SDN-based DMM

Distributed Mobility Management (DMM) [1] is a new trend to overcome the limitations of the current mobility management protocols (such as Mobile IPv6 and Proxy Mobile IPv6) which are highly centralized and hierarchical. The key concept of DMM is that instead of having a centralized mobility anchor, the anchors are distributed among the network entities and placed as close as possible to the mobile node e.g., at the router edge of the access network. DMM also offers dynamic mobility features (per prefix granularity). There are two DMM schemes: the fully distributed (F-DMM) and the partially distributed (P-DMM). In F-DMM, the control and data planes are distributed while in P-DMM, the central entity still exist but for the control plan only.

Software Defined Networking (SDN) [2] is a new, 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, acting as a decision-making entity, is centralized in a software-based controller, while the network devices perform the packet forwarding function. In other words, the network devices act as a simple forwarding hardware instead of running routing protocols as in the traditional network. In SDN, the controller has a global view of the underling network infrastructure. The centralized control plane architecture and the global knowledge of the central controller make SDN a potential mechanism to work with P-DMM.

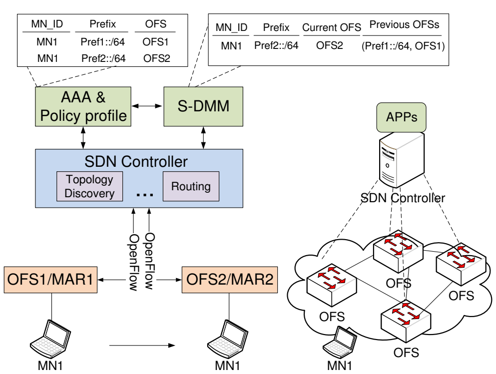

Figure 1. S-DMM architecture.

MN - Mobile Node
OFS – OpenFlow-enabled switch
AAA – Authentication, Authorization and Accounting
MAR – Mobility Access Router
CN – Corresponding Node

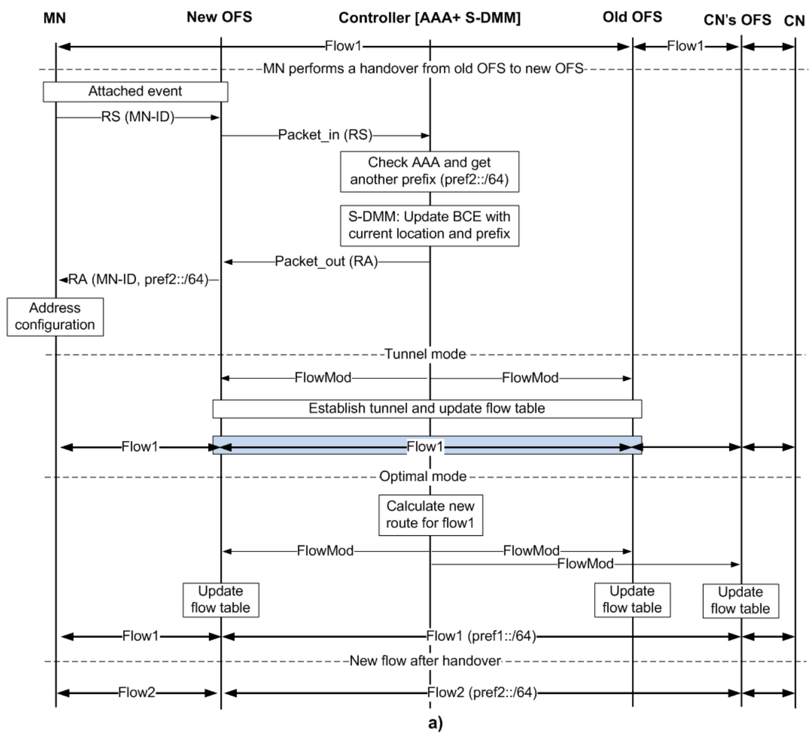

Figure 2. Signaling for handover procedure.

In our laboratory, the partially distributed scheme was implemented as an application on top of the open-source SDN controller platform RYU [3] using Python (called S-DMM [4]). The architecture and signaling for handover procedure are illustrated in Figure 1 and Figure 2, respectively. As can be seen in Figure 2, there are two modes: tunnel and optimal mode. In the former mode, the ongoing flows will be redirected between the previous and the current OFSs using tunneling mechanism, as similar to the traditional DMM approaches. In the latter one, the controller calculates the new route for the ongoing flows (based on the existing routing protocols) and populates the new forwarding rules to all intermediate OFSs along the new route between the new OFS and the CN's OFS. In the current implementation, only the tunnel mode was implemented. However, there exists some limitations:

- In terms of number of supported switches due to using Virtual Local Area Network (VLAN) as a tunneling mechanism: another tunneling mechanisms e.g., IP-in-IP encapsulation and Multiprotocol

Label Switching (MPLS) should be investigated.

- Tunnel is uni-directional: how to make the tunnel bi-directional?
- The IPv6 addressing mechanism is hardcoded: the implementation should get the IPv6 address of switches from the network (if available), from a configuration file, or by using REST API.

This project aims at examining the current SDN-based implementation to address its limitations (as well as implement the optimal mode). The outcome of this project should be a new version of the existing SDN-based implementation. Also, the experiments regarding the realistic traffic models should be conducted.

References

1. DMM Working Group, *https://datatracker.ietf.org/wg/dmm/documents/*
2. B.A.A. Nunes, M. Mendonca, X. -N. Nguyen, K. Obraczka, and T. Turletti, ``*A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks*'', IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1617-1634, 2014.
3. http://osrg.github.io/ryu/
4. T. -T Nguyen, C. Bonnet, and J. Härri, `` SDN-Based Distributed Mobility Management for 5G Networks'', submitted to WCNC 2016.