*Proceedings of the Asia Pacific Advanced Network*

# Intelligent Proxy Server in TANet/TWAREN

**Ya-Ling Wang [1], Ying-Chieh Chang [2] and Quincy Wu [1,*]**

1 National Chi Nan University / Department of Computer Science and Information Engineering
2 National Chi Nan University / Department of International Business

E-Mails: {s96321013,s97212905,solomon}@ncnu.edu.tw

* Corresponding author; Tel.: +886-49-2910960 # 4736

---

**Abstract:** This paper studies the choice of web proxy servers. Usually there are plenty of proxy servers that are available to users, so it is important for users to choose the best proxy server. When visiting web servers in different countries, it is a common practice to divert the requests to different proxy servers in order to get better performance. However, generally the users do not exactly know which proxy server provides the fastest service for a specific website. Therefore, this paper proposes to run a priori speed tests in webpage retrieval, and choose the fastest proxy server according to the measurement. The experimental result shows that the performance of this strategy is better than the naïve approach which simply makes up the decision according to the nationality of websites.

**Keywords:** Intelligent; Learn; Proxy server; TANet; TWAREN.

---

## 1. Introduction

For server administrators, it has always been a tough technical decision to adjust the strategy of web proxy servers in his/her site to obtain the best performance [1]. For a multi-homed site like National Chi Nan University (hereinafter referred to as NCNU), this is even more difficult. Just as many universities in Taiwan, NCNU is connected to two national research/education networks (NRENs) – Taiwan Academic Network (TANet) and Taiwan Advance Research & Education Network (TWAREN), each of which provides a proxy server to serve connecting sites [2]. Because TWAREN provides high speed connection through APAN to many countries, the original configuration of NCNU is to access all webpages through the TWAREN proxy server. However, some tests demonstrate that this strategy is not always the optimal, especially for

commercial websites or domestic websites. This violates the goal to reduce download time through a proxy server [3]. Therefore, depending on the destination websites, we may want to use different upstream proxy servers. For web browsers like Microsoft Internet Explorer, users need to manually modify the proxy server name in Internet Options to choose a different proxy server. For other browsers like Mozilla Firefox and Seamonkey, add-ons like *foxyproxy* were developed to allow users selecting multiple proxy servers and route requests to one of them according to regular expression matching of URLs [4].

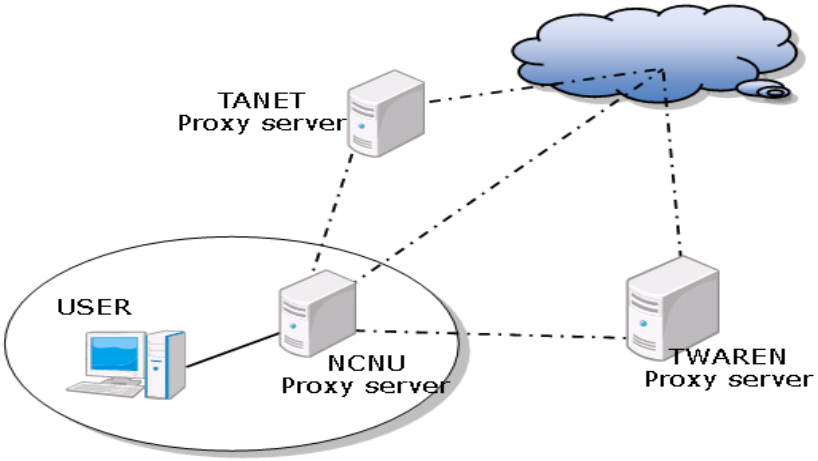


**Figure** 1. Multi-proxy servers architecture

According to users' feedback, configuring multiple proxy servers requires the knowledge of network topology, and constant measurement of download time. Therefore, although some network experts may enjoy the flexibility of foxyproxy, many users still prefer to choose a single web proxy server, which can *usually* provides them the fastest connection. Consequently, NCNU decided to provide users a single web proxy server, which will intelligently determine to access a webpage via the TWAREN proxy server or the TANET proxy server, depending on which one provides faster service for the target site (see Figure 1). In some cases, the NCNU proxy server will decide to bypass both the TANet proxy server and the TWAREN proxy server, and access the website directly, because the overhead for retrieving the website indirectly through another proxy server is larger than the benefit it can obtain. The goal is to promote all NCNU users to surf the web through the proxy server, so that the average download time can be minimized [5].

## 2. Architecture and Components

NCNU proxy server is running on the CentOS 5.4 operating system. Its web proxy server was upgraded to Squid 3.1 in February of 2010 to support IPv6 websites. The access log is analyzed by Webalizer and the report is generated by SARG (Squid Analysis Report Generator)

to provide statistical charts. At the beginning, the policy is that all non-local webpages were retrieved via the TWAREN proxy server.

As shown in Figure 1, the current test environment consists of three proxy servers:

1. (local) NCNU proxy server
   `http://proxy.ncnu.edu.tw:3128/`
2. TANet proxy server
   `http:// proxy.moe.edu.tw:3128/`
3. TWAREN proxy server
   `http:// proxy1.twaren.net:3128/`

*2.1. Measurement Tools*

To measure the download speed of each webpage, a tool `http_ping` is utilized. The behavior of `http_ping` is similar to the command `ping`. The difference is that `ping` sends ICMP echo requests while `http_ping` sends HTTP requests. The command `ping` used to be a popular utility for network engineers to test the round-trip time between two hosts. However, recently many firewall policies simply discard ICMP requests to prevent hackers from scanning on-line hosts. On the contrary, HTTP requests can always successfully pass through the firewall, so it is suitable to support our measurement. Moreover, `http_ping` supports IPv6. Through a command-line option `-proxy` it can specify which proxy server to use. The -count option specifies the number of iterations of HTTP request, and the data transmission time will be calculated, as shown in Figure 2.

```
[linlin@proxy2 ~]$ http_ping -count 3 -proxy proxy.ncnu.edu.tw:3128 http://www.ncnu.edu.tw
135 bytes from http://www.ncnu.edu.tw: 0.402 ms (0.072c/0.309r/0.021d)
135 bytes from http://www.ncnu.edu.tw: 0.4 ms (0.041c/0.341r/0.018d)
135 bytes from http://www.ncnu.edu.tw: 0.475 ms (0.046c/0.409r/0.02d)

--- http://www.ncnu.edu.tw http_ping statistics ---
3 fetches started, 3 completed (100%), 0 failures (0%), 0 timeouts (0%)
total    min/avg/max = 0.4/0.425667/0.475 ms
connect  min/avg/max = 0.041/0.053/0.072 ms
response min/avg/max = 0.309/0.353/0.409 ms
data     min/avg/max = 0.018/0.0196667/0.021 ms
```

**Figure** 2. `http_ping` screenshots

In our experiment, we shall repeat the `http_ping` test 3 times to retrieve each webpage from a designated proxy server, so the command will look like:

`http_ping -count 3 -proxy` *proxy server*`:`*port* `http://`*URL*

*2.2. Data Collection Process*

The NCNU proxy server performs a download test regularly every day. From the access.log file of the proxy server, URLs which were retrieved by users are extracted, and `http_ping` tries to retrieve these webpages via different proxy servers. According to the download speed of each site, new rules will be added into the configuration file of the NCNU proxy server. It then reloads the profile so that the proxy server will choose the fastest proxy server between TWAREN proxy server and TANet proxy server as the upstream server. To reduce the total number of download pages, for popular websites like Yahoo! which receives thousands of visits in each day, our experiment will only take http://news.yahoo.com/ as a representative and download the webpage 3 times. Figure 3 show the flow chart of the data collection process.

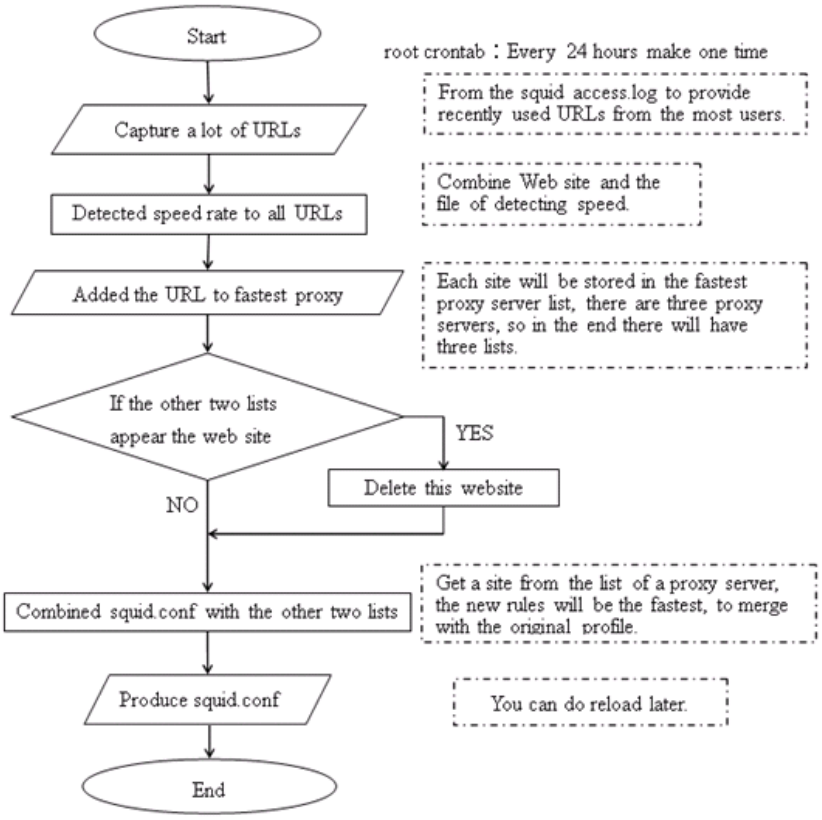


**Figure** 3. Flow chart of the data collection process

Each test will generate two lists. One list contains all websites whose download speed will be maximal via the TWAREN proxy server. The other list contains websites whose download

speed will be maximal via the TANet proxy server. These two lists will be included into `squid.conf` to produce a new configuration file (see Figure 4 and Figure 5). For other sites not in these two lists, retrieving them directly from the NCNU proxy server provides the fastest download speed, without going through any further proxy servers.

A

```
cache_mgr ycc@ncnu.edu.tw

acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl localhost src ::1/128
acl localhost src 163.22.0.0/19
acl localhost src 163.22.32.0/23
acl localhost src 163.22.34.0/24
acl localhost src 2001:e10:6840::/48
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl to_localhost dst ::1/128
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7   # RFC 4193 local private network range
acl localnet src fe80::/10  # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnet
http_access allow localhost
http_access deny all
http_port 3128
```

B

```
cache_peer proxy1.twaren.net parent 3128 0 no-query no-netdb-exchange no-digest
acl twaren dstdomain .edu .org .gov .jp .cn .hk .vn .th .kr .in .my .ph .sg .ca
cache_peer_access proxy1.twaren.net allow twaren
```

C

```
hierarchy_stoplist cgi-bin ?
cache_dir aufs /var/cache/squid 20480 16 256
maximum_object_size 32768 KB
coredump_dir /var/cache/squid
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
```

**Figure** 4. Squid original configuration

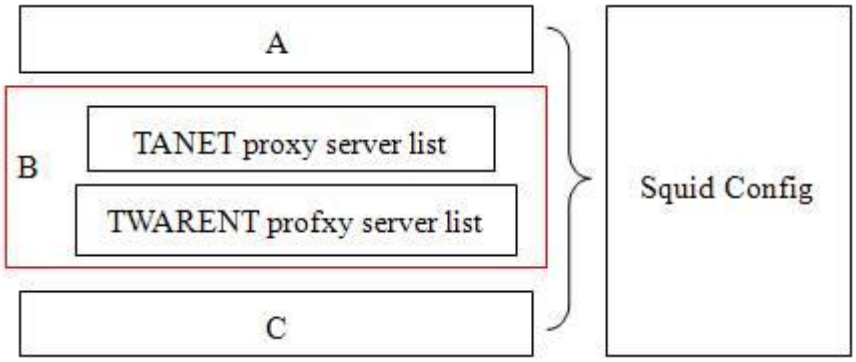


**Figure** 5. Squid new profile

## 3. Experimental Results

Although `http_ping` does not consume much bandwidth [6], the tests were required to be scheduled at dawn (4AM in this experiment) to prevent it from causing network congestion or bringing any inconvenience to other users. However, NCNU has enough uplink bandwidth. According to Table 1 which shows the statistics of NCNU inbound traffic flows during 2010/04/29 to 2010/05/10, these `http_ping` tests did not significantly increase the network traffic. Take April 29th for example. It only consumed 7.9Gbyte of bandwidth, which was ranked as top 15 among all hosts in NCNU.

Table 1. Network traffic of the NCNU proxy server

| IP address : 163.22.12.5 | | | |
|---|---|---|---|
| date | TOP | Inflow (bytes) | FQDN |
| 2010 / 04 / 29 | 15 | 7,946,919,322 | proxy.ncnu.edu.tw |
| 2010 / 04 / 30 | 15 | 6,082,848,187 | proxy.ncnu.edu.tw |
| 2010 / 05 / 01 | 36 | 1,633,764,588 | proxy.ncnu.edu.tw |
| 2010 / 05 / 02 | 17 | 4,105,709,382 | proxy.ncnu.edu.tw |
| 2010 / 05 / 03 | 26 | 2,947,806,027 | proxy.ncnu.edu.tw |
| 2010 / 05 / 04 | 15 | 9,523,143,539 | proxy.ncnu.edu.tw |
| 2010 / 05 / 05 | 18 | 6,607,241,838 | proxy.ncnu.edu.tw |
| 2010 / 05 / 06 | 24 | 3,740,517,234 | proxy.ncnu.edu.tw |
| 2010 / 05 / 07 | 17 | 7,855,203,024 | proxy.ncnu.edu.tw |
| 2010 / 05 / 08 | 17 | 6,211,977,904 | proxy.ncnu.edu.tw |
| 2010 / 05 / 09 | 196 | 349,948,151 | proxy.ncnu.edu.tw |
| 2010 / 05 / 10 | 20 | 5,091,735,935 | proxy.ncnu.edu.tw |

Figure 6 shows the test for 3191 randomly chosen URLs from the access.log data on May 7th, 2010. The horizontal axis is the data amount while the vertical axis is the cumulative access time in millisecond. The blue line indicates the access time required to download those webpages through the NCNU proxy server; the average time for a site is 455.54ms. The red line indicates the access time required to download those webpages without the NCNU proxy server; the average time for each site is 1014.49ms, which is twice longer than the scenario with the NCNU proxy server. This figure clearly shows that utilizing the NCNU proxy server can reduce total access time.

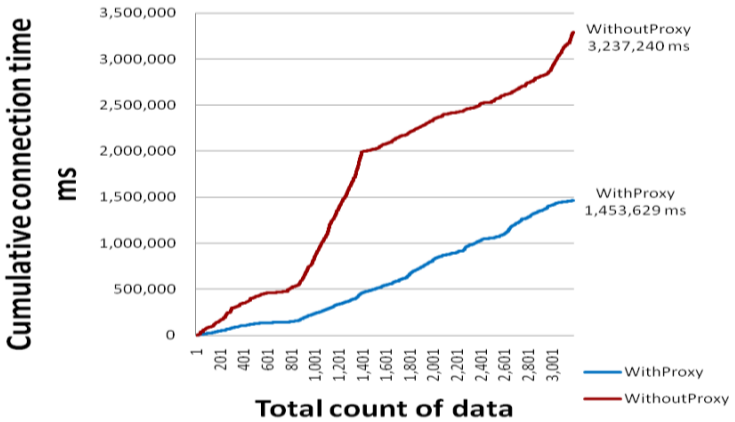


**Figure** 6. 2010/05/07 Speed test

Figure 7 shows another experiment on May 11th with 157 randomly selected documents. The horizontal axis is the total amount of data. The vertical axis is the cumulative access time in millisecond. The blue line indicates that, for intelligent proxy server with adaptive rules, averagely each webpage only takes 481.78 ms to download. The green line indicates that, for traditional proxy server with fixed rules, the average time to download a webpage is 1069.82 ms. The red line indicates the time to retrieve webpages without proxy servers, where the average time to download a page is 1837.06 ms.

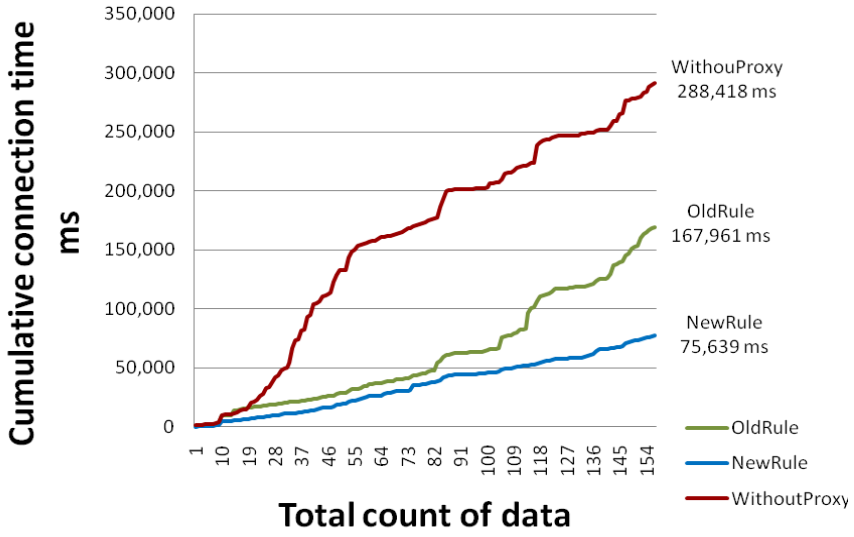


**Figure** 7. 2010/05/11 Comparison Test

## 4. Conclusions and Future Work

Through experiments in this paper, it was shown that with proxy servers, the access time to retrieve webpages will be significantly decreased. If the proxy server can intelligently adjust its rule to utilize upstream proxy servers to retrieve data, it can further reduce the access time profoundly.

As more users utilize this proxy service, there will be more websites included in the testing list, and further retrieval to webpages on those sites can automatically choose the fastest upstream proxy server to retrieve data.

However, the testing process downloads webpages which consume network bandwidth, so it is suggested that in the future, it can re-use the statistical information from some existing measurement system like perfSONAR, so that it need not download webpages frequently..

**Acknowledgements**

The Computer and Network Center in National Chi Nan University provides perfect equipment and support to this experiment, which is highly appreciated.

**References**

1. R. Howard, B.J. Jansen, "A proxy server experiment: an indication of the changing nature of the Web", Proceedings of the 7th International Conference on Computer Communications and Networks, pp. 646 – 649, 1998.
2. C.L. Jeffery, S.R. Das, G.S. Bernal, "Proxy-sharing proxy servers", Proceedings of First Annual Conference on Emerging Technologies and Applications in Communications, pp. 116-119, 1996.
3. S. Hussain, R.D. McLeod, "Intelligent prefetching at a proxy server", Proceedings of 2000 Canadian Conference on Electrical and Computer Engineering, pp. 209-211 vol.1, 2000.
4. Jeremy Clark, P. C. van Oorschot, Carlisle Adams, "Usability of anonymous web browsing: an examination of Tor interfaces and deploy ability", ACM Proceedings of the 3rd Symposium on Usable Privacy and Security, pp. 41-51, 2007.
5. M. Wijnants, P. Monsieurs, P. Quax, W. Lamotte, "Exploiting Proxy-Based Transcoding to Increase the User Quality of Experience in Networked Application", Proceedings of First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA'05), pp. 73-80, 2005.
6. Wei-hsiu Ma and David H.C. Du, "Reducing bandwidth requirement for delivering video over wide area networks with proxy server", Multimedia, IEEE Transactions, pp. 539-550, 2002.