

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | HTTP代理服务器的设计与实现 | | | | | |
| 姓名 | 朱明彦 | | 院系 | 计算机科学与技术学院 | | |
| 班级 | 1603109 | | 学号 | 1160300314 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物213 | | 实验时间 | 2018年10月27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 | | | | | | |
|  | | | | | | |

****

|  |
| --- |
| 实验目的： |
| 熟悉并掌握Socket网络变成的过程与技术；深入理解HTTP协议，掌握HTTP代理服务器的基本工作原理；掌握HTTP代理服务器设计用于变成实现的基本技能。 |
| 实验内容： |
| 1. 设计并实验一个基本HTTP代理服务器。要求在制定端口(例如8080)接收来自客户的HTTP请求并且根据其中的URL地址访问该地址所指向的HTTP服务器(原服务器)，接收HTTP服务器的相应报文，并将相应报文转发给对应的客户进行浏览。 2. 设计并实现一个支持Cache功能的HTTP代理服务器。要求能缓存原服务器相应的对象，并能够通过修改请求报文(添加if-modified-since头部行)，向原服务器确认缓存对象是否是最新版本。 3. 扩展HTTP代理服务器，支持网站过滤，允许/不允许访问某些网站。 4. 扩展HTTP代理服务器，支持用户过滤，支持/不支持某些用户访问外部网站。 5. 扩展HTTP代理服务器，支持网站引导，将用户对某个网站的访问引导至一个模拟网站(钓鱼)。 |
| 实验过程： |
| 1. Socket编程的客户端和服务器端的主要步骤    1. socket客户端   对于socket客户端而言，编程的思路比较清晰，明确目的服务器的IP地址、端口号以及传输层协议(TCP or UDP)，根据以上信息构造socket用于通信即可；在接受消息后，注意适时关闭socket连接。   * 1. socket服务器端   对于socket服务器端而言，思路也比较固定，主要区分一下UDP协议和TCP协议上的编程即可。  c) 对于UDP协议上的通信，无需提前建立连接，只需在开始时建立相应的socket，进入无限循环，接收消息后直接与源地址进行通信即可。  d) 对于TCP协议上的通信，服务器需要有一个socket负责控制，在进入无限循环前建立绑定指定的端口号，并在无限循环内，对于每一个连接新建TCP连接与源主机进行通信即可。  2. HTTP代理服务的基本原理  本次实验中实现的HTTP代理服务器，主要是通过转发源主机的HTTP请求报文至目的服务器，并且将接收到的HTTP相应报文，再次转发到源主机上实现的。  3. HTTP代理服务器的程序流程图  见下页图    4. 实现HTTP代理服务器的关键技术及解决方案  a) 解析HTTP请求报文  解析HTTP的请求报文主要是解析HTTP的头部行，在这里我们使用`\r\n`对整个请求报文进行划分，得到的就是每一个头部行的信息。  其中最为重要的就是头部行的第一行，即**Request Line**，标注着method、URL和协议的版本号，并使用1个空格进行划分，如下：    可以通过解析Request Line获得目的服务器的URL。  b) 实现请求报文和响应报文的转发  在proxy的实现中主要涉及了3种socket，分别为：  1. 代理服务器用于处理TCP请求的socket，在本次实验中，将这个socket的端口绑定在12138端口；  2. 用于直接与源主机连接的socket，用于接受来自源主机的HTTP请求报文和从proxy将HTTP的响应报文转发至源主机，对于源主机的TCP请求，在一个线程中开启1个此种socket用于处理；  3. 其三为proxy代源主机与目的主机进行连接的socket，主要负责将源主机的HTTP请求转发发送至其目的主机，并获取从目的主机返回的HTTP响应报文。  c) cache的实现  对于cache功能的实现，在这次实验中采取了一种比较朴素的思想，即将所有的请求的文件保存在磁盘上。当源主机再次访问相同的文件时，proxy首先获得缓存文件的时间，然后构造条件GET方法(增加if-modified-since头部)访问目的服务器，如果得到的结果为304，则不再从目的服务器获得请求文件，转而由本地磁盘直接将信息读出发送给源主机；否则，说明目标文件已经发生了变化或者缺少Last-Modified头部，此时认为请求的对象\*\*发生了改变(即使可能没有发生改变)\*\*，继续向目的服务器发送请求，并更改本地缓存。  d) 钓鱼、限制用户和限制网址的实现  本质上，对于上面三种功能的实现，都是基于对HTTP请求报文的解析，从相似的角度解决。总的来说，使用`JSON`配置文件，配置相应的限制信息，格式如下：    其中的host对应的是网址的限制，IP对应的是用户的限制以及fishing对应的是钓鱼网站的实现。  实现是基于配置信息，根据HTTP请求报文中的URL和源IP地址，**如果其出现在相应的配置信息中，就将其禁用**，并返回本地对应的html文件，告知用户相应信息。 |
| 实验结果： |
| 1.**基本功能**  对于代理服务器的基本功能实现，访问http://jwts.hit.edu.cn，访问的结果如下：    可以看到左侧为访问的结果，右侧可以看到相应的HTTP请求报文。  2.**网站限制**  对于代理服务器的网站限制，此处我们遵循上面提到的配置文件，访问http://www.whu.edu.cn，结果如下：    即理应得到的网页为武汉大学官网，但是由于网页限制得到的是限制后得到的404页面。  3.**用户限制**  对于代理服务器的用户限制，仍然遵循上面提到的配置文件，直接在限制本地环回地址，访问的结果是：    访问http://today.hit.edu.cn，原本正常可以访问，但是在此处我们进行了限制，所以得到的是403 Forbidden。  4.**cache的实现**  在上面的实验中我们访问了http://jwts.hit.edu.cn`，因此得到了本地的缓存文件，如下：    但是由于http://jwts.hit.edu.cn本身返回的报文中不包含Last-Modified头部，所以实现读取本地缓存，只有对应的css文件可以返回304代码，所以访问的结果如下：    根据右边命令行最下面一行，可以看到，css文件从本地读出。  5.**"钓鱼"的实现**  在这里钓鱼的实现，主要是将配置文件中对应的网站进行重定向，此处访问http://www.fudan.edu.cn，结果如下：  可以看到，原本访问复旦大学的主页，但是我们进行重定向，得到的是百度的主页，实现了钓鱼的功能。 |
| 问题讨论： |
| 在实验中禁用IP和某些网站的实现，原本的实现是直接丢弃HTTP请求报文，但是这样会使Firefox陷入对禁用网站的无限请求。因此，在之后改为，在本地构建1个简单的html文件，将其作为禁用的结果返回源主机，就可以解决。 |
| 心得体会： |
| 实验中实现的式基本的HTTP代理服务器，仅仅能实现的是对于HTTP协议的某些网站的访问，对于一些https协议的网站，还无法处理。与实际中所使用的代理服务器相比差距还很大，但是通过实现基本的代理服务器的功能，了解了代理服务器的基本工作原理，为了解socket编程提供了很多帮助。 |