Implementation and Evaluation of a Network-Controlled Mobility Management Protocol (IP^2MM):

Performance Evaluation Compared with Mobile IPv6

Katsutoshi Nishida
NTT DoCoMo Inc.
3-5 Hikarinooka,
Yokosuka-shi, Kanagawa,
239-8536 Japan
nishidak@nttdocomo.co.jp

Shin-ichi Isobe
NTT DoCoMo Inc.
3-5 Hikarinooka,
Yokosuka-shi, Kanagawa,
239-8536 Japan
isobes@nttdocomo.co.jp

Tomohiko Yagyu
NEC Corporation
31753 Shimonumabe,
Nakahara, Kawasaki
211-8666, Japan
yagyu@cp.jp.nec.com

Ippei Akiyoshi
NEC Corporation
31753 Shimonumabe,
Nakahara, Kawasaki
211-8666, Japan
i-akiyoshi@ah.jp.nec.com

Abstract— **The 3rd-generation mobile network, IMT-2000, was launched some years ago and multimedia traffic in the mobile Internet has been increasing with the growth of wireless access links. Next-generation mobile networks need to efficiently transport huge volumes of multimedia traffic with high-quality mobility management. We have proposed an IP-based IMT Network Platform (IP^2) for the next-generation mobile network architecture and have studied its mobility management scheme, IP^2 Mobility Management Protocol (IP^2MM). IP^2MM is a network-controlled mobility protocol that deploys mobility functions inside the network and conceals IP mobility from the mobile node. It is also designed to satisfy the requirements of mobile operators. In this paper, we introduce an overview of IP^2MM and show the comparative evaluation results with Mobile IPv6. For the evaluation, we implemented IP^2MM and Mobile IPv6 in our experimental system and compared their performance. The results show that IP^2MM provides better mobility performance than Mobile IPv6, and we conclude that IP^2MM is an effective and feasible mobility protocol for the next-generation networks of mobile operators.**

Keywords; IP mobility, network-controlled, IP^2MM

I. INTRODUCTION

3G network service has started and wireless access speed, a mobile-specific bottleneck, has improved. With the advent of broadband wireless links, mobile Internet traffic has dramatically increased. Moreover, future wireless technologies such as HSDPA (High Speed Downlink Packet Access) [1] will make the mobile Internet more common, and its traffic will keep increasing for the future. Therefore, next-generation mobile networks will need to economically transport huge amounts of multimedia traffic in higher quality than current mobile networks.

As for the transport technology in the next-generation mobile networks, our shared vision is discussed in ITU-R that all telecommunication networks are eventually going to shift to IP networks [2]. 3GPP members also discussed network evolution from IMT-2000 to all IP networks under the name of IP Multimedia Sub-system (IMS) which aims to provide IP-based real-time multimedia services [3]. IP technology is assumed to be effective in realizing next-generation mobile networks.

On the other hand, in the Internet world, the Internet Engineering Task Force (IETF) has discussed IP mobility and standardized Mobile IPv4 (MIP) [4] that provides IP mobility to the mobile node (MN). Mobile IPv6 (MIPv6) [5] has been also standardized recently, and it has been improved from MIP and supports IPv6. Although MIP and MIPv6 achieve MN's IP mobility, they are terminal-based mobility protocols that follow the fundamental rule of the Internet, the so-called End-to-End principle. Therefore, a number of issues such as handover performance degradation due to the IP address configuration by the MN have been raised. There are many proposals to overcome MIP and MIPv6 issues, and some efforts to tackle them take the approach where the network supports MN mobility functions such as Proactive Handover [6] for MIP and Fast Handovers for Mobile IPv6 (FMIP) [7]. Another approach has been taken by the BRAIN Candidate Mobility Management Protocol (BCMP) [8], for example, where the network controls and manages the MN's mobility differently from the MIP family.

We have proposed the IP-based IMT Network Platform (IP^2) [9] which evolves from IMT-2000 and aims to achieve an all-IP next-generation mobile network. As for mobility management in IP^2, we have also studied IP^2 Mobility Management Protocol (IP^2MM) [10]. IP^2MM is a network-controlled mobility protocol where mobility management functions are deployed on the network side as much as possible and the network controls the MN mobility. It has been designed to satisfy the mobile operators' requirements described in [11][12] as well. To evaluate IP^2MM performance, we implemented it in our experimental system, and for comparison, we also installed MIPv6.

In this paper, we discuss the design policies and basic mobility control procedures of IP^2MM and present experimental results. Then, we analyze them while comparing them with MIPv6 and conclude that IP^2MM provides better mobility performance than MIPv6 and is suitable for the next-generation networks of mobile operators.

In section 2 of this paper, we briefly discuss the MIPv6 procedures and its issues, and analyze previous works that improve handover performance identified as the issue of MIPv6. In section 3, we clarify the mobile operators' requirements and present the protocol designs and basic mobility control procedures of IP^{2}MM. In section 4, we present the measurement results of IP^{2}MM and MIPv6 regarding two performance aspects, i.e. mobility control and packet transportation, and analyze them. Finally, we summarize the paper by offering conclusions and items for further study in Section 5.

II. RELATED WORK

In IP networks, the IP address has two roles: one is the identifier which specifies the MN and the other is the locator which indicates the MN's location. Used as the locator, IP addresses have to be changed when a MN moves from one network to another. Thus, the MN cannot continue to communicate because the identifier, i.e. IP address, for the application also changes when the MN moves. To solve the IP mobility issue, MIP was discussed in IETF and its IPv6 version, MIPv6, has been standardized. In MIPv6, the Home Address (HoA), which is assigned by the home network, is used in the upper layers above Transport to identify the terminal, whereas the Care-of Address (CoA), which the MN configures at the currently connected network, is used for actual packet transportation in the Network layer. Controlling these two different IP addresses and concealing IP address changes from its application, MN provides IP mobility by itself. MIPv6 introduces Home Agent (HA) in the home network, and the mapping of the HoA and the CoA, which is called Binding Cache, is managed by the HA as location information. Sending Binding Update periodically and when the MN moves to another network, the MN updates its location information maintained in HA. HA also has the function to forward a packet designated to the HoA of the MN, which is currently away from home, to its CoA by referring to the Binding Cache. In this case, packet transmission between the MN and the correspondent node (CN) is performed through the tunnel between MN and HA that is called bi-directional tunneling. In MIPv6, the route optimization function is also provided by MN informing its CoA to the CN after handling the Return Routability procedure.

Studied as the extension of IPv6, MIPv6 is a terminal-based mobility protocol and requires MN to configure the IP address used in its connecting subnet through Stateless Address Autoconfiguration [13], for example. In this procedure, the MN has to confirm the uniqueness of configured IP address in the subnet, which is called Duplicate Address Detection (DAD) specified in [13]. As DAD takes more than 1 second, MIPv6 has difficulty providing fast handover to the MN, so that it is critical for the real-time applications.

To solve this issue, FMIP [7] has been proposed where the network has the intelligence and improves the handover latency by conducting time-consuming procedures such as DAD instead of the MN. In FMIP, the MN or network is assumed to have the function of acquiring the MN's handover trigger. Given the intelligent function, upon receiving the handover trigger of the MN, the access router (AR) conducts the DAD procedure before the MN actually moves between ARs. When the network obtains the MN handover trigger, the old AR (oAR), to which the MN is currently connected, indicates MN handover to a new AR (nAR), which the MN is going to attach. Then, the nAR performs DAD for the MN and informs the MN of its IP address valid under the nAR link in advance of the MN handover. In this way, FMIP provides fast handover by locating the mobility control function in the AR and supporting MN mobility procedures. The same approach was taken in the Proactive Handover study [6] based on MIP where the oAR and nAR cooperate with each other and create new CoA and complete the Binding Update before the MN moves.

On the other hand, the network-controlled mobility management approach has been employed in the BCMP [8] instead of employing the terminal-based mobility protocol, MIP, as its base protocol. BCMP introduces an anchor router (ANR) which aggregates multiple ARs and provides mobility service to the MN through the collaboration between ARs or AR-ANR. When the MN connects to an AR, the upper ANR allocates one IP address to the MN. In the case of intra-ANR handover, obtaining MN handover trigger, the oAR sets up a tunnel from oAR to nAR, for example. After the MN moves to nAR and nAR finds the MN's handover, ANR sets up a routing path between ANR and nAR according to the handover indication from the nAR. In this way, BCMP processes mobility management within the network, and the MN has minimal functions such as movement detection and notification.

III. IP^2 MOBILITY MANAGEMENT PROTOCOL

As discussed above, we can roughly categorize the mobility protocols into two approaches: one is terminal-based and the other is network-controlled. We have proposed the IP^2 Mobility Management Protocol (IP^{2}MM) which employs the network-controlled approach not only to achieve high-quality mobility to the MN but also to achieve the mobile operators' requirements that are not considered in the Internet world. In designing IP^{2}MM, we have placed great importance on these requirements to build a mobility protocol suitable for mobile operators.

A. Protocol Requirement

In the current mobile operator's network, the location of the MN is hidden from the CN. Location privacy is one of the important requirements when the operator provides mobility services (R1). It is also desirable to achieve route optimization at any time to minimize packet transmission delays (R2). This is especially important if the MN roams into a visited network far away from home. As for the mobile circumstances, the battery saving of the MN should be considered (R3). Furthermore, the new mobility protocol should be able to provide higher quality mobility services than current ones such as MIPv6 or GPRS

[14] in 3G networks (R4). As discussed concerning security issues in the Internet world, the new mobility protocol must eliminate possible security risks, such as DDoS Attack, by concealing topology information of the network including nodes or servers addresses that control mobility functions located inside the network (R5). Due to the possible bottleneck, wireless link resource usage should be minimized by reducing the packet overhead and the number of signalings in the wireless link (R6). In addition, when the number of subscribers increases or the traffic pattern changes, the network should be able to expand the mobility control and packet transmission capacity independently for the sake of cost efficiency (R7).

B. Design Policy

As we discussed in section 2, if a mobility protocol provides high-quality handover, it needs, to some extent, network assistance instead of depending all mobility functions on the MN. Furthermore, although there are protocols, such as FMIP, which extend the MIPv6 and place some mobility functions in the network, they cannot achieve mobile operators' requirements. This is because these protocols assume that the MN sends Binding Update directly to the HA, so that the HA address should be informed to the MN. Thus, the requirement of topology hiding (R5) cannot be achieved by revealing the address of HA, which is the important mobility control server inside the network. One of the other reasons is that MIP-based protocols require the MN to process mobility functions such as IP address management or periodical location update, causing the MN to require more battery power. It is thus a concern that battery saving (R3) cannot be achieved. From the above discussion, we assumed that the mobility protocols that achieve mobile operators' requirements should not be the extension of terminal-based mobility protocols. We therefore employed the network-controlled approach and designed IP^{2}MM based on the four design policies, illustrated in Figure 1.

IP^{2}MM is characterized by separating the two functions of the legacy IP address -- the identifier, and the locator -- where the MN uses only an IP address that is used as the MN identifier. In other words, IP^{2}MM defines the IP-host Address (IPha) as the identifier and the IP-routing Address (IPra) as the locator (Policy A). The Packet is transported by the IPha between the MN and the AR, whereas the routers in the network use the IPra to forward it. The address of the packet is translated from IPha to IPra and visa versa at the AR, so that the MN can send/receive a packet only by using IPha. The mapping between IPha and IPra is managed by Routing Manager (RM) and it informs AR to create, update and delete IPha/IPra mappings. Employing this function, the network can provide location privacy to the MN as it can communicate with CN by the IPha that does not show the MN's location information (R1).

Next, we locate the intelligent functions at network edge nodes such as the AR and the Boarder Router (BR) (Policy B). Those edge nodes perform the proxy functions for the MN such as address translation or mobility control. By the edge nodes routing a packet with the IPra, which indicates the current location of the MN, the packet can be routed by the optimal route (R2). Moreover, as the AR terminates the signaling from the MN, the MN does not need to know the addresses of mobility control servers, so that the network topology is hidden at the same time (R5).

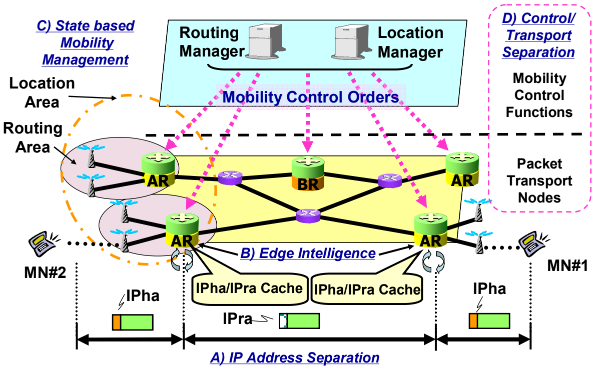


Figure 1: IP^2 Mobility Management Architecture

We also defined two states of MN, Active and Dormant. In the Active state, the MN is in communication, while in the Dormant state, the MN is not in communication. MNs in Active and Dormant states are managed by the Routing Manager (RM) and Location Manager (LM), respectively (Policy C). The IPra is allocated only for the Active MNs, and it changes when the MN moves to another Routing Area (RA). In IP^{2}MM, we treat the RA as the subnet. Maintaining the IPha/IPra mapping, RM manages the routing information of Active MNs. On the other hand, the LM tracks the location of the Dormant MN in the granularity of Location Area (LA), which consists of multiple RAs. The LA is identified by the Location Area ID (LAID), and the mapping of IPha and LAID is maintained by the LM. Introducing a state-based mobility management scheme, the MN in Dormant state does not need to send an IPha/IPra update to the RM for every movement between RAs. Instead, the Dormant MN updates its location only when it moves between LAs. When the Dormant MN is called, the LM performs Paging and wakes it up. This policy enables MN battery conservation by reducing MN's frequent location update to the mobility control nodes in the network (R3), and it also reduces wireless resource consumption (R6).

Finally, we separated mobility control and packet forwarding functions (policy D). By the separation, a network operator can independently install mobility control or packet forwarding functions depending on such situations as traffic pattern changes or subscriber increase. By this function, no unnecessary functions are installed so that a cost efficient network can be built (R7).

C. Protocol Overview

As shown in Figure 1, IP^{2}MM transports a packet using the IPha between the MN and the AR, whereas the AR translates the IPha with the corresponding IPra and the routers in the network transport the packet by the IPra. Therefore, the sending and receiving ARs are both required to maintain the information of the MN's IPha/IPra mapping. This information consists of two different caches, one is the Cache for Source Terminal

(CST) that is the IPha/IPra of the source MN, and the other is the Cache for Destination Terminal (CDT) that is the IPha/IPra of the destination MN. These caches are maintained only for Active MNs. To transit the MN's states, IP^{2}MM uses two procedures: Activation and Deactivation. The other two procedures are defined for the Dormant MN: one is Paging that forces the MN to transit to the Acitve state, the other is Location Registration that is used when the MN registers its location.

Figure 2 illustrates the Location Registration procedure in IP^{2}MM. When the Dormant MN#1 detects a change in its location, 1) MN#1 sends Location Registration to AR. Upon receiving it, 2) the AR sends the Location Update to LM, and 3) LM updates the entry of MN#1 in the Location Cache Table. Then, 4) LM sends a Location Update Ack. to AR, and 5) the AR sends a Location Registration Ack. to the MN. In this way, the Location Registration procedure is completed.

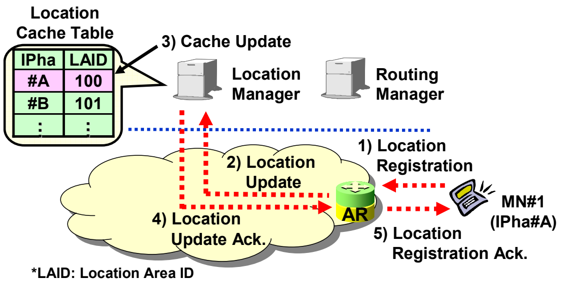


Figure 2: IP^{2}MM Location Registration Procedure

Figure 3 shows the Activation procedure. When the Dormant MN#1 starts communication, 1) it sends Activation to inform its state transition to Active. Receiving it, 2) the AR allocates IPra from the IPra address pool and informs RM of it by sending Activation Notification (AN). Upon receiving this message, 3) RM creates the mapping of IPha and IPra, and 4) RM sends IPra Update (IPU) to the AR for approval notification of IPra allocation for the MN. Receiving IPU, 5) the AR caches the IPha/IPra mapping in the CST. Then, 6) the AR sends an Activation Ack. to the MN#1, and the Activation procedure is completed. In a contrary vein, the Deactivation procedure is conducted when Active MN moves to the Dormant state, and gives the trigger to delete the IPha/IPra mapping of the MN in the AR and the RM [10].

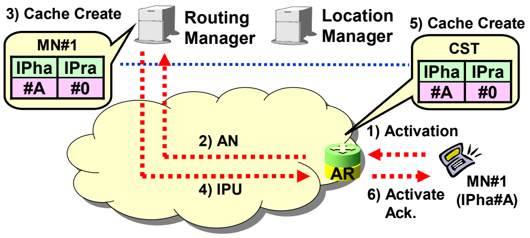


Figure 3: IP^{2}MM Activation Procedure

Figure 4 shows the message flow in the session initiation between Active MNs. 1) When Active MN#1 sends a packet designated to the IPha of MN#2, it is transported to AR1. In failing to find the corresponding IPra in the CDT while trying to translate the address, AR1 buffers the packet, and 2) AR1 sends IPra Inquiry (IPI) to the RM to acquire IPha/IPra mapping of MN#2. Upon receiving IPI and finding the mapping of MN#2, 3) the RM updates cache tables for MN#1 and MN#2 by adding the IPha/IPra mapping entries of MN#2 and MN#1, respectively. Then, informing IPha/IPra mapping of MN#1, 4) the RM sends IPU to the AR2, where the MN#2 connects, to allow AR2 to forward the packet from MN#2 to MN#1 without a query on MN#1 mapping. Receiving IPU, 5) AR2 caches the mapping of MN#1 in CDT, and 6) AR2 sends an IPU Ack. to the RM. Then, to inform the IPha/IPra mapping of MN#2, 7) the RM sends an IPU to AR1. AR1 receives IPU and 8) AR1 caches the mapping in the CDT, then it is ready to translate the packet. After that, 9) AR1 releases the buffered packets, translates to IPra and transmits them to AR2. 10) At AR2, packets are translated again to the IPha and sent to the MN#2.

If the MN#2 is in Dormant state when MN#1 initiates the communication, the RM sends the paging request to the LM, and LM performs Paging procedure by referring the IPha and LAID mapping information. Then, MN#2 conducts the Activation procedure and transits to Active state [10].

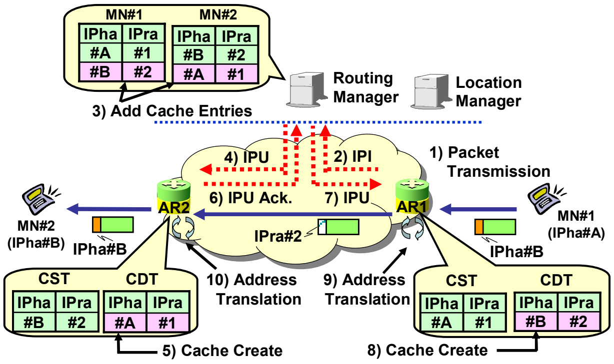


Figure 4: Session Initiation Message Flow in IP^{2}MM

Figure 5 shows the message flow of the handover. The handover is invoked by the Active MN moving between ARs and sending Activation to a new AR. 1) When MN#2 moves to AR3 and detects its handover, 2) MN#2 informs handover by sending Activation to AR3. Upon receiving it, 3) AR3 sends AN to the RM to inform it of the allocated IPra, and 4) the RM updates IPha/IPra mapping of MN#2 in the cache table for MN#1 and MN#2. Then, 5) the RM notifies the cache update of MN#2 to AR1 by sending IPU, and 6) AR1 updates the cache. After that, 7) AR1 sends the IPU Ack. to the RM, and 8) the RM sends IPU to AR3 to order the cache creation for MN#1 and MN#2. Receiving IPU, 9) AR3 creates the IPha/IPra mappings for MN#1 in the CDT and for MN#2 in the CST, and 10) AR3 notifies the handover completion to MN#2 by sending an Activation Ack.

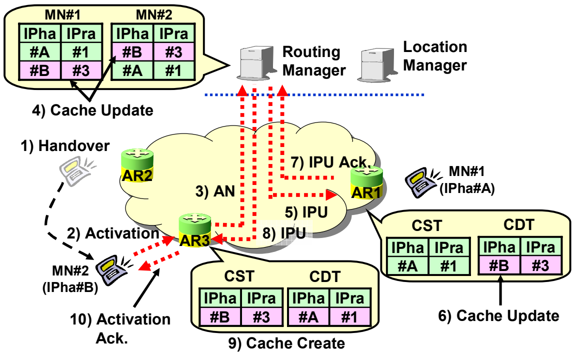


Figure 5: Handover Message Flow in IP^{2}MM

IV. EVALUATION

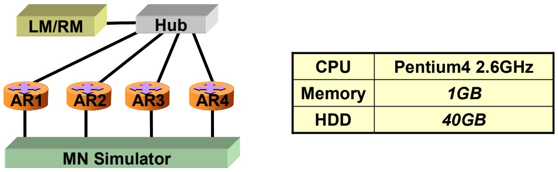
Employing the network-controlled approach, IP^{2}MM has been designed to satisfy the mobile operators' requirements. Due to its design to separate the mobility control nodes and packet transport nodes, signalings between these nodes are required for mobility control functions such as notification of IPha/IPra mapping to the AR. IP^{2}MM is also designed to provide a proxy function to the MN, since the AR needs to perform mobility procedures such as address translation.

To evaluate how the IP^{2}MM designs and their characteristics will influence protocol performance such as handover and confirm that the IP^{2}MM can provide mobility services of higher quality than a conventional protocol, MIPv6, we implemented IP^{2}MM and MIPv6 in our experimental system and measured mobility control and user packet transmission performance. Then, we conducted a comparative evaluation and provided an analysis.

A. Experimental System Configuration

Figure 6 shows the IP^{2}MM experimental system configuration and node specification that are common to the ARs and LM/RM. The MN simulator, which is connected to four ARs over a wired link, is hardware to emulate multiple MNs. The software installed in the ARs and LM/RM is based on NetBSD 1.6.1, and consists of an IP^{2}MM kernel, which is a modified NetBSD kernel, and the user land program that controls the kernel. The LM/RM function is achieved by the user land program, and manages databases such as IPha/IPra mapping and Location Cache Table. On the other hand, the AR function is achieved by both the IP^{2}MM kernel and user land program. The IP^{2}MM kernel has the IPha/IPra mapping database and performs address translation, whereas the user land program controls the IPha/IPra database in the kernel. The IP^{2}MM software is based on the NetBSD, thus the experimental system can be easily configured with PCs of any platform if they support NetBSD.

For the MIPv6 experimental system, we used HA and an IPv6 router instead of RM/LM and AR, respectively. Hardware specifications are the same for MIPv6 entities, and we used a MIPv6 kernel for the NetBSD developed in Kame Project [15] as the software for the HA and MN simulator.



CPU	Pentium4 2.6GHz
Memory	*1GB*
HDD	*40GB*

Figure 6: IP^{2}MM Experimental System Configuration

B. Mobility Control Performance

In this section, we evaluate how the IP^{2}MM design influences mobility control performance compared to MIPv6. We conducted two measurements for the evaluation: single processing time of mobility control, and simultaneous processing capacity. The single processing time was measured for Location Registration and Handover to obtain their basic processing delays. The simultaneous processing capacity was the experiment to give simultaneous mobility control procedures on the network, and to evaluate the processing capacity of IP^{2}MM and MIPv6 when multiple terminals connect to the network. In these experiments, the Location Registration in MIPv6 is defined as a procedure where the MN powers on and registers its HoA/CoA information by sending Binding Update to HA.

Table 1 shows the processing time of Location Registration and Handover in the scenario where only one MN in the network performs single mobility control procedure at a time. In this measurement, we gave no background traffic so that we could observe the pure processing time. We also registered the minimal number of caches at AR/RM or HA to reduce the impact of the implementation dependent cache searching schemes. These results do not include movement detection time, as it is outside the scope of IP^{2}MM. The measurement legs are shown below.

Location Registration: For IP^{2}MM, the section is from 1) to 5) in Figure 2, and for MIPv6, from the MN receiving Router Advertisement, creating CoA, performing Duplicate Address Detection (DAD), to completion CoA registration by MN receiving Binding Acknowledgement.

Handover: For IP^{2}MM, the section is from 2) to 10) in Figure 5. For MIPv6, we cannot measure the exact handover time due to the MIPv6 software problem, so we measured the leg from the MN sending Binding Update to receiving Binding Ack, which excludes the delay by the DAD procedure, as a reference.

	Location Registration	Handover
IP^{2}MM	0.90msec	2.87msec
MIPv6	2000.93msec	N/A (0.25msec*)

*Processing Time without DAD procedure

Table 1: Control Procedure Delay

From the results, we can see that the IP^{2}MM Location Registration processing time is considerably faster than that of

MIPv6. This is because, in MIPv6 Location Registration, both the MN and HA perform DAD in creating CoA and it takes more than 2 seconds, whereas IP^{2}MM does not need DAD since the MN uses a permanently allocated IPha regardless of its location. The same results can be assumed for handover processing time when we consider MIPv6 handover would be more than 1000.25msec, as it requires the DAD procedure which takes more than 1 second.

Next, in Figure 7 we present the measurement results on simultaneous processing capacity. Simultaneous processing in the network would occur when multiple MNs connect to the network and perform the same mobility control procedures, such as Location Registration, at a time. The horizontal axis shows the number of procedures invoked by the MN simulator in a second, and the vertical axis indicates the average delay increment between the processing times of two successive procedures. We define the simultaneous processing capacity as the maximum number of mobility control procedures processed in a second with the same time to conduct single procedure shown in Table 1. For instance, the simultaneous handover capacity in IP^{2}MM Handover is 700, and the network can complete 700 procedures arrived within 1 second by the same time of 2.87msec as shown in Table 1.

As for the measurement conditions, we configured MIPv6 cache search time of the binding cache to be minimal for eliminating its impact, as the MIPv6 software employs an inefficient cache search algorithm. For this reason, the results of MIPv6 procedures, shown in Figure 7, are better than they actually are. In the measurements of IP^{2}MM Location Registration and MIPv6 handover, we cannot gauge the upper limits of these simultaneous processing capacities due to limitations in the MN simulator performance. Therefore, the actual capacities of these procedures were considered to be larger than the experimental results.

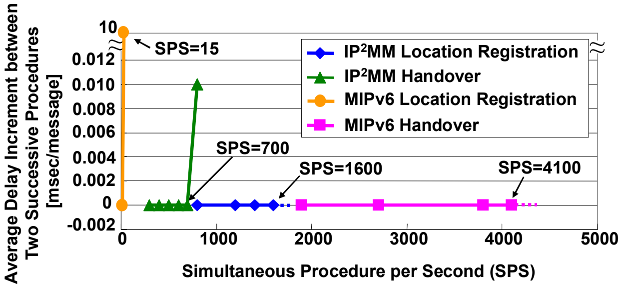


Figure 7: Simultaneous Processing Capacity

From the results, it is observed that the capacity regarding Handover is more than 4100 in MIPv6, while the IP^{2}MM is up to 700. On the other hand, the result of the IP^{2}MM Location Registration shows better capacity than the IP^{2}MM Handover and worse than the MIPv6 Handover. It should be noted that although the performance of Location Registration in MIPv6 is extremely worse, it is not the issue of Mobile IPv6 protocol itself. The reason is that the KAME MIPv6 kernel employs an inefficient cache management algorithm for Location Registration, as it is not assumed to support a huge number of MNs as seen in the mobile operator's network.

Looking at Table 1, it is observed that the simultaneous processing capacity of IP^{2}MM Location Registration, Handover and MIPv6 Handover reflect the results of a single procedure delay. In this analysis, we employ the reference value as MIPv6 Handover results, since the DAD procedure in MIPv6 does not affect the number of simultaneous procedures sent by the MN simulator. We analyzed that the reason is the difference in the number of messages occurred in each procedure: the MIPv6 Handover takes 2 messages, the IP^{2}MM Handover exchanges 10 messages, and the IP^{2}MM Location Registration takes 4 messages including acknowledgements. Therefore, the number of signalings in the procedure is assumed to have an impact on the processing delay, thus it is expected that the delay will be reduced if the number of messages is decreased by the procedure optimization, for example. A reduction in the number of messages is also effective for the propagation delay in the network that is the potential factor leading to the increase of mobility procedure time. The other possible reason is that the mobility control signalings are processed in a different part of the software: the MIPv6 is processed in the OS kernel, while the IP^{2}MM is done in the user land program.

From these evaluations, we can conclude that the IP^{2}MM signaling processing time of Handover and Location Registration is considerably faster than the conventional method, MIPv6. It is also observed that the simultaneous processing capacity in IP^{2}MM Handover is smaller than that of MIPv6, but the performance will improve if the number of messages in the procedure is reduced.

C. User Packet Transmission Performance

In this section, we measured the single user packet transmission delay to evaluate how the address translation method employed in the IP^{2}MM design influences its performance. The measurement was conducted in the scenario where only two MNs exist in the network and exchange a user packet with no signaling and no other user traffic as the background load. To reduce the influences of software dependent cache search algorithms, we registered only required entries in ARs/RM or HA. The packet length of a user packet is 56 bytes including IP header, and we used the average value of 1000 times measurement for the result. In MIPv6, MN processing times such as option adding or encapsulation are not included in the measurement.

Figure 8 shows the measurement results of user packet transmission delay. From the results of MIPv6 (Tunneling) and IP^{2}MM, we can observe the packet processing delay at HA, 56.7 µsec, is relatively larger than that of the AR in IP^{2}MM, which is 0.63 µsec. From this result, although the AR, in IP^{2}MM, translates the addresses between IPha and IPra, we can assume that the load is considerably small. On the other hand, looking at the MIPv6 (Router Optimization) and IP^{2}MM, where the packet travels the same distance and the processing time at MNs are not included in the results of MIPv6, we can observe that the IP^{2}MM can transport a packet faster than MIPv6 (Route

Optimization). This is because the IP^{2}MM does not increase the packet header, whereas the MIPv6 needs Home Address Destination Option or Routing Header Type2.

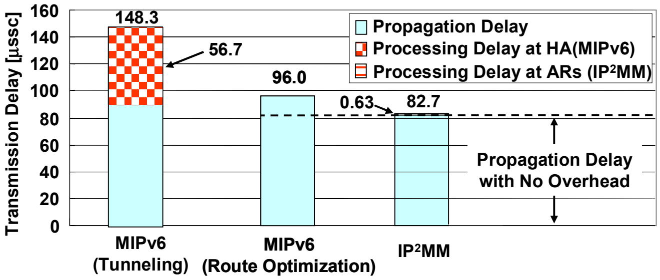


Figure 8: User Packet Transmission Delay

From this evaluation, we can conclude that IP^{2}MM is effective in reducing propagation delay by avoiding the increase of header length, and the address translation method employed in the IP^{2}MM is a lightweight procedure compared to MIPv6 packet routing method.

V. CONCLUSTION

This paper describes an overview of the IP^2 Mobility Management Protocol, which is designed to satisfy mobile operators' requirements for next-generation mobile networks. Achieving the mobile operators' requirements, IP^{2}MM employs the network-controlled approach, which is different from conventional terminal-based ones such as MIPv6, and four design policies: IP address separation, edge intelligence, state-based mobility management, and control/transport separation. Therefore, IP^{2}MM needed to be evaluated in terms of how high-quality mobility services, such as fast handover, can be provided without giving additional load on the network. To evaluate the mobility protocol performance, we implemented IP^{2}MM and MIPv6 in our experimental system, measured mobility control performance and user packet transmission performance, and conducted a comparative evaluation. From these measurements and analysis, we confirmed that the IP^{2}MM provides better performance in mobility control and packet transmission compared to the MIPv6, even though it processed complex tasks to achieve the mobile operators' requirements that the MIPv6 could not satisfy. Therefore, we can conclude that the IP^{2}MM is the feasible and effective mobility protocol for the next-generation networks of mobile operators.

In this analysis, signaling reduction in IP^{2}MM mobility control procedure is identified as a topic for further study to reduce the mobility procedure load and to prevent performance degradation caused by signaling propagation delays in the network. Other topics such as hierarchy in the mobility control nodes to support local mobility efficiently also need to be considered.

REFERENCES

[1] 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network: "UTRA High Speed Downlink Packet Access (Release 4)," 3GPP TR.

[2] ITU-R draft recommendation, "Vision, framework and overall objectives of the future development of IMT-2000 and systems beyond IMT 2000," November 2002.

[3] 3GPP specification, "IP Multimedia Subsystem (IMS) – Stage2 V6.5.0," March 2004.

[4] RFC2002, "IP Mobility Support," October 1996.

[5] RFC3775, "Mobility Support in IPv6," June 2004.

[6] Pat R. Calhoun, et al, "Foreign Agent Assisted Hand-off," Internet Draft draft-calhoun-mobileip-proactive-fa-03.txt, November 2000.

[7] R. Koodli, et al., "Fast Handovers for Mobile IPv6", Internet Draft draft-ietf-mobileip-fast-mipv6-02.txt, Work in Progress, July 2004.

[8] C. Keszei, et al., "Evaluation of the BRAIN Candidate Mobility Management Protocol," Proceeding IST Summit '01, September 2001.

[9] H. Yumiba, et al., "IP-based IMT Network Platform," IEEE Personal Communication Magazine, Vol.8, No.5, pp. 18-23, October 2001.

[10] T. Okagawa, M.Jo, et al., "IP Packet Routing Mechanism Based on Mobility Management," ICIN2003, March 2003.

[11] ITU-T Q.SNFB Ver 2.1 "Service and Network Capabilities Framework of network aspects for systems beyond IMT-2000," April 2004.

[12] 3GPP TR21.902 Ver 1.3.0 "Evolution of 3GPP system (release X)," March 2003.

[13] RFC2462, "IPv6 Stateless Address Autoconfiguration," December 1998.

[14] C. Ferrer, et al., "Overview and capacity of the GPRS (General Packet Radio Service)", IEEE, 1998.

[15] Kame Project, http://www.kame.net