INFO中写入 [SYN,ACK] 来标示的。      1. 下面的图片展示的就是 TCP 三次握手的过程。首先客户机想服务器端发送 SYN 请求报文；之后服务器向客户机回复 SYN，ACK 报文；最后客户机向服务器回复 ACK 报文段，完成三次握手。      1. 包含 HTTP POST 命令的 TCP 报文的序号是 1（绝对值为 1187582186）。      1. 第六个报文段的序号为 5889（绝对值为 1187588074）。接受时间为 Nov 26, 2020 10:24:33.814944000 中国标准时间。对应的 ACK 接受时间为 Nov 26, 2020 10:24:33.847832000 中国标准时间。          1. 前六个 TCP 报文段的长度各是 128、1440、1440、1440、1440 和 1440。      1. 在整个跟踪过程中，接收端公示的最小的可用缓存空间是 515，限制发送端的传输以后，接收端的缓存是够用。 2. 没有重传的片段。依据为发送端的报文段序号始终在增加，没有出现重复发送某一个序号的报文段的情况，故没有重传的。      1. 最后 ACK 包的序列号为 156318，计时器为 7.249878000 秒，而第一个包的计时器为 5.658626000 秒，所以吞吐量为 。 2. **IP 分析** 3. 我的 IP 地址为 172.20.100.197。      1. 通过分析 IP 数据包，可以分析出上层协议为 ICMP。      1. 通过上面的 Header Length 行和 Total Length 行可以分析得出 IP 头有 20 字节，该IP数据包的净载为 56。 2. 观察 Flags 区，可以发现 More fragments 标志为空，没有其余的帧并且帧的偏移为0，可以推断出没有进行分片。      1. 观察 IP 数据包的可以发现 Identification、TTL 和 Checksum 的值总是变化。      1. 必须保持常量的是版本号、首部长度、Differentiated Services Field 以及协议（始终为ICMP）。必须改变的是 TTL、Checksum 和 Identification，TTL 为生存时间，每次转发必然改变；由于TTL的改变，Checksum 自然也会改变；Identification 则是用于区分不同的 ICMP 报文。 2. Identification 自 1200 开始，接下来的包依次增加 1。 3. Identification 段为 64430，TTL 为 64。      1. Identification 段发生变化，这样可以区分不同的 ICMP time-to-live exceeded 消息；但 TTL 保持不变，为 64（均为一次转发）。 2. 该消息被分解为不止一个数据包。      1. IP 头部可以在 Flags 域中，看到 More fragments 被置为 1 且偏移量为 0，表示该分片不为最后一片。该分片的长度为 1500 字节。      1. 原始数据被分成了3片。      1. 标志位部分、偏移量和 Checksum 部分发生了变化。      1. **抓取 ARP 数据包** 2. ARP 表的格式如下。在 ARP 表中，每一项表示一个 IP 地址到物理地址的映射。每一项第一列是IP地址，第二列是物理地址，第三列是类型。      1. ARP数据包的格式如下图所示，共由九部分构成：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），目的MAC地址（6 字节），目的IP地址（4 字节）。      1. 可以通过 Opcode 字段判断，若为 1 则是请求包；若为 2 则是应答包。      1. 因为进行 ARP 查询时并不知道目的 IP 地址对应的 MAC 地址，所以需要广播查询；而 ARP 响应报文知道查询主机的 MAC 地址（通过查询主机发出的查询报文获得），且局域网中的其他主机不需要此次查询的结果，因此 ARP 响应要在一个有着明确目的局域网地址的帧中传送。 2. **抓取 UDP 数据包** 3. 消息是基于 UDP 的。 4. 本机 IP 地址为 172.20.100.197，目的主机 IP 地址为 111.30.159.62。      1. 主机发送 QQ 消息的端口号为 4022，QQ 服务器的端口号是 8000。      1. UDP 数据报由五部分构成，分别是源端口号（4 字节），目的端口号（4 字节），长度（4 字节），校验和（4 字节）和应用层数据。      1. 因为 UDP 是不可靠的数据传输，需要上层协议来实现可靠数据传输，因此每次发送 ICQ 报文后又回复一个 ICQ 数据包来确认。UDP 是无连接的，因为可以看到发送数据之前没有连接的建立过程（如 TCP 的三次握手），没有序列号，因此为无连接数据传输。 2. **利用 Wireshark 进行 DNS 协议分析**   利用 Wireshark 进行 DNS 协议抓包的结果如上。 |
| 问题讨论： |
| 见实验结果的分析。 |
| 心得体会： |
| 通过本次实验，我熟悉并掌握 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。在对于不同的协议报文分析的过程中，我对于各个网络协议的理解更为深入，对于部分细节也有了全新的认识。 |