May-June 2020
ISSN: 0193-4120Page No. 10845 - 10850

Performance Analysis of Proxy Mobile IPv6 and Software-Defined Mobility

Dr. Sridevi
Associate Professor, Department of Computer Science, K.U., Dharwad
devisris@yahoo.com

Article Info
Volume 83
Page Number: 10845 - 10850
Publication Issue:
May- June 2020

Abstract

In this paper a new mobility management approach which called Software-Defined mobility (SDN) and has shown that SDN can be implemented without IP mobility protocol for providing mobility like as Proxy Mobile IPv6 (PMIPv6) that is the solution adopted by 3GPP, with some performance gain.

Keywords:*SDN-Mobility, PMIPv6, Mobile Node, etc.*

Article History
Article Received*: 19 November 2019*
Revised*: 27 January 2020*
Accepted*: 24 February 2020*
Publication*: 19 May 2020*

1. INTRODUCTION

Generally, the basic problem in supporting Internet mobility is to deliver data to a mobile receiver, whose location in the network topology is dynamically changing. Mobile IP (MIP) [1] [2] is one of the earliest and most well-known protocol. Mobile IP is an Internet Engineering Task Force (IETF) standardized protocol which allows Mobile Nodes (MN) to move from one network to another without losing connectivity. Several solutions were proposed to improve IP mobility management such as Fast Mobile IPv6 (FMIPv6) [3], Hierarchical Mobile IPv6 (HMIPv6) [4] and Proxy Mobile IPv6 (PMIPv6) [5]. Even FMIPv6 and HMIPv6 improved handover latency, they are host-based mobility management protocols. The MN needs to modify its protocol stack to support mobility signaling. Consequently, IETF decided to develop a Network-based Localized Mobility Management (NetLMM) solution where the network entities take the responsibility of exchanging mobility signaling on behalf of the MN. Mobility is achieved without requiring the nodes to have some specific configuration or software installation. The main advantage of PMIP compared to other MIP solutions that are host-based is that only the network is concerned by the mobility management, not the mobile. As mobile has no signaling to exchange, it is a more flexible and convenient solution to use in a real network. While the PMIP solution has advantages that made the 3GPP standardizes it, its relevance in the context of new network architectures arises.

Originally, each network device could be controlled and managed individually. But, each device of the different vendor has different firmware, and the forwarding and control planes are coupled into one box. Thus, it is not flexible and hard to manage. The Software-Defined

Networking (SDN) [6] [7] aims to introduce flexibility by leveraging the software components of the network. It is a new approach to computer networking that can help the network administrators to configure, update and monitor the different network devices and the various manufacturers more accessible through a software application. It makes the addition of network function easier. Thus, it is not surprising that many network deployments are using SDN for flexible and more comfortable management.

2. SDN-MOBILITY ARCHITECTURE

The SDN Mobility architecture has two main functional entities, the controller and the Access Router (AR) as shown in Figure 1 in the red label. An OpenFlow controller is located in the same network as ARs. Its duty is to be responsible for the flow table to all AR in SDN Mobility network. Access Routers (ARs) are the OpenFlow switched that are located on the access network. They are responsible for the movement of MN and the OpenFlow message exchange with its controller. In the following, we distinguish the Previous Access Router (PAR) from the New Access Router (NAR).

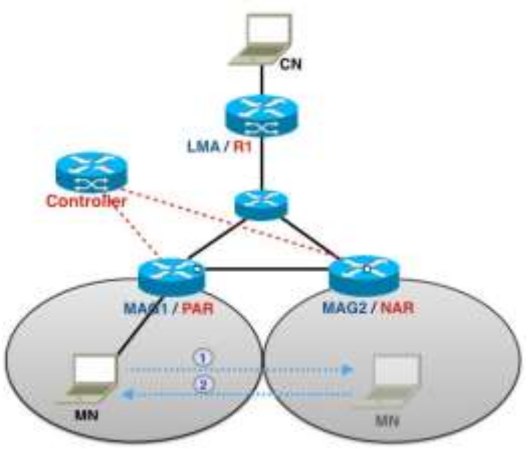

Figure 1: PMIPv6/SDN Mobility Architecture

3. SDN-MOBILITY OPERATION

In SDN network, when the MN moves to the new location and attaches to a new Access Router (NAR), the routing path in the Access Routers (ARs) is not updated. So, the packets flow continue to be forwarded to the Previous Access Router (PAR). This problem is being solved by modifying the function of the response of the MN status and using the SDN concept to update the controller with the MN status and the routing path of MN. We did this solution in this paper. Our approach, SDN-Mobility uses OpenFlow protocol for communication between the controller and OpenFlow switches. The OpenFlow messages are sent to exchange the information between the controller and every ARs for mobility management in SDN network. The SDN Mobility operation can be separated in two procedures: MN registration and MN handover. Both procedures transmit OpenFlow messages for notifying MN event and for updating the routing path. The SDN Mobility signaling illustrated in Figure.2, looks similar as PMIPv6.

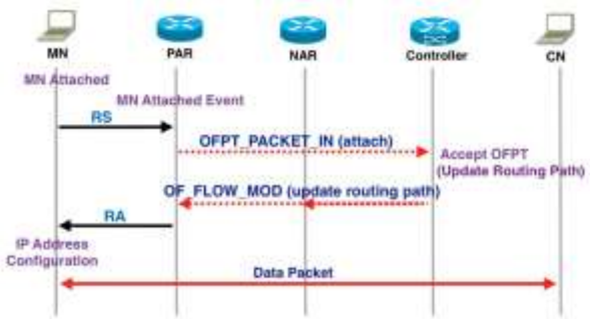

Figure 2: SDN Mobility Signaling

MN Registration: when the MN enters in the SDN Mobility network, PAR detects the attached event and sends the OFPT_PACKET_IN message to its controller for informing this event. After the controller received and processed that message, it sends the OF_FLOW_MOD message to all ARs for adding the routing flow of MN. Then, CN directly communicates with MN. Note that, we

Published by: The Mattingley Publishing Co.,

May-June 2020
ISSN: 0193-4120Page No. 10845 - 10850

assume that the controller has the network policy allowing the MN to access network.

MN Handover: When the MN is detached from PAR, PAR sends OFPT_PACKET_IN message to its controller to inform it of this event. The controller immediately sends the OF_FLOW_MOD message to all ARs for deleting the routing flow of MN. At the same time, the MN moves tothe new location, attaches to the NAR, NAR informs of the attached event to its controller by sending OFPT_PACKET_IN message. After the controller received and processed that message, it sends the OF_FLOW_MOD message to all ARs for adding the routing flow of MN. Then, the connection of MN and CN continues.

4. IMPLEMENTATION

The caching mechanism is proposed for handover management in the LTE cellular network. Controlled by the Serving Gateway (SGW) which has a central view. The data are stored on the source E-UTRAN Node B (eNB) when the Radio Link Control protocol (RLC) detects the losses of data. Then, the source eNB transfers the data to the target eNB that will store the data until SGW received signaling from the target eNB, indicating that user data plane has been switching to the target eNB.

Unlike this context, we have enrolled in a network without connection signaling, without data link level 2 able to number the frames and therefore without buffering mechanism defined for mobility management.

SDN-Mobility and PMIPv6 are network-based mobility approaches that do not require the signaling from MN which is suitable to be implemented for localized mobility network. SDN-Mobility operates an IPv6 autoconfiguration same as PMIPv6 for obtaining an IPv6 address. It uses the OpenFlow message for immediately updating the routing path, leading to avoid IP-in-IP tunneling establishment. But PMIPv6 approach uses the PMIPv6 messages to setup IP-in-IP Tunnel for communication between MN and CN. They are very easy and smooth to be in the real network. However, both methods need to add extra functions at the routers or switches for the mobility management. The problem of both methods is the packet loss during the MN handover. The size of each message in PMIPv6 and SDN-Mobility is represented in TABLE1. The comparison of PMIPv6 and SDN-Mobility can be illustrated in TABLE 2 and TABLE 3.

Considering the number of exchanged messages in PMIPv6 and SDN-Mobility as shown in TABLE 4, the signaling cost of SDN-Mobility is almost the same as PMIPv6 during MN registration and MN handover. Although PMIPv6 and SDN-Mobility can be implemented to support mobility, PMIPv6 uses IP-in-IP tunneling technique which makes some overhead cost that depends on the number of transferred packets. Thus, if the data size is large, it will make more IP-in-IP overhead, leading to increasing the packet transmission time. SDN-Mobility can provide packet forwarding directly without tunneling establishment by immediately updating the routing flow as the MN changes the point of attachment.

TABLE-1: PMIPv6 and SDN-Mobility Messages Summary

Message	Size (in bytes)
IPv6 Address Configuration	
Router Solicitation (RS)	**16**
Router Advertisement (RA)	**56**
PMIPV6	
Proxy Binding Update (PBU)	**84**
Proxy Binding Acknowledgement (PBA)	**76**
SDN Mobility	

Published by: The Mattingley Publishing Co.,

May-June 2020
ISSN: 0193-4120Page No. 10845 - 10850

OFPT_PACKET_IN (attached/detached)	**156**
OF_FLOW_MOD (update routing path)	**120**

TABLE-2: PMIPv6 and SDN-Mobility summarization

Approach	PMIPv6 Operation	OpenFlow Operation
PMIPv6	-notifying the attachment of MN - forwarding the HNP - add/delete/update BCE - setup the tunnel	
SDN Mobility		- notifying the attachment of MN - setting up the routing path

TABLE-3: PMIPv6 and SDN-Mobility Advantages and Limitations

Approach	Advantages	Requirement and Limitations
PMIPv6	network based mobility decreases the long handover latencyof MN problem in MIPv6	localized mobility management. IP-in-IP Tunneling overhead. packets loss during MN handover. PMIPv6 implementation requirement
SDN Mobility	network based mobility avoids IP-in-IP tunneling establishment. decreases the handover latency of MN problem.	localized mobility management. packets loss during MN handover. OpenFlow implementation requirement..

TABLE-4: The Number of Signaling Comparison in PMIPv6 and SDN-Mobility in MN Registration Period and Handover Period.

	MN Registration Period		MN Handover Period	
	PMIPv6 (Messages)	SDN-Mobility (Messages)	PMIPv6 (Messages)	SDN-Mobility (Messages)
IP Address Configuration				
Router Solicitation (RS)	1	1	1	1
Router Advertisement (RA)	1	1	1	1
Binding Address and Tunnel				
PBU	1	-	1	-
PBA	1	-	1	-
Movement Detection				
OF_PACKET_IN	-	1	-	1
OF_FLOW_MOD	-	2	-	2
OF_PACKET_OUT	-	-	-	-
Tunneling Establishment				
IP-in-IP Tunnel	Yes	No	Yes	No
Total	4	5	6	5

5. EXPERIMENTAL COMPARISON: SDN-MOBILITY VS PMIPV6

An objective is to justify that SDN can be used to provide the mobility of MN without the implementation of the legacy Mobility protocol. We set up an experimental network to compare the performance of SDN-Mobility and PMIPv6, in Figure 1. Labels represent usual notation in PMIPv6 and SDN for each component. We use Mininet [9] to generate topology. The MN connects to switches by using the wire channel. We update the source code to enable the MN to attach/detach with the switch in Mininet. It acts like a MN movement by hard handover scheme.

Compiled the kernel and installed UMIP mobility patch for PMIPv6 and uses RYU controller for SDN network with a set of parameters as shown in TABLE 5. Iperf tool generates UDP and TCP traffics and performs the performance measurement. We also use Wireshark to capture TCP traffic and use TShark to classify the data for performance analysis.

Published by: The Mattingley Publishing Co.,

TABLE 5: Simulation Parameters of PMIPv6 and SDN-Mobility

Parameter	Setting	
	PMIPv6	SDN-Mobility
Simulation Tool	Mininet 2.1.0p2	Mininet 2.1.0p2
Mobility Patch	PMIPv6-v0.4.1	-
Bandwidth on edge	10Mbps	10Mbps
All Link delay	$0.5\text{x}10^{-6}$ s	$0.5\text{x}10^{-6}$ s
Controller	-	Ryu 3.18
OpenFlow Message	-	v1.3.0
Testing Tool	Iperf v2.0.5	Iperf v2.0.5
UDP Datagram	1450 Byte	1450 Byte

6. PERFORMANCE ANALYSIS

Two scenarios were proposed for measuring performance on the different transport protocols (UDP and TCP).

6.1 Scenario 1: UDP

This scenario was designed for UDP performance measurement that we separated into two sub-experiments: UDP throughput and Packet Loss.

UDP Throughput

We ran Iperf server and Iperf client at CN and MN. We generate UDP traffic from MN to CN for 50 seconds and report the result every 0.5 seconds. In this scenario, MN moves two times. Five seconds after the simulation start, MN moves to the other attachment and will move back to a home network at 20 seconds later. The result of UDP throughput of PMIPv6 and SDN-Mobility can be illustrated in Figure 3. Considering the result in Figure 3, in y-axis, the result shows that the UDP throughput of SDN-Mobility is higher than PMIPv6 about 1 Mbps that is caused by the tunnelling overhead of PMIPv6. The UDP throughput of both methods significantly dropped when the MN changed the point of attachment to the other access router after second 5.0 and reached 0 Mbps in second 5.5 for both methods. Then, the UDP throughput increases when the MN already attaches again and obtains an IPv6 address. The UDP throughput of SDN-Mobility began to increase at second 6.0 and second 7.0 in PMIPv6.

This difference in times is due to a handover latency which is about 1.0 second for SDN-Mobility and about 2.0 seconds for PMIPv6.

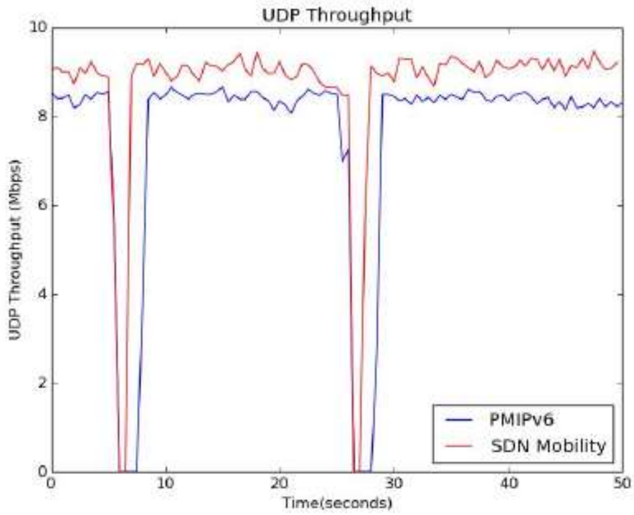

Figure 3 PMIPv6 and SDN Mobility UDP Throughput

6.2 Packet Loss

In another experiment, we ran Iperf server and Iperf client at CN and MN. We generate UDP traffic from MN to CN for 50 seconds and report the result every 0.5 seconds. We did two sets of experiments: the first with one MN handover and the second with two MN handovers. For each set, we repeat simulation for 20 times and average the results. The number of packet loss can be shown in Figure 4. For one MN handover experiment, MN moves only one time. MN moves to the other attachment 5 seconds after the simulation start. MN will move back to the previous attachment at 20 seconds later for two MN handovers experiment. Considering the percentage of packet loss in Figure 4, the percentage of packet loss of PMIPv6 is 9.82% and 15.85% for one MN handover and two MN handovers. The packet loss is 3.85% and 7.0% for SDN-Mobility in one MN handover and two MN handovers experiment. As this result shows SDN-Mobility gives a lower

Published by: The Mattingley Publishing Co.,

May-June 2020
ISSN: 0193-4120Page No. 10845 - 10850

percentage of packet loss compared to PMIPv6 which is about twice.

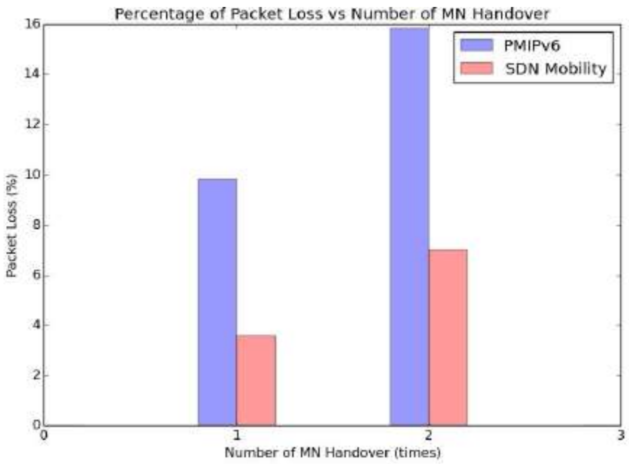

Figure 4 Percentage of Packet Loss versus Number of MN Handover

6.3 Scenario 2: TCP

In TCP scenario, We generated TCP traffic by running Iperf between CN and MN for 50 seconds. During simulation times, MN moves to other attachment at second 5 and will move back to the previous attachment at second 25. Figure 5 shows TCP sequence of PMIPv6 and SDN-Mobility, from this we can learn that during handover time, TCP packets cannot be sent to the CN. So the TCP sequence number is held and will be counted after the MN connection restores. This result shows that SDN-Mobility took a shorter handover delay than PMIPv6.

7. CONCLUSION

The experimental results of the wired experiment have supported our proposal that SDN Mobility can be implemented for Mobility Management like PMIPv6 without Mobility protocol implementation. Moreover, SDN-Mobility is better than PMIPv6 in two advantages: provides higher throughput and faster management.

REFERENCES

[1] C. Perkins, “IP Mobility Support for IPv4,” RFC 3344, August 2002.

[2] D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6,” RFC3775, June 2004.

[3] R. Koodli, “Mobile IPv6 Fast Handovers,” RFC 5568, July 2009.

[4] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, “Hierarchical Mobile IPv6 (HMIPv6) Mobility Management,” RFC 5380, October 2008.

[5] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy Mobile IPv6,” RFC 5213, Auguest 2008.

[6] K. Tantayakul, R. Dhaou, and B. Paillassa, “Experimental Analysis in SDN Open Source,” IEEE ECTI-CON 2017, pp. 337–340, June 2017.

[7] Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks (ONF White Paper),” ONF, April 2012.

[8] C. Alaettinoglu, “Software Defined Networking,” PACKET DESIGN, 2013.

[9] “Mininet: An Instant Virtual Network on your Laptop (or other PC),” Available: http://mininet.org.

Published by: The Mattingley Publishing Co.,