

Evaluation of Performance of TCP on Mobile IP SHAKE

KENJI KOYAMA,[†] YOSUKE ITO,[†] HIROSHI MINENO^{††}
and SUSUMU ISHIHARA^{†††}

We extensively evaluated a method of sharing multiple paths in a temporary network using MobileIP, especially its influence on the performance of the TCP. It uses multiple links between a temporary group of mobile computers and the Internet to deliver IP packets between a mobile computer in the group and its correspondent host on the Internet. We implemented the protocol on Linux and tested it on both emulated wireless and real IMT-2000 networks. We evaluated the performance of MobileIP SHAKE under various delay jitter conditions and found that it was degraded by out-of-order packets, which increased as delay jitter increased degraded the performance of the TCP. However, when there were two or more TCP flows, sufficient performance was obtained from the viewpoint of total throughput for each TCP flow by using multiple paths.

1. Introduction

Whenever people access the Internet while outdoors or on the move, they use mobile communication services (e.g., 2G, 2.5G, 3G mobile phones or public wireless LANs). Long-distance wireless services like mobile phones have narrow bandwidths, at most a few hundred kbps and low reliability compared to wired networks. Currently, many mobile computers have multiple network interfaces, such as mobile phones, wired LANs, wireless LANs, Bluetooth, or IrDA. Furthermore, many people carry such mobile information tools as laptops, i.e., PDAs or browser-phones. These multiple network interfaces on multiple mobile computers can be used for fast and reliable long-distance wireless communications. Our research group previously proposed a method of increasing communication speed and reliability in such environments, which we called Sharing multiple paths procedure for a cluster network environment, or *SHAKE*²⁾ for short. SHAKE assumes mobile computers have multiple network interfaces. Mobile computers with a fast wireless link (e.g., wireless LAN, and Bluetooth) temporarily form a network (called a cluster). When a mobile computer in a cluster accesses the Internet, it uses not only its own wireless link but also the links between other mobile computers to improve data transmission speed, reliability, and connectivity of communi-

cation. Our research group proposed a method to implement SHAKE on the IP layer, MobileIP SHAKE¹⁾, which requires no additional software on a correspondent host unlike²⁾ itself. This is accomplished by modifying the Mobile IPv4⁴⁾ home agent to distribute IP packets destined for a mobile host to multiple links between the Internet and cluster. To date, some technologies that extend Mobile IP to use multiple network interfaces for high speed or reliable communication have been proposed. Zhao et al. extended Mobile IPv4 to support multiple network interfaces and QoS⁵⁾. The WIDE InternetCAR project⁶⁾ extended Mobile IPv6 to support multiple network interfaces. This technology can be used to select one of the gateways between a mobile network on a vehicle and the Internet with the host in that network according to the host's policy. Although these technologies use multiple interfaces simultaneously or dynamically select one of the interfaces depending on the network environment, they do not use them simultaneously for a single flow. Our method uses multiple paths offered by temporarily connected multiple mobile computers to increase communication speed even for a single flow.

We observed that the throughput of a TCP flow improved by using MobileIP SHAKE¹⁾, when the bit rate of each link was sufficiently slow against the bit rate of a link within a temporary group of mobile computers. However, we also observed that the throughput did not improve as the bandwidth and delay of each path to be used increased. In this paper, we evaluated the performance of MobileIP SHAKE

[†] Graduate School of Engineering, Shizuoka University

^{††} Faculty of Engineering, Shizuoka University

^{†††} Faculty of Information, Shizuoka University

on an emulated and a real IMT-2000 network. We first increased the TCP transmitting buffer size. We then evaluated the influence out-of-order packets had on the TCP of MobileIP SHAKE. Finally, we evaluated the performance of MobileIP SHAKE when there were two or more different flows simultaneously. The rest of the paper is organized as follows. Section 2 outlines the MobileIP SHAKE and Section 3 describes its implementation. Experimental results obtained from a wireless emulation network and a real IMT-2000 wireless network are presented in Section 4. Section 5 summarizes the paper.

2. Mobile IP SHAKE

2.1 SHAKE on IP

To use multiple paths between the Internet and a temporary group of mobile computers (a cluster), the mobile nodes (MN) in that cluster, and either a correspondent node (CN) on the Internet or intermediate nodes on their paths, have to know the multiple paths between the Internet and the cluster. However, it is not realistic to assume that the CN of MNs will support such a function. Although a SHAKE-like protocol can be implemented in the application layer (like our first implementation of SHAKE), to do so requires that the CN knows the multiple paths to a cluster and this requires additional software. Additional application layer software on a CN should not be required because this prevents an MN in a cluster from communicating with general hosts on the Internet. This problem is solved by using mobile IP. If a route optimization function is not used, Mobile IP uses a home agent (HA) to deliver IP packets to an MN. Even if the CN does not know the current location and the care-of-address (CoA) of the destination MN, IP packets to the MN are received by the HA and forwarded to the MN. If the HA knows the IP addresses of MNs in the cluster to which the destination MN belongs, it can forward the IP packets to the MNs in the cluster through the multiple paths between the cluster and the Internet even if the CN does not know the destination MN is in the cluster. We called this method *Mobile IP SHAKE* and Fig. 1 has an overview of it.

2.2 Mobile IP SHAKE

The mechanism for Mobile IP SHAKE is explained below. We assumed MNs in a cluster would share information about their network resources, e.g., the maximum bandwidth and

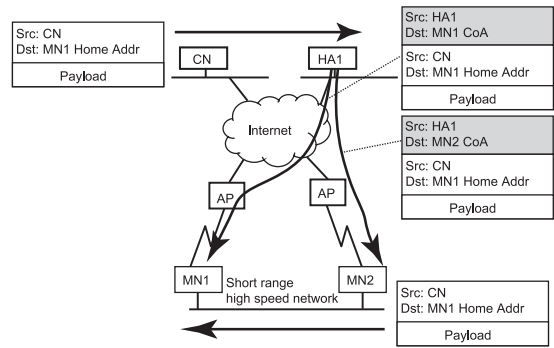


Fig. 1 Mobile IP SHAKE.

cost of the link to its Internet access point, the IP addresses for outside links and for the cluster, and other computing resources with messages broadcast every two seconds. An MN in a cluster can select other MNs with which to cooperate to communicate with a host outside the cluster, by using this information and requesting an alliance. If a mobile node (MN1) wants to use the link of another mobile node (MN2) (Fig. 1), the MN1 sends an *alliance request* to the MN2 to order it to cooperate in communicating with hosts outside the cluster. If the MN2 receives the request from the MN1 and agrees with it, it sets a route from outside the cluster to the MN1 through the MN2 and the reverse route from the MN1 to the Internet. The MN2 then sends an affirmative reply to the MN1. Hereafter, we will call a node like the MN2 that offers communication resources to another node a relay node (RN). If an MN receives an affirmative reply to an alliance request from an RN, it registers the IP address for the outside link of the RN with its HA. The MN also adds a route destined for networks outside the cluster via its RN. The HA of an MN distributes IP packets to the MN to the care-of-address of the MN and the registered care-of-address of the RN. This is achieved by setting the destination address of the header of encapsulated IP packets to the address of the MN's RN. To use multiple paths to an MN in a cluster efficiently, it is important to distribute packets to an adequate path according to the condition of each path. The details of this will be discussed later. When an RN receives an IP packet from the HA of its client MN, it decapsulates the packet and forwards it to the client according to its routing table, which was configured for its client MN when it received the alliance request message. Packets from an MN

in a cluster are distributed by the MN itself. Some of the packets are sent through a link between the MN and the Internet. Others are forwarded to RNs and sent to the packet destination via their links to the Internet. To avoid a loop within the cluster, each MN must only distribute packets from itself to its RN. Packets from MNs are not encapsulated as well as in the original Mobile IP.

2.3 Control Messages

An alliance request and reply message are exchanged by an MN and its RN to create an MN alliance. The same format is used for the messages. A Mobile IPv4 registration request message is used to register the care-of address of an RN. A reserved bit in the format of the registration request is used to tell the HA that the registered care-of address is not owned by the registering MN but by SHAKE's RN. In addition to these messages, MNs and HAs periodically exchange short messages to measure the round trip delay between them. The MNs in a cluster also periodically send broadcast messages to the cluster to advertise of their capabilities. When an MN does not receive an advertisement message from its RN for a given length of time, it sends a registration request to its HA to remove the RN's care-of address. When an MN receives an advertisement from a new MN in the cluster, it sends an alliance request message to the new MN if it decides to cooperate with the new MN.

2.4 Distribution of Packets by a Home Agent

Forwarding packets via an appropriate path to an MN in a cluster at the HA is important for efficient communications. To ensure high reliability, the HA should send the same packets to an individual path to the destination MN. To archive high speed communications, on the other hand, the HA should distribute packets along multiple paths to the destination MN via RNs so that the bandwidth of each link is efficiently used. If the required bandwidth of each flow is smaller than the maximum available bandwidth of each path, packets should be distributed by the per-flow rule not by the per-packet rule. This is because the per-packet rule may cause many out-of-order arrivals at the receiver. However, if the required bandwidth of a flow is larger than the maximum available bandwidth of the paths, e.g., rich streaming video, the per-packet rule must be used. If the per-packet rule is used, distributing packets to each

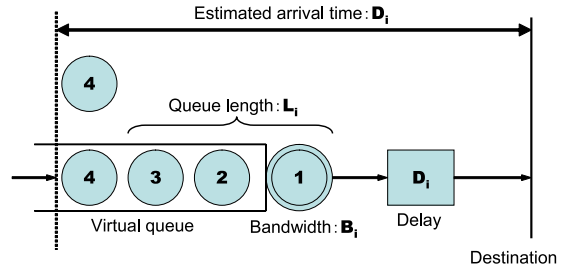


Fig. 2 Model of virtual transmission queue.

path is important so that out-of-order arrivals that degrade the performance of the TCP and cause frames to be dropped in streaming video and so on can be avoided at the receiver. Thus far, we have implemented the per-packet rule in the HA because it can use a larger bandwidth for each path than the per-flow rule. However, the per-packet rule has a problem where out-of-order packets may frequently be generated by using multiple paths.

2.5 Traffic Dispersion Method

To avoid out-of-order packets, we implemented the following two policies enabling us to adopt the conditions of the paths in the prototype.

D The HA selects a path randomly according to the ratio of measured round trip delay between the HA and each MN.

DQ In our packet dispersion method, each packet is sent to a path where the sum of the waiting time at the transmission queue and the delay time reported from the client is the shortest. Virtual transmission queues are used to predict the transmission delay at each path. **Figure 2** shows the model.³⁾

The parameter i indicates the ID of each path. Here, S , B_i , L_i , τ_i , and t are packet size, effective bandwidth, queue length, the time when a packet was sent, and the present time, respectively. L_i is recalculated using S , B_i , and $\tau_i - t$ when a new packet is transmitted. Waiting time, W_i in the transmission queue is calculated by the following equation (see **Fig. 3**):

$$W_i = \frac{S}{B_i}(L_i + 1) + \tau_i - t$$

The time passing before the packet arrives is estimated by the following equation:

$$D_i = W_i + D_i$$

where d_i is the estimated transmission delay time, which is predicted from a feedback message from the client. Each packet is sent to the path, i , that has the shortest

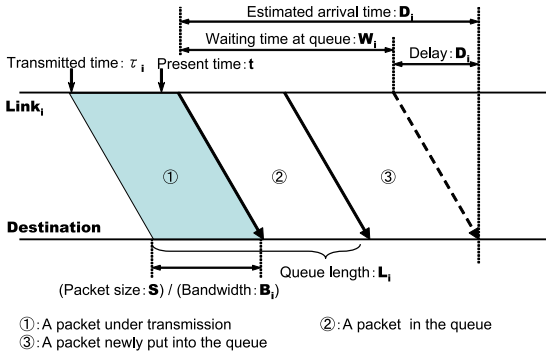


Fig. 3 Waiting time at a transmit queue and arrival prediction time.

D_i .

In our implementation, the round trip delay between HAs and MNs were measured by exchanging periodic packets. The maximum bandwidth of each link between the access point and each MN was assumed to be static.

2.6 Distribution of Packets from a Mobile Node

Packets relayed by an HA to MNs in a cluster can be distributed according to the measured conditions of paths between the HA and MNs. Packets from an MN to its CN, on the other hand, have insufficient information for distribution because they are not relayed by an HA that has the ability to monitor the path between itself and MNs, and the CN does not have such ability. The maximum bandwidth of the outside link of each MN, the round trip delay at the last hop from the Internet to each MN, the load of each MN are the criteria for selecting a path. Of course, packets from an MN can be relayed to the HA through encapsulation. Then, the same strategy as when an HA forwards packets to MNs can be used.

3. Implementation

We implemented a Mobile IP SHAKE prototype on Linux (kernel 2.2). To distribute IP packets to MNs at an HA, we modified an implementation of “simultaneous binding” function⁸⁾ on Dynamics HUT Mobile IP⁷⁾. Simultaneous binding is a technology that enables the same IP packets to be forwarded to the multiple care-of addresses of one MN to enable smooth handover. We modified the behavior of the HA so that it could distribute packets to multiple addresses instead of sending the same packets to them. To notify the HA to use SHAKE instead of simultaneous binding, we added a 1-bit

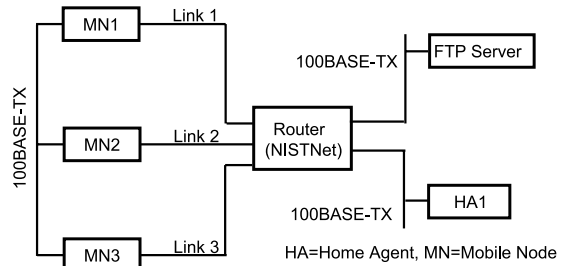


Fig. 4 Experimental network for emulating wireless network.

option to the registration request message of the mobile IP. In addition to this extension to Mobile IPv4, we added the message exchange function stated in the previous section. However, the per-flow traffic distribution, and the traffic distribution at MNs were not implemented in this prototype.

4. Performance Evaluation

We evaluated the performance of the Mobile IP SHAKE in an emulated wireless and a real IMT-2000 wireless network.

4.1 Experiments on an Emulated Wireless Network

Figure 4 outlines the network topology for the experimental network. Three MNs with two fast Ethernet interfaces were connected, and they were also connected to a PC with multiple network interfaces and runs the NISTNet⁹⁾ network emulator. The PC is denoted as a router in Fig. 4. The HA of one of the MNs and an FTP server were connected to the different network interfaces of the router. The throughput when a 1-MB (or 4-MB) file was transferred from the FTP server to one of the MNs was measured. The bandwidth of the link between the MNs and the router (Link 1-3) was set to 64, 128 or 384 kbps, and each link was symmetrical. The delay in the link between the MNs and router was set to 300 msec. A configuration where only one MN was connected to the router was also tested for comparison.

Figure 5 shows the results when the jitter of the link between the router and MNs was set to 0, 50, or 100 msec. The packet loss rate for all the links was set to 0%. Figure 5 (a) and (b) shows that the DQ outperformed D because it estimated the packet arrival time more accurately. Generally, large jitter causes out-of-order packets especially when multiple links are used. Out-of-order packets prevent the TCP congestion window from increasing and

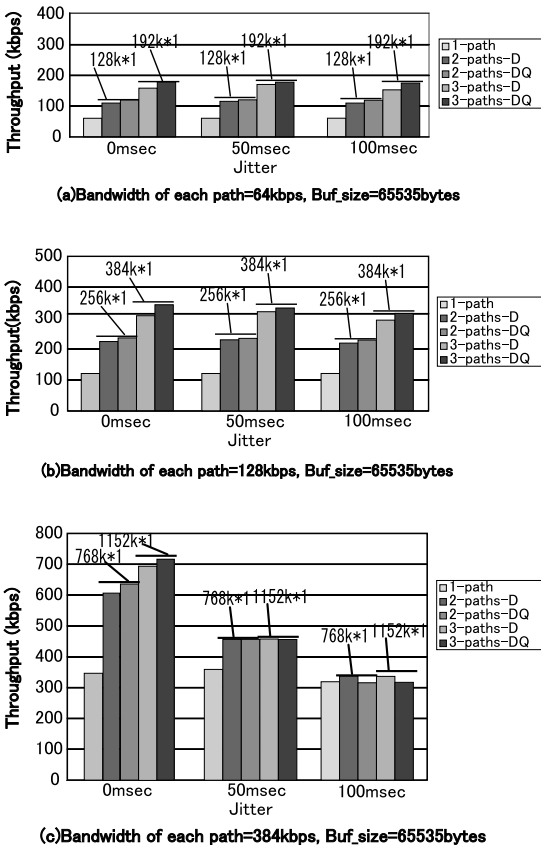


Fig. 5 Throughput of TCP flow before change TCP transmitting buffer size. (a) 64 kbps, (b) 128 kbps, (c) 384 kbps.

degrade TCP performance. However, the performance of MobileIP SHAKE (D and DQ) in this case was almost equal to that of a network with a single path with the same bandwidth as the total bandwidth of links shared by MobileIP SHAKE. Figure 5 (c) shows that performance was not improved when the bandwidth of each path was 384kbps. The reason for the decrement performance of the TCP may have been due to the deflection of the correspondent host's TCP transmitting buffer. As the total bandwidth of each path increases at a correspondent host, the amount of TCP transmitting buffer consumed (default value on Linux is 65,535 bytes) was increased because of the large bandwidth and the low delay in paths. To improve the TCP performance, we increased the TCP transmitting buffer so that it was larger than the default value on Linux. **Figure 6** plots the results when the correspondent host's TCP transmitting buffer was increased from 65,535 to 256,000 bytes. Performance was

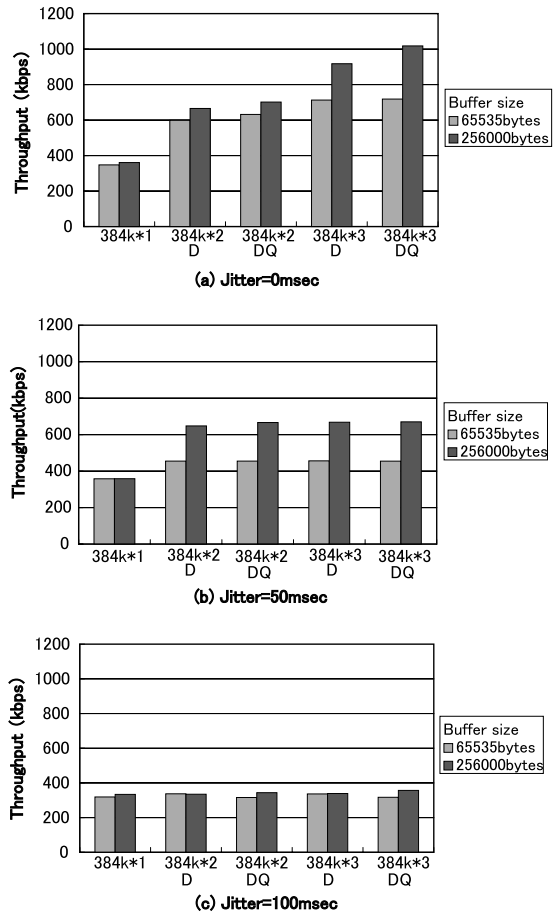


Fig. 6 Comparison of throughput before and after change in TCP transmitting buffer size. (a) jitter= 0 msec, (b) jitter= 50 msec, and (c) jitter= 100 msec.

evaluated near the total maximum bandwidth of each path when the delay jitter was 0 msec (Fig. 6 (a)). However, as the jitter was increased (Fig. 6 (b) and (c)), the TCP performance on MobileIP SHAKE declined even with the larger TCP transmitting buffer.

Figure 7 plots the time transition for throughput and the number of TCP sequences with jitter of 100 msec. We can see that out-of-order packets occurred frequently and TCP performance decreased as the number of paths increased. However, using this solution forces the correspondent host to change the size of the TCP transmitting buffer. Consequently, a method that does not force any change on the correspondent host is required. To use on HA to terminate a TCP connection like Indirect TCP would be effective.

In practice, there are simultaneous multiple

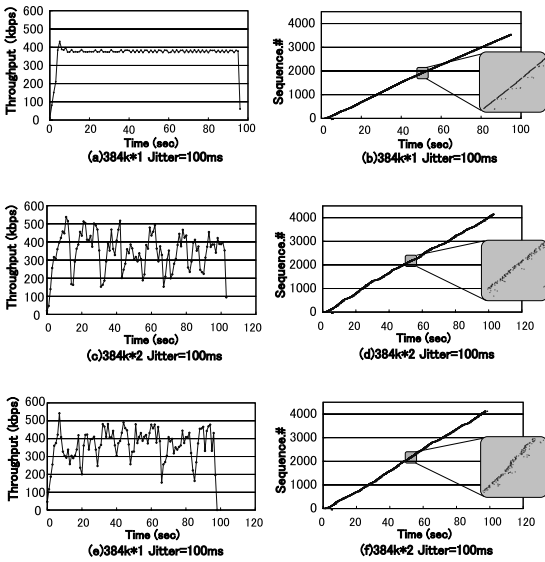


Fig. 7 Time transition for throughput and number of TCP sequences with jitter of 100 msec.

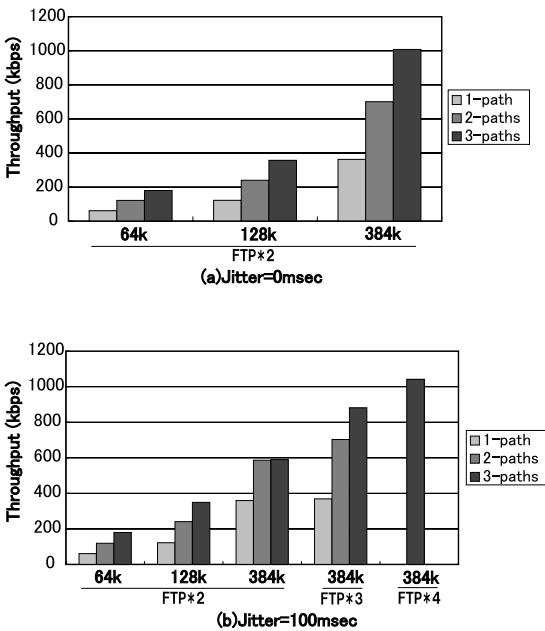


Fig. 8 Results for emulated network with two or more FTP flows (a) jitter = 0 msec and (b) jitter = 100 msec.

flows at one of the MNs, for example, when we use multiple TCP flows to deliver images. The measured results assuming that the flow of multiple FTP-connections occur simultaneously in an MN are plotted in Fig. 8. The method of distribution was DQ. The throughput was calculated as the total for all FTP connection throughput. As we can see in Fig. 8 (a),

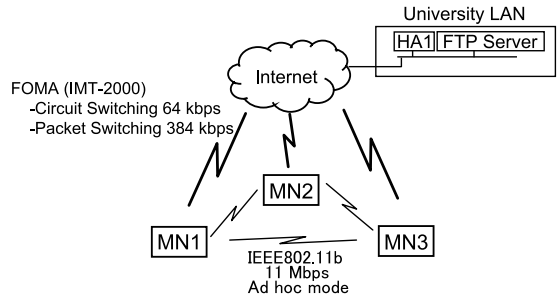


Fig. 9 Experiment network topology.

when delay jitter was 0 msec and multiple paths were used, throughput near the total maximum bandwidth for each path was obtained. When there was only one FTP connection, although we used multiple paths, there was only 300 kbps of throughput (Fig. 6 (c)). However, as we can see from Fig. 8 (b), when there were two FTP connections, the total throughput for all connections improved to 600 kbps. Moreover, when 3 or 4 FTP connections were used, throughput near the total bandwidth for all paths was obtained. This means the per-packet rule can also be used as a distribution method for MobileIP SHAKE to improve TCP throughput.

4.2 Experiments on a Real IMT-2000 Wireless Network

Figure 9 outlines the network topology for the experiment in a real IMT-2000 wireless network. Two or three mobile PCs (MNs) were connected by an IEEE 802.11b wireless LAN. They were also connected to the Internet via NTT DoCoMo's IMT-2000 service, FOMA¹⁰. An FTP server and the HA of an MN were connected to our university's campus LAN. The link between the university LAN and the Internet had a sufficiently wide bandwidth. The wireless links between MNs and the Internet were bottlenecks. We measured the throughput when a 1-MB (or 4-MB) file was transferred from the FTP server to one of the MNs. FOMA offers two types of services for data communication. One is a 64-kbps circuit switching service, the other is a 384-kbps packet switching service. We used both these services in our experiments. We evaluated the performance of MobileIP SHAKE in a real IMT-2000 wireless network. Figure 10 shows the results before and after changing the size of the correspondent host's TCP transmitting buffer.

The results when a 64-kbps circuit switch service was used are in Fig. 10 (a). The delay and packet loss rate between the HA and each MN

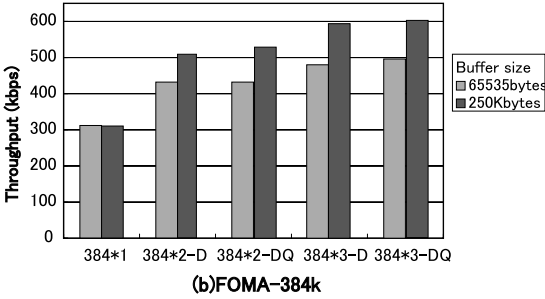
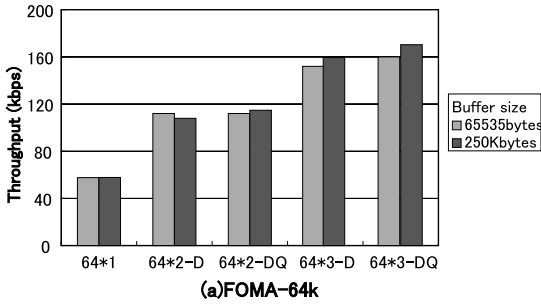


Fig. 10 TCP throughput on FOMA (a) 64 kbps circuit switching service, (b) 384 kbps packet switching service.

were almost 200 msec and 0.08%. When we used the DQ packet distribution policy, we observed nearly double and triple throughput for a single 64-kbps link. Additionally, even when the correspondent host's TCP transmitting buffer was increased from 65,535 to 256,000 bytes, there was no difference in performance. Figure 10(b) has the results for when 384-kbps packet switching services were used. The delay and packet loss rate between the HA and each MN were almost 210 msec and 0.08%. The figure shows that the performance improved by increasing the correspondent host's TCP transmitting buffer. For example, when two MNs were used with the DQ policy and the correspondent host's TCP buffer was increased, the throughput was about 1.7 times more than when a single 384-kbps link was used. However, the performance when three MNs were used was almost the same as when two MNs were used. One of the reasons for this is that many out-of-order packets occurred at the receiver. This was caused by the addition of a link with large jitter. FOMA links, basically, have long delay and large jitter. Sharing such links caused many out-of-order packets at the receiver. This degraded the performance of the TCP. Additionally, Fig. 10 shows that the performance

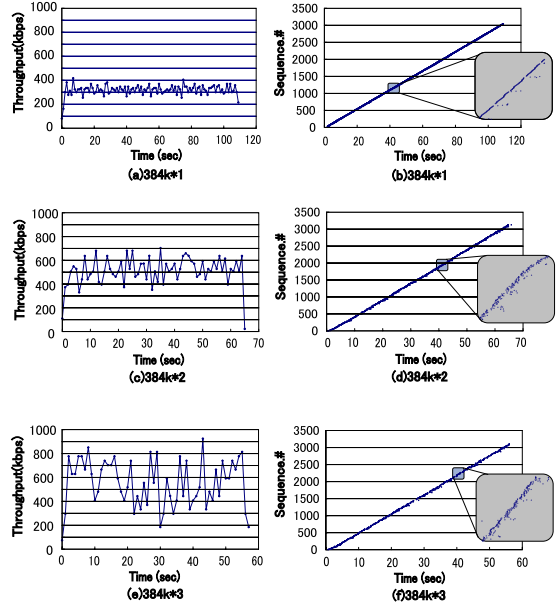


Fig. 11 Throughput and number of sequences using FOMA.

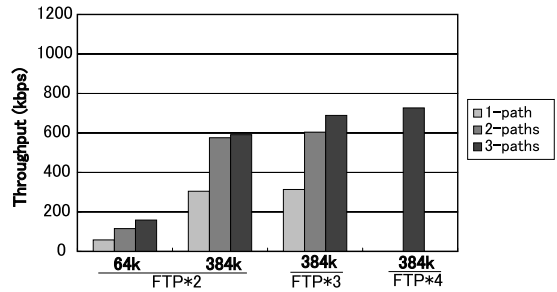


Fig. 12 Results for FOMA with two or more FTP flows.

of MobileIP SHAKE on the IMT-2000 network was close to that when jitter was 50 msec on the emulated network (Fig. 6 (b)). **Figure 11** plots the time transition for throughput and the number of TCP sequences when using the 384-kbps packet switching service. It also shows that TCP performance was degraded by out-of-order packets. To avoid this, a smart packet distribution policy and a smart path monitoring method would be required.

Figure 12 plots the results when there were multiple FTP connections. Even when there were two or more TCP flows, the throughput did not improve like it did in the emulation environment (Fig. 8). This was because the uplink bandwidth was insufficient when using FOMA's 384-kbps packet switching service and multipaths. This was because FOMA's packet

switching service had an asymmetrical link with an uplink bandwidth of 64 kbps. To solve this problem, packets from MNs to the CN have to be distributed to each link from the cluster to the Internet. However, at the moment, traffic distribution at MNs have not been implemented.

5. Conclusion

We evaluated a method of sharing multiple paths in a temporary network using MobileIP, especially its influence on the performance of the TCP when the per-packet traffic dispersion rule was used. The experiment results revealed that when the bandwidth per path was sufficiently small, e.g., 64 kbps or 128 kbps, throughput near the total maximum bandwidth per path could be obtained by using multiple paths, and when there were multiple TCP flows, MobileIP SHAKE's performance was improved compared to when a single flow was used. This meant that the per-packet traffic dispersion rule could be useful for MobileIP SHAKE. However, our results also revealed that when the jitter of delay of shared multiple links were large, the performance of MobileIP SHAKE was degraded by out-of-order packets frequently caused by transmission through multiple paths. The development of a traffic distribution mechanism to avoid out-of-order packets, and the implementation of a distribution mechanism for up-link traffic remain topics for future research.

References

- 1) Ishihara, S., Koyama, K., Ito, Y., Mineno, H. and Ohta, K.: Sharing multiple paths for temporarily connected mobile computers using Mobile IP, *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, Vol.12 (Aug. 2003).
- 2) Mineno, H., Ishihara, S., Ohta, K., Aono, M., Idecuchi, T. and Mizuno, T.: Multiple paths protocol for a cluster type network, *International Journal of Communication Systems*, Vol.12, pp. 391-403 (Dec. 1999).
- 3) Saito, Y., Ishihara, S., Mizuno, T. and Watanabe, T.: Evaluation of Traffic Dispersion Methods for Synchronous Distributed Multimedia Data Transmission on Multiple Links, *International Conference on Advanced Information Networking and Applications (AINA)*, pp. 644-649 (Mar. 2003).
- 4) Perkins, C.: IP Mobility Support, RFC 2002 (Oct. 1996).
- 5) Zhao, X., Castelluccia, C. and Baker, M.: Flexible Network Support for Mobility, *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom1998)* (Oct. 1998).
- 6) Wakikawa, R., Uehara, K. and Murai, J.: Multiple Network Interfaces Support by Policy-Based Routing on Mobile IPv6, *Proc. ICWN 2002*, pp. 9-17 (June 2002).
- 7) Dynamics HUT Mobile IP.
<http://www.cs.hut.fi/Research/Dynamics/>
- 8) Hartenstein, H., Jonas, K. and Schmitz, R.: Seamless Inter-Domain Handoff via Simultaneous Bindings, *Proc. European Wireless 2000* (2000).
- 9) NIST Net.
<http://snad.ncsl.nist.gov/itg/nistnet/>
- 10) NTT DoCoMo FOMA/3G.
<http://www.nttdocomo.com/foma/3g/index.html>

(Received April 15, 2004)

(Accepted July 1, 2004)



Kenji Koyama was born in 1980. He received his Diploma in System Engineering from the Shizuoka University in Japan in 2003. He is a master candidate of the Shizuoka University. His current work is in the field of Mobile TCP/IP networkings. His research interests are in the field of Wireless Communications for the support of multimedia services. He is a member of IPS Japan.



Yosuke Ito was born in 1981. He received his Diploma in System Engineering from the Shizuoka University in Japan in 2003. He is a master candidate of the Shizuoka University. His current work is in the field of Mobile TCP/IP networkings. His research interests are in the field of Wireless Communications for the support of multimedia services. He is a member of IPS Japan.



Hiroshi Mineno was born in 1974. He received the BE and the ME degrees from Shizuoka University, Japan in 1997 and 1999, respectively. Between 1999 and 2002 he was a researcher in the NTT Service Integration Laboratories. In 2002, he joined the Department of Computer Science of the Shizuoka University as an Assistant position. His research interests include network architectures and protocols for ubiquitous networking, including wireless communication and mobility. A member of the Information Processing Society of Japan (IP SJ), and the Institute of Electronics, Information and Communication Engineers (IEICE).



Susumu Ishihara was born in 1972. He received his BSEE and Ph.D. degrees from Nagoya University in 1994 and 1999, respectively. From 1998 to 1999 he was a Special Researcher of JSPS fellowship. Since 1999, he has been with Shizuoka University, where he is an Associate Professor in the Department of Systems Engineering. His research interests are in the areas of mobile TCP/IP networking, mobile computing, and mobile ad hoc networking. He is a member of IEEE, ACM, IEICE Japan and IPS Japan.
