

# 哈尔滨工业大学

## <<计算机网络>>

### 实验报告

(2018 年度春季学期)

姓名:	
学号:	
学院:	计算机科学与技术学院
教师:	

## 实验一 HTTP 代理服务器的设计与实现

### 一、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。

### 二、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) 扩展 HTTP 代理服务器，支持如下功能：

- a) 网站过滤：允许/不允许访问某些网站；
- b) 用户过滤：支持/不支持某些用户访问外部网站；
- c) 网站引导：将用户对某个网站的访问引导至一个模拟网站（钓鱼网站）

### 三、实验过程及结果

#### 1. Socket 编程的客户端和服务端主要步骤：

#### 服务器：

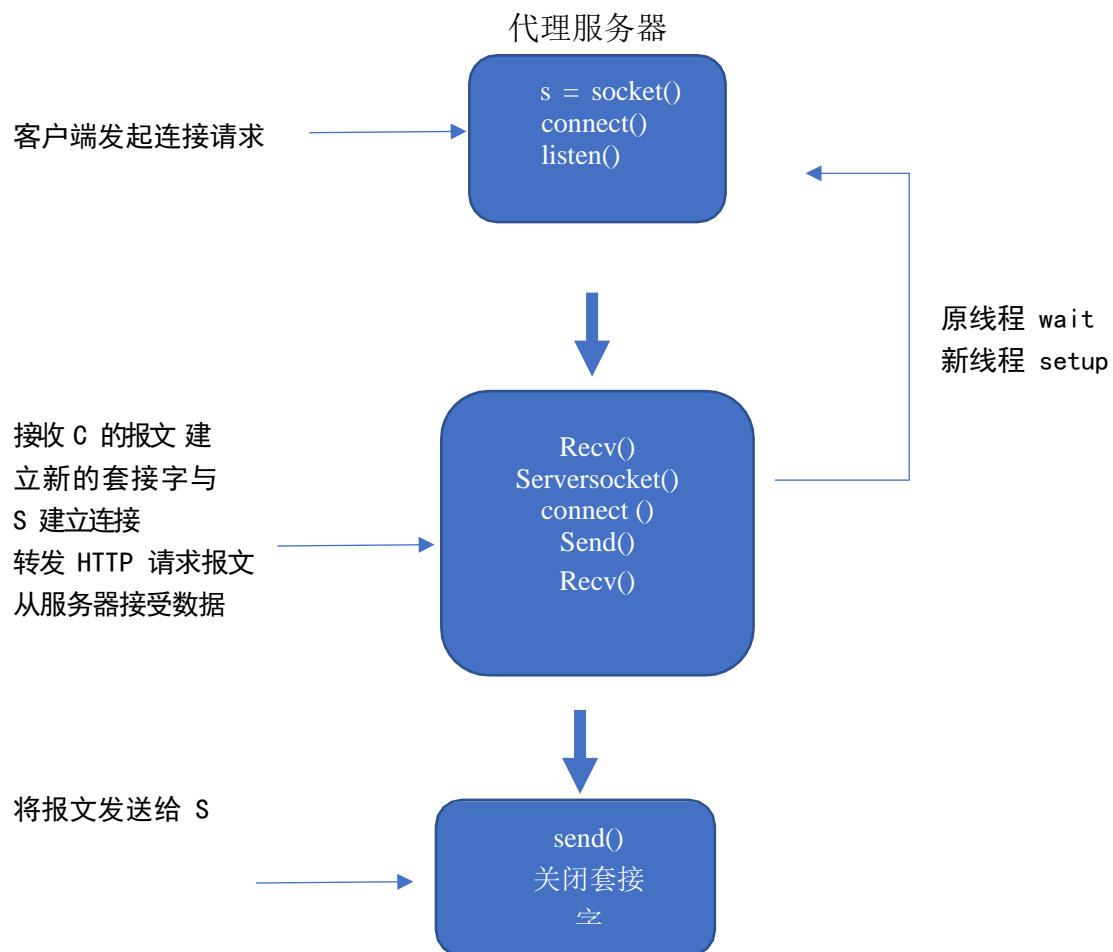
- (1) 创建一个 socket，用函数 `socket()`；
- (2) 设置 socket 属性，用函数 `setsockopt()`；\* 可选
- (3) 绑定 IP 地址、端口等信息到 socket 上，用函数 `bind()`；
- (4) 开启监听，用函数 `listen()`；
- (5) 接收客户端上来的连接，用函数 `accept()`；
- (6) 收发数据，用函数 `send()` 和 `recv()`，或者 `read()` 和 `write()`；

- (7) 关闭网络连接;
- (8) 关闭监听;

## 客户端:

- (1) 创建一个 socket, 用函数 `socket()`;
- (2) 设置 socket 属性, 用函数 `setsockopt()`; \* 可选
- (3) 绑定 IP 地址、端口等信息到 socket 上, 用函数 `bind()`; \* 可选
- (4) 设置要连接的对方的 IP 地址和端口等属性;
- (5) 连接服务器, 用函数 `connect()`;
- (6) 收发数据, 用函数 `send()` 和 `recv()`, 或者 `read()` 和 `write()`;
- (7) 关闭网络连接;

## 2. 总体设计



### 3. 关键技术的解析和设计

#### 1. web 缓存

建立 cache 结构体，存储 cache 的数据。我设计的 cache 结构存储了第一次客户端发来的请求报文，以及客户主机要访问的目的主机发回数据报文，并从中提取 last-modified 字段并存储。在 ProxyThread 函数中，当收到请求报文时，在对报文头部处理后，首先在 cache 中查找该请求。如果找到，在请求报文之中-第三行加入 if-modified-since: date，发送给服务器。接收到服务器返回的响应报文，对响应报文进行处理，检查头部是否为 304 not modified，如果是，直接将 cache 中的响应报文返回给浏览器，如果不是，首先将该响应报文存入 cache 中，即对 cache 进行更新—仍存储在之前的那个位置上，然后将响应报文返回给浏览器。如果在 cache 中没有找到该请求，将处理后的请求报文头部存入 Cache，得到响应报文之后，对响应报文进行解析，得到 date，然后将响应报文和 date 存入 cache。

#### 2. 网站引导（钓鱼）

当user 在访问某一个网站时，http 代理服务器将其引导到另外一个网站上；

实现方法有 2 种：

- 1) 当用户发送请求报文后，经过对 host 进行匹配，确定是否是被钓鱼的网站，如果是，此时向客户端发送一个携带钓鱼后的网站地址的 302 报文。接收到 302 报文的客户端就会发送一个对钓鱼网站的请求报文，代理服务器会回复钓鱼的网站信息；
- 2) 当用户发出访问被钓鱼的网站请求后，我们事先存储了正常访问钓鱼网站的请求报文，经过将 IP 等头部信息替换之后，代理服务器将把访问原主机的报文丢弃，然后使用已经更改好的钓鱼网站的请求报文发送请求。那么，这次回复的也是回复钓鱼的网站信息；

#### 2. 用户过滤：

当建立起浏览器和代理服务器的链接时，解析出请求报文头部，得到浏览器的地址信息，也就得到浏览器端的 ip 地址，与被禁用户 ip 比较，如果相同，认为链接没有建立，不进行任何操作，代理服务器等待下一次访问请求；

#### 3. 网站过滤：

设置一个全局数组，存放的是被禁止访问的网站的主机。在 ProxyThread 函数中解析出请求报文头部之后，将请求报文头部中的 host 与全局数组中的数据进行比较，如果出现相同的表示访问的网站被禁止访问，直接跳转到结束位置。

## 4. 实验结果：

### 1. 基本模块

当浏览器输入 `softmargin.com` 后可以看到代理服务器的后台接受了请求，浏览器正确显示了页面：

```
C:\Users\Surflyan\Documents\Visual Studio 2015\Projects\Proxy2\Debug\Proxy2.exe
代理服务器正在启动
初始化...
代理服务器正在运行, 监听端口 1028
CONNECT hm.baidu.com:443 HTTP/1.1

关闭套接字
CONNECT hm.baidu.com:443 HTTP/1.1

关闭套接字
CONNECT hm.baidu.com:443 HTTP/1.1

关闭套接字
CONNECT hm.baidu.com:443 HTTP/1.1

关闭套接字
GET http://www.softmargin.com/ HTTP/1.1
http://www.softmargin.com/
代理连接主机 www.softmargin.com 成功
关闭套接字
CONNECT cdn.staticfile.org:443 HTTP/1.1

关闭套接字
GET http://api.share.baidu.com/s.gif?l=http://www.softmargin.com/ HTTP/1.1
http://api.share.baidu.com/s.gif?l=http://www.softmargin.com/
代理连接主机 api.share.baidu.com 成功
关闭套接字
CONNECT clients4.google.com:443 HTTP/1.1
```



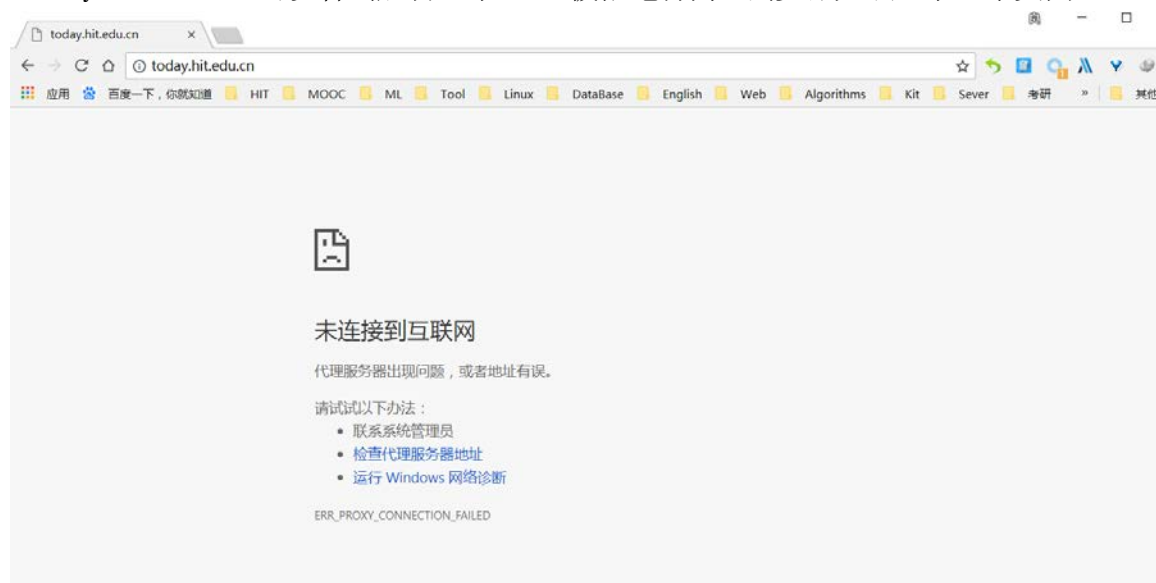
## 2. 用户过滤

建立禁止主机数组。通过在调用 `accept` 函数时获知请求方的 `ip`，如果对方的 `IP` 为禁止主机数组里的 `IP`，则 `FORBIDDEN`。

```
C:\Users\Surflyan\Documents\Visual Studio 2015\Projects\Proxy2\Debug\Proxy2.exe
代理服务器正在启动
初始化...
代理服务器正在运行，监听端口 1028
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
此用户访问受限！
```

## 3. 网站过滤

网站过滤就是通过对请求的 `host` 进行比对后，确定是否屏蔽。我屏蔽了 `today.hit.edu.cn`。可以看到后台显示 `url` 被拒绝访问，浏览器无法显示正常页面。



## 4. 钓鱼网站

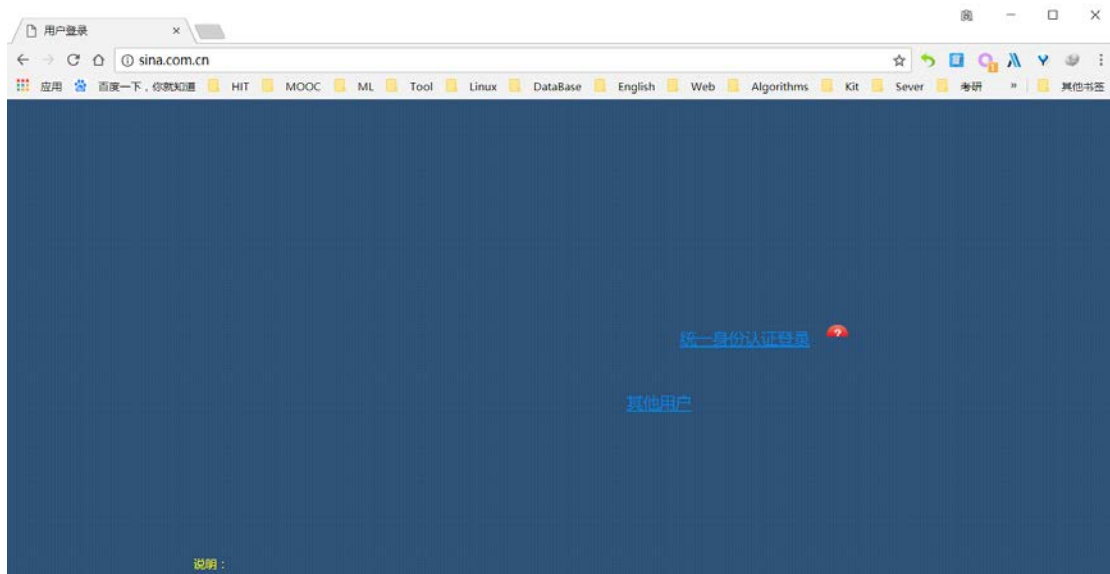
当user 在访问某一个网站时, http 代理服务器将其引导到另外一个网站上;

实现方法有 2 种:

1) 当用户发送请求报文后, 经过对 host 经行匹配, 确定是否是被钓鱼的网站, 如果是, 此时向客户端发送一个携带钓鱼后的网站地址的 302 报文。接收到 302 报文的客户端就会发送一个对钓鱼网站的请求报文, 代理服务器会回复钓鱼的网站信息;

2) 当用户发出访问被钓鱼的网站的请求后, 我们事先存储了正常访问钓鱼网站的请求报文, 经过将 IP 等头部信息替换之后, 代理服务器将把访问原主机的报文丢弃, 然后使用已经更改好的钓鱼网站的请求报文发送请求。那么, 这次回复的也是回复钓鱼的网站信息;

当输入 sina.com.cn 时, 被劫持到 jwt.s.hit.edu.cn:



## 5. cache 验证

第二次访问 softmargin.com, 代理端为请求报文加入 If-Modified-Since: 字段。目的服务器返回 304 Not Modified 报文。

```
C:\Users\Surflyan\Documents\Visual Studio 2015\Projects\Proxy2\Debug\Proxy2.exe
GET http://www.softmargin.com/ HTTP/1.1
http://www.softmargin.com/
关闭套接字
代理连接主机 www.softmargin.com 成功
-----条件性Get报文-----
GET http://www.softmargin.com/ HTTP/1.1
Host: www.softmargin.com
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7
Cookie: Hm_lvt_5580aadc4c87d87d2202096d9e278ef3=1514715188; bdshare_firsttime=1514724030754; _ga=GA1.3.1757834714.1517234528; Hm_lvt_3747566a81b32444a5bb7052b11946c3=1524618407,1524831755,1524904492,1525221324; _gid=GA1.3.450394259.1525485820; Hm_lpv_3747566a81b32444a5bb7052b11946c3=1525485820; _gat=1
If-None-Match: W/"5a6f2948-794c"
If-Modified-Since: Mon, 29 Jan 2018 14:01:44 GMT
```

```
-----Server返回报文-----
HTTP/1.1 304 Not Modified
Server: nginx/1.10.3 (Ubuntu)
Date: Sat, 05 May 2018 02:10:58 GMT
Last-Modified: Mon, 29 Jan 2018 14:01:44 GMT
Connection: keep-alive
ETag: "5a6f2948-794c"

0.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7
Cookie: Hm_lvt_5580aadc4c87d87d2202096d9e278ef3=1514715188; bdshare_firsttime=1514724030754; _ga=GA1.3.1757834714.1517234528; Hm_lvt_3747566a81b32444a5bb7052b11946c3=1524618407,1524831755,1524904492,1525221324; _gid=GA1.3.450394259.1525485820; Hm_lpvt_3747566a81b32444a5bb7052b11946c3=1525485820; _gat=1
If-None-Match: W/"5a6f2948-794c"
If-Modified-Since: Mon, 29 Jan 2018 14:01:44 GMT

将cache中的数据返回给客户端
```

## 四、实验心得

通过此次实验，我更加深刻的理解 HTTP 协议的原理以及工作方式，熟悉了 Socket 网络编程的技术，掌握了 HTTP 代理服务器的设计与实现。相对清楚的知道了在浏览器里键入一个 URL 是如何获得返回的数据报文的流程。也对一些网络攻击有了一点初步的体会。更重要的是通过此次实验，极大的激发了我对计算机网络的兴趣。