

Web SHAKE: A fast WWW access method for mobile terminals on temporary cluster networks

Yousuke Konishi, Susumu Ishihara and Tadanori Mizuno

Abstract—Recently mobile computing has become very popular. Many people can access the internet through mobile hosts. However, wireless links used by mobile hosts have problems such as narrow bandwidth and low reliability. To offer high-speed communication on wireless links, we have proposed *SHAKE* (SHARING multiple paths procedure for cluster network Environment). In *SHAKE*, mobile hosts that are connected by a fast local link use multiple wireless links owned by each host simultaneously to communicate with hosts on the internet. An experimental system was implemented and evaluated. In this system, however, not only mobile hosts but also correspondent hosts on the internet require special software for *SHAKE*. In this paper, we propose a fast WWW access method with *SHAKE* (*Web SHAKE*). The feature of *Web SHAKE* is that it does not require any special software on web servers on the internet to use *SHAKE*. The feature is realized by an use of HTTP Proxy Server that works on each mobile host. We present the result of a performance evaluation of the *Web SHAKE*.

Index Terms—HTTP, WWW, Mobile computing, distributed system, wireless communication, Ad-Hoc network, multiple access methods.

I. INTRODUCTION

Recently, the number of internet users has increased rapidly because of the remarkable spread of personal computers and the progress of computer network technologies in the last decade. In addition, users who access the internet from mobile (wireless) computers are common because mobile hosts (e.g., portable computers, PDAs, etc.) and cellular phones have become popular.

Because of the increase in internet users, internet content and services demand higher quality. For example, on the WWW (World Wide Web), web pages including multimedia data (pictures, sounds, movies, etc.) are common, and the data size of web content increases daily. However, current personal wireless communication technologies are not sufficient to transfer multimedia data in comparison with wired communication because of the narrow bandwidth and low reliability of wireless links.

To offer fast and high speed communication on wireless links, we have already proposed *SHAKE* (SHARING multiple paths procedure for a cluster network Environment). In *SHAKE*, mobile hosts that are connected by a fast local link (e.g., wireless LAN [IEEE802.11b], Bluetooth, etc.) use multiple wireless links owned by each host simultaneously to communicate with hosts on the internet. An experimental system

has been implemented and evaluated [1]. In this system, however, not only mobile hosts but also correspondent hosts on the internet require special software for *SHAKE*.

In this paper, we propose a fast WWW access method with *SHAKE* (we call this method the *Web SHAKE*). The feature of *Web SHAKE* is that it does not require any special software on web servers on the internet. The feature is achieved by the use of HTTP proxy server that works on each mobile host. The results of a performance evaluation of the *Web SHAKE* system are described in this paper.

II. SHAKE (SHARING MULTIPLE PATHS PROCEDURE FOR A CLUSTER NETWORK ENVIRONMENT)

When we access the internet to browse web pages while outdoors or on the move, we utilize mobile communication services (e.g., the cellular phone or personal handy phone system (PHS)). However, wireless links used by mobile hosts have problems such as narrow bandwidth and low reliability in comparison with a wired network.

To solve these problems, we proposed a multiple paths protocol procedure. In this procedure, mobile hosts with a fast wireless link (e.g., wireless LAN, Bluetooth, etc.) temporarily form a network (we call this network the Cluster Network), and when a mobile host accesses the internet, the mobile host uses not only wireless link to connect with the internet, but also the wireless links of the other mobile hosts. This system can offer high quality and high speed communication on wireless links.

We call this procedure *SHAKE* (Sharing multiple paths procedure for a cluster network environment). An example of communication with *SHAKE* is shown in **Fig. 1**. The advantages of *SHAKE* are as follows:

- Improvement of data transfer speed on wireless links.
- Even when the mobile host does not have an external link or cannot utilize external wireless links, if it uses *SHAKE*, it can utilize the wireless links of other mobile hosts.
- Cluster networks can be formed dynamically. They do not depend on the physical location of mobile hosts.

Similar ideas are found in the PPP multilink protocol [2] [3] and 384-Kbps PHS experimental system [4], which use special devices. Parallel data transmission at the datalink layer is achieved by use of network interfaces owned by a mobile host. However, *SHAKE* is different from these systems because *SHAKE* offers communication in parallel at the network layer or higher.

Another technique for fast WWW access method is that using distributed Cache. For example, There are “Sharp Su-

Y. Konishi belongs to the graduate school of information, S. Ishihara is with the faculty of Engineering, and T. Mizuno are with the faculty of information in Shizuoka University, Shizuoka, Japan. E-mail: y-konisi@mizulab.net.

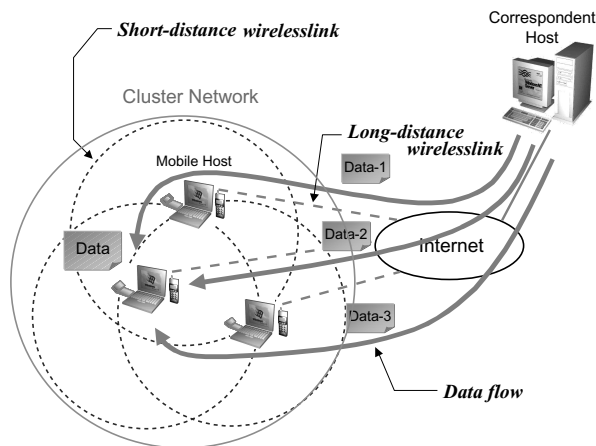


Fig. 1. Example of communication with SHAKE

per Proxy Script [5]” and “Squid WWW Cache system [6]”. These realize improvement in the speed by using two or more distributed cache servers simultaneously.

An experimental system of SHAKE has already been implemented and evaluated so far [1]. In this experimental system, the function of SHAKE has been implemented as a library that utilizes TCP for reliable socket-level communication. This library disperses the traffic on multiple paths, and sorts and combines data blocks transmitted via multiple paths.

III. WEB SHAKE

A. Overview of Web SHAKE

In the experimental system of SHAKE, both mobile hosts and correspondent hosts need special software for SHAKE to communicate with hosts on the internet. However, it is desirable that only mobile hosts need a SHAKE module.

We propose a fast WWW access method with SHAKE (we call this method the WebSHAKE). The Web SHAKE solves the problems encountered by the experimental SHAKE implementation, although it limits the available protocol to HTTP (Hyper Text Transfer Protocol). With Web SHAKE, mobile hosts in a cluster can communicate with any hosts on the internet through a HTTP proxy server that works on each mobile host. Special software for SHAKE is not required for the correspondent hosts (web server).

B. Web SHAKE architecture

HTTP (Hyper Text Transfer Protocol) [7] [8] consists of some request methods (e.g., GET method, POST method, HEAD method, etc.). In Web SHAKE, we use the HEAD method and partial GET method (GET request including Range header field). We can transmit only part of the data entity using the partial GET method. The HEAD method is used to obtain the size information of the target file, and the partial GET method is used to transfer data via multiple links.

The Web SHAKE architecture is shown in Fig. 2. SHAKE HTTP Proxies (HTTP Proxies that include SHAKE modules)

run on mobile hosts in a cluster. If a SHAKE HTTP Proxy (SHP) receives the HTTP request from a HTTP client (e.g., Internet Explorer, Netscape Navigator, etc.) running on the same mobile host, it sends partial GET requests to SHPs working on other mobile hosts in the same cluster network. When a SHP receives a partial GET request from a SHP running on the other mobile host in the same cluster network, the SHP behaves as a normal HTTP proxy server. That is, the SHP that receives a partial GET request from another mobile host sends a HTTP request instead of the mobile host that receives a GET request directly from a HTTP client.

Details of the data transmission on the Web SHAKE are as follows.

- 1) A HTTP client sends a GET request to a SHP running on the same host. (The HTTP client is configured to use the SHP as the HTTP proxy).
- 2) The SHP that receives a GET request from a HTTP client sends a HEAD request to a web server and receives a HTTP header including the file size information of the target file.
- 3) The SHP determines the data size for each wireless link according to the whole data size and the bandwidth of each link. It then sends n partial GET requests to the other SHPs on the same cluster network. The SHPs working on each mobile host run in parallel, forward the request and receive a part of the target data from the web server by use of the partial GET method.
- 4) The partial data received by each SHP is sent to the SHP that receives the GET request directly from the HTTP client.
- 5) The SHP that receives a GET request directly from the HTTP client sorts and combines the partial data and sends the combined data to the HTTP client. At that time, the SHP removes the header of the response of the partial GET requests because each replying message includes a useless header, and combines the extracted data

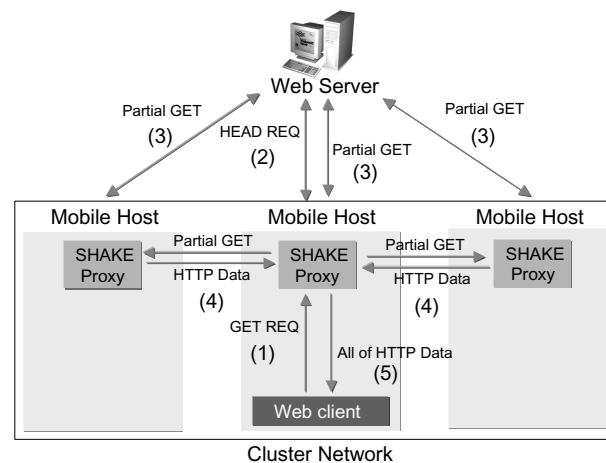


Fig. 2. Web SHAKE Architecture

in the replying message and the original header received in step (2).

We call this data transfer process the F-H REQ (First-Head REQest) because the SHP receiving a GET request directly from a HTTP client sends a HEAD request first. Sending the HEAD method and partial GET method, controlling the cluster network, and sorting and combining parts of the data are performed by SHPs.

C. The optimization of the data transfer process

If we use the F-H REQ to transfer small data, we cannot obtain a larger throughput than the normal WWW access method. This is because the overhead of the first HEAD method is not negligible when the size of the data entity is small. To prevent this degradation, we propose a new architecture that replaces the HEAD method ((2) in Fig. 2) with partial GET method. We can receive not only partial data, but also the size of the whole data from the partial content header by using the partial GET method. On account of this, if we use the partial GET method instead of the HEAD method, the SHP can receive the head of the data including the size information. Therefore, when the data size is small, the SHP can receive all the data by a message exchange between the SHP and web server at once. We call this data size that the mobile host receives by using the first partial GET method the **Maximum Data Size at First Request (MDFR)**. When the size of the data is large, the SHP may receive the rest of the data that cannot be received with the first partial GET method from the web server by multiple wireless links in parallel. We call this data transfer process the F-P REQ (First-Partial GET REQest), and discriminate between the F-H REQ and the F-P REQ.

IV. PERFORMANCE EVALUATION

A. Experimental environment

1) *Overview:* We used three note PCs (Personal Computers) with PCMCIA card for PIAFS (PHS Internet Access Forum Standard) [9] communication and a wireless LAN (IEEE802.11b). The note PCs were Panasonic SS3010 (CPU: PentiumII 300Mhz, Memory: 64Mbytes, OS: Linux kernel ver. 2.2) and two Panasonic CF-M2 (CPU: Celeron 500Mhz, Memory: 128 Mbytes, OS: Linux Kernel ver. 2.2). The PCMCIA cards for the PIAFS communication were two MC-P210 (Seiko Instruments Inc.) and one P-in comp@ct (NTT DoCoMo). The cluster network consisted of Ethernet (100BASE-T) or wireless LAN (IEEE802.11b).

2) The experiment:

Experiment-1: Ideal environment (2 same wireless links):

We measured the throughput the case where the throughputs of all wireless links are almost equal in the Web SHAKE system. **Fig.3** shows the network environment for experiment-1. We measured the data transmission time and throughput when the wireless links used one and shared two. We define the data transmission time as the time required to receive a whole requested file after starting to send a GET request from a web

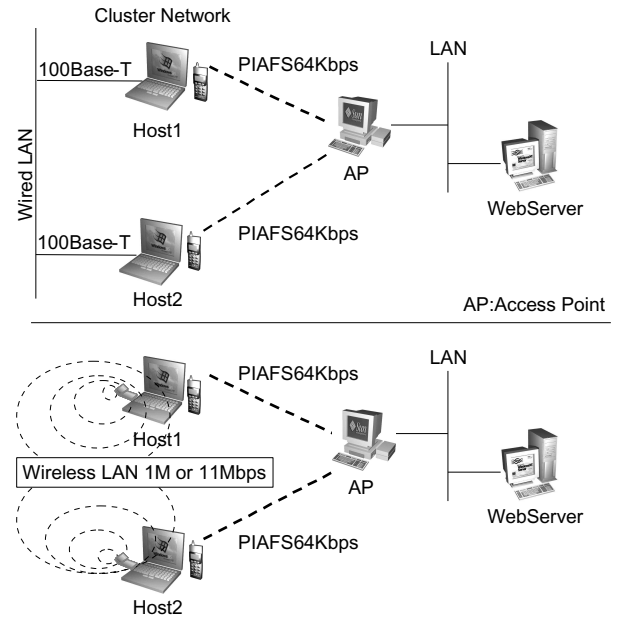


Fig. 3. Experimental environment-1

client. Host1 in Fig. 3 requests 1K – 1M[bytes] files from the web server.

The Access point (AP) and the web server were connected to Shizuoka University's local area network not via the internet. The maximum bandwidth of each wireless link was 64Kbps. The cluster network consisted of a wired LAN (100BASE-T) and wireless LAN (IEEE802.11b). We evaluated the performance of the Web SHAKE system in three cases in this environment. The first case was the cluster network that consisted of the wired LAN (100BASE-T). The second case was the cluster network that consisted of the wireless LAN (transfer rate: 11Mbps). The third case was the cluster network that consisted of the wireless LAN (transfer rate: 1Mbps). The SHP on the same host running web client dispersed the same amount of data traffic among each path. The data transfer process was F-H REQ.

Experiment-2: A comparison of F-H REQ with F-P REQ:

Fig. 4 shows the network environment for experiment-2. We compared the performance of the F-H REQ with the F-P REQ in the case where the HTTP client requested small data. In the environment of experiment-1, both Host1 and Host2 were connected to the same AP, and the AP and web server were not connected to the internet. However, in the environment of experiment-2, Host1 and Host2 were connected to different APs, and AP1 and AP2 were connected to the internet so that we gave each wireless link a different throughput and transmission delay. The method of measurement was the same as in experiment-1. However, the data size that the mobile host received was not 1[KB] – 1[MB], but 1[KB] – 128[KB] because we focused on evaluating the performance of the Web SHAKE when the data size that Host1 requested to the web

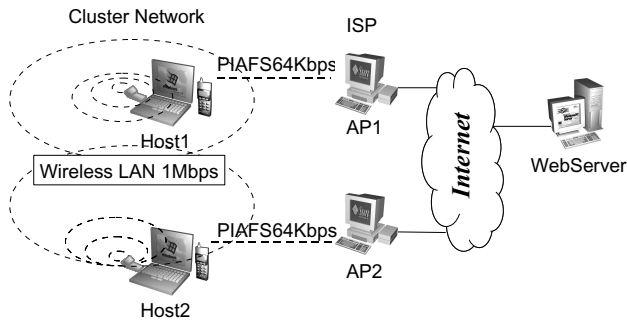


Fig. 4. Experimental environment-2

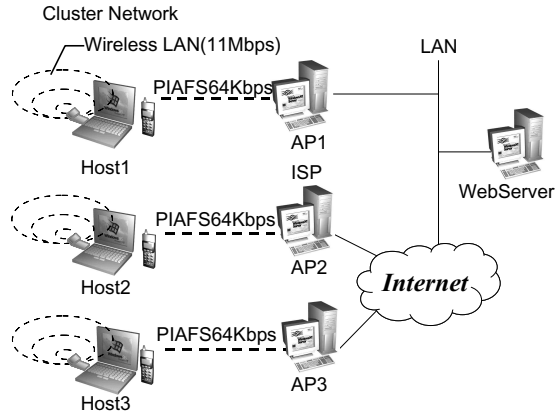


Fig. 5. Experimental environment-3

server was small. We evaluated the performance of two different implementations of the Web SHAKE system using of this environment. In the first implementation, the data transfer process was F-H REQ. In the second implementation, the data transfer process was F-P REQ (MDFR = 10KB, 20KB, 30KB).

Experiment-3: Multiple links with different characteristics: We measured the throughput on the Web SHAKE in the case where the throughput of each wireless link was different with experiment-2. In the environment of experiment-3, Host1, Host2, and Host3 were connected to different APs, and AP2 and AP3 were connected to the internet so that we gave each wireless link a different throughput and delay. **Fig. 5** shows the network environment of experiment-3. In this network, 1 – 3 wireless links (64Kbps) were used simultaneously. The data transfer process was F-H REQ and F-P REQ (MDFR = 20KB). The method of measurement was the same as in experiment-1 and experiment-2. We dispersed the same amount of data traffic among each path. When the number of shared wireless links was two, the used links were Host(1)+AP1 and Host(2)+AP2. When only one wireless link was shared, Host(1)+AP1 was used. The receiver was Host2 in Fig. 5 (i.e., Host2 sends the HTTP request and receives the HTTP data).

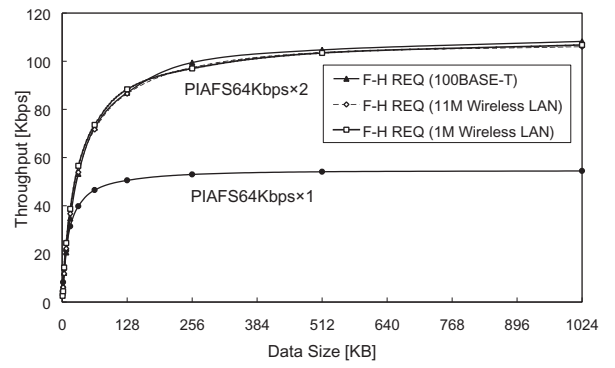


Fig. 6. Result of experiment-1

B. Results and evaluation

1) **Experiment-1:** The results of experiment-1 are shown in **Fig. 6**. This graph shows the throughput of the data transmission on the Web SHAKE and normal WWW access. We can see that as the data size increases, the throughput becomes larger. In particular, when the data size is 1M[bytes], the throughput on the Web SHAKE (2 wireless links shared) is about twice as much as normal WWW access. We tested three variations of the cluster networks (100BASE-T, wireless LAN [11Mbps], and wireless LAN [1Mbps]). However, the throughput remained almost equal.

In SHAKE, mobile hosts that are connected by cluster links use multiple wireless links owned by each host simultaneously to communicate with hosts on the internet to offer high speed communication on wireless links. The cluster link needs to be a high-speed network in comparison with the sum of the external links as a premise of SHAKE. If the bandwidth of the cluster link is small, this link will cause a bottleneck and not be able to offer high speed communication. In this experiment, the bandwidth of the cluster link was large in comparison with the external link (PIAFS: 64Kbps) because the cluster network consists of a wireless LAN (1Mbps or 11Mbps) or wired LAN (100Mbps).

While the throughput is large when the data size is large, the throughput on the Web SHAKE is smaller than normal WWW access when the data size is small. However, in experiment-1, when mobile hosts receive small size data of less than 16K[bytes], the throughput is worse than normal WWW access (one wireless link is used). One of the reasons for this is the overhead of establishing the TCP connections establishment. Because the cluster network consists of a high speed LAN (100BASE-T) in this experiment, the time for the TCP's 3 Way Hand Shake (3WHS) between the mobile hosts can be ignored. However, the time for the TCP's 3WHS between the mobile hosts and the Web server is not small enough to be ignored.

2) **Experiment-2:** **Fig.7** shows the results of experiment-2. This graph shows the ratio between the throughput on the Web SHAKE (F-H REQ and F-P REQ) and the throughput on

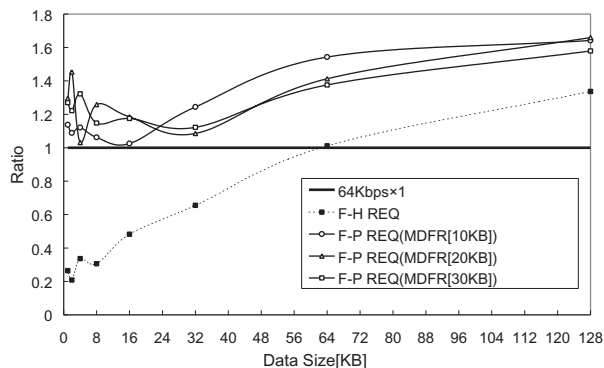


Fig. 7. Result of experiment-2(Throughput ratio)

normal WWW access. We call this ratio the performance improvement ratio. In this graph, Ratio F-H REQ is substantially smaller than 1 when the data size is 0 – 64 [KB]. When the F-H REQ is used to transfer data, the SHP first sends a HEAD request to the web server to obtain file information. When the transmission data size is small, the overhead of transmitting the HEAD information cannot be ignored.

On the other hand, Ratio F-P REQ is substantially greater than 1 even when the transmission data size is small. Because we can receive not only partial data but also the size of the whole data from the partial content header by using the F-P REQ, the SHP can receive all the data via a message exchange between the SHP and the web server at once of the small data. Therefore, the overhead of the F-P REQ is smaller than the F-H REQ, and the throughput is high when the data size is small.

3) *Experiment-3:* **Fig.8** illustrates the results of experiment-3. Fig. 8 shows the throughput on the Web SHAKE (F-H REQ and F-P REQ) and normal WWW access. From these graphs, we can see that even if the throughput of each wireless link is different, the throughput becomes large with use of the Web SHAKE. When the data size is 1M[bytes], the throughput on the Web SHAKE (2 wireless links shared) is about 1.9 times as much as normal WWW access. Furthermore, when three wireless links are shared, the throughput is about 2.8 times as much as normal WWW access through the use of both the F-H REQ and the F-P REQ. We can see that the performance throughput on the F-H REQ is substantially smaller than normal WWW access when the data size is 0 – 64 [KB]. However the throughput of the F-P REQ is greater than normal WWW access at all ranges (1[KB] – 1[MB]). If we use the F-P REQ, the Web SHAKE is effective all the time.

V. CONCLUSION AND FUTURE WORKS

We have proposed the Web SHAKE as a fast WWW access method using SHAKE, and implemented the SHAKE HTTP Proxy. Web SHAKE enables mobile hosts in a cluster network to communicate with any hosts on the internet through a HTTP

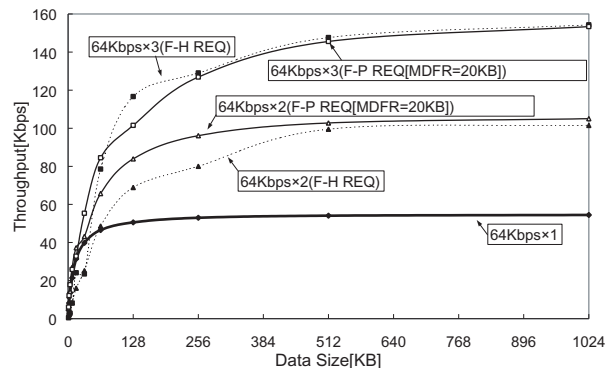


Fig. 8. Result of experiment-3 (Throughput)

proxy server that works on each mobile host. Although the available protocol is limited to HTTP, special software for the Web SHAKE is not required on the web servers.

We have shown that data transfer speed was improved by Web SHAKE in comparison with normal WWW access through three experiments. We proposed two variations of the Web SHAKE, the FH-REQ and F-P REQ. When the data size was large, the data transfer speed was improved substantially on both the F-H REQ and F-P REQ. The throughput on the Web SHAKE (2 wireless links shared) was about 1.9 times as much as normal WWW access. When three wireless links were shared, the throughput was about 2.8 times as much as normal WWW access through the use of both the F-H REQ and F-P REQ. In addition, even when the data size was small, the throughput on the F-P REQ was larger than normal WWW access.

We assume that the cluster network is Ad-Hoc. Therefore, the cluster network is dynamic and mobile hosts may join the cluster and leave at any time. In our current implementation of Web SHAKE, a cluster network is controlled by a SHP. However, because the cluster table that the SHP has is not configured dynamically, Web SHAKE cannot handle a mobile host that joins the cluster network halfway. In our future work, we plan to control the members and the dispersing rate in the cluster network dynamically.

REFERENCES

- [1] H. Mineno, et. al., "Multiple paths protocol for a cluster type network," *Int. J. Commun. Syst.*, Vol. 12, pp. 391 – 403, December 1999
- [2] K. Sklower, B. Lloyd, G. McGregor, D. Carr "The PPP Multilink Protocol (MP)," RFC1717, November 1994
- [3] K. Sklower, B. Lloyd, G. McGregor, D. Carr "The PPP Multilink Protocol (MP)," RFC1990, August 1996
- [4] Y. Kamio, F. kojima, M. fujise, "Implementation and Performance Evaluation of 384kbps-PHS Experimental system," *IEICE*, Vol. E83-B, No. 8, pp. 1844 – 1853 July 2000
- [5] Sharp, "Sharp Super Proxy Script – distributed proxy servers by URL hashing", <http://naragw.sharp.co.jp/sps/>
- [6] Squid WWW Cache system, <http://www.squid-cache.org/>
- [7] R. Fielding and H. Frystyk, "Hypertext Transfer Protocol – HTTP1.0," RFC1945, May 1996.
- [8] R. Fielding, et. al., "Hypertext Transfer Protocol – HTTP1.1," RFC2616, June 1999.
- [9] PHS Internet Access Forum meeting, "PIAFS Specifications," March 1997.