山东大学 计算机 学院

计算机网络 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202400130039 | 姓名：张汇智 | | 班级：智能 |
| 实验题目：  Wireshark\_HTTP | | | |
| 实验学时：2h | | 实验日期： 2025. | |
| 实验目的： explore several aspects of the HTTP protocol | | | |
| 硬件环境： AMD ryzen R9 7900HX ; NVIDIA RTX4070LAPTOP ; RAM SAMSUNG 16GB\*2 ; ROM WD770 1T+2T; | | | |
| 软件环境：Windows11 23H2 (KB5056580) | | | |
| 实验步骤与内容：   1. The Basic HTTP GET/response interaction     304NMDF没清除缓存。    不知道为啥删不掉 禁用缓存200 0K DISPLAYED  Is your browser running HTTP version 1.0 or 1.1? **1.1.**  What version of HTTP is the server running? **HTTP/1.1**  2. What languages (if any) does your browser indicate that it can accept to the server?  **zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6 简中 中 英 不列颠英 美英。**  3. What is the IP address of your computer? **101.76.244.249**  Of the gaia.cs.umass.edu server? **128.119.245.12**  4. What is the status code returned from the server to your browser? **200**  5. When was the HTML file that you are retrieving last modified at the server?  **Tue, 16 Sep 2025 07:55:00 GMT**  6. How many bytes of content are being returned to your browser? **128**  7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? **True**  If so, name one.  **Keep alive**  **//呵呵没到解释一分钟我就遇到了304,,,**  //自此笔者上课来到实验室，由以太网变为WLAN   1. The HTTP CONDITIONAL GET/response interaction     无IF modified since请求。    有文件内容    取消禁用缓存，刷新，有IF MODIFIED SINCE 字段，value  Tue, 16 Sep 2025 05:59:02 GMT    304 not modified  没有文本信息  很明显，GET请求有一个若改动则的字段请求，server发现无改动（modified）不就不给新文本了。🤭  3. Retrieving Long Documents    1个。在frame32580中。  Frame 32605中响应了200 OK    4个segment TCP。   1. HTML Documents with Embedded Objects | | | |
| Q16:可以看到发了3GET请求。通往128.119.245.12；128.119.245.12；178.79.137.164。 | | | |
| Q17:时间差2.36s，串行下载的特征，怀疑网络问题，重新抓一下    0.29s认为是**并行下载**。  5 HTTP Authentication | | | |
| 输入账密    Q18: 401 UNAUTHORIZED  Q19: Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n | | | |
| 结论分析与体会：  实验总结与分析  基础请求与响应机制 通过分析简单 HTML 文件的请求与响应，我掌握了 HTTP 报文的结构，包括请求行、请求头、响应状态码、响应头和实体内容。浏览器与服务器之间的通信是建立在 TCP 连接之上的，Wireshark清晰地展示了这一层层封装。  条件 GET 与缓存机制 实验中观察到浏览器在第二次请求时会自动添加 If-Modified-Since 和 If-None-Match 字段，用于判断资源是否更新。这种机制显著提升了网页加载效率，也体现了 HTTP 的轻量级优化能力。  长文档的分段传输 当请求较大的 HTML 文件时，服务器会将响应内容分成多个 TCP 段进行传输。Wireshark 的“TCP segment of a reassembled PDU”提示让我理解了 TCP 的重组机制，以及 HTTP 本身并不具备分段能力。  嵌入对象的并行加载 浏览器在加载包含图片的 HTML 文件时，会并行发起多个 GET 请求，分别向不同服务器请求嵌入资源。这种并行处理方式体现了现代浏览器的高效资源调度能力。  HTTP认证机制的安全隐患 在访问受密码保护的页面时，浏览器通过 Base64 编码将用户名和密码附加在 Authorization 字段中。虽然看似“加密”，但其实只是编码，任何人都可以轻易解码，这让我深刻认识到：HTTP基本认证并不安全，必须依赖 HTTPS 或更高级的认证机制来保护用户隐私。  个人体会  通过本次实验，我不仅掌握了 HTTP 协议的核心机制，还提升了使用 Wireshark 进行网络抓包与分析的能力。每一个数据包背后都隐藏着协议的设计哲学与工程智慧。尤其是在认证部分，我意识到网络安全不仅仅是技术问题，更是用户隐私与信任的基础。  这次实验让我真正做到了“看得见、摸得着”的网络协议学习，也让我对 Web 通信的底层逻辑有了更清晰的认知。 | | | |