This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

An Efficient Framework for Local Mobility

Feng Zhong, Yang Xia, Chai Kiat Yeo, Bu Sung Lee, Teck Meng Lim
Center for Multimedia and Network Technology
School of Computer Engineering
Nanyang Technological University, Singapore 639798
Email: {zhon0026, xiayang, asckyeo, ebslee}@ntu.edu.sg, tmlim@starhub.com

***Abstract*—To improve the local mobility performance of a wireless network, we present a new framework, namely Efficient Framework for Local Mobility (EFLoM). In EFLoM, we introduce three entities: Local Anchor Router (LAR), Wireless Access Gateway (WAG) and Mobile Node (MN). LAR is a router that is in charge of MN's IP mobility management. It keeps track of MN's movement, and delivers the MN's packets to its current location. A WAG is an access router that is responsible for managing the traffic flows within the same local mobility domain. In EFLoM, when Corresponding Node (CN) and MN connect to the same local mobility domain, packets are sent directly between CN and MN by WAG, without suffering sub-optimal routing problems. A MN is a mobile host which updates its data packets' header with its current location address before packets are sent out. To evaluate our new framework, we compare EFLoM with Hierarchical Mobile IPv6 Mobility Management (HMIPv6) [4] which is an existing protocol that has good performance for local mobility. Both analytical analysis and simulation result show that EFLoM can attain substantial improvements over HMIPv6 in terms of handover delay, end to end delay, traffic overhead, which therefore presents EFLoM as a new engineering alternative to existing MIPv6 based techniques.**

I. Introduction

With the development of mobility management protocols, the mobility management protocols are divided into two kinds: the local mobility management and the global mobility management. As Fig.1 shows, local mobility mainly involves movements across the same administratively and geographically contiguous sets of networks, and it has great application scenarios. For example, a wireless network in a metropolitan city can be viewed as a local mobile network. People traveling around this city can be modeled as MN moving between different networks. Since people move within a city frequently, local mobility is deemed to have greater application and thus the performance of local wireless network will play a more important role.

Although a lot of mobility management protocols, such as Mobile IPv6 (MIPv6) [1], Host Identity Protocol (HIP) [11] and Stream Control Transmission Protocol (SCTP) [10], have been proposed in the past few years, all of them have their own disadvantages. Take MIPv6 as an example. Firstly, MIPv6 is a global mobility management protocol and it is not specifically designed for local mobility. Secondly, MIPv6 introduces suboptimal routing problem, and this problem results in an increase in transmission delay and traffic overhead. Thirdly, MIPv6's signal cost and handover delay are high for some delay-sensed applications. To overcome these disadvantages of MIPv6, some extension protocols have been proposed. HMIPv6 [4] is one of them. HMIPv6 is a protocol aimed at reducing MIPv6's handover delay and signal cost for local mobility, but it also has its disadvantages. For example, when packets flow between two MNs under the same Mobile Anchor Point (MAP), packets need to be passed to MAP first, and the packets cannot be delivered directly between the two MNs. This paper proposes a mobility protocol to improve local mobility performance in terms of throughput, handover delay and end to end delay.

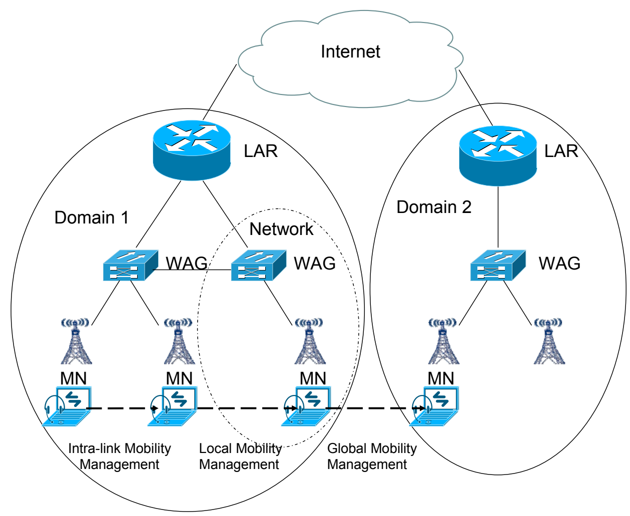

Fig. 1. Intra-link Mobility, Local Mobility and Global Mobility

In this paper, we present a new framework, namely Efficient Framework for Local Mobility (EFLoM)†, for handling local mobility issues. There are three kinds of entities in EFLoM - LAR, WAG and MN. LAR is an interface router of wireless domain to outer networks, and its responsibility is to grant service to MN in its domain and to update the data flow between the MN and CN. A LAR can deliver data packets to MN's current location. WAG is a gateway router which MN attaches directly with. It is in charge of authorizing MN's request and changing IP address for in-domain flows between MN. MN is similar to MN in HMIPv6 except that a user agent is added to EFLoM's MN. Detailed operations of the three entities will be covered in the later section of this paper.

†This work is supported by the Agency for Science, Technology and Research(A*STAR) of Singapore. (Grant number: 0721010028 - M47020034)

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

HMIPv6 [4] is one of the most common mobility protocols that aims to reduce the signal messages and handover delay in local mobile domain. Through comparison between EFLoM and HMIPv6, we can conclude that EFLoM has the following advantages over HMIPv6 [4]:

1) EFLoM does not incur traffic overhead for delivering packets. In HMIPv6, the packets are delivered to MN's current location through IP tunnel, and a tunnel header is added to each packet. However, in EFLOM, the packets' header is updated with MN's new current on-link address, and no tunnel header is required. Hence, EFLoM gets higher overall throughput than HMIPv6.
2) To support route optimization, HMIPv6 needs involvement of CN while EFLoM does not. EFLoM keeps CN transparent about the movement. But in HMIPv6, the CN has to store a cache information of MN's new on-link address. This becomes a heavy load for a CN if it has numerous customers to serve.
3) HMIPv6 suffers triangular routing problems for in-domain flows. For example, when a MN communicates with CN in the same domain, the packets are firstly sent from CN to MAP. Later, MAP encapsulates these packets and sent them to MN. However, with help of WAG, EFLoM can avoid this sub-optimal routing issue.
4) Both experimental results and mathematical analysis show that EFLoM incurs less delay in handover procedure than HMIPv6.

The remainder of this paper is organized as follows. Section II provides a brief description of related work and identify the existing issues in HMIPv6. These issues will be addressed in our new framework. Section III describes our new framework for local mobility - EFLoM. Three subsections are presented to introduce EFLoM's entities, location updating and packets forwarding. To make comparisons with HMIPv6, we will provide our mathematical analysis and simulation of EFLoM in Section IV. Section V concludes the paper and gives the future work for EFLoM.

II. Related Work

HMIPv6 [4] is an extension to MIPv6 and it aims to reduce signal cost and handover delay when MN roams within one domain. If MN moves from one network to another within the same domain, it does not have to update its home agent (HA). In HMIPv6, when a mobile node attaches to a new link, it will receive a router advertisement with a MAP option. With the MAP option, a mobile node can get MAP's IP address and form its regional care-of-address based on MAP's prefix. Through MAP's IP address, MN can judge whether it moves into a new domain.

If MN moves into a new domain, it first sends a binding update to MAP and HA, carrying its regional care-of-address and its on-link care-of-address, in order to build tunnels between HA and MAP and between MAP and MN. After that, HA will redirect all data packets destined for MN to MAP, which later tunnels these packets to MN in the new link. If MN just moves within a domain, the MAP IP address in the new network does not change. MN will just send a binding update to MAP, informing it to rebuild the tunnel to MN. The MN will not update the HA because MAP does not change.

Although HMIPv6 can reduce MIP's signaling cost, it also suffers suboptimal routing. Since data packets should go through MAP first, the data cannot be delivered between CN and MN directly. This suboptimal routing is obvious especially when CN and MN are in the same domain. At the same time, since HMIPv6 also employs tunnel to deliver packets, the traffic overhead is high. So, we have proposed a new framework to overcome these issues in HMIPv6.

III. Efficient Framework for Local Mobility

Efficient Framework for Local Mobility (EFLoM) is a new framework specific for local mobility. EFLoM mainly aims to improve the performance of mobile node in terms of handover delay, protocol signal cost and traffic overhead. In EFLoM, packets are not sent through tunnels between HA and MN. Instead, it relies on the cooperation among LAR, WAG and MN to manipulate packets' IP headers. Through updating packets' header with MN's new on-link address, EFLoM can avoid adding tunnel header in each packet. At the same time, in order to update the mapping association between home address and on-link address, a registration mechanism is required in EFLoM. After MN enters a new network and acquires a on-link address, a Register Request (RR) is firstly sent to WAG and then received by LAR to apply for service in this new network. After authentication check, LAR returns a Registering Acknowledgement (RA) message to MN, and LAR creates a mapping association between MN's home address and the new on-link address. LAR then notifies WAG about the updated mapping association, which will be used in packet forwarding. When data packets need to be transmitted, LAR or WAG substitutes the IP address in a packet's header with MN's new on-link IP address to make sure packets can be delivered directly to MN's current location.

Fig.2 illustrates the work flow of EFLoM. Under EFLoM, MN can communicate directly with its peer and get rid of triangular routing issues. The following sections detail the entities employed in EFLoM and the operation of EFLoM.

A. EFLoM Entities

Three kinds of entities are employed in this framework: LAR, WAG and MN. LAR, as an interface router of one domain, provides an interface to out-domain nodes. Since LAR is a converging router, all data packets from outside go through LAR first, and then these packets are delivered to mobile nodes in this domain. This topology is commonly used in many scenarios. For example, in a metropolitan city network, the whole city networks get access to other city through border routers. This kind of topology has many advantages such as hierarchical management and scalability. Note that, EFLoM's LAR is different from border gateway router in commonly used scenario. Additional functions are added to LAR for mobility management. Firstly, LAR needs to handle binding request from MN and to create mapping association for MN.

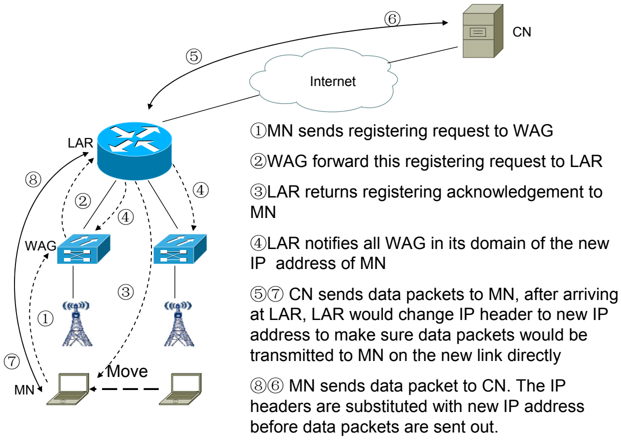

Fig. 2. Work flow of EFLoM

Then, LAR needs to notify WAG to update the mapping associations. Lastly, LAR needs to update IP header with new on-link address upon receiving packets from outside.

WAG is a gateway router for MN. It is responsible for advertising Router Advertisement, handling protocol messages and processing data flows. Specifically speaking, a WAG is responsible for advertising its Router Advertisement to all nodes within its network periodically. Based on this advertisement, MN can judge whether it has entered a new network and then acquire its new on-link address. At the same time, because WAG is the gateway router for a network, MN's register request is firstly sent to WAG and then forwarded by WAG to LAR. Moreover, WAG involves transmitting in-domain flows. When WAG receives a packet from MN, it checks the destination of this packet. If this packet is sent to MN's peer of the same domain, WAG searches its mapping table to find the mapping relationship between the peer's home address and the new on-link address. Then, WAG changes IP address of packet's IP headers and transmits the packet to the next hop.

MN in EFLoM is similar to the MN in HMIPv6, but EFLoM's MN has an IP switching agent. This agent is responsible for updating the IP header with MN's new on-link address for out-going packets, and restoring the home address of these received packets.

B. Register, Update and Deregister

Registration and updating occurs when MN enters a new network. MN receives Router Advertisements from neighboring WAG, and the advertisements contain several important information such as network prefix, IP address of LAR, lifetime, sequence number and so on. With these information, a MN can judge whether it has moved into a new network. Fig.3 illustrates the registration work flow. If MN detects it enters a new network, a register request (RR) message is sent from MN to WAG. After receiving RR, WAG does the authorization check for MN and forwards RR to LAR if the check passes. Of course, the forwarded RR is attached with MN's new on-link address by WAG. Upon receiving the RR, LAR will create a mapping association for this MN. This association is about the mapping relationship between MN's home address and its new on-link address, and both addresses are contained in the RR message. Then, a register authorization message is sent from LAR to MN, indicating that MN gets the granted service in the new network. At the same time, LAR sends an updating message to WAG in its domain, informing WAG to update relevant mapping association for this MN. These mapping associations in LAR and WAG will be used for packets' forwarding. With these mapping associations, LAR and WAG can cooperate with each other to manipulate the packets' header and to deliver packets to MN's current location successfully.

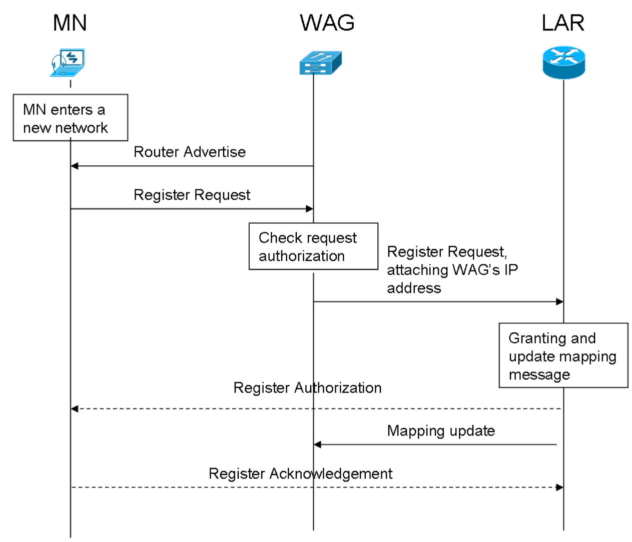

Fig. 3. A Handover scheme for EFLoM

Deregister procedure occurs as soon as MN returns to its home network. A deregister message is sent to inform LAR and MAG that MN has returned. Subsequently, the LAR and MAP remove the relevant mapping entries and do not manipulate the header of packets which destines at MN. Hence, when a packet is sent from CN to MN with MN's home address, the LAR knows MN has returned to its home network. The LAR does not change the packet's header and transmits the packet according to the destination address in the packet. Therefore, the packet is sent to the home network of MN, which is also MN's current location.

C. Packets Forwarding

In EFLoM, every Mobile Node has a home address. This home address is permanent during MN's movement, and it is used as the source address by MN's applications to establish connections with their peers. At the same time, MN is granted a new on-link address every time it enters a new wireless network.

Fig.4 and Fig.5 show the two kinds of scenario for packet forwarding. This is implemented at WAG which serves as either the first hop or the last hop for MN's packets. When packets arrive at WAG, WAG checks IP header of the packets. If the flow's two communication enders are both within the same local mobility domain, we call this flow a *iFlow*. Other flows are called *oFlow*. These two kinds of flows are illustrated as follows:

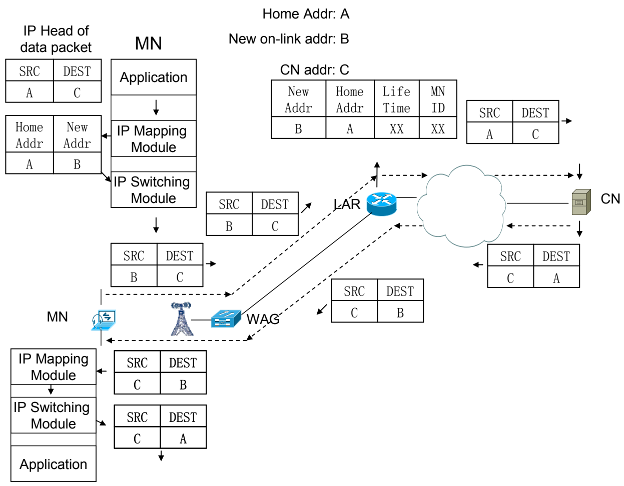

Fig. 4. oFlow pathway illustration

(i) ***oFlow*:** As stated above, the *oFlow* packets are between the CN, which is outside this domain, and MN which is inside this domain. When a MN enters another wireless network, it first acquires a new address on this new link. This new address serves as an interface for MN on this link. When MN's application wants to send a data packet to its CN, MN's applications use home address and pass it to IP Switching Module before the packet is sent out. In the IP Switching Module, the source address is replaced by new on-link address and then passed to WAG. For *oFlow*, WAG just performs a forwarding function as a usual router. These packets will arrive at LAR because LAR is the interface router to the domain. Upon receiving the packet, LAR checks its ip header and searches the IP mapping association for the source MN, restores the packet's source address with the home address of source MN. Thus, CN has no idea about the movement of MN because the received packet's source address and destination address are always the same during the session no matter which network MN moves in. In the reverse direction, it is quite similar. The destination field of a packet, which is from CN, will get updated by LAR with the new on-link address, and then delivered to destination MN at its current location. Before the packet is passed up to the application, the IP Mapping Module will substitute the destination address with MN's home address to keep its application from sensing the movement.

(ii) ***iFlow*:** As illustrated in Fig.5, the forwarding function is implemented in the first hop of WAG and the last hop of WAG. For example, when a MN wants to send a packet to its peer MN which is in the same domain, the IP Mapping Module in source node will update source address as stated

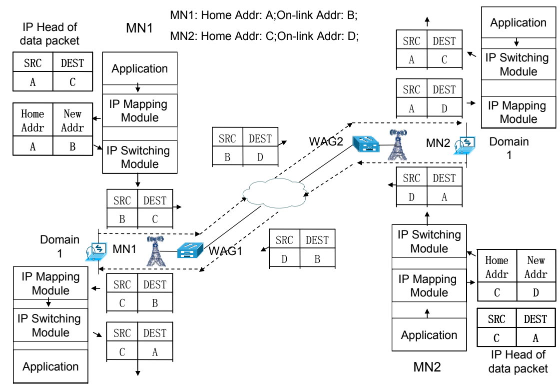

Fig. 5. iFlow pathway illustration

before and forward it to its WAG. The first-hop WAG will find this flow is *iFlow* and it will search the mapping entry for destination node. If the destination node enters another network, the WAG replaces the address's destination field with the destined MN's current link address. Since the destination is updated, the packet is directly passed to the current network of the destined MN. The last-hop WAG finds that the packet is transmitted to the network it is in charge of, so it updates the packet's source address with the source MN's home address before transmitting it to the destined MN. The destined MN restores the destination address for incoming packet and passes it to upper layer applications. The reverse is the same for the return packets.

IV. Performance Analysis And Simulation

In this section, we compare EFLoM with HMIPv6 in terms of handover delay, transmission delay and traffic overhead. In our opinion, the EFLoM has many advantages over HMIPv6. For example, the EFLoM supports the optimized route without requiring any revision on CN, while HMIPv6 incurs transmission delay between MAP and Home Agent. By substituting IP address, EFLoM do not add traffic overhead during the transmission. Therefore, EFLoM's throughput is higher than HMIPv6 which adds a tunnel header to each packet.

Let us first check handover delay of EFLoM and HMIPv6. As introduced before, the registration procedures of these two protocols are similar. When a MN enters a new network, a series of messages will be exchanged among nodes and routers before MN finishes the registration. Now, we define $T_{handover}$ as the handover duration the mobile node experiences. This handover duration lasts from the time a MN enters a new network to the time MN finishes registration. It can be divided into two components: T_{detect} which is the time a MN spends to detect the on-link change and $T_{register}$ that is defined as the duration when a MN starts to send a RR until it registers successfully. We assume the time, T_0, which packets spent on every hop is the same and end to end delay is mainly decided by the number of hops traversed by a packet. So, we define the symbol N_{O-D} as the hop count from origin node O to destination node D. From [4], we can get

$$\begin{aligned} T_{handover} &= T_{detect} + T_{register} \\ &= (N_{detect} + N_{register}) * T_0 \quad (1) \\ N^{EFLoM}_{detect} &= 4 * N_{MN-DHCP} \quad (2) \\ N^{HMIPv6}_{detect} &= 4 * N_{MN-DHCP} \quad (3) \\ N^{EFLoM}_{register} &= 2 * N_{MN-WAG} + 2 * N_{WAG-LAR} \quad (4) \\ N^{HMIPv6}_{register} &= 2 * N_{MN-AR} + 2 * N_{AR-MAP} \\ &\quad + 2 * N_{MAP-LAR} \quad (5) \end{aligned}$$

From the above equations, we can see that T_{detect} is almost the same in comparison because both protocols depend on DHCP to get a new on-link address when MN enters a foreign network. For $T_{register}$, EFLoM saves more time than HMIPv6 because only after the registration message arrives at LAR and LAR updates its mapping association, then data packets will be sent to MN on the new link. Whereas $T_{register}$ in HMIPv6 depends on the MAP's level. The lower the MAP level, the lesser is the MN-to-MAP Round Trip Time(RTT) and the more frequently MN switches its MAP. Since MN picks the same level MAP, N_{AR-MAP} and $N_{WAG-LAR}$ are equal and it is obvious that HMIPv6 will take more time to finish handover because HMIPv6 needs a RTT time of $2 * (N_{MN-AR} + N_{AR-MAP}) * T_0$ plus a tunnel-establishing time of $2 * N_{MAP-LAR} * T_0$.

To prove it, we have conducted an experiment based on ns2 to simulate the handover procedure. Firstly, we create a domain with 3 networks. Each network covers certain area of the domain and all networks converge to one LAR which serves as an interface of the wireless domain. Then, we create MN and designate one network as its home network. With TCL script, we force MN to move from its Home Network to a foreign network across another foreign network, and then back to home agent. The traced throughput graph is shown in Fig.6. From this figure, we can see steep fall in throughput at the time when MN processes a handover as it enters a new network. The duration taken for MN to recover from handover can be measured from the width of the dip. Hence, we can conclude that EFLoM incurs less handover delay than HMIPv6.

Similarly, we can calculate the transmission delay of EFLoM and HMIPv6. Just take a flow from CN to MN as an example. It is easy to observe that no matter which type of topology is used, EFLoM will ensure the transmission delay is less than HMIPv6 because EFLoM does not depend on tunnel transmission and there is no suboptimal routing issue. By substituting IP address of data packet with new on-link address of MN, EFLoM can transmit flows directly between MN and its peer along the shortest path compared to HMIPv6. As stated in [5], the packet transmission delay depends on network's topology. The focus of EFLoM is the local mobility, hence we assume MN chooses the same level MAP in HMIPv6 during its movement and the packets are forwarded directly to MN without transversing between the Home Agent and the Mobility Anchor Point. (6), (7) and (8) have given the transmission delay of EFLoM and HMIPv6 separately, where

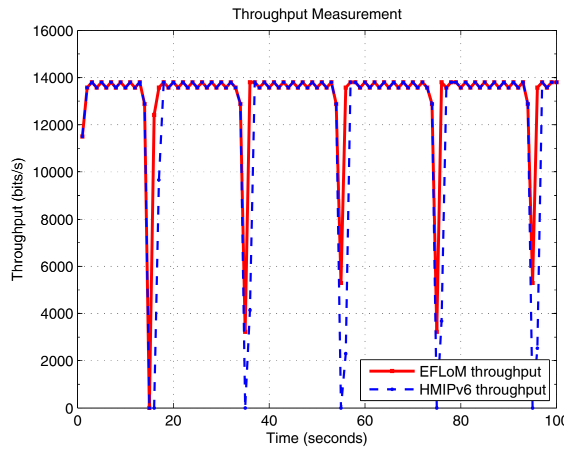

Fig. 6. Handover delay comparison

$N^{HMIPv6opt}_{TransDelay}$ means the hop counts where route optimization is used in HMIPv6 and $N^{HMIPv6}_{TransDelay}$ is the hop counts of HMIPv6 without route optimization.

$$\begin{aligned} N^{EFLoM}_{TransDelay} &= N_{CN-LAR} + N_{LAR-MN} \quad (6) \\ N^{HMIPv6}_{TransDelay} &= N_{CN-HA} + N_{HA-MAP} \\ &\quad + N_{MAP-MN} \quad (7) \\ N^{HMIPv6opt}_{TransDelay} &= N_{CN-MAP} + N_{MAP-MN} \quad (8) \end{aligned}$$

Obviously, transmission delay of EFLoM and HMIPv6 with route optimization is the same if MN in HMIPv6 takes the same level MAP during its movement. Hence, we mainly conduct our experiment comparing EFLoM with HMIPv6 without route optimization. Firstly, we construct the same scenario with the experiment of handover time and keep the CN sending packets to its MN all the time. We randomly move MN to test the packet transmission delay during MN's movement. Then, we record the delay time packets experienced in these two frameworks and depict them in Fig.7. It is obvious that packets in EFLoM incur less delay than the packets in HMIPv6. What is more, the jitter of packets in EFLoM is quite smaller than HMIPv6. Hence, EFLoM is beneficial for those applications which are sensitive to jitter.

As for traffic overhead, since HMIPv6 transmit packets based on tunnel technology, an extra IP header is added to every packet. Let's assume a CBR traffic is sent from CN to MN. The CBR source generates a total of S = R * (8L) * t bits in a duration of t. For HMIPv6, the amount of traffic generated onto the network from CN to MN (C^{HMIPv6}) is shown in (9).

$$\begin{aligned} C^{HMIPv6} &= 8 * Rt(L + L^{IPinIP}) N^{HMIPv6}_{MN-MAP} \\ &\quad + 8 * RtL N^{HMIPv6}_{CN-MAP} \quad (9) \\ C^{EFLoM} &= 8 * RtL N^{EFLoM}_{CN-LAR} \\ &\quad + 8 * RtL N^{EFLoM}_{MN-LAR} \quad (10) \end{aligned}$$

where L^{IPinIP} is the size of packets' outer header. N^{HMIPv6}_{CN-MAP} and N^{HMIPv6}_{MN-MAP} are the number of hops from

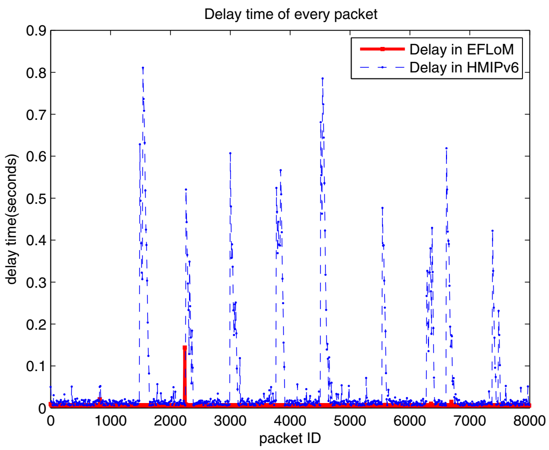

Fig. 7. Average packet delay

CN to MAP or MN to MAP respectively.

However, in EFLoM, the amount of traffic generated into the network is $S = R * (8L) * tbits$ and no additional IP header is added along the way. So the total cost of EFLoM (C^{EFLoM}) can be calculated as in (10). For transmitting the same amount of CBR traffic, the traffic cost of EFLoM over the traffic cost of HMIPv6 is:

$$\frac{8 * RtLN^{EFLoM}_{CN-LAR} + 8 * RtLN^{EFLoM}_{MN-LAR}}{8 * Rt(L + L^{IP-in-IP})N^{HMIPv6}_{MN-MAP} + 8 * RtLN^{MIPv6}_{CN-MAP}}$$

which is obviously less than 1. So EFLoM can save bandwidth resources compared to HMIPv6 when transmitting data packets.

V. Conclusion And Future Work

In this paper, we proposed a new Efficient Framework for Local Mobility (EFLoM) to improve the performance of local wireless network in terms of handover delay, traffic overhead and end to end delay. By manipulating the data packets, EFLoM enables MN to roam within local wireless networks without any involvement of CN. At the same time, with the help of a IP mapping technology, EFLoM avoids traffic overhead and saves a lot of bandwidth resource. Moreover, the mathematical analysis and simulation result show EFLoM's handover delay and end to end delay are smaller than HMIPv6. EFLoM is therefore a new engineering alternative to MIPv6 based schemes.

Of course, EFLoM is not a complete local mobility protocol. For future work, we plan to further develop the security function and failure handling function of EFLoM.

References

[1] J. Kempf, “Problem Statement for Network-Based Localized Mobility Management (NETLMM)”, RFC 4830, April 2007
[2] D. Johnson, C. Perkins and J. Arkko, ” Mobility Support in IPv6 ”, RFC 3775, June 2004.
[3] Lee. B.S, Lim.T.M, Yeo.C.K and Le.Q.V “ A Mobility Scheme for Personal and Terminal Mobility ”, in IEEE International Conference on Communications 2007, Glasgow, Scotland, 24-28 June 2007.
[4] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier ” Hierarchical Mobile IPv6 Mobility Management (HMIPv6) ”, RFC 4140, August 2005.
[5] H. Soliman, “Mobile IPv6 : Mobility in a Wireless Internet,” Addison-Wesley, 2004.
[6] J. Kempf, “Goals for Network-Based Localized Mobility Management (NETLMM)”, RFC 4831, April 2007
[7] C. Vogt, J. Kempf, “Security Threats to Network-Based Localized Mobility Management (NETLMM)”, RFC 4832, April 2007
[8] E.R.Koodli, “ Fast Handovers for Mobile IPv6”, RFC 4068, July 2005.
[9] P.Srisuresh, J.Kuthan, J.Rosenberg, A.Molitor, and A.Rayhan, ” Middlebox communication architecture and framework”, RFC 3303, Aug.2002.
[10] R. Stewart, ”Stream Control Transmission Protocol”, RFC 4960, September 2007
[11] Moskovitz, R. and P. Nikander, ”Host Identity Protocol (HIP) Architecture”, RFC 4423, May 2006.