The University of Tokyo

Master Thesis

# SBMP: An SDN-based Mobility Management Protocol to Support Seamless Handover
## (SBMP: SDNによる移動端末のシームレスハンドオーバーの実現)

*Author:*
Xiaozhou Jia

*Supervisor:*
Prof. Yuji Sekiya

*A thesis submitted in fulfilment of the requirements*
*for the degree of Master of Engineering*

*in the*

Department of Electrical Engineering and Information Systems (EEIS)
Graduate School of Engineering

August 2015

# Declaration of Authorship

I, Xiaozhou Jia, declare that this thesis titled, 'SBMP: An SDN-based Mobility Management Protocol to Support Seamless Handover ' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

THE UNIVERSITY OF TOKYO

# *Abstract*

Department of Electrical Engineering and Information Systems (EEIS)
Graduate School of Engineering

Master of Engineering

**SBMP: An SDN-based Mobility Management Protocol to Support Seamless Handover**

by Xiaozhou Jia

The proliferation of mobile devices has created an extraordinary mobile data growth, which stimulates interest in delivering better user experience. Among all possible improvements, maintaining connectivity while moving around is very attractive to end users. Recently many research efforts have been made on applying SDN in the field of wireless or mobile networking. SDN enables a centralized control over computer network by manipulating programmable devices. This thesis focuses on applying the SDN paradigm to mobility management, which has been little studied until now. A testbed on Mobile IPv6 in laboratory environment was fist studied and results show shortcomings of this traditional mobility management protocol. Then, the protocol design, implementation, evaluation of SBMP are presented in this thesis. Through evaluations from both quantitative and qualitative perspective, results show that SBMP is able to provide a high performance handover in term of latency.

# *Acknowledgements*

I would like to express my sincere gratitude to Prof. Yuji Sekiya, for the continuous support of my master study and research, for his instructions and guidance on how to conduct meaningful research, for his expertise, patience, kindness, enthusiasm, and immense knowledge. Second, I would like to express my heartfelt gratitude to Dr. Hajime Tazaki for sparing his time to join the weekly discussion on my research progress and providing his valuable comments.

I'm also greatly indebted to Dr. Panagiotis Georgopoulos for giving me the opportunity for a five-month academic visit at Computer Engineering and Networks Laboratory of ETH Zurich. And my thanks go to Prof. Laurent Vanbever and Dr. Karin Anna Hummel in the same group as well. This visit truly helped me getting the first hand experience on a Mobile IPv6 testbed and the discussion there made a lot of meaningful inputs into this thesis.

My sincere thanks also go to Prof. Nakayama, Prof. Ogawa and Dr. Miyamoto, for their encouragement, insightful comments and thought-provoking questions in the lab seminars. I thank my fellow labmates for their kindness and studying with them was really an unforgettable experience.

Last but not the least, I would like to thank my family: my parents and my sister for supporting me spiritually throughout writing this thesis and my my life in general. I would also like to thank my fiancée Ms Shasha Yao, for all her love and support.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **SBMP** | **S**DN **B**ased **M**obility Management **P**rotocol |
| **BA** | **B**inding **A**cknowledgment |
| **BU** | **B**inding **U**pdate |
| **CN** | **C**orresponding **N**ode |
| **CoA** | **C**are-of **A**ddress |
| **CoT** | **C**are-of **T**est |
| **CoTI** | **C**are-of **T**est **I**nit |
| **DAD** | **D**uplicate **A**ddress **D**etection |
| **HA** | **H**ome **A**gent |
| **HoT** | **H**ome **T**est |
| **HoTI** | **H**ome **T**est **I**nit |
| **IETF** | **I**nternet **E**ngineering **T**ask **F**orce |
| **LMA** | **L**ocal **M**obility **A**nchor |
| **MAG** | **M**obile **A**ccess **G**ateway |
| **MIPv6** | **M**obile **IPv6** |
| **MN** | **M**obile **N**ode |
| **PBA** | **P**roxy **B**inding **A**cknowledgment |
| **PBU** | **P**roxy **B**inding **U**pdate |
| **PMIPv6** | **P**roxy **M**obile **IPv6** |
| **RA** | **R**oute **A**dvertisement |
| **RO** | **R**oute **O**ptimization |
| **RS** | **R**oute **S**olicitation |
| **SDN** | **S**oftware **D**efined **N**etworking |

*To my family*

# Chapter 1

# Introduction

## 1.1 Background

The past few years witnessed an increasing mobile data traffic, thanks to the proliferation of mobile terminals - such as smart phones, tablets and so on. And this trend is expected to continue in the future. According to Ericsson's most recent mobility report [1], world mobile subscriptions are expected to reach 9,500 million by 2020 while total mobile data traffic are expected to reach 25EB per month in 2020 (Figure 1.1). Such pace of increasing impose significant challenges for today's mobile network operators.

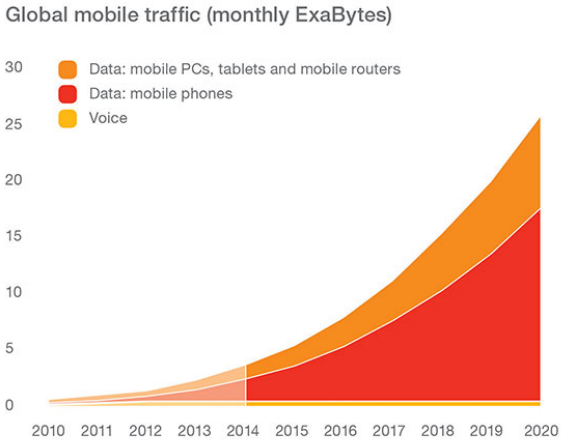



FIGURE 1.1: Global mobile traffic trend (monthly ExaBytes) (Figure is taken from [1])

One of the most important responsibilities of mobile network operators is mobility management, which refers to a set of mechanisms to keep ongoing-sessions continuity while a mobile user changes his/her physical channel, access network or communication protocol. Mobility management could benefit both end clients and the network operators. On one hand, end users could leverage the benefit of heterogeneous network, getting access of broader network resources while enjoying seamless roaming. On the other hand, network operators could benefit from this via traffic offload [4].

## 1.2 Problem statement

Imagine one is talking to his friend via Skype, which is a telecommunications software based on the Internet [5]. During the talk he has to move to another room, which has different access network (probably different Wifi SSID). During his movement, chances are very high that the talking session would be broken or at least be interrupted. To maintain an ongoing session alive during movement, we need some mechanisms to provide mobility support.

In order for users to benefit from mobility support, it is important to keep the upper layer sessions uninterrupted by the mobility. If the duration of the sessions are short (e.g., web browsing), the probability is high that the sessions will finish before the handover happens; even if those sessions are interrupted by the handover, the cost is usually low (e.g., refresh the web page). However, if the TCP sessions are typically long (e.g., video conferencing, Voice over IP (VoIP), Game net, download/upload of large size files), the interruptions during the handover would become unacceptable.

While there are many existing solutions for this problem, most of them are not deployed in commercial network - the complexity of those protocols sometimes make deployment impossible. Usually, mobility support is traded with performance degradation and extra overhead (heavy modification on protocol stack, complicated signaling process and so on) in these solutions (details in Chapter 2). We argue that Software Defined Networking (SDN) could be another promising candidate to provide mobility management, since its flexibility enables the possibilities to introduce new functions in an efficient way. However, how to utilize the SDN paradigm is still open to the research community and has been little studied.

## 1.3 Contributions

The main contribution of this work is the design and development of SBMP: a system that provides mobility management using the concept of SDN. And to evaluate the performance of this proposal, we chose to mainly focus on the handover performance. Other key contributions of this thesis are as follows:

- Investigate how mobile IP performs on a testbed in lab environment
- Figure out what SDN could bring in, design and develop an SDN-based mobility management protocol
- Compare and contrast the above mentioned protocols
  - Focus on handover - bottleneck of mobility management
    - In what way handover is carried out
    - Handover times performance
    - How handover affects the connection

## 1.4 Thesis organization

The rest of this thesis is organized as follows:

Chapter 2 presents related work in the area of mobility management and some relevant publications on applying SDN in mobile or wireless network.

Mobile IPv6, one traditional solution to the problem and a testbed on it is presented in Chapter 3.

In Chapter 4 the system design and development of SBMP is presented.

And in Chapter 5 the evaluation is done in both quantitative and qualitative perspective. The methods of evaluating the system and experiments performed are discussed, along with the corresponding results.

Finally, Chapter 6 contains concluding statements and points out some possible future directions continuing this thesis.

# Chapter 2

# Related Work

Various mobility management protocols for enabling end user mobility have been proposed. In particular, mobility management implemented at the network layer has being developed by the Internet Engineering Task Force (IETF). This chapter gives an overview of two main groups of approaches to support mobility management, host-based and network-based. The former group is represented by Mobility IP [6][7], which expects end host's participation in the signaling process. While in the network-based ones, the total process is transparent to the end clients.

One should notice that there are also other ways for grouping research on mobility management. For example, mobility management could also be broadly divided into routing-based and mapping-based approach [8]. In routing-based approach, a mobile keeps its IP address no matter where it goes. As a result, the routing system must continuously keep track of a mobile's movements and reflect its current position on the routing infrastructure. A famous example of routing-based protocol called Connexion was designed by Boeing. The mapping-based approach is to provide mobility support by a mapping between a mobile node's stable identifier and its dynamically changing locator.

For two reasons we didn't apply this taxonomy. Firstly, as discussed in [9], routing-based proposals would not be suitable for large network since the whole network must be informed of every movement by every mobile nodes and it will be feasible only in small networks with a small number of mobile nodes. Secondly, our proposal (will be shown in Chapter 4) is a hybrid of routing-based and mapping-based mobility management.

At the end of this chapter, an overview of SDN and some existing work using SDN into mobile and wireless network will also be discussed.

## 2.1 Host-based mobility

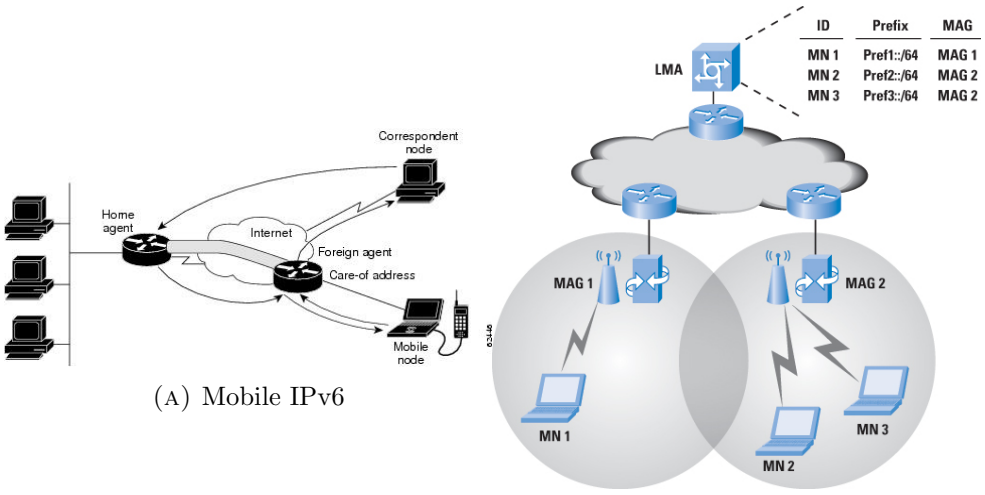
Equipped with a specialized protocol stack, end clients are required to be involved in the management of the mobility. Mobility IP [6, 7] enables global reachability and session continuity by introducing the home agent (HA), an entity located at the home network which anchors the permanent IP address used by the MN, called home address (HoA). The HA is responsible for redirecting received traffic to the MN's current location. When away from its home network, MN acquires a temporal IP address from the visited network - called care-of address (CoA) - and informs the HA about its current location via Binding Update (BU). An IP bidirectional tunnel between the MN and the HA is then used to redirect traffic from and to the MN [10].

The employment of a fixed HA brings the triangular routing problem when the MN is away from HA. All the packets from CN to MN will have to take a detour to pass the HA, which in most cases is no longer on the shortest path between MN and CN. And triangle routing also leads to large handover signaling cost, as well as heavy load on HA due to the same reason. Though some modifications on MIPv6 have been proposed to mitigate those problems, for example using multiple HAs to support one MN [11]. These extensions make the signaling burden even heavier and can hardly been deployed in a scalable way. And usually different levels of modifications of operating system will be required on the end users' side. We will cover more details of Mobile IPv6 in Chapter 3.

One apparent trade off of host-based mobility is that a mobile node (MN) is required to have a specialized protocol stack to support mobility. Thus, modifications or upgrades on MNs become a must. This obviously increases the operation expense and complexity for the MNs. Accordingly, network-based mobility protocols are being developed by the research community.

## 2.2 Network-based mobility

In a network-based mobility support scheme, end host do not need to engage in exchange of any mobility-related signaling with other entities. The Proxy




(A) Mobile IPv6

(B) Proxy Mobile IPv6

FIGURE 2.1: MIPv6 and PMIPv6
(Figures are taken from [2, 3]

Mobile IPv6, standardized in RFC 5213 [12], is one of the most widely known approach. The PMIPv6 proposed special mobility agents in networks that include an local mobility anchor (LMA) and a mobility access gateway (MAG), for performing mobility-related signaling and handling mobility management on behalf of each MN. An MAG is responsible for detecting and registering the movement of the MN in its access network. When an MAG detects the MN's movement, it sends a proxy binding update (PBU) to the LMA. Upon receiving the PBU, a LMA keeps a record of the MN's attachment point and create/update the binding cache. The LMA also sends out a proxy binding acknowledgement (PBA) as a response to PBU to MAG. After MAG receives the PBA, it sends the router advertisement (RA) message to MN, which will configure its address.

Note that for both Mobile IPv6 and Proxy Mobile IPv6, there are many extensions grown from the original version. To name a few standardized by IETF, Hierarchical Mobile IPv6 (HMIPv6) [13] introduces extensions to Mobile IPv6 and IPv6 Neighbour Discovery to allow for local mobility handling. Fast Mobile IPv6 (FMIPv6) [14] updates the packet formats for the Handover Initiate (HI) and Handover Acknowledge (HAck) messages to provide a shorter handover. Fast Proxy Mobile IPv6 (FPMIPv6) [15] introduces several modifications to FMIPv6 to adapt to the network-based mobility management.

The benefit of this group of extensions is that they aim at either transparency to end users or they aims at providing a fast handover. And there are many

comparison work done in the literature [16–19]. Generally, a shorter handover delay could be expected in PMIPv6-like protocol as the LMA is a local network entity and sending signaling to it produces less delay than sending signal to a remote HA. Accordingly, the logic now sits in the network, which increases the burden and complexities.

## 2.3 Software Defined Networking

SDN is one of the key outcomes of extensive research efforts over the last decade towards to a more open, programmable, reliable, secure and manageable network infrastructure. By separating the control logic from the underlying infrastructure, it brings new tool set into network research community, enabling researchers to implement new ideas by simply writing software. One of the representative implementation of this concept is OpenFlow [20] , which is discussed extensively in research community and sometimes are used to refer SDN itself.

Recently, using SDN to handle mobile and wireless network becomes of interest. A number of surveys has been published. For instance, [21] recognizes challenges for future mobile and wireless network and points out SDN could efficiently address those challenges significantly and benefit the future mobile network.

Applying SDN in mobile context is still in its infancy. And there are already a lot of work focusing on this [22–25]. Basically, there are two groups of work trying to apply SDN. One is on cellular networks, the other is on WLAN.

With cellular networks, SoftRAN [26] is a software defined centralized control plane of the radio access network. It regards base stations in a geographical area as a virtual big-base station. One point worth mention here is that the authors also implement some control plane functionalities down at the radio elements. Specifically, the controller in the big-base station handles the cross radio elements decision, and underlying radio elements will need to deal with the frequently parameters. OpenRAN [27] is another software defined RAN architecture from another perspective by introducing cloud computing. It consists of wireless spectrum resource pool, cloud computing resource pool and an SDN controller.

CellSDN [28] focuses on the software defined core part in the LTE network. It covers both access and core network, and aims at implement a network

operating system on LTE networks. SoftCell [29] is the successive research of CellSDN. SoftCell introduces the software defined access switches and they are used to implement the packet classification.

When it comes to applying SDN to WLAN, OpenRoad [30] proposes a networking environment that users could move freely between any wireless infrastructure. The proposal was based on OpenFlow and they deployed it on Stanford campus [31]. Odin [32] is a prototype software defined wireless network framework for enterprise WLANs. Odin builds on a light virtual access point (AP) abstraction that has stable associations with users. The decision module in Odin is an application on top of the OpenFlow controller. In this way, Odin benefits several applications such as seamless mobility, load balancing. OpenRadio [33] is another work trying to apply SDN on WLAN. OpenRadio proposes a programmable wireless data plane by enabling programmability of the Physical and MAC layer. It decomposes wireless protocols into processing plane and decision plane with a modular and declarative APIs.

The above mentioned two groups of research are highly related to each other, especially under the assumption that future cellular network are expected to have an IP core. This means that all traffic leaving access networks will become IP-based. As a result, IP mobility management becomes a key role to support future wireless systems. Current and future researches on Internet mobility management will continue to contribute to cellular networking as well.

# Chapter 3

# Mobile IPv6 Testbed

## 3.1 Mobile IPv6

As mentioned briefly in Chapter 2, Mobile IPv6 [7] is derived from Mobile IPv4 [6], which was designed to provide mobility to nodes with IPv4. Generally, when the mobile node (MN) changes its location from one access network to another, the IP address corresponds to the MN's location will change. In Mobile IPv6, the home agent (HA) is used to keep a binding between MN's identifier (Home IP address) and its location (Care of IP address).

Since the Mobile IPv6 specification was published [7], many extensions have been developed. For example, Fast Mobile IPv6 [34] aimed at reducing the handover latency. Proxy Mobile IPv6 [12] released the requirement that end clients have to be modified to support the protocol. In this thesis, we chose to investigate the standard version of Mobile IPv6. To measure the performance, we used an linux-based open source Mobile IPv6 implementation called Universal Mobile IP (UMIP) [35].

## 3.2 Mobile IPv6 Handover Process

A complete Mobile IPv6 handover process will have both low layer latency and network layer latency, as shown in Figure 3.1.

The low layer latency includes scanning, authentication and association to the new access point. After the low layer handover finishes, the MN performs the movement detection process. The MN will detect whether it is receiving

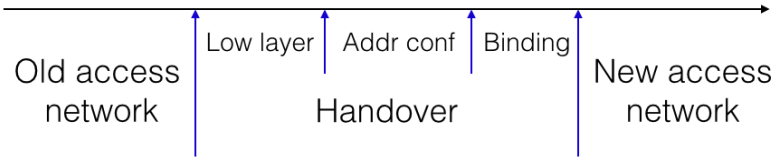



FIGURE 3.1: Handover latency time line for Mobile IPv6

TABLE 3.1: Equipment details

| Role | Hardware | Operating System | CPU |
|---|---|---|---|
| CN | Macbook Air | OS Yosemite | 1.3 GHz |
| HA, AP1, AP2, MN | Thinkpad T410 | Ubuntu 14.04 | 2.4 GHz |
| Switch | PLANET GSD-805 | Unknown | Unknown |

router advertisements and neighbor discovery messages from a new access point. Then, the MN will have to go through the Duplicate Address Detection (DAD) to confirm the uniqueness of the IPv6 address, after which the MN will form a new Care of Address (CoA). Note that the address configuration is based on the prefix information included in the RA messages. The last step is to notify the home agetn (HA) the MN's CoA via Binding Update (BU) and Binding Acknowledgement (BA). If the routing optimization is enabled, the MN will also need to notify corresponding node (CN).

## 3.3 Handover latency on Mobile IPv6

### 3.3.1 Testbed setup

This section explains our testbed setup and relevant hardware in a lab environment. By the time we made the testbed, we didn't have access to existing routers or equipment have Mobile IPv6 enabled by default. After a careful study, we chose to have multiple Linux boxes functioning as accessing points and home agent. That also goes along with our decision to use UMIP. The testbed is shown in Figure 3.2 and the equipment details is shown in Table 3.1.

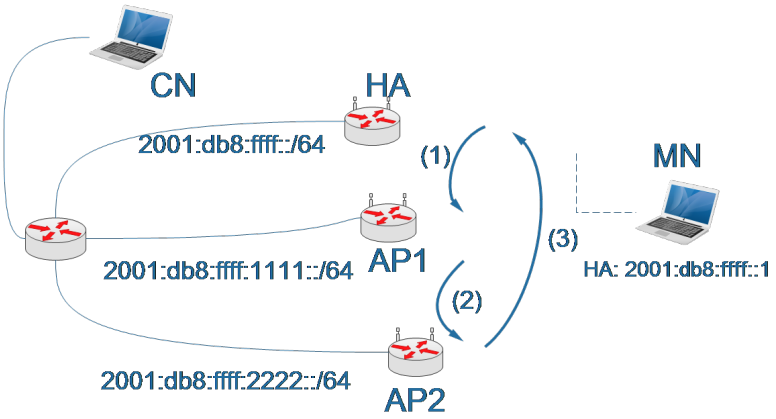



FIGURE 3.2: Mobile IPv6 testbed

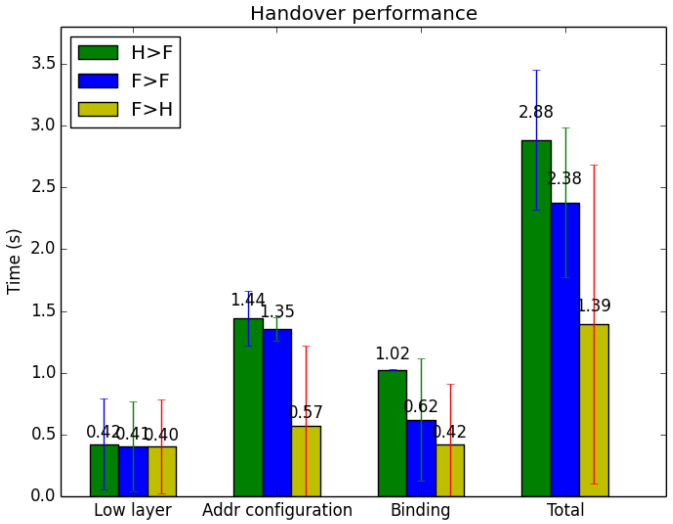



FIGURE 3.3: Handover latency

### 3.3.2 Handover latency results

As shown in Figure 3.3, We collected handover latency under three types of handover scenarios: from home network to foreign one, movement among foreign networks, and from foreign network back to home.

One clear conclusion we could reach is that three types of handover have almost the same low layer latency. And it has been observed that DAD is causing the most latency during a handover. And moving from foreign network to home network comparatively cost less time for IPv6 address configuration. That is because when the MN moves back to home network, it will not be

TABLE 3.2: Handover success rate for a fast mover

| Movement pattern | Success rates |
|---|---|
| Linear | 26.7% (4/15) |
| Back and forth (home-foreign) | 46.7% (7/15) |
| Back and forth (foreign-foreign) | 73.3% (11/15) |

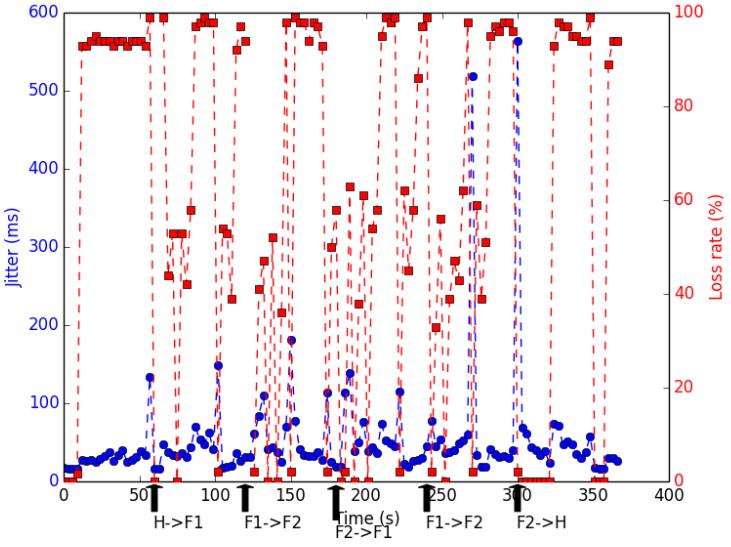



FIGURE 3.4: How handover affects the connection

necessary for the MN to acquire a new CoA as the home address could be used to reach the MN.

Another experiment we tried is that we made a "faster mover" scenario. In this case, the MN will only stay at one station 2 seconds and then move away. (We let the MN stay at each site for 10 seconds in the former experiment). As we have shown that a handover will cost more than 2 seconds on most cases, we could expect that some of handovers will not finish before the MN moves away. And the results are shown in Table 3.2.

We also collected the Jitter and loss rate using iperf on the testbed. The results are shown in Figure 3.4. We could see clearly each handover will cause jitter increase nearly 200%. Meanwhile, we could see that loss rate is on a relatively high level along the experiment, we couldn't conclude that is caused by MN's movement, as there are wireless interference we couldn't reduce on the testbed. But a pattern do exist that loss rate will increase in respond to each handover.

## 3.4 Summary and motivations

Based on our experiment carried out on the testbed, we showed that Mobile IPv6 protocol experiences averagely 2 seconds of handover latency. To our life experience, a 2 seconds break is long enough to interrupt our chatting session on Skype. And a MN could be totally unreachable if he/she moves faster enough to jump away before the handover could ever finish. Also, the loss rate and jitter increases along with each movement. Those matrics are critical in a video/audio streaming session.

That gives much space to do optimization on handover latency. And in this work we chose to use SDN, an innovative network paradigm which provides flexibility.

# Chapter 4

# SBMP: SDN-based Mobility Management Protocol

In this chapter we present our proposal of using SDN to support seamless roaming. We will first give an overview of the proposal and then go to the protocol details. At last, we will also present some implementation details using Pyretic [36].

## 4.1 Overview of protocol

As mentioned in Chapter 2, SDN decouples the control logic from the data plane. And it enables researchers to evolve new protocols easily by programming. In our proposal, the concept of an "Anchoring point" is lifted up to the SDN controller(s). Each time the MN changes its location, an update is necessary in the binding information database. Based on MN's new location reported by the new attachment accessing switch, the controller will install new rules to the switches effected in this movement.

To reduce the necessary traffic between two layers, we decided to install provisional rules in advance to the switches. By using location tags to identify a location, the routing infrastructure is running based on those location tags. To ensure packets don't get lost during one's movement, we are putting flooding logic onto the edge switches as the MN moves. Local agents reside on each edge switch to detect attachment/detachment event on that switch and trigger corresponding rules accordingly.

After an MN attaches to a new access switch, switch there will report this event to the controller. After the controller sees this, it will update the binding information database, which were consulted by all edge switches who want to locate a mobile node.

We are setting the case when a network operator have control over its own domain, consisted both wireless and mobile network. The corresponding node (CN) is transparent from the movement of MN.

Generally, we will have three different roles in our protocol.

- Core switches
  - Provisional flow rules based on tag
  - Depend on configuration, will have different behavior upon seeing a *up-k* tag
    - Flood to both upstream and downstream switches (modify tag to *down* before flooding down)
    - Modify tag to *down*, only flood to downstream switches
- Edge switches
  - Incoming packets
    - Consult the controller, which will track this switch as well
    - Put on a tag (keep the original destination)
  - Outgoing packets
    - Normal traffic: Strip off the tag and forward to the MN
    - Flood traffic (with a flood tag *down*): Strip off the tag and check destination, forward to MN if matches, drop otherwise
- Controller
  - Updates the location database upon receiving a port up
  - Remembers switches ever hit a specific MN, update those switches if the MN moves

## 4.2 Protocol details

### 4.2.1 Tagging

In order to reduce the amount of traffic between SDN controller and the switches, we decided to rely the entire routing infrastructure on a tagging system. This could also relax the limitation on controller that it needs to support a large number of switches. Generally, we will have two types of tagged packets in the network.

One type would be the normal packets, which will have a destination tag on them. A destination tag will correspond to an edge switch if we think about the network topology in a spanning tree way. Another type of packets would be those corresponds to movement and flooding. The flooding mechanism is introduced in Section 4.2.3. And here we want to bring this up because of these tags should comply to the high level decision - routing should based on tags.

The tagging system brings us at least two benefits. Firstly, it would be no longer necessary to consult the SDN controller each time a switch sees a new header. Secondly, the tagging system in fact separated an MN's location from its identifier.

### 4.2.2 Controller's role

In our proposal, SDN controller keeps a binding information database which binds each nodes' identifier (IP address) and locator (location tag). The controller will have the following responsibilities.

- Install forwarding rules to all the core switches based on location tag, thus the entire routing infrastructure is based on tags
- Answer requests from edge switches asking which tag should be put, given a destination's identifier (IP address)
- Update edge switches with most recent binding information database when notified MN's movement by an edge switch

### 4.2.3 Flooding mechanism

In our proposal, we carefully designed a flooding mechanism to reduces the number of lost packets during MN's movement. Apparently, we made an assumption that end users usually move within a relatively small region. That's the logic behind the idea of using a spanning tree to do flooding. However, how many levels would be necessary to cover an average user's movement, is still an open question. As we will cover in Chapter 5, we will have two types of movement and they have different implications on the movement pattern.

**Priority with flooding rules** Flooding brings not only benefit: we are replicating packets and that will soon become a serious burden to the network. To address this, we designed a reducing priority mechanism: we will take half of the rules' priority each time it got up one level on the spanning tree. This will make sure flooded packets have lower priority in the core network compared to the edge part. As it would be much more busy, in terms of more traffic, near the core network and those flooded packets will be dropped since their low priority. Again, how far will those flooded packets go will depend on the configuration of flood tag value $k$. Details in Section 4.3.2.

### 4.2.4 An example

We will walk through an example in this section and try to cover the points mentioned in 4.1. As shown in Figure 5.7, here we have ten switches, six of which are located at the edge of the network (`s5, s6, s7, s8, s9, s10`). And the other four are referred as the "core switches". A corresponding node (CN) sits behind `s6` and we assume it will be stationary during the whole process. A mobile node (MN) starts its journey from `s7`, first move to `s8`, and then continues to move further, end up in `s9`.

Let's say now CN wants to talk to MN, it will send out a packet which will reach `s6`. First time seeing this, the edge switch `s6` will consult the controller which tag should be put on this packet, providing MN's IP address. The controller will check the binding information database and returns `s7`. And then all packets comes from CN destined to MN will have a tag `s7` on them. And at each switch along the path `s6 - s2 - s1 - s3 - s7` there would be rules pre-installed for how to handle a tag s7 - usually would be forwarding on the shortest distance path. When the packets actually arrives at `s7`, the

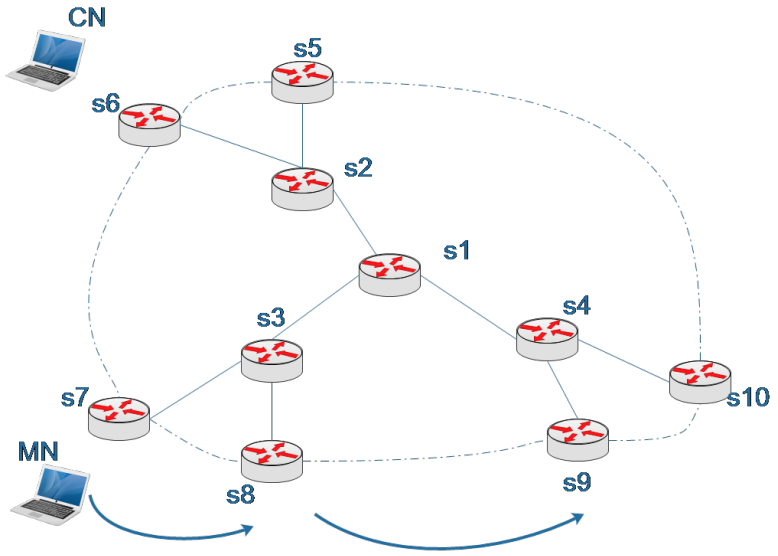



FIGURE 4.1: Walk through an example

edge switch at this side will strip off the tag and forwards those packets to MN.

The interesting part starts from this moment: when the MN starts to move to `s8`, the local agent locates at `s7` will respond quickly by triggering a pre-defined set of flood rules. These flood rules say that all the packets with a tag `s7` should now be flooded along the spanning tree at a pre-defined level $k$. In this case, `s7` will modify the tag `s7` to a special flood tag and send the packets to an upper stream switch in the spanning tree, in this case `s3`. Upon receiving a packet with a flood tag, it will behave differently based on configuration of $k$ value. If $k$ is configured to be one level (in this case would be enough), `s3` will stop sending packets to upper stream switches, modify the flood tag and flood the packets to all down stream switches except where it comes from, in this case `s8`.

Again, `s8` will see a packet with some tagged packets, it will strip off those tags as usual. And then it will check the destination field of the packet to see whether this is a packet looking for itself. In one case, the packet's destination node do sits behind this edge switch, it will strip off whatever the tag on it and forward the packet to the end node as expected. However, the packets will simply be dropped otherwise. It's a natural decision to make as we don't want everyone else got notified by one's movement. In this case, `s8` will strip off the flood tag and then forward the packets to MN.

When MN moves further, to `s9` in this case, the local agent at `s8` will again trigger flood rules. And `s8` will bouncing packets addressed to MN back into the network after put on a flood tag. So the packets will reach `s3`. To this moment, it's not hard to tell that flooding one level on the spanning tree will not be enough. So we will configure the $k$ value to be (at least) two. Thus, `s3` will not only floods the packet to downstream switches on the spanning tree, it will also send the packets to its upper stream switches (in this case `s1`). When `s1` receives packets with a flood tag, it will simply flood the packets to all its down stream switches except where it comes from (in this case, `s1` will forward the packets to `s2` and `s4`, which will in turn flood the packets again to all edge switches). And this will finally make those packets reach their intended target, the MN.

Along all of the flooding process, the edge switches who see a new end clients coming should also reports this to the controller. Thus, the controller could update the binding information database and install new set of rules to other edge switches. The reaction of this event is quite important since it affects the performance of the network and more details could be found in 4.3.

## 4.3 Discussion

### 4.3.1 Communication between mobile nodes

In our setup, the CN is stationary during the whole process, which is not necessary. In fact, communication between mobile nodes is expected to be quite frequent since the number of mobile nodes is expected to be growing. In our proposal, the handover is transparent to the end clients and the binding information is maintained in a centralized controller. This makes our proposal also applicable for communication between two mobile nodes.

### 4.3.2 Selection of k value

The selection of k value matters when we consider the MN's movement range. As we are flooding the packets along the spanning tree, starting from the very edge, if k is too small, flooding might not be able to cover all movement. For example, let's consider second movement in the example mentioned in 4.2 from s8 to s9, the packets will never reach MN if k is configured to one. On the other hand, if k value is configured too large, we will have duplicate traffic

in many places. And too much duplicate traffic will increase the burden of the network. So how to collect a proper value worth more consideration.

As for one of the future work, we will learn from users' movement pattern and try to figure out a better way to apply the results to this part of logic.

## 4.4 Development

We implemented our proposal based on Mininet 2.2.0 [37]. In Mininet, movement is simulated by detaching a host from one switch and then attaching it to another. The local agent at each edge switch are simulated by a precalculated flooding rules based on MN's detachment point and attachment point. When an edge switch detects the detachment, it will trigger the installation of the flooding rules with priorities calculated by a specific set of rules.

The forwarding tags are implemented with the field `vlan_id` and `vlan_pcp`. The controller will calculate forwarding rules based on shortest path routing algorithm and translate those rules to OpenFlow rules and assign them different `vlan_id`.

We chose to use Pyretic [36] as our SDN controller. Pyretic enables network programmers and operators to write succinct modular network applications by providing powerful abstractions. We defined flooding policies and schedule a list of actions the controller should make in react to each handover.

# Chapter 5

# Evaluation

## 5.1 Qualitative comparison with Mobile IP

Mobile IP is the most well known existing solution to mobility management. As the preliminary evaluation, we here give a theoretical comparison between SBMP and Mobile IP.

The first metric we considered is network type, our SDN based solution is advantageous both from end users' perspective and network operators'. An end user would be happy to have the feature of changing from Wifi to LTE seamlessly. And the network operators would benefit in diverting traffic from busy ones to others which would have higher performance.

End users don't really care neither where the functionality is implemented nor where the anchor is. But they would prefer a shorter delay and higher bandwidth. Also, the triangular routing would degrade the performance.

## 5.2 Analytical performance evaluation

In this section, the performance of SDN-based mobility protocol (SBMP) is evaluated and compared with Mobile IPv6 and Proxy Mobile IPv6 on an analytical basis.

TABLE 5.1: Theoretical comparison with Mobile IP

| | Mobile IP | SDN-based solution |
|---|---|---|
| Function position | Access network & core network | Core network |
| Anchor position | Home agents | SDN controllers |
| Triangular routing | Can't been avoided | Could be quickly removed by controllers |
| Communications between MNs | Detour to two HAs | Same as with stationary nodes |
| Deployment | Modification in protocol stack & end clients | Initial SDN infrastructure & Software |

TABLE 5.2: Role assignment in the network model

| | Mobile IP | Proxy Mobile IPv6 | SBMP |
|---|---|---|---|
| Gateway switch | Gateway switch | LMA | Root of spanning tree |
| Edge switch | Edge switch | MAG | Edge switch |
| Protocol-dependent entity | Home Agent (HA) | None | SDN controller |

### 5.2.1 Network model

As shown in Figure 5.1, we will carry out our analytical performance evaluation based on this model. The network is composed of several access networks. Each access network is assumed to be circular in shape, and all access networks are assumed to be identical for simplicity [19]. The nodes are connected through a number of hops. The average distance (hop counts) between different nodes are also depicted in Figure 5.1. In each access network, there will be at least one gateway (GW) and several edge switches. For a fair comparison, we assigned different roles on the identical entity in Figure 5.1, shown in Table 5.2. The CN (shown in orange) is expected to be stationary in the Internet and the MN (shown in red) will go through one intra-mobility and one inter-mobility.

It's also necessary to take one-way packet transmission delay into consideration [18]. We will have both wired link and wireless link in out network model. The MN is connected to edge switches via wireless links and the other parts in the network are connected with wired links. For transmission over a wired

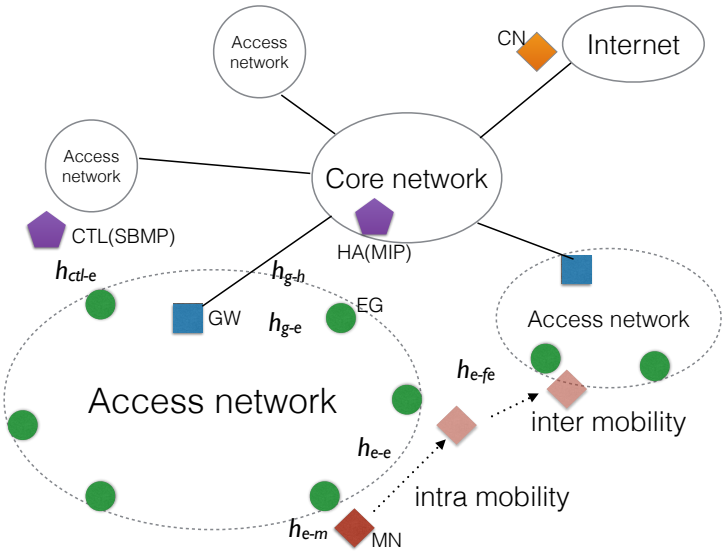



FIGURE 5.1: Access network configuration

link, we assume that packets will not be lost and will reach the destination without failure. The one-way packet transmission delay $d_{wd}(L_p, h)$ can be expressed as

$$d_{wd}(L_p, h) = \frac{L_p \times h}{BW_{wired}} + D_{wired} \tag{5.1}$$

where $L_p$ is the packet size and $h$ is the average number of hops from source node to the destination node. $BW_{wired}$ and $D_{wired}$ are the bandwidth and the delay of wired link, respectively.

But wireless links are unreliable compared to wired links. Also, one-way transmission delay over a wireless link is expected to be longer than the one over a wired link. Results in [38] and [18] are listed here. Suppose that $\tau$ and $\rho_f$ is the interframe time and the frame error rate over the wireless link, respectively. Let $p_{i,j}$ be the probability that the first frame sent from the MN arrived at the edge switch successfully, being the $i$th retransmitted frame at the $j$th retransmission trial. Then, the one-way frame transmission delay $d_{frame}$ between the MN and edge switch via the wireless link is expressed as [18]

$$d_{frame} = D_{wl}(1 - \rho_f) + \sum_{i=1}^{n} \sum_{j=1}^{i} p_{i,j}(2i \times D_{wl} + 2(j-1)\tau) \tag{5.2}$$

where $D_{wl}$ is the wireless link delay mainly depending on which layer 2 technology is being used. Also, $i<n$, $j<i$. $p_{i,j}$ could be expressed as follows [18]:

$$p_{i,j} = \rho_f(1-\rho_f)^2(\rho_f(2-\rho_f))^{(i^2-i)/2+j-1} \tag{5.3}$$

Suppose $k$ denotes the necessary numbers of frames per packet over the wireless link. Then we have

$$k = \lceil \frac{L_p}{L_f} \rceil \tag{5.4}$$

where $L_p$ and $L_f$ are the packet size and frame size, respectively. Thus, by combing 5.2 and 5.4, the one-way packet transportation delay over the wireless link $d_{wl}(L_p)$ is shown as [38]

$$d_{wl}(L_p) = d_{frame} + (k-1)\tau \tag{5.5}$$

### 5.2.2 Movement model

We will consider two types of movement in our evaluation: intra-mobility and inter-mobility. Intra-mobility refers to the cases in which a MN moves within a domain while inter-mobility refers to ones in which a MN moves to another domain. We adopted an existing model from [19] as a starting point. The model [19] made the following assumptions

- The residence time of an MN staying in a particular edge switch or in a domain is an exponentially distributed random variable
- The session arrival process to an MN follows a Poisson distribution, and $\lambda_S$ is the rate of exponentially distribution which inter-arrival time complies.
- The MN moves at an average speed $\nu$ and in a direction uniformly distributed over the range $[0, 2\pi]$

We omit the derivation of the model and list the key variables here. The average number of movements can be expressed as

$$E(N_l) = \frac{\mu_C}{\lambda_S}(1 - \frac{1}{\sqrt{N}}) \quad \text{and} \quad E(N_d) = \frac{\mu_d}{\lambda_S} \tag{5.6}$$

where $E(N_l)$ and $E(N_d)$ are numbers of intra-moblity and inter-mobility respectively, $N$ is the number of nodes in a single domain. And $\mu_C$ and $\mu_d$ are the edge switch crossing rate and access network crossing rate, which could be written as

$$\mu_C = \frac{2\upsilon}{\pi R} \quad \text{and} \quad \mu_d = \frac{2\upsilon}{\pi R\sqrt{N}} \tag{5.7}$$

### 5.2.3 Handover delay

We will derive formulas of handover delay for each protocol in this section. As layer 2 delay $D_{L2}$ will depend on L2 technology and manufacture choice and all protocols will have layer 2 delay, we ignored detailed discussion of this portion and focused on the different protocol implications.

#### 5.2.3.1 SBMP handover delay

As shown in Figure 5.2, the handover time line of SBMP could be separate into two parts: actions involve the controller and those without controller.

The green items in the graph refers to the actions without involvement of the controller. The local agents resides at edge switches will trigger flooding rules on that switch and packets reach that point later will be referred as "flooded packets". This indicates that our protocol reduces the amount of traffic got lost during the MN's movement.

The blue ones in the graph are actions involving the SDN controller. The controller will compute a new set of policies based on the updated topology and flush away all the flooding flow rules. After the moment when new rules are installed, the MN could be reached via direct routing.

Let $D^{SBMP}$ be the handover delay of SBMP, which could be expressed as

$$D^{SBMP} = D_{L2} + D_{flood} + D_{ctl} + d_{wd}(L_{pi}, h_{ctl-e}) + d_{wd}(L_{fr}, h_{ctl-e}) \tag{5.8}$$

Where $L_{pi}$ and $L_{fr}$ stand for the packet size of a PACKET_IN and FLOW_MOD. And $D_{flood}$, $D_{ctl}$ are the flooding delay and controller processing delay, respectively. The flooding delay will depend on how many levels are configured

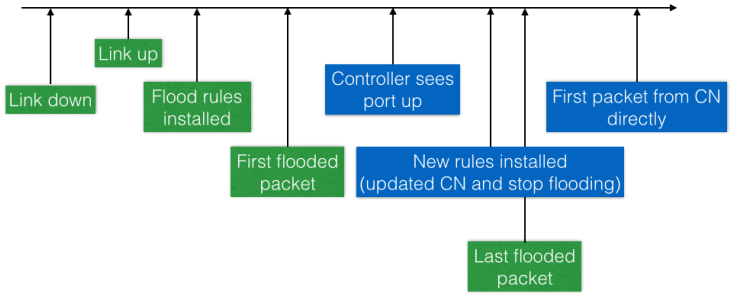



FIGURE 5.2: Handover time line for SBMP

on the spanning tree and the propagation delay over those wired links. After receiving the link up message, controller will need to recompute the flow rules based on new topology and then push the rules down to edge switches closest to the CN.

#### 5.2.3.2 MIPv6 handover delay

Figure 5.3 shows the timing diagram for MIPv6 handover. And the detailed handover process is already discussed in 3.2. Note that we didn't enable routing optimization (RO) on our testbed (Chapter 3). So we will analysis the implications of enabling RO as a supplementary discussion.

Suppose the $L^{MIPv6}$ is the handover delay of MIPv6, which could be expressed as

$$D^{MIPv6} = D_{L2} + D_{MD} + D_{DAD} + D_R \tag{5.9}$$

where $D_{L2}$, $D_{MD}$, $D_{DAD}$, $D_R$ are layer 2 delay, movement detection delay, DAD delay and registration delay, respectively.

The movement detection consists the MN sending out router solicitation (RS) and receiving router advertisement (RA). According, $D_{MD}$ could be expressed as

$$D_{MD} = d_{wl}(L_{RS}) + d_{wl}(L_{RA}) \tag{5.10}$$

In Figure 5.3, $D_{RH}$ presents the binding delay for registering with HA, which could be expressed as

$$\begin{aligned} D_{RH} = {} & d_{wl}(L_{BU-HA}) + d_{wd}(L_{BU-HA}, h_{h-e}) \\ & + d_{wl}(L_{BA-HA}) + d_{wd}(L_{BA-HA}, h_{h-e}) \end{aligned} \tag{5.11}$$

Similarly, $D_{RC}$ refers the binding delay for updating the CN, expressed as

$$\begin{aligned} D_{RC} = {} & d_{wl}(L_{BU-CN}) + d_{wd}(L_{BU-CN}, h_{c-e}) \\ & + d_{wl}(L_D) + d_{wd}(L_D, h_{c-e}) \end{aligned} \tag{5.12}$$

And $h_{c-e}$ could be calculated by adding $h_{c-h}$ and $h_{h-e}$.

From [39] we know that DAD will depend on the value of *RetransTimer*. And the registering delay $D_R$ could now be rewritten as [18]

$$D_R = D_{RC} + \max\{D_\alpha, D_\beta\} \tag{5.13}$$

Where $D_\alpha$ is the required time to exchange the HoTI and HoT messages via the HA and and $D_\beta$ is the required time to exchange the CoTI and CoT messages directly with the CN. The two items could be expressed as [18]

$$\begin{aligned} D_\alpha &= d_{wl}(L_{HoTI}) + d_{wd}(L_{HoTI}, h_{c-e}) + d_{wl}(L_{HoT}) + d_{wd}(L_{HoT}, h_{h-e}) \\ D_\beta &= d_{wl}(L_{CoTI}) + d_{wd}(L_{CoTI}, h_{c-e}) + d_{wl}(L_{CoT}) + d_{wd}(L_{CoT}, h_{c-e}) \end{aligned} \tag{5.14}$$

To put all above equations together, we could have a formula expressing handover delay of Mobile IPv6 when RO is enabled.

$$D^{MIPv6} = D_{L2} + d_{wl}(L_{RS}) + d_{wl}(L_{RA}) + D_{DAD} + D_{RC} + \max\{D_\alpha, D_\beta\} \tag{5.15}$$

Meanwhile, if RO is disabled as discussed in Section 3.3, the $D_R$ should only contain $D_{RH}$ as now notifying CN is not being notified MN's movement.

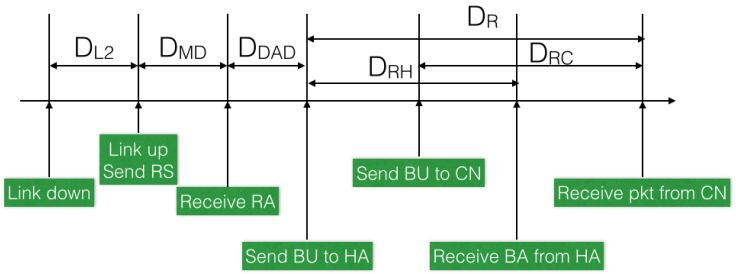



FIGURE 5.3: Handover time line for MIPv6

$$D^{MIPv6} = D_{L2} + d_{wl}(L_{RS}) + d_{wl}(L_{RA}) + D_{DAD} + D_{RH} \tag{5.16}$$

#### 5.2.3.3 PMIPv6 handover delay

Figure 5.4 shows the timing diagram of handover process of Proxy Mobile IPv6 (PMIPv6). Unlike Mobile IPv6, PMIPv6 manages the movement of the mobile nodes in a localized manner [12]. Suppose the $D^{PMIPv6}$ is the handover delay of PMIPv6, which could be expressed as

$$D^{PMIPv6} = D_{L2} + D_{LMA} \tag{5.17}$$

where $D_{LMA}$ involves the required time to (1) send the RS message, (2) exchange PBUPBA pairs between the MAG and the LMA, (3) receive the first packet sent from the LMA. Then $D_{LMA}$ could be rewritten as

$$D_{LMA} = d_{wl}(L_{RS}) + d_{wd}(L_{PBU}, h_{g-e}) + d_{lma-packet} \tag{5.18}$$

where $d_{lma-pcaket}$ is the time which the first data packet sent from the LMA arrives at the MN. Note that when LMA receives the PBU from MAG, it sends data packets destined for MN *along* with the PBA via the bidirectional

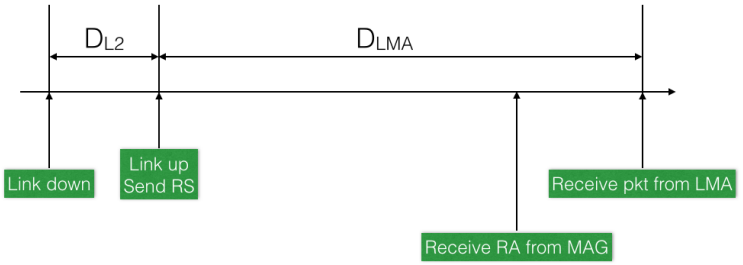



FIGURE 5.4: Handover time line for PMIPv6

tunnel between the LMA and the MAG. Accordingly, $d_{lma-packet}$ could be expressed as

$$d_{lma-packet} = d_{wl}(L_D) + d_{wd}(L_D + L_T, h_{g-e}) \tag{5.19}$$

### 5.2.4 Protocol cost

To evaluate SBMP and to compare it with MIPv6 and PMIPv6, we will analysis protocol cost in this section. Mainly, we will cover signaling cost $C_S$ and packet delivery cost $C_{PD}$. The signaling cost is the accumulative mobility signaling overhead for supporting mobility service and will be calculated by summing up the product of mobility signaling message and the hop distance [40]. The packed delivery cost is calculated as the product of the data packet size and the hop distance.

An MN's movement will incur packet transmission cost and processing cost on mobility management related entities. The first part, packet transmission cost, is proportional to the distance between source and destination nodes. Assume the unit transmission costs on a wired and a wireless link are $\alpha$ and $\beta$, respectively. Then the transmission cost on a wired and wireless link would be

$$C_{wd} = \alpha d \quad \text{and} \quad C_{wl} = \beta d \tag{5.20}$$

where $d$ is the average distance between any two nodes in the network.

The second part, processing cost will depend on the protocol design and computing power assigned to those mobility management entities. For each protocol, we will consider intra-mobility and inter-mobility respectively and assume the total signaling cost would be the sum of those two parts.

#### 5.2.4.1 SBMP cost modeling

For SBMP, the signaling cost mainly consists following parts:

- The local agent at each edge switch will trigger installation of flooding rules
- The controller re-compute the topology and update the edge switches closest to CN

Note that there are other parts of signaling cost with SBMP. For example, an edge switch generate a PACKET_IN message and the controller responds with a location tag. These signaling cost are ignored in this discussion as we are now focusing on those cost incurred only by MN's movement.

The signaling cost for SBMP $C^{SBMP}$ could be expressed as

$$C_S^{SBMP} = E(N_l)(L_{port-up}\alpha h_{ctl-e} + \lambda_S L_{flow-mod}\alpha h_{ctl-e}) \tag{5.21}$$

where $\lambda_S$ is the average session arrival rate to MN's wireless interface.

#### 5.2.4.2 MIPv6 cost modeling

Signaling cost for Mobile IPv6 mainly consists two parts

- Registering with Home Agent (HA)
- Registering with Corresponding Node (CN) (when Route Optimization is enabled)

For the first part, signaling cost for registering with HA $C_{HA}^{MIPv6}$ could be expressed as

$$C_{HA}^{MIPv6} = L_{BU-HA}(\alpha h_{h-e} + \beta h_{e-m}) + L_{BA-HA}(\alpha h_{h-e} + \beta h_{e-m}) \quad (5.22)$$

where $L_{BU-HA}$ and $L_{BA-HA}$ are the size of BU message and BA message, respectively. While $\alpha$ and $\beta$ are weighting factors for a wired link and a wireless link as mentioned in Section 5.2.1. Similarly, signaling cost for registering with CN $C_{CN}^{MIPv6}$ could be expressed as [17]

$$\begin{aligned} C_{CN}^{MIPv6} =& (L_{HoTI} + L_{HoT})(\alpha(h_{h-e} + h_{c-h}) + \beta h_{e-m}) \\ &+ (L_{CoTI} + L_{CoT})(\alpha h_{c-e} + \beta h_{e-m}) \\ &+ L_{BU-CN}(\alpha h_{c-e} + \beta h_{e-m}) \end{aligned} \quad (5.23)$$

The signaling cost of MIPv6 can be rewritten as

$$C_S^{MIPv6} = E(N_l)(C_{HA}^{MIPv6} + C_{CN}^{MIPv6}) \quad (5.24)$$

where $E_{N_l}$ is the expected number of handover counts.

#### 5.2.4.3 PMIPv6 cost modeling

Proxy Mobile IPv6 removes the necessities for an MN involving in the entire signaling process. Thus, signaling cost of PMIPv6 $C^{PMIPv6}$ could be written as

$$C_S^{PMIPv6} = E(N_l)(2L_{PBU}\alpha h_{g-e} + 2L_{PBA}\alpha h_{g-e}) \quad (5.25)$$

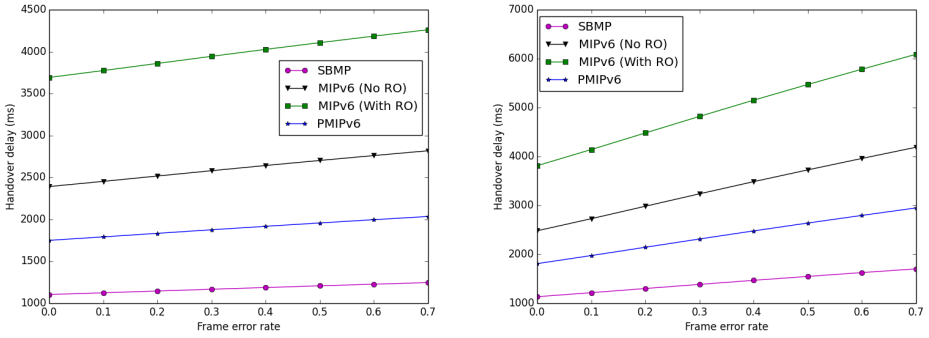
Since there is no involvement of MN in the signaling process, there is no cost over wireless links.

### 5.2.5 Numerical analysis results and discussions

We ran a numerical evaluation based on models discussed above, and the parameter values are adopted from [18, 19, 38] as shown in Table 5.3.

TABLE 5.3: Parameters used for numerical results

| Notation | Meaning | Value |
|---|---|---|
| $h_{ctl-e}$ | Distance between SDN controller and edge switches (hops) | 1 |
| $h_{g-h}$ | Distance between a gateway to Home Agent (hops) | 4 |
| $h_{g-e}$ | Distance between a gateway to edge switch (hops) | 4 |
| $h_{m-e}$ | Distance between Mobile Node to edge switches (hops) | 1 |
| $D_{L2}$ | Layer 2 handover delay (ms) | 40 |
| $D_{DAD}$ | DAD delay (ms) | 500 |
| $BW_{wired}$ | Bandwidth of a wired link (Mbps) | 100 |
| $L_s$ | Size of the control signaling message (bytes) | 8 |
| $L_{pi}$ | Size of the PACKET_IN message (bytes) | 128 |
| $L_{RS}$ | Size of the route solicitation message (bytes) | 52 |
| $L_{RA}$ | Size of the route advertisement message (bytes) | 80 |

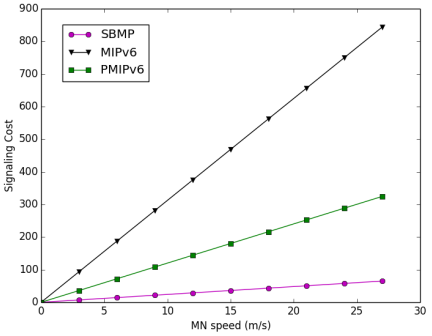



(A) $D_{wl} = 10ms$ (B) $D_{wl} = 40ms$

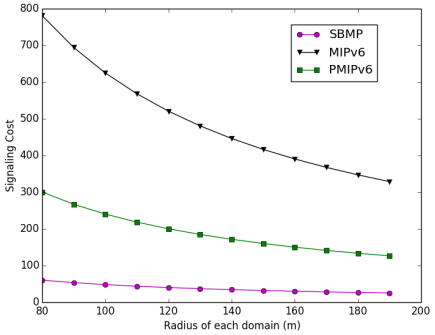
FIGURE 5.5: Handover delay versus $\rho_f$

#### 5.2.5.1 Handover delay

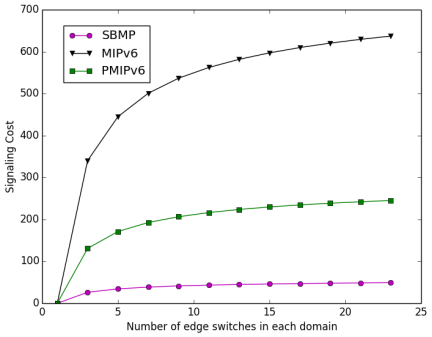
We first calculated the handover latency based on models in Section 5.2.3 and the results are shown in Figure 5.5a and Figure 5.5b. Higher $\rho_f$ will lead to erroneous packet transmission, and causing the number of mobility signaling retransmissions increasing. The value of $D_{wl}$ will also effect the handover delay. We could see that SBMP outperforms the other protocols in term of handover latency. It is mainly because that the flooding mechanism enables handling mobility in a local fashion that we are not seeing too much signaling in SBMP.




(A) R=100m, M=20




(B) Speed=20m/s, M=20




(C) Speed=20m/s, R=100m

FIGURE 5.6: Signaling cost

#### 5.2.5.2 Signaling cost

We also ran the model to measure signaling cost. And the results are shown in Figure 5.6. We could see that SBMP introduces the least cost compared to Mobile IPv6 and Proxy Mobile IPv6. The signaling cost in this model is defined by multiplying the hop numbers and signal message size. Since SBMP is flooding the traffic as the first reaction and do signaling at a later stage, plus the fact that the signaling only happens between SDN controller and the edge switch, the signaling cost of SBMP is the lowest compared to Mobile IPv6 and Proxy Mobile IPv6.

## 5.3 Experiment on a simple topology

As mentioned in Section 4.4, we implemented our proposal using Mininet 2.2.0 and Pyretic. We ran a series of evaluations based on a customized network topology shown in Figure 5.7.

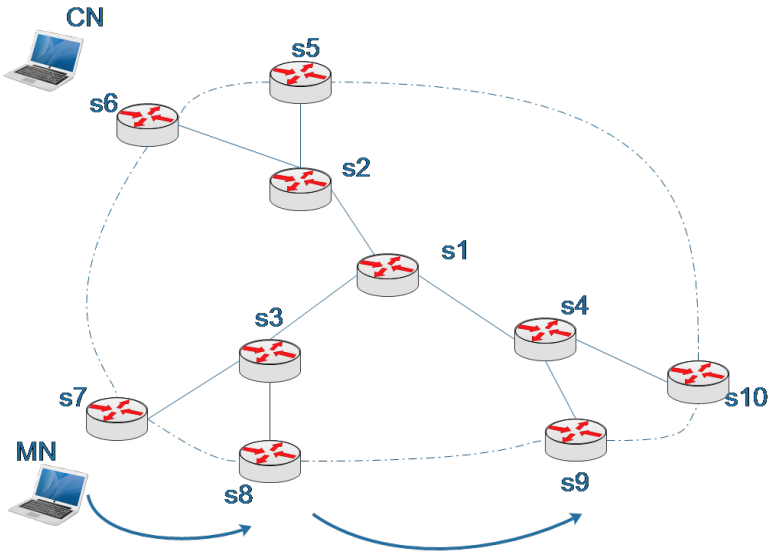
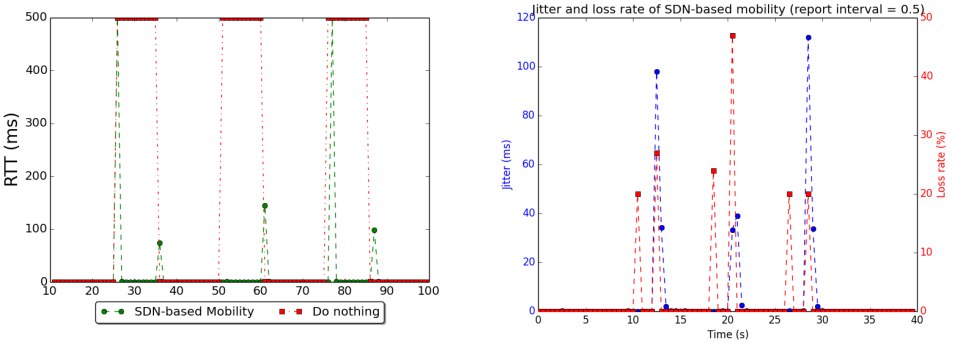



FIGURE 5.7: Experiment on a simple topology



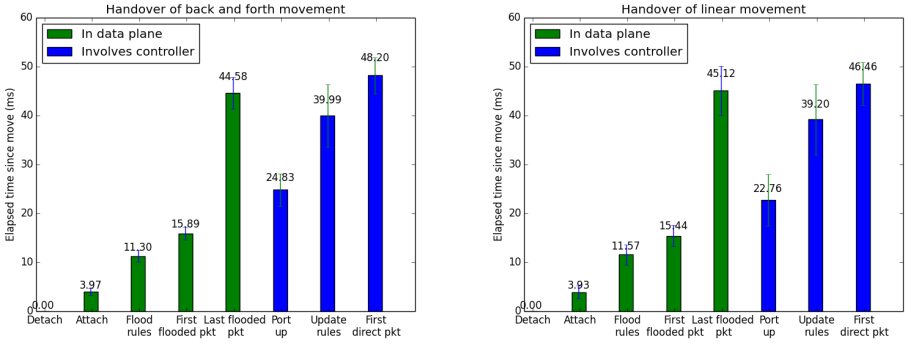

(A) Reduce the number of lost packets during movement

(B) Jitter and loss rate of SDN-based mobility management

FIGURE 5.8: Simulation results with `ping` and `iperf`

Our first simulation is to ping the mobile node while from a corresponding node (CN) and keep the mobile node (MN) moving. The MN will perform three handovers along its moving path `s7 - s8 - s9 - s10`. As present in Figure 5.8a, our flooding mechanism reduced the number of lost packets during a MN's movement.

We also ran iperf [41] between CN and MN to collect packet loss rate and jitter information. Shown in Figure 5.8b, we could see that there are two spikes along with each handover. The first corresponds to the moment when flooding rules are triggered, while the second one refers to the moment when new rules from the controller is installed. If we refer back to the same experiment conducted on the Mobile IPv6 testbed, shown in Figure 3.4, we can tell the




(A) Back and forth movement (B) Linear movement

FIGURE 5.9: Handover delay on a simple topology with simulation

differences. And we need to address here: those two experiment are not conduced on the same basis, here we have a simulation with Mininet, while the results of Mobile IPv6 were collected from a physical testbed.

As shown in Figure 5.9a and Figure 5.9b, the reachability is recovered from the moment flooding rules are installed. And after the new computed rules replaced those flooding rules, the MN could be reached via direct routing, which happened roughly 50 ms after the detachment.

# Chapter 6

# Conclusion

## 6.1 Conclusion

In this work we address mobility management using the concept of SDN, which provides flexibilities that was hard to leverage in the legacy network. After setting up a testbed on Mobile IPv6 and conducted handover related experiment, we analysis handover latency caused by multiple sources. Motivated by the the results we saw on the testbed, we carefully design SBMP: a mobility management protocol based on SDN. And we also implemented SBMP on Mininet with Pyretic.

In SBMP, the tagging mechanism reduces the amount of necessary traffic between controller and under layer devices. By introducing a flooding mechanism, the mobility is supported in a series of local actions.

We conducted analytical comparison with Mobile IPv6 and Proxy Mobile IPv6. Also, we collected handover latency in mininet as well. Those results show that SBMP is able to provide a faster handover.

## 6.2 Future work

One clear future work would be taking users' movement pattern into consideration. We could possibly learn patterns of users' movement and apply those as discussed in Section 4.3.2.

As for other future work, implementing the proposal based on a physical testbed and compare that with what we collected from Mobile IPv6 testbed

would be the next step. As for now, we don't have a direct comparison between two protocols except a simulation and computational results. Comparing different protocols on the same basis would provide fairness to our evaluation.

Considering security issues will also become an interesting future work. Especially, as one MN is moving, traffic addressed to it will be flooded in the network, and this will potentially lead to a privacy issue. On the other hand, usually we can't predict where the MN will end up to. A possible solution is to apply some existing user mobile models and only flood traffic to those possible destinations.

# Appendix A

# Related publication

Xiaozhou Jia, Panagiotis Georgopoulos,Laurent Vanbever, Karin Anna Hummel, Yuji Sekiya, Hajime Tazaki, "SDN-based Mobility Management", in *Proceedings of IEICE IN Conference*, Hokkaido, Japan, July 2015.

# Bibliography


[1] Ericsson. Ericsson Mobility Report. http://www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf, 2015.

[2] Cisco. Introduction to Mobile IP. http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/mobile_ip/mobil_ip.html, .

[3] Cisco. PMIPv6: A Network-Based Localized Mobility Management Solution. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-3/133_pmipv6.html, .

[4] Younes Khadraoui, Xavier Lagrange, and Annie Gravey. A Survey of Available Features for Mobile Traffic Offload. pages 421–424, 2014.

[5] Skype. http://www.skype.com/en/.

[6] C. Perkins. Ip mobility support for ipv4, revised. Internet Requests for Comments, November 2010. ISSN 2070-1721. URL http://www.rfc-editor.org/rfc/rfc5944.txt. http://www.rfc-editor.org/rfc/rfc5944.txt.

[7] C. Perkins, D. Johnson, and J. Arkko. Mobility support in ipv6. RFC 6275, RFC Editor, July 2011. URL http://www.rfc-editor.org/rfc/rfc6275.txt. http://www.rfc-editor.org/rfc/rfc6275.txt.

[8] Zhenkai Zhu, Ryuji Wakikawa, and Lixia Zhang. A survey of mobility support in the internet. *RFC 6301*, March 2011. URL https://tools.ietf.org/html/rfc6301.

[9] Lixia Zhang, Ryuji Wakikawa, and Zhenkai Zhu. Support mobility in the global internet. In *Proceedings of the 1st ACM Workshop on Mobile Internet Through Cellular Networks*, MICNET '09, pages 1–6, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-753-0. doi: 10.1145/1614255.1614257. URL http://doi.acm.org/10.1145/1614255.1614257.

[10] H Anthony Chan, Hidetoshi Yokota, Jiang Xie, Pierrick Seite, and Dapeng Liu. Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues. *Journal of Communications*, 6(1):4–15, February 2011. ISSN 1796-2021. doi: 10.4304/jcm.6.1.4-15. URL http://ojs.academypublisher.com/index.php/jcm/article/view/4086.

[11] Abdullah Gani, Golam Mokatder Nayeem, Muhammad Shiraz, Mehdi Sookhak, Md Whaiduzzaman, and Suleman Khan. A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing. *Journal of Network and Computer Applications*, 43:84–102, August 2014. ISSN 10848045. doi: 10.1016/j.jnca.2014.04.009. URL http://linkinghub.elsevier.com/retrieve/pii/S1084804514000927.

[12] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy mobile ipv6. RFC 5213, RFC Editor, August 2008. URL http://www.rfc-editor.org/rfc/rfc5213.txt. http://www.rfc-editor.org/rfc/rfc5213.txt.

[13] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical mobile ipv6 (hmipv6) mobility management. RFC 5380, RFC Editor, October 2008. URL http://www.rfc-editor.org/rfc/rfc5380.txt. http://www.rfc-editor.org/rfc/rfc5380.txt.

[14] R. Koodli. Mobile ipv6 fast handovers. RFC 5568, RFC Editor, July 2009.

[15] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia. Fast handovers for proxy mobile ipv6. RFC 5949, RFC Editor, September 2010.

[16] Xavier Pérez-Costa, Marc Torrent-Moreno, and Hannes Hartenstein. A performance comparison of mobile ipv6, hierarchical mobile ipv6, fast handovers for mobile ipv6 and their combination. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(4):5–19, October 2003. ISSN 1559-1662. doi: 10.1145/965732.965736. URL http://doi.acm.org/10.1145/965732.965736.

[17] Jong-hyouk Lee, Thierry Ernst, and Tai-myoung Chung. Cost Analysis of IP Mobility Management Protocols for Consumer Mobile Devices. 56 (2):1010–1017, 2010.

[18] Jong Hyouk Lee, Jean Marie Bonnin, Ilsun You, and Tai Myoung Chung. Comparative handover performance analysis of IPv6 mobility management protocols. *IEEE Transactions on Industrial Electronics*, 60(3): 1077–1088, 2013. ISSN 02780046. doi: 10.1109/TIE.2012.2198035.

[19] Christian Makaya and Samuel Pierre. An analytical framework for performance evaluation of ipv6-based mobility management protocols. *Wireless Communications, IEEE Transactions on*, 7(3):972–983, 2008.

[20] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008. ISSN 0146-4833. doi: 10.1145/1355734.1355746. URL http://doi.acm.org/10.1145/1355734.1355746.

[21] Mao Yang, Yong Li, Depeng Jin, Lieguang Zeng, Xin Wu, and Athanasios V. Vasilakos. Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey. page 12, August 2014. doi: 10.1007/s11036-014-0533-8. URL http://arxiv.org/abs/1409.0079.

[22] C. Giraldo, F. Gil-Castineira, C. Lopez-Bravo, and F. J. Gonzalez-Castano. A Software-Defined mobile network architecture. *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 287–291, October 2014. doi: 10.1109/WiMOB.2014.6962184. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6962184.

[23] Ntonio D E La. An Architecture for Software Defined Wireless Networking. *IEEE Wireless Communications*, pages 52–61, 2014.

[24] Slavica Tomovic, Milica Pejanovic-Djurisic, and Igor Radusinovic. SDN Based Mobile Networks: Concepts and Benefits. *Wireless Personal Communications*, 78(3):1629–1644, July 2014. ISSN 0929-6212. doi: 10.1007/s11277-014-1909-6. URL http://link.springer.com/10.1007/s11277-014-1909-6.

[25] Guolin Sun, Feng Liu, Junyu Lai, and Guisong Liu. Software Defined Wireless Network Architecture for the Next Generation Mobile Communication : Proposal and Initial Prototype. *Journal of Communications*, 9(12):946–953, 2014. doi: 10.12720/jcm.9.12.946-953.

[26] Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. Softran: Software defined radio access network. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, pages 25–30, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2178-5. doi: 10.1145/2491185.2491207. URL http://doi.acm.org/10.1145/2491185.2491207.

[27] Mao Yang, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. Openran: A software-defined ran architecture via virtualization. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 549–550, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2056-6. doi: 10.1145/2486001.2491732. URL http://doi.acm.org/10.1145/2486001.2491732.

[28] Li Erran Li, Z Morley Mao, and Jennifer Rexford. Cellsdn: Software-defined cellular networks. *Techinical Report, Princeton University*, 2012.

[29] Xin Jin, Li Erran Li, Laurent Vanbever, and Jennifer Rexford. Softcell: Scalable and flexible cellular core network architecture. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 163–174, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2101-3. doi: 10.1145/2535372.2535377. URL http://doi.acm.org/10.1145/2535372.2535377.

[30] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. Openroads: Empowering research in mobile networks. *SIGCOMM Comput. Commun. Rev.*, 40(1):125–126, January 2010. ISSN 0146-4833. doi: 10.1145/1672308.1672331. URL http://doi.acm.org/10.1145/1672308.1672331.

[31] Kok-Kiong Yap, Masayoshi Kobayashi, David Underhill, Srinivasan Seetharaman, Peyman Kazemian, and Nick McKeown. The stanford openroads deployment. In *Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization*, WINTECH '09, pages 59–66, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-740-0. doi: 10.1145/1614293.1614304. URL http://doi.acm.org/10.1145/1614293.1614304.

[32] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. Towards programmable enterprise wlans with odin. In

*Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pages 115–120, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1477-0. doi: 10.1145/2342441.2342465. URL http://doi.acm.org/10.1145/2342441.2342465.

[33] Manu Bansal, Jeffrey Mehlman, Sachin Katti, and Philip Levis. Openradio: A programmable wireless dataplane. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pages 109–114, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1477-0. doi: 10.1145/2342441.2342464. URL http://doi.acm.org/10.1145/2342441.2342464.

[34] Rajeev Koodli. Fast handovers for mobile IPv6. https://tools.ietf.org/html/rfc4068, 2005.

[35] UMIP - Mobile IPv6 and NEMO for Linux. http://umip.org/.

[36] Pyretic. http://frenetic-lang.org/pyretic/.

[37] Mininet. http://mininet.org/.

[38] N. Banerjee, K. Basu, and S.K. Das. Hand-off delay analysis in sip-based mobility management in wireless networks. In *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, pages 8 pp.–, April 2003. doi: 10.1109/IPDPS.2003.1213412.

[39] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), September 2007. URL http://www.ietf.org/rfc/rfc4861.txt. Updated by RFC 5942.

[40] PetroPesha Ernest, H.Anthony Chan, OlabisiEmmanuel Falowo, and LinohA. Magagula. Distributed mobility management with distributed routing management at access routers for network-based mobility support. *Wireless Personal Communications*, pages 1–25, 2015. ISSN 0929-6212. doi: 10.1007/s11277-015-2602-0. URL http://dx.doi.org/10.1007/s11277-015-2602-0.

[41] iperf. https://iperf.fr/.