Performance Evaluation of Handover Protocols in Software Defined Networking Environment

Dong-Ru Lee[1], Shan Jaffry[2], Syed Faraz Hasan[2], Yaw-Wen Kuo[1], and Xiang Gui[2](✉)

[1] Electrical Engineering, National Chi Nan University, Puli, Taiwan
ywkuo@ncnu.edu.tw
[2] School of Engineering and Advanced Technology, Massey University, Palmerston North, New Zealand
{S.jaffry,F.hasan,X.gui}@massey.ac.nz

Abstract. Smooth handovers of Mobile Nodes (MN) between different points of attachments to the network ensures seamless connectivity when users move from one place to another. In IP based wireless networks, Mobile IPv6 (MIPv6) protocol manages handover issues, which has evolved with time to take many different forms. In this paper, we explore the softwarization of wireless networks with special regard to mobility management. We modify the legacy mobility management protocols to accommodate Software Defined Networking (SDN) features and measure their performance using simulations. The considered performance indicators include time to complete the handover process and the number of steps required for the same.

Keywords: Mobile IPv6 · Software Defined Networking · Handover Mobility

1 Introduction

Global Internet data transfer is largely enabled by a protocol called Internet Protocol (IP). The identifiers used by this protocol are called IP addresses. A mobile node (MN) will have to change its IP address as it changes the point of connection to the network. In order to enable seamless connectivity for MNs, a new protocol called Mobile IP (MIP) was proposed by Internet Engineering Task Force (IETF) [1]. With the introduction of IPv6 on the network layer, the mobile versions of the protocol also evolved as MIPv6 [2].

In a typical wireless network, a MN is connected to an Access Point (AP) over the radio interface. This AP is the first Point of Attachment (PoA) for the MN, which also acts as its gateway to the Internet. The APs are typically meant for indoor environments and have limited communication range (typically 200–300 m). Multiple APs are linked together to extend the range of a wireless

P. H. J. Chong et al. (Eds.): SmartGIFT 2018, LNICST 245, pp. 240–249, 2018.
https://doi.org/10.1007/978-3-319-94965-9_24

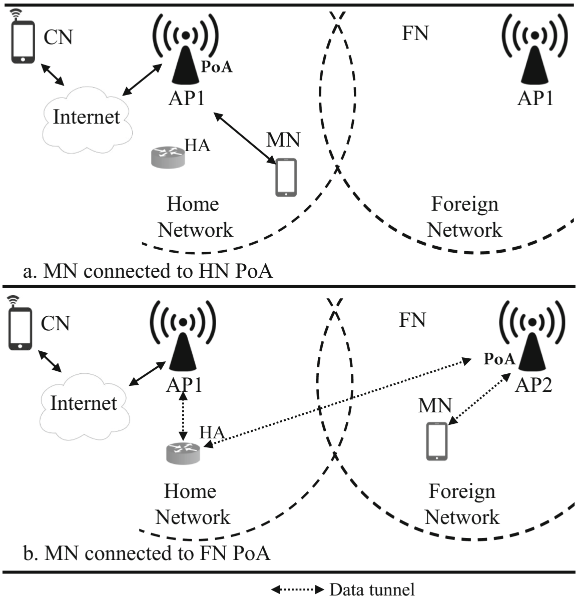

Fig. 1. An example of MIPv6 packet delivery in HN and FN

network. Due to mobility, a MN may change its PoA while still maintaining the connection via the next PoA. This process is called handover (see Fig. 1).

While multiple solutions have been proposed for handover management, they all lack adaptability due to the legacy hardwired infrastructure. However, the recent advances in Software Defined Networking (SDN) have provided opportunities for adaptable network deployment [3–5]. For example, the authors in [6] have shown that SDN can be programmed to support IP mobility, along with re-routing shortest path between a MN and its intended destination called the Corresponding Node (CN). The authors have shown that the hop-counts required in handover and the associated signaling overhead can be reduced using SDN. On the other hand, the authors in [3] have used SDN to realize intra-technology handover. In a previous work, the authors have proposed an SDN based implementation for Hierarchical MIPv6 (HMIPv6) protocol in [7]. Basu et al. have presented a framework for software enabled mobility management in [8] by distributing main controller's functionalities to a set of multiple sub-controllers. The distributed architecture is aimed to reduce signaling overhead in making local and global handover decisions at different controllers.

In this paper, we evaluate the performance of different variants of the legacy MIPv6 protocol with SDN. The performance parameters considered are signaling steps and handover latency. This paper is organized as follows. Section 2 discusses mobility protocols. Section 3 explains SDN and its underlying functions. SDN-enabled simulation is presented in Sect. 4 and conclusions are presented in Sect. 5.

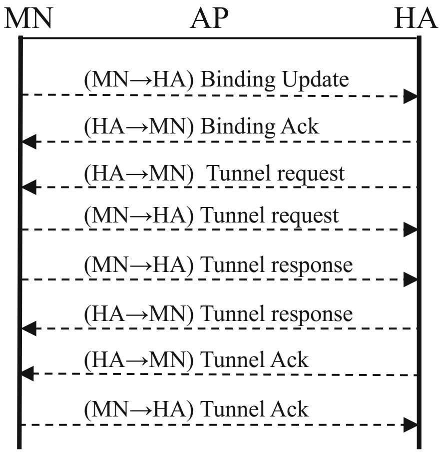

Fig. 2. An example of MIPv6 packet delivery in HN and FN

2 Handover Protocols

MIPv6 is the legacy protocol that handles handovers for MNs. Multiple variants of MIPv6 protocols have been proposed by IETF over the past years. Two of the most popular MIPv6 variants are Hierarchical MIPv6 (HMIPv6) and Proxy MIPv6 (PMIPv6).

2.1 Mobile IPv6

According to MIPv6, a MN connected to its Home Network (HN) is assigned an IP address called the Home Address (HoA), as shown in Fig. 1a. A MIPv6 based network must have at least one entity called Home Agent (HA) that stores HoA of the MN. A MN is always identified by its HoA when in home network. When a MN travels from its HN to another network, called Foreign Network (FN), it must performs handover to seamlessly change the PoA. While within FN, a MN is assigned a Care of Address (CoA). The MN must send its current CoA to HA. HA then binds the HoA to the CoA to form a data tunnel between MN and CN. The handover process completes when the MN's CoA is updated in HA. During handover, HA buffers all the incoming traffic for MN. The data tunnel enables data exchange from CN to the CoA of the MN, and back. As shown in Fig. 1b, all IPv6 packets destined for the MN's HoA are routed to its CoA by HA. Note that CoA may change but HoA remains the same until the session ends. The process of building tunnel is shown in Fig. 2. Note that at least one HA must be configured on the home network, and MN must know the IP address of HA.

2.2 Proximity Mobile IPv6

PMIPv6 is a network-controlled variant of MIPv6 in which, contrary to the legacy method, handover process is initiated by the network [9]. The initiation of the handover depends on parameters such as signal strength, etc. Hence, PMIPv6 offloads handover decision from MN to the network. This preserves the battery power for MN that is critical for its operation. The main entity in PMIPv6-based architecture is the Local Mobility Anchor (LMA), which basically acts as a HA. Each LMA manages the set of Media Access Gateways (MAGs). Each MAG is further responsible for managing and reporting the connection of MN with LMA. Most of the mobility management signals flow between MAG and LMA over wired links. The wired connectivity between LMA and MAG makes control signaling faster and more reliable [10].

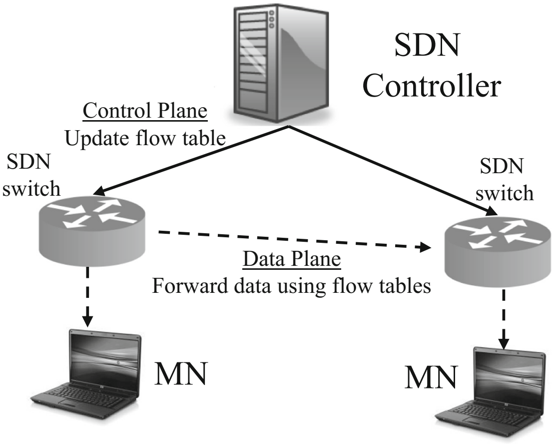

Fig. 3. A depiction of Software Defined Networking

2.3 Hierarchical Mobile IPv6

MIPv6 manages global (inter-network) and local (intra-network) handovers in the same way. HA may be located at multi-hop distances from the MN or it may even be in another network, which incurs additional signaling every time MN sends binding updates. This overhead grows larger in situations where local handover rate is high. In order to treat the local and global handovers differently (so to incur less overhead in the former), IETF has proposed Hierarchical MIPv6 (HMIPv6) [11]. HMIPv6 introduced a new node, called Mobility Anchor Point (MAP), which manages local handovers. On the other hand, the global handovers occur in same manner as in MIPv6. Instead of a single CoA, HMIPv6 assigns a Local CoA (LCoA) and Regional CoA (RCoA). For a handover within the same network, LCoA is changed whereas RCoA remains the same. The new LCoA is updated in the MAP though binding update process. Note that in HMIPv6, each AP is connected to the MAP. A set of APs and their MAP constitute a MAP mobility domain. When a MN travels between two different networks (i.e. two different MAP), both LCoA and RCoA change. The layering in HMIPv6 reduces

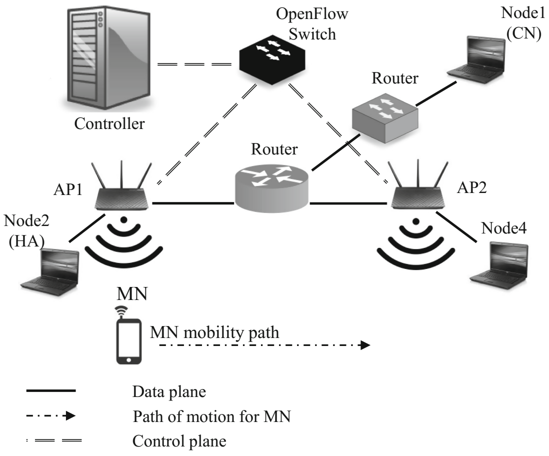

Fig. 4. SDN-enabled MIPv6 implementation

the redundant signaling, especially in high mobility environments because local and global mobility is managed separately. We have also modified legacy HMIPv6 as Network-Assisted HMIPv6 (NA-HMIPv6).

To this end, we have briefly discussed MIPv6 and its two variants that are relevant to this work. In the next section, we explain SDN so that its implementation for mobile IP can be explained in Sect. 4.

3 Software Defined Networking

SDN transforms a conventional hard-wired network into a programmable network by decoupling the control and data planes. For example, legacy routers forward incoming packets from one port to an out-going port based on routing table information. Hence, a router can be viewed as having two planes: (i) decision making plane (control plane), and (ii) packet forwarding plane (data plane). SDN decouples these two planes, providing more flexibility to control routing during run time, as shown in Fig. 3. This way, SDN manages to keep the network behaviour adaptive and dynamic.

The communication link between the SDN controller and its routers carries the information that updates the flow tables. For this purpose, several communication protocols have been proposed. OpenFlow has been the most popular choice [12].

3.1 OpenFlow Based SDN

OpenFlow (OF) is a communication interface between control and data planes of SDN. OF, which is also referred to as an OF-Switch, enables direct control of the routers' data plane via the controller. An OF-Switch maintains flow table(s),

Table 1. OF controller flow table (packet (P_n) of size N for flow k)

Packets	Actions
$P_{k,0}$	Re-direct to controller
$P_{k,i}$	Next hop to R_2 if $k = 0$, $\forall i \in N$
$P_{k,i}$	Next hop to R_3 if $k \neq 0$, $\forall i \in N$

which are used to control packet flow. The flow-table entries in an SDN-enabled router can be modified (i.e. additions, deletion of flow entries) by the OF controller in run time. Basic entries in a flow table are 'flows' and 'actions'. The flow-entry extracts packet information (e.g. MAC address, or incoming port etc.). The action table keeps information about the actions associated with the flow-entry (e.g. forward packet to a dedicated port, or block packet etc.). By default, the first packet of each flow is sent to the controller, for example, a router simply routes the packet to the controller. After decoding the packet header (examining information like MAC address, or IP address etc.), the controller sets the action field for this particular flow. The controller also informs the router of these actions. Table 1 shows a sample flow table in an OF-Switch. Each flow has packet size N. If there are k flows, the first packet of each flow is redirected to the controller. The controller then decides the action based on header information from the first packet, e.g. all the packets from the first flow are routed to R_2 while the rest of the flow is directed towards R_3.

In this research, we have used OpenFlow-enabled SDN to simulate mobile IP protocols.

Table 2. Address tables of devices

Device name	Address
$Node_1$ (CN)	1.0.4.2
AP_1 (HN)	1.0.2.1
$Node_2$ (HA)	1.0.2.2
MN	1.0.2.3
AP_2 (FN)	1.0.3.1
Node4	1.0.3.2

4 SDN-Enabled Mobile IP

In this section, we discuss implementation of SDN-enabled mobile IP protocols in Estinet simulator [13,14]. We have compared following four protocols: (i) Legacy MIPv6, (ii) PMIPv6, (iii) HMIPv6, and (iv) Network Assisted HMIPv6.

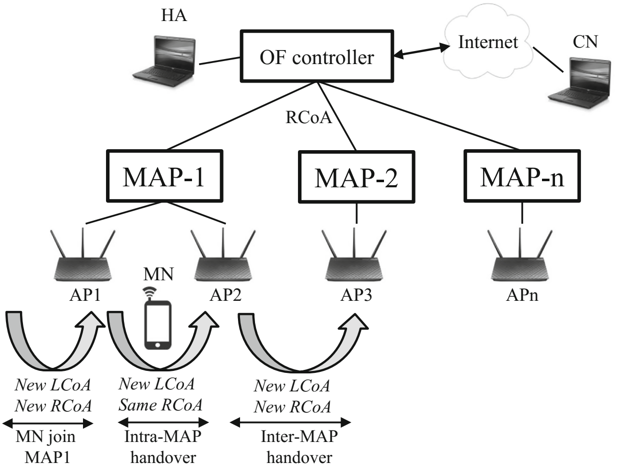

Fig. 5. SDN-enabled HMIPv6 and NA-HMIPv6 implementation

4.1 Simulation Environment

The implementation of SDN-enabled MIPv6 is presented in Fig. 4. Access point AP_1 is the PoA for MN residing in its home network. The CN ($Node_1$) is sending data to MN. The data is first sent to the controller, which directs the route CN-to-MN through AP_1. When MN leaves the communication range of AP_1 and enters into the range of AP_2, it receives a temporary CoA. All the forthcoming data from CN will now buffer into HA until MN updates its new CoA information to HA ($Node_2$). After binding-update is completed, HA sends buffered data to AP_1, which directs data through the router to AP_2 to MN. This process is called tunneling, which continues until the session ends. Tunneling stops when MN returns to home network, or when it enters a new FN. Table 2 shows all the IP addresses associated with the communicating nodes in our simulation. The architectural implementation of MIPv6 and PMIPv6 required minor changes. For example, MAG in PMIPv6 is analogous to HA in MIPv6.

We ran simulations for HMIPv6 setup shown in Fig. 5. Recall that the motivation behind using hierarchical MIPv6 is to benefit from the layered architecture to reduce signaling overhead. The hierarchical architecture comprises of three layers: OF controller (OFC), MAPs, and APs. All the APs in a domain are listed inside OFC. As MN moves, OFC configures LCoA and RCoA for MN given the prefix of the new PoA. In our simulation settings, we have considered three kinds of handovers. In the first instance, a MN joins a network (Join Handover). The MN is assigned a new LCoA and RCoA as it joins MAP-1. When MN leaves communication range of AP_1 to enter the range of AP_2 (intra-MAP handover), the MN is assigned a new LCoA, while the RCoA remains the same. The third instance is when MN leaves the domain of MAP-1 to enter the domain of MAP-2 (inter-MAP handover). MN is assigned a new LCoA and RCoA in this case. The binding between RCoA and LCoA is managed by MAP.

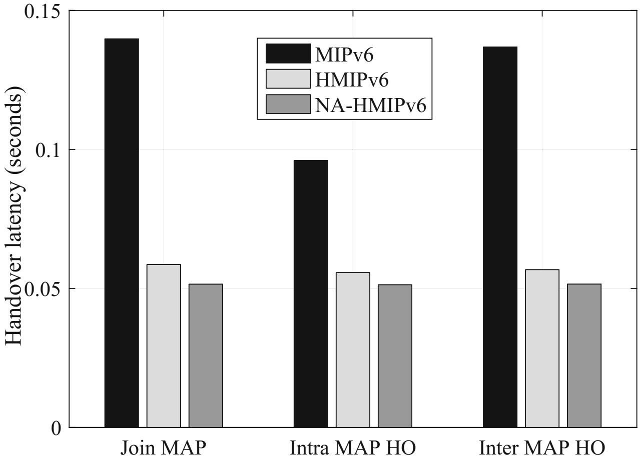

Fig. 6. Comparison of handover latency

Table 3. Handover latency and signaling between MIPv6 and PMIPv6

Protocol	MIPv6	PMIPv6
Total time (sec)	0.1802	0.0068
Number of steps	10	14

Later, we also implemented the Network Assisted HMIPv6 (NA-HMIPv6) in which the MN will initiate the handover process as in HMIPv6. The goal here is to minimize the MN involvement in the handover process while still maintaining the layered architecture of HMIPv6. The architecture for NA-HMIPv6 is same as shown in Fig. 5.

4.2 Simulation Results

Here we examine the performance of SDN-enabled mobility protocols based on the time required to complete the handover, and the number of steps required for this purpose. As mentioned in Sect. 4.1, we have considered three types of handovers in our simulation for HMIPv6 and NA-HMIPv6. Figure 6 shows that SDN-enabled NA-HMIPv6 reduces the handover latency compared to SDN-enabled HMIPv6 and MIPv6.

Since PMIPv6 and HMIPv6 are based on different architecture, direct comparison between two protocols is not possible. Similarly, comparing MIPv6 and PMIPv6 in Table 3, it is also evident that SDN-enabled PMIPv6 performs better than SDN-enabled MIPv6.

Figure 7 shows the number of steps required to complete the handover process. It can be observed that legacy MIPv6 takes lesser number of steps compared to HMIPv6, NA-HMIPv6 and PMIPv6 (see Table 3). However, in our simula-

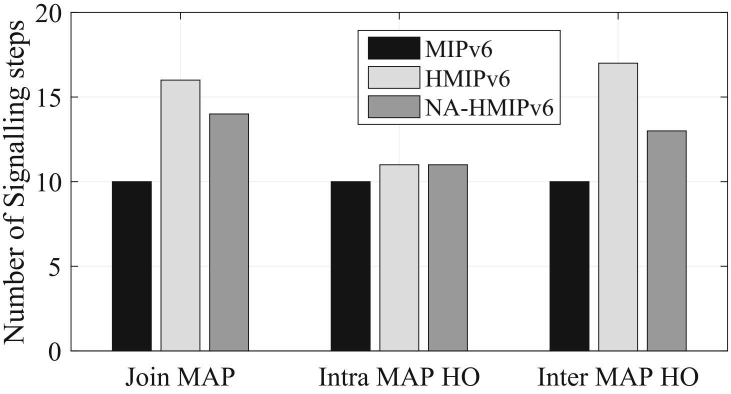

Fig. 7. Comparison of steps required to complete handover

tion, it was observed that the cumulative latency to accomplish handover with MIPv6 was higher than its variants (i.e. PMIPv6, HMIPv6 and NA-HMIPv6).

5 Conclusion

In this paper, we have examined the performance of different mobile IP protocols with software defined network. We have done extensive simulation using Estinet simulator to demonstrate our results. It was found that SDN-enabled PMIPv6 takes lesser time to complete the handover process than MIPv6 protocol. Similarly, handover latency for SDN-enabled MIPv6 is higher when compared to SDN-enabled HMIPv6 and NA-HMIPv6 in different mobility scenarios. Furthermore, NA-HMIPv6 performs better compared to legacy HMIPv6 in terms of handover latency. On the other hand, SDN-enabled MIPv6 takes least number of steps to complete the handover process. A detailed performance comparison between SDN-enabled HMIPv6 and SDN-enabled PMIPv6 needs further investigation and is considered as part of our future research.

Acknowledgement. This work has been funded through Internet New Zealand's research grant of 2016/2017.

References

1. Koodli, R.S., Perkins, C.E.: Mobile Inter-networking with IPv6: Concepts, Principles and Practices. Wiley, Hoboken (2007)
2. Johnson, D., Perkins, C., Arkko, J.: Mobility support in IPv6. Technical report (2004)
3. Guimaraes, C., Corujo, D., Aguiar, R.L., Silva, F., Frosi, P.: Empowering software defined wireless networks through media independent handover management. In: Global Communications Conference (GLOBECOM). IEEE (2013)

4. Farhady, H., Lee, H., Nakao, A.: Software-defined networking: a survey. Comput. Netw. **81**, 79–95 (2015)
5. Jeon, S., Guimarães, C., Aguiar, R.L.: SDN-based mobile networking for cellular operators. In: Proceedings of the 9th ACM Workshop on Mobility in the Evolving Internet Architecture, pp. 13–18. ACM (2014)
6. Wang, Y., Bi, J.: A solution for IP mobility support in software defined networks. In: 2014 23rd International Conference on Computer Communication and Networks (ICCCN), pp. 1–8. IEEE (2014)
7. Hasan, S.F.: A discussion on software-defined handovers in hierarchical MIPv6 networks. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 140–144. IEEE (2015)
8. Basu, D., Hussain, A.A., Hasan, S.F.: A distributed mechanism for software-based mobility management. In: 2016 7th IEEE International Conference on Software Engineering and Service Science (2016)
9. Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., Patil, B.: Proxy mobile IPv6. Technical report (2008)
10. Li, Y., Wang, H., Liu, M., Zhang, B., Mao, H.: Software defined networking for distributed mobility management. In: 2013 IEEE Globecom Workshops (GC Wkshps), pp. 885–889. IEEE (2013)
11. Soliman, H., Castelluccia, C., Elmalki, K., Bellier, L.: Hierarchical mobile IPv6 (HMIPv6) mobility management. Technical report (2008)
12. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
13. EstiNet Technologies Inc.: Estinet Simulator (2017). http://www.estinet.com/ns/
14. Wang, S.-Y., Chou, C.-L., Yang, C.-M.: Estinet openflow network simulator and emulator. IEEE Commun. Mag. **51**(9), 110–117 (2013)