PAPER Special Section on New Technologies and their Applications of the Internet

HIMIPv6: An Efficient IP Mobility Management Protocol for Broadband Wireless Networks

Hyunku JEONG[†a)], ***Student Member***, **Seungryoul MAENG**[†], ***Nonmember***, ***and*** **Youngsu CHAE**[††], ***Member***

SUMMARY With the increasing deployment of mobile devices and the advent of broadband wireless access systems such as WiBro, mWiMAX, and HSDPA, an efficient IP mobility management protocol becomes one of the most important technical issues for the successful deployment of the broadband wireless data networking service. IETF has proposed the Mobile IPv6 as the basic mobility management protocol for IPv6 networks. To enhance the performance of the basic MIPv6, researchers have been actively working on HMIPv6 and FMIPv6 protocols. In this paper, we propose a new mobility management protocol, HIMIPv6 (Highly Integrated MIPv6), which tightly integrates the hierarchical mobility management mechanism of the HMIPv6 and the proactive handover support of the FMIPv6 to enhance the handover performance especially for the cellular networking environment with high frequent handover activities. We have performed extensive simulation study using ns2 and the results show that the proposed HIMIPv6 outperforms FMIPv6 and HMIPv6. There is no packet loss and consequent service interruption caused by IP handover in HIMIP.
key words: *IP mobility management protocol, seamless, handover*

1. Introduction

IP mobility management is one of the most essential functions in all-IP networks where the various wireless access technologies have been converged, such as WLAN, WiBro, mWiMAX, HSDPA, and etc [1]–[3]. To provide convenient and continuous services to users, IP mobility management protocols should be *seamless*, that is, all necessary configurations should be adjusted autonomously and on-going services should not be interrupted or broken.

Mobile IPv6 (MIPv6) [4] is a standard IP mobility management protocol proposed by IETF. MIPv6 installs the binding, an ordered pair of Home Address (HoA) and Care-of Address (CoA), in binding caches of the Home Agent (HA) and Correspondent Nodes' (CNs) and keeps them up-to-date. MIPv6 is not seamless and suffers from the heavy signaling overhead [5], [6]. Hierarchical MIPv6 (HMIP) [7] introduces a hierarchical registration structure into MIPv6 in order to reduce the signaling overhead. Within the domain, a set of Access Routers (ARs), HMIP performs the binding updates to a Mobility Anchor Point (MAP) only. HMIP reduces the signaling overhead, but it is still not seamless [6]. To overcome the service interruption in MIPv6 and HMIP, IETF has introduced Fast Handovers for MIPv6 (FMIP), predicting the next CoA before handover occurs [8]. But, it is hard to predict when or where the mobile node moves [5], [9].

In the literature, there also have been extensive research activities on IP mobility management. SMIP [9] improves on FMIP by predicting the next CoAs based on the mobility patterns, but it is tuned for the pedestrian speed under some limited environments such as an airport. RHO [10] improves the performance of MIPv6 by optimization, but it ignores the L2 handover latency and focuses on the real-time traffic only such as VoIP while the reliable transport protocols such as TCP are more vulnerable by packet loss.

In this paper, we propose a seamless IP mobility management protocol suitable for all-IP networks. For the performance evaluation, we have implemented MIPv6, HMIP, FMIP, and HIMIP on ns2 simulator [11]. We have compared FMIP over HMIP (FHMIP), FHMIP with buffering (FHMIP-b), and HIMIP. We have selected FHMIP and FHMIP-b as our reference protocols. The results show that HIMIPv6 outperforms FHMIP and FHMIP-b. There is no packet loss and consequent service interruption caused by IP handover in HIMIP.

The rest of this paper is organized as follows. Section 2 shows the reference model and the analysis of other protocols in detail. Section 3 describes the procedures of HIMIP for each case of intra-domain handover and inter-domain handover. Section 4 discusses the design issues. Section 5 analyzes the performance of our protocol. In Sect. 6, the results of performance evaluation and their meanings are presented. Section 7 concludes the paper.

2. Background and Related Work

2.1 Reference Model

Figure 1 and Table 1 show our reference model. A Mobile Node (MN) moves through domains while communicating with many CNs or servers which provide various services such as streaming, file downloading, etc. In one location the MN can see a number of APs attached to distinct ARs. Selecting the next AP depends on many kinds of conditions such as moving speed or pattern, interference, the load of networks, etc. So we have assumed that the protocol does not know which AR the MN will move to until it actually does.

Manuscript received February 12, 2009.
Manuscript revised May 21, 2009.
[†]The authors are with the Department of EECS, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea.
[††]The author is with CMAX Wireless Co., Ltd., Korea.
a) E-mail: hkjeong@camars.kaist.ac.kr, Corresponding author.
DOI: 10.1587/transinf.E92.D.1857

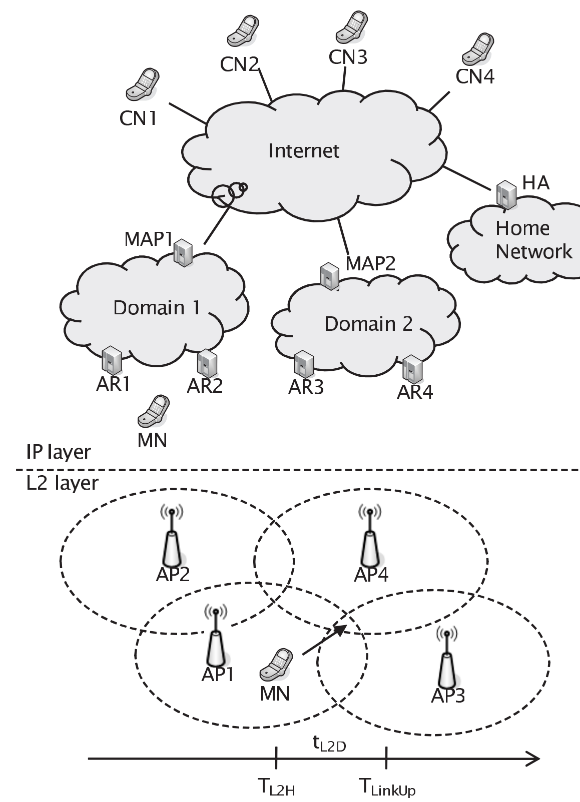

Fig. 1 A reference model.

IP Handover Latency (IPL) is defined as a period from T_{LinkUp} to the moment when the MN receives the first data packet sent directly to the next CoA. *Service Disruption Time* (SDT) is defined as a sum of periods where the services are interrupted because of the L2 handover, packet loss and recovery, side-effects of the protocols, and so forth. SDT is an effective metric for the performances of proactive protocols.

2.2 Analysis of MIPv6

The procedure of MIPv6 consists of following steps: 1) Movement Detection (MD), 2) CoA configuration and Duplicate Address Detection (DAD), 3) Standard Binding Update (SBU) † to the HA followed by Return Routability Test (RRT), and 4) SBU to CNs [4], [12], [13]. Let t_{MD}, t_{DAD}, and t_{RRT} denote the time occupied by MD, DAD, and RRT respectively. Then the SDT and IPL of MIPv6, which are not affected by other factors such as flow/congestion control in TCP, are presented as below.

$$\begin{aligned} &\mathit{IPL(MIPv6)} = t_{\mathrm{MD}} + t_{\mathrm{DAD}} + t_{\mathrm{RRT}} + t_{\mathrm{SBU}} \\ &\quad \text{where } t_{\mathrm{RRT}} = 2t_{\mathrm{CN}} \ (\because \text{HOTI and HOT}) \\ &\mathit{SDT(MIPv6)} = t_{\mathrm{L2D}} + \mathit{IPL(MIPv6)} \end{aligned}$$

There are several problems in MIPv6. First, the MN cannot receive packets from CNs during SDT and those packets are lost. Next, every time the MN changes its IP point-of-attachment, MIPv6 has to perform RRT and SBU with every CNs. Finally, the execution times of MD and DAD †† are so long that the MN can start another handover while the previous one is not complete.

Table 1 Summary of parameters and symbols.

Literal	Meaning
t_{AR}	Packet delivery latency between MN to AR
t_{AM}	AR to MAP
t_{MC}	MAP to CN, MAP to HA, or HA to CN
t_{MAP}	MN to MAP ($= t_{\mathrm{AR}} + t_{\mathrm{AM}}$)
t_{CN}	MN to CN ($= t_{\mathrm{AR}} + t_{\mathrm{AM}} + t_{\mathrm{MC}}$)
t_{AA}, t_{MM}	AR to AR and MAP to MAP
T_{L2H}, T_{IPH}	L2 and IP handover start time
T_{LinkUp}	L2 handover end time
t_{L2D}	L2 handover latency ($= T_{\mathrm{LinkUp}} - T_{\mathrm{L2H}}$)
λ	packet arrival rate
Prefix p-	present or previous ex.) pAR, pMAP, ...
Prefix n-	next or new ex.) nAR, nMAP, ...
ARij	ARj which belongs to MAPi
RCoAi	Regional CoA derived from MAPi
LCoAij	On-link CoA derived from ARij
B(a1, a2)	Binding of two addresses a1 and a2
SS	Slow Start
CA	Congestion Avoidance
CA-SS1	CA followed by SS type 1 (explained in 6.2)
CA-SS2	CA followed by SS type 2 (explained in 6.2)

2.3 Analysis of HMIP

HMIP introduces two kinds of CoAs: an on-Link CoA (LCoA) and a Regional CoA (RCoA). An LCoA is used within a domain, and to outside only a RCoA is exposed. The MN performs only LBU if the RCoA is not changed. When the RCoA is changed, it does SBU to the HA and CNs after LBU. The SDT and IPL of HMIP in intra-domain handover are as follows.

$$\begin{aligned} \mathit{IPL(HMIP\text{-}intra)} &= t_{\mathrm{MD}} + t_{\mathrm{DAD}} + 2t_{\mathrm{MAP}} \\ \mathit{SDT(HMIP\text{-}intra)} &= t_{\mathrm{L2D}} + \mathit{IPL(HMIP\text{-}intra)} \end{aligned}$$

Compared with *SDT(MIPv6)*, t_{RRT} is eliminated and the time required by the binding update is reduced. But *SDT(HMIP-intra)* is still long because it contains t_{MD} and t_{DAD}. The SDT and IPL of HMIP in inter-domain handover are as follows.

$$\begin{aligned} \mathit{IPL(HMIP\text{-}inter)} &= 2t_{\mathrm{MAP}} + \mathit{IPL(MIPv6)} \\ \mathit{SDT(HMIP\text{-}inter)} &= t_{\mathrm{L2D}} + \mathit{IPL(HMIP\text{-}inter)} \end{aligned}$$

2.4 Analysis of FMIP over HMIP (FHMIP)

In FMIP the MN selects one nAR among all possible candidates and creates a nLCoA from the prefix information of the nAR. Then it sends B(pLCoA, nLCoA) to the pAR,

† We use SBU to refer a binding update to the HA or CNs. We call a binding update to the MAP as a Local Binding Update (LBU).

†† According to [10] and [13], the values of t_{MD} and t_{DAD} are 2.2∼2.8 s and 1∼2 s respectively.

which is called a Fast Binding Update (FBU). The nAR tests the uniqueness of the nLCoA by its prior knowledge and sends back the result to the pAR. After that, the pAR intercepts and tunnels packets to the nLCoA. We will call this tunneling action *pAR-tunneling* for later discussion.

Let t_{TEST} be the latency of nCoA validity test. The SDT and IPL of FHMIP-intra are expressed like below.

$$\begin{aligned} &\mathit{IPL(FHMIP\text{-}intra)} = t_{\mathrm{MD}} + 2t_{\mathrm{MAP}} \\ &\mathit{SDT(FHMIP\text{-}intra)} = t_{\mathrm{WAIT}} + t_{\mathrm{L2D}} + 2t_{\mathrm{AR}} (\because \text{UNA}) \\ &\text{where } t_{\mathrm{WAIT}} = (T_{\mathrm{L2H}} - T_{\mathrm{IPH}}) - t_{\mathrm{TEST}} \text{ and} \\ &t_{\mathrm{TEST}} = 2(t_{\mathrm{AR}} + t_{\mathrm{AA}}) \ (\because \text{FBU/FBAck and HI/HAck}) \end{aligned}$$

In FHMIP (and other proactive protocols) the MN can receive data packets even before the IP handover procedure ends. So there is no harm to omit IPL. t_{WAIT} is a waiting time during which the MN cannot receive packets though it is still connected to the pAR. t_{MD} and t_{DAD} are eliminated from *SDT(FHMIP-intra)*. So, if t_{WAIT} becomes zero, then FHMIP-intra can achieve almost optimal performance. The SDT of FHMIP-inter is similar too, but in this case t_{TEST} becomes longer because pAR and nAR reside different domains.

$$\begin{aligned} &\mathit{SDT(FHMIP\text{-}inter)} = t_{\mathrm{WAIT}} + t_{\mathrm{L2D}} + 2t_{\mathrm{AR}} \\ &\text{where } t_{\mathrm{WAIT}} = (T_{\mathrm{L2H}} - T_{\mathrm{IPH}}) - t_{\mathrm{TEST}} \text{ and} \\ &t_{\mathrm{TEST}} = 2(t_{\mathrm{AR}} + t_{\mathrm{AA}}) = 2(t_{\mathrm{MAP}} + t_{\mathrm{MM}} + t_{\mathrm{AM}}) \end{aligned}$$

FHMIP suffers from two kinds of prediction problems. 1) It is hard to decide or predict T_{IPH}, T_{L2H}, and t_{TEST} accurately. So if FHMIP starts the handover earlier, then t_{WAIT} will become longer. If t_{WAIT} is negative, then FHMIP operates in a reactive mode and packet loss is likely to occur. 2) At T_{IPH} the MN selects a nAR, but the MN can move to an unexpected AR or even does not move.

Another source of problems is pAR-tunneling. 1) All packets tunneled by pAR have additional delay as t_{AA}. This *tunneling delay* becomes longer in the inter-domain handover. 2) After the MN performs LBU or SBU, the packets tunneled from the pAR and forwarded directly to the MN meet at the incoming link of the nAR at a rate of 2λ. In condition of high λ and many MNs, the incoming link should be congested. 3) Packet interlacing occurs. The protocols, which guarantee reliable transfer like TCP, can handle that case. But, packet interlacing may degrade the performance. It will be shown in Sect. 6.

3. Protocol Description

The purpose of HIMIP is to provide seamless IP mobility management. These are the design considerations of HIMIP: 1) We have chosen the proactive approach to achieve seamlessness. The reactive protocols should be supported by or cross-optimized with all L2 layer protocols to compensate for the packet loss incurred by the L2 handovers. This cancels out the overhead of the proactive protocols. 2) We have considered inter-domain handovers as well as intra-domain handovers. Most micro-mobility protocols have dealt with the intra-domain handovers only. Even if not happened frequently, the inter-domain handovers occur and impact the performance. The detail procedures for both cases are explained in the following subsections. 3) We have assumed that L2 layer does hard (break-before-make) handovers and reports to our protocol the information about access points and their status for our protocol to operate on any wireless access technologies.

We have defined three ICMP messages and two Mobile IP messages.

HR (Handover Ready) The ICMP message sent by the master MAP to ARs or neighbor MAPs to initiate a handover. It has three flags: B flag: buffer packets, T flag: create tunnel, and A flag: require ack.

HRAck (HR Ack.) The ICMP message sent in response to the HR message with A flag.

HC (Handover Complete) The ICMP message used to inform the master MAP that the handover is complete.

FBU+ A set of extended bindings expressed as B(RCoA*i*, [LCoA*ij*, LCoA*ik*, ...]).

D-FBU+ (Delegated FBU+) A subset of the FBU+ message.

3.1 Intra-Domain Handover

Figure 2 shows the procedure of our protocol for an intra-domain handover. The logical relationship expresses that all ARs belong to MAP1 and they do not share any APs. The MN is about to move from AR11 to AR12 or AR13.

When the MN detects new APs, it queries the logical relationship and gets [AP-ID, AR-Info, MAP-info] tuples by exchanging the RtSolPr and PrRtAdv messages with the pAR just like FMIP. Based on these tuples and the status reports from L2 layer, HIMIP selects multiple nARs and makes nLCoAs derived from each nAR. Then it sends the pMAP, the present serving MAP, the FBU+ message at T_{IPH}. In our example, the FBU+ message looks like B(RCoA1, [LCoA12, LCoA13]).

According to the FBU+ message, the pMAP updates its binding cache and sends the pAR the HR message with B flag which makes the pAR buffer packets before forwarding them to the MN. This buffering is served as a precaution against packet loss caused by false-alarm or ping-pong. After that the pMAP starts to simulcast to pLCoA and all of nLCoAs. For all packets simulcasted it sets the S flag and adds a sequence number. HIMIP requires one bit from the IPv6 header for S flag and a new IP header option for sequence number additionally.

Followed by the HR message the pMAP sends the nARs the HI messages to initiate validity tests for each nLCoA (AR*ij* checks LCoA*ij*). Each nAR reply the result as the HAck messages and it start to buffer the simulcasted packets. If the test result is negative, pMAP creates a tunnel

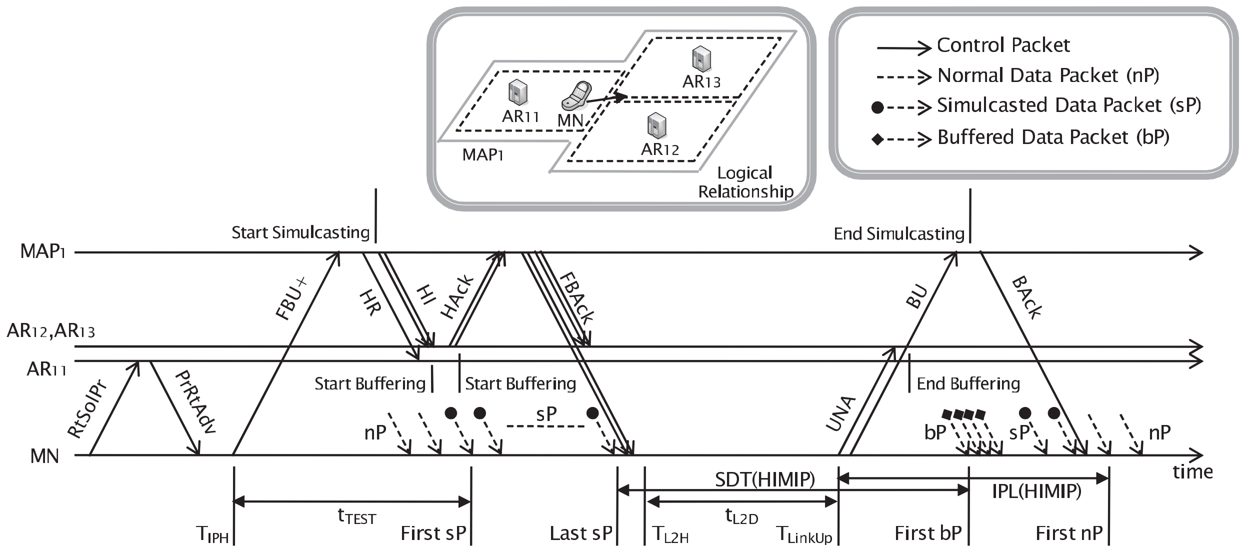

Fig. 2 Intra-domain handover procedure of HIMIP.

with the nAR. For each test result the pMAP notifies the MN with the FBAck message. The FBAck message is sent to pLCoA and nLCoA.

It should be noted that the pMAP starts the simulcasting right after receiving the FBU+ message. There are two reasons. First, the validity test takes some time and if the pMAP waits then the MN may experience the service interruption. The second reason is explained in Sect. 4.2. Therefore the pMAP makes copies for each packet and then sends one to the pLCoA and saves the others for the nLCoAs into buffers. The buffered packets for LCoAij are flushed when the HAck message comes from ARij.

When the MN is connected to a new AP, the MN learns from the tuples that which nAR it has moved to and that which nLCoA it should use. Then the MN informs the nAR of its presence via the UNA message which causes nAR to stop buffering and forward the buffered packets. After that the MN sends the pMAP the BU message. The pMAP finishes the handover by stopping simulcasting, clearing the binding cache, and sending the BAck message to the MN.

3.2 Inter-Domain Handover

Figure 3 represents the procedure of an inter-domain handover. As shown by the logical relationship, AR_{11} is connected to MAP_1 and AR_{22} and AR_{23} are connected to MAP_2. The MN is about to move from AR_{11} to AR_{22} or AR_{23}. Because the procedure of inter-domain handover is similar to the previous one, only the difference is explained.

The MN starts the handover by selecting the multiple nARs. For each nAR the MN looks up the tuples to find the nMAP where the nAR belongs. Then the MN creates nRCoA for the nMAP. After creating all nRCoAs and nLCoAs, the MN composes the FBU+ message whose content is {B(RCoAi, [LCoAij, ...]), B(RCoAr, [LCoArs, ...]), ...} according to the relationship tuples.

The pMAP investigates the FBU+ message and finds RCoAi derived from MAPi which is not itself. For each MAPi found, the pMAP sends the HR message with the T and A flags. The context transfer occurs at this stage if exists. After MAPi acknowledges the HR message with the HRAck message, a tunnel between the pMAP and MAPi is created. Then the pMAP sends MAPi the D-FBU+ message whose content is B(RCoAi, [LCoAij, ...]). Because the D-FBU+ message contains the extended binding, MAPi behaves just like the pMAP does in the intra-domain handover except sending the HR message.

After the L2 connection is recovered, the MN sends the UNA to the nAR and the BU messages to nMAP. By the BU message the nMAP resets the entry in the binding cache and stops simulcasting. Then it transfers the BAck message to the MN and the HC messages to the pMAP to tell that the MN has moved to itself.

Finally, finishing LBU, the MN performs SBU. After that, for some period, the nMAP receives packets sent to pRCoA and nRCoA. HIMIP uses the n-buffer scheme to resolve the packet interlacing which can be occurred at nMAP.

3.3 Putting It All Together

So far we have described the handover procedures separately, but in common cases these two types of handovers will happen together. HIMIP handles them without any difference. Let's assume the MN is about to move from AR_{11} to AR_{12}, AR_{23}, or AR_{24}. As a result, the binding caches of MAP_1 and MAP_2 become B($RCoA_1$, [$LCoA_{11}$, $LCoA_{12}$, MAP_2]) and B($RCoA_2$, [$LCoA_{23}$, $LCoA_{24}$]) respectively.

According to the binding caches, MAP_1 simulcasts packets to $LCoA_{11}$, $LCoA_{12}$, and MAP_2, and MAP_2 simulcasts to $LCoA_{23}$ and $LCoA_{24}$. MAP_1 lets all ARs have same simulcasted packets in their buffers, which is managed in the soft state mode.

If the MN moves to AR_{23}, it sends the UNA message

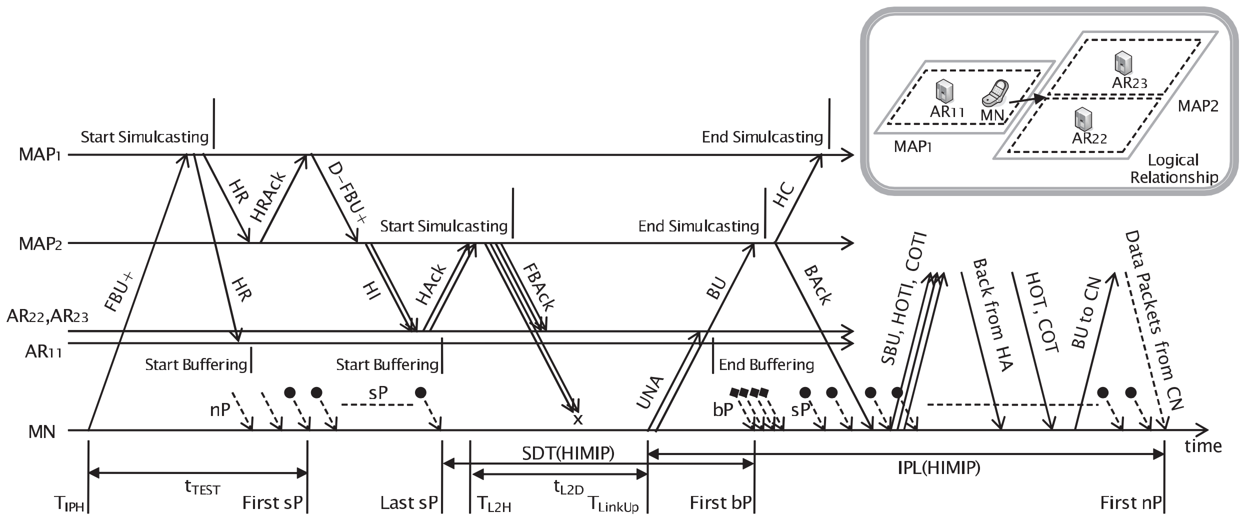

Fig. 3 Inter-domain handover procedure of HIMIP.

including the last sequence number which it has seen so far. In HIMIP the MN can receives the simulcasted packets from the pAR just before the handover. So without the sequence number, packet duplication will occur. AR_{23} filters out the packets whose sequence number are less than or equal to the last sequence number seen and transfers the rest to the MN.

The HC message makes the pMAP prune the simulcasting branches to unnecessary paths. After the HC message, the binding cache of MAP_1 becomes B($RCoA_1$, MAP_2) and MAP_2 of B($RCoA_2$, $LCoA_{23}$).

4. Design Issues

In this section, the design issues and the differences between HIMIP and other protocols such as FHMIP or SMIP are discussed.

4.1 Selecting Multiple nARs

Unlike FHMIP which chooses only one nAR, HIMIP selects the candidate nARs as many as it needs. So the probability of false-alarm decreases dramatically. Unlike SMIP which has severe drawbacks on predicting nARs, HIMIP does not need to predict where to move. Nevertheless, it can rapidly switch among several pre-selected nARs by just sending the UNA message (without LBU) to continue its session with the packets saved at each nAR. This enables the MN to move at high speed.

To prevent the problems of pAR-tunneling which FHMIP exploits, HIMIP uses pMAP-simulcasting which has duplication overhead inherently. This overhead, however, will not be so serious practically: 1) APs are to be deployed to provide the maximum coverage with the minimal numbers. 2) Several APs are connected to one AR. 3) The MN chooses only a fraction of the APs which meet certain threshold. 4) It can use the connection history.

4.2 Implicit Acknowledgement

When the MN gets the simulcasted packets, it can be sure that the pMAP receives the FBU+ message, although the uniqueness of the nLCoAs can be known after receiving the FBAck messages. We call it *implicit acknowledgement*. If the MN sees the implicit acks, it can move freely without worry about packet loss because all simulcast packets are buffered at nARs.

There is another advantage in implicit acknowledgement. Because the pMAP starts to simulcast right after it receives the FBU+ message, the MN expects that it will receive the implicit ack after $2t_{\mathrm{MAP}} + \frac{1}{2\lambda} \simeq 2t_{\mathrm{MAP}}$ where λ is high. The value of t_{MAP} can be measured by periodic LBU. So, if good L2 triggers are provided, T_{IPH} can be estimated with high accuracy.

4.3 Managing Simulcast Tree

The simulcasting is performed in a tree structure. The initial simulcast tree is constructed from the first FBU+ message. After that, the MN can add or remove leafs by sending additional FBU+ messages. Let's assume AR_{11} and AR_{22} are leafs. If the MN finds AR_{23} after moving to AR_{22}, it sends MAP_2 the FBU+ message, B($RCoA_2$, $LCoA_{23}$). MAP_2 already has the binding for $RCoA_2$, so it simply appends $LCoA_{23}$ to the binding. If the MN finds AR_{34}, it sends the root, MAP_1, the FBU+ message, B($RCoA_3$, $LCoA_{34}$). In this way, AR_{23} and AR_{34} are added. Removing is also simple. If the MN wants to remove AR_{11} from the tree, it sends the FBU+ messages where the lifetime of the $LCoA_{11}$ is set to zero.

Although the MN selects multiple nARs to avoid the problems such as false-alarm, still there is a chance of erroneous movement due to various movement patterns and speed, irregularity in L2 layer, and so forth. Managing simulcast tree enables HIMIP to handle those situations more securely and to use the network resource more effectively.

4.4 Preventing Interlacing at the MN

Although simulcasting is used, the packets tunneled from the pMAP and directly sent to nRCoA after SBU are interlaced at the MN in case of inter-domain handovers. HIMIP uses the n-buffer scheme to deal with this problem. After the nMAP sends the HRAck message it creates the n-buffer storing n-packets whose destination address is nRCoA. It flushes the n-buffer after no more p-packets, whose destination address is pRCoA, arrive from the pMAP.

HIMIP calculates the exponential weighted moving average (EWMA) for the inter-arrival time of p-packets as TCP estimates RTT, because if the waiting time is too short the packets are interlaced, and if it is too long, it wastes time.

4.5 Preventing Interlacing at the CN

The interlacing at the CN can happen by reverse tunneling, and the reverse tunneling occurs to bypass the ingress filtering at ARs or MAPs under inter-domain handovers.

Let's assume that P(s, d, data) is a packet whose elements represent sources address, destination address, and the content in order. Until SBU ends, the packet sent from the MN looks like P_{out} = P(nLCoA, pMAP, P_{in}) where P_{in} = P(pRCoA, CN, data). The source address of P_{out} has to be the nLCoA otherwise P_{out} will be filtered out at the nAR. The destination address of P_{out} must be the pMAP, if it is nMAP, the inner packet P_{in} will be dropped at the nMAP. Moreover the source address of P_{in} cannot be nRCoA, because the CN does not know nRCoA yet. This reverse tunneling causes interlacing.

To prevent the problem, HIMIP uses the trust relationships between the pMAP and nMAPs. From the trust relationships between the MN and pMAP, and the pMAP and nMAP, the nMAP can trust the MN. In HIMIP P_{out} looks like P(nLCoA, nMAP, P_{in}). Even though the source address of P_{in} is not nRCoA, the nMAP forwards P_{in} to the CN.

5. Performance Analysis of HIMIP

The meaning of t_{TEST} is the latency of nLCoA validity test. But, in high λ, it will serve as the latency of the implicit ack. Here are the SDT and t_{TEST} of HIMIP-intra.

$$\mathit{SDT(HIMIP\text{-}intra)} = t_{\mathrm{L2D}} + 2t_{\mathrm{AR}} + \frac{1}{2\lambda} \simeq t_{\mathrm{L2D}} + 2t_{\mathrm{AR}}$$
$$t_{\mathrm{TEST}} = 2t_{\mathrm{MAP}} + \min\{\frac{1}{2\lambda}, 2t_{\mathrm{AM}}\} \simeq 2t_{\mathrm{MAP}}$$

Compared with *SDT(FHMIP-intra)*, t_{WAIT} is eliminated because the MN can receive packets just before the handover. Also t_{TEST} does not affect *SDT(HIMIP-intra)* if $t_{\mathrm{TEST}} < T_{\mathrm{L2H}} - T_{\mathrm{IPH}}$. So HIMIP removes all redundant delays from SDT. The SDT and t_{TEST} of HIMIP-inter is as follows.

$$\mathit{SDT(HIMIP\text{-}inter)} = t_{\mathrm{L2D}} + 2t_{\mathrm{AR}} + \frac{1}{2\lambda} \simeq t_{\mathrm{L2D}} + 2t_{\mathrm{AR}}$$
$$t_{\mathrm{TEST}} = 2t_{\mathrm{MAP}} + \min\{\frac{1}{2\lambda}, 2t_{\mathrm{AM}} + 4t_{\mathrm{MM}}\} \simeq 2t_{\mathrm{MAP}}$$

One of the advantages of HIMIP is that the SDT and t_{TEST} for the intra/inter-domain handover are same. So the MN can handle them same way. There is, however, the case resulting in service interruption:

$$t_{\mathrm{TEST}} \geq T_{\mathrm{L2H}} - T_{\mathrm{IPH}} \text{ and}$$
$$T_{\mathrm{L2H}} - T_{\mathrm{IPH}} + t_{\mathrm{L2D}} < 2(t_{\mathrm{MAP}} + t_{\mathrm{AM}} + 2t_{\mathrm{MM}})$$

It means that the MN does not receive any acks and that the handover is done so quickly that the nAR does not be notified the MN yet. Even if the MN operates in optimistic mode, it only receives the RtAdv-NAAck message and it should lose some packets. This is solved if good L2 triggers are provided. Because t_{TEST} is measurable, T_{IPH} can be estimated appropriately.

6. Performance Evaluation

To evaluate the performances, we have implemented FHMIP, FHMIP-b, and HIMIP on the ns2 [11] from scratch based on MobiWan [14], the extension supporting MIPv6.

6.1 Simulation Model

Figure 4 shows an example of our simulation model. Total simulation time is 80 seconds. Starting at AR_{11}, the MN moves from $\mathrm{AR}x_i$ to $\mathrm{AR}x_{(i+1)}$ between 10 s to 70 s. The handovers occur evenly. In this example, 2 intra-domain handovers happen at 25 s and 55 s, and 1 inter-domain handover at 40 s. T_{IPH} is 0.2 s earlier than T_{L2H}. t_{L2D} is 0.15 s. The number of ARs and MAPs and delays of each link vary according to the simulation scenarios. The traffic model is FTP over TCP-Reno and the flow starts at 5 s and terminates at 75 s. The packet size is 1000 B and the window size are 30. There is no packet loss or error in wireless link.

6.2 TCP Behaviors

Figure 5 shows the case of burst packet loss which is typical

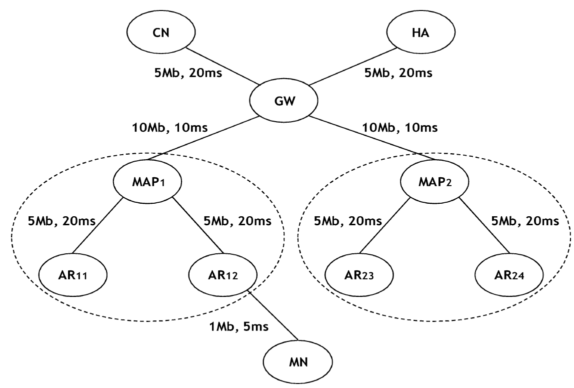

Fig. 4 Simulation model.

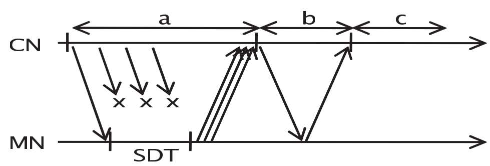

Fig. 5 Three sections where timeout interval can fall.

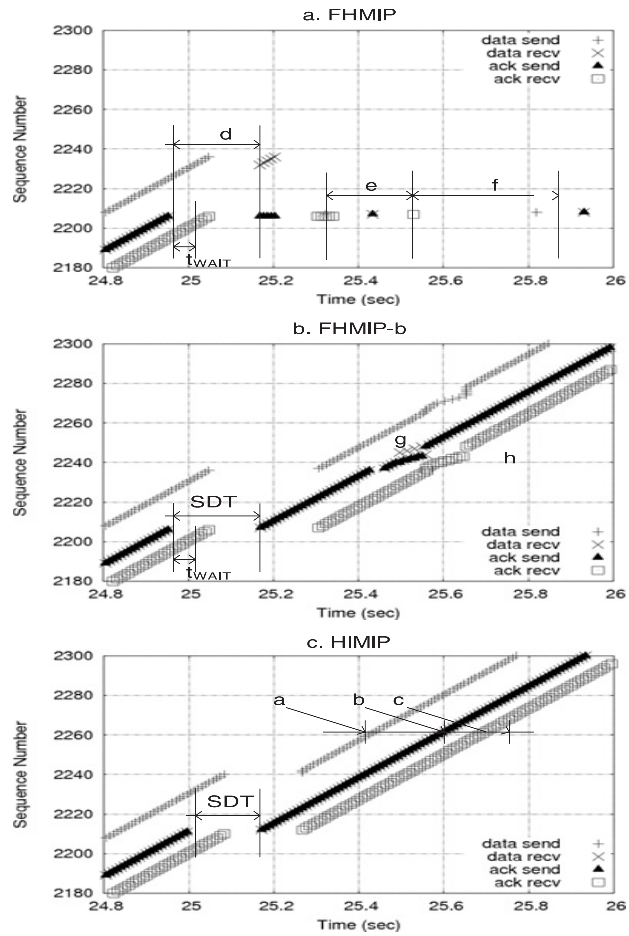

Fig. 6 TCP sequence numbers for intra-domain handovers.

in FHMIP. If timeout occurs in *a*, TCP does *SS*. For timeout in period *b*, it handles the case as *CA-SS1*. TCP sends the r-pkt against 3dup-acks but timeout occurs before the a-pkt. After all, it does *SS*. *CA-SS2* happens in *c*. Even though TCP receives the a-pkt, it has to wait for other lost packets. In general, the SDT of *CA* is the shortest, and *SS*, *CA-SS1*, and *CA-SS2* in ascending order.

$$a = \text{RTT} + \text{SDT} = 2t_{\text{CN}} + t_{\text{WAIT}} + t_{\text{L2D}}$$
$$b = \text{RTT} + \text{tunneling} = 2(t_{\text{CN}} + t_{\text{AA}})$$

6.3 Impact of the Intra-Domain Handover

The x and y axes of Fig. 6 represent time in second and sequence numbers of packets. Let's see Fig. 6.c first. The marks *a*, *b* and *c* represent the sequence numbers of the data and the ack packets respectively. If there is packet loss, these can be different like *g* in Fig. 6.b.

Figure 6.a depicts the result of FHMIP. The period (mark *d*) is about 0.2 seconds which is a sum of t_{WAIT} and t_{L2D} (= 0.05 s + 0.15 s). About 25 packets are dropped during handover. After connected to nAR, the MN receives 5 data packets, but it incurs 3dup-acks. So the CN retransmits the missing data packet for recovery (named r-pkt, for convenience) and goes into the fast recovery mode. After about 0.2 seconds, mark *e*, the CN receives the ack for the r-pkt (named a-pkt), but it does not send any data packets, because the congestion window is full even though it shrinks by half. After about 0.3 seconds (mark *f*), the TCP timeout occurs and the CN does slow start. (This is an example of *CA-SS2*.) So, the SDT of FHMIP-intra is about 1.83 seconds even if it is not shown in the graph.

FHMIP-b, shown in Fig 6.b, eliminates the packet loss and achieves relatively good SDT (= 0.22 s) except t_{WAIT}. The marks *g* and *h* show the packet interlacing occurs. TCP can reorder the packets, so there is no effect of interlacing. But, it can be a source of performance degradation. We will explain it in Sect. 6.6.

The SDT of HIMIP-intra (also HIMIP-inter) is about 0.18 seconds. HIMIP achieves the optimal SDT and there are no packet loss or interlacing. As explained in Sect. 5, the SDTs of *HIMIP-inter* and *HIMIP-intra* are the same.

6.4 Impact of the Inter-Domain Handover

The impact of the inter-domain handover is shown in Fig. 7. In FHMIP-inter, total 18 packets are lost (mark *a*). During mark *b*, the CN does recovery. But different from FHMIP-intra, the timeout for the r-pkt occurs and then the a-pkt arrives after a while. The reason is that the trip time of the a-pkt increases because of reverse tunneling. (This is an example of *CA-SS1*.)

SDT(FHMIP-inter) is about 1.55 seconds. Compared with FHMIP-intra, it is reduced. There are two reasons. 1) In the inter-domain handover, the pAR and nAR are placed in different domains, so t_{TEST} takes more time. As a result, t_{WAIT} decreases. 2) In case of massive packet loss, the sooner TCP goes into slow start, the better result comes. The fast recovery mechanism is nothing but an delay here. These, however, do not mean that the performance of inter-domain handover is better than intra-domain handover. These are only unexpected benefits. If t_{TEST} is more longer, FHMIP will operate in a reactive mode, and there will be a service interruption in the order of seconds.

As the reason explained above, t_{WAIT} in FHMIP-b is also almost zero. The space marked as *c* is caused by increment of round trip time due to tunneling. Also the SBU occurs around the mark *c*, so there is no interlacing.

6.5 Impact of t_{L2D}, Handover Type and Interval

Figure 8 shows TCP goodputs according to the combination of *handover type*, *the number of handovers*, and t_{L2D}. For example, *Intra10* on x-axis means that the MN performs 10

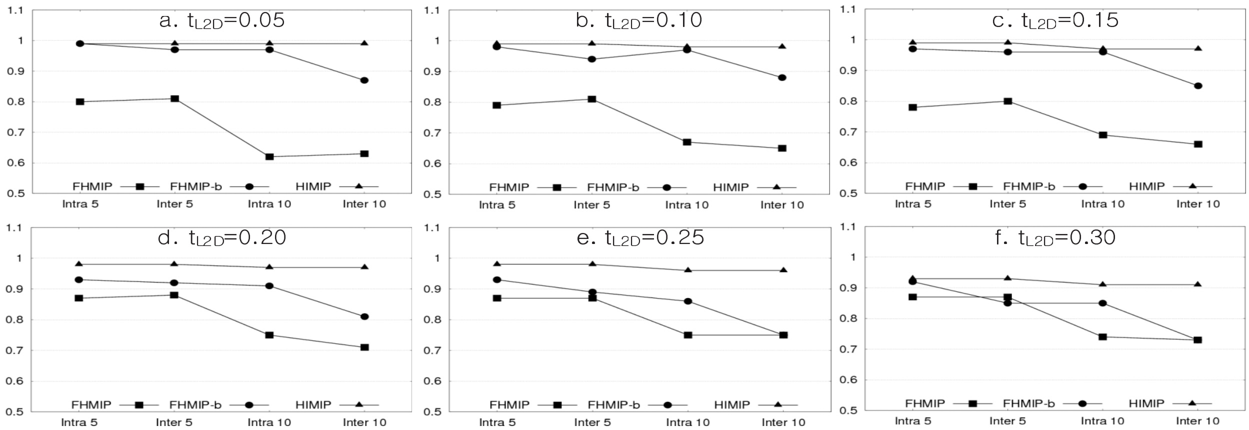

Fig. 8 TCP goodputs according to the impact of various parameters.

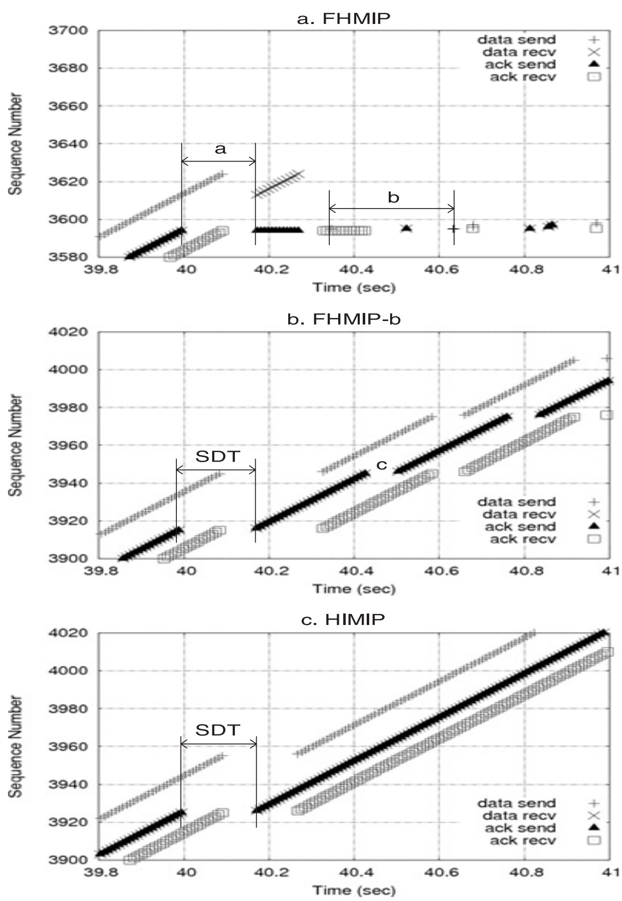

Fig. 7 TCP sequence numbers for inter-domain handovers.

Table 2 The # of events in FHMIP varying t_{L2D} in second.

FHMIP-intra5	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	0	5	5	5
CA-SS1	1	1	1	0	0	0
CA-SS2	4	4	4	0	0	0
FHMIP-inter5	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	0	5	5	5
CA-SS1	3	3	4	0	0	0
CA-SS2	2	2	1	0	0	0
FHMIP-intra10	0.05	0.10	0.15	0.20	0.25	0.30
SS	1	5	7	10	10	10
CA-SS1	7	4	1	0	0	0
CA-SS2	2	1	2	0	0	0
FHMIP-inter10	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	4	6	8	10	10
CA-SS1	10	4	1	1	0	0
CA-SS2	0	2	3	1	0	0

Table 3 The # of events in FHMIP-b varying t_{L2D} in second.

FHMIP-b-intra5	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	0	5	5	5
CA once	0	0	1	0	0	0
FHMIP-b-inter5	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	0	2	5	5
CA once	3	1	0	2	1	4
CA twice	0	2	1	1	1	1
FHMIP-b-intra10	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	0	5	10	10
CA once	1	0	1	0	0	2
FHMIP-b-inter10	0.05	0.10	0.15	0.20	0.25	0.30
SS	0	0	2	4	10	10
CA once	0	3	4	3	4	4
CA twice	5	3	1	3	1	1

intra-domain handovers. Each result is normalized to the one where the MN does not performs any handovers. Table 2 and 3 show how the results come.

FHMIP, 0.05 s~0.15 s: Bulk packet loss happens whenever the MN does handover. If handover interval is long (here, 5 handovers), the congestion window can grow to incur queuing delay at nAR. In consequence, timeout interval becomes longer to cause *CA-SS2*. In Inter5, however, the path of reverse tunneling becomes longer than the one of Intra5. Hence *CA-SS1* comes about more frequently. If handover interval is short (10 handovers), the congestion window cannot grow much. Thus timeout interval becomes short and the TCP events get sensitive to t_{L2D}.

As t_{L2D} is getting longer, the goodputs of 5 handovers decrease and the ones of 10 handovers increase. The reason for 5 handovers case is that the number of lost packet also increases and the reason for 10 handover case is the number

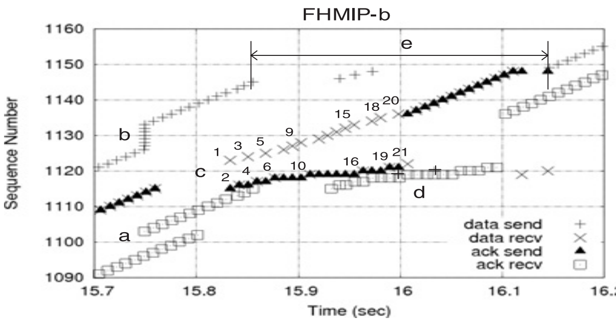

Fig. 9 Impact of packet interlacing.

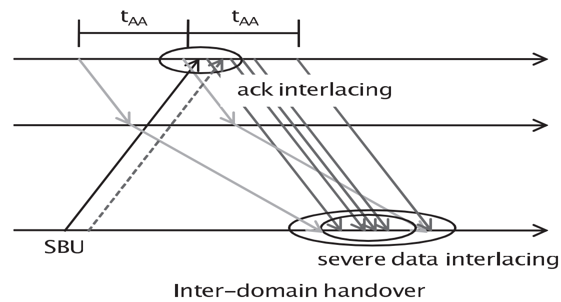

Fig. 10 Packet interlacing in detail.

of *CA-SS* reduced.

FHMIP, 0.20 s~0.30 s: Because of long t_{L2D} *CA-SS1* and *CA-SS2* disappear. As the side-effect of fast recovery vanishes, the goodputs are getting better.

FHMIP-b, 0.05 s~0.15 s: In FHMIP-b, the main reason that depreciates goodputs is not packet loss but interlacing. Because timeout does not occur, it is still better than FHMIP (See 6.6 and 6.7).

FHMIP-b, 0.20 s~0.30 s: Due to long t_{L2D}, *SS* takes place along with *CA*. This is the reason why the goodput of FHMIP-b at 0.30 s is worse than the one of FHMIP.

HIMIP, 0.05 s~0.30 s: The degradation which happens at 0.30 s comes from long L2 handover latency. In this case, there are not much things to do in IP layer. To get more better result, the helps of other layers are required.

6.6 Impact of Packet Interlacing

Figure 9 shows how packet interlacing depreciates the performance. Although not presented in the figure, the handover took place at 15 s and there was no packet loss. The SDT was short enough not to cause service interruption.

About at 15.75 s, marked as *a*, the CN encounters interlaced ack packets caused by reverse tunneling and it transmits several (about 7) data packets as a response at once (mark *b*). About at 15.83 s, marked as *c*, the MN gets 21 out-of-order packets. For every interlaced packet, the MN acknowledges, but only three acks are effective for the CN and the others are just served as 3dup-acks. The mark *d* shows fast recovery mechanism is activated twice and the congestion window shrinks to one quarter. So as mark *e*, the service interruption happens. It lasts about 0.29 seconds.

This result shows that preventing interlacing at both sender and receiver side is important as stated in Sects. 4.4 and 4.5.

6.7 Packet Interlacing in Detail

Figure 10 explains why packet interlacing gives a bad effect to goodputs and why it is more serious in inter-domain handovers. As we can see, the MN sends an ack directly to the CN after SBU. The ack with higher sequence number makes the CN at once to send data as many as the TCP windows allow. So, data interlacing becomes severe. After all, multiple *CAs* occur as presented in Table 3. In intra-domain handover, ack interlacing does not happen since there is no SBU.

7. Conclusion

In this paper, we propose a proactive IP mobility management protocol for providing seamless services. HIMIP minimizes the probability of false-alarm, provides measurable testing time, and handles intra/inter-domain handovers altogether. Also the performances are analyzed, evaluated, and discussed. HIMIP achieves almost optimal performance. We also analyze the specific TCP behaviors during handovers and impact of packet interlacing. From that we can learn that it is important to reduce packet loss and out-of-order delivery even it costs some redundancy.

Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-2009-C1090-0902-0045).

References

[1] I.F. Akyildiz, J. Xie, and S. Mohanty, "A survey of mobility management in next-generation all-IP-based wireless systems," IEEE Wireless Commun., [see also IEEE Pers. Commun.], vol.11, pp.16–28, 2004.

[2] F.M. Chiussi, D.A. Khotimsky, and S. Krishnan, "Mobility management in third-generation all-IP networks," IEEE Commun. Mag., vol.40, pp.124–135, 2002.

[3] P. Newman, "In search of the all-IP mobile network," IEEE Commun. Mag., vol.42, pp.S3–S8, 2004.

[4] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," RFC3775, June 2004.

[5] Y. Gwon, J. Kempf, and A. Yegin, "Scalability and robustness analysis of mobile IPv6, fast mobile IPv6, hierarchical mobile IPv6, and hybrid IPv6 mobility protocols using a large-scale simulation," IEEE International Conference on Communications, pp.4087–4091, 2004.

[6] X. Perez-Costa, M. Torrent-Moreno, and H. Hartenstein, "A performance comparison of Mobile IPv6, Hierarchical Mobile IPv6, fast handovers for Mobile IPv6 and their combination," SIGMOBILE Mob. Comput. Commun. Rev., vol.7, pp.5–19, 2003.

[7] H. Soliman, C. Castelluccia, K.E. Malki, and L. Bellier, "Hierarchical mobile IPv6 mobility management (HMIPv6)," Aug. 2005.

[8] R. Koodli, "Fast handovers for mobile IPv6," RFC4068, July 2005.
[9] R. Hsieh, Z.G. Zhou, and A. Seneviratne, "S-MIP: A seamless handoff architecture for mobile IP," INFOCOM, pp.1774–1784, IEEE, 2003.
[10] Y.H. Han, J. Choi, and S.H. Hwang, "Reactive handover optimization in IPv6-based mobile networks," IEEE J. Sel. Areas Commun., vol.24, pp.1758–1772, 2006.
[11] http://nsnam.isi.edu/nsnam/, "nsnam."
[12] T. Narten, E. Nordmark, and W. Simpson, "Neighbor discovery for IP version 6 (IPv6)," RFC2461, Dec. 1998.
[13] H. Soliman, Mobile IPv6: Mobility in a Wireless Internet, Addison-Wesley Professional, April 2004.
[14] http://www.inrialpes.fr/planete/mobiwan/, "MobiWan."

Hyunku Jeong received hi B.S. degree in Computer Science from Handong University in 1999 and M.S. degree in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 2002. He has been a Ph.D. student in KAIST since 2002. His current research areas include mobile ad hoc networks, mobility management and future network architecture.

Seungryoul Maeng received the B.S. degree in Electronics Engineering from Seoul National University, Korea, in 1977, and the M.S. and Ph.D. degrees in Computer Science from KAIST, in 1979 and 1984, respectively. Since 1984 he has been a faculty member of Department of Computer Science of KAIST. From 1988 to 1989, he was with the University of Pennsylvania as a visiting scholar. His research interests include micro architecture, parallel computer architecture, cluster computing, and embedded systems.

Youngsu Chae received the B.S. and M.S. degrees in Computer Science from Postech, Korea and Ph.D. in Computer Science from Georgia Tech, in 1994, 1996 and 2002, respectively. He also had been affiliated with Samsung Electronics and Yeungnam University, Korea, as a senior research engineer and assistant professor from 2003 to 2008. Since 2007 he has been affiliated with CMAX Wireless, Korea, as the founder and CEO. His research interests include 4G small cell systems, mobile broadband networks and future Internet architecture.