SDN-Based Distributed Mobility Management for 5G Networks

Tien-Thinh Nguyen, Christian Bonnet, and Jérôme Härri
EURECOM, Department of Mobile Communications
Sophia-Antipolis, France
Email: {Tien-Thinh.Nguyen, Christian.Bonnet, Jerome.Haerri}@eurecom.fr

Abstract—**Software-Defined Networking (SDN) is transforming the networking ecosystem. SDN allows network operators to easily and quickly introduce new services and flexibly adapt to their requirements, while simplifying the network management to reduce the cost of operation, maintenance and deployment. On the other hand, mobility is a key aspect for the future mobile networks. In this context, Distributed Mobility Management (DMM) has been recently introduced as a new trend to overcome the limitations of the today's mobility management protocols which are highly centralized and hierarchical. Driven from the fact that DMM and SDN share the same principle in which the data and control plane are decoupled, we propose a DMM solution based on SDN architecture called S-DMM. This solution offers a lot of advantages including no need to deploy any mobility-related component at the access router, independence of the underlying technologies, and per-flow mobility support. On one hand, the numerical results prove that S-DMM is more scalable than the legacy DMM. On the other hand, the experiment results from a real implementation show that S-DMM comes at no performance penalty (in terms of handover latency and end-to-end delay) compared to legacy DMM, yet at a slightly better management cost, which makes S-DMM a promising candidate for a mobility management solution in the context of 5G networks.**

Index Terms—**IP Mobility Management, Distributed Mobility Management, Software-Defined Networking, 5G Networks.**

I. INTRODUCTION

The mobile network operators are now facing with several challenges such as a huge traffic demand with more sophisticated services (thus, posing different connectivity requirements), the need for seamless delivery services across different technologies, and a rapid changing of business environment [1], [2]. Thus, the operators are seeking for innovative solutions to improve their network performance and efficiency, as well as to reduce the costs expended on network operation, maintenance, and new service deployment. Considering the deployment of heterogeneous 5G networks, including a mix of femto and pico cells of different technologies, simplifying the network architecture and optimizing the data transmission costs are driving 5G networks to evolve toward flat architectures.

In line with this trend, the 3GPP[1] proposed such flat optimization techniques as Local IP Access/Selected IP Traffic Offload (LIPA/SIPTO) and IP Flow Mobility (IFOM) [3]. Following the same idea, the Internet Engineering Task Force (IETF) has recently chartered the Distributed Mobility Management (DMM) Working Group[2], which specifies flat IP mobility management architecture separating data and control planes to address the limitations of current centralized mobility management such as sub-optimal routing, scalability, and reliability issues [4], [5]. Flat architectures are designed to improve the data plane yet at the cost of an increased complexity at the control plane. On the other hand, Software-Defined Networking (SDN) together with Network Functions Virtualization (NFV) also decouple data and control planes, and offer a new network architecture particularly designed to improve the management of the control plane, both at the mobile backhaul and core network levels [2], [6]. Accordingly, network operators can reduce the cost expended on network operation, maintenance and service deployment.

Since DMM and SDN have complementarity assets and share the same decoupled data and control plane paradigm, we propose in this paper a DMM solution based on SDN architecture (namely S-DMM). The mobility management intelligence is deployed as a software application on top of the controller, thus taking full advantages of SDN architecture. S-DMM has several salient features, such as: (i) distributed data plane - no need to deploy any mobility-related component at the access router, and mitigation of tunneling overhead and sub-optimal routing; (ii) centralized control plane - full view and control of network switches, and mitigation of tunneling and flow management (e.g., maintenance and keep-alive signaling), which are one of the major issues of the legacy DMM; (iii) per-flow mobility support - management of the mobility of traffic flows rather than nodes; and (iv) network independence of the underlying technologies. The experiment results from a real implementation show that a similar performance as the legacy DMM in terms of handover latency and end-to-end delay can be achieved, yet at a reduced complexity of the control plane. Altogether, S-DMM can be considered as a promising candidate for mobility management in 5G networks.

The rest of this paper is organized as follows. Section II presents the background information related to DMM and SDN. Section III describes the proposed solution regarding its design principles, architecture and operations. Section IV provides the evaluation of the solution including both quantitative and qualitative results. Also, the experiment results from a real

---

[1] Third Generation Partnership Project, http://www.3gpp.org/

[2] IETF DMM Working Group: https://ietf.org/wg/dmm/

978-1-4673-9814-5/16/$31.00

implementation are presented. Finally, section V concludes the paper and provides perspectives for future work.

II. RELATED WORK

A. Distributed Mobility Management (DMM)

Today's mobility management protocols rely on a central entity such as Home Agent (HA) in Mobile IPv6 (MIPv6), and Local Mobility Anchor (LMA) in Proxy Mobile IPv6 (PMIPv6) [7] to maintain the mobile node's (MN) reachability when it is away from home. As a result, they have several major limitations from their centralized and hierarchical nature. For example, centralizing both the control and data plane functions at the central mobility anchor introduces scalability and reliability issues (the central entity represents a bottleneck and single point of failure) [4]. It also leads to sub-optimal paths between the MNs and their corresponding nodes (CNs). Therefore, it affects the overall network performance in terms of routing efficiency and end-to-end delay transmission [4].

To address these limitations, DMM approaches have been introduced. The key concept of DMM is that instead of having a centralized mobility anchor, the mobility management function is distributed among the network entities at the network edge [4]. DMM also offers dynamic mobility features (per prefix granularity). In a DMM domain, the MN gets different prefixes when changing its point of attachment. The MN's flows are anchored (if necessary) at the access router (called Mobility Access Router - MAR) in which the MN's prefix in use is allocated. For the flows anchored at the current MAR, they are routed using the normal Internet routing manner. In case of mobility, these flows can be redirected via the tunnel from the anchor to the current MAR.

Depending on whether the control plane is distributed, there are two different DMM schemes including the partially (P-DMM) and the fully distributed (F-DMM) (see Fig. 1) [8], [9]. More specifically, in the first scheme, the control plane is still centralized while the data plane is distributed. In other words, a central entity (namely Central Mobility Database - CMD) still exists but for the control plane only. Thanks to the central entity, this scheme offers the operators the possibility to fully control the system. Unlike the first scheme, both data and control plane are distributed in F-DMM which eliminates the central entity, however, with a cost of an external mechanism (such as IEEE 802.21 Media Independent Handover Services (MIH)) and additional signaling cost [10]. Hence, the first scheme seems to be a better choice for the network operator.

B. Software-Defined Networking (SDN)

Software-Defined Networking (SDN) [11] is a new and a very promising approach that refers to the ability to control, change and manage the behavior of the network and the network devices in a dynamic manner. It is achieved by decoupling the control plane from the data plane. The control plane is moved to a logical controller which determines how the packets should flow through the network and configures the forwarding plan accordingly. In other words, the network devices act as a simple forwarding hardware instead of running a full stack of routing protocols as in the traditional network. Thus, SDN allows the network to offer almost limitless capabilities for implementing new services which are deployed on top of the network abstraction layer and flexibly adapting to their requirements. Also, the network operator can significantly reduce the cost expended on network operation, maintenance and service development as well.

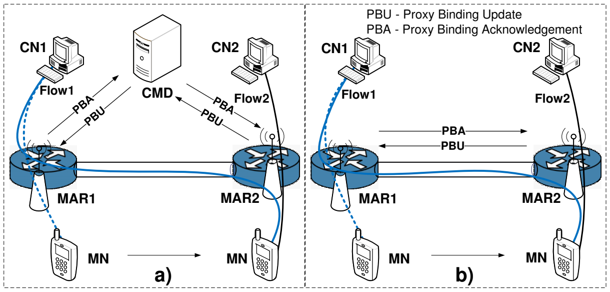

Fig. 1: DMM architecture: a) partially distributed, b) fully distributed.

In the SDN architecture, the controller uses the southbound application program interfaces (APIs) to communicate with the forwarding devices. One typical example of southbound API is OpenFlow protocol[3]. The OpenFlow switches (OFS) perform the packet forwarding functionality based on their flow tables in which each entry determines how the packets will be processed and forwarded. By using OpenFlow protocol, the controller can apply a set of actions (for example, install, update and delete flow entries) to the flow tables depending on a set of conditions. In this architecture, the network applications/services will be deployed as an application on top of the controller. The communication between the controller and the applications will be done by the northbound API.

C. Applying SDN to IP Mobility Management Solution

Already some work has been considered IP mobility management in the context of SDN. In [12], the authors presented their vision of 5G as a group of applications. The authors proposed that mobility management could be provided as a service on top of the SDN controller. In [13], the authors discussed how to apply SDN to the Telecom domain, specifically, focusing on the forwarding data across networking layers. The authors then presented IP mobility as a use-case. Elaborating this idea, the authors in [14] proposed an SDN-based mobile networking with two different types of controller model: single and hierarchical controller. In [15], the authors also suggested that DMM could act as an SDN application. However, it is not clear how it works. In addition, all of them only mentioned the idea of applying SDN in IP mobility management without providing any detailed information. In [16], the authors presented a proposal for route optimization in DMM with SDN. The idea was that, the controller could update the flow tables at the CN's OFS to route the packets directly to the current MN's OFS bypassing the previous MN's OFS. It is noted that after handover, first, the packets have to follow the route from the old to the new OFS. The packets

---

[3] https://www.opennetworking.org/sdn-resources/openflow

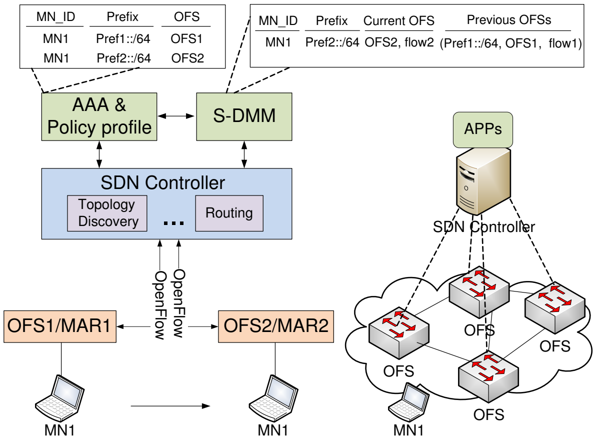

Fig. 2: S-DMM architecture.

are then redirected between the new OFS and the CN's OFS. Consequently, it may lead to a complex process and a high signaling cost.

III. SDN-BASED DMM (S-DMM)

In this section, we first describe the design principles and architecture of the solution. We then explain its operations regarding two different phrases: initial registration and handover.

A. Design Principles and Architecture of S-DMM

To cope with SDN philosophy, we propose an SDN-based DMM (S-DMM) architecture as shown in Fig. 2. In this architecture, DMM is provided as a service deployed on top of a controller. This architecture takes full advantages of SDN for following reasons:

- DMM (and authentication, authorization and accounting (AAA) and policy profile service as well) is provided as a service on top of SDN controller;
- MAR can act as a simple forwarding hardware (a basic OFS) that does not require any mobility-related component;
- Centralized control: The centralized controller allows the operators to efficiently control the network;
- Network-based mobility: The mobility service can be provided to all legacy devices (without any additional mobility-related function);
- DMM service is independent of the underlying technologies and provides seamless handover across different access technologies;
- SDN-based DMM architecture helps to reduce the complexity of the tunnel and flow management mechanism in the conventional DMM solution (e.g., maintenance of the tunnel and keep-alive signaling).

Putting our solution in the context of LTE network architecture, the SDN controller could be co-located with the Mobility Management Entity (MME). Furthermore, the functionalities (control plane) of the other nodes such as Home Subscriber System (HSS), Serving Gateway (S-GW), and Packet Data Network Gateway (P-GW) are implemented as SDN controller applications while the data plane is managed by the basic OFSs. It is noted that in a DMM-like LTE architecture, the MAR plays the role of both the S-GW and the P-GW.

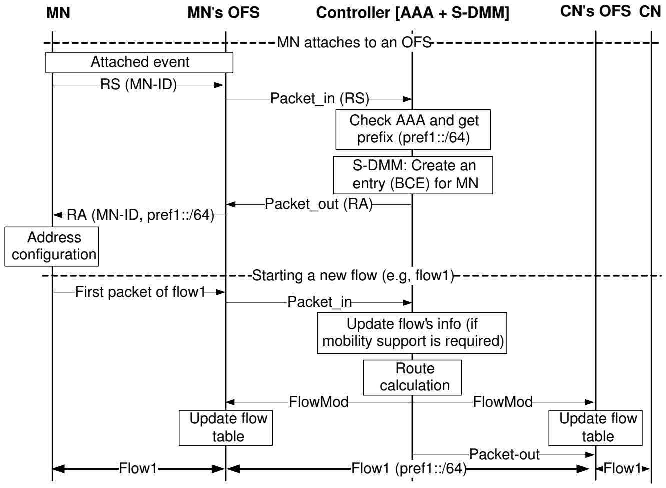

Fig. 3: Signaling flow for initial registration.

B. Operations

1) Initial Registration: The OFS, where the MN attaches for the first time, can detect the new attachment by means of a Router Solicitation (RS) message[4]. Since this OFS does not have any information (forwarding rules) related to the MN, it forwards this message to the controller for inspection by means of a Packet-in message. The AAA, acting as an SDN application, is registered to listen to this message, verifies whether the MN is allowed to use the mobility service. Upon the successful authorization, it returns the MN profile including its identifier (MN-ID) and the allocated prefix (e.g., pref1::/64). S-DMM in turn obtains the MN's related information. S-DMM then creates an entry in its database (Binding Cache Entry or BCE) to keep track the location as well as the MN's related information as shown in Fig. 2. After that, S-DMM creates a Router Advertisement (RA) message including the allocated prefix and sends it to the current OFS via an OpenFlow protocol message (Packet-out). The current OFS forwards this message to the MN which then can configure its address (pref1::MN/64) and start new communication flows with the CN (e.g., flow1). The first packet of the flow will be forwarded to the controller for inspection. Based on a routing service, the controller updates the forwarding rules in the corresponding OFSs (via a message FlowMod). Henceforth, the flow1's packets are handled by the switches without further involvement of the SDN controller. In addition, if mobility support is required for this flow, the flow information is added to the MN's BCE (e.g., source address, destination address, flow label, and traffic class, etc.).

2) Handover Operations: When the MN moves from the old OFS to a new one, similar procedures as described in the registration process are executed to obtain a new prefix

---

[4] It is noted that the attachment detection can be done using either a layer 2 mechanism (e.g., 802.21 MIH and OpenFlow-specified) or a layer 3 one.

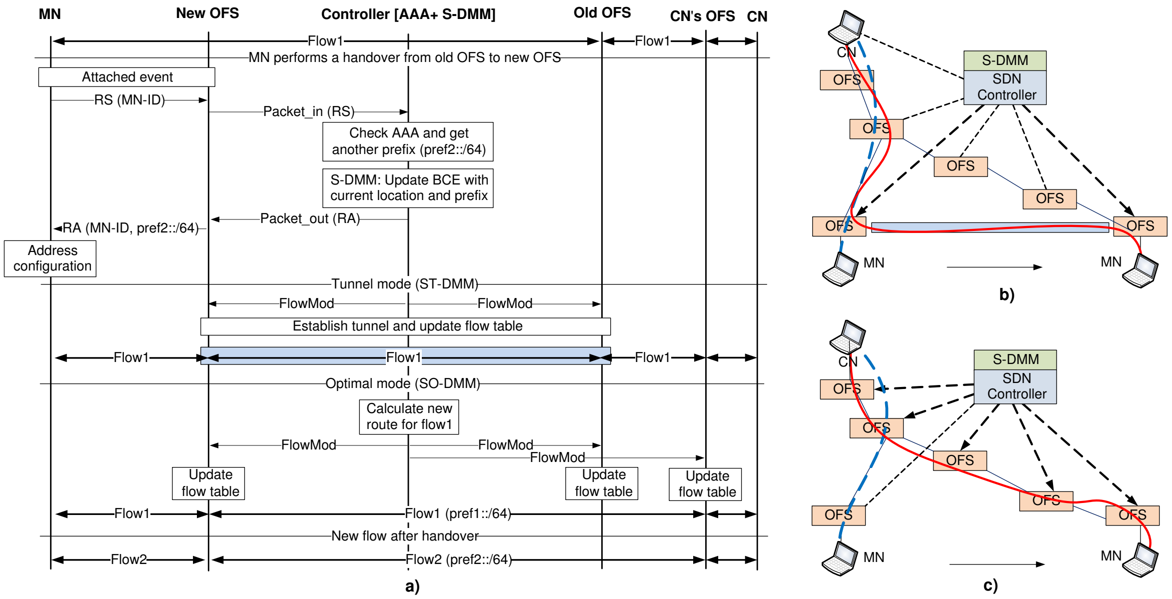

Fig. 4: Handover operations: a) handover signaling, b) ST-DMM, c) SO-DMM.

(pref2::/64) and update the new location of the MN at S-DMM. The major difference is that the RA message is sent to the new OFS including a list of the previous prefixes (with a lifetime set to 0) and the new one (with a lifetime set to a value greater than 0). It is noted that only the prefixes used by the on-going flows, which require mobility support, are included. On one hand, the forwarding rules for the new flows using the new prefix are set similar to that in the previous subsection. On the other hand, the controller updates the forwarding rules for the on-going flows (which require mobility support) at the old OFS and the new one. These flows then will be redirected between the two OFSs using tunneling mechanism (called tunnel mode, or ST-DMM), as similar to the traditional DMM approaches.

In addition, there is another possibility (called optimal mode, or SO-DMM) in which the controller calculates the new route for the on-going flows and populates the new forwarding rules to all the intermediate OFSs along the new route between the new OFS and the CN. In other words, an explicit path between the MN and CN is defined and established by the controller (e.g., following the routing policy as specified in [17]). In this way, tunneling mechanism can be avoided. However, it may lead to a complex task giving the large number of switches/flows to handle. Besides, the simplicity of the first scheme can come at the cost of tunneling overhead and sub-optimal routing. In addition, the two schemes can be dynamically switched up on the service requirements. For example, the optimal mode is likely more suitable for latency-sensitive services while the tunnel mode seems to be better for the packet loss- and interruption- sensitive services.

IV. EVALUATION OF THE SOLUTION

In this section, we first present the qualitative evaluation. The scalability concerns will then be discussed in terms of signaling cost. Finally, the experiment results from a real implementation are introduced regarding the handover latency and end-to-end delay.

A. *Qualitative Evaluation*

Table I provides a summary of the proposed approach characteristics in comparison with the legacy DMM proposals including P-DMM and F-DMM (see Fig. 1). A detail of the legacy DMM proposals is provided in [8], [9]. Sharing the same principle, S-DMM (including both SO-DMM and ST-DMM) and P-DMM have several common characteristics e.g., partially distributed, centralized control, and no-required an external mechanism. These features are different from those of F-DMM approach. On the other hand, S-DMM does not require a specific hardware/software to deploy the functionalities of an MAR as in legacy DMM approach (including both P-DMM and F-DMM). Instead, in S-DMM the functionalities of the MAR/CMD are responsible by an SDN application. As a result, it makes the process of deployment and maintenance easier compared to those in the legacy DMM solutions. Unlike the legacy DMM solutions and the centralized approaches (MIPv6 and PMIPv6) which provide mobility support per-prefix and per-mobile node, respectively, S-DMM offers a per-flow support. Driven from the fact that different applications/services have very different characteristics, requirements and policies, per-flow support can lead to a better quality of service. Furthermore, based on the per-flow support feature, flow mobility can be easily supported. Also, the handover flow can be routed in an optimal manner in case of SO-DMM.

Regarding the tunnel and flow management cost, thanks to a simple mechanism in OFS which allows the switch to delete the flow-related rules after a fixed time interval (i.e., a hard time-out) or a specified period of inactivity (i.e., a soft time-out), signaling overhead for refreshing BCE at S-DMM

TABLE I: Comparison between S-DMM and the legacy DMM proposals.

Metrics/Scheme	**P-DMM**	**F-DMM**	**SO-DMM**	**ST-DMM**
Type of mobility management	Network-based	Network-based	Network-based	Network-based
Distributed level	Partially distributed	Fully distributed	Partially distributed	Partially distributed
Centralized control	Yes	No	Yes	Yes
External mechanism	Not required	Required	Not required	Not required
Specialized S-GW/P-GW/MAR	Yes	Yes	No	No
Granularity level	Per-prefix	Per-prefix	Per-flow	Per-flow
Route optimization	Sub-optimal	Sub-optimal	Optimal	Sub-optimal

(including both SO-DMM and ST-DMM) can be avoided. Also, based on this mechanism, it is easy to verify whether the flow is still alive. It is very important since if there is no on-going flow, the handover process can be considered as simple as the initial registration. Additionally, the legacy DMM approaches can leverage on an external mechanism e.g., Deep Packet Inspection (DPI) to verify that the on-going flows are still alive or not. Yet, it may increase the processing delay, which may play an important role since the expected end-to-end delay in 5G is very small (5ms). Altogether, S-DMM seems to be more flexible and manageable than conventional DMM.

B. Scalability Analysis

Since scalability is one of the main concerns regarding SDN deployment, in this subsection, we investigate the scalability analysis regarding the signaling cost. According to [18] the scalability limitations are not restricted to SDN, traditional control protocol also faces the same challenges. Consequently, in the scope of this document, we consider only the signaling messages generated by mobility-related procedures which may affect the scalability of the solution. In other words, we do not consider the OpenFlow messages for statistics and configurations.

1) Signaling Cost: The signaling cost ($SC_{(.)}$) is the signaling overhead for updating the location ($C^u_{(.)}$) as well as for refreshing the bindings ($C^r_{(.)}$) for the MN. Although different signaling messages have different sizes, we assume that they have the same size for simplicity. Also, the cost for transmitting a signaling message is supposed to be proportional to the distance between the source and the destination. We have:

$$SC_{(.)} = \mu \left(C^u_{(.)} + C^r_{(.)} \right), \tag{1}$$

where μ is the subnet border crossing rate.

The hop-count distances between the entities for performance analysis are defined as follows: h_{mc} is the average number of hops between the MAR and the CMD; between the OFS and the SDN controller as well. h_{mm} is the average number of hops between the anchor and the current MAR. h_{mo} is the average number of hops between the MN and its current MAR/OFS. It is noted that in case of ST-DMM, only two flow modification messages are enough to set up a tunnel between the old and the current OFS for the on-going flows. Whilst in SO-DMM, flow modification message is dedicated to each on-going flow. Thus, $C^u_{(.)}$ can be given by:

$$C^u_{P-DMM} = 2h_{mo} + 2(N_p + 1)h_{mc}, \tag{2}$$

$$C^u_{F-DMM} = 2h_{mo} + 2N_p h_{mm}, \tag{3}$$

$$C^u_{ST-DMM} = 2h_{mo} + 4h_{mc}, \tag{4}$$

$$C^u_{SO-DMM} = 2h_{mo} + 2h_{mc} + 2N_o N_f h_{mc}, \tag{5}$$

where N_p, N_o, N_f is the number of active prefixes, the number of OFSs along the route between the MN and the CN, and the number of on-going flows, respectively. According to [19], N_p is calculated as:

$$N_p = 1 + \frac{\mu}{\delta}, \tag{6}$$

where $1/\delta$ is the mean value of the active prefix lifetime while the MN is visiting a foreign network.

In the legacy DMM approaches, even the MN remained at the same subnet, the signaling for refreshing the bindings is sent periodically when the binding timer expires. For a sake of simplicity, we suppose that the binding cache entry lifetime (T_{BCE}) is identical in case of P-DMM and F-DMM. Thus, the refreshing procedure is executed on average $R_r = \lfloor 1/(\mu T_{BCE}) \rfloor$ times. On the other hand, refreshing BCE is not required in S-DMM (including both SO-DMM and ST-DMM) as mentioned earlier. Instead, one message is needed to notify the controller when the flow is terminated. Thus, in ST-DMM one message is needed while N_f messages are required in SO-DMM. $C^r_{(.)}$ can be therefore given by:

$$C^r_{P-DMM} = 2R_r h_{mc}, \tag{7}$$

$$C^r_{F-DMM} = 2R_r h_{mm}, \tag{8}$$

$$C^r_{ST-DMM} = 1, \tag{9}$$

$$C^r_{SO-DMM} = N_f. \tag{10}$$

2) Numerical Results: In this paper, we consider the case where the MN always moves from MAR/OFS to MAR/OFS as if they were linearly deployed (the user is moving further away from the first attached MAR/OFS and never attaches back to a previously visited MAR/OFS). Hence, we have [20]:

$$h_{mm} = N_p h_m, \tag{11}$$

where h_m is the average hop distance between two adjacent MARs/OFSs. The default parameter values for the analysis are introduced in Table II in which some of them are taken from [19], [20].

TABLE II: Parameters for the performance analysis.

Parameter	**Value**	**Parameter**	**Value**
h_{mo}	1 hop	h_{mc}	8 hops
h_m	1 hop	N_f	3
N_o	8	$1/\delta$	600s
T_{BCE}	300s		

Fig. 5 shows the signaling cost as a function of the average number of active prefixes (N_p). We can clearly see that ST-DMM outperforms the other approaches in terms of

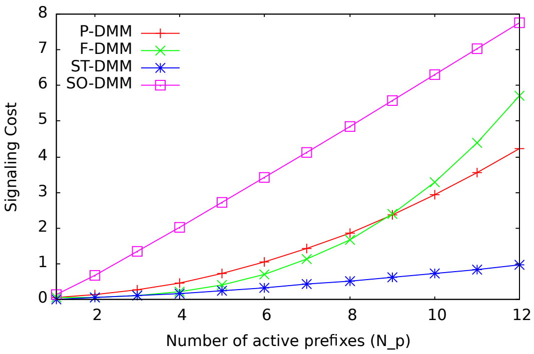

Fig. 5: Signaling cost as a function of N_p.

signaling cost. The cost is slightly increased in ST-DMM while significantly increased in the other approaches as N_p increases. In other words, when N_p is small, the difference between the signaling cost in case of ST-DMM and the others is small. When N_p increases, the difference is getting much higher. Additionally, since F-DMM leverages on an external mechanism to get the address of the previous MAR (which we do not consider in the analysis), thus may introduce a noticeable additional cost. Interestingly, when N_p is small, the cost in F-DMM is smaller than that in P-DMM. However, when N_p increases, the cost in F-DMM is getting higher than that in P-DMM. The reason is that while the distance between MAR and CMD is supposed to be fixed, the distance between the anchor and the current MAR is proportional to the value of N_p. Consequently, when the MN moves far away from its anchor, the signaling cost in F-DMM is notably increased and becomes higher than that in P-DMM. On the other hand, the cost in case of SO-DMM is supposed to be high since it is proportional to the number of switches along the path from the MN to the CN. Thus, optimal route can be achieved in SO-DMM at a cost of signaling overhead. In conclusion, the low signaling cost makes ST-DMM more scalable solution in comparison with other approaches.

C. Experimental Evaluation

The proof-of-concept was deployed based on Open vSwitch[5] and Mininet[6] with 5 OpenFlow switches, one MN, and one CN as illustrated in Fig. 6 (a). We adopted the Python based open-source Ryu[7] as a controller. The main reason why Ryu was selected among the open-source controller platforms (such as OpenDaylight, ONOS, and POX, etc. [21]) is that only Ryu fully supports IPv6. Additionally, it is very flexible and well-documented. Accordingly, S-DMM and AAA are implemented (in Python) as the applications on top of Ryu controller platform. At this stage, only ST-DMM mode is evaluated. The SO-DMM mode will be left for future work. Also, to eliminate the need for IP-in-IP tunneling between the previous and the current OFS, such a tunneling mechanism as Multiprotocol Label Switching (MPLS), and Q-in-Q Virtual Local Area Network (VLAN) will be implemented. For instance, a Q-in-Q VLAN tunneling scheme was deployed.

The experiment scenario is as follow. The MN first attaches to the OFS1. After configuring its IPv6 address, the MN starts a new communication flow with the CN (e.g., using Iperf[8]). The MN then moves to OFS2 and to OFS3, respectively. It is noted that for the improvement of the credibility, we performed the experiment a large amount of times.

1) Handover (HO) latency and Packet loss: Handover latency is defined as a period when a node fails to receive the packets for the on-going session, or cannot start a new one. It consists of both layer 2 (L2) and layer 3 (L3) handover duration: L2 latency is due to the reattachment process from the previous switch to the new one while L3 duration is caused by the IP-related procedures such as location update and address acquisition. As can be seen in Table III, the average handover latency is 76.6ms while the L3 latency is only 15.8ms. With this value, our solution can satisfy the requirement in terms of service disruption which is 300ms for a real time service, and 500ms for a normal one (according to [22]). Also, there is no packet loss in this experimentation.

2) End-to-end delay: As can be seen in Table III, after handover, the delay is slightly increased. It is obvious since the traffic is forwarded following a non-optimal route, thus, introducing a small additional delay. However, the delays (0.61ms and 0.81ms, before and after handover, respectively) are still far lower than the expected service-level delay in 5G, which is 5ms [23]. Furthermore, the delay (after handover) can be reduced by routing the packet via an optimal path between the MN and CN, however, at a cost of increased signaling and processing.

3) Comparison with a P-DMM Implementation: In our laboratory, a Linux-based P-DMM has been implemented on top of the OAI PMIP implementations[9]. To compare S-DMM with P-DMM, a similar test-bed was deployed (see Fig.6 (b)), which consists of 6 virtual machines: one CMD, 3 MARs, one MN, and one CN. From the experimentation, we obtain a similar L3 latency as in S-DMM (see Table III) while the handover latency is slightly higher than that in S-DMM.

Regarding the end-to-end delay, there is a small difference between the delay in P-DMM and S-DMM for the flow before handover. Additionally, the delay in both cases is very small. However, after handover, while the delay in S-DMM is slightly increased, that in P-DMM is significantly increased. As a result, the impact of tunneling mechanism on the end-to-end delay is obvious in P-DMM. In other words, tunneling in P-DMM introduces a significant additional delay. As a result, when the MN moves far away from its anchor MAR, the requirement in terms of end-to-end delay in 5G networks (which is 5ms [23]) may not be satisfied.

---

[5] http://openvswitch.org/
[6] http://mininet.org/
[7] http://osrg.github.io/ryu/
[8] https://iperf.fr
[9] http://www.umip.org/contrib/umip-oai-pmipv6.html

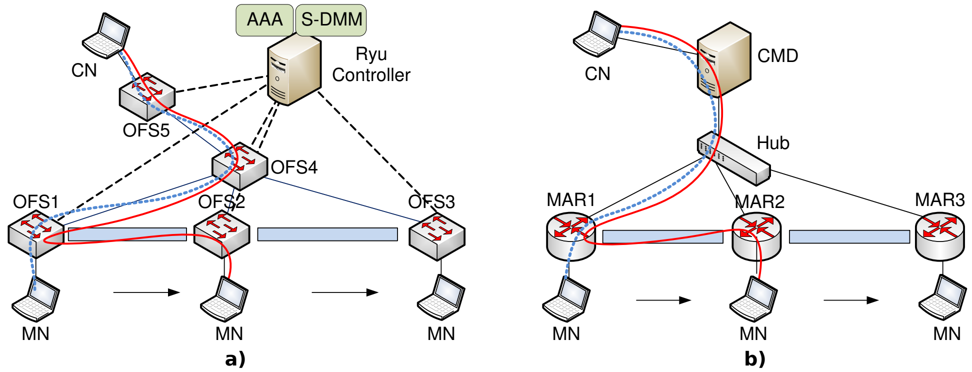

Fig. 6: Test-bed setup: a) S-DMM; b) P-DMM.

TABLE III: Experimental results: handover latency and end-to-end delay (mean and standard deviation (std)).

	L3 handover latency		**Handover latency**		**E2E delay (before HO)**		**E2E delay (after HO)**	
	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)
S-DMM	15.80	2.62	76.60	6.96	0.61	0.16	0.81	0.31
P-DMM	15.71	4.85	79.12	7.45	0.51	0.13	1.34	0.21

V. CONCLUSION

SDN is transforming the telecommunication industry and emerging with a lot of areas. To cope with the evolution of SDN into the network, this paper introduced an SDN-based DMM (S-DMM) solution which acts as an application on top of the SDN controller. S-DMM offers a lot of additional benefits brought from SDN characteristics compared to the legacy DMM approaches. Obtained results show that a similar performance as the legacy DMM in terms of handover latency and end-to-end delay can be achieved, yet at a reduced complexity of the control plan as well as a more scalable solution. All of them make S-DMM one candidate for mobility management solution in the context of 5G networks.

ACKNOWLEDGMENT

This work has been partially supported by the French project SYSTUF. EURECOM acknowledges the support of its industrial members: BMW Group Research & Technology, IABG, Monaco Telecom, Orange, SAP, SFR, ST Microelectronics, Symantec. Authors would like to thank Lucas Croixmarie and Monia Chouaibi for their help in implementing S-DMM.

REFERENCES

[1] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019”, Feb 2015.
[2] ONF Solution Brief, “OpenFlow-Enabled Mobile and Wireless Networks”, Sept 2013.
[3] C.B. Sankaran, “Data Offloading Techniques in 3GPP Rel-10 Networks: A Tutorial”, IEEE Commun. Mag., vol. 50, no. 6, pp. 46-53, Jun 2012.
[4] H.-A. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen, “Requirements for Distributed Mobility Management”, RFC 7333, Aug 2014.
[5] H.-A. Chan, H. Yokota, J. Xie, P. Seite, and D. Liu, “Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues”, Journal of Communications, vol. 6, no. 1, 2011.
[6] A. Hakiri, A. Gokhale, P. Berthou, D. Schmidt, and T. Gayraud, “Software-Defined Networking: Challenges and Research Opportunities for Future Internet”, Computer Networks, Dec 2014.
[7] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy Mobile IPv6”, RFC 5213, Aug 2008.
[8] D.-H. Shin, D. Moses, M. Venkatachalam, and S. Bagchi, “DMM for Efficient Video Delivery over All-IP Mobile Networks: Competing Approaches”, IEEE Network, vol. 27, no. 2, pp. 28-33, Mar 2013.
[9] C.J. Bernardos, A. de la Oliva, F. Giust, “ A PMIPv6-based Solution for Distributed Mobility Management”, IETF-Draft (work-in-progress), Sept 2015.
[10] MEDIEVAL project, Deliverable D4.4, “Final Operational Mobility Architecture”, Dec 2012.
[11] B.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of SDN: Past, Present, and Future of Programmable Networks”, IEEE Commun. Surveys Tut, vol. 16, no. 3, 2014.
[12] J. Costa-Requena, R. Kantola, J. Llorente, V. Ferrer, J. Manner, A.Y. Ding, L. Yanhe, and S. Tarkoma, “Software Defined 5G Mobile Backhaul”, Proc. of International Conference on 5G for Ubiquitous Connectivity, Nov 2014.
[13] G. Hampel, M. Steiner, and T. Bu, “Applying Software-Defined Networking to the Telecom Domain”, Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Apr 2013.
[14] S. Jeon, C. Guimarães, and R. L. Aguiar, “SDN-based Mobile Networking for Cellular Operators”, Proc. of the 9th ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch’ 14), 2014.
[15] F. Giust, L. Cominardi, and C.J. Bernardos, “DMM for Future 5G Networks: Overview and Analysis of Existing Approaches”, IEEE Commun. Mag., vol. 53, no. 1, pp. 142-149, Jan 2015.
[16] H. Yang, K. Sun, and Y. Kim, “Routing Optimization with SDN”, IETF-Draft (work-in-progress), Jul 2015.
[17] N. Kang, Z. Liu, J. Rexford, and D. Walker, “Optimizing the “One Big Switch” Abstraction in Software-Defined Networks”, Proc. of ACM CoNEXT, Dec 2013.
[18] S.-H. Yeganeh, A. Tootoonchian, Y. Ganjali, “On Scalability of Software-Defined Networking”, IEEE Commun. Mag., vol. 51, no. 2, pp. 136-141, Feb 2013.
[19] F. Giust, C.J. Bernardos, and A. De La Oliva, “Analytic Evaluation and Experimental Validation of a Network-based IPv6 Distributed Mobility Management Solution”, IEEE Trans. Mobi. Comp., no. 99, Feb 2014.
[20] T.-T Nguyen, C. Bonnet , “A Hybrid Centralized-Distributed Mobility Management for Supporting Highly Mobile Users”, Proc. of IEEE International Conference on Communications (ICC), Jun 2015.
[21] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, “SDN: A Comprehensive Survey”, Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan 2015.
[22] 3GPP, TR 25.913, “Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)”, Release 9, Dec 2009.
[23] NetWorld2020 Whitepaper, “5G: Challenges, Research Priorities, and Recommendations”, Aug 2014.