

# Leveraging Proxy Mobile IPv6 with SDN

Syed M. Raza, Dongsoo S. Kim, DongRyeol Shin, and Hyunseung Choo

**Abstract:** The existing Proxy Mobile IPv6 suffers from a long handover latency which in turn causes significant packet loss that is unacceptable for seamless realtime services such as multimedia streaming. This paper proposes an OpenFlow-enabled proxy mobile IPv6 (OF-PMIPv6) in which the control of access gateways is centralized at an OpenFlow controller of a foreign network. The proposed OF-PMIPv6 separates the control path from the data path by performing the mobility control at the controller, whereas the data path remains direct between a mobile access gateway and a local mobility anchor in an IP tunnel form. A group of simple OpenFlow-enabled access gateways performs link-layer control and monitoring activities to support a comprehensive mobility of mobile nodes, and communicates with the controller through the standard OpenFlow protocol. The controller performs network-layer mobility control on behalf of mobile access gateways and communicates with the local mobility anchor in the Proxy Mobile IPv6 domain. Benefiting from the centralized view and information, the controller caches the authentication and configuration information and reuses it to significantly reduce the handover latency. An analytical analysis of the proposed OF-PMIPv6 reactive and proactive handover schemes shows 43% and 121% reduction in the handover latency, respectively, for highly utilized network. The results gathered from the OF-PMIPv6 testbed suggest similar performance improvements.

**Index Terms:** IP mobility, OpenFlow, proxy mobile IPv6 (PMIPv6), software defined networks (SDN).

## I. INTRODUCTION

RAGING technological advancements in the last decade and the increasing availability of high speed Internet have exponentially increased the usage of mobile devices. As the hardware in mobile devices is getting more advanced and the data rates growing exponentially, users are more inclined to use media enriched, interactive and realtime services on the go. Mobility management is essential to provide seamless and delay free experience to the users.

The current architecture of the IP protocol and the Internet client/server model does not facilitate network layer mobility management. A logical connection between a client and a server is based on a socket comprising of IP addresses, port numbers

of the two terminals and the protocol. The connection has to be reestablished in case any of these parameters alter. In a mobile environment, the IP address and port number of a client changes, as soon as it attaches to a different gateway. The client reestablishes a logical connection with the server using the new IP address. This disruption in connection causes delay and degradation in service and user experience.

Proxy mobile IPv6 (PMIPv6) [1] is a network based mobility management protocol, standardized by the Internet Engineering Task Force (IETF). In PMIPv6, mobile access gateways (MAGs) are responsible to perform mobility control signaling with the local mobility anchor (LMA) on behalf of a mobile node (MN). The anchor acts as a home agent within a PMIPv6 domain. As shown in the Fig. 1, the gateways perform authentication of a mobile node using an authentication server; and creates a bi-directional IP tunnel with the anchor after the required control signaling. In a PMIPv6 domain, a home network prefix (HNP) is assigned to a mobile node by the AAA server. A constant IPv6 address is maintained at the mobile node via the stateless auto IPv6 address configuration mechanism, as it moves within a PMIPv6 domain. Handover of a mobile node to the next gateway requires the gateway to perform control signaling with the anchor, which causes excessive and unacceptable delay for realtime services.

Vigorous research has been conducted on the PMIPv6 and many schemes have been proposed to reduce the handover delay and packet loss [2]–[4]. For example, fast proxy mobile IPv6 (FPMIPv6) [5] requires participation of a mobile node in the proactive handover process, and hence does not conform to the fundamental concept of the PMIPv6 network based protocol. Other pure network based proposed schemes reduce the handover delay and packet loss to great extent. The limitations that current schemes incur are, high bandwidth requirement due to of excessive control signaling; and intense computation intensity either on the gateway or on the anchor [3], [4]. These limitations result in higher operational cost and pose major hurdle in deployment of these schemes in realistic scenarios.

This paper introduces the concept of software defined networks (SDN) in the PMIPv6, by separating the PMIPv6 control and data planes. In SDN based PMIPv6, a central control entity is responsible to perform all the mobility related computation and control signaling with an anchor on behalf of all the gateways in a PMIPv6 domain. A major benefit of the control and data plane separation and centralization of control signaling is; simple, cost-effective and easy to develop gateways, which are only responsible for the layer two functionalities, data forwarding and minimal layer three control signaling. A global view of a PMIPv6 domain at the central control entity, enables the development of comprehensive and efficient mobility management schemes while providing the backward compatibility. As central entity takes the responsibility of control signaling, the signaling

Manuscript received October 22, 2014; approved for publication by Choong Seon Hong, Division III Editor, August 23, 2015.

This manuscript is an extended version of the paper presented and published in ACM IMCOM 2014.

This research was supported in part by IITP and PRCP of MOE, Korean government, under the IITP(14-911-05-006) and NRF (NRF-2010-0020210), respectively.

S. M. Raza, D. Shin, and H. Choo (corresponding author) are with College of Information and Communication Engineering, Sungkyunkwan University, Korea, email: {s.moh.raza, drshin, choo}@skku.edu.

D. S. Kim is with School of Engineering and Technology, Indiana University Purdue University Indianapolis, USA, email: dskim@iupui.edu.

Digital object identifier 10.1109/JCN.2016.000061

This is the author's manuscript of the article published in final edited form as:

Raza, S. M., Kim, D. S., Shin, D., & Choo, H. (2016). Leveraging proxy mobile IPv6 with SDN. *Journal of Communications and Networks*, 18(3), 460-475. <http://dx.doi.org/10.1109/JCN.2016.000061>

overhead at the gateways and the anchor reduces in comparison with the other schemes, and management of the gateways become easier.

In this paper the SDN concept of logically centralized control is implemented in PMIPv6 through OpenFlow [6], the de facto implementation of the software defined networks, resulting an OpenFlow based PMIPv6 (OF-PMIPv6). In the proposed OF-PMIPv6, the controller resides in the backbone network and connects to all the gateways and the anchor. The gateways implement the OpenFlow protocol, upon which the controller communicates with them. The communication between an anchor and the controller is over IPv6, hence an anchor does not implement the OpenFlow protocol. An OpenFlow mobile access gateway (OMAG) notifies the controller about a mobile node attachment through PMIPv6 control message in the OpenFlow protocol, and the controller performs all the PMIPv6 related mobility control signaling with the anchor and authentication server on behalf of the OMAG. For the data communication, an OMAG directly creates IP tunnel with the anchor. The Fig. 2(a) depicts the separation of control and data paths in OF-PMIPv6.

The rest of the paper is managed as follows. Section II presents our motivation for this work and background concepts related to the PMIPv6 and the OpenFlow. The OF-PMIPv6 architecture, its different components and their functioning are thoroughly discussed in Section III. Section IV presents the analytical models for the PMIPv6 and the OF-PMIPv6 which are later utilized in the evaluation. A brief discussion related to the OF-PMIPv6 testbed implementation is presented in Section V. Results based on the PMIPv6 and OF-PMIPv6 analytical models, and the OF-PMIPv6 testbed, are used for the performance evaluation, in Section VI. Finally, the conclusions are drawn and future directions are discussed in Section VII.

## II. MOTIVATION AND BACKGROUND

Handovers in the IEEE 802.11, suffer from high latency due to the scanning for a new access point [7]. It has also been reported that for IPv6, 90% of the total (layer two + layer three) delay is caused by layer three handover mainly due to the duplicate address detection and move detection phase. Perkins *et al.* [8] presents how mobility can be supported in IPv6 through binding a mobile nodes care of address in a foreign network with its home agent in a home network. A lot of work has been done in order to reduce the handover delay in the IPv6. Koodi *et al.* [9] presents the Fast Mobile IPv6 which enables a mobile node to quickly detect that it has moved to a new subnet by providing a new access point and an associated subnet prefix information when the mobile node is still connected to its current subnet. In [10] the authors have provided a draft proposal which discusses about how the IEEE 802.11 Management frames can be extended and utilized to advertise the capability information of the access points. They recommended an improved moment detection mechanism by avoiding the unnecessary layer three messaging over wireless link. McNair *et al.* [11] proposes a two-path handover mechanism in the Mobile IPv6, which maintains the QoS for multimedia traffic while enabling large-scale support in the Internet. These mobile based mobility management schemes require a mobile node to be actively involved during

the handover decision and the execution procedure. A network based mobility management standard of PMIPv6 [1], handles the mobility related signaling without involving a mobile node. This opens up the opportunities for innovation in the networks to handle mobility management without enforcing any changes in a mobile node, which is an important factor in terms of compatibility of mobile nodes with the network.

### A. Proxy Mobile IPv6 (PMIPv6)

In the PMIPv6, the entire data communication between a mobile node (MN) and a corresponding node (CN) happens through a local mobility anchor (LMA). The anchor serves as a home agent for a mobile node. It is considered as the topological anchor point for the mobile nodes home network prefix and manages a mobile nodes binding state [1]. When a mobile node first enters in a PMIPv6 domain, it needs to register itself with the anchor. This control signaling is performed by a mobile access gateway (MAG) on behalf of a mobile node. After completion of the layer two connectivity, the mobile node sends a router solicitation (RS) message to the gateway [1]. On receiving the RS message, the gateway sends the mobile node ID to an authentication, authorization, and accounting server (AAA server) for authentication. In response, the AAA server authenticates the mobile node and provides the gateway with a home network prefix (HNP) of the mobile node. The gateway then sends a proxy binding update (PBU) message to the anchor, which contains the mobile node ID and the HNP. The anchor creates/updates an IP tunnel with the gateway and makes appropriate entries in its routing table before responding with a proxy binding acknowledgement (PBA) message. Once the gateway receives the acknowledgement, it makes a routing table entry and creates/updates the IP tunnel with the anchor. This sets up a bi-directional IP tunnel between the gateway and the anchor for data communication. The gateway sends a router advertisement (RA) message to the mobile node before the data flow begins between the mobile node and the corresponding node via the anchor.

In case of handover, a mobile node disconnects from its previous gateway (pMAG) and sends an RS message to a next gateway (nMAG). The rest of the control signaling is similar to a mobile node registration process. The corresponding node is oblivious to the mobile node handover, as the anchor makes necessary changes in its routing table and routes the received data packets from the corresponding node over the new IP tunnel created with the next gateway while maintaining the mobile node IP address. The control signaling involved in the handover procedure increases the time during which mobile node is disconnected, causing the packet loss which is unacceptable for the realtime services.

Many schemes have been proposed to improve the PMIPv6 handover latency. FPMIPv6 [5] reduces the handover latency and packet loss considerably by allowing a mobile node in the control signaling which does not go along with the fundamental concept of PMIPv6. Jeon *et al.* [12] have proposed a solution of two-phase tunnel control based on the IEEE 802.21 to handle early packet forwarding problem in the FPMIPv6. Smart buffering [4] reduces the packet loss on the expense of control signaling and extra memory requirement; however handover de-

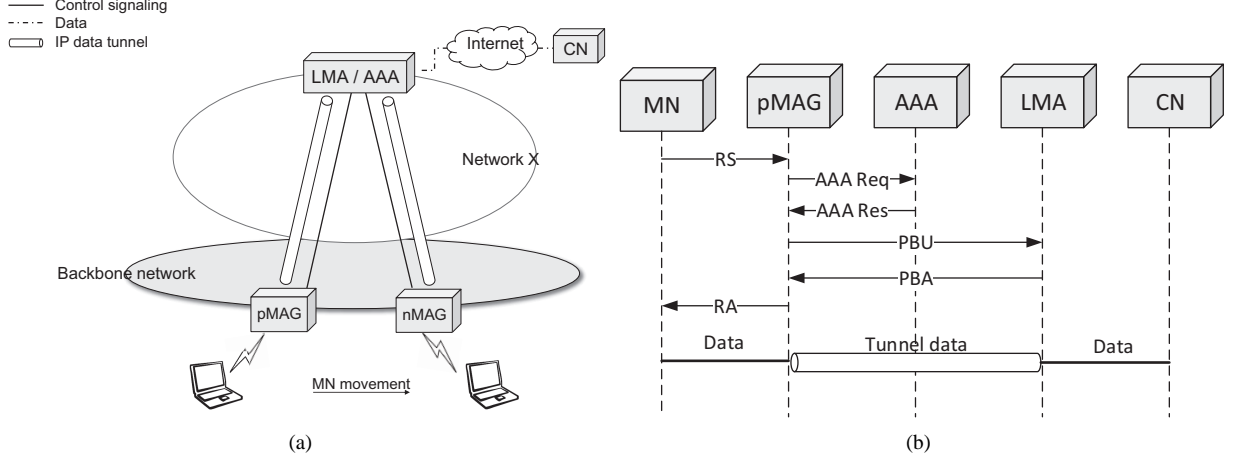


Fig. 1. PMIPv6 architecture and signaling call flow: (a) PMIPv6 architecture and (b) mobile node registration signaling call flow.

lay remains same as the standard PMIPv6. Kim *et al.* [2] proposed a solution to mitigate the packet loss during the handover in PMIPv6 by introducing a buffering mechanism at the anchor, requiring larger buffering space at the anchor to cater the needs of all the mobile nodes for a large scale network. Oh *et al.* [3] also proposed a mechanism to relieve the packet loss during the handover by buffering the packets at the optical buffering module of the anchor and reduces the handover latency as well by simplifying the authentication mechanism during the PMIPv6 handover process.

### B. OpenFlow

The concept of SDN has been floating in the academic circles for a long time, and OpenFlow (OF) [6] has recently emerged as the first implementation of SDN. The initial motivation for the OpenFlow was to enable researchers to perform their research on real production networks. Later, OpenFlow is standardized by the open networking foundation (ONF) [13]. A central control entity in the OpenFlow network is known as controller. The OpenFlow has been opted to implement the SDN concept in different types of networks, e.g., wireless sensor networks, mesh networks, data centers, etc. These networks consist of a set of OF-enabled network devices (switch/router/access point) and a controller. A network device consists of a data plane and a control plane. The data plane is responsible for packet forwarding whereas the control plane takes care of communication between the OF switch and the controller over a secure TCP connection. The main objective is to make the network control functions more centralized rather than distributed. The controller performs all the control logic and manages all the forwarding elements using the OF protocol. The OF switch consists of a flow table which performs packet lookup and forwarding. An entry in a flow table consists of three main fields, header, action and statistics. Any incoming packet is matched against the header fields of the entries in the flow table; if the match is found then action mentioned in the action field of the matched entry is performed. In case of no match (table miss) the packet is sent to the controller which installs a new flow entry in the flow table corresponding to that packet.

### III. PROPOSED OF-PMIPV6 ARCHITECTURE

This paper proposes the OF-PMIPv6 architecture, which integrates the OpenFlow with the PMIPv6 without modifications. The mobile access gateways in the OF-PMIPv6 (OMAG) are only responsible for the layer two functionalities, while the layer three related PMIPv6 control signaling is taken care by the central controller. Our design of the OF-PMIPv6 architecture is motivated by a low handover latency, reduced packet loss, simplicity, extensibility, scalability, sustainability, compatibility with the existing PMIPv6 and scalability. The proactive scheme in the proposed architecture needs to achieve the handover latency and the packet loss within an acceptable range for realtime services, while not burdening the gateways and the anchor with the extensive control signaling. The proposed architecture should be easily extensible to incorporate different mobility management schemes for reducing the handover latency and the packet loss, and should provide scalability for addition/removal of different network elements. Sustainability and compatibility are key design features, where the proposed architecture should cater the requirements of the production network and should work with the standard PMIPv6 [1] as well.

In the OF-PMIPv6 architecture the OMAGs are considered as the access network to which a mobile node makes the layer two attachment and they also serve as a gateway for a mobile node. Over the wired link, OMAGs perform the PMIPv6 control signaling with the controller in the backbone/core network of the service provider. The anchor and the AAA server resides in the home network of a mobile node. On behalf of the OMAGs the controller performs the PMIPv6 control signaling with the anchor and the AAA server and this communication travels through either the home network of the service provider or any foreign network, presented as network 'X' in the Fig. 2(a). Bi-directional IP data tunnel is created between the anchor and the OMAG for the transmission of data packets. The Fig. 2(a) presents the architecture of the OF-PMIPv6 where the control signaling is through the controller, and the data transmission is through the IP tunnel between the OMAGs and the anchor. The Fig. 2(b) shows the signaling call flow for the mobile node registration in the OF-PMIPv6.

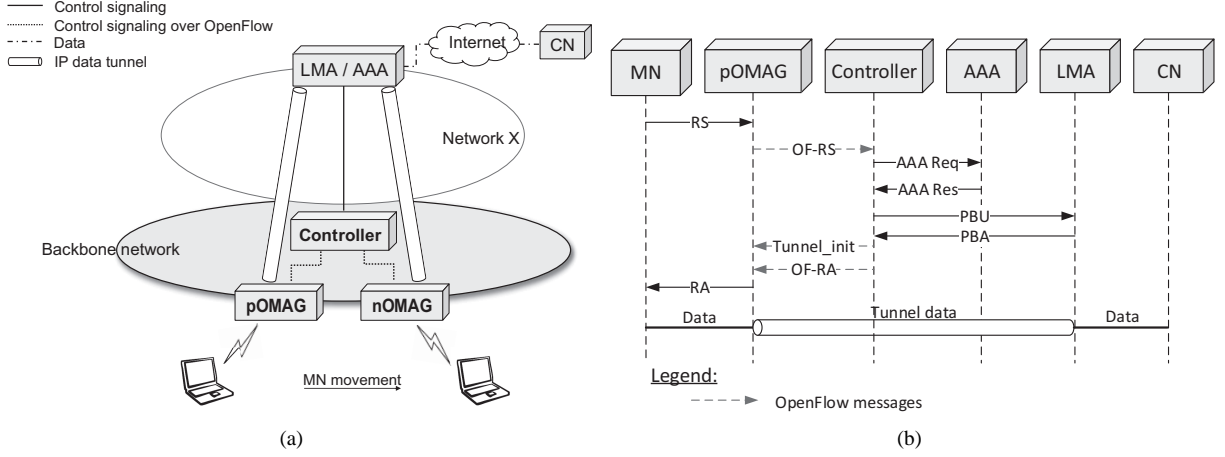


Fig. 2. OF-PMIPv6 architecture and signaling call flow: (a) OF-PMIPv6 architecture and (b) mobile node registration signaling call flow.

Both the control signaling and the data communication in the PMIPv6 takes place over the same path between the gateway and the anchor. The OF-PMIPv6 separates the control signaling path from the data communication path. The controller performs the PMIPv6 control signaling with the anchor/AAA on behalf of all the OMAGs, thereby logically virtualizing the multiple OMAGs as one OMAG, therefore we consider it as a virtual mobility access gateway (vMAG). Communication between the OMAGs and the vMAG (in the controller) is over the OpenFlow protocol. This logical separation of the paths is evident from the Fig. 2(a). As the control signaling has been offloaded from the OMAGs to the controller, the OMAGs are mainly responsible for layer two functionalities and IP tunnel management.

#### A. OF-PMIPv6 Components

##### A.1 OpenFlow Enabled Mobile Access Gateway (OMAG)

In comparison with PMIPv6, an OMAG is the main evolved component in the OF-PMIPv6. The OMAGs responsibility of the PMIPv6 control signaling with the anchor is offloaded to the controller, and the OMAG communicates with the controller over the OF protocol. Therefore, the main functionalities of an OMAG are layer two forwarding of data packets, IP tunnel management and MNs link state monitoring for the proactive scheme.

##### Link State Monitoring:

An OMAG monitors the link state of the mobile nodes attached to it, by recording the received signal strength (RSS) value of the arrived beacon or any other frames/packets. An OMAG also samples the RSS values of the mobile nodes that are not attached to the OMAG, by overhearing their communication. In single sample OMAG can receive multiple frames/packets from a mobile node. These link state values are maintained as a moving average, and are reported to the controller in case of an event. The event occurs when the link state of a mobile node crosses over either a lower threshold (too bad) or a higher threshold (too good). Appropriately weighted moving average of the link state values and the two thresholds ensure that the controller is not overwhelmed by the link state values, and enable the controller to make handover decision in proactive

scheme with reasonable accuracy, and prevents the system to become unstable through the ping pong effect. To monitor the link state of the mobile nodes which are not attached to the OMAG, monitor mode of the IEEE 802.11 is utilized.

##### OpenFlow Protocol:

An OMAG implements the OF protocol, to communicate the mobile node link state values and the PMIPv6 related signaling to the controller. The standard OF protocol does not support PMIPv6 related control signaling. An experimenter message type is provided in the OF protocol to facilitate the addition of new message types and support for other protocols. The experimenter message type is used in the OF-PMIPv6, to provide support for the PMIPv6 control signaling. This extension of the OF protocol includes three new messages (Tunnel\_init, S\_Report and L\_Report). The controller sends the Tunnel\_init message to the next OMAG (nOMAG) in the reactive/proactive handover, to initiate an IP tunnel from the next OMAG to the anchor. The OMAGs report mobile nodes link state to the controller through S\_Report message. The OMAGs uses the L\_Report message to send the lost report to the controller, if no signal is received from a mobile node for a certain period.

##### Tunnel Management:

From the PMIPv6 perspective, the main responsibility of an OMAG is to maintain the status of IP tunnels between an OMAG and the anchor for each associated mobile node. In the proactive OF-PMIPv6 scheme an OMAG maintains the status of IP tunnel for a mobile node as provisional or conclusive. When the tunnel status is provisional for a particular mobile node, the OMAG buffers the data packets received from the IP tunnel for that mobile node, without forwarding them to the mobile node. These buffered data packets along with the normally received data packets are forwarded to the mobile node without buffering once the status is changed to conclusive.

##### A.2 OpenFlow Controller

The controller resides in the backbone network as presented in Fig. 2(a). In OF-PMIPv6 the controller acts as a central control entity and performs the PMIPv6 related control signaling with the anchor on behalf of all the OMAGs in the OF-PMIPv6

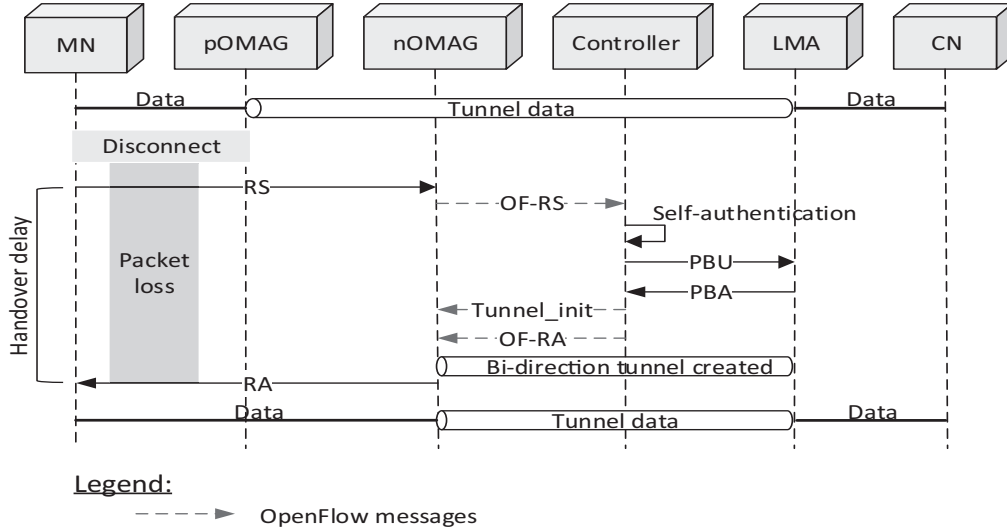


Fig. 3. Reactive OF-PMIPv6 handover signaling call flow.

domain, therefore it requires to communicate both with the OMAGs and the anchor. The controllers in built modules takes care of much of the OpenFlow communication with the OMAG, however newly added OF-PMIPv6 modules are required to support the mobility related communication with the OMAG and the anchor. The controller has a complete view of entire network and the added OF-PMIPv6 mobility management module utilizes this information to perform the reactive and the proactive handovers while maintaining the proper state of a mobile node through the finite state machines.

#### OpenFlow Module:

OpenFlow module is responsible for communicating the PMIPv6 control and mobility related messages with OMAG, by using under-laying in-built functions of the controller. The OpenFlow module also facilitates communication between different OF-PMIPv6 modules in the controller. The main purpose of the OF protocol is to handle the forwarding plane of the network elements, and offers no support for the mobility management. The experimenter message type of the OF protocol is used to provide required support for the OF-PMIPv6 control signaling between the controller and OMAGs, which is handled by the OpenFlow module in the controller.

#### PMIPv6 Module:

The PMIPv6 module is responsible for performing standard PMIPv6 control signaling with the anchor and the AAA server. On directions from the OpenFlow module, it constructs the PMIPv6 message (e.g., PBU message) accordingly and sends it to the anchor or the AAA server. Similarly it provides the required information to the OpenFlow module once a response is received from the anchor or the AAA server.

#### Mobility Management Module:

Mobility Management module consists of a connectivity database (C-DB) which maintains the information of the mobile nodes such as MN ID, LMA ID, attached OMAG ID and MN link state values from different OMAGs. For each link state value, a timestamp is also recorded and is considered expired

after a particular time period. Once all the link state values for a mobile node are expired then it is considered to have left the OF-PMIPv6 domain. To perform the proactive handover, the link state values in the C-DB are utilized to determine the handover initiation instant and the next OMAG. During the reactive or proactive handover the proper state of a mobile node is maintained in the mobility management module.

#### B. Operation of OF-PMIPv6

A mobile node registration and handover are two major operations of the OF-PMIPv6. Upon entering the OF-PMIPv6 domain, a mobile node associates itself to an OMAG and sends a RS message. The OMAG forwards the RS message to the controller via OF protocol (OF-RS message). The controller extracts the mobile ID from the OF-RS message. On finding no entry against the received mobile node ID in the C-DB, the controller sends it to the AAA server for authentication. The AAA server authenticates the mobile node ID and provides the home-network prefix (HNP) of the mobile node to the controller. Using the received HNP, the controller sends a PBU message to the anchor. In the PMIPv6, a PBU message contains the mobile node ID and the mobile node HNP, whereas in the OF-PMIPv6, a PBU message also contains the OMAG ID from whom the RS message is received, as the anchor is required to create an IP tunnel between itself and the OMAG. After making the routing table entries, the anchor replies to the controller with a PBA message. On receiving the PBA message, the controller sends a message to the OMAG for creation of an IP tunnel with the anchor. The OMAG updates its routing tables and creates the IP tunnel with the anchor if it does not exist already. Also the controller sends a RA message to the mobile node through the OMAG (OF-RA). The C-DB is updated with the information of the newly registered mobile node. Registration completes when the mobile node receives the RA message. The signal call flow presented in the Fig. 2(b) shows the mobile node registration procedure.

In case of handover, the OF-PMIPv6 works either in reactive or proactive mode depending on the controller and the OMAG configuration.

### B.1 Mobile Node Handover in Reactive OF-PMIPv6

In the reactive OF-PMIPv6, the mobile node handover is almost similar as handover in the PMIPv6 [1] with minor differences. First is that upon receiving the RS message from a mobile node, the OMAG does not send the PBU message to the anchor, instead forwards the OF-RS message to the controller, and the controller takes care of rest of the control signaling with the anchor which is similar to the mobile node registration procedure in OF-PMIPv6. Secondly in PMIPv6 next gateway has to re-authenticate the mobile node with the AAA server, as it cannot distinguish between a mobile node handover and registration. This causes the increase in PMIPv6 handover latency. Re-authentication of a mobile node is not required in the reactive OF-PMIPv6, because of the cached mobile node information in the C-DB., and this is presented in the Fig. 3 through Self-Authentication internal message in the controller.

To make sure that there is no bogus mobile node record in the C-DB, the controller only keeps the record of those mobile nodes which are connected with the OF-PMIPv6 network at the given time. In the reactive mode correct state of the C-DB is ensured through the mobile node lost report sent by the OMAG, as in the reactive mode there are no link state messages. The mobile node connection lost report triggers a timer (validation timer) at the controller, and if the controller does not receive the same mobile nodes RS message from the other OMAG before the expiration of validation timer the mobile node record is removed, otherwise it is kept. Exclusion of the re-authentication process, reduces handover latency in the reactive OF-PMIPv6 comparing to PMIPv6. It is important to note that the controllers internal Self-Authentication message does not imply that some authentication protocol or algorithm is incorporated.

### B.2 Mobile Node Handover in Proactive OF-PMIPv6

In the proactive OF-PMIPv6, a mobile node current OMAG (pOMAG) constantly monitors link state of the mobile node and as soon as the values drops below the lower threshold it reports it to the controller. The controller updates the C-DB from the received information. Other OMAGs in the vicinity overhears the communication of the mobile node with pOMAG and reports the mobile node link state to the controller if it is above the higher threshold. Once the controller determines that the mobile node link state reported by some other OMAG is better than the link state of the mobile node with the pOMAG, it initiates the handover process. The handover decision and selection of the nOMAG is presented in the Fig. 4 through the controller internal Handover\_Init message. The controller sends the Tunnel\_Init message to the nOMAG to create the IP tunnel to the anchor. Simultaneously the controller sends the PBU message to the anchor with mobile node ID, HNP and nOMAG ID. The anchor updates the binding entry accordingly and creates the IP tunnel with the nOMAG before sending the PBA message to the controller. Here onwards the anchor forwards the received data packets from the corresponding node to the nOMAG, where the

nOMAG buffer the received packets.

Meanwhile the controller, sends a redirect request message to the mobile node through the pOMAG and starts an expiration timer. On receiving the redirect request message, the mobile node disconnects from the pOMAG and sends a RS message to the nOMAG after association. The nOMAG forwards the RS message to the controller (OF-RS). On receiving the OF-RS message the controller terminates the expiration timer. If the value of expiration timer is elapsed and no OF-RS is received, then the controller resends the redirect request message to the mobile node and restarts the expiration timer. In the meantime controller has received the PBA message from the anchor. Therefore on receiving the OF-RS message, the controller creates a RA message and sends it to the mobile node through the nOMAG (OF-RA). On receiving the OF-RA message nOMAG forwards the RA message to the mobile node and also flushes all the buffered packets to the mobile node. During the proactive handover, correct state of the mobile node is maintained in the C-DB through the finite state machine, similarly the anchor also maintains the state of the mobile nodes to ensure consistent state of the system at any given time. States of the OMAGs are also maintained at the controller. The Fig. 4 presents the aforementioned explained proactive mode of the OF-PMIPv6.

The handover latency and packet loss in the proactive OF-PMIPv6 is considerably less comparing to PMIPv6 or the reactive OF-PMIPv6. The desired reduction in the handover latency is mainly because; in PMIPv6 a gateway is responsible for the layer two connectivity, layer three connectivity, control signaling and data transmission. As all of these are dependent on each other, the gateway cannot introduce parallelism. Whereas in the OF-PMIPv6 the presence of the controller provides a unique opportunity to offload the layer three control signaling from the gateways and enables to perform various control signaling simultaneously, as shown in the Fig. 4.

## IV. ANALYTICAL MODELING

This section presents a detailed analytical model of the PMIPv6, OF-PMIPv6 reactive and proactive schemes for analyzing their performance under different parameters. Different variables that are used in the analytical model, are presented in the Table 1 along with their definitions.

### A. Handover Latency

Handover latency is defined as the time elapsed from the moment mobile node layer two connection is established with the nMAG/nOMAG to the moment when the mobile node receives the RA message. This definition is used to model the handover latency of the PMIPv6, OF-PMIPv6 reactive and proactive.

#### A.1 PMIPv6

Based on the PMIPv6 handover description, the handover latency for PMIPv6 ( $HD_{PMIPv6}$ ) can be calculated as:

$$HD_{PMIPv6} = T_{\text{Const}} + T_A + T_{PU} \quad (1)$$

where  $T_{\text{Const}} = T_{WRS} + T_R$ . The  $T_{WRS}$  is the delay between the mobile nodes layer two connection establishment and

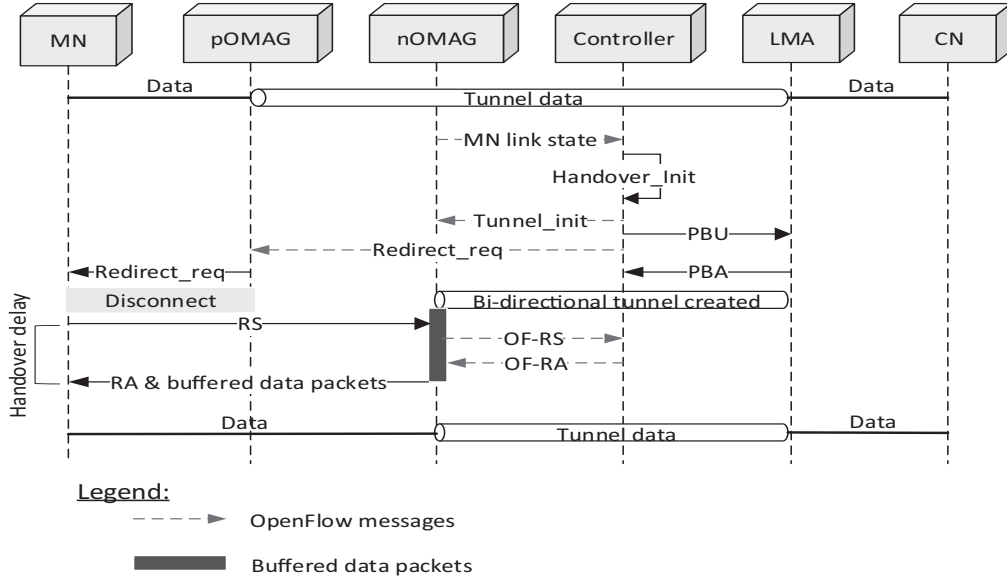


Fig. 4. Proactive OF-PMIPv6 handover signaling call flow.

transmission of the RS message. We have modeled the  $T_{WRS}$  as a random variable distributed normally between the interval  $[0, \text{MAX\_SOLICITATION\_Delay}]$  [14], [15]. The cumulative time taken by the RS and RA message is presented by the  $T_R = T_{RS} + T_{RA}$ . The  $T_{RS}$  and  $T_{RA}$  consists of the wireless channel propagation delay and service delay on the gateway. Hence, the  $T_R$  can be calculated as:

$$T_R = \frac{P_f(W_{MAG} + L_\beta)}{1 - P_f} + \frac{P_f(L_\beta)}{1 - P_f} \quad (2)$$

$$T_R = \frac{P_f(W_{MAG} + 2L_\beta)}{1 - P_f} \quad (3)$$

where  $P_f/(1 - P_f)$  presents the odds that a transmission over the wireless link will fail. The  $W_{MAG}$  is the service delay at the gateway once it receives a message, and we have modeled it using the  $M/M/1$  queues while assuming there is no packet drop during the processing. We have considered the link service delay ( $\mu_p$ ) to be same for all the nodes, and the  $\lambda_{MAG}$  is the packet arrival rate at the gateway. Using the mean wait time of the  $M/M/1$  queues, the  $W_{MAG}$  can be defined as:

$$W_{MAG} = \frac{1/\mu_p}{1 - \rho_{MAG}} \quad (4)$$

where  $\rho_{MAG} = \lambda_{MAG}/\mu_p$ . The processing delay at the mobile node once it receives the RA message is not considered as the part of handover delay, hence the  $T_{RA}$  only depends on the wireless link propagation delay.

In the PMIPv6, the next gateway needs to authenticate the mobile node with the AAA server, before sending the PBU message to the anchor.  $T_A$  is the time taken to authenticate the mobile node with the AAA server, and is defined as  $T_A = T_{AAAreq} + T_{AAARes}$ . Considering there is no transmission failure in the wired network and using the  $M/M/1$  queuing model

the  $T_A$  is calculated as:

$$T_A = (H_{MAG-AAA}(W_p + L_\alpha) + W_{AAA}) + (H_{MAG-AAA}(W_p + L_\alpha) + W_{MAG}) \quad (5)$$

$$T_A = 2H_{MAG-AAA}(W_p + L_\alpha) + W_{AAA} + W_{MAG} \quad (6)$$

There can be multiple network nodes (e.g., switches and routers) present in the network. It is assumed that mean wait time of these network nodes is the same, and is presented as  $W_p$  in (6). The  $W_p$  can be calculated as shown in (4), with the packet arrival rate  $\lambda_p$ . Once the AAA request message arrives at the AAA server, the processing time consumed for the authentication is presented as the  $W_{AAA}$ , and is calculated as shown in (4), with the packet arrival rate  $\lambda_{AAA}$ . After authentication process the AAA server sends back the AAA response message to the gateway. The processing time of the gateway for the AAA response message is presented as the  $W_{MAG}$ .

$T_{PU}$  represents the time taken by the PBU and PBA messages, and it can be defined as  $T_{PU} = T_{PBU} + T_{PBA}$ . Based on (6), the  $T_{PU}$  can be calculated as:

$$T_{PU} = 2H_{MAG-LMA}(W_p + L_\alpha) + W_{LMA} + W_{MAG} \quad (7)$$

where the  $W_{LMA}$  is the mean wait time of the anchor, and can be calculated as shown in (4), with the packet arrival rate  $\lambda_{LMA}$ .

## A.2 Reactive OF-PMIPv6

In the reactive OF-PMIPv6 the handover does not involve the AAA server, as the controller uses the mobile node cached information to authenticate it. Based on the signal call flow in the Fig. 3, the handover delay for the reactive OF-PMIPv6 can be calculated as:

$$HD_{OFPMIP-R} = T_{Const} + T_{OF-R} + T_{OF-PU} \quad (8)$$

In the OF-PMIPv6, gateways are replaced by the OMAGs. In our analytical model we have considered OMAGs to be same

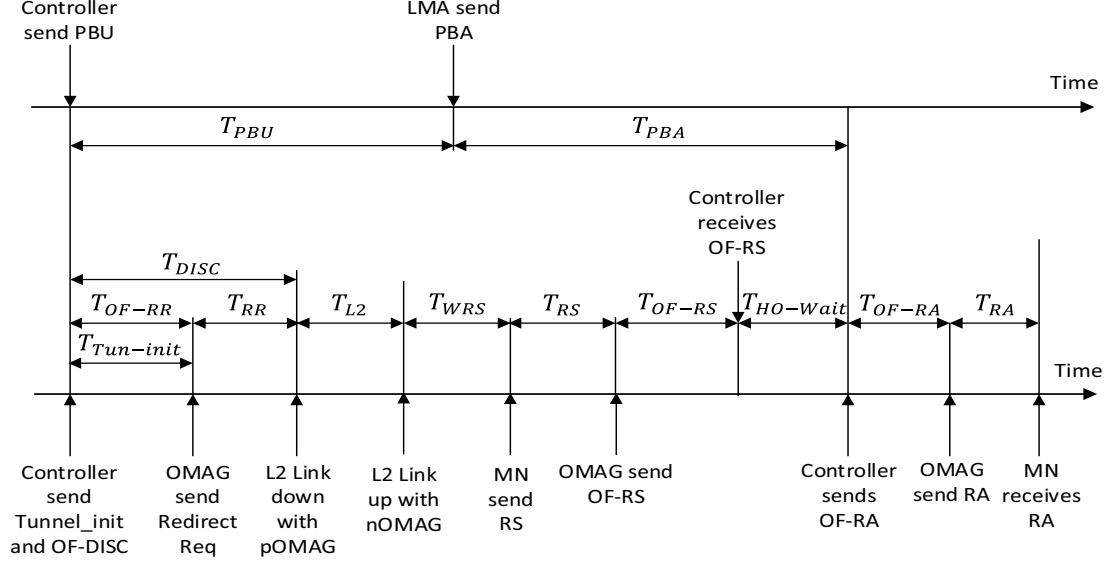


Fig. 5. OF-PMIPv6 proactive handover timing diagram.

as gateways in terms of their processing delay. Hence, the  $T_{Const}$  and the  $W_{MAG}$  in the OF-PMIPv6 remains same as the PMIPv6. The communication delay between the OMAG and the controller is encompassed by the  $T_{OF-R}$ , which can be defined as:  $T_{OF-R} = T_{OF-RS} + T_{OF-RA}$ . Based on (5) the  $T_{OF-R}$  can be calculated as:

$$T_{OF-R} = 2H_{MAG-CON}(W_p + L_\alpha) + W_{CON} + W_{MAG} \quad (9)$$

where the  $W_{CON}$  is the mean wait time of the controller, and can be calculated as shown in (4), with the packet arrival rate  $\lambda_{CON}$ .

In the OF-PMIPv6, the controller is responsible to communicate with the anchor on behalf of the OMAGs. The  $T_{OF-PU}$  represents the communication delay between the controller and the anchor, and can be defined as:  $T_{OF-PU} = T_{OF-PBU} + T_{OF-PBA}$ . Similar to (9), the  $T_{OF-PU}$  can be calculated as:

$$T_{OF-PU} = 2H_{CON-LMA}(W_p + L_\alpha) + W_{CON} + W_{LMA} \quad (10)$$

### A.3 Proactive OF-PMIPv6

Unlike the PMIPv6 and the reactive OF-PMIPv6, in the proactive OF-PMIPv6, a mobile node initiates the handover on the controller directive. The mobile node disassociates with the pOMAG because of the redirect request message from the controller. The handover start point is same as in the PMIPv6 and the OF-PMIPv6 reactive. The handover delay for the proactive OF-PMIPv6 can be calculated as:

$$HD_{OFPMIP-P} = T_{Const} + T_{OF-R} + T_{HO-Wait} \quad (11)$$

As shown in the Fig. 5, the controller in proactive mode communicates with the nOMAG, anchor and a mobile node without waiting for the response from the first. The controller first of all sends the tunnel\_init message to the nOMAG, then sends

the PBU message to the anchor and while waiting for the PBA message it sends the redirect request message to the mobile node through the pOMAG. This simultaneous communication enables us to reduce the handover delay, because when the controller receives the OF-RS message from the nOMAG it have already finished its communication with anchor and right away sends the OF-RA message. However, there can be a case where controller is still waiting for a PBA message from the anchor when it receives an OF-RS message from the nOMAG. This case only occurs when the anchor is located at far off destination with large communication delay with the controller, and in this case the controller takes extra time in dispatching the OF-RA message to the nOMAG. This extra wait time is expressed as the  $T_{HO-Wait}$  in (11) and can be calculated as:

$$T_{HO-Wait} = \text{MAX}[T_{RR-OFRS}, T_{OF-PU}] - T_{RR-OFRS} \quad (12)$$

The time elapsed between the redirect message sent from the controller and the OF-RS message received at the controller is expressed as the  $T_{RR-OFRS}$ . If the  $T_{RR-OFRS}$  is greater than  $T_{OF-PU}$ , it means that the controller will have the OF-RA message ready once it receives the OF-RS message from the nOMAG and in this case the  $T_{HO-Wait}$  will be zero. If the  $T_{OF-PU}$  is greater than the  $T_{RR-OFRS}$ , in this case the  $T_{HO-Wait} = T_{OF-PU} - T_{RR-OFRS}$ . The  $T_{RR-OFRS}$  can be calculated as:

$$T_{RR-OFRS} = T_{DISC} + T_{L2} + T_{WRS} + T_{RS} + T_{OF-RS} \quad (13)$$

where the  $T_{DISC}$  is the delay for redirect request message to reach the mobile node from the controller via the pOMAG, and can be calculated as:  $T_{DISC} = T_{OF-RR} + T_{RR}$ . The delay of redirect request message from the controller to the pOMAG is expressed as the  $T_{OF-RR}$  and is same as the  $T_{OF-RA}$ . The delay from the pOMAG to the mobile node is expressed as the  $T_{RR}$  and is same as the  $T_{RA}$ .

### B. Registration Latency

When a mobile node enters in the PMIPv6 domain, it registers itself with the anchor by sending an RS message to the gateway. The registration latency is defined as the time elapsed from the moment mobile nodes layer two connection is established to the moment when it receives the RA message. In the PMIPv6, the registration latency is similar to the handover latency. Whereas, in the OF-PMIPv6 the registration latency is slightly more than the handover latency, because of the authentication process. The registration latency in the OF-PMIPv6 can be calculated as:

$$RD_{OFPMIP} = HD_{OFPMIP-R} + T_{OF-A} \quad (14)$$

where the  $T_{OF-A}$  is the communication delay between the controller and the AAA server, and can be calculated as:

$$T_{OF-A} = 2H_{CON-AAA}(W_p + L_\alpha) + W_{CON} + W_{AAA}. \quad (15)$$

### C. Buffering Cost

Buffering of the data packets is required to minimize the packet loss which might occur during the mobile nodes handover. The space required for the buffering is referred to as buffering cost. Depending on the scheme in use, buffering of the data packets is generally performed either on the pMAG or nMAG during a handover process, and buffered data packets are forwarded to the mobile node by the pMAG or nMAG upon its attachment to an nMAG.

The standard PMIPv6 [1] does not provide buffering mechanism and so does the OF-PMIPv6 in the reactive mode, because the identity of the nMAG is not known before the handover and there is no communication between the pMAG and the nMAG. However, in the OF-PMIPv6 proactive mode identity of the nMAG is known to the controller and hence the buffering is performed at the nMAG to minimize the packet loss which might occur during the handover. Buffering for a mobile node starts at the nMAG on the completion of the bidirectional tunnel between itself and the anchor, and continues until it receives the OF-RA message from the controller.

The required space for a mobile node at the nMAG is the accumulative size of the data packets received at the nMAG in the time period starting from the arrival of the PBU message at the anchor with the nMAG ID and ending when the nMAG receives the OF-RA message from the controller. For a mobile node proactive handover in the OF-PMIPv6, the buffering cost can be calculated as:

$$BC_{OFPMIP-P} = \lambda_s E(S)(T_{BUF-Wait}) \quad (16)$$

where the  $\lambda_s$  is the average session arrival rate and the  $E(S)$  is the average session length [14]. The  $T_{BUF-Wait}$  is the duration for which data packets are buffered at the nMAG, and it is defined as the difference between the buffering end time and the start time  $T_{BUF-Wait} = T_{BE} - T_{BS}$ . The buffering end and start time can be calculated as:

$$T_{BS} = \text{MAX}[T_{OF-PU}, T_{Tun-init}] \quad (17)$$

$$T_{BE} = \text{MAX}\left[\frac{T_{RR-OFR}}{P_a}, T_{BS}\right] \quad (18)$$

where the  $T_{Tun-init}$  is the delay for the tunnel initiate message from the controller to the nMAG, and is same as the  $T_{OF-RA}$ . The time elapsed between the redirect request message sent from the controller and the OF-RA message received by the nMAG is defined as the  $T_{RR-OFR}$ , and is calculated as:  $T_{RR-OFR} = T_{RR-OFRS} + T_{OF-RA}$ .

The controller selects the nMAG with the highest RSS value at the time of handover initiation. On receiving the redirect request message the mobile node disconnects itself from the pMAG and connects to the nMAG with the highest RSS value and is selected by the controller. In case where multiple candidate nMAGs are available with the same RSS value, the mobile node can attach to any one of them. The probability that the mobile node will attach to the nMAG selected by the controller, when multiple candidate nMAGs are available with the same RSS value is presented by  $P_a$ . Clearly, if the mobile node attaches to the nMAG which is not selected by the controller, then buffering end time at the controller selected nMAG will extend till infinity. Hence, the buffering end time is inversely proportional to the success probability of the mobile node attachment to the controller selected nMAG. The  $P_a$  depends on the number of candidate OMAGs, and it can be calculated as:

$$P_a = \frac{1}{N} \quad (19)$$

where  $N$  is the number of candidate nMAGs.

### D. Packet Loss

During a handover the mobile node stops receiving the packets as soon as it disconnects from the pMAG/pOMAG, and resumes after receiving the RA message from the nMAG/nOMAG. In our packet loss model we have not considered the mobile nodes layer two connection establishment time. In case of no buffering mechanism the packets are lost during the handover. In PMIPv6 and reactive OF-PMIPv6 there is no buffering mechanism, hence their packet loss can be calculated as:

$$PL_{PMIPv6} = \lambda_s E(S) HD_{PMIPv6}, \quad (20)$$

$$PL_{OFPMIP-R} = \lambda_s E(S) HD_{OFPMIP-R}. \quad (21)$$

In the proactive OF-PMIPv6 buffering is performed at the nMAG, however packet loss can occur if: 1) As a result of redirect request message, the mobile node disconnects from the pMAG before the IP tunnel at the anchor is updated to the nMAG, 2) the data packets reach the nMAG through the tunnel but are not buffered because the communication delay for tunnel initiate message is more than  $T_{OF-PU}$ . Once the buffering starts at the nMAG there is no further packet loss. When the nMAG receives the OF-RA message from the controller, it sends all the buffered data packets along with the RA message to the mobile node and starts forwarding the data packets to the mobile node without buffering, therefore the packet loss for the proactive OF-PMIPv6 can be calculated as:

$$PL_{OFPMIP-P} = \lambda_s E(S)(T_{TUD} + T_{TID}). \quad (22)$$

Where the  $T_{TUD}$  is the tunnel update delay at the anchor and represents the above mentioned case one for the packet loss. The

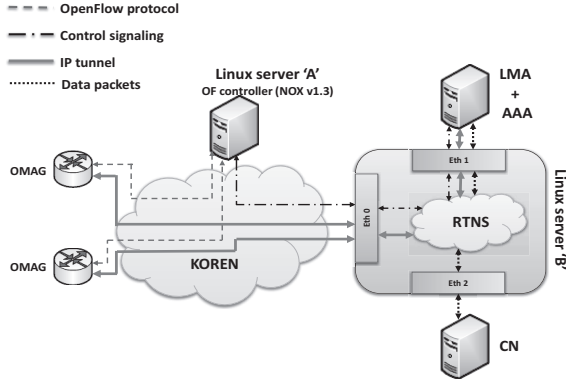


Fig. 6. OF-PMIPv6 Testbed.

tunnel initiate delay at the nOMAG for the above mentioned case two is expressed as the  $T_{TID}$ . Both  $T_{TUD}$  and  $T_{TID}$  can be calculated as:

$$T_{TUD} = \text{MAX}[T_{DISC}, T_{OF-PU}] - T_{DISC}, \quad (23)$$

$$T_{TID} = \text{MAX}[T_{Un-init}, T_{OF-PU}] - T_{OF-PU}. \quad (24)$$

As both the cases for packet loss does not depend on the association of the mobile node with the nOMAG, therefore the probability  $P_a$  does not apply on the packet loss.

## V. IMPLEMENTATION

To evaluate the performance of the proposed schemes, an integrated PMIPv6 and OpenFlow testbed is developed to advocate the feasibility, practicality and benefits of the OF-PMIPv6 in the realistic situations of the production networks. The Fig. 6 presents the architecture of the OF-PMIPv6 testbed. The OpenFlow version 1.3 [13] is used in the OF-PMIPv6 testbed, because of its comprehensive support for IPv6. An open source implementation by the Open Air Interface [16], is used for the PMIPv6 module in the controller and the anchor.

### A. OMAG Implementation

The testbed currently has five OMAGs which are connected to the KOREN (Korean advance research Network), which is an international research collaboration network, spreading over multiple countries and continuously growing [17]. As presented in the section 4.2.1 the OMAG is required to implement partial functionalities of the PMIPv6 gateway, a complete OF protocol 1.3 and control messages which are not supported by the OF protocol. To suffice these requirements open source router implementation for embedded device (OpenWRT) [18] is deployed on the Alix 2d2 development kit [19]. An open source implementation of the OF protocol v1.3 for the switches [20] is utilized on the OMAGs, and is extended through the experimenter message type, to support the OF-PMIPv6 control signaling. The developed tunnel management module manages and creates an IP in IP tunnel with the anchor and makes appropriate entries in the routing table (in Linux kernel), through which

user data is exchanged between the mobile node and the anchor. A link state monitoring module is developed in the user space of the OMAG to sample the RSS values of all the nearby mobile nodes by utilizing the monitor mode of the IEEE 802.11 for a predefined sample duration. The collected link state information of the mobile nodes is sent to the controller using the S\_report type experimenter message.

### B. OpenFlow Controller Implementation

The controller is a dedicated Linux server with 32 GB memory, 2.5 TB hard disk and Intel Xeon CPU (3.10 GHz quad core). The controller is connected to the KOREN and serves as a central control entity in the OF-PMIPv6. The open source implementation of the NOX supporting OF protocol v1.3 [20] is used for the controller. The underlying library functions of the NOX are used to perform all the communication with the OMAG either through the standard OF protocol message types or the experimenter message type. The OpenFlow module in the controller subscribes with the event handlers in the NOX to receive the standard and experimenter OF protocol messages from the OMAGs. The required standard OF protocol and experimenter messages are created in the OpenFlow module through the NOX library functions. To send the standard and experimenter OF protocol messages to the OMAG, the OpenFlow module utilizes the API functions exposed by the NOX. The mobility management module is developed as part of the OpenFlow module, and is used to store the OMAGs and mobile nodes information, as well as to maintain the mobile node state during the reactive or proactive handovers. The PMIPv6 module is used to communicate with the anchor. The Role of the PMIPv6 module is to generate the PMIPv6 control messages from the information provided by the OpenFlow module, and send them to the anchor. Also, it provides the information to the OpenFlow module based on the response received from the anchor. The PMIPv6 module executes as a separate process and communicates with the OpenFlow module through IPC.

State of the mobile node is maintained as not registered, already registered or updated registration through the C-DB. The OpenFlow module performs the mobile node state transition on receiving the OF-PMIPv6 protocol messages.

**NOT REGISTERED:** It occurs when a mobile node connects to the OF-PMIPv6 domain and no prior entry exists in the C-DB. In response, the OF module creates a new entry for the mobile node in the C-DB after the authentication with the AAA server, and via the PMIPv6 module communicate with the anchor in order to establish the IP in IP tunnel. On receiving an acknowledgement from the anchor, the PMIPv6 module responds to the OF module with an RA message, which is then sent to the mobile node through the OMAG.

**ALREADY REGISTERED:** It occurs when the controller receives the RS message from the current OMAG of the already registered mobile node. In response, the controller simply re-sends the RA message to the mobile node via its current OMAG, as it already has the RA message in the C-DB.

**UPDATE REGISTRATION:** If an entry for the mobile node already exists in the C-DB for which the RS message is received, but the ID of the OMAG in the C-DB is not similar to the ID of the OMAG which has sent the OF-RS message, then this is

Table 1. Notation.

Variables	Description	Values
$H_{MAG-AAA}, H_{MAG-LMA}, H_{MAG-CON}$	Number of hops between MAG and AAA server, LMA, Controller, respectively	[1]–[19]
$H_{CON-LMA}, H_{CON-AAA}$	Number of hops between Controller and LMA, AAA server, respectively	[1]–[19]
$\mu_p$	Link service rate	10 Mbps
$\lambda_p$	Packet arrival rate at the network nodes e.g., switches and routers	1–9 Mbps
$\lambda_{MAG}, \lambda_{LMA}, \lambda_{AAA}, \lambda_{CON}$	Packet arrival rate at MAG, LMA, AAA and Controller, respectively	1–9 Mbps
$\lambda_s$	Average session arrival rate	0.7
$E(S)$	Average session length	20
$\rho_p$	Utilization of network nodes e.g., switches and routers	0.1–0.9
$\rho_{MAG}, \rho_{LMA}, \rho_{AAA}, \rho_{CON}$	Utilization of MAG, LMA, AAA and Controller, respectively	0.1–0.9
$L_\beta$	Wireless link propagation delay	15 ms
$L_\alpha = d/\omega$	Wired link propagation delay, where distance ( $d$ )=1500 m and link speed ( $\omega$ )= $2 \times 10^8$ m/s	0.0075 ms
$W_p$	Mean wait time of the network nodes e.g., switches and routers	–
$W_{MAG}, W_{LMA}, W_{AAA}, W_{CON}$	Mean wait time of the MAG, LMA, AAA and Controller, respectively	–
$P_f$	Failure probability of wireless link	0.3
$P_a$	Success probability of a mobile node attachment with nOMAG	–
$N$	Number of candidate nOMAGs	2
$T_{L2}$	Time delay for establishing the layer two connection	450 ms
$T_{WRS}$	Time from established layer two connection to dispatch of first RS message	220 ms

considered as the UPDATE REGISTRATION case. This case occurs when the mobile node handover from one OMAG to the other. In response, the information of the new OMAG is updated in the C-DB, the PMIPv6 module communicates with the anchor in order to update the IP in IP tunnel, and meanwhile the OF module sends the OF-RA message (which is already in the C-DB) and Tunnel-Init message to the new OMAG.

### C. Real Time Network Simulator (RTNS)

The OF-PMIPv6 testbed aims to provide a realistic production network environment in which the anchor is usually present in the mobile nodes home network along with the AAA server, and a corresponding node can be anywhere in the Internet. To depict this we have developed a Real Time Network Simulator (RTNS) in NS-3, which emulates the background traffic and network delay as experienced in the real network environment. The motivation behind the development of the RTNS module is to show the effectiveness of the proposed OF-PMIPv6 architecture under realistic scenarios, however it is not the integral part of the OF-PMIPv6 architecture of the OF-PMIPv6 testbed.

A virtual network in the RTNS connects to the outside world through a TAP and Emu NetDevice of the NS-3. An Emu NetDevice connects to the real Ethernet card of the server [21]. For an Emu NetDevice to receive the packets from the outside of NS-3, the real Ethernet card is set to promiscuous mode and the

Ethernet device name (e.g., eth0) is provided to the Emu NetDevice. NS-3 utilizes the MAC addresses of the interfaces to distinguish between the real and emulated interfaces [22]. To avoid the occurrence of packet duplication, the IP forwarding must be disabled in the Linux system. In the RTNS, there are two border nodes and each border node has two interfaces, one interface is Emu Netdevice to connect to the real Ethernet card, and the other interface is to connect to the virtual network in the RTNS. A border node receives a packet on its Emu NetDevice interface from the Ethernet card and forwards it to the virtual network through its other interface. In the virtual network the packets are routed to the other border node which forwards them outside to the real Ethernet card.

The RTNS runs on a dedicated Linux server, attached to the KOREN. The Linux server consists of 8 GB memory, 2.5 TB hard disk and an Intel CPU with quad cores where each core speed is 3.10 GHz. As presented in the Fig. 6, the control messages from the controller to the anchor and vice versa goes through the RTNS in order to introduce the real network delay. Similarly the IP in IP tunnel between the OMAG and the anchor goes through the RTNS, so the data packets experience the same delay as the control messages do. The anchor strips off the tunnel IP header from the data packets received through the IP in IP tunnel and forwards them back to the RTNS and this time the RTNS forwards the data packets to the correspond-

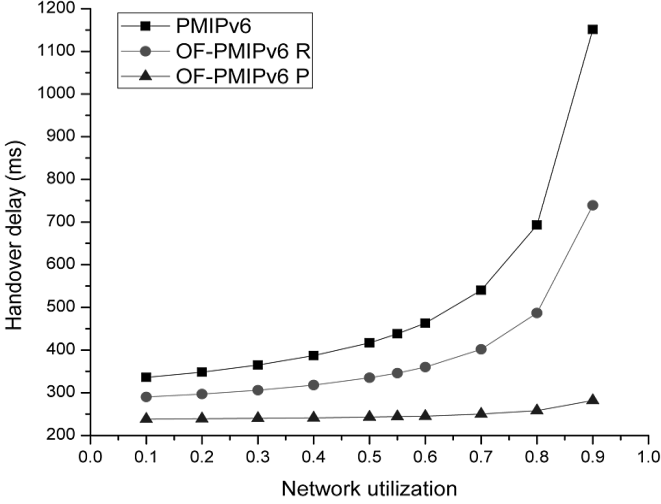


Fig. 7. Handover latency comparison with increasing utilization of network elements and PMIPv6/OF-PMIPv6 nodes.

ing node after introducing a corresponding delay, and the same is performed in the reverse direction. In other words the RTNS works as a router, but also introduces the delay in the packets according to different realistic scenarios.

## VI. PERFORMANCE EVALUATION

This section discusses the performance evaluation of the OF-PMIPv6 based on the analytical model and experimental results from the testbed.

### A. Analytical Evaluation

Analytical evaluation between the PMIPv6 and the OF-PMIPv6 is drawn to show the performance gains of the reactive and proactive schemes in OF-PMIPv6 over the PMIPv6. The values of parameters presented in the Table 1 are used in the evaluation, unless stated otherwise [3], [23], and [24].

The layer two connection delay is heavily dependent on the hardware of wireless card in the mobile node and the OMAG [25], as different manufacturers follow different message sequences and development techniques. For our analysis we have considered the value of  $T_{L2}$  averaged from the results presented in [7], [25], and [26]. In all the performed evaluations, the AAA server is considered to coexist with the anchor, hence the authentication delay is considered same as the binding update delay.

#### A.1 Handover Latency Comparison

Because of city wide WiFi deployments and exponentially increasing usage of the data over the mobile networks, the traffic load on the networks is increasing which also affects the handover performance. Based on the values in the Table 1 and in (1), (8), and (11), an analytical performance comparison for the handover latency in the PMIPv6, reactive OF-PMIPv6 and proactive OF-PMIPv6 is presented in the Fig. 7, as a function of network utilization. A mobile node is considered to be in a foreign network and the controller is present in the same backbone as the current OMAG, hence the number of MAG/Controller

to anchor hops and OMAG to Controller hops are set to be 19 and 1, respectively. The results in the Fig. 7 show that the PMIPv6 handover latency exponentially increase with the increase in network utilization. The reactive OF-PMIPv6 follows the same trend as the PMIPv6 but has 50% less handover latency. The proactive OF-PMIPv6 has a slight increase in the handover latency because the communication delay with the anchor ( $T_{OF-PU}$ ) becomes greater than  $T_{RR-OFRS}$  in (12), hence the controller has to wait extra time before sending the OF-RA message to the OMAG.

A distinguishing characteristic of the proposed OF-PMIPv6 is its capability to sustain under the high network traffic load and communication delays, where the PMIPv6 performance degrades rapidly. The relative handover performance gain of the OF-PMIPv6 over the PMIPv6 is calculated as [15]:

$$PG_{OFPMIP-R} = \frac{HD_{PMIPv6}}{HD_{OFPMIP-R}}, \quad (25)$$

$$PG_{OFPMIP-P} = \frac{HD_{PMIPv6}}{HD_{OFPMIP-P}}. \quad (26)$$

Using (25) and (26), consolidated handover performance gain of the proposed OF-PMIPv6 reactive and proactive schemes against the increasing network utilization and the anchor distance from the gateway/controller, is shown in the Figs. 8(a) and 8(b), respectively. The increase in performance gain of the reactive OF-PMIPv6 is because of the cached mobile node information in the controller, and for the proactive OF-PMIPv6 the increasing performance gain is the result of the preemptive handover decision by the controller and in advance control signaling. It can be inferred from the Figs. 8(a) and 8(b) that under realistic network scenarios the proposed OF-PMIPv6 shows high handover performance gains over the PMIPv6.

#### A.2 MN Registration Latency Comparison

The MN registration process is similar in the both OF-PMIPv6 reactive and proactive mode, and has slightly higher latency than the standard PMIPv6 because of an extra control step in form of the controller and the required authentication with the AAA server. Based on (14) the difference in registration latencies of the OF-PMIPv6 and the standard PMIPv6 is calculated to be approx. 0.04sec and it remains constant for any value of communication delay with the anchor. As the mobile node registers only once upon its entry in the PMIPv6 or OF-PMIPv6 domain, therefore this increase in delay is insignificant.

#### A.3 Packet Loss Comparison

The packet loss comparison is drawn between the PMIPv6 and the proposed OF-PMIPv6 (reactive and proactive modes) based on (22), (23), and (24), respectively. The Fig. 9 shows the packet loss comparison results as a function of network utilization. The average session length ( $E(S)$ ) at the mobile node is set to be 20 packets, and the average session arrival rate ( $\lambda_s$ ) is considered as 0.7 sessions per second. Results in the Fig. 9 are measured when the number of hops between the gateway/controller and the anchor are set to be 19. The packet loss for the OF-PMIPv6 reactive and the standard PMIPv6 is directly propor-

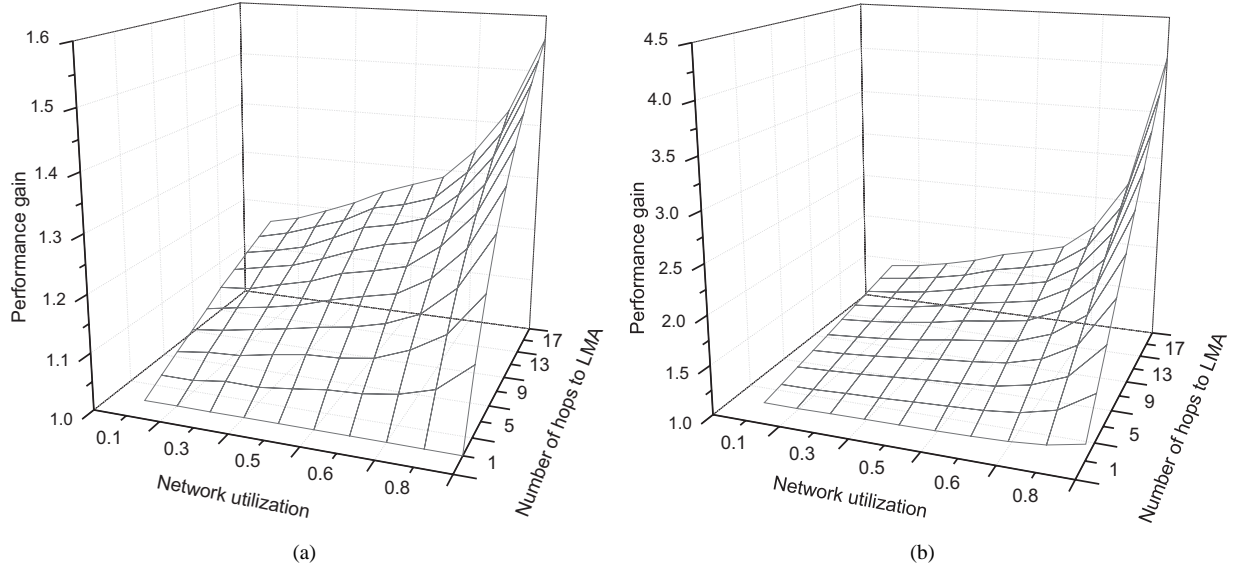


Fig. 8. OF-PMIPv6 relative handover performance gain over PMIPv6: (a) Reactive OF-PMIPv6 handover performance gain against network utilization and distance from the anchor and (b) proactive OF-PMIPv6 handover performance gain against network utilization and distance from the anchor.

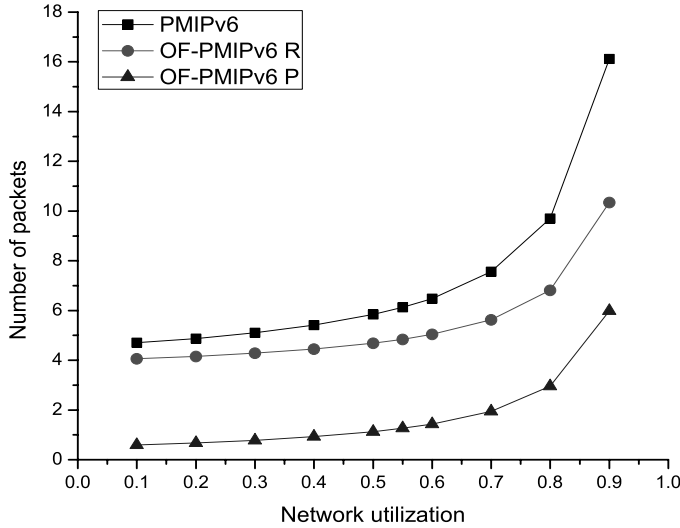


Fig. 9. Packet loss comparison results with increasing utilization of network elements and PMIPv6/OF-PMIPv6 nodes.

tional to the handover latency as there is no buffering mechanism, therefore the PMIPv6 suffers with the maximum amount of packet loss as its handover latency is the largest. In the Fig. 9 proactive OF-PMIPv6 show a gradual increase in the packet loss because of the early arrival of the redirect request message at the mobile node, whereas the PBU message to the anchor is delayed due to the multiple hops with high background traffic.

#### A.4 Control Signaling Comparison

The control signaling overhead of the proposed schemes is defined as the number of extra bytes transmitted during the control signaling, comparing to the control signaling in the standard PMIPv6 handover. The Table 2 lists the control messages and their sizes, which are transmitted in the PMIPv6 and in the proposed OF-PMIPv6 reactive and proactive schemes. The con-

trol message sizes mentioned in the Table 2 are based on the traces obtained from the testbed. The AAA protocol used in the PMIPv6 is RADIUS. The AAA request/response message size is based on the access request and response message mentioned in the RFC 2865 [27]. Based on the Table 2, the number of bytes transmitted during the control signaling in the PMIPv6 handover are 706 bytes, whereas in the reactive and proactive OF-PMIPv6 handover transmitted control messages bytes are 1190 and 1362, respectively. For the proactive OF-PMIPv6, the link status message sent by the OMAG is not considered as part of the handover process as it is not a handover trigger. A handover is initiated by the controller, when it decides that better links are available for a mobile node. A link state message is sent to deliver the mobile node link status value to the controller, and only when an event is occurred on the OMAG. Size of a link state message is approx. 300 bytes.

#### B. Experimental Evaluation

In the experimental evaluation we present the results for mobile node handover latency in the reactive OF-PMIPv6 scheme. The testbed is set up as described in the Fig. 6. The pOMAG and nOMAG are placed in a way that there is minimum footprint overlap between the two OMAGs. A person holds the mobile node and moves from one OMAG to the other on a predefined path with walking speed. Initially the mobile node connects to the pOMAG and registers its self to the OF-PMIPv6 domain. As the person walks away from the pOMAG and enters in footprint of the nOMAG, the mobile node disconnects from the pOMAG and connects to the nOMAG. The handover latency of the mobile node in the reactive OF-PMIPv6 is calculated as the time difference between the RS and RA messages sent and received by the mobile node, respectively. Delay between the layer two connection establishment and transmission of RS message ( $T_{WRS}$ ) is not considered while collecting the handover results, because the value for  $T_{WRS}$  is implementation dependent and differs for different operating systems and ven-

Table 2. Control messages and their sizes in PMIPv6 and OF-PMIPv6.

PMIPv6		Reactive OF-PMIPv6		Proactive OF-PMIPv6	
Message	Size	Message	Size	Message	Size
RS	70 bytes	RS	70 bytes	PBU	156 bytes
AAArequest	120 bytes	OF-RS	254 bytes	Tunnel-init	266 bytes
AAAresponse	120 bytes	OF-RA	204 bytes	Redirect Req	172 bytes
PBU	156 bytes	Tunnel-init	266 bytes	PBA	130 bytes
PBA	130 bytes	PBU	156 bytes	RS	70 bytes
RA	110 bytes	PBA	130 bytes	OF-RS	254 bytes
		RA	110 bytes	OF-RA	204 bytes
				RA	110 bytes

dors. However, to make the testbed results consistent with the analytical model, value of  $T_{WRS}$  from the table 1 (220 ms) is added in each testbed handover latency result. The reactive OF-PMIPv6 handover results from, the testbed and the analytical model, are presented in the Fig. 10. The testbed results for the reactive OF-PMIPv6 handover are collected over a period of one week. For each day of the week 50 experiments are performed at the similar time, and their average is considered as the handover latency for that day. The analytical model based reactive OF-PMIPv6 handover latency in the Fig. 10 considers, single hop between the OMAGs and the controller and four hops between the controller and the anchor, with the utilization for all the OF-PMIPv6 nodes and the network node is 0.5. The results in the Fig. 10 show that testbed results for the reactive OF-PMIPv6 are close to the results from the analytical model. At the time of experiment, on the day four, much activity in the environment caused the dynamic and lossy wireless channel. Therefore, the testbed based handover results from day four are much closer to the analytical model results when the wireless link failure probability is 0.7.

It is worth mentioning that a mobile node handover latency in the OF-PMIPv6 testbed is not calculated from the time it disconnects from the pOMAG. This is because the disconnection from the pOMAG and decision to connect to the nOMAG is done by network manager in the mobile node operating system. During the experiments it was found out that network manager in different operating systems may work in different manner and that the network manager in the Windows 7 OS connects to the nOMAG most promptly after disconnecting from the pOMAG. However the network manager in Windows 7 OS stays connected to the pOMAG even when the received signal strength is not sufficient for transmitting/receiving any data. This phenomenon is presented in the Fig. 11. From the Fig. 11 it is evident that the goodput at the mobile node drops to zero way before the handover is triggered by the network manager in the Windows 7 OS. In the Fig. 11, a mobile node receives no data after 30 s till 68.5 s causing huge amount of packet loss, whereas the layer three handover initiates at the 67 s. To provide the seamless mobility and disruption free realtime services, it is necessary that the handover at the mobile node is triggered before it moves out of the effective range of the OMAG. Result in the Fig. 11 advocates the need and importance of our proposed proactive OF-

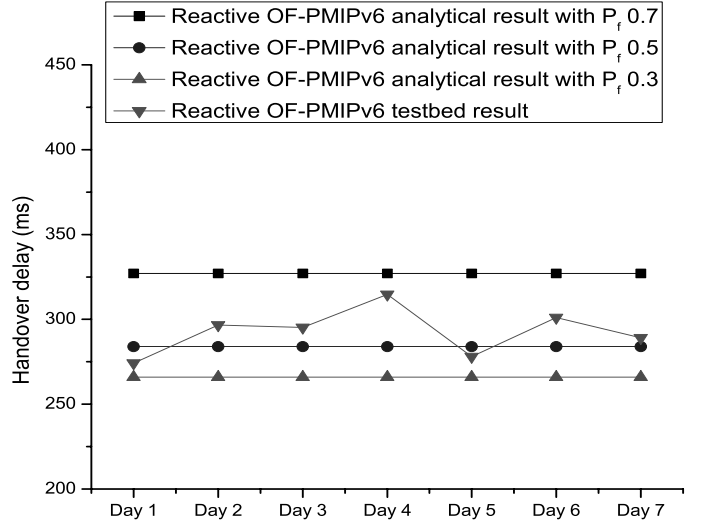


Fig. 10. Reactive OF-PMIPv6 handover latency in the testbed and its comparison with the analytical model based results.

PMIPv6 handover scheme, where handover is triggered by the network instead of a mobile node. It is important to note that the Fig. 11 shows the result from single experiment and the throughput and service disruption period values change with every trial.

## VII. CONCLUSION AND FUTURE WORK

A proposed OpenFlow supported PMIPv6 architecture (OF-PMIPv6) is presented in this paper, which separates the control path from the data path and centralizes the control at the controller. Although, an extra step of control signaling is introduced in the OF-PMIPv6, but the results presented in this paper show that its effects are minimal on the handover latency and packet loss. Benefits of the OF-PMIPv6 are discussed along with a detailed explanation of the architecture and specific functionalities of the different components. Based on the OF-PMIPv6 architecture, reactive and proactive handover schemes are presented in this paper, which achieve 43% and 121% improvement over the standard PMIPv6 in terms of handover latency and 46% and 90% improvement in terms of packet loss. Brief description of the OF-PMIPv6 testbed and initial results from it are also in-

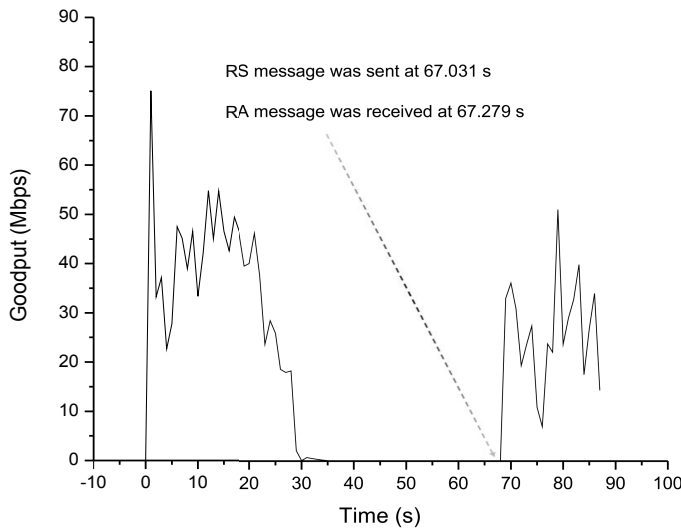


Fig. 11. The goodput at a mobile node during the reactive OF-PMIPv6 handover.

cluded in this paper. In future, the proactive OF-PMIPv6 handover scheme will be implemented on the OF-PMIPv6 testbed, and results will be collected while using RTNS module to create realistic production network environment. We plan to implement different schemes on our testbed to do performance comparison with OF-PMIPv6. Finally the OF-PMIPv6 protocol, architecture and testbed will be extended to support the vertical handovers.

## REFERENCES

- [1] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy mobile IPv6," RFC 5213, Aug. 2008.
- [2] K. Kim, H. Lee, H. Choi, and Y. Han, "Efficient buffering scheme in the LMA for seamless handover in PMIPv6," *IEICE Trans. Commun.*, vol. 95, no. 2, pp. 382–391, Feb. 2012.
- [3] S. Oh and H. Choo, "Low latency handover scheme based on optical buffering at LMA in proxy MIPv6 networks," in *Proc. ICCSA*, July 2009.
- [4] H. Choi, K. Kim, H. Lee, S. Min, and Y. Han, "Smart buffering for seamless handover in proxy mobile IPv6," *J. Wireless Commun. Mobile Comput.*, vol. 11, no. 4, pp. 491–499, Apr. 2011.
- [5] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast handover for proxy mobile IPv6," IETF RFC 5949, Sept. 2010.
- [6] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," in *Proc. ACM SIGCOMM*, Apr. 2008.
- [7] M. Siksik, H. Alnuweiri, and S. Zahir, "A detailed characterization of the handover process using mobile IPv6 in 802.11 networks," in *Proc. PACRIM*, Aug. 2005.
- [8] C.E. Perkins and D.B. Johnson, "Mobility support in IPv6," in *Proc. ACM MOBICOM*, Nov. 1996.
- [9] R. Koodli, "Fast handovers for mobile IPv6," RFC 4068, July 2005.
- [10] P. Tan, "Recommendations for archiving seamless IPv6 handover in 802.11 networks," IETF draft-paultan-seamless-ipv6-handoff-802-00.txt, Mar. 2003.
- [11] J. McNair, I.F. Akyildiz, and M.D. Bender, "Handoffs for real-time traffic in mobile IP version 6 networks," in *Proc. IEEE GLOBECOM*, Nov. 2001.
- [12] S. Jeon, N. Kang, and Y. Kim, "Enhanced predictive handover for fast proxy mobile IPv6," *IEICE Trans. Commun.*, Nov. 2009.
- [13] OpenFlow specifications, [Online]. Available: <https://www.opennetworking.org/sdn-resources/onf-specifications>
- [14] J.H. Lee, Z. Yan, and I. You, "Enhancing QoS of mobile devices by a new handover process in PMIPv6 networks," *Wireless Pers. Commun.*, vol. 61, no. 4, pp. 591–602, Dec. 2011.
- [15] J.H. Lee, T. Ernst, and N. Chilamkurti, "Performance analysis of PMIPv6 based network mobility for intelligent transportation systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 74–85, Jan. 2012.
- [16] Open source implementation of PMIPv6, [Online]. Available: <http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6>
- [17] Korean Advanced Research Network, [Online]. Available: <http://www.koren.kr/koren/eng/>
- [18] Open source router kernel for embedded devices, [Online]. Available: <https://openwrt.org/>
- [19] Alix development boards by PCEngine, [Online]. Available: <http://www.pcengines.ch/alix.htm>
- [20] Implementation of OpenFlow v1.3 for controller and switch, [Online]. Available: <https://github.com/CPqD/>
- [21] NS-3 interaction with the real world, [Online]. Available: [https://www.nsnam.org/wiki/HOWTO\\_make\\_ns-3\\_interact\\_with\\_the\\_real\\_world](https://www.nsnam.org/wiki/HOWTO_make_ns-3_interact_with_the_real_world)
- [22] Emulation in NS-3, [Online]. Available: <https://www.nsnam.org/docs/models/html/fd-net-device.html#emufdnetdevicehelper>
- [23] C. Makaya and S. Pierre, "An analytical framework for performance evaluation of IPv6-based mobility management protocols," *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 972–983, Mar. 2008.
- [24] S. Jeon, N. Kang, and Y. Kim, "Resource-efficient network mobility support in proxy mobile IPv6 domain," *AEU Int'l J. Electron. Commun.*, vol. 66, no. 5, pp. 390–394, May 2012.
- [25] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM Comput. Commun. Rev. Newslett.*, vol. 33, no. 2, pp. 93–102, Apr. 2003.
- [26] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *Proc. IEEE ICC*, Apr. 2003.
- [27] C. Rigney, S.W. Livingston, A.R. Merit, and W.S. Daydreamer, "Remote authentication dial in user service (RADIUS)," RFC 2865, June 2000.
- [28] C.M. Mueller and O. Blume, "Network-based mobility with proxy mobile IPv6," in *Proc. IEEE PIMRC*, Sept. 2007.
- [29] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," RFC 3775, July 2004.
- [30] J.H. Lee and T. Ernst, "Lightweight network mobility within PMIPv6 for transportation systems," *IEEE Syst. J.*, vol. 5, no. 3, pp. 352–361, Sept. 2011.
- [31] G. Jo, H.J. Choe, and H. Choo, "Predictive handover scheme using mobility history in PMIPv6," in *Proc. ACM RACS*, Oct. 2013.



**Syed M. Raza** received the B.S. degree in Computer Information Sciences from Pakistan Institute of Engineering and Applied Sciences, and the M.S. degree in Wireless Communication from Lund University, Lund, Sweden in 2006 and 2009, respectively. During his master degree, from 2008 to 2009, he enjoyed working in Ericsson EuroLabs Aachen as an intern student. In 2011, he joined the COMSATS University, Islamabad, Pakistan as a Lecturer in the Computer Science Department. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, South Korea. His current research interests include software defined networking, IP mobility across heterogeneous networks, software defined wireless, and cellular networks.



**Dongsoo S. Kim** received the M.S. degree in Computer Science from the University of Texas at Dallas, TX, USA, in 1994, and the Ph.D. degree in Computer Science and Engineering from the University of Minnesota, Minneapolis, MN, USA, in 1998. Dr. Kim worked as a Research Scientist for Electronics and Telecommunications Research Institute from 1986 to 1992, and as a project leader for Megaxess Inc. from 1998 to 2000. In 2000, he joined the Department of Electrical and Computer Engineering, Indiana Univ. Purdue Univ. Indianapolis, USA. He has joined with the Department of Electrical and Computer Engineering of Sungkyunkwan University in 2013–2015. He is currently an Associate Professor of ECE, IUPUI. His research includes switch networks, optical switches, network survivability, protection switching, network planning, QoS provisioning in the Internet, mobile ad-hoc networks, mobility of wireless networks, sensor networks, power-aware routing, Software-Defined Networks (SDN), bio-inspired networks, and complex network analysis.



**DongRyeol Shin** received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Sungkyunkwan University in 1980, the Korea Advanced Institute of Science and Technology (KAIST) in 1982, and the Georgia Institute of Technology in 1992, respectively. During 1992-1994, he had worked for Samsung Data Systems, Korea, where he was involved in the research of Intelligent Transportation Systems. Since 1994, he has been with the Department of Computer Science and Engineering at Sungkyunkwan University where he is currently a

Professor in Network Research Group. His current research interests lie in the areas of mobile network, ubiquitous computing, cloud computing, and bioinformatics.



**Hyunseung Choo** received the M.S. in Computer Science from the University of Texas at Dallas in 1990 and the Ph.D. from the University of Texas at Arlington in 1996. He is currently a Professor in College of Information and Communication Engineering and Director of the Convergence Research Institute, at Sungkyunkwan University, Korea, which he joined in 1998. From 2005 to 2013, he had been the Director of the Intelligent HCI Convergence Research Center supported by the Korean government. Since 2013, he has been an Advising Professor of Samsung Electronics.

Dr. Choo has been Editor-in-Chief of the Journal of Korean Society for Internet Information for three years, Journal Editor of ACM Transactions on Internet Technology, Journal of Supercomputing, and Founding Editor of Transactions on Internet and Information Systems since 2010. He is a Member of the ACM, IEEE, and IEICE. He has published over 350 papers in international journals and conferences, and his current research interests include embedded networking, mobile computing, and clouds.