



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 16893

The contribution was presented at AINA 2016 :
<http://voyager.ce.fit.ac.jp/conf/aina/2016/>

To cite this version : Tantayakul, Kuljaree and Dhaou, Riadh and Paillassa, Béatrice *Impact of SDN on Mobility Management*. (2016) In: 30th Advanced Information Networking and Applications (AINA 2016), 23 March 2016 - 25 March 2016 (Crans-Montana, Switzerland).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Impact of SDN on Mobility Management

Kuljaree Tantayakul
IRIT-ENSEEIH
University of Toulouse, France
Email: kuljaree.tantayakul@etu.enseeiht.fr

Riadh Dhaou
IRIT-ENSEEIH
University of Toulouse, France
Email: Riadh.Dhaou@enseeiht.fr

Beatrice Paillassa
IRIT-ENSEEIH
University of Toulouse, France
Email: Beatrice.Paillassa@enseeiht.fr

Abstract—The software integration with new network architectures via SDN (Software Defined Network) axis appears to be a major evolution of networks. While this paradigm was primarily developed for easy network setup, its ability to integrate services has also to be considered. Thus, the mobility service for which solutions have been proposed in conventional architectures by defining standardized protocols should be rethought in terms of SDN service. Mobile devices might use or move in SDN network. In this paper, we have 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.

Index Terms—Software-defined Network, PMIPv6, SDN Mobility.

I. INTRODUCTION

Currently, the use of wireless mobility communication and mobile devices such as laptops and handheld devices are increasing rapidly. This can be supported by Mobile consumer Internet traffic in exabytes per month [1]. Since 2013, the Mobile consumer are increasing every year and is predicted to increase even more. Each of mobile devices, called a Mobile Node (MN), may change its point of attachment, leading to a possible disconnection. Originally, IETF solved this problem by Mobile IP (MIP) which supported both IPv4 and IPv6 for MN Mobility. Several solutions were proposed to improve IP mobility management such as Fast Mobile IPv6 (FMIPv6) [2], Hierarchical Mobile IPv6 (HMIPv6) [3] and Proxy Mobile IPv6 (PMIPv6) [4]. 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.

Main advantage of PMIP compare 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 it to be standardized by the 3GPP, its relevance in the context of new network architectures arises.

Originally, each network device could be controlled and managed individually. But, each device of different vendor has different firmware and the forwarding and control planes

are coupled within one box. Thus it is not flexible and hard to manage. The Software-Defined Network (SDN) [5][6] aims to introduce flexibility by leveraging the software components of the network. It is an new approach to computer networking that can help the network administrators to configure, update and monitor the different network devices and the different manufacturers easier by through software application. It makes addition of network function easier. Thus, it is not surprising that there are many network deployments using SDN for flexible and easier management.

To the extent that software components can be easily deployed in the network through the SDN approach, PMIP solution can be easily integrated. However, the benefit of this deployment raises questions. Is there no redundancy between the components of mobility and the components of SDN? This is the question that we answer in this paper.

We examine the interest of the PMIP solution in the context of the SDN architecture. This paper began as an examination of Mobility IPv6 and SDN issues. We had hoping to find the ways to improve the mobility management in SDN. During that examination it was noticed that we can simply use SDN concept to manage the mobility operation without mobility protocol implementation. Consequently, we propose a new possible solution by using the powerful OpenFlow protocols [7][8][9].

At first the paper introduces the basic elements of MIP and SDN and presents related works. At the second, our proposed approach which is called SDN Mobility is detailed. In the fact this solution hides the PMIP protocol. We evaluate the solution on an experimental topology and analyses the obtained results.

II. MOBILITY AND SDN SERVICES

A. Proxy Mobile IPv6 (PMIPv6)

Management of IP mobiles postulates no changes at the IP level. Mobiles use their IP addresses and if they move and so get a new address, it is on the responsibility of the management to hide this change in a such way that the communication has not to be interrupted. Mobile management has to follow the mobile movement in order to localized it and forward its data to the new localization by tunneling. It is done by the consultation of an association table, named Binding Cache, containing the user reference, that is its home IP address (HoA), and its localization that is the address of the router element next to the user (CoA for Care of Address). This table is filled by control messages sent by network

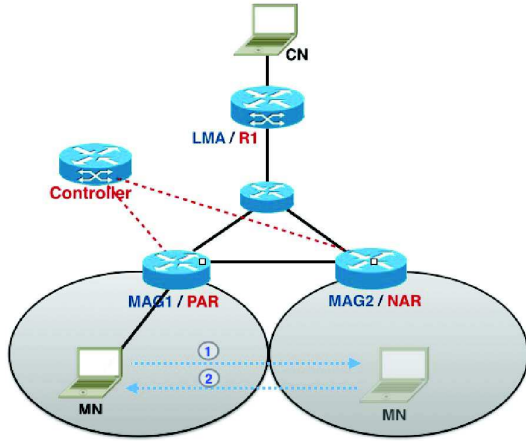


Fig. 1. PMIPv6/SDN Mobility Architecture

elements. There are Proxy Binding Update and Proxy Binding Acknowledgement (PBA) messages.

1) *PMIPv6 Architecture*: More precisely PMIPv6 architecture has two main functional entities, a Local Mobility Anchor (LMA) and a Mobile Access Gateway (MAG) as illustrated in Fig. 1.

MAG: The MAG is an entity which handles mobility signaling on the behalf of the MN while MN attached to its access links. It uses the acquired identifier of Mobile Node (MN-ID) for a modified Proxy Binding Update (PBU) and for the authentication process.

LMA: It keeps a Binding Cache Entry (BCE) for each registered MN. LMA acts as the Home Agent (HA) of the MN in Proxy Mobile IPv6 domain. It is responsible for managing the reachability state of the MN and for providing the Home Network Prefix (HNP).

As usually, we distinguish the Previous Access Router (PAR) from the New Access Router (NAR).

2) *PMIPv6 Operation*: In Fig. 2 is illustrated in blue full line, the signaling of basic PMIPv6 operation. When MN enters to PMIPv6 network, MAG1 senses the MN attached event. MAG1 uses the acquired identifier of Mobile Node (MN-ID) to send a Proxy Binding Update (PBU) to LMA for the registration process. After LMA received the PBU message, it checks the MN-ID in Binding Catch Entry (BCE). If the MN-ID does not have entry in BCE, it is added. Then, LMA provides the Home Network Prefix (HNP) of MN to MAG1 by sending a Proxy Binding Acknowledge (PBA) back to MAG1. LMA has configured the IP tunneling of its side at the same time. When MAG1 receives the PBA from LMA, it establishes tunnel on its side. Then, Router Advertisement (RA) message is advertised in the access link by MAG1. It provides HNP for the MN. If MN does not receive the RA, it sends a Router Solicitation (RS) message to get the RA. The IPv6 address of MN can be modified by using the IPv6 auto-configuration, based-on EUI-64 standard algorithm [10]. All data communication between MN and Corresponding Node

TABLE I
PMIPv6 AND SDN MOBILITY MESSAGES SUMMARY

Message	Size (Bytes)
IPv6 Address Configuration	
Router Solicitation (RS)	16
Router Advertisement (RA)	56
PMIPv6	
Proxy Binding Update (PBU)	80
Proxy Binding Acknowledgement (PBA)	76
SDN Mobility	
OFPT_PACKET_IN (attached/detached)	180
OF_FLOW_MOD (update routing path)	120

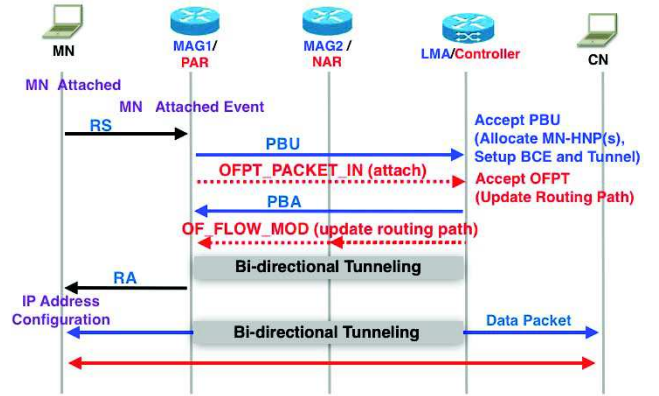


Fig. 2. Redundancy of PMIPv6 and SDN Signaling

(CN) are transmitted through the established bi-directional tunneling.

While the MN is roaming to another attachment in localized network, it detaches from MAG1, MAG1 senses this event and sends the PBU for de-registration procedure to LMA. LMA receives the PBU message and starts the BCEDelete timer for deleting the entry of MN in BCE. After that, a PBA message is replied from LMA to the MAG1. When MAG2 senses the MN attached event, it sends PBU message to LMA. LMA adds the new MN entry in BCE, configures the IP tunnel and sends PBA message to MAG2. Then the tunnel between MAG2 and LMA is established. MAG2 advertises the HNP to MN by sending a RA message. After the MN autoconfigured its IPv6 address, all packet data are transmitted through the established tunneling between MAG2 and LMA.

Messages involved in the mobility management process are summarized in TABLE I.

B. Software-defined Networking (SDN)

The concept of SDN is based-on a centralized intelligence by dividing the operation into two parts: the control plane and the data plane. The operation methods require some protocols such as the OpenFlow protocol. The key component of SDN architecture is the controller which operates like as the brain: it coordinates and manages all network devices in the SDN.

OpenFlow protocol is the first standard protocol that was defined by Open Network Foundation (ONF). It is a Layer 2

method which can be used for providing the communication between the centralized controller and the network devices in SDN architecture. It works on top of the Transmission Control Protocol (TCP) [11] and was released in many versions, OpenFlow1.5.1 (march 2015) is the current version.

Main communication between the network device and the Controller during the network operation concerns the routing. As the routing is a control function it is decided at the controller level, then decision is transmitted to the network element on a pull model based. Forwarding elements request the routing decision to their controller, it is then cached for a given time at the forwarding elements.

Combining, SDN approach to Network mobility offers some benefit in terms of overhead. It is possible to avoid tunneling during the mobile movement, since the central management can decide to send the data directly to the new localization of the Mobile. Meanwhile, as we can see in Fig. 2, there are some signaling redundancy. This redundancy is logically understandable. SDN is essentially done to centrally manage network services, and PMIP architecture manages in a centralized way the mobile network service. It is then obvious to consider that the network mobility service may be a SDN service, we named it SDN mobility service.

C. Related works

Various approaches are proposed to combine SDN and IP mobility protocol in order to improve the PMIP performance in terms of signaling and/or the handover latency. Some approaches modify the legacy mobility protocol for co-operation with SDN, other uses SDN for managing IP Mobility.

OPMIPv6 [12] uses the advantages of the OpenFlow for PMIPv6 network. But, it still uses the PMIPv6 operation for notifying the attachment of MN and forwarding the Home Network Prefix to MN. It avoids the tunnel establishment by using the OpenFlow message to setup the routing path. Meanwhile, the method requires the PMIP implementation and must implement LMA and MAG functionality at controller and switches. Mobility can not be considered as an SDN service.

OF-PMIPv6 [13] uses also PMIPv6 and SDN. OpenFlow is for sending the information in three purposes, for authentication of MN, for setup tunneling at nOMAG (next OpenFlow Mobile Access Gateway) side, and for informing of the handover event to the controller. PMIPv6 protocol is still used for forwarding the assigned Home Network Prefix (HNP), for add/delete/update Binding Cache Entry, and for setup the tunnel at LMA side. MN authentication is required at AAA server before OMAG advertises the HNP.

Although OPMIPv6 scheme introduces some proactive handover management at the controller level, and improves the performance compare to PMIPv6 require both PMIPv6 and SDN implementation.

You Wang and Jun Bi [14] proposed SDN-based to enhance Mobile IP network using OpenFlow. This approach is a network based mobility management. It uses OpenFlow messages for managing the routing path of all MN packets by mapping a MN's HoA to CoA. When MN attaches a switch,

switch assigns a CoA to MN and sends Binding Update to its controller. In this approach, CoA is the IP address of MN's first-hop switch. Then, the controller knows the CoA of MN and adds it in Binding Cache. When CN connects to MN, MN's first-hop switch downloads Binding Cache and rewrites all MN packets destination to CoA. The MN's first-hop switch rewrites all packets destination of MN to MN's IP address and forwards to MN.

Even though SDN-based is a protocol design for IP mobility by using SDN concept, it rewrites all MN's packets on network side. This can imply that the switches and the controllers suffer heavy loads and an increasing complexity as the large number of MNs.

III. PROPOSED APPROACH

In the fact the mobility protocols may run in SDN network, but the operation of mobility protocols is not relevant to SDN components, it has been done individually. There are two ways to achieve this aim. The first way needs to modify the mobility protocol for co-operation with SDN signaling and the second way needs to find a new method based on SDN signaling to provide mobility service without the legacy mobility protocol. We adopt the second way in this paper.

In this section, we propose a new approach based on SDN concept for providing IP mobility in localized network that is called SDN Mobility service. Based-on SDN concept, without legacy IP mobility protocol implementation, it is simply for real implementation, solves the overhead problem, decreases the handover latency, and so reduces the number packet losses compared to PMIP.

A. SDN Mobility Architecture

The SDN Mobility architecture has two main functional entities, the controller and Access Routers (ARs) as shown in Fig. 1 in red label.

1) *Controller*: It is an OpenFlow controller which locates in the same network as ARs. Its duty is to be responsible of the flow table to all AR in SDN Mobility network.

2) *Access Routers (ARs)*: Access Routers (ARs) are the OpenFlow switch located on the access network. They are responsible for the movement of MN and for the OpenFlow message exchange with its controller. In the following, we distinguish the Previous Access Router (PAR) from the New Access Router (NAR).

B. SDN Mobility Operation

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 Fig. 3, looks similar as PMIPv6 one.

1) *MN Registration*: First, 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

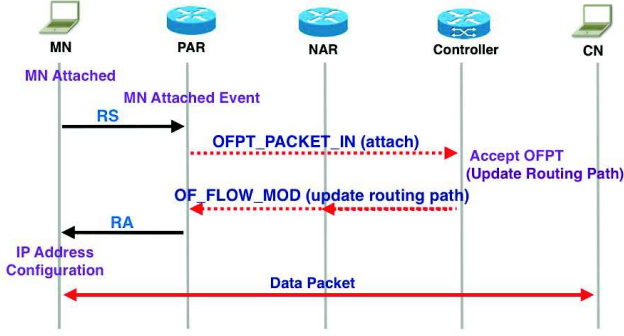


Fig. 3. SDN Mobility Signaling

TABLE II
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	0.5×10^{-6} s	0.5×10^{-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

to all ARs for adding the routing flow of MN. Then, CN directly communicate with MN. Note that, we assume that the controller has the network policy allowing the MN to access network.

2) *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 to the 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.

IV. EXPERIMENTAL TOPOLOGY

We setup an experimental network to compare the performance of SDN Mobility and PMIPv6, in Fig. 1. Labels represent usual notation in PMIPv6 and SDN for each component.

We use Mininet [15] to generate topology. Mininet supports only wire topology, and does not support models of wireless channel and mobility. So, in this work, MN connect to switches by using wire channel. We wrote a few of source code for the MN attaches/detaches with switch in Mininet, it acts like a MN movement by hard handover scheme.

We compiled the kernel and installed UMIP mobility patch [16][17] for PMIPv6 and uses RYU [18] controller for SDN network that set parameters as in TABLE II. Iperf [19] tool

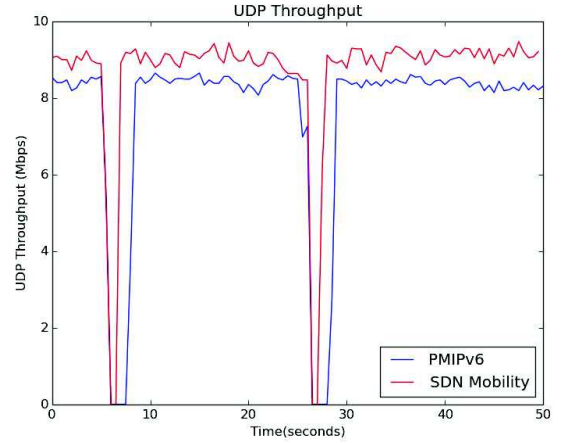


Fig. 4. PMIPv6 and SDN Mobility UDP Throughput

generates UDP and TCP traffics, and performs the performance measurement. We also use Wireshark [20] to capture TCP traffic and use tshark [21] to classify the data for performance analysis.

V. PERFORMANCE ANALYSIS

This section shows the experimental result and analysis. After we setup the experimental topology which is described in the previous section. We use the same scenario for testing PMIPv6 and SDN Mobility. Two scenarios were proposed for measuring performance on the different transport protocols (UDP and TCP).

Scenario 1: UDP

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

1) 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 home network at 20 seconds later. The result of UDP throughput of PMIPv6 and SDN Mobility can be illustrated in Fig. 4.

Considering the result in Fig. 4, 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 tunneling 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 in PMIPv6. This difference in times is due to a handover latency which is

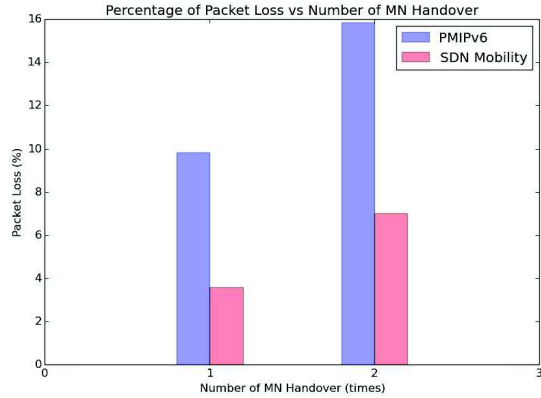


Fig. 5. Percentage of Packet Loss versus Number of MN Handover

about 1.0 second for SDN Mobility and about 2.0 seconds for PMIPv6.

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 Fig. 5.

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 Fig. 5, 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 handovers and two MN handover experiment. As this result shows that SDN Mobility gives a lower percentage of packet loss compared to PMIPv6 which is about twice.

Scenario 2: TCP

In TCP scenario, we did an experiment quite similar as UDP throughput measurement experiment. We generated a 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. Fig. 6 shows TCP sequence of PMIPv6 and SDN Mobility, from this we can learn that during handover time, TCP packets cannot send to 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 took the handover delay than PMIPv6.

3) Impact of the mobility on the loss rate

In the following, we focus on the packets loss during handover between two WIFI access points. We study the impact of the mobility during handover duration on the loss rate. In order to evaluate the number of lost packets while MN

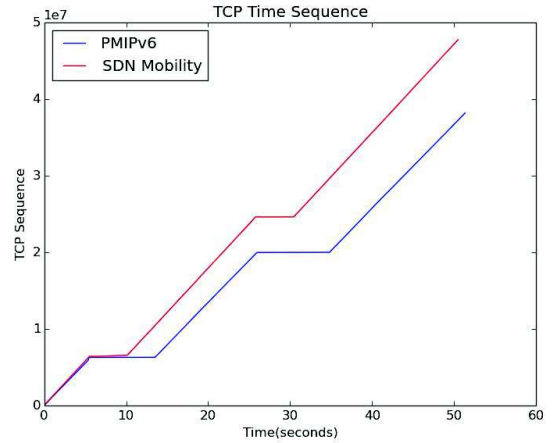


Fig. 6. PMIPv6 and SDN Mobility TCP Time Sequence

handover, we set up a MN roaming scenario (it is a situation where the MN is sending data to the CN while moving and changing its point of attachment as shown in Fig. 1).

We use Iperf to generate UDP traffic by sending the data where the size is 4 MBytes and the UDP datagram of each packet is set to 1400 Bytes. MN starts to send the data at the same position as 161 meters from OFSW-AP1. We vary the data rate from 0.5 Mbps to 1Mbps and vary the velocity of MN from 0 to 5.5 m/s. The obtained performance while varying the velocity of Mobile Node can be illustrated in Fig. 7. Considering static MN (velocity= 0 m/s), the loss equals zero for all traffics that use data rate lower than and equal to 0.9 Mbps. For data rate equals 1 Mbps, the percentage of packet loss is about three percent, which is compliant with the result of the previous scenario. Because while MN lives in a weak signaling area, OFSW-AP decreases the bandwidth capacity lower than the sending rate of the MN, leading to packets loss. We refer to WiFiAdapLoss in this case. Considering the data rate is lower than 0.9 Mbps, the percentage of loss is constant for all MN speeds. This loss, referred as, handover loss, is due to MN handover. The handover loss is directly proportional to the transmitted data (Handover Loss = Handover Delay x Data Rate).

Considering the data rate is higher than or equal to 0.9 Mbps, the percentage of loss is not constant for all MN speeds. The number of packets loss will significantly increase when the velocity of the MN is less (Total of Loss = Handover Loss + WiFiAdapLoss + BufDataLoss).

The handover loss is the number of loss during MN handover which comes from handover delay multiplied by the data rate. WiFiAdapLoss is the number of loss when the MN stay in the weak signaling area, the WiFi management decreased data rate of MN that shown the experimental results in previous scenario. BufDataLoss is the number of packets loss which are buffered in the previous AP before the MN move to attach new AP. Because while the MN is moving

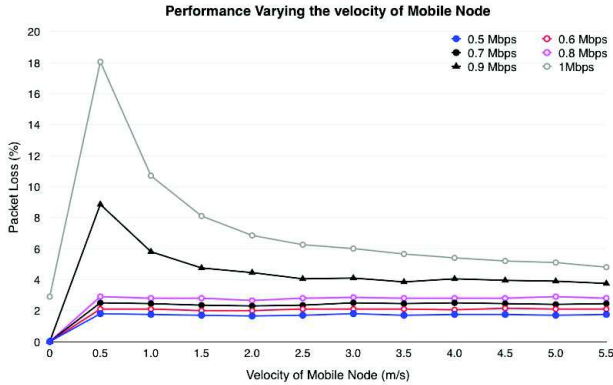


Fig. 7. Impact of the velocity on the Packet loss during handover

in the weak signaling area, the AP decreases the data rate by buffering some packets in the AP. This implies that if the MN stays a long time in the weak signaling area, more data packets will be buffered in the AP. Then the MN moves to the other AP, all old buffered packets in the previous AP are lost, leading to the number of packets loss obtained for low speed MN more than in case of faster MN. Because this latter spends a long time in a weak signaling area. To overcome these loss issues, caching techniques can be used to buffer data for the handover duration. In the future we will implement caching. In the same direction used by CEOVDS (Cross Site Evaluation of an Open flow assisted Video on Demand Distribution) project [22], we intend to use the software capacity of SDN to cache data during handover.

VI. CONCLUSION AND FUTURE WORK

In this paper, the existing IP mobility management protocol, called PMIPv6 and the SDN Mobility service we have proposed to replace it, have been implemented. Their performance has been measured in terms of UDP throughput, TCP sequence and percentage of packet loss. From the results analysis, we conclude that the SDN Mobility can be used for mobility management like as PMIPv6, but without the legacy IP mobility protocol implementation. Moreover, as SDN routing is directly managed by the centralized controller, the well-known direct routing problem is resolved, there is no need of tunnel to forward the data in case of localization change (only need to change the routing).

Meanwhile, PMIP protocol would be useful to manage mobility between heterogeneous networks with or without SDN approach. Concerning mobility management between different SDN administration, it would be interesting to standardize some SDN mobile services. Until some mobility services will be standardized, PMIP would also be useful.

The proposed solution, based on OpenFlow protocol for communicating between control plane and data plane, leading to several advantages: It is a network-based mobility solution without participation of the MN, which is easy to use in real network and suitable for localized domain network; It avoids

creating the tunneling, avoids transferred tunneling packet overhead, decreases the handover latency time, and mitigates the percentage of packet loss.

Although, our solution improves somewhat the PMIP mobility service, there are still performance problems due to the unreliability of the final network segment. Software processing capabilities at the heart of the network that offers the SDN could be implemented to address these problems. In the future work, we are planning to handle the suffered packet loss by using cache scheme.

REFERENCES

- [1] P. Georgopoulos, M. Broadbent, and N. Race. Open Call Deliverable OCC-DS2.1 Design of a Multi-site OpenFlow-assisted Video-on-Demand Distribution Architecture (CEOVDS), GANT, May 2014.
- [2] R. Koodli. Mobile IPv6 Fast Handovers, RFC 5568, July 2009.
- [3] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical Mobile IPv6 (HMIPv6) Mobility Management, RFC 5380, October 2008.
- [4] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6, RFC 5213, August 2008.
- [5] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks (ONF White Paper), April 2012.
- [6] Cengiz Alaettinoglu. Software Defined Networking, PACKET DESIGN, 2013.
- [7] Open Networking Foundation. OpenFlow Switch Specification Version 1.2 (Wire Protocol 0x03), December 2011.
- [8] Open Networking Foundation. OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04), June 2012.
- [9] Open Networking Foundation. OpenFlow-enabled Transport SDN (ONF Solution Brief), May 2014.
- [10] R. Hinden, and S. Deering. IP version 6 Addressing Architecture, RFC 4291, February 2006.
- [11] J. Postel. Transmission Control Protocol, RFC 793, September 1981.
- [12] S. Kim, H. Choi, P. Park, S. Min, and Y. Han. OpenFlow-based Proxy Mobile IPv6 over Software Defined Network (SDN), CCNC 2014, January 2014.
- [13] S. M. Raza, Dongsoo S. Kim, and H. Choo. Leveraging PMIPv6 with SDN, IMCOM (ICUIMC) '14, January 2014.
- [14] Y. Wang, and J. Bi. A Solution for IP Mobility Support in Software Defined Networks, ICCCN 2014, August 2014.
- [15] Mininet: An Instant Virtual Network on your Laptop (or other PC). Available: <http://mininet.org>
- [16] UMIP: Mobile IPv6 and NEMO for Linux. Available: <http://umip.org>
- [17] EURECOM Sophia Antipolis: OPENAIRINTERFACE PROXY MOBILE IPV6 (OAI PMIPv6). Available: <http://openairinterface.eurecom.fr/openairinterface-proxy-mobile-ipv6-oai-pmipv6>
- [18] Ryu 3.18 documentation: WELCOME TO RYU THE NETWORK OPERATING SYSTEM (NOS). Available: <http://ryu.readthedocs.org/en/latest/index.html>
- [19] iPerf - The network bandwidth measurement tool. Available: <https://iperf.fr/>
- [20] Wireshark. Available: <https://www.wireshark.org/#learnWS>
- [21] tshark - Dump and analyze network traffic. Available: <https://www.wireshark.org/docs/man-pages/tshark.html>
- [22] The GEANT Open Call Deliverable OCC-DS2.1, Design of a Multisite Open-flow-assisted Video-on-Demand Distribution architecture (CEOVDS), 20 May 2014.