

**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 | 张智雄 | | 院系 | 计算学部 | | |
| 班级 | 2103601 | | 学号 | 2021112845 | | |
| 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 | 2023.11.4 | | |
| 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： |
| 必做内容：   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** |
| 实验过程及结果： |
| 1. **Wireshark的使用**   启动Web浏览器和Wireshark，选择“capture”下拉菜单中的“Capture Options”命令，设置分组俘获的选项后，开始分组捕获。    开始分组捕获，出现分组捕获窗口：    浏览器打开<http://www.hit.edu.cn>，捕获并提取显示HTTP报文如下：     1. **利用Wireshark分析HTTP协议** 2. HTTP GET/response 交互 3. 启动Web browser，然后启动Wireshark分组嗅探器； 4. 在窗口的显示过滤说明处输入“http”，开始Wireshark分组俘获； 5. Web browser窗口中输入地址<http://hitgs.hit.edu.cn/zhxw/list.htm>并跳转； 6. 停止分组俘获，得到捕获结果如下：     **问题思考：**   1. **你的浏览器运行的是HTTP1.0，还是HTTP1.1？你所访问的服务器所运行HTTP协议的版本号是多少？**     **答：**由GET请求报文知，浏览器运行HTTP1.1；而由返回报文知，服务器同样运行HTTP1.1   1. **你的浏览器向服务器指出它能接收何种语言版本的对象？**     **答：**由‘Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6’可知，服务器支持zh-CN (中文简体，中国)、zh (中文)、en (英文)、en-GB (英文，英国)、en-US (英文，美国)（按照优先级从高到低）   1. **你的计算机的IP地址是多少？服务器**[**http://hitgs.hit.edu.cn/zhxw/list.htm**](http://hitgs.hit.edu.cn/zhxw/list.htm)**的IP地址是多少？**   **答：**根据GET命令的Source和Destination字段可知，   * IPv4下，本机IP地址为172.20.245.248；服务器地址为219.217.226.25；      * 图形用户界面, 文本, 应用程序    描述已自动生成IPv6下，本机IP地址为2001:250:fe01:130:317d:48bf:5884:9289；服务器地址为2001:da8:b800:253::dbd9:e219  1. **从服务器向你的浏览器返回的状态代码是多少？**   **答：**一般情况下均为200，存在404等情况。  图形用户界面, 应用程序  描述已自动生成   1. HTTP 条件 GET/response 交互 2. 启动浏览器，清空浏览器的缓存。 3. 启动Wireshark分组俘获器，开始Wireshark分组俘获。 4. 在浏览器的地址栏中输入URL：<http://hitgs.hit.edu.cn/zhxw/list.htm>跳转并刷新。 5. 停止Wireshark分组俘获，结果如下。     **问题思考：**   1. **分析你的浏览器向服务器发出的第一个HTTP GET请求的内容，在该请求报文中，是否有一行是：IF-MODIFIED-SINCE？**   **答：**观察得，在首次向服务器发送GET请求的内容中，没有IF-MODIFIED-SINCE行。     1. **分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？**   **答：**服务器明确返回了文件内容，可以从返回报文中直接看出，也可从服务器返回的状态码200获知。     1. **分析你的浏览器向服务器发出的较晚的“HTTP GET”请求，在该请求报文中是否有一行是：IF-MODIFIED-SINCE？如果有，在该首部行后面跟着的信息是什么？**   **答：**第二次发送的GET请求并不包含IF-MODIFIED-SINCE首部，但相较于第一次报文增加了“Cache-Control: max-age=0”字段（这通常是强制刷新导致的），指定了缓存的最大寿命（时间）。在这里，"max-age" 设置为0，表示该响应的内容应立即过期，不应该被缓存。表示客户端不希望使用缓存数据，而是要求服务器始终提供最新的内容。  根本原因是，服务器在返回报文时返回了“Pragma: No-cache”和“Cache-Control: no-cache”字段，表示不应使用缓存来处理请求的响应，即禁用缓存。  但正常的流程是再次访问时加入IF-MODIFIED-SINCE首部，表示浏览器内容最后更新的时间（eg. Thu, 25 Nov 2022 09:48:50 GMT），从而向服务器发送条件GET，询问是否可以直接使用本地缓存。       1. **服务器对较晚的HTTP GET请求的响应中的HTTP状态代码是多少？服务器是否明确返回了文件的内容？请解释。**   答：由于上述浏览器在第二次发送的报文中包含“Cache-Control: max-age=0”字段，因而返回的HTTP状态代码为200，并返回了文件的内容。    事实上，对于条件GET，如果服务器返回的状态码为200，则表明内容需要更新，并明确返回了文件的内容；如果服务器返回的状态码为304，则表示网页的内容没有更新，可以直接使用本地缓存的内容。   1. **利用Wireshark分析TCP协议** 2. 俘获大量的由本地主机到远程服务器的TCP分组 3. 启动浏览器，打开<http://gaia.cs.umass.edu/wireshark-labs/alice.txt>网页，保存ALICE'S ADVENTURES IN WONDERLAND文本到本地。 4. 打开<https://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>网页，输入保存的本地文件。 5. 启动Wireshark，开始分组俘获。 6. 单击“Upload alice.txt file”按钮，将文件上传到gaia.cs.umass.edu服务器。 7. 停止俘获。      1. 浏览追踪信息   在显示筛选规则中输入“tcp”,可以看到在本地主机和服务器之间传输的一系列tcp和http报文，你应该能看到包含SYN报文的三次握手。也可以看到有主机向服务器发送的一个HTTP POST报文和一系列的“http continuation”报文。  **问题思考：**   1. **向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址和TCP 端口号是多少？**   **答：**客户端IP地址为172.20.180.171（电脑重启了IP地址有变化），TCP端口号为57776。  **表格  低可信度描述已自动生成**   1. **Gaia.cs.umass.edu 服务器的 IP 地址是多少？ 对这一连接， 它用来发送和接收 TCP 报文的端口号是多少？**   **答：**服务器的IP地址为128.119.245.12，TCP端口号为80。  **表格  低可信度描述已自动生成**   1. TCP基础   **问题思考：**   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（sequence number）是多少？在该报文段中，是用什么来标示该报文段是SYN报文段的？**  * 客户服务器之间用于初始化TCP连接的TCP SYN报文段序号是0； * 报文段中将Flags中的SYN标志位置1以标识该报文段为SYN报文段。   图形用户界面, 文本, 应用程序  描述已自动生成   1. **服务器向客户端发送的SYN ACK报文段序号是多少？该报文段中，Acknowledgement字段的值是多少？Gaia.cs.umass.edu服务器是如何决定此值的？在该报文段中，是用什么来标示该报文段是SYNACK报文段的？**  * 服务器向客户端发送的SYN ACK报文段序号为0； * 报文段中，Acknowledgement字段的值为1； * Gaia.cs.umass.edu服务器通过将客户端发送过来的报文段决定此ACK的值； * 通过将Flags中ACK和SYN标志位同时置1来标示报文段是SYN ACK报文段。   **图形用户界面, 应用程序, Word  描述已自动生成**   1. **你能从捕获的数据包中分析出TCP三次握手过程吗？**      * **第一次握手：**客户端会向服务器发送一个SYN报文，初始序列号Seq为0，不携带其他任何数据，进入 SYN\_SEND 状态，等待服务器确认连接； * **第二次握手：**服务器收到SYN报文段，确认客户的SYN（），同时发送SYN报文，即SYN ACK报文，服务器进入SYN\_RECV 状态 * **第三次握手：**双方建立起连接，客户端确认收到服务器的SYN ACK报文，回复ACK报文段（SYN置0），同时可以发送数据。  1. **包含HTTP POST命令的TCP报文段的序号是多少？**   **答：**筛选http报文，发现包含HTTP POST命令的TCP报文段序号为151613。  **图形用户界面, 文本, 应用程序  描述已自动生成**   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，那么该TCP连接上的第六个报文段的序号是多少？是何时发送的？该报文段所对应的ACK是何时接收的？**  * 第六个报文段的序号是6093； * 在HTTP POST命令之前，TCP连接建立之后发送的； * 该报文段对应的ACK是在该报文段发送之后，HTTP POST命令之后接收的   **图形用户界面, 应用程序, Word  描述已自动生成**   1. **前六个TCP报文段的长度各是多少？**   **答：**如上图，除第一个TCP报文段为706 Bytes，其余都为1414 Bytes。实际包含数据内容为652 Bytes和1360 Bytes（会有54 Bytes的部等信息）。     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多少？限制发送端的传输以后，接收端的缓存是否仍然不够用？**     **答：**最小的可用缓存空间是239字节。本次实验数据量较小，并未出现接收端缓存不够用的情况，过程中缓存空间还呈现逐渐增大的趋势。  但是限制发送端传输后，接收端的缓存仍然可能出现不够用的现象，原因可能是数据处理速度较慢、接收端缓冲区过小等因素。   1. **在跟踪文件中是否有重传的报文段？进行判断的依据是什么？**   **答：**没有，因为没有出现重复的序列号。   1. **TCP连接的throughput (bytes transferred per unit time)是多少？请写出你的计算过程。 起始时间**     **结束时间**  **答：**数据总长度为字节（算上首部行等字段），发送第一个段的时间节点为，发送最后一个段的时间节点为，发送时间间隔为，因此吞吐率为。   1. **利用Wireshark分析IP协议** 2. 通过执行traceroute 执行捕获数据包 3. 启动Wireshark并开始数据包捕获； 4. 启动pingplotter并“Address to Trace Window”域中输入目的地址。   图形用户界面, 表格  中度可信度描述已自动生成   1. 对捕获的数据包进行分析 2. 第一个主机发出的ICMP Echo Request消息，在packet details窗口展开数据包的Internet Protocol部分。   图形用户界面, 应用程序, Word  描述已自动生成  **数据包大小**  **上层协议**  **源IP地址**  **问题思考：**   1. **你主机的IP地址是什么？**   **答：**172.20.184.210   1. **在IP数据包头中，上层协议（upper layer）字段的值是什么？**   **答：**在IP数据包头中，上层协议字段的值是“Protocol”字段，它的值是1。这表示上层协议是 ICMP（Internet Control Message Protocol）。   1. **IP头有多少字节？该IP数据包的净载为多少字节？并解释你是怎样确定该IP数据包的净载大小的？**   **答：**IP头部大小为20字节（由IP数据包的Header Length字段）。该IP数据包的净载大小为36字节，通过Total Length字段的值为56字节，减去首部字段20字节确定。   1. **该IP数据包分片了吗？解释你是如何确定该IP数据包是否进行了分片**   **答：**没有分片。原因是Flags和Fragment Offset字段均为0，表明该IP数据包不存在偏移，未经过分片。   1. 单击Source按钮对捕获的数据包按源IP地址排序。选择第一个主机发出的ICMP Echo Request消息， 在packet details窗口展开数据包的Internet Protocol部分。   **问题思考：**   1. **你主机发出的一系列ICMP消息中IP数据报中哪些字段总是发生改变？**  * TTL字段，指示数据包可以经过的最大跳数； * Seq字段，标识和追踪不同 ICMP 消息之间的关系，通常以顺序号的方式递增。 * Checksum字段，由数据内容决定，因此会发送变化。  1. **哪些字段必须保持常量？哪些字段必须改变？为什么？**   **答：必须保持常量：**   * 源端口和目标端口，这些字段在整个TCP连接的生命周期内保持不变。它们用于标识通信的源和目标端点； * ICMP消息类型和代码（ICMP Type和Code）； * Identification（标识）字段，用于标识和关联Ping请求和Ping响应，在Ping响应中保持不变，以确保正确地关联响应与请求。   **必须改变：**   * 序列号字段Seq，在不同的Ping消息中通常会递增，以标识不同的Ping请求； * 校验和Checksum字段，由数据内容决定，因此会发生变化。  1. **描述你看到的IP数据包Identification字段值的形式。**   **答：**16位二进制数，用十六进制表示  **图形用户界面, 应用程序, Word  描述已自动生成**   1. 找到由最近的路由器（第一跳）返回主机的ICMP Time-to-live exceeded消息。   图形用户界面, 应用程序  描述已自动生成  **问题思考：**   1. **Identification字段和TTL字段的值是什么？**   **答：**Identifier字段值为0x0000；TTL字段值为   1. **最近的路由器（第一跳）返回给你主机的ICMP Time-to-live exceeded消息中这些值是否保持不变？为什么？**   **答：**Identifier字段和TTL字段均保持不变。相同的Identifier标识是为了分段后组装为一段数据，并不代表信号。每经过一个路由器（一跳），TTL都会减小1，因此在初始TTL相同的情况下，都会返回相同的。   1. 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字节后主机发送的第一个ICMP Echo Request消息。   **问题思考：**   1. **该消息是否被分解成不止一个IP数据报？**   **答：**是的，该消息被分解成两片。       1. **观察第一个IP分片， IP头部的哪些信息表明数据包被进行了分片？ IP头部的哪些信息表明数据包是第一个而不是最后一个分片？ 该分片的长度是多少？**  * Flags字段的值为0x1，其中最后一位为1，表示"More fragments"标志被设置，表明这是一个分片。 * Fragment Offset字段的值为0，表示这是分片的第一个部分，因为第一个分片的偏移量通常为0。 * Total Length字段的值为1500字节，表明该分片的总长度为1500字节，包含1480字节的数据和20字节的首部。  1. 找到在将包大小改为3500字节后你的主机发送的第一个ICMP Echo Request消息。   **问题思考：**   1. **原始数据包被分成了多少片？**   **答：**原始数据包被分成了3片     1. **这些分片中IP数据报头部哪些字段发生了变化？**   **图形用户界面, 文本, 应用程序  描述已自动生成**  **第0片**  **图形用户界面, 文本, 应用程序  描述已自动生成**  **第1片**  **图形用户界面, 文本, 应用程序  描述已自动生成**  **第2片**   * More Fragment 段的值：前两片为 1，最后一片 0； * Fragment Offset字段的值不同：分别为0，1480，2960； * Total Length字段大小不同：前两片为1500，最后一片为540字节；  1. **利用Wireshark分析ARP协议** 2. 利用MS-DOS命令：arp或c:\windows\system32\arp查看主机上ARP缓存的内容   命令提示符  **问题思考：**   1. **说明ARP缓存中每一列的含义是什么？**   答：每一列分别表示 IP 地址所对应的物理地址和类型（动态配置或静态配置）。   1. 在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）   命令提示符   1. 启动Wireshark，开始分组俘获。     **问题思考：**   1. **ARP数据包的格式是怎样的？由几部分构成，各个部分所占的字节数是多少？**   **表格  描述已自动生成**  **答：**ARP数据包格式如图所示，共28字节，具体由9部分构成：   |  |  |  |  | | --- | --- | --- | --- | | 部分名称 | 字节数 | 部分名称 | 字节数 | | 硬件类型 | 2 | 发送端MAC地址 | 6 | | 协议类型 | 2 | 发送端IP地址 | 4 | | 硬件地址长度 | 1 | 目的MAC地址 | 6 | | 协议地址长度 | 1 | 目的IP地址 | 4 | | OP | 2 |  |  |  1. **如何判断一个ARP数据是请求包还是应答包？**   **答：**判断一个ARP分组是ARP请求还是应答的字段是“OP”，当其值为0×0001时是请求包，为0×0002时是应答包。   1. **为什么ARP查询要在广播帧中传送，而ARP响应要在一个有着明确目的局域网地址的帧中传送？**   **答：**ARP查询需要广播，因为它需要询问整个局域网中的设备来找到目标设备的MAC地址，而ARP响应是为了回应特定的ARP查询，局域网中的其他主机不需要此次查询的结果，所以它需要在一个有着明确目的局域网地址的帧中传送。   1. **利用Wireshark分析UDP协议** 2. 启动Wireshark，开始分组捕获； 3. 发送QQ消息给你的好友； 4. 停止Wireshark组捕获； 5. 在显示筛选规则中输入“udp”并展开数据包的细节。   图形用户界面, 表格  中度可信度描述已自动生成  图形用户界面, 表格  中度可信度描述已自动生成  **问题思考：**   1. **消息是基于UDP的还是TCP的？ 答：**UDP。 2. **你的主机ip地址是什么？目的主机ip地址是什么？**   **答：**主机IP地址为172.20.247.120，目的主机IP地址为39.156.132.120**。**   1. **你的主机发送QQ消息的端口号和QQ服务器的端口号分别是多少？**   **答：**主机发送端口为4001，QQ服务器接收端口为8000。   1. **数据报的格式是什么样的？都包含哪些字段，分别占多少字节？**  |  |  |  |  | | --- | --- | --- | --- | | **源端口号** | **目标端口号** | **长度** | **校验和** |   **答：**UDP（用户数据报协议）数据报的格式如图所示，共8字节，具体由4部分构成：   |  |  | | --- | --- | | 部分名称 | 字节数 | | 源端口号（Source Port） | 2 | | 目标端口号（Destination Port） | 2 | | 长度（Length） | 2 | | 校验和（Checksum） | 2 |  1. **为什么你发送一个ICQ数据包后， 服务器又回给你的主机一个ICQ数据包？这UDP的不可靠数据传输有什么联系？对比前面的TCP协议分析，你能看出UDP是无连接的吗？**   **答：**发送一个ICQ数据包后，服务器回复一个数据包，这种回应通常用于确认数据包的接收、处理请求或者传递状态信息。  服务器回复一个数据包仅确认数据包的接收，并不维护连接状态或数据包的顺序，  数据有可能乱序到达或者丢失，为不可靠数据传输。  TCP协议是一种面向连接的协议，它在通信的两端维护一个连接状态，确保数据包的可靠传递和顺序传输。TCP使用序列号、确认号和连接建立过程来实现可靠性和连接状态维护，而UDP则不具备这些特性，因此是无连接的。   1. **利用Wireshark分析DNS协议** 2. 打开浏览器键入：www.baidu.com； 3. 打开Wireshark，启动抓包； 4. 在控制台回车执行完毕后停止抓包，查看wireshark捕获的 DNS 报文如下。   图形用户界面, 应用程序, 表格  描述已自动生成  分析得知，查询的目的地址均为相同的10.128.1.114，经查询是内网IP。   1. **利用Wireshark分析Ethernet数据帧**   以太网数据帧（Ethernet data frame）是在计算机网络中用于在物理层和数据链路层之间传输数据的基本单位，提供了底层的数据传输机制，而TCP和HTTP等协议则建立在它们之上，用于管理连接、数据分段、数据可靠性和应用层通信。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 前同步码 | 目的地址 | 源地址 | 类型 | 数据 | CRC |   以太网数据帧结构如图所示，主要包含了物理地址（MAC地址）和一些控制信息等六个字段，以确保数据帧的传输和完整性，在上述分析过程中已有提及。  文本  描述已自动生成  例如上述数据帧报文，它包含了以下信息：   * Src：源MAC地址，指示了数据帧的发送者f0:9e:4a:04:3c:79 * Dst：目标MAC地址，指示了数据帧的接收者44:ec:ce:d2:ff:c2 * Type：这是数据帧的类型字段，指示数据帧中的数据是IPv4（0x0800），表示数据帧中包含了IPv4协议的数据。   关于MAC地址的解释，主要用于局域网内的通信：   * LG bit（Locally/Group bit）：这位用于指示MAC地址是否是本地地址）还是组地址。0表示是全局唯一地址，通常由硬件制造商分配的； * IG bit（Individual/Group bit）：这位用于指示MAC地址是单播（数据帧只传递给一个设备）还是组播（数据帧被传递给一组设备）。   发送适配器在一个以太网帧中封装了一个IP数据报，并把该帧传递到物理层。接受适配器从物理层收到这个帧，提取出IP数据报，并把该IP数据报传递给网络层。  所有的以太网技术都需网络提供不可靠的无连接服务。 |
| 问题讨论： |
| 1. **为什么访问网页时既有IPv4又有IPv6？**   在发送网页请求时，根据不同的网络配置和服务器支持，可能会出现两种IP版本（IPv4和IPv6）交替使用的情况，这被称为"双栈"（Dual-Stack）支持。  当客户端（通常是浏览器）与服务器通信时，它可以通过DNS解析域名来获取服务器的IP地址。如果域名的DNS记录同时包含IPv4和IPv6地址，那么双栈的客户端会尝试使用IPv6连接。如果IPv6连接失败或服务器不支持IPv6，客户端将回退到IPv4连接。这种方式允许在IPv4和IPv6之间进行无缝的过渡，并确保尽可能多的设备可以访问网络资源。   1. **为什么有的网站再次访问没有IF-MODIFIED-SINCE字段？**   网站和服务器可能使用缓存控制头（如"Cache-Control"和"Expires"）来指导浏览器是否应该缓存资源，以及在何时重新验证资源。如果服务器配置了缓存策略，浏览器可能会根据策略来处理资源的访问，而不需要 "If-Modified-Since" 字段。 |
| 心得体会： |
| 1. 对HTTP、TCP、UDP、IP、DNS等协议的报文格式有了更深入的了解，理解了每个标志位的含义和作用，深入研究了这些协议之间的互动过程和工作原理。 2. 更加了解了网络协议实体之间的交互以及报文传递的一般流程和情景。 3. 了解学习了如何使用Wireshark工具来捕获数据包，以及如何使用PingPlotter工具来进行网络性能分析。 |