IEICE TRANS. INF. & SYST., VOL.E100–D, NO.5 MAY 2017

INVITED PAPER *Special Section on the Architectures, Protocols, and Applications for the Future Internet*

NAPT-Based Mobility Service for Software Defined Networks

Shimin SUN[†a)], **Li HAN**[††], **Xianshu JIN**[†††], ***Nonmembers***, ***and*** **Sunyoung HAN**[†††b)], ***Member***

SUMMARY For IP-based mobile networks, efficient mobility management is vital to provision seamless online service. IP address starvation and scalability issue constrain the wide deployment of existing mobility schemes, such as Mobile IP, Proxy Mobile IP, and their derivations. Most of the studies focus on the scenario of mobility among public networks. However, most of current networks, such as home networks, sensor networks, and enterprise networks, are deployed with private networks hard to apply mobility solutions. With the rapid development, Software Defined Networking (SDN) offers the opportunity of innovation to support mobility in private network schemes. In this paper, a novel mobility management scheme is presented to support mobile node moving from public network to private network in a seamless handover procedure. The centralized control manner and flexible flow management in SDN are utilized to provide network-based mobility support with better QoS guarantee. Benefiting from SDN/OpenFlow technology, complex handover process is simplified with fewer message exchanges. Furthermore, handover efficiency can be improved in terms of delay and overhead reduction, scalability, and security. Analytical analysis and implementation results showed a better performance than mobile IP in terms of latency and throughput variation.

key words: *software defined networking (SDN), mobility management, private networks, network address port translation (NAPT)*

1. Introduction

In Internet-of-Things (IoT), most devices are expected to access Internet and may expect to provide network-accessible service even during roaming. On account of high QoS requirements of mobile users, it is essential to provide uninterrupted communication even mobile node changes its network attachment. However, the diversity of IoT devices and mobility patterns limit the deployment of existing mobility management schemes.

Most of the current studies focus on the mobility management in IPv6 networks, since it is much more facile than in IPv4 network due to abundant IPv6 addresses and its auto-configurable characteristics [1]–[3]. Although IPv6 have already been extensively used in some scenarios, a great proportion of commercial networks are still IPv4-based networks, such as home networks, enterprise networks, campus networks, and wireless sensor networks, which are included as the units in IoT. In those networks, the IP addresses used for communication of all local hosts with external networks are based on Network Address Translation (NAT) or Network Address Port Translation (NAPT) mechanisms [4].

To provide roaming service, Mobile Node (MN) should have its own public IP address as its identifier in home and foreign networks. Most of IP mobility management approaches, such as Mobile IP (MIP) [5] and Proxy Mobile IP (PMIP) [6], devote to the scenarios that all those networks are public networks with sufficient public IP address for each host. Some approaches [7]–[9] tend to encapsulate packets with private IP address in tunnel, which degrades data transmission efficiency due to the tunneling overhead.

We try to break away from traditional mobility management schemes with a new architecture evolving from Software Defined Networking (SDN) [10], [11]. All or most of the network devices (e.g., routers, switches) are OpenFlow protocol [12] supported. To support mobility in private networks, SDN controller needs to manage NAT service for terminal networks. The issue is that current version of OpenFlow protocol are still with deficiency of NAT functions. Therefore, we attempt to realize NAT service by manipulating flow tables of OpenFlow devices (e.g., OpenFlow Switches, OFSs). Two restrictions need to be solved to realize NAT service in SDN. (1) Connections can only be originated inside the NAT domain. External networks cannot connect to inside computer proactively (e.g., remote desktop). (2) IP checksum needs to be recalculated after the change of IP address. If TCP header is encrypted (for instance, IPSec transport mode), TCP checksum field cannot be modified. Any modification to the packet header results in the failure of authentication and abandon of the packet.

In this paper, a novel mobility management scheme is presented for the scenario that MN moves from a public network to a private network. The objective is to offer a universal mobility support without involving the participation of MNs in handover process. This is much suitable for various IP-based IoT devices, particularly for resource-constraint MNs which may incapable to deploy mobility functions.

The rest of this paper is organized with following structure: Section 2 is an overview of the literatures of mobility management. Section 3 presents the proposed architecture and NAT/NAPT realization methods. Section 4 elaborates the principles of mobility support. Prospective of perfor-

Manuscript received July 21, 2016.
Manuscript revised January 3, 2017.
Manuscript publicized February 13, 2017.

[†]The author is with School of Computer Science & Software Engineering, Tianjin Polytechnic University, Tianjin, 300387, China.
[††]The author is with School of Computer & Communication Engineering, Tianjin University of Technology, Tianjin, 300384, China.
[†††]The authors are with Department of Computer Science & Engineering, Konkuk University, Seoul, 143–701, Korea.
a) E-mail: sunshimin@tjpu.edu.cn
b) E-mail: syhan@cclab.konkuk.ac.kr (Corresponding author)
DOI: 10.1587/transinf.2016NTI0001

mance is evaluated by comparison with other mechanisms in Sect. 5, and conclusion is given in Sect. 6.

2. Related Work

Although various mobility management approaches have already been proposed, most of them have drawbacks in different aspects. For example, MIP requires MN to take part in mobility functions with deployed new host stack, while PMIP introduces tunneling which results in traffic overhead. Mobile PPC [13] conceives MN's IP layer to resolve IP address variation and notifies Corresponding Nodes (CNs) about new address. It also proposed a NAT traversal scheme on MN to support MN moves through private networks. In Mobile PPC, all the functions were put on MN with kernel modification. REBEKAH-IP server [14] was designed to integrate Home Agent (HA) function and Foreign Agent (FA) function to form MIP. RPX IP-in-FQDN (REBEKAH-IP with Port Extension in Fully Qualified Domain Name) tunneling is used to support multiple MNs with same IP address moving to the same foreign network. NTMobile (Network Traversal with Mobility) [15] creates a tunnel route between NTM terminals (between MN and CN) and applications use virtual IP addresses to achieve a continuous communication during handover. The drawback of RPX IP-in-FQDN is to deploy HA/FA and tunneling, while NTMobile needs MN to join the handover and builds tunneling.

Recently, there is a tendency to applying SDN technology to mobile Internet. Some on-going researches pay attention to the deployment of SDN to cellular networks [16], [17]. OpenRoads [18] provide a wireless extension for OpenFlow to improve robustness of data delivery during handover by applying multicast. The problem is that OpenRoads require MNs have multiple interfaces, so that the flows can be multicast to MN to provide lossless handover. It is impractical to have multiple interfaces for many IoT nodes, especially capacity restricted sensor nodes. Different from OpenRoads, we prefer to realize mobility support using basic OpenFlow functions, rather than to introduce new features to OpenFlow protocol. Pupatwibul et al. [19] proposed to enhance MIP using OpenFlow, where OFSs just simply realize the functions of HA and FA. Latest researches [20]–[22] are inclined to extend SDN paradigm to mobility management by keeping up-to-date identifier-to-locator mapping on OFSs. Each MN need to maintain its own identifier (public IP address from its home network (HoA)) as well as a locator (address prefix of MN's first-hop OFS of visited network). In our previous research, a secure mobility support approach was presented for IPv6 networks based on OpenFlow [23].

Most of above proposals were directed at MN with public IP address and moves among public networks. Tunneling based approaches lead to large traffic overhead or non-optimal route. Efficient mobility management scheme is necessary for the scenario that MN moves among public networks and private networks. In this paper, we propose a novel NAPT-based mobility service using OpenFlow technology.

3. NAPT-Based Mobility Support Scheme

Instead of complying with traditional handover procedure, we attempt to rethink the whole process and redesign mobility related functions from movement detection to binding updates. Traditional handover process in Fig. 1 (e.g., MIP) is generally completed in eight steps, in which *Agent Discovery* is the most time-consuming step. Because MN needs to wait a random time to receive agent advertisement. The proposal try to achieve the handover within five steps without time-consuming step in Fig. 2.

MIP handover process is explained as follows. When MN visits a foreign network, FA assigns a CoA to MN. MN starts *Agent Discovery*, obtains CoA, and launches *Duplicated Address Detection* as well as *Authentication* with FA. After that, MN registers its current location (CoA) with FA and HA during *Registration*. Finally, HA set up a reciprocal tunnel to the CoA to route packets to MN as it roams.

For SDN-based mobility management, three factors should be confirmed beforehand. First, SDN controller totally manages handover procedure on behalf of MNs. Second, movement detection is achieved by the controller receiving the first packet of MN from a foreign network. The packet is encapsulated in *Packet_in* message and forwarded by the OFS of visited network to the controller. MN's MAC and Home Address (HoA) in this packet can be naturally used as the identifier for authentication. Third, ARP Table of MN should be modified to let all the outgoing packets transmitted to the visited OFS by defining automatic ARP reply mechanism.

Centralized control feature of SDN facilitates the possibility to abnegate MIP architecture. The proposed mechanism reduces handover delay by shifting some pivotal steps, such as agent discovery and IP address configuration, which are the most time-consuming steps. Some steps in Fig. 1 can be abandoned according to the new design as in Fig. 2. MNs are not required to participant in mobility related functions, and transparent to its movement in network layer.

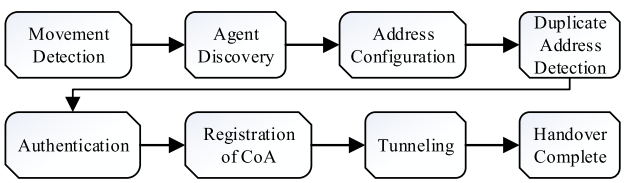

Fig. 1 Traditional handover process

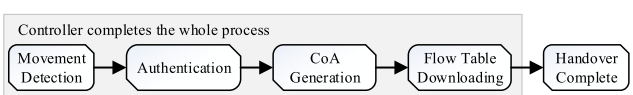

Fig. 2 Proposed handover process in SDN

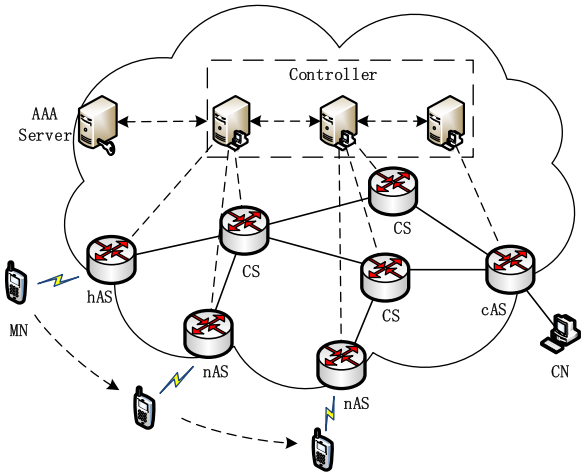

Fig. 3 Mobility management architecture

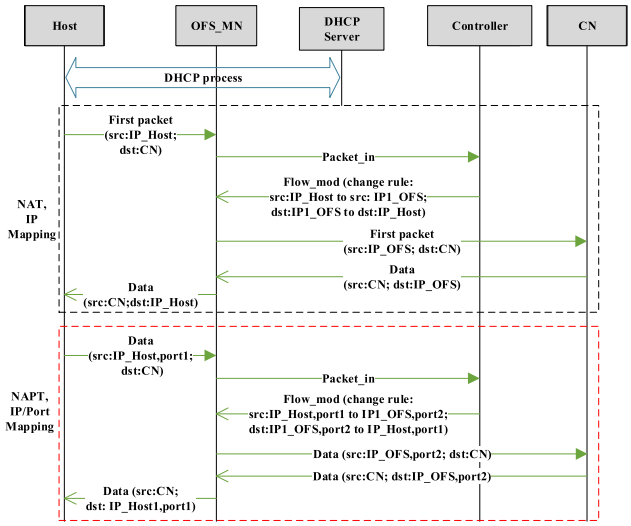

Fig. 4 NAT/NAPT realization method

3.1 SDN-Based Mobility Management Architecture

Agent or anchor based mobility schemes usually lead to non-optimal routing, traffic overhead, and requires extra agents. On the contrary, we tends to manage mobility by SDN controller manipulating OFSs with optimal routing, low traffic overhead and no agents. To support global mobility, ISPs should negotiate for the coordination among their controllers to provide large-scale and heterogeneous collaborative mobile service.

Three network entities are included in the scheme: SDN controllers with Mobility Management Entity (MME), OFSs, and AAA server. Controller is the basement to realize network functions, such as link discovery, topology manager, device manager, as well as MME. OFSs are responsible for data forwarding and data manipulation following the rules preserved in flow tables. AAA server is deployed for the distribution and update of certifications of MNs and controllers. Mobility management architecture is illustrated in Fig. 3, where the dotted lines and solid lines indicate control flow and data flow respectively. Access Switch (AS) is the OFS allocated in terminal network with wireless service. hAS is the AS in MN's home network, while nASs is the one in foreign networks and cAS is the AS of CN's network. Cross Switches (CSs) are the OFSs located in backbone network to offer data forwarding and routing service.

In proposed scheme, functionalities of data plane is relatively simple. All packets are forwarded by OFSs following flow table entries that defined by controller. When MN's movement detected, the controller downloads specific flow table entries to relative OFSs to keep continuous connections between MN and CNs.

3.2 NAT/NAPT Realization Method

MN needs to use HoA all the way before and after handover to keep on-going sessions uninterrupted. The problem is that HoA cannot be used in visited networks due to the ingress filter of routers [24] on outgoing route. CoA needs to be assigned to MN as the locator for data delivery to external networks. The CoA need to be changed back to MN's HoA before data reaches CN. The process is achieved by controller manipulating OFSs using pure OpenFlow functions. Take Fig. 3 as an example, MN obtains HoA when it at the network of *hAS*. Once MN visits *nAS*, MN still uses HoA. The controller assigns a CoA to MN for outgoing messages. When messages reach cAS of CN, cAS performs the address translation for MN from CoA to HoA.

For private networks, DHCP server is assumed deployed for IP allocation in LAN. To ensure DHCP service works as normal, a flow table entry of OFS is added in previous to forward DHCP discovery messages (from IP: 0.0.0.0 to IP: 255.255.255.255) to DHCP server.

The realization of NAT/NAPT service is depicted in Fig. 4 and explained as follows. Initially, a local host obtains an IP address from DHCP server. For NAT service, an IP pool is maintained by OFS_MN. Source address of all packets from local host to external networks is modified to one of the OFS_MN's IP addresses, while destination address of all packets from external networks to local host is modified to host's private IP address. For NAPT, both IP address and TCP/UDP port of on-going sessions of local host are matched to OFS_MN's IP and port. The translation of IP/port is realized by controller manipulating the flow entries of the OFS, which requires the controller to monitor the available TCP/UDP ports of OFS_MN.

3.3 Data Structure in MME

Controllers in a federation share a MN Info. Table (MIT) (Table 1), which contains the basic information of MN, such as HoA, MAC, public key, IP of hAS, and current attached AS. The controller of MN's home network (home controller) maintains a Session Table (ST) (Table 2), which contains the information of MN's active sessions when it

Table 1 MN Info. Table (MIT) at federated controllers

MN's home address
MN's MAC address
MN's public key
MN's home AS
MN's current AS

Table 2 Session Table (ST) at home controller

MN's home address
MN's port number
CN's IP address
CN's port number

Table 3 Session Mapping Table (SMT) at visited controller

MN's home address
MN's port number
nAS's IP address
nAS's port number
Anchor point address

was at home network. This information is used in fast session resumption after handover. The controller of visited network (visited controller) maintains a Session Mapping Table (SMT) (Table 3), which contains the NAPT information for MN in visited network and the IP of anchor point where to revert the packet header to original as from MN. The anchor point is a switch on the route between MN and CN, which should be on the border or outside of access domain of nAS to avoid router's ingress filter problem.

4. Mobility Support Mechanism

4.1 Before Mobility Service

MN registers its basic information (IP/MAC, public key, and hAS) to controller before acquiring the provision of mobility service. MIT (Table 1) need to be shared with other controllers and updated in the federation for global mobility support.

4.2 Movement Detection Method

When MN attaches to a new network (nAS), it will send packets as normal rather than launch DHCP discovery immediately, since it is ignorant of attachment alteration. nAS encapsulates the first packet of MN in Packet_in message and transmits to controller. After received the Packet_in, controller decapsulates it and checks MIT to make sure whether the source IP of the packet is in the table or not. If in, the controller launches authentication using MN's public key. Upon the success of authentication, mobility service to MN is supplied by running NAPT and mobility support mechanism.

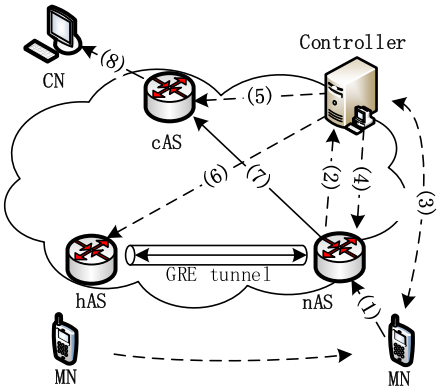

Fig. 5 Mobility support when data initiated from MN

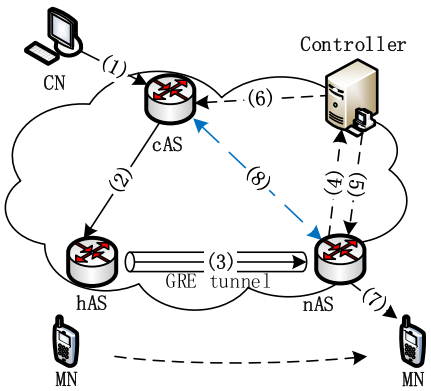

Fig. 6 Mobility support when data initiated from CN

4.3 Mobility Support Mechanism

After attachment detection, visited controller retrieves MN's ST from MN's home controller. Then, it builds a SMT for MN as well as a tunnel between hAS and nAS for information exchange between MN and its home network. The tunnel is for data delivery from home network to MN or from MN to home network. An intermediate CS on the path of MN and CN is selected for IP header conversion on behalf of MN. Mobility support can be completed in two phases through the operations (Fig. 5 and Fig. 6). The operation for data initiated from MN in visited network is illustrated in Fig. 5, while the data initiated from CN to MN's home network is illustrated in Fig. 6 with eight steps.

In Fig. 5, when the first packet from MN reaches nAS (1), nAS encapsulates it in Packet_in message and sends to its controller (2). The controller checks MIT and initiates authentication with MN if MN is in the table (3). If authentication succeed, the controller downloads proper flow table entries to nAS, cAS, and hAS using Flow_mod messages. For nAS, MN's IP/Port is modified to nAS's IP/Port (4). For cAS, nAS's IP/Port is changed back to MN's IP/Port (5). For hAS, a GRE tunnel between hAS and nAS is built for message exchange (e.g. ARP, ICMP, and data from external network reaches hAS) (6). Upon receiving Flow_mod

message, nAS immediately modifies IP header of packets belong to the same session and sends to CN passing by cAS (7). cAS performs header translation between MN's IP/Port and nAS's IP/Port according to the rules defined in Flow_mod message (8).

A new session may be originated from CN to MN after MN's handover. In this situation, data is still forwarded to hAS. The operation is depicted in Fig. 6. First, data is forwarded to hAS by cAS (1) (2). hAS forwards the data to nAS through established GRE tunnel (3). Since the data is from a new session, there is no matched flow entry in nAS. nAS sends Packet_in message to controller for further operation (4). The controller inserts a SMT entry and issues flow rules to nAS and cAS (5) (6). nAS performs NAPT function for the session and sends data to MN (7). Finally, data of the session is transmitted between cAS and nAS directly (8).

4.4 Automatic Reply Messages

To let MN works as in its home network, some message exchanging through the tunnel between hAS and nAS is required to keep MN online. Basically, ARP requests for MN need to be replied and ARP requests from MN to other hosts need to be sent to MN's home network. If DHCP service is deployed in home network, DHCP request/ACK messages need to be transmitted through the tunnel. If multicast group service is deployed, IGMP messages should also be transmitted through the tunnel.

5. Analytical Evaluation and Implementation

5.1 Analytical Evaluation of Handover Latency

For MIP, two phases lead to the major handover delay: movement detection and registration, which can be presented by Eq. (1).

$$T_{MIP\_handover} = T_{move\_detect} + T_{reg} \tag{1}$$

Where T_{move_detect} is the delay of movement detection and T_{reg} is MN registration delay. T_{reg} is determined by the *RTT* between MN and HA, and normally less than 100ms in WAN. Lazy Cell Switching (LCS) and Eager Cell Switching (ECS) [25] are two major movement detection algorithms in MIP. LCS acts handover until the primary network is confirmed unreachable (lifetime of its IP address reaches 0), while ECS initiates handover every time agent advertisement with new network prefix received. The drawbacks are that LCS causes high handover delay and ECS performs poor if MN moves back and forth in two networks.

HA/FA advertisement interval should be shorter than 1/3 of the lifetime and is recommended one advertisement per second in [5]. Assuming the agent advertisement lifetime is $3x$ and interval is x, the delay caused by above two algorithms can be estimated by their mean values (Eq. (2) and Eq. (3)).

$$T_{move\_detect\_LCS} = 2.5x \tag{2}$$

$$T_{move\_detect\_ECS} = 0.5x \tag{3}$$

If $x = 1$ second, the experiments results showed that LCS causes UDP delay for about 6s and TCP for 10s, while ECS causes UDP delay for 3s and TCP for 6s [26]. The throughput of TCP traffic decreases sharply after handover due to slow start algorithm.

For proposed mobility support mechanism, the delay can be indicated by Eq. (4).

$$T_{SDN\_handover} = T_{move\_detect} + T_{auth} + T_{flow\_mode} \tag{4}$$

Where T_{move_detect} is one-way delay from MN to controller, T_{auth} is the *RTT* between controller and MN, while T_{flow_mode} is the one-way delay between nAS and the controller. Because MN and nAS is one-hop connection, delay between them is small. The total delay can be estimated by two times *RTT* between MN and controller (Eq. (5)).

$$T_{SDN\_handover} \approx 2 \times RTT_{MN\text{-}controller} \tag{5}$$

The controller should be in the same domain of nAS. Hence, delay between the controller and nAS is also relatively small, not more than 100ms commonly.

Above all, the delay caused by handover in SDN-based scheme is much lower than in MIP scheme. Low delay of handover conduces to smaller throughput variation and lower packet loss.

5.2 Network Environment of Implementation

Since MIP is not supported in SDN simulation tool Mininet [27] yet, we built a testbed using several virtual machines with custom topology and link status (Fig. 7) using Virtualbox. In SDN testbed, S1, S2, S3, S4 and S5 were OFSs implemented by OpenFlow vSwitch [28]. SDN controller was deployed by python-based programmable software Ryu [29]. MIP was implemented on the same network using the code of dynamic mobile IP [30], where S1 was replaced by HA, while S2 and S3 were replaced by FAs.

In the experiment, *ping* was used for delay test between MN and CN. Test interval was set to 0.2s. *Iperf* [31] was used to generate TCP or UDP traffic for the test of throughput. Link status was defined by Linux traffic control tool *tc*. *tc* can be used to set link latency and bandwidth. The delay

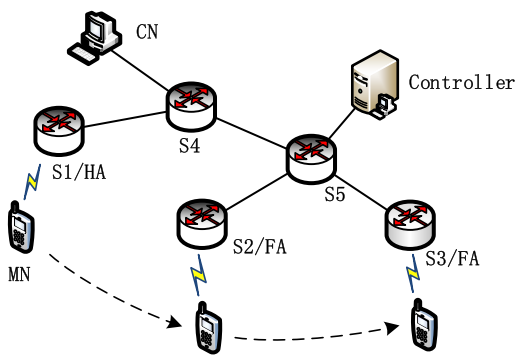

Fig. 7 Network environment for implementation

of each link was set to 20ms and the bandwidth was set to 10mpbs. The test time was set to 40s with two handovers executed at the time 10s and 30s.

5.3 Performance Evaluation

The objective of mobility solutions is to reduce handover delay and traffic overhead. For the evaluation of mobility management approaches, variation of delay and throughput during handover process is the most considerable factor. Several tests are carried out based on top of the afore-presented testbed. Latency and throughput variation of MIP and proposed schemes are compared in the evaluation.

5.4 Comparison with MIP

The comparison between SDN-based mobility and MIP are shown in Fig. 8 and Fig. 9 in terms of throughput and *RTT*.

Figure 8 shows the variation of throughput for a session between MN and CN. In this experiment, 10mbps UDP traffic (packet size 1500 bytes) was transmitted during handover process. The result reveals that the throughput of MIP decreases sharper and recovers slower than SDN-based approach. This is caused by the longer handover delay of MIP. The degradation of traffic rate after handover results from the tunneling.

Figure 9 illustrates the variation of delay during handover process. *RTT* values of both approaches raise to a higher value rapidly when MN changes network attachment. Proposed approach shows better restoration speed than MIP

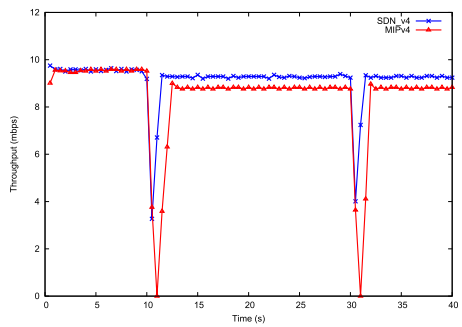

Fig. 8 Throughput variation during handover

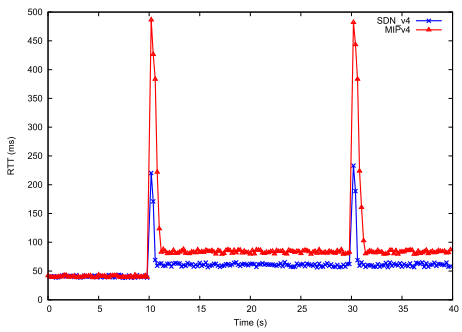

Fig. 9 RTT variation during handover

in same experiment scenario. After handover in MIP, the delay between MN and CN regresses to a higher level due to triangular routing of MIP. The proposed approach avoids triangle routing by changing flow entries of related OFSs to setup a new optimal path between MN and CN after handover. The new path follows basic Internet routing methods, such as OSPF or RIP, which are commonly used to find the optimal path.

The proposed approach also performs better than MIP In some other aspects, such as packet loss, traffic overhead and signaling cost, which is because the abnegation of tunneling for data delivery and few messages exchange in handover process. Packet loss was tested during the experiments for throughput. For proposed approach, the number of packet loss is about 1030 for the first handover and 820 for the second handover. For MIP, the number of packet loss is about 7000 for both handover since they has same distance between FA and HA (between MN and CN) after handover.

6. Conclusion

This paper presented a SDN-based mobility management scheme for node movement through private IP networks. The mobility functions were realized in SDN controller. Controller manipulates OFSs to achieve mobility support without introducing new entities to network or new functions to OpenFlow protocol. The evaluation results indicate that the proposed scheme reveals better performance than MIP.

Although the proposal provides mobility service for IP-based devices in the specific cases, more scenarios and test environments could be considered in future work. We will focus on the mobility support for MN with private IPv4/IPv6 addresses among private/public networks as well as other network environments to offer various mobility support service.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) under B0126-16-1078.

References

[1] C. Perkins, D. Johnson, and J. Arkko, “Mobility support in IPv6,” IETF RFC 6275, July 2011.
[2] S. Lee, H. Latchman, and B. Park, “Efficient handover scheme of proxy mobile IPv6 in wireless local area networks,” Int. J. Multimedia and Ubiquitous Engineering, vol.5, no.2, April 2010.
[3] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy mobile IPv6,” IETF RFC 5213, Aug. 2008.
[4] P. Srisuresh and K. Egevang, “Traditional IP network address translator (traditional NAT),” IETF RFC 3022, Jan. 2001.
[5] C. Perkins, “IP mobility support for IPv4, revised,” IETF RFC 5944, Nov. 2010.
[6] K. Leung, G. Dommety, P. Yegani, and K. Chowdhury, “WiMAX forum/3GPP2 proxy mobile IPv4,” IETF RFC 5563, Feb. 2010.

[7] H. Levkowetz and S. Vaarala, "Mobile IP traversal of network address translation (NAT) devices," IETF RFC 3519, April 2003.
[8] G. Montenegro, "Reverse tunneling for mobile IP, revised," IETF RFC 3024, Jan. 2001.
[9] A. Idoue, H. Yokota, and T. Kato, "Proposal of hierarchical mobile IP supporting private addresses utilizing NAT function and its implementation on Unix operating system," IEICE Trans. Commun., vol.E84-B, no.12, pp.3155–3165, Dec. 2001.
[10] ONF White Paper, "Software-defined networking: the new norm for networks," Open Networking Foundation, April 2012.
[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.69–74, April 2008.
[12] OpenFlow Networking Foundation, "OpenFlow switch specification, version 1.5.1," March 2015.
[13] H. Suzuki, K. Terazawa, and A. Watanabe, "Implementation of NAT traversal for mobile PPC with the principle of hole punching," Proc. IEEE Region 10 Conference (TENCON 2009), Singapore, pp.1–6, Jan. 2009.
[14] S. Rattananon, B. Landfeldt, A. Seneviratne, and P. Chumchu, "Mobility support in private networks using RPX," Proc. IEEE Conf. Local Computer Networks 30th Anniversary (LCN '05), Sydney, NSW, pp.729–739, Nov. 2005.
[15] K. Kamienoo, H. Suzuki, K. Naito, and A. Watanabe, "Development of mobile communication framework based on NTMobile," Seventh Int. Conf. Mobile Computing and Ubiquitous Networking (ICMU), Singapore, pp.27–32, Jan. 2014.
[16] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A.V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: a survey," Mobile Networks and Applications, vol.20, no.1, pp.4–18, Feb. 2015.
[17] N.A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: a survey," ACM Computing Survey, vol.47, no.2, Jan. 2015.
[18] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: empowering research in mobile networks," ACM SIGCOMM Computer Communication Review, vol.40, no.1, pp.125–126, Jan. 2010.
[19] P. Pupatwibul, A. Banjar, A.A. Sabbagh, and R. Braun, "Developing an application based on OpenFlow to enhance mobile IP networks," IEEE 38th Conf. Local Computer Networks Workshop on Wireless Local Networks, Sydney, Australia, pp.936–940, Oct. 2013.
[20] Y. Wang, J. Bi, and K. Zhang, "Design and implementation of a software-defined mobility architecture for IP networks," Mobile Networks and Applications, vol.20, no.1, pp.40–52, Feb. 2015.
[21] Y. Wang and J. Bi, "A solution for IP mobility support in software defined networks," 23rd Int. Conf. Computer Communication and Networks (ICCCN), Aug. 2014.
[22] S.-M. Kim, H.-Y. Choi, P.-W. Park, S.-G. Min, and Y.-H. Han, "OpenFlow-based proxy mobile IPv6 over software defined network (SDN)," IEEE Consumer Communications and Networking Conference (CCNC 2014): Mobility Management in the Networks of the Future World, Jan. 2014.
[23] S. Sun, L. Han, and S. Han, "Secure IP mobility support in software defined networks," Mobile and Wireless Technology 2015 (ICWMT2015), Springer Berlin Heidelberg, vol.310, pp.127–136, June 2015.
[24] P. Ferguson and D. Senie, "Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing," IETF RFC 2827, May 2000.
[25] Q. Zhao, L. Feng, Z.C. Li, and J. Yang, "Movement detection delay analysis in mobile IP," Computer Communications, vol.28, no.5, pp.550–556, March 2005.
[26] Y. Min-hua, W. Zhe-wei, Z. Hui-min, and L. Yu, "Performance analysis of TCP/UDP during mobile IP handoffs," Proc. Int. Conf. Info-tech and Info-net, Beijing, China, vol.2, pp.724–729, Oct. 2001.
[27] K. Karamjeet, S. Japinder, and S.G. Navtej, "Mininet as software defined networking testing platform," Int. Conf. Communication, Computing & Systems, pp.139–142, Ferozepur, Punjab, India, 2014.
[28] Open vSwitch (OVS), http://openvswitch.org/
[29] Ryu SDN Framework, http://osrg.github.io/ryu/
[30] Dynamics Mobile IP, http://dynamics.sourceforge.net/
[31] iPerf-The TCP, UDP and SCTP network bandwidth measurement tool, https://iperf.fr/

Shimin Sun received M.S. and Ph.D Degree in Computer & Information Communication Engineering from Konkuk University of Korea in 2009 and 2016 respectively. He is currently an instructor in Tianjin Polytechnic University, China. His research interests include Software Defined Networking, Mobility Management, Wireless Sensor Networks, Information Centric Networking, and Future Internet.

Li Han received M.S. and Ph.D Degree in Computer & Information Communication Engineering from Konkuk University of Korea in 2012 and 2016 respectively. She is currently an instructor in Tianjin University of Technology, China. Her research interests include QoS routing, Software Defined Networking, and Network Security.

Xianshu Jin received M.S. Degree in Computer & Information Communication Engineering from Konkuk University of Korea in 2009. She is currently a Ph.D. student in Dept. of Computer & Information Communication Engineering at Konkuk University, Korea. Her research interests include Internet of Things, Distributed Mobility Management, Multicast, and Software Defined Networking.

Sunyoung Han received M.S. degree and Ph.D. degree in Computer Science from Korea Advance Institute of Science & Technology in 1979 and 1988 respectively. Since 1981, he has been working in Dept. of Computer Science & Engineering in Konkuk University as a Professor. His research interests include Multicasting, Wireless Sensor Networks, Internet of Things, and Future Internet.