



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	任永鹏		院系	计算机		
班级	1737102		学号	1173710107		
任课教师	张英涛		指导教师	张英涛		
实验地点	格物 213		实验时间	周六 12 节		
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						



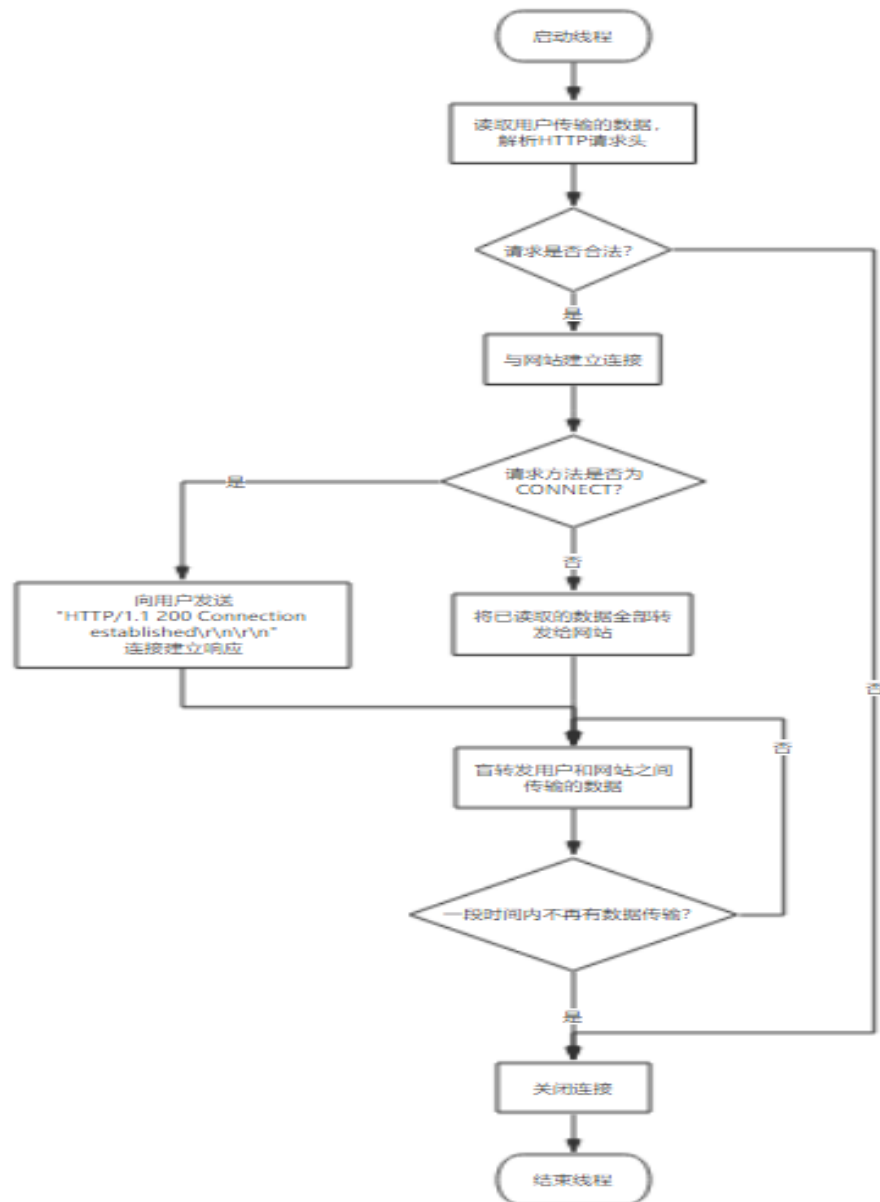
计算机科学与技术学院 SINCE 1956...  
School of Computer Science and Technology



```

static void CloneStream(ByteArrayOutputStream CloneResult, InputStream InputStream) throws IOException {
    byte[] buffer = new byte[1024];
    int length;
    while ((length = InputStream.read(buffer)) != -1) {
        CloneResult.write(buffer, 0, length);
    }
    CloneResult.flush();
}
    
```

流程框图:



HTTPS代理服务器的实现与HTTP代理服务器的实现大体相同,但是在转发过程中使用的是HTTP协议的隧道技术。在一开始Client建立与代理服务器的连接之后,发送Connect HTTP数据报,告知代理服务器 目标服务器的 host和port,然后代理服务器需要与制定的host和port建立TCP链接,如果链接成功,就返回给Client一个HTTP Response:HTTP/1.1 200 Connection established\r\n\r\n,告知Client连接成功,之后同HTTP代理服务器,将Client发送给代理服务器的信息全部交付给目标服务器,将目标服务器发送来的信息全部交付给Client。

在代理服务器收到来自Server的HTTP Response 时,将请求的URL作为HashMap的key,将响应的内容作为value。当客户端再次请求相同的URL时,代理服务器根据URL取出对应的缓存信息,提取出其中的Last-Modified字段的日期,向目标服务器发送一个HTTP请求, 使用GET方法,包含If-Modified-Since: 日期, 如果服务器返回304状态的响应,则代理服务器直接将缓存的副本返回给Client,如果目标服务器返回的是正常的HTTP响应,则代理服务器将新的数据返回给Client,并更新对应的缓存。

拓展代理服务器功能

(a) 为了实现屏蔽用户访问某些网站的功能,我建立了一张屏蔽Host的Json格式的文件,然后使用程序读取并解析这个Json文件,从中提取出需要屏蔽的Host,在Client建立链接并发送请求后,解析出请求的Host,判断该Host是否在屏蔽列表中,如果在其中,就立即中断与Client的连接,不再代理转发请求。

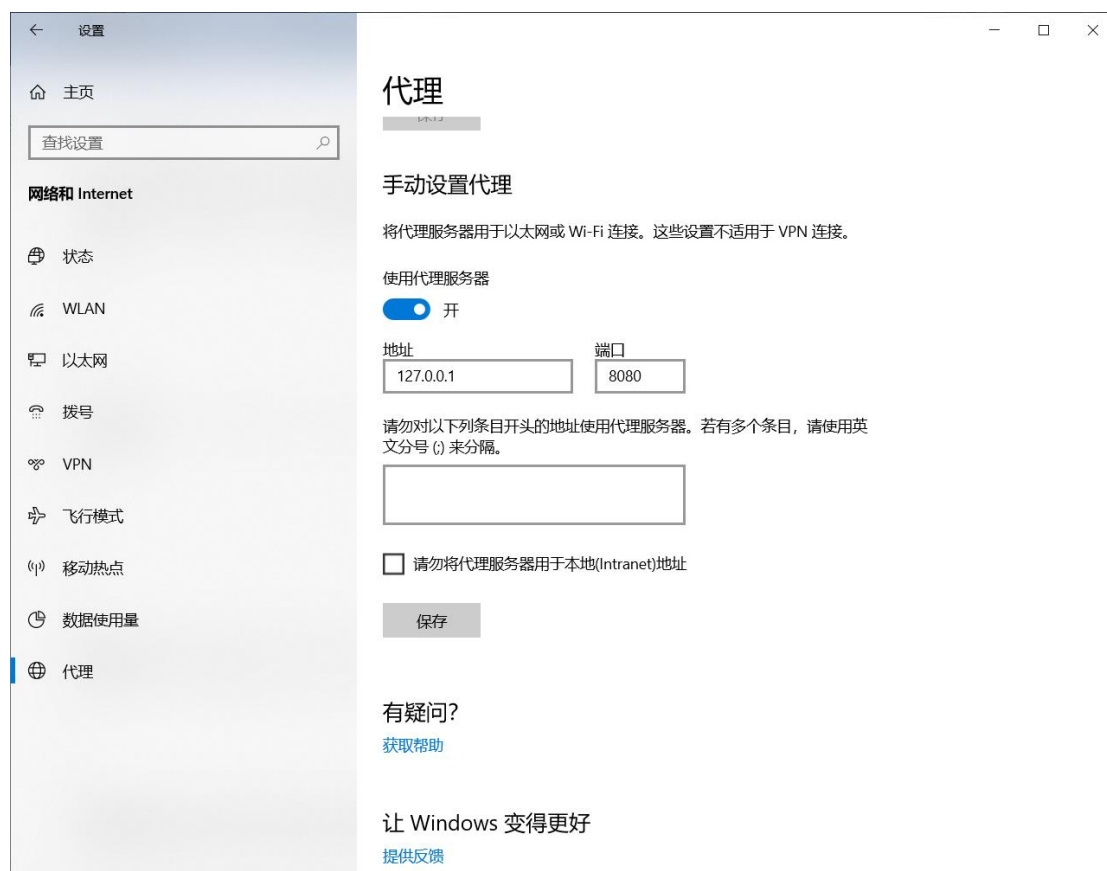
(b) 为了实现屏蔽某些用户访问代理服务器,我在屏蔽Host的列表中建立了一张屏蔽制定用户的列表,处理方式同上,在Client与代理服务器建立链接后,首先检查Client Socket的Host,如果在屏蔽的列表中,就立即中断与该Client的连接。

(c) 钓鱼网站的实现: 在Client与代理服务器建立连接后,并不将请求转发给目标服务器,而是直接返回一个自己构造好的HTTP响应,然后Client就收到了一个钓鱼网站,而不是真正的网站。

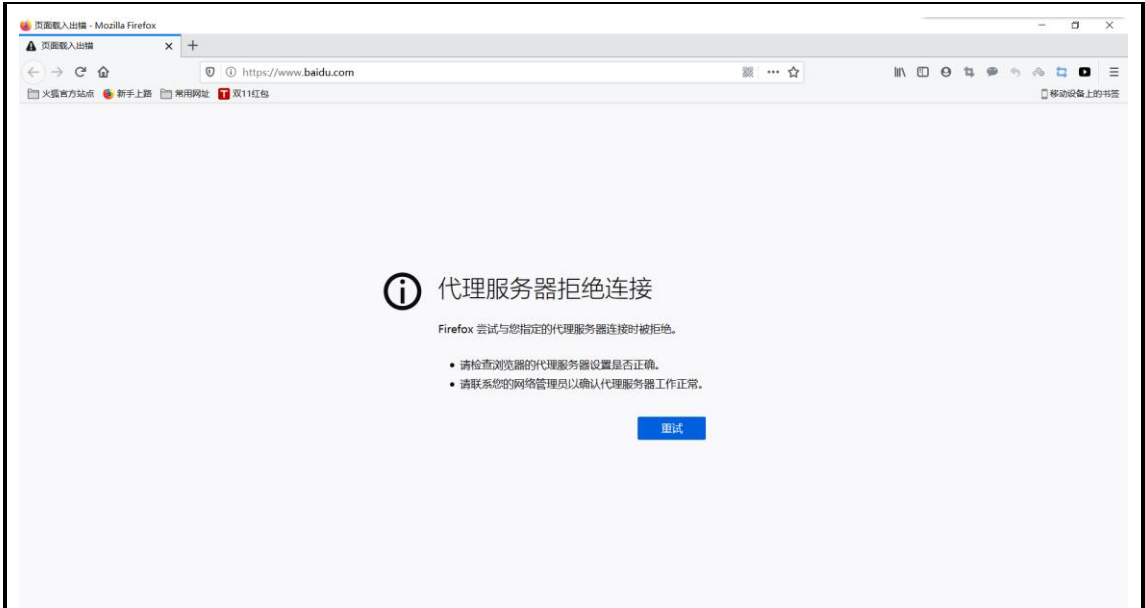
实验结果:

采用演示截图、文字说明等方式, 给出本次实验的实验结果。

开启代理:

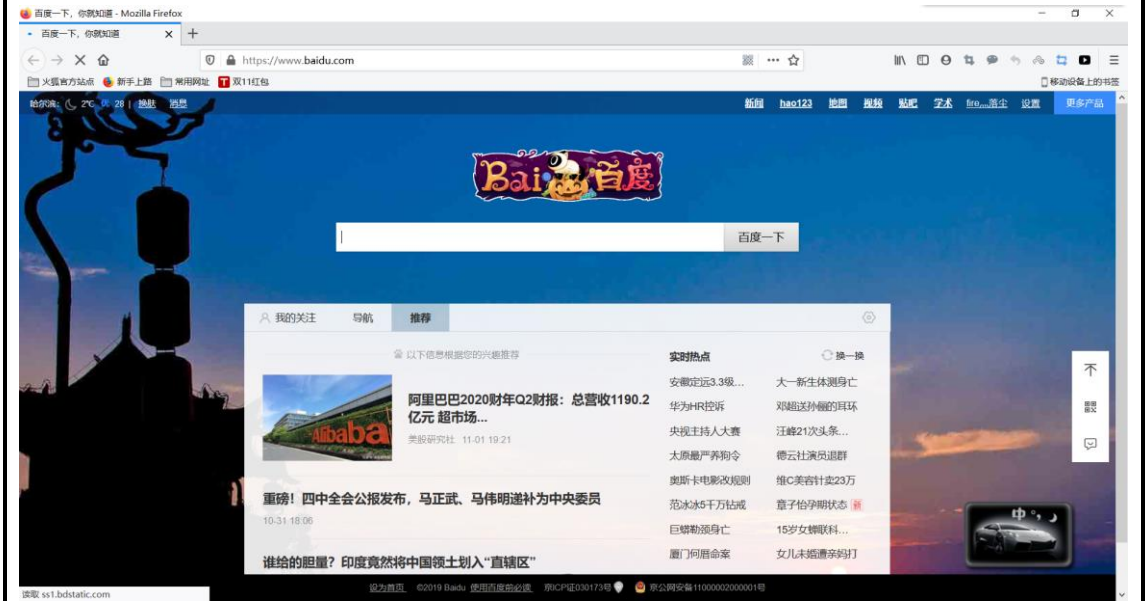


打开浏览器效果

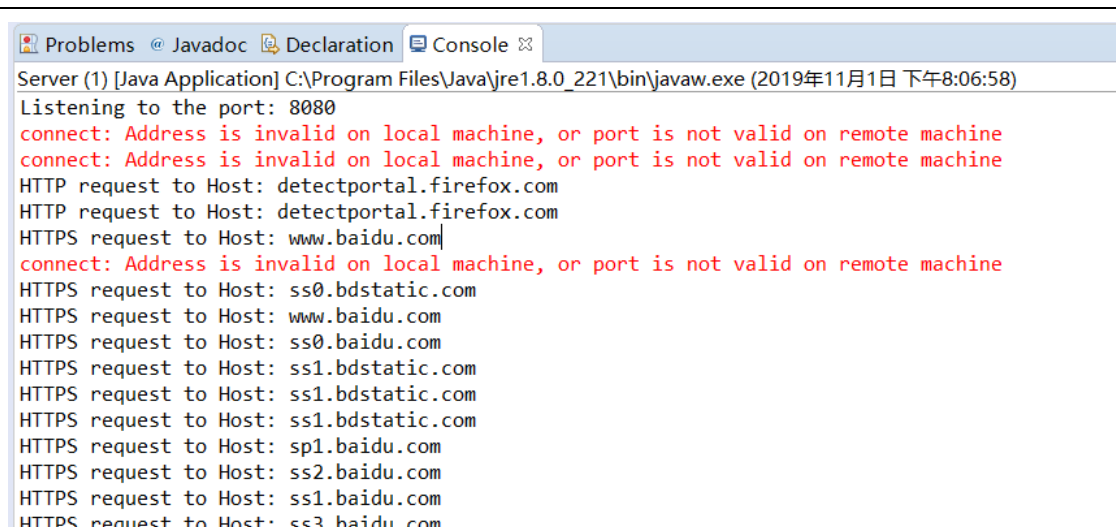


打开eclipse运行代码

打开浏览器



Eclipse控制台信息

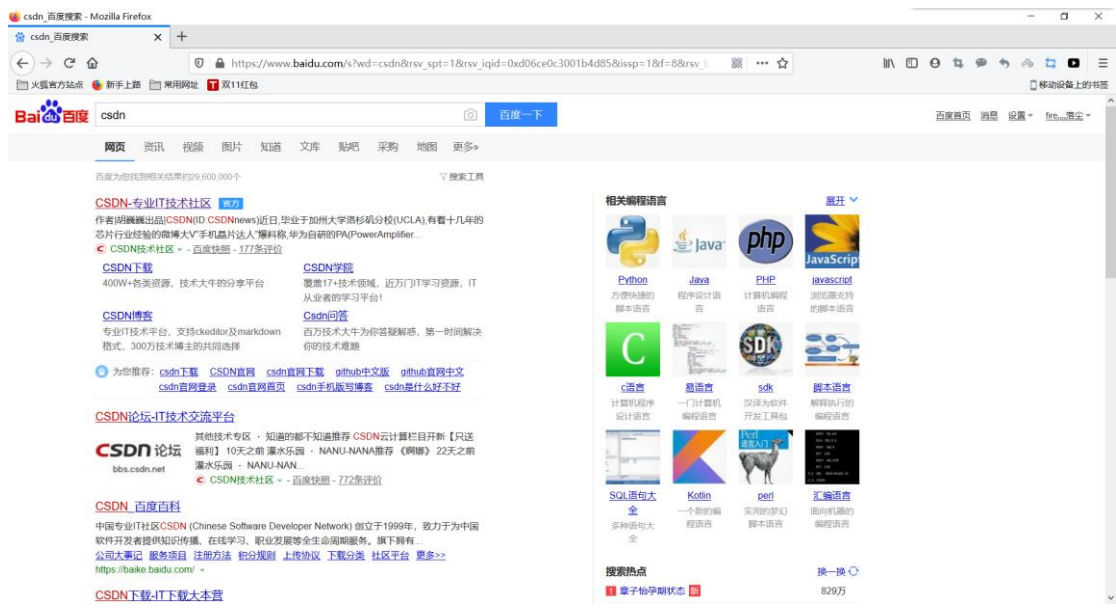


屏蔽某些网站的访问:  
在config.json中,我添加了

```

{
  "forbidden_hosts": [
    "www.csdn.net",
  ],
  "forbidden_clients": [
    "#127.0.0.1"
  ]
}
    
```

准备打开csdn



看看控制台信息

### 控制台截图

## 钓鱼网站截图

此处是钓鱼网站



对实验过程中的思考问题进行讨论或回答。

在读取socket的inputStream时,出现了会不断循环的情况。

上网查找相关问题后发现,Socket的inputStream的数据传输只有在连接结束时才会结束,要想结束读取,必须设置计时器。设置了Socket的 outTime,当超时时会抛出SocketTimeoutException异常,通过捕捉这个异常就可以退出inputStream的读取。



心得体会:

结合实验过程和结果给出实验的体会和收获。

实验设计非常好,就是感觉平时看mooc和实验有点脱轨,直接上来编程有点难受

