HTTP proxy server based on Real-time link

Guang-fei Du
School of Information and Engineering
Zheng-zhou University
Zheng-zhou, China
zzudgf@163.com

Zhi-hong Zhang
School of Information and Engineering
Zheng-zhou University
Zheng-zhou China
iezhzhang@zzu.edu.cn

Xiao-ying Wu
School of Information and Engineering
Zheng-zhou University
Zheng-zhou China
mysister517@163.com

***Abstract*—This paper introduces the design and implementation of HTTP proxy server based on real-time link. The real-time link packages UPLink and PSLink into a link. The UPLink is the link between user and proxy server, and PSLink is the link between proxy server, and target server. These two links exist and close together, and it can solve the problem of link's closed incompletion. This approach can reduce network congestion and improve the efficiency of proxy server.**

Keywords-Proxy server; Real-time link; C/S mode; Internet; HTTP

I. Introduction

Proxy server is an important safety feature. It works mainly in the dialogue layer of Open Systems Interconnection (OSI)[1], playing the role of firewall. Proxy servers are mainly used to connect INTERNET and INTRANET.

There is a problem of common proxy server, which is link's closed incompletion: Proxy server analyzes the user's request, and sends the request information to target server, then reads out returned data of target server. If user closes UPLink while data is incompletely read, and PSLink is not closed, the proxy server will read out the remaining returned data of target server, then return the data to user, however, UPLink has been closed. This phenomenon wastes network traffic and resources of proxy server.

This paper proposes Http proxy server based on real-time link. Proxy server uses real-time link control UPLink and PSLink, if one of the two links is closed, the other will be closed too. This approach solves the problem of link's closed incompletion.

II. Proxy Server Technology

Http proxy server receives user's request, and sends request information to target server, and target server responses to it, then proxy server sends the responsed information to user. Designing principles are as follows:

1. Using C/S Mode For Structural Design

Figure1 C/S Mode Proxy Server

Fig.1 is C/S Mode Schematic diagram. In this mode, user sends application request to proxy server, and proxy server responses it, at this time, proxy serves as a server. Proxy server receives requested information from user, then sends the information to target server, while proxy is a client of target server[2].

2. Caching Improve Efficiency Of Proxy Service

Proxy server uses caching deal with mutual information of client and server[3,4] so as to improve access efficiency of client. In this mode, if user sends request to proxy server, proxy will find cache information firstly. If there is a cache for this request, proxy returns the cache information to user; otherwise, proxy sends this

978-0-7695-4258-4/10 $26.00 © 2010 IEEE
DOI 10.1109/MINES.2010.44

request to target server, and target server responses to it, proxy receives the response and returns it to user. Studies have shown that cache improves capability of WWW extently[5].

3. User Request Management

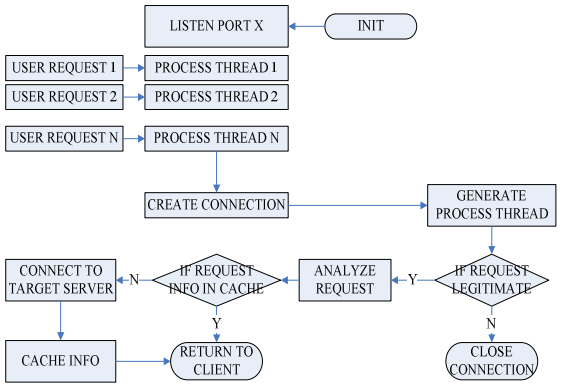

Figure 2 User Request Management of Common Proxy Server

Proxy server listens a port, and waits for user's connection request. When users send request to proxy through the port, proxy sends authentication request to users, if the returned information from users is correct, the connection between user and proxy is established. This is typical process to create a TCP connection. Proxy server controls user's request and server's response. It can carry out authentication, connection control and competence control on users[6]. User management of common proxy likes Fig.2. User management mainly considers three cases as follows: First of all, proxy server listens an open port continuously. If a user initiates a connection request, proxy creates a socket to receive the request and creates a thread to deal with it. Secondly, it creates TCP Connection dynamically, and proxy creates Two-way Connection with user through authenticating the user. Thirdly, Multi-thread mechanism deals with user requests, and proxy creates a thread to deal with user's connection request, all of the threads are independent except accessing to protect resources.

4. Connection Control Mechanism

Proxy controls user's access to target server, all requests from users to target server must be sent to proxy firstly, then proxy server returns the response information from target server to user. In this mechanism, user do not access target server directly, and the access connects to target server through proxy, therefore, proxy can be used as firewall by controlling user's connection request[7].

III. Real-time link

Real-time link is a connection among client, proxy server and target server, it is constituted by UPLink and PSLink, furthermore these two links exist and close together.

This paper proposes Request-Parsing algorithm, real-time link uses this algorithm to parse request information submitted by user, so we can get target host from it. This algorithm limits the length of buffer, which is used to store the data read from sockets, therefore, makes proxy server disregard of deformity of data packages received from client.

A. Real-time link Algorithm

Real-time link is expressed as follows:

```
class link {
    int UPLink;
    int PSLink;
}
```

UPLink is the link between user and proxy server, while PSLink is the link between proxy server and target server.

Real-time link algorithm is expressed as follows:

0: listening a port;
1: receive user request;
2: create UPLink;
3: add read() and close() events in UPLink;
4: read UPLink_socket;
5: if(!UPLink_socket.contains(host)), turn to step 6;
6: UPLink.close(), turn to step 0;
7: else if(UPLink_socket.contains(host)), turn to step 8;
8: create PSLink;
9: add read() and close() events in PSLink.
10: if UPLink can be read, read out the data and write into PSLink.
11: else PSLink.close(),turn to step 0;
12: if PSLink can be read, read out the data and write into UPLink.
13: else UPLink.close(), turn to step 0;

In this algorithm, UPLink_socket is the socket of UPLink. It creates UPLink after receiving user's request, reads and parses UPLink_socket to get target host, then

creates PSLink, then the whole link is created. In real-time link, if we close one of the two links, two sockets will be closed and the whole link is closed.

B. Request-Parsing Algorithm

Request-Parsing algorithm is used to parse user's request, so as to get target host which the host user wants to visit, the algorithm is expressed as follows:

0: nsize=0;
1: while nsize>=0;
2: if(!proxyBuffer.contains(host)), turn to step 3;
3: nlength = read(UPLink_socket, proxyBuffer += nlength, nsize-=nlength);
4: turn to step 1;
5: else if(proxyBuffer.contains(host)), turn to step 6;
6: return host; exit loop;
7: if(nsize<0), turn to step 8;
8: return -1;

In this algorithm, read() can read the information of UPLink_socket into proxyBuffer, UPLink_socket is the socket of UPLink, proxyBuffer is the buffer for proxy to read socket, nsize is the size of proxyBuffer, nlength is the length of the data read from socket.

C. Application of Real-time link on Proxy server

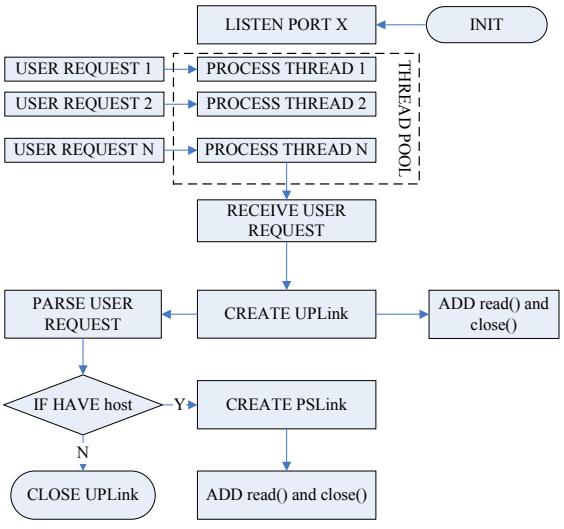

Figure 3 Real-time link of Proxy server

Multithread is used to applying real-time link in proxy server. For the first time, we create an IO server, and there is a thread pool, the thread pool is used to deal with user's request. When proxy receives a user's request, it will call function IOServer.push(Socket, Event_Connect, do_connect), this method is used to push socket, event and function into IO server, IO server will call function do_connect asynchronously to create connection when Event_Connect occurs.

The process of creating real-time link likes Fig.3. Proxy server listens port X, if there is a request form user, proxy will create a thread to deal with it. The thread creates UPLink firstly, then uses Request-Parsing algorithm to parse user's request information. If there is a host in the request information, it will create PSLink, so real-time link is created; otherwise there is no host in the requested information, it will close UPLink. The workflow of real-time link is shown in Fig.4.When real-time link is created, if UPLink can be read, it will read out the data and write into PSLink; and if PSLink can be read, it will read out the data and write into UPLink. As these two links interrelated with each other, if one of the two links is closed, the whole link will be closed.

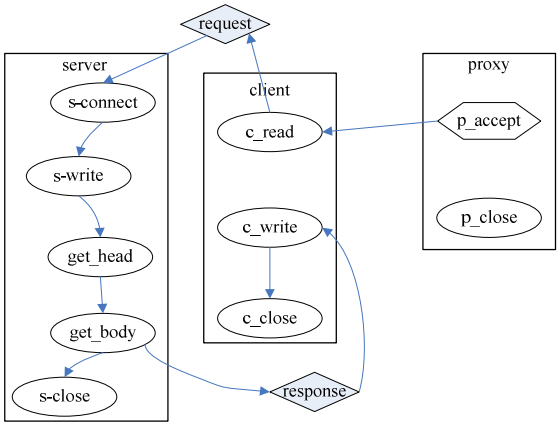

Figure 4 Workflow of Real-time link

IV. Experiment and Analysis

A. Experiment Environment Setup

In this paper, we design and implement prototype system of HTTP proxy server based on real-time link with Java in MyEclipse. In this prototype system, we don't care of cache and legitimacy of user connection request, just implement the basic functions of real-time link with Request-Parsing algorithm. When real-time link is created in this system, UPLink and PSLink are packaged into a link, as if one of the two links is closed, the whole real-time link is closed.

B. Experimental Results Analysis

We deploy prototype system of HTTP proxy server based on real-time link and Broadband HTTP proxy server AdslHttpProxy V1.2[8] in local server, and visit a website www.people.com through the two proxy server from local client, then capture the response data from target server by Ethereal on proxy server[9].

We collect experimental data by following method. Firstly, user initiates a request and sends it to proxy server, then closes the link between user and proxy instantly before the response data is received completely. In this process, we capture the responsed data from target server by Ethereal, and calculate the flow of it. Ethereal captures the data twice. Firstly, the local client visits the website through prototype system of HTTP proxy server based on real-time link, part of the capture data package is shown as follows:

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
The document has moved here
<hr>
<address>Apache Server at www.people.com Port 80</address>
</body></html>

Secondly, the local client visits the website through AdslHttpProxy V1.2, part of the capture data package is shown as follows:

HTTP/1.1 200 OK
Date: Fri, 30 Apr 2010 01:37:28 GMT
Server: Apache
ETag: "5f7c-2c2b3f00"
Accept-Ranges: bytes
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding,X-Catmap-Header,User-Agent,*
Content-Encoding: gzip
P3P: CP='PHY ONL UNI PUR FIN COM NAV INT DEM CNT STA PRE CUR ADMa DEVa TAIo PSAo PSDo IVAo IVDo CONo TELo OTPi OUR UNRo PUBi OTRo IND DSP CAO COR', CP='PHY ONL UNI PUR FIN COM NAV INT DEM CNT STA PRE CUR ADMa DEVa TAIo PSAo PSDo IVAo IVDo CONo TELo OTPi OUR UNRo PUBi OTRo IND DSP CAO COR'
Content-Length: 19058
Keep-Alive: timeout=5
Connection: Keep-Alive

The two packages shows that, if we visit a web page through HTTP proxy server based on real-time link, then close the web page quickly before read out all of the data, as real-time link is closed because the link between user and proxy is closed, there will be 301 error, and proxy won't receive the whole data from target server. If we visit a web page through AdslHttpProxy V1.2, and user closes the connection with proxy server, as the link between proxy and target server is still alive, the proxy will receive all of the data returned from target server, so that the amount of data transmitted is 19058bytes. Therefore we can conclude that the application of real-time link can reduce network traffic.

V. Conclusion

This paper proposes Http proxy server based on real-time link. Real-time link can control the connection among user, proxy server and target server, because these two links UPLink and PSLink exist and close together, it solves the problem of link's closed incompletely, and can reduce network traffic so as to improve the efficiency of proxy server. This paper proposes Request-Parsing algorithm, and real-time link uses this algorithm to analyze requested information which is submitted by user, so we can get target host from it. This algorithm limits the length of buffer, makes proxy not care for format and length of packet in order to improve the efficiency of proxy server.

REFERENCES

[1] OSI model: From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/OSI_model, 2010

[2] IlJin Lee, Jae Cheon Han, Wook Hyun and ShinGak Kang, "A Study on Keepalive Mechanism between SIP UA and Proxy server for Internet Telephony Service," The 9th International Conference on Advanced Communication Technology, Proceedings Volume III, Toward Network Innovation beyond Evolution, 2007, pp. 1117～1120

[3] Michael Rabinovich and Oliver Spatscheck, "Web Caching and Replication," ACM SIGMOD Record Volume 32, Issue 4, 2003, pp. 107～108

[4] Lei Shi, Lin Wei and Zhiming Gu, “Study on Cachability of Web Objects and Acceleration Method,” Computer Engineering, 2005, 31(18) , pp. 74～75， 89

[5] B.M.Duska, D.Marwood and and M.J.Feelay, “The measured access characteristics of World Wide Web client proxy caches, Proceedings of USENIX Symposium on Internet Technologies and Systems,”(http://cs.ubc.ca/spider/feeley/wwwap/wwwap.html)

[6] Lima Livio, Taubman David and Leonardi Riccardo, “JPIP proxy server for remote browsing of JPEG2000 images,” Multimedia Signal Processing, 2008 IEEE 10th Workshop on, Cairns, Queensland, Australia, 2008, pp 844～849

[7] Hong Liu, “Design and Implementation of Web-Based WAP Page Transformation Proxy,” Graduate University of Chinese Academy of Sciences(Institute Of Computing Technology) , 2001

[8] AdslHttpProxy V1.2, http://www.skycn.com/soft/11288.html, 2010

[9] Ethereal: The world's most popular network protocol analyzer, http://www.ethereal.com/, 2006

This paper is based on Important National Science & Technology Specific
Serial number:2009ZX03001-018