



哈尔滨工业大学
Harbin Institute of Technology

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	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 10 月 31 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 Socket 网络编程的过程与技术; 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 代理服务器设计与编程实现的基本技能。

实验内容:

1. 设计并实现一个基本 HTTP 代理服务器。要求在指定端口(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 服务器的响应报文,并将响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响应的对象,并能够通过修改请求报文(添加 if-modified-since 头行),向原服务器确认缓存对象是否是最新版本。
3. 网站过滤:允许/不允许访问某些网站;
4. 用户过滤:支持/不支持某些用户访问外部网站;
5. 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. 浏览器使用代理**


在实验过程中, 为了方便测试, 我使用 Proxy SwitchyOmega 插件进行代理。



Proxy SwitchyOmega
by Firefox user 12962115

Manage and switch between multiple proxies quickly & easily.

 Remove

 This add-on is not actively monitored for security by Mozilla. Make sure you trust it before installing. [Learn more](#)

需要使用代理时可以使用不同的代理方案, 方便地使用插件进行切换。

2. Socket 编程的客户端和服务端的主要步骤

(1) 对于 Socket 客户端而言, 明确目的服务器的 IP 地址、端口号以及传输层协议(TCP 或 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。

(2) 对于 Socket 服务器端而言, 思路同客户端类似, 主要区分一下 UDP 协议和 TCP 协议上的编程即可。

(3) 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 , 进入忙循环, 接收消息后直接与源地址进行通信即可。

(4) 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, 并在忙循环内, 对于每一个连接新建 TCP 连接与源主机进行通信即可。

3. HTTP 代理服务器的基本原理

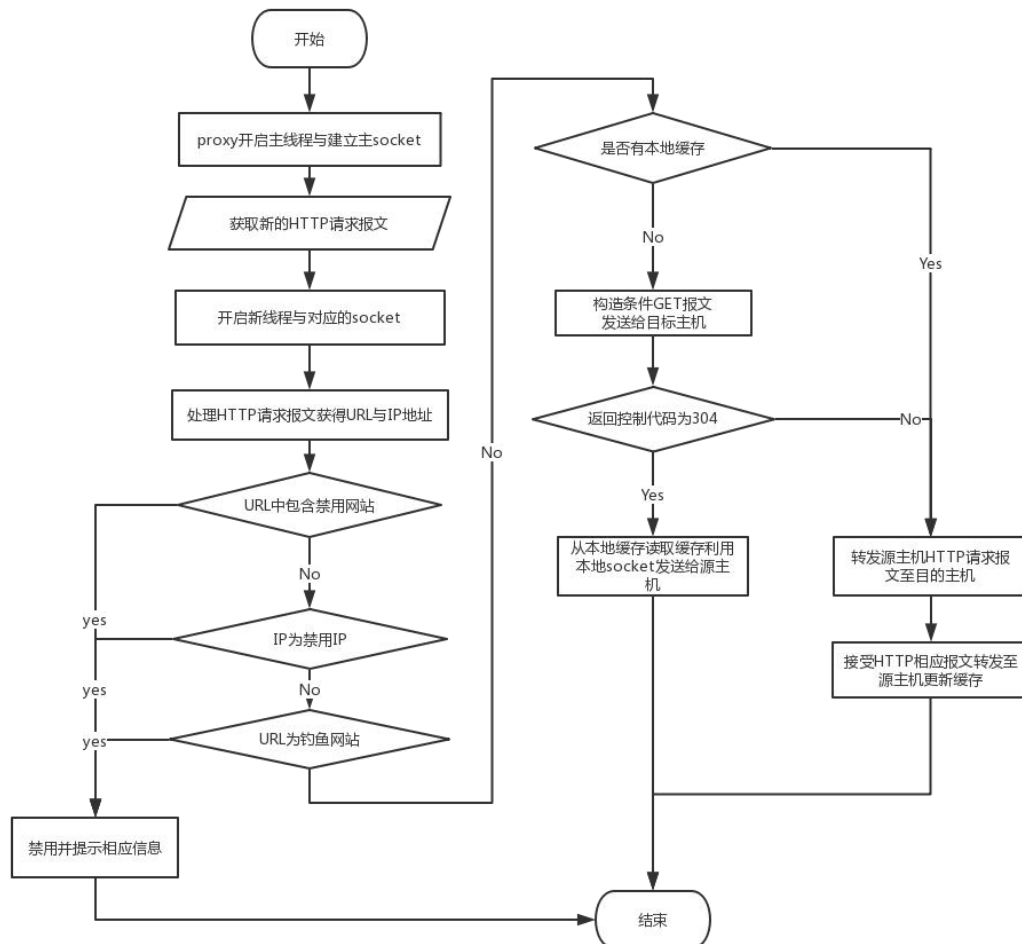
HTTP 代理服务器的基本原理是利用 HTTP 代理服务器转发源主机的 HTTP 请求报文至目的服务器, 在接收到对应的 HTTP 报文后再次将其转发到源主机上。

若需要实现缓存功能, 则需要缓存目的服务器的 HTTP 报文, 在源主机再次访问时代理服务器向目的服务器判断是否产生过变化, 若无变化则直接返回本地缓存; 否则再次请求

报文。

网站和用户过滤则是判断源主机和目的服务器的 IP 地址来决定是否需要转发，网站引导则是修改目的服务器的地址。

4. HTTP 代理服务器的程序流程图



5. 实现 HTTP 代理服务器的关键技术及解决方案

(1) 解析 HTTP 报文

对于 HTTP 报文，代理服务器需要做的主要是解析出 HTTP 报文的头部信息，从头部信息中解析出目的服务器的主机名和 URL 地址。我们将整个请求报文按照"
"进行划分。

```

# 将报文进行划分
headers = message.split('\r\n')
    
```

之后在 headers 的第一行（即 Request Line）中解析出 URL 地址，从 headers 中找到 Host 键值对解析出目的服务器的主机名。

```
# 获取目的服务器的主机名和端口
hostname, port = None, 80
for header in headers:
    if header[:4].lower() == 'host':
        arr = header[6:].split(':')
        hostname = arr[0]
        if len(arr) == 2:
            port = int(arr[1])
        break
# 解析出 URL 地址
request_line = headers[0].strip().split()
url = request_line[1].rstrip('\n')
```

(2) 将源主机和目的服务器的 HTTP 报文进行转发

在代理服务器的实现中主要涉及了三种 Socket，分别为：

其一为代理服务器用于处理 TCP 请求的 Socket。在本次实验中，将这个 Socket 的默认绑定在本地的 19267 端口；

其二为用于直接与源主机连接的 Socket，用于接受来自源主机的 HTTP 请求报文和从代理服务器将 HTTP 的响应报文转发至源主机。对于源主机的一个 TCP 请求，服务器会启动一个线程中并开启一个此种 Socket 用于处理该连接；

其三为代理服务器代替源主机与目的服务器进行连接的 Socket，主要负责将源主机的 HTTP 请求转发发送至其目的服务器，并获取返回的 HTTP 响应报文。

(3) 缓存机制的实现

对于缓存功能的实现，在这次实验中我使用了一种比较朴素的思想，即将所有的请求目的服务器返回 HTTP 报文以二进制文件的形式保存在磁盘上。

当源主机再次访问相同的文件时，代理服务器首先获得上次保存缓存文件的时间，然后构造条件 GET 方法（增加 If-Modified-Since 头部）访问目的服务器，如果得到的状态码为 304，则不再从目的服务器获得请求文件，转而在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，此时应当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。

(4) 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。

若访问的主机名为限制的网址主机或者用户的 IP 地址为限制的用户地址，则不访问目的服务器而直接返回 404 状态。

若访问的主机名为需要引导的主机，则替换需要访问的主机名，其访问路径不变（便于加载资源文件），之后从新的 URL 地址中获取 HTTP 报文。

```
ban_user:
  # - "127.0.0.1";
ban_web:
  - "ban.com"
  - "today.hit.edu.cn"
fishing:
  "fishing.com": "www.hit.edu.cn"
```

在此次实验中，我使用 YAML 格式的文件来存储对应的配置信息。其中 `ban_user` 为限制用户的 IP 地址，`ban_web` 为限制网址的主机名，`fishing` 为引导网站对应的键值对。

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://today.hit.edu.cn/>，访问的结果如下：

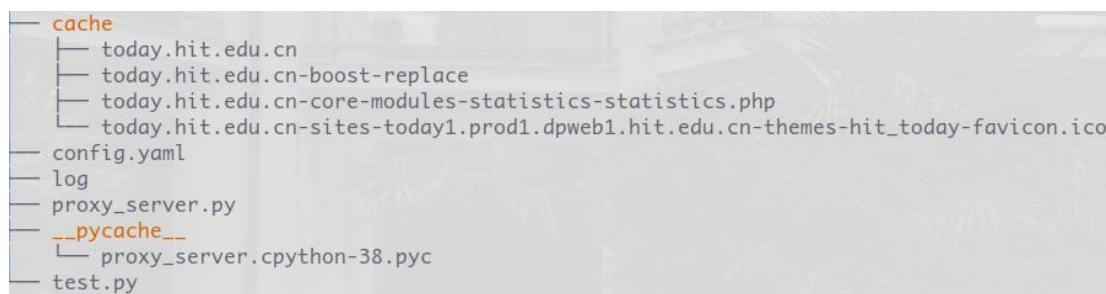


The screenshot displays a web browser window with two panes. The left pane shows the HTTP request and response logs, including headers like Host, Proxy-Connection, User-Agent, and cookies. The right pane shows the website content, which includes a blue header with the HIT logo and a list of recommended articles under the heading '网站推荐'.

可以看到右侧为访问的结果，左侧可以看到相应的 HTTP 请求报文和代理日志。

2. 缓存机制

在上面我们访问了今日哈工大，本地会有对应的缓存文件



The screenshot shows a file explorer window with a tree view of the cache directory. The root directory is 'cache', and it contains several subdirectories and files, including 'today.hit.edu.cn', 'today.hit.edu.cn-boost-replace', 'today.hit.edu.cn-core-modules-statistics-statistics.php', 'today.hit.edu.cn-sites-today1.prod1.dpweb1.hit.edu.cn-themes-hit_today-favicon.ico', 'config.yaml', 'log', 'proxy_server.py', '__pycache__', 'proxy_server.cpython-38.pyc', and 'test.py'.

再次访问今日哈工大，返回的就是缓存文件中内容。

```
[CACHE INFO] Read from file cache/today.hit.edu.cn-sites-today1.prod1.dpweb1.hit.edu.cn-themes-hit_today-favicon.ico
[CACHE] Get http://today.hit.edu.cn/sites/today1.prod1.dpweb1.hit.edu.cn/themes-hit_today-favicon.ico from cache file.
```

3. 网站限制

```
(28) < vi config_vml
      < cycle@arch: ~/Documents/CS-Homework/计算机网络/Lab1/src <master X [?]> [03f5a68] (0:27)
38)
      < py test.py [03f5a68]
[INFO] Start Proxy Server on 127.0.0.1 19267
GET http://today.hit.edu.cn/ HTTP/1.1
Host: today.hit.edu.cn
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.1367356385.1594734110

[REJECT NOSENAME] today.hit.edu.cn
```

该网站无法访问

在配置文件中直接在限制本地环回地址，访问的结果是：

```
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.1367356385.1594734110

[REJECT USER] 127.0.0.1
GET http://cs.hit.edu.cn/ HTTP/1.1
Host: cs.hit.edu.cn
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,application/signed-exchange;v=b3;q=0.9
Accept-encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.1367356385.1594734110

[REJECT USER] 127.0.0.1
```

你被禁止访问网络

在配置文件中配置对应的选项后，我们访问 <http://fishing.com/> 其结果如下：

```
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: uid=6cf0627d-8140-4867-8ffc-0b22ca547b58; _ga=GA1.2.2057531077.1604139406; _fbp=fb.1.1604139407300.146819039; _hjid=44ec116-6c64-48b4-bd74-52b542dac8
[FIGISH] fishing.com www.hit.edu.cn
GET http://fishing.com/_upload/tpl/02/bc/700/template700/images/favicon.ico HTTP/1.1
Host: fishing.com
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://fishing.com/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: uid=6cf0627d-8140-4867-8ffc-0b22ca547b58; _ga=GA1.2.2057531077.1604139406; _fbp=fb.1.1604139407300.146819039; _hjid=44ec116-6c64-48b4-bd74-52b542dac8
[FIGISH] fishing.com www.hit.edu.cn
```

可以看到，原本访问的 <http://fishing.com/>，但是我们进行重定向，得到的是工大的主页，实现了钓鱼的功能。

在实验的过程中,我发现不同的网址其编码格式不同,还有部分资源文件难以编码为文本。这导致我难以确定一个通用的缓存文件编码格式。最后我使用二进制文件的方法存储缓存文件,实现了报文的缓存同时避免了编码问题。

实验过程中，我实现了一个最基本的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，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，我了解了代理服务器的基本工作原理，对 Socket 编程有了初步的理解，对于计算机网络的认识更为深刻。