Mobility Management in New Internet Architectures

Kuljaree Tantayakul

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :

Mme KULJAREE TANTAYAKUL

le lundi 17 septembre 2018

Titre :

Mobility Management in New Internet Architectures

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

MME BEATRICE PAILLASSA

M. RIADH DHAOU

Rapporteurs :

M. ANIS LAOUITI, TELECOM SUD PARIS

M. MICHEL MISSON, UNIVERSITE CLERMONT-FERRAND 1

Membre(s) du jury :

Mme MICHELLE SIBILLA, UNIVERSITE TOULOUSE 3, Président

M. KAMAL SINGH, UNIVERSITE DE SAINT-ETIENNE, Membre

Mme BEATRICE PAILLASSA, INP TOULOUSE, Membre

M. RIADH DHAOU, INP TOULOUSE, Membre

M. SIDI MOHAMMED SENOUCI, UNIVERSITE DE BOURGOGNE, Membre

Acknowledgement

First of all, I would like to thank my supervisor, Prof. Dr. Béatrice PAILLASSA, a professor at Institut National Polytecnique de Toulouse. This thesis would not have been possible without her feedback. She has given the value opportunity to me for a student under her care. She has encouraged to do this thesis. She has devoted a lot of time and has always given the suggestions and support to me.

I would like to thank another my supervisor, Assoc. Prof. Dr. Riadh DHAOU. He has devoted a lot of time to me. He has always given the comments and the guidance to me.

I would like to express my sincere thanks to Prof Michel MISSON, Professor at the Université Clemont Auvergne and Associate Professor Anis LAOUITI at TELECOM Sud-Paris, who have given me the honor of accepting to be the reviewer for this thesis.

I would also like to thank Prof. Michelle SIBILLA, Professeur at the Université Paul Sabatier, and Prof. Sidi-Mohammed SENOUCI, Professeur at the Université de Bourgogne, and Asst. Prof. Kamal SINGH at Telecom Saint Etienne / Université Jean Monnet for their interest in my research by participating as the jury.

I would like to thank the administration team at the ENSEEIHT office, namely Sam, Isabelle, Annabelle and Fred Peyré for their kindness and assistance. I express my deepest gratitude to all and wish them well.

I would like to thank my friends, Dr. Wasimon PANICHPATTANAKUL, Dr. Chakadkit THAENCHAIKUN and his wife, Dr. Charlie KREY, KEEGAN family, CONQUY family, and Nattawee SAE-Heng, for their support to come here and spend my life here. In addition, I would like to thank my colleagues and friends in IRIT laboratory for their assistance. Among them, I would like to thank Qiankun, Ahmad, Farouk, Élie, Yoann, Dorin, Oana, Adrien, Mouna, and Amal.

Last but not least, I would like to thank my parent for their love and everything.

Abstract

The software integration with new network architectures via Software-Defined Networking (SDN) axis appears to be a major evolution of networks. While this paradigm was primarily developed for easy network setup, its ability to integrate services has also to be considered. Thus, the mobility service for which solutions have been proposed in conventional architectures by defining standardized protocols should be rethought in terms of SDN service. Mobile devices might use or move in SDN network. In this thesis, we proposed a new mobility management approach which called "SDN-Mobility" and has shown that SDN can be implemented without IP mobility protocol for providing mobility like as Proxy Mobile IPv6 (PMIPv6) that is the solution adopted by 3GPP, with some performance gain. However, PMIPv6 and SDN-Mobility have some packets loss during Mobile Node (MN) handover. Thus, in this thesis, we proposed a new paradigm based on caching function to improve the quality of transfer during handover. Caching policy cooperates with SDN controller for automatic buffering of the data during the handover. We proposed two caching policies that are compared through a performance analysis regarding the quality of transfer for the user and for the operator. This thesis also presented that SDN-Mobility with caching policy can be applied easily for mobility management in heterogeneous network architectures able to integrate the future Internet based on the Information-Centric Networking (ICN).

Résumé

L'intégration logicielle avec les nouvelles architectures réseau via l'axe SDN (Software-Defined Network) apparaît comme une évolution majeure des réseaux. Bien que ce paradigme ait été principalement développé pour faciliter la configuration du réseau, sa capacité à intégrer les services doit également être prise en compte. Ainsi, le service de mobilité pour lequel des solutions ont été proposées dans des architectures classiques en définissant des protocoles normalisés devrait être repensé en termes de service SDN. Les appareils mobiles peuvent utiliser ou se déplacer dans le réseau SDN. Dans cette thèse, nous avons propose une nouvelle approche de gestion de la mobilité appelée "SDN-Mobility" et montré que SDN peut alors sans protocole de mobilité IP fournir une mobilité comme Proxy Mobile IPv6 (PMIPv6) qui est la solution adoptée par 3GPP, avec un gain de performance. Toutefois, PMIPv6 et SDN-Mobility présentent des pertes de paquets lors du transfert du nœud mobile (MN). Ainsi, dans cette thèse, nous avons proposé un nouveau paradigme basé sur la fonction de mise en cache pour améliorer la qualité du transfert lors du déplacement du mobile. La stratégie de mise en cache coopère avec le contrôleur SDN pour la mise en mémoire tampon automatique des données pendant le transfert. Nous avons proposé deux politiques de mise en cache qui sont comparées à travers une analyse de performance concernant la qualité du transfert pour l'utilisateur et pour l'opérateur. Cette thèse a également présenté que SDN-Mobility avec la politique de mise en cache peut être facilement appliquée pour gérer la mobilité dans des architectures de réseau hétérogènes capables d'intégrer le futur Internet basé sur les réseaux centrés sur l'information (ICN).

Contents

List of Tables

List of Abbreviations

AAA	Authentication, Authorization and Accounting
AP	Access Point
API	Application Programming Interface
AR	Access Router
BCE	Binding Cache Entry
CCN	Content-Centric Networking
CDN	Content Delivery Network
CN	Correspondent Node
CV	Connected Vehicle
DASH	Dynamic Adaptive Streaming over HTTP
DMM	Distributed Mobility Management
DMM-GW	Distributed Mobility Management Gateway
eNB	E-UTRAN Node B
FMIPv6	Fast Mobile IPv6
HMIPv6	Hierarchical Mobile IPv6
HNP	Home Network Prefix
ICN	Information Centric Networking
LMA	Local Mobilility Anchor
MAG	Mobile Access Gateway
MIP	Mobile IP
MIPv4	Mobility support for IPv4
MIPv6	Mobile Internet Protocol version 6
MN	Mobile Node
MN-ID	Mobile Node Identifier
NAR	New Access Router
NDN	Name Data Networking
NetLMM	Network based Localized Mobility Management
NSPs	Network Service Providers
OF-PMIPv6	OpenFlow PMIPv6
OPMIPv6	OpenFlow-based Proxy Mobile IPv6
OFSW-AP	OpenFlow Switch-Access Point
PAR	Previous Access Router
PBA	Proxy Binding Acknowledgement
PBU	Proxy Binding Update
RA	Router Advertisement

RLC	Radio Link Control protocol
RS	Router Solicitation
RSU	Road Side Unit
RTP	Real-time Transport Protocol
SaDMM	SDN-aided Distributed Mobility Management
SDMM	Software-Defined Mobility Management
SDN	Software Defined Networking
SW	Switch
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Chapter 1

Introduction

Contents

1.1 Context

Generally, the basic problem in supporting Internet mobility is to deliver data to a mobile receiver, whose location in the network topology is dynamically changing. Mobile IP (MIP) [1] [2] is one of the earliest and most well-known protocol. Mobile IP is an Internet Engineering Task Force (IETF) standardized protocol which allows Mobile Nodes (MN) to move from one network to another without losing connectivity. Several solutions were proposed to improve IP mobility management such as Fast Mobile IPv6 (FMIPv6) [3], Hierarchical Mobile IPv6 (HMIPv6) [4] and Proxy Mobile IPv6 (PMIPv6) [5]. Even FMIPv6 and HMIPv6 improved handover latency, they are host-based mobility management protocols. The MN needs to modify its protocol stack to support mobility signaling.

Consequently, IETF decided to develop a Network-based Localized Mobility Management (NetLMM) solution where the network entities take the responsibility of exchanging mobility signaling on behalf of the MN. Mobility is achieved without requiring the nodes to have some specific configuration or software installation. The main advantage of PMIP compared to other MIP solutions that are host-based is that only the network is concerned by the mobility

management, not the mobile. As mobile has no signaling to exchange, it is a more flexible and convenient solution to use in a real network. While the PMIP solution has advantages that made the 3GPP standardizes it, its relevance in the context of new network architectures arises.

Originally, each network device could be controlled and managed individually. But, each device of the different vendor has different firmware, and the forwarding and control planes are coupled into one box. Thus, it is not flexible and hard to manage. The Software-Defined Networking (SDN) [6] [7] aims to introduce flexibility by leveraging the software components of the network. It is a new approach to computer networking that can help the network administrators to configure, update and monitor the different network devices and the various manufacturers more accessible through a software application. It makes the addition of network function easier. Thus, it is not surprising that many network deployments are using SDN for flexible and more comfortable management.

Information-Centric Networking (ICN) [8] [9] is a proposed future Internet architecture. ICN is a clean-slate redesign of the current Internet architecture that shifted from host-based to content-based location-independent communication. It is an appealing technology to provide not only mobility but also security and storage as native properties of the network architecture. The principal concern of the network is to expose, find and deliver information rather than the reachability of end-hosts and the maintenance of conversations between them.

1.2 Research Significance and Reasons

SDN is applied to the Internet for managing network services, but IP mobility management protocols are applied to manage mobile services. SDN is an emerging network architectural approach whose key idea is to separate control and data plane of networks. As the most of IP mobility rely on a centralized entity which tied of both control and data planes together, when the network individually runs SDN and IP mobility, it possibly occurs some redundancy between the components of mobility and the components of SDN. This leads to question. Is it possible to manage mobility in SDN? There are two feasible ways to achieve mobility management in SDN. The first way needs to modify the mobility protocol for co-operation with SDN signaling and the second way needs to find a new method based on SDN signaling to provide mobility service without the legacy mobility protocol. In this thesis, we proposed a novel mobility management approach without the legacy mobility protocol which is called "SDN-Mobility" by using the powerful OpenFlow protocol [10] [11] [12]. This approach has the advantage over a traditional method based on mobility management by IP protocols, such as the PMIP approach well suited to operator networks, to avoid the use of tunnels which

saves time and bandwidth. Tunnel avoidance can also be done in the traditional method, by direct routing as soon as the source knows the location of the destination, i.e., the destination IP address. Meanwhile, security and privacy objectives considerably increase the mobility management complexity. The implementation of the approach, which we have explained in Chapter 4, is based on access switches signaling to the controller, through OpenFlow protocol, the location of the mobiles they discover instead of using the traditional mobility signaling (the PMIP messages) to indicate to a mobility agent the location of the users.

The SDN-Mobility approach improves the operation of the network as the loss rate is decreased thanks to the simplification of the mobility control. However, the data transfer service suffers from losses when changing mobile access points. The loss impacts on the user's Quality of Experience (QoE). In general, the loss problem can be solved by caching scheme which can be separated into two schemes based on the caching place: user caching and network caching. In this thesis, we study QoE metrics of both user (Dynamic Adaptive Streaming over HTTP (DASH)) and network caching that acts quite differently. This leads to difficulty to compare the QoE of user caching (DASH) and network caching. The user caching scheme (DASH) buffers the small data at the user side. This technique improved QoE that allows a user to smoothly playback video by buffering and adapting the requested video bitrate. However, when a mobile node changes its point of attachment, the requested video still suffers from loss. While the network caching scheme prevented/minimized the loss during handover. We propose an innovative approach that solved the packet loss problem by adding a caching function to the SDN equipment. We show the interest of such a function on the SDN equipment and study the parameters of the caching policy to answer the questions: Which content to cache? and for how long? We propose to use a caching policy with an SDN-Mobility approach to improve the packet loss. The caching duration is based on the network quality evaluation with various policies. Two caching schemes, ON-OFF caching and Adaptive caching, are introduced and compared in terms of user quality and operator quality.

The coexistence of traditional IP network and future network (ICN) for mobility is a hot topic that challenges in communications and networking since ICN shifted from host-based to content-based communication. It was designed with no backward compatibility with IP network architecture regarding different network architecture, different routing systems, and various software applications. Hence, IP and ICN will coexist for a longer period. Thus, to maintain connectivity of a mobile node when it moves in heterogeneous network architectures (IP and ICN), solutions are required. In this thesis, we proposed the concept of SDN mobility solution which can be applied for the coexistence of heterogeneous network architectures.

1.3 Thesis Organization

This thesis is organized into eight chapters. The brief description of the chapters is presented as follows:

Chapter 1: Introduction

The first chapter, the Introduction, provides the significance and reasons of this thesis. In addition, it presents the organized outline of this thesis in this section. It also contains the contribution and research objectives.

Chapter 2: Technological Background

This chapter provides the technological background and related works done in relation to this thesis. It presents the mobility problems and the existing mobility protocols. It also explains future network architectures.

Chapter 3: State of the Art of Mobility Management in Future Internet Architecture

This chapter provides a review of mobility management approaches for the future Internet architecture. It also presents the comparison and analysis of each approach.

Chapter 4: Proposal for Software-Defined Mobility Management (SDMM)

This chapter presents our first proposal for Software-Define Mobility Management (SDMM) which called "SDN-Mobility". It presents the designed SDN-Mobility architecture and operation that is based-on SDN concept. It also provides the experimental network topology for comparison of performance between PMIPv6 and SDN-Mobility. Moreover, this chapter explains the characteristics of WiFi connectivity by the analysis of experimental result in the different scenarios. It also evaluates the data loss during MN handover which will be improved in the next chapter.

Chapter 5: Proposal Improvement with Caching Policy

This chapter, first explains the impact of loss on the perceived video quality that motivates to improve SDN-Mobility with caching policy. It indeed gives the design and implementation of caching algorithms.

Chapter 6: Evaluation of SDN-Mobility with Caching Policy

This chapter provides the evaluation of SDN-Mobility with caching policy in pedestrian and vehicular context for improving the performance of SDN-Mobility in two aspects: user and operator.

Chapter 7: Mobility Management in Heterogeneous Network Architectures

This chapter explains the mobility problems in heterogeneous network architecture and presents the possible solutions that can be applied including SDN-Mobility with caching policy for seamless mobility management in heterogeneous network architectures.

Chapter 8: Conclusion and Perspectives

The final chapter gives the conclusion of this thesis and shows the advantages of the proposed approach. It also gives some perspectives for future works.

1.4 Contributions

The main contributions of this thesis can be summarized as follows:

- We propose SDN-Mobility which is a new mobility management approach to allow the mobile node freely moves in the localized SDN network without the legacy Mobility protocol implementation. The proposed approach minimizes the handover cost and decreases the control operation.
- We propose two caching policies: Adaptive and ON-OFF caching policy to improve the quality of transfer during handover in a pedestrian context. For a vehicular network, an ON-OFF policy is applied, the data of vehicles can be uploaded to the target network before the handover.
- We propose the possible approaches to manage mobility services in heterogeneous network architectures by applying SDN-Mobility with caching policy.

2

Chapter 2

Technological Background

Contents

This chapter deals with the context of the thesis regarding mobility issues and network architecture. First, we present the problem of mobility as it has been studied for several years, that is to say in a distributed Internet IP context. In a second time, we present the environment of the new Internet architectures to highlight the points of innovation as well as the potential solutions for the management of the mobility in these new network architectures. We are going to pay particular attention to Software-Defined Networking (SDN) architectures, by highlighting the interest of SDN in a mobile application, the vehicular application.

2.1 Mobility Problem

Let us first clarify the notion of mobility and then examine the different approaches of solving that have been studied for many years to manage the change of network, the handover.

2.1.1 What is Mobility?

With the ability to freely and easily move of the users thanks to its equipment and wireless access networks, it is possible to move to another network while communicating with another via the Internet infrastructure. When a user moves to another network, depending upon the network architecture, it can also change its IP network and so the IP socket used by the communication in progress is no longer available. The mobility management function has the task of making the network change transparent to the application so that there is no need to restart the application. It assures the continuity of the communication.

The mobility management is achievable in different ways, by acting at various levels (IP, Transport, Application SIP..) and by different methods. In this thesis, we focus on the IP level management, and we are going to rely on two approaches: mapping based, and routing based, explained in the following.

2.1.2 Overview of the Mobility Approaches

Mobility in IP networks is a well-known problem which has been the subject of many solution proposals. They can be classified into three basic approaches: routing-based, mapping-based, and combination approaches [13]. The classification of the existing mobility protocols according to the above basic approaches are shown in TABLE A.1 (in Appendix). The indicated works have been carried out for more than 20 years. The earliest and the latest works reported in the table concern respectively Columbia [14] and LISP approach [15], of which the RFC was published in 2013.

The concept of routing-based approaches is to mask the user mobility by the routing. The user can keep her address while she has changed her location.

In order to keep the user mobility, two strategies have been proposed. One is broadcast-based, and the other is a path-based. In the former case, either the mobile location information is actively broadcasted to the whole network or a proactive broadcast query is needed to obtain the location information such as Columbia [14] and Connexion [16]. On the other

hand, a path is maintained by the routing system instead such as Cellular IP [17], HAWAII [18], TIMIP [19], etc. This approach, which has also been used in MANET (Mobile Ad Hoc Networks), is particularly suitable for small-scale environments. On the other hand, the signaling and routing updates generated by location tracking do not pass the Internet scale.

In contrast, mapping-based approaches allow the user to change her IP address but with keeping a piece of stable information, known as an identifier (while IP address is rather known as a locator) which does not change during handover. Instead of notifying the world of every movement, a user only needs to update a single binding location about her location change. These approaches can be achieved through either a network-based strategy or host-based strategy. Almost protocols are based on a host-based strategy that requires any modification of user such as Mobile IP (MIP), Hierarchical Mobile IP (HMIP) [4], Fast Handover for Mobile IPv6 (FMIPv6) [3], LISP-Mobility [15], etc. The only PMIPv6 protocol is a network-based mobility protocol which does not require any software implementation at the user equipment.

Since a routing-based approach is not scaling to provide mobility in the global Internet, this thesis focuses on mapping-based approaches with IPv6 (because all the studies and perspectives are thought in IPv6 even if IPv4 is still in deployment). In particular, the old basic solution MIP, the most recent LISP-Mobile Node (LISP-MN) and the network operator PMIPv6 will be highlighted.

In few words, LISP-MN differs from MIP by replacing the current IP address with two separated namespaces: Endpoint Identifiers (EIDs), and Routing Locators (RLOCs) and PMIPv6, relies on identifiers similar to those of MIP with a protocol implementation supported in the network rather than by the machine, i.e., the host.

2.1.3 The Basic Approach: Mobile IP (MIP)

Normally, the operation of Internet routers is to forward packets to the destination network as indicated by the destination IP address of the packet header. Routing tables typically maintain the next-hop information for each IP network destination. The IP address identifies a network interface to deliver the packet. However, as the Internet architecture has evolved, the network interfaces have come to be numbered by which network they are attached. Thus, in practice, the IP address has two functions: the identification of the interface but also its localization with regards to its network attachment. This means that the IP address has to change at a new point of attachment. To alter the routing of IP packets intended for mobile to its new point of attachment requires a new IP number associated with that new point of

attachment. However, to maintain the existing transport protocol layer connections as the mobile moves, the mobile's IP address must remain the same.

This problem has been solved by the mobility standard protocol which is called Mobile IP or MIP. MIP uses a so-called permanent address that does not change while communicating. It plays the role of the identifier and a temporary address that makes it possible to locate the mobile via its attachment network. IETF has developed the mobility protocol to support both IPv4 (Mobility support for IPv4 (MIPv4) [1]) and IPv6 (Mobile Internet Protocol version 6 (MIPv6) [2]). These standards provide a distinction between two roles of IP address, by allocating: Home Address (HoA), as an identifier and Care-of Address (CoA), as a locator. The mobile has to support the mobility protocol for handling the mobility signaling in charge of information exchanges about addresses. It is thus classified as a host-based protocol. It may cause some problems of maintainability, each time the protocol evolves, and a host has to be set up, but also there is some loss of control from the network operator.

Let's briefly fix how MIP works and what is its limitation.

2.1.3.1 MIPv6 Operation Overview

A mobile (Mobile Node: MN on Figure 2.1) uses its Home Agent (HA) when communicating with other nodes. The bidirectional tunnel between its HA and its current location (CoA) is established after mobile moved from its home network to the others. CoA will be changed as the mobile roamed. The association between the mobile's HoA and CoA is called a binding that is performed at HA.

When the mobile moves away from its home network, it sends a Binding Update (BU) message to inform its HA with new CoA. Then HA receives this BU, and it sends a Binding Acknowledgement (BA) to mobile for indicating the status of the registration. When the packets from the Corresponding Node (CN) are routed to the home network, the HA intercepts them and forwards them over a tunnel to mobile. Each intercepted packet is tunneled to the mobile's CoA. Packets to the correspondent are be tunneled from the MN to the HA (reverse tunnel), after that the packets are routed normally from the home network to the correspondent as illustrated in Figure 2.1.

2.1.3.2 Triangle Routing Limitation

The triangle routing situation appears when the correspondent node of the mobile sends its data to the home network of the mobile, that afterward forwards them to the actual network

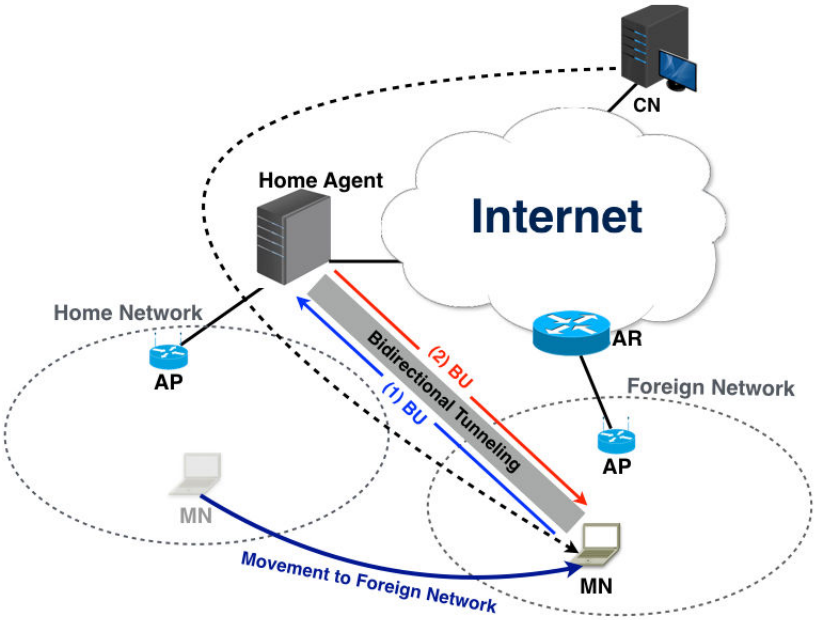

Figure 2.1 MIPv6 Operation

attachment of the mobile, while the correspondent could transmit them directly to the actual mobile attachment. The inverse situation is also possible when the mobile transmits to its correspondent. More precisely, the correspondent does not know the current mobile's address (CoA), it sends all packets to mobile's HoA. Then the HA intercepts and tunnels them to CoA, as illustrated in Figure 2.2. This problem can be solved by Route Optimization technique [20] which provides missing addresses to the correspondent/mobile. It enables the correspondent to maintain a binding cache giving the CoA of the mobile for the tunneling establishment in case of mobility.

The triangle routing effect causes some performance degradation, but meanwhile, it avoids some security issues, as the user's privacy is not exposed to a correspondent.

In this thesis, we will work on a network mobility approach that overcomes the performance problem while keeping the user privacy by not exposing the user localization.

2.1.4 The New Approach by Name: LISP

The current addressing scheme identifies host and its location by the same item leading to mobility problems in the existing IP network. Therefore, Locator/ID Separation (LISP) protocol [15] was proposed to separate into two different numbering spaces: Endpoint

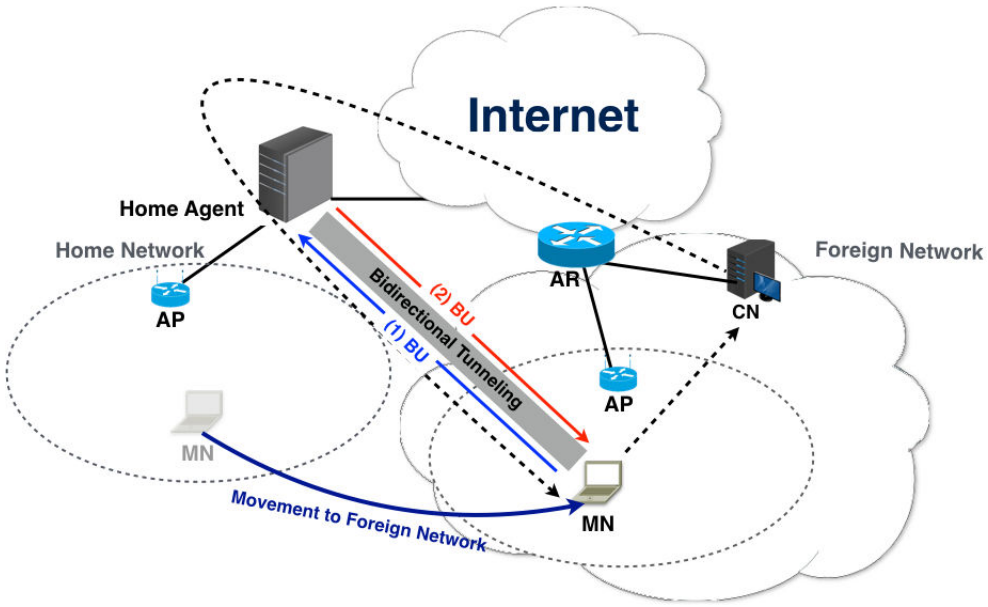

Figure 2.2 Triangle Routing Problem

Identifiers (EIDs) and Routing Locators (RLOC). LISP-Mobile Node (LISP-MN) [21] is based on the LISP protocol which obtains a permanent EID to each mobile for all transport connections. Then, when the mobile changes its point of attachment, only the RLOC of the mobile is changed, leading to enabling seamless endpoint mobility by allowing the application to bind to the EID. LISP-MN supports mobility in both IPv4 and IPv6 networks with the map-and-encapsulate scheme. The mapping of EID to RLOC is achieved by some routers through a mapping resolver system acting as an overlay over the Internet.

LISP-MN is a promising approach that provides the seamless mobility. Meanwhile, it is a host-based mobility protocol based on an encapsulation scheme, generating overhead. As for us, we have based the thesis over the existing network approach, PMIP.

2.1.5 The Standard Network Approach: Proxy Mobile IPv6 (PMIPv6)

Because of Mobile IP can suffer from a long handover latency, IETF designed several solutions by extension for Mobile IP such as Fast Mobile IPv6 (FMIPv6) [3], that introduce communication between previous and next access routers, Hierarchical Mobile IPv6 (HMIPv6) [4] based on an hierarchical architecture that circumvents the signaling, and so improves the delay and the scalability, and Proxy Mobile IPv6 (PMIPv6) [5]. Even though FMIPv6 and HMIPv6 improved handover latency, they are host-based approaches protocol

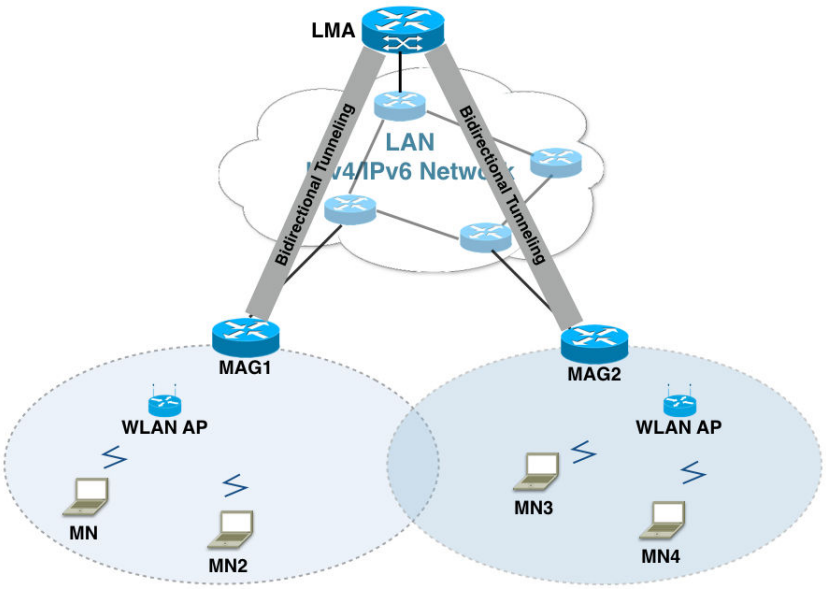

Figure 2.3 The Architecture of PMIPv6

in which mobile needs to modify its protocol stack to support mobility signaling, with a lack of flexibility.

Consequently, IETF decided to develop a Network-based Localized Mobility Management (NetLMM) [22] solution where the network entities take the responsibility of exchanging mobility signaling on behalf of the MN. Mobility is achieved without requiring the nodes to have some specific configuration or software installation.

The IETF has developed proxy Mobile IPv6 (PMIPv6) as one network-based localized mobility management approach which uses IP Tunneling and without any modifications to the mobile. It has been specified in RFC 5213 and has been adopted by 3GPP. The work presented in this thesis is based on PMIPv6, and so, we explain in more details its architecture and operation.

2.1.5.1 PMIPv6 Architecture

The PMIPv6 architecture considers two main functional entities, the Local Mobility Anchor (LMA) and the Mobile Access Gateway (MAG) as illustrated in Figure 2.3.

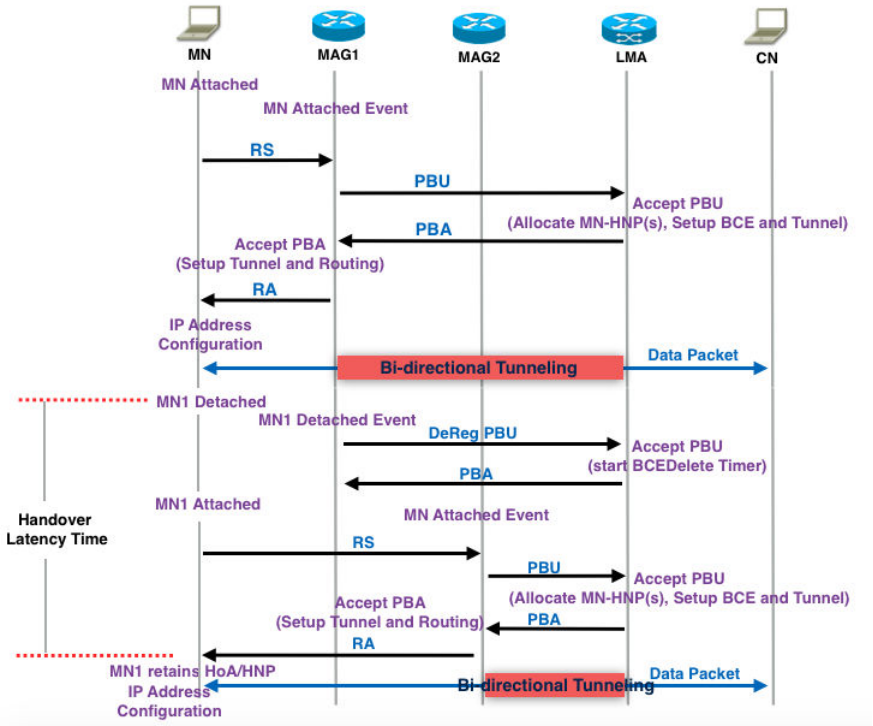

Figure 2.4 The PMIPv6 Signaling

- **Access Gateway (MAG)**

  The entity named Mobile Access Gateway (MAG) handles mobility signaling on behalf of the mobile while it is attached to its access link. MAG is responsible for detecting the mobile movement event and for signaling from LMA. It uses the acquired identifier of Mobile Node (MN-ID) to elaborate a modified Proxy Binding Update (PBU) for the authentication process. Note that, the authentication process is under the control of the operator, that is for the operator an interesting characteristic of the network scheme.

- **Local Mobility Anchor (LMA)**

  Local Mobility Anchor (LMA) is the home agent of the mobiles in a Proxy Mobile IPv6 domain. It provides the Home Network Prefix (HNP) and manages the mobile's binding state. It also keeps a Binding Cache Entry (BCE) for each registered mobile.

- **Mobile Node (MN)**

  Mobile Node (MN) is an IP host which does not require any software modification on MN.

2.1.5.2 PMIPv6 Operation

An illustration of the PMIPv6 functioning is given in Figure 2.4 through the exchanges of messages between the entities of the architecture composed of one LMA and two MAGs, named MAG1 and MAG2, according to the following scenario.

The first part is the registration process.

The MN enters to a PMIPv6 network, MAG1 senses the MN attached the event. The MAG1 uses the acquired identifier of Mobile Node (MN-ID) to modify a PBU and sends Proxy Binding Update (PBU) to LMA for the registration process.

After the LMA received the PBU message, it checks the MN-ID in Binding Catch Entry (BCE). If not present in BCE, it is added by LMA. Then, LMA provides the Home Network Prefix (HNP) of the mobile to MAG1 by sending back a Proxy Binding Acknowledgement (PBA). LMA configures the IP tunneling of its side at the same time.

When MAG1 receives the PBA from LMA, the IP tunnel between LMA and MAG1 is established. A Router Advertisement (RA) message is advertised in the access link by MAG1. It provides a prefix for the mobile.

If the mobile does not receive RA, it sends a Router Solicitation (RS) message for requesting the RA. The IPv6 address of the mobile can be modified thanks to the IPv6 autoconfiguration, based-on EUI-64 standard algorithm [23].

All data communication between MN and Corresponding Node (CN) are transmitted through the established bi-directional tunneling.

Concerning the handover process there are:

MN is roaming to the other attachment, it detaches from MAG1, MAG1 senses this event and sends PBU for the de-registration procedure to LMA.

LMA receives the PBU message and starts the BCEDelete timer for deleting the entry of MN in BCE. After that, the PBA message is replied from LMA to MAG1.

During the time that MAG2 senses the mobile attached event and sends the PBU message to LMA, LMA adds the new entry in BCE, configures the IP tunnel and sends PBA message to MAG2. Then, the tunnel between MAG2 and LMA is established.

MAG2 advertises the HNP to mobile by sending a RA message. After the mobile autoconfiguration is completed, all data packets are transmitted through the established tunnel between MAG2 and LMA.

2.1.5.3 Basic PMIPv6 Operation Messages

There are 4 basic messages, two are linked to the IPv6 autoconfiguration (RFC4861 [24] and RFC4862[25]), Router Solicitation (RS) message and a Router Advertisement (RA) message, and the two other messages are for the location mapping update, Proxy Binding Update (PBU) message and Proxy Binding Acknowledgement (PBA) message are used to update BCE in PMIPv6 operation. More details are explained in the Appendix.

2.1.5.4 Central routing limitation for localized communication

The localized routing occurs when both endpoints are located in the same PMIPv6 domain [26]. In this case, the packets are always forwarded through the central LMA instead of directly be routed by the MAG to each other. It induces some useless increased of delay with some possible bottleneck in the central point. It increases transport cost and traffic load at the LMA. To address this problem, two new signaling messages were defined [27], Localized Routing Initiation (LRI) and Local Routing Acknowledgement (LRA). These signaling messages are used to initial localized routing that the mobility entities (LMA or MAG) could decide to initiate in three different scenarios as follows.

The first scenario is two MNs attached to the same MAG and LMA as illustrated in Figure 2.5. The localized routing can be enabled in this scenario by LMA which sends a LRI message to MAG after detecting that the two MNs are attached to the same MAG.

In scenario 2: the MNs are attached to different MAGs but with the same LMA, (Figure 2.6). It is similar to scenario 1 as it is the LMA that decides to do localized routing. The LMA informs an IPv6 address of another MAG with a LRI message to establish the tunneling between MAGs for localized routing.

Scenario 3: Two MNs attached to the same MAG with different LMAs, this time it is the MAG which determines that MNs are attached to the same MAG. It informs to both LMAs for localized routing with the LRI messages as illustrated in Figure 2.7.

The above scenarios illustrate the problem of routing, already reported in the MIP solution with the triangular routing, with, also, a centralized control problem at the LMA that can be

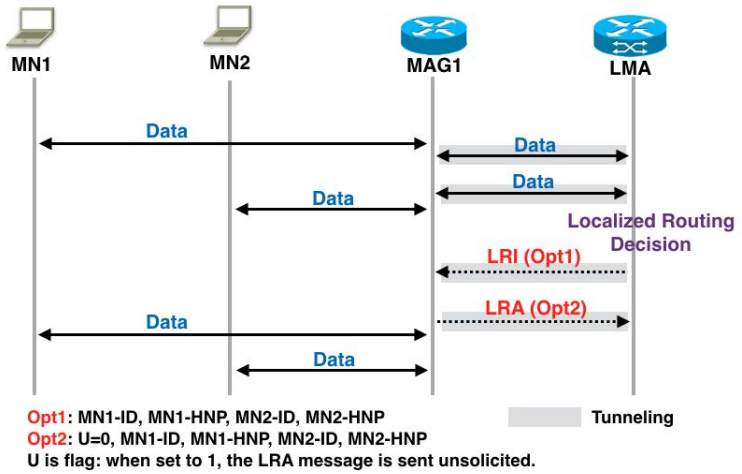

Figure 2.5 Scenario 1: Two MNs attached to the same MAG and LMA

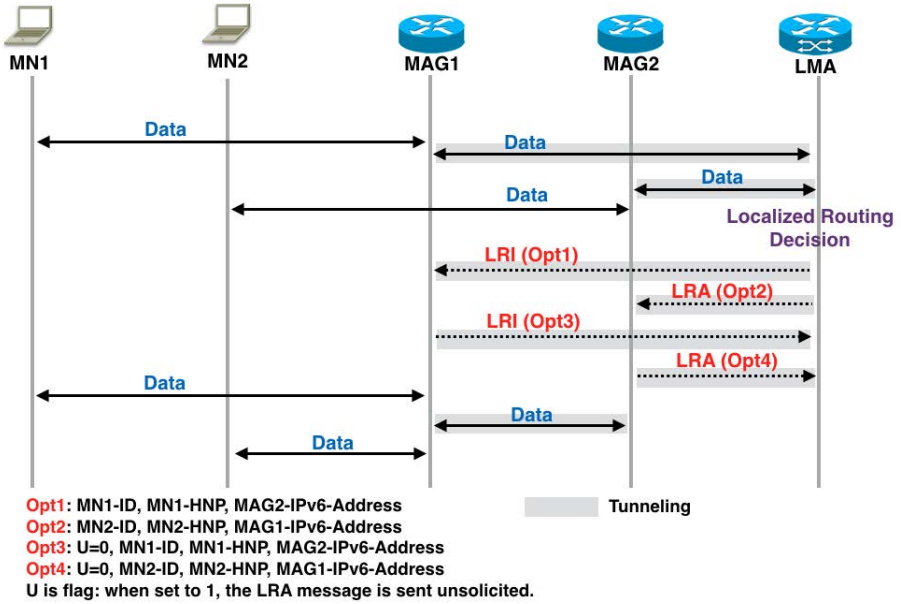

Figure 2.6 Scenario 2: Two MNs attached to different MAGs but the same LMA

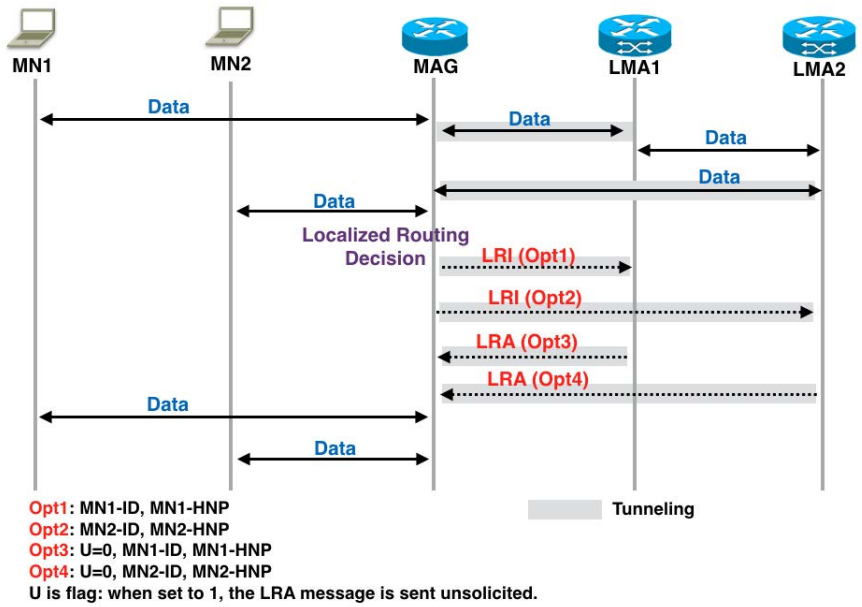

Figure 2.7 Scenario 3: Two MNs attached to the same MAG with different LMAs

solved by a routing shortcut at the MAG. We will see in Chapter 4 how these two features are particularly suitable for implementation by Software Defined Architecture.

Now that we have explained the context of the thesis regarding mobility, we will in the second part present the innovative context of the network architectures that we considered.

2.2 Characteristics of Future Network Architectures

As the Internet was not designed for its current level of usage, Internet today is a complex agglomerate of protocols that inherits the grown legacies of decades of patchwork solutions. The future network architectures are required to simple construct to allow the network to do better in terms of flexibility security, mobility, quality of service, and quality of experience. To achieve this goal, the networks should be designed to have the following properties as key points to future network architectures: centralization, virtualization, network caching, and mobile edge computing.

Within architectural approaches able to meet these requirements, we are particularly interested in SDN and Information-Centric Networking (ICN) architectures. The first one can easily act on the routing as we have shown as being a major concern for the mobility management, while the second includes a naming paradigm that could be helpful for mobility. Moreover,

caching is very interesting when applied together with edge computing, as we will see in the next section.

2.2.1 Virtualization and Centralization: SDN and NFV

Because the networks are deployed with a large and increasing variety of proprietary hardware appliances, it makes the network operators difficult to launch a new network service often because it requires to find the space and power to accommodate by using yet another variety of procedure. Therefore, Network Functions Virtualization (NFV) [28] [29] [30] was designed to address this problem. The concept of NFV is the transformation of the network architectures through the implementation of network functions and network operations in software. The software can dynamically be added or moved to various locations in the network as required, without the need for installation of new equipments.

The deployment of NFV requires large-scale dynamic network connectivity both in the physical and virtual layers to interconnect virtual network function endpoints. The MANO (network functions virtualization management and orchestration) system, actually in standardization progress, would take advantage of the SDN architecture, because this last is intended to set a programmable network, by a separation between the data plane and the control plane.

Traditionally, the data and the control plane in the networking devices (and most of the communication principles) have been tied together. This mean, the prevailing operating system and its features with the provided hardware are implemented in a single device. Therefore, network devices such as switches, routers, firewalls, etc., are built with the intelligence of handling traffic relative to the adjacent devices. This makes the intelligence distributed and scattered in the network. In Figure 2.8 is illustrated the switch architecture of both traditional and SDN.

As explained above, it is now possible to add the network processes by dynamically installing new network functions. This is the basic assumption of our work. We can think of new treatments, and they could be usable in the real. While in the basic Internet, it would be too complex.

2.2.1.1 Software Defined Networking Architecture and Devices (SDN)

Software-Defined Networking (SDN) [6][7] was designed to allow the network administrators to automatically and dynamically manage, control and monitor a large number of network

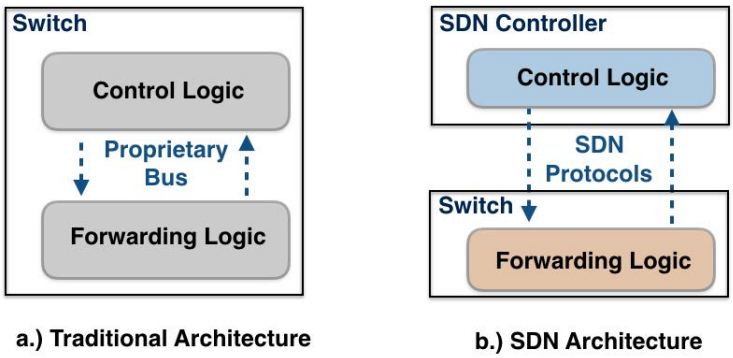

Figure 2.8 Traditional and SDN Architecture

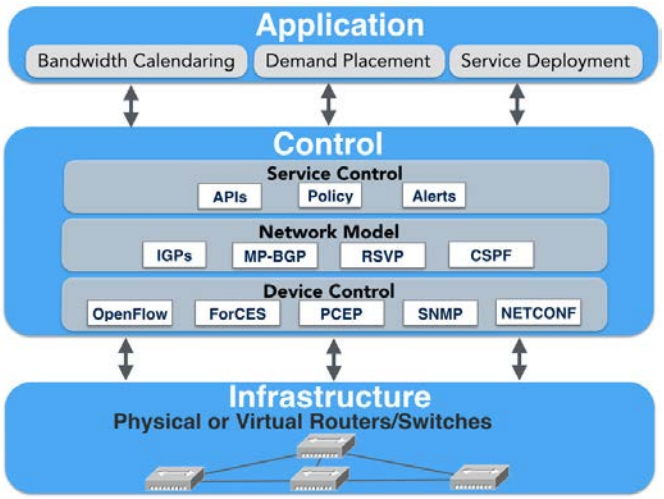

Figure 2.9 SDN Architecture

devices with the network programmability, and it is intented to improve network management and configuration methods [31]. The concept of SDN is based on a centralized management system by dividing the operation into two parts: the control plane and the data plane. Therefore, some protocols are required for the operation methods between the control plane and the data plane such as NETCONF [32], Border Gateway Protocol (BGP) [33], Open vSwitch Database Management Protocol (OVSDB) [34], MPLS Transport Profile (MPLS-TP) [35], and the OpenFlow protocol [10] [11].

In our work, we focus on SDN network with OpenFlow protocol. Its architecture is conformed to the one standardized by Open Networking Foundation a major standard actor in the area.

The SDN architecture model is separated into three layers which are application layer, control layer, and the infrastructure layer as illustrated in Figure 2.9.

- **SDN Application Layer**

  The application layer consists of the end-user business applications that consume the SDN communication services. These applications interact with the SDN control layer via Northbound Application Programming Interfaces (API). Optimized routing is a basic application of SDN. In the thesis, we are going to consider the Mobility Management application.

- **SDN Control Layer**

  The SDN control layer provides a mean to dynamically and deterministically control the behavior of network resources (such as for data transport and processing), as instructed by the application layer. The SDN applications specify how network resources should be controlled and allocated, by interacting with the SDN control layer via application-control interfaces. Then the control signaling from the SDN control layer to the network resources is delivered via resource-control interfaces. The configuration and/or properties exposed to SDN applications are abstracted by the meaning of information and data models. The level of abstraction varies according to the applications and the nature of the services to be delivered. For example, considering the routing process, the controller has to control the data transport in function of the application, as for example an energy routing [36].

  Presently, routing is the main controllable process; meanwhile, quality of service with buffering management are on the way. It seems us envisagable to introduce some additional process like the one we proposed in Chapter 5 for data buffering.

- **SDN Infrastructure Layer**

  The lowest layer is the infrastructure layer, also called the data plane. It comprises the forwarding network elements, generally referred as switches whatever the transport technology in used (MPLS, IP or Ethernet). The responsibility of this layer is mainly data forwarding, as well as monitoring local information and gathering statistics. On the information reception, it applies the rules sent by the controller. So, in the case of mobility, this element could directly transmit the information in the case of an aware mobility controller as we will propose in Chapter 4.

2.2.1.2 Implementation of SDN Architectures

To ending this SDN insight we precise at follows some implementation of the SDN architecture on which we worked. More details could be found in our paper about the opentools evaluation [37].

The controller operates as the network brain receiving and sending information from switch devices. There are many different available SDN controllers such as POX [38], RYU [39] [40], Pyretic [41], Trema [42], FloodLight [43], ONOS [44] and OpenDaylight [45]. As for us, we have chosen RYU controller because it is widely used, well documented and defines API for creating various SDN application [46] [47]. Also, it is very fast python based OpenFlow controller [48] as compared to the performance of the application with POX and Pyretic controller. Work on RYU controller [49] is in line with this choice, comparing to the other four controllers: POX, Trema, FloodLight, and OpenDayLight [50], it is claimed as the best. Last but not the least, only RYU controller was fully supported IPv6 at the start of our work [51].

In few lines, RYU [39] [40] is a component-based software-defined networking framework that is located between the SDN applications and OpenFlow switches as illustrated in Figure A.1 (a) and Figure A.1 (b) (in Appendix). All of the code is freely available under the Apache 2.0 license. RYU communicates with OpenFlow protocol, the layer 2 communication protocol on top of the Transmission Control Protocol (TCP) [52]. OpenFlow1.5 is the current version. OpenFlow version 1.2 is the first version to support IPv6 matching such as IP protocol number, IPv6 source/destination address, traffic class, flow label, and ICMPv6 types/codes. More IPv6 function was added in OpenFlow 1.3 such as the ability to rewrite packet header via flexible match support and more ability to match IPv6 header fields such as protocol number (next header, extension header) and tunneling.

2.2.1.3 Limitations

Even though SDN brings some simplification and centralization of network management, it raises some question in terms of delay, scalability, and reliability aspects [53] [54],we can partially reply to these limitations by the multiple ongoing works.

- **Control plan delay inefficiency**

  Because the concept of SDN separates the control plane and data plane, everything must be done in the control plane, that means the first packet will regularly be forwarded to the SDN controller for a flow table setup. The first packet takes high delay due to the control plan delay [55] [56]. While any subsequent packets will not involve the controller at all and will be handled directly by the switches. This control plan delay impacts the sensitive delay applications. This problem can be avoided by proactive flow insertion before the arrival of the packets, which will allow them to be handled only by the switch without control plane delay.

- **Scalability problem**

  SDN moves the control plane out of data plane and uses only one controller in the beginning phase. But as the size of the network scales up, more events and requests are sent to the controller, and at some point, the controller cannot handle all the incoming requests. This could be a significant problem for SDN scalability with high flow initiation rate [57]. Various works have proposed to improve the scalability of SDN. That can be done in two directions. The first direction is to design a distributed architecture of SDN control planes, such as Hyperflow [58], Onix [59], and Kandoo [60]. The other direction is to move some control function of the SDN controller to switches, reducing the event requests submitted to the controller significantly, such as DIFANE [61] and DevoFlow [62].

- **Reliability problem**

  SDN architecture was designed to separate the control plane and data plane for centralized network management. Further, a new network plane between the control plane and data plane is created for connection network domain which is called a control path network [63]. In the investigation of network reliability, it is usually assumed that at any time there is at most one physically failed component in an SDN-enabled network. Then, the SDN reliability problems will occur in the following cases: failure on the master SDN controller, failure on the OpenFlow switches, and failure on the link between the SDN controller and OpenFlow switches. For example, if a SDN controller failed, the control signaling is lost, leading to whole switches not have forwarding rules then cannot forward the packets to the right destination.

  Many works have studied and proposed SDN reliability solutions that can be classified into three focal areas: data plane reliability, control path reliability, and control plane reliability.

  So for this thesis, we considered that reliability is not a major obstacle to the use of SDN.

The second architecture we will tackle is the ICN architecture, on the contrary to SDN it is a distributed architecture and it innovates by introducing caching in the network as well as a paradigm of naming, rather than addressing, that can highly influence the mobility management.

2.2.2 Naming and Network Caching: ICN

Caching technique is the process of storing data in the cache. This technique has been extensively studied in the DNS context then in the Web context and more recently in the video context with Content Delivery network. The objective is to improve the user experience and to avoid network or server congestion [64] [65], by reducing delay and load. For Content Delivery Networks (CDN) popular contents are replicated on many servers spread throughout the Internet to reduce latency and traffic load [66]. But CDN also has its own drawbacks and limitations, such as manageability, scalability, mobility, and security [67]. The Information-Centric Networking (ICN) proposal is to integrate directly the caching in the network. This aspect that is a key point of the ICN architecture is completed by the name paradigm. The name idea is inspired by new communication uses that seek information regardless of its location. The important thing is not where (location) but what(content). Communication takes place by exchanging named data instead of transmitting packets from a source to a destination. In ICN, the network elements are in charge of the right forwarding regardless of a location indicated by an IP address contained in the packet header.

This paradigm seems well adapted to the LISP mobility management itself based on naming [15]. For mobility management, the interest of the naming will depend on the kind of mobility, source or receiver, as it will be studied in Chapter 3.

Compared with traditional web caching and CDN caching, ICN caching takes on several new characteristics such as cache is transparent to applications, the cache is ubiquitous, and content is cached as granulation [68]. For granulating cache, most ICN proposals use the technique of slicing large files into small self identifiable chunks and perform cache operations on the unit of chunks [69] [70] [71] [72] [73] [74]. For example, Figure 2.10 illustrates the operation of coordinated in-network caching (CINC). The content is cached ubiquitous and granular in each network device. First, the host requests with the number of chunk to closest switch but does not know where the content is cached. After the closest switch received, if it does not have the requested chunk, it forwards the request to the other switches until matching the number of the chunk. Then, the switch replies the requested chunk data to the host.

In this thesis, we are going to propose in-network caching scheme as in ICN. We will consider network elements able to store source data. This function could be implemented as a NFV.

To end the context of the thesis, we present the vehicular application, that is a mobile application taking advantage of the new Internet architectures, especially by integration of processes at the network level by Mobile Edge Computing based on NFVSDN architectures.

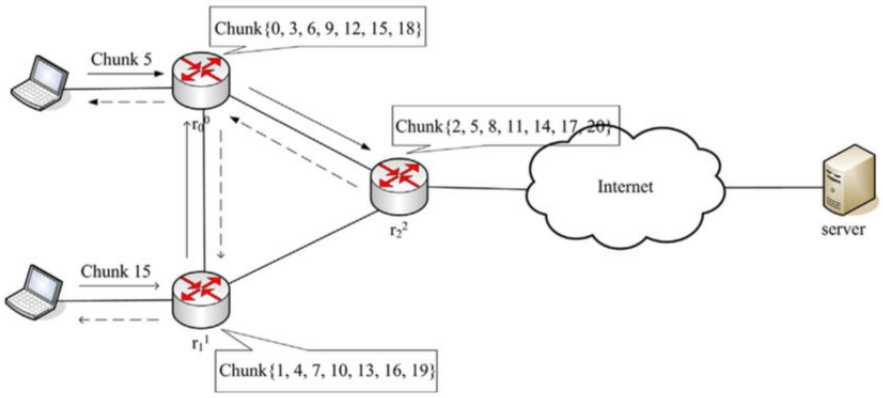

Figure 2.10 The operation of coordinated in-network caching (CINC) [74]

2.2.3 Mobile Edge Computing Use Case: Vehicular Application

Mobile Edge computing (MEC) is an understudy paradigm in next-generation wireless networks, enabling the cloud-computing capabilities close to mobile devices [75].

MEC solutions can be separated into two common trends [76]. The first trend is based on virtualization techniques exploiting NFVs principles that MEC can use to manage virtualized resources flexibly. The second trend is a decoupling of the control and data planes by taking advantage of the SDN paradigm, which allows a dynamic adaptation of the network to change traffic patterns and users requirements. The use of SDN for MEC is also in trend for current mobility network including vehicular network [51] [77] [78].

J. Liu et al. [79] proposed the SDN-enabled architecture of heterogeneous vehicular network as illustrated in Figure 2.11. SDN is used to facilitate flow entries management. This SDN-enabled architecture also makes the network management more flexible for data or resource plane exchanges.

In Figure 2.11, a vehicle communication can be exchanged information with other vehicles (V2V), surrounding infrastructure (V2I), the Internet (V2N), roadside pedestrian (V2P), and a back-end cloud server. The MEC cloud servers are responsible for computing resources and storage spaces. They are placed at the edge of the vehicular access network and are close to the mobile vehicle, which decreases the round-trip time of data packets. Also, MEC can offload traffic load from the backbone network.

In the thesis, we are going to apply our proposals for improving mobility on a use case relying on new network environments. It is the management of vehicles in a MEC architecture with dynamic adaptation of network devices.

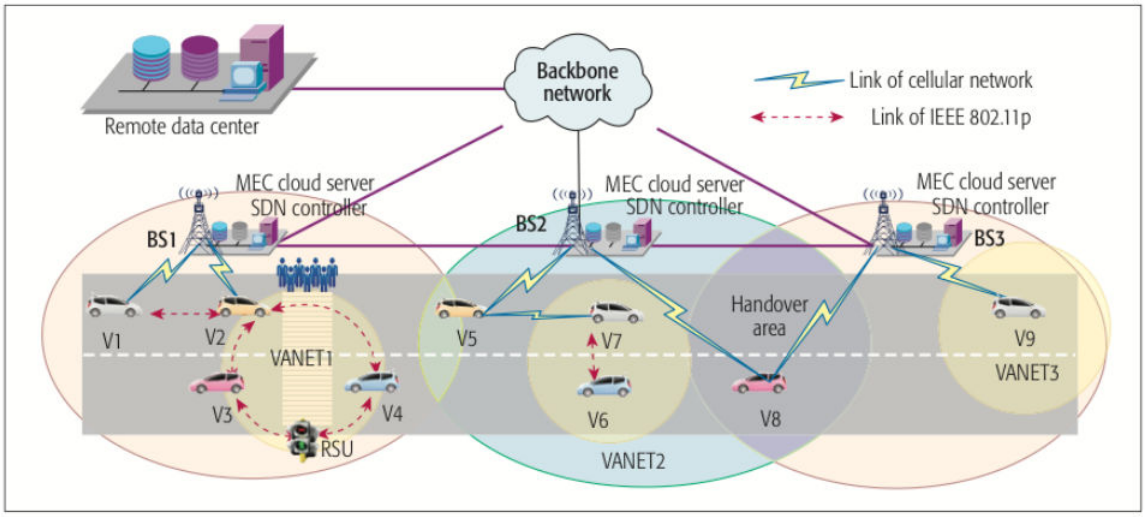

Figure 2.11 The SDN-enabled architecture of heterogeneous vehicular network [79].

2.3 Conclusion

In this chapter, we have introduced the context of Mobility. We sought to present a panorama in this field by highlighting the method, routing or mapping, and the architecture of the solution. Besides, we showed the main problems to solve to improve performance: namely triangular routing, and centralized routing for local communications.

Note that the scaling up problem is not specifically addressed, it is more the responsibility of the implementation of mobility management. When specifying the solution, which we will present in the rest of this document, we will try to reduce the signaling because even if a single message is saved when we consider the number of terminals to manage, the gain can be significant. A step beyond our work would be the implementation of a distributed architecture to relieve the load of the centralized processes. We motivate our choice to concentrate our research on PMIPv6 solution that is better suited to achieve transparent operation to the user and is in line with future network architectures that tend to increase the network intelligence.

We continued by presenting a quick overview of the characteristics of the new internet architectures based on SDN and ICN and by analyzing what could be imported from these new technologies for innovative mobility management. We have shown that the intelligent routing proposed by the SDN is an interesting track to solve the triangular routing and that the caching possibilities of the ICN in the network should be equally interesting regarding performance.

We will explore these two possibilities in the next chapter.

Chapter 3

State of the Art of Mobility Management in Future Internet Architecture

Contents

This chapter dedicates to mobile management studies in the context of new network architectures. Insofar, as we are part of a pioneering study theme, work in the field is limited and lacks synthesis. We propose an original classification and an analysis of bibliographic works. We show the different schemes and architectures as well as the limitations of the approaches implementation.

At first, we interested the mobility in an SDN network architecture context, then in an ICN context. Details of the existing approaches are discussed and compared.

3.1 SDN and Mobility Management

This section analyzes recent works in the area of Software-Defined Mobility Management (SDMM) that apply the leverage of SDN to provide mobility. Mobility management in IP networks normally requires mobility protocols implementation such as presented in the previous chapter. It is possible that mobility protocols may run in the SDN network, but it appears that operation of mobility protocols is not really relevant without considering the SDN process. At follows, we analyze the architectures approaches, and then we are going to study more particularly literature proposals based on cooperative architecture between PMIP and SDN, and SDN standalone architecture, able to support the mobility without the legacy PMIP protocol.

3.1.1 Architecture Approaches Classification

Among the many papers dealing with the SDN in telecom networks, mobility is an application generally cited, but at the time of this thesis, only a few papers provided details on its implementation. Considering more precisely this application, it appears that a key feature is its organization, it is either in a centralized scheme or a distributed one.

Mobility management as defined by the IETF is a centralized management (with for example a central point which is the home agent or the LMA) it is simply because the central entity is able to know the user movement and to reroute by tunneling the packet to the access router on which the mobile node is currently connected. However, the central point represents a problem of reliability as a scalability issue. Thus, IETF has proposed some requirements for Distributed Mobility Management (DMM) [80].

Recently [81], the DMM group has proposed to decouple the control and transfer functions to distribute them with the aim of avoiding the accumulation of traffic on a central point, to this end several models are possible including the centralization of the control, by SDN controller to better distribute traffic. For example RFC 7429 in 2015 [82] proposes to distributes the PMIP operation by deploying several LMAs and to use some selection criteria to assign LMAs for attaching mobile nodes. Moreover, they integrate localized routing function. An example of such deployment is proposed in [83] where PMIPv6 is extended at the MAG entity with some links directly connected to the Internet and also by the ability to route directly the packet from MAG to another MAG.

One of the first papers that discuss the general positioning of SDN in telecom architectures analyzes the mechanisms of tunnels and gateways and where SDN equipment can be installed. There is no clarification to indicate that some agents, as a mobile agent, can be installed on

SDN equipment and that in case of an agent or access failure, the controller can automatically transfer the traffic to another agent [84].

The solution proposed by DMM PMIP in [85] relies on access routers, and on a control point to allocate prefixes. Unlike a conventional PMIP, the traffic transmitted by the correspondent does not go through the LMA, but goes directly to the MAG to which the user is connected. In the case of handover, the mobile requests a new prefix which is then communicated to the old MAG. The latter can then forward the packets to the new access point.

Paper [86] is more focused on the study of mobility. Nevertheless, PMIP architecture is seen as a use case, without details on its implementation. Paper is interested in different ways to combine mobility architectures and SDN, using one or more controllers and maintaining mobility exchanges and SDN exchanges. In their approach, the traditional mobility management locates users, while OpenFlow is just in charge of the routing. A domain is seen as a routing domain and a mobile domain. The SDN controller is in charge of the routing domain, while the mobile PMIP domain is managed by the LMA.

When considering several mobility domains, it is possible to associate a controller to each domain and to introduce a higher hierarchical level, with a father controller responsible for the inter-domain mobility. So that, as the user moves from one domain to another, the destination domain controller warns her father controller, which in turn will warn the controller of the previous domain that it has to forward the data to the new domain.

SDN controller can improve the performance compared to a distributed PMIP approach where the data are received by the known access gateways and then relayed to next forwarding equipment until reaching the right location of the mobile. The SDN architecture has the advantage of improving the timely delivery of information since data can be directly transmitted on the right access gateway.

F. Giust et al [83] compared a distributed PMIP architecture with a SDN architecture and a BGP architecture. They focus on the data forwarding and the way to modify the path, either by PMIP or by OpenFlow or by BGP at the router level. From some Linux experiment, they show that pure PMIP solution and pure SDN solution are quite comparable in terms of handover measurement, while BGP is less attractive (we can suppose that it is due to the long BGP convergence time).

[87] considered a centralized PMIPv6 architecture combined with SDN architecture. The control of the routing and the mobility are centralized on the same equipment, and two kinds of switches are proposed, access switches and intermediate switches. SDN access switches support MAG functionality, with PMIP v6 protocol, intermediate switches only support OpenFlow (Figure 3.2). The LMA and the MAG communicate each other through PMIPv6

to notify the attachment of the mobile and to communicate the network prefix assigned by the LMA, while OpenFlow is used for routing. The two controls can also be integrated as shown in scheme (c) of Figure 3.3. In this case, PMIP signaling may be omitted.

An objective of the distributed mobility architecture is to design a flat architecture to solve some scalability problems; nevertheless, it increases the complexity of the control plane, and in case of centralized control model, it can also generate some centralization issues. In our work we have studied a centralized scheme with a SDN centralized control, easier to develop in the laboratory.

From literature analysis, we synthesize two ways to provide mobility support in the SDN network. The first way needs to modify the mobility protocol in such a way it can cooperate with SDN signaling in a cooperative architecture. The second way needs to find a new method based on SDN signaling to provide mobility service without the use of legacy mobility protocols.

3.1.2 Cooperative Architecture: SDMM with legacy PMIPv6

As follows, we examine the ways to combine SDN and PMIPv6 mobility protocol that aims to improve the PMIPv6 performance regarding signaling and/or the handover latency. The works focus on centralized PMIP architecture.

3.1.2.1 *OpenFlow-based Proxy Mobile IPv6 (OPMIPv6)*

OPMIPv6 [87] was proposed the architecture which uses OpenFlow for PMIPv6 network. Authors retain the PMIPv6 operation for notifying the attachment of mobile and forwarding the Home Network Prefix to it and use the OpenFlow messages to set up the routing path. Operations of OPMIPv6 are illustrated in Figure 3.1. The main advantage of the proposal is to avoid the tunnel establishment thanks to OpenFlow signaling.

Paper [87] compared the signaling cost of PMIPv6 with the OpenFlow PMIP architecture proposed in two modes, integrated (c) and separated (b) as in Figure 3.3. The signaling cost of the integrated control is naturally lowest because it does not use PMIPv6 mobility signaling but only OpenFlow. Also, the signaling of the separated control plane is lower than the PMIPv6 control plane one. Concerning the data plane efficiency, authors consider some given cost of transmission depends upon the element geographical localization, so by enumerating the signaling; they show a packet delivery cost of PMIPv6 which is the highest, compared to the proposal (the packet delivery cost is the same whatever the control scheme).

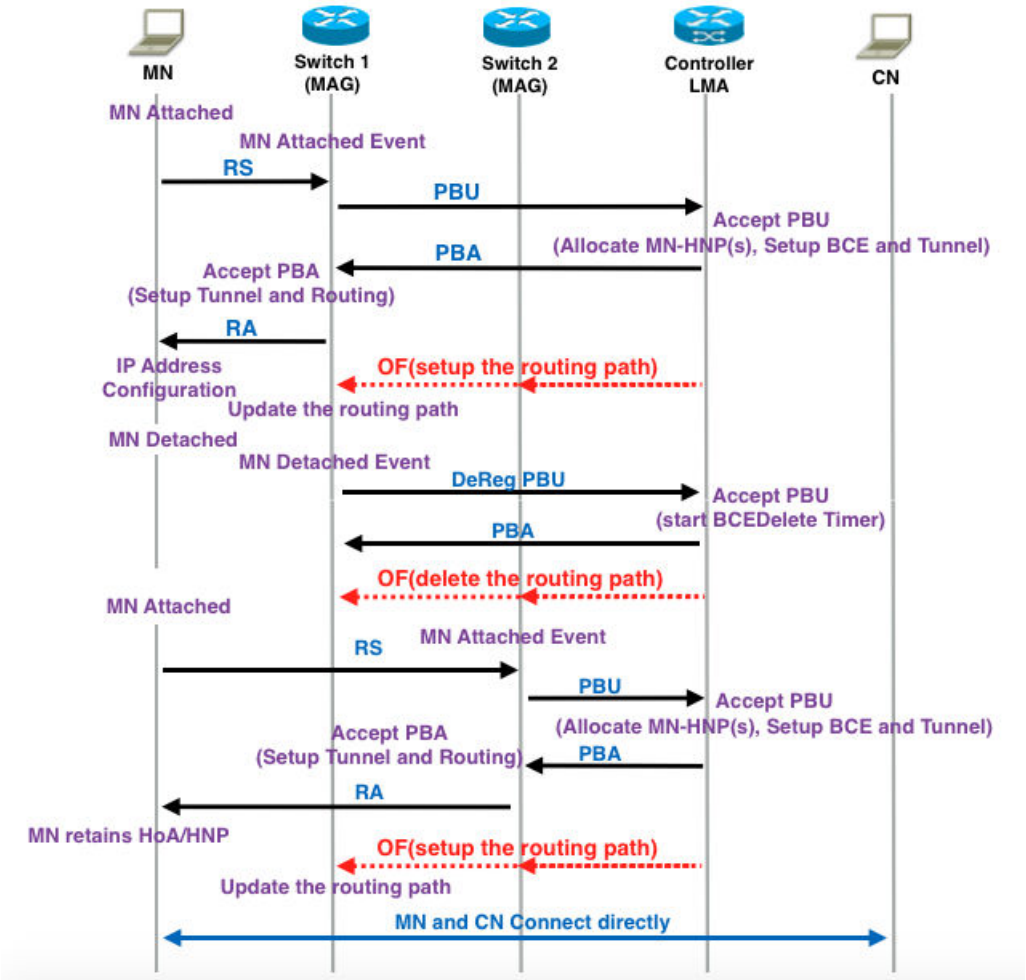

Figure 3.1 OPMIPv6 Signaling

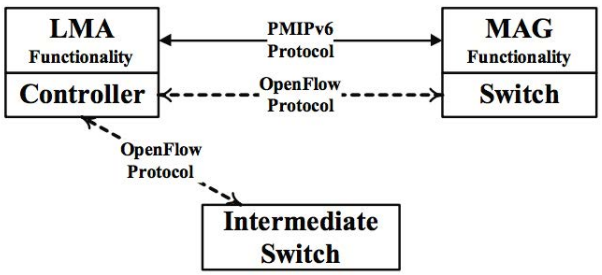

Figure 3.2 The Architecture of OpenFlow-based PMIPv6 [87]

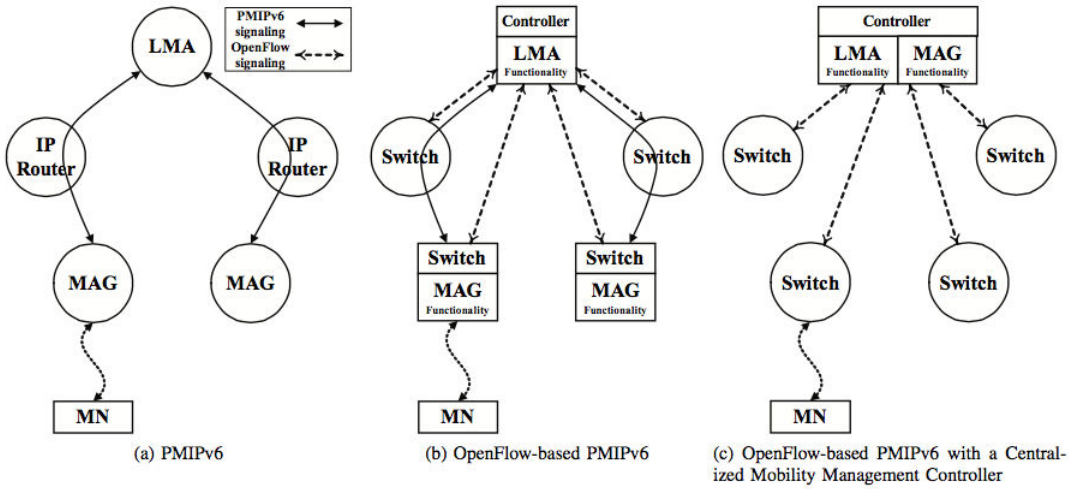

Figure 3.3 Control Plane Configuration of PMIPv6 and OpenFlow-based PMIPv6 [87]

- **Advantages and Limitations**

  We retain that OPMIPv6 method can mitigate the IP-in-IP tunnel overhead problem. Its performance regarding signaling cost and packet delivery cost is more efficient than PMIPv6. However, this method requires PMIPv6 implementation and must implement the LMA and MAG functionality at the controller and the switches.

3.1.2.2 *OpenFlow PMIPv6 (OF-PMIPv6)*

OF-PMIPv6 [88] introduced an AAA server in the architecture and it separated the SDN controller to the LMA that brought the controller closer to the mobile access gateways. Authors proposed a SDN switch for access gateway with an OpenFlow controller.

The functioning of the solution is illustrated in Figure 3.4 for the registration phase. For the handover process are illustrated in Figure 3.5 and Figure 3.6.

On mobile node arrival, the SDN access gateway (labeled OMAG: OpenFlow Mobile Access Gateway) contacts its OpenFlow controller (OF-Controller) and then the controller contacts the AAA server before, in case of success, to contact the LMA. Signaling is OpenFlow for mobile authentication, for setup at the gateway side the tunnel between next access gateway and LMA, and for informing the handover event to the controller. PMIPv6 protocol is the signaling for forwarding the assigned Home Network Prefix (HNP), for add/delete/update Binding Cache Entry, and for setup the tunnel at LMA side. The mobile node is authenticated

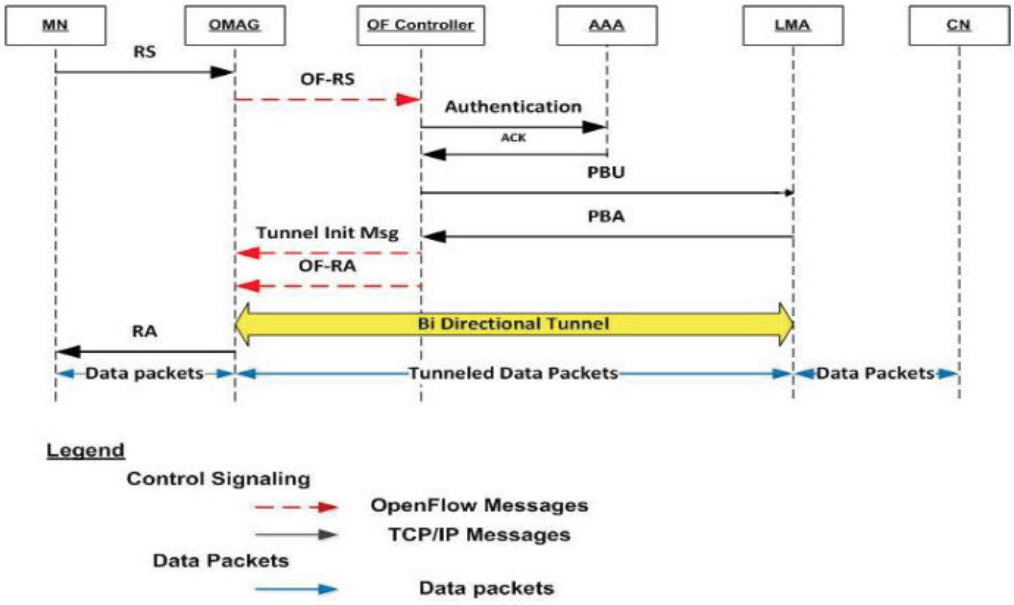

Figure 3.4 OF-PMIPv6 Registration[88]

at AAA server before the OpenFlow mobile access gateway advertises the network prefix (Home Network Prefix).

The handover procedure is considered in two modes, reactive and proactive. The predictive handover gained less latency than reactive handover. For the two cases of handover, the proposal provides better performance (authors indicate respectively, 45% and 91% decrease for reactive and proactive handover).

In reactive mode, the handover is almost similar to the handover of PMIPv6. But the mobility signaling messages (PBUs) are sent by the OpenFlow controller and also the controller re-authenticate the mobile by itself (does not need to re-authenticate mobile with AAA server). The OF-PMIPv6 reactive mode is illustrated in Figure 3.5.

In OF-PMIPv6 proactive mode, the OpenFlow controller has a handover decision function to determine the next access gateway. So the tunnel is established before the mobile is disconnected from the previous Mobile Access Gateway to new Mobile Access Gateway. Then, the packets will be buffered at LMA and will be sent after the mobile finished handover. The OF-PMIPv6 proactive mode is illustrated in Figure 3.6.

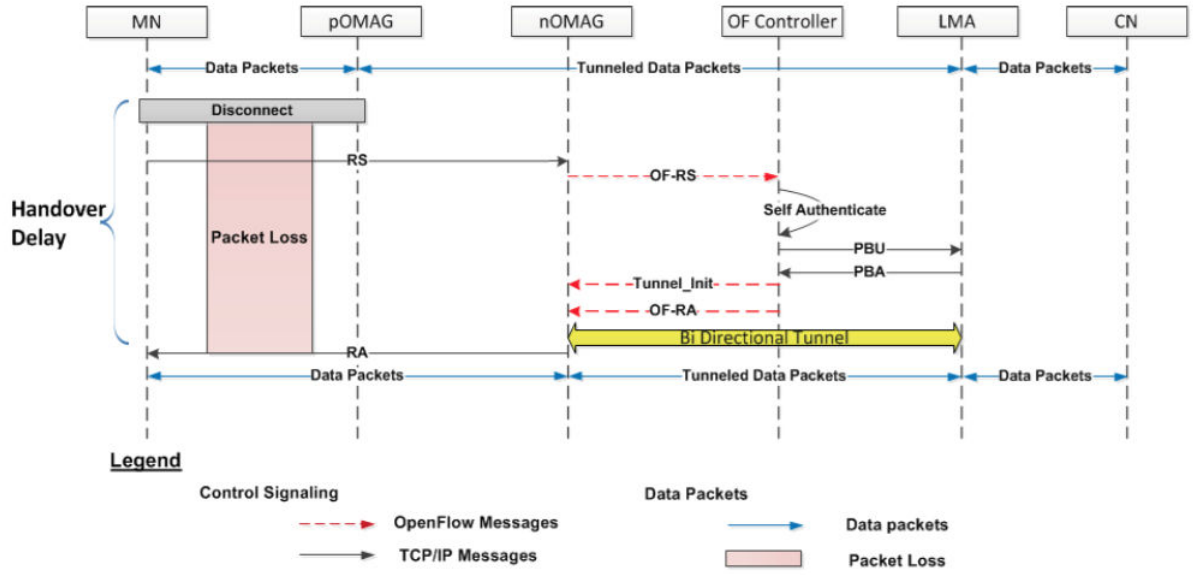

Figure 3.5 Reactive mode handover in OF-PMIPv6 [88]

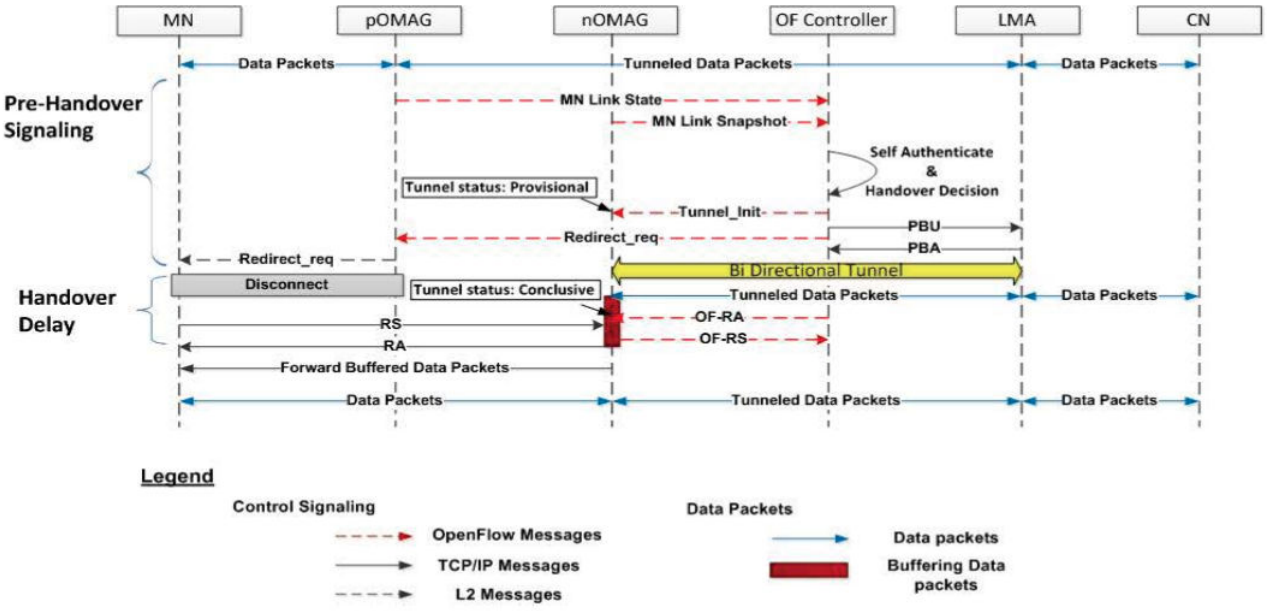

Figure 3.6 Proactive mode handover in OF-PMIPv6 [88]

TABLE 3.1 PMIPv6, OPMIPv6 and OF-PMIPv6 summarization

Approach	**PMIPv6 Operation**	**OpenFlow Operation**
PMIPv6	- notifying the attachment of MN - forwarding the HNP - add/delete/update BCE - setup the tunnel	-
OPMIPv6	-notifying the attachment of MN Switch (MAG)	Setting up the routing path - forwarding the HNP
OF-PMIPv6	- forwarding the HNP - add/delete/update BCE - setup the tunnel	Send the information for: - authentication of MN - setup tunnel at nOMAG - inform the handover to controller

TABLE 3.2 Number of Signaling Comparison in MN Registration period.

	PMIPv6	**OPMIPv6**	**OF-PMIPv6**
	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)
IP Address Configuration			
Router Solicitation (RS)	1	1	1
Router Advertisement (RA)	1	1	1
Binding Address and Tunneling			
PBU	1	1	1
PBA	1	1	1
Movement Detection			
OF_PACKET_IN	-	-	1
OF_FLOW_MOD	-	2	-
OF_PACKET_OUT	-	-	2
Tunnel Establishment			
IP-in-IP Tunnel	Yes	No	Yes
Total	**4**	**6**	**7**

- **Advantages and Limitations**

  In terms of the handover latency, the result of OF-PMIPv6 is better than PMIPv6. But this approach still suffers from tunneling overhead problem.

3.1.3 Analysis of SDN-PMIP Cooperation

We analyze the methods cooperation between SDN and PMIP of the literature by first identifying what is achieved by OpenFlow signaling and what is achieved by PMIP in TABLE 3.1, then we detail the signaling to evaluate the signaling cost and their specific interest.

We enumerate the signaling in the registration period and handover period as shown in TABLE 3.2 and TABLE 3.3. To get comparable results, we do not consider the AAA process.

TABLE 3.3 Number of Signaling Comparison in MN Handover period.

	PMIPv6	**OPMIPv6**	**OF-PMIPv6**	
			Reactive	**Proactive**
	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)
IP Address Configuration				
Router Solicitation (RS)	1	1	1	1
Router Advertisement (RA)	1	1	1	1
Binding Address and Tunneling				
PBU	1	2	2	2
PBA	1	2	2	2
Movement Detection				
OF_PACKET_IN	-	-	1	3
OF_FLOW_MOD	-	4	-	-
OF_PACKET_OUT	-	-	3	4
Tunnel Establishment				
IP-in-IP Tunnel	Yes	No	Yes	Yes
Total	**4**	**10**	**10**	**13**

It appears that the cost of OPMIPv6 and OF-PMIPv6 are higher than PMIPv6 because they use both signalings of PMIPv6 and OpenFlow, while the integrated solution generates less signaling.

The two proposals OPMIPv6 and OF-PMIPv6 aim to improve the efficiency of PMIPv6 in the different problems. The OPMIPv6 approach focuses on solving the IP-in-IP overhead by using OpenFlow. The OF-PMIPv6 approach focuses to decrease the handover latency and to mitigate the number of packet loss during MN handover. Each method has the different advantages and limitations that can be shown in TABLE 3.4.

From TABLE 3.4, it appears that their solutions can be improved at least by thinking the mobility management with a SDN point of view.

3.1.4 Stand-alone Architecture: SDMM without legacy mobility protocols

In this second part of the related works in SDN mobility management, we focus on works that consider the mobility management only through the SDN architecture. At the beginning of the thesis, quite no works were available on the subject that can be separated into cellular and WiFi context as follows.

TABLE 3.4 PMIPv6, OPMIPv6 and OF-PMIPv6 Advantages and Limitations

Approach	Advantages	Limitations and Requirement
PMIPv6	- network based mobility. - decreases the handover latency of MN in MIPv6.	- localized mobility management. - IP-in-IP Tunnel overhead. - packets loss during MN handover. - PMIPv6 implementation requirement.
OPMIPv6	- network based mobility. - avoids IP-in-IP tunnel establishment. - decreases signaling cost in integrated architecture (OPMIPv6-C).	- localized mobility management. - packets loss during MN handover. - increases signaling cost in OPMIPv6. - PMIPv6 implementation requirement. - OpenFlow implementation requirement.
OF-PMIPv6	- network based mobility. - decreases the long handover latency of MN problem. - decreases the number of loss during MN handover in OF-PMIPv6 proactive mode. - MN authentication with AAA server.	- localized mobility management. - IP-in-IP Tunnel overhead. - packets loss during MN handover in OF-PMIPv6 reactive mode. - PMIPv6 implementation requirement. - OpenFlow implementation requirement.

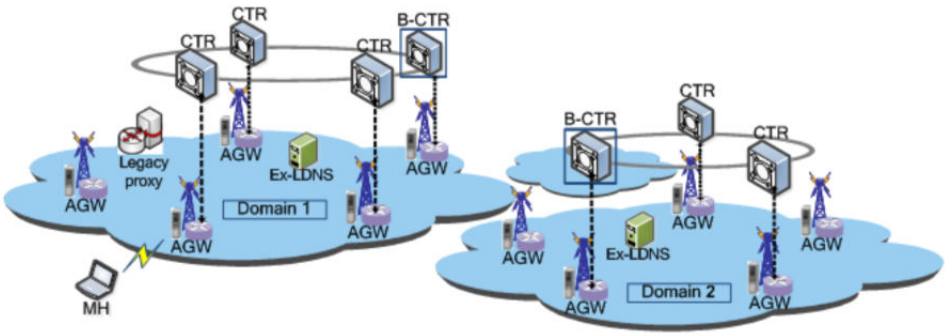

Figure 3.7 SaDMM Architecture [89]

- **Cellular Context**

3.1.4.1 *SDN-aided Distributed Mobility Management (SaDMM)*

In 2013, Y. Li et al proposed a new approach to realize the distributed mobility management in the cellular network using SDN instead of the existing mobility management protocols that called SDN-aided Distributed Mobility Management (SaDMM) [89].

The SaDMM architecture illustrated in Figure 3.7 covers many domains and proposes many controllers (CTR) associated to access gateways (AGW) and border controller (B CTR) to manage interdomain movement. SaDMM also defined the Mobile Host identifier (MH-ID) to replace the source and destination addresses to send and receive packets. To maintain MH's connectivity during MHs handover, the CTRs are

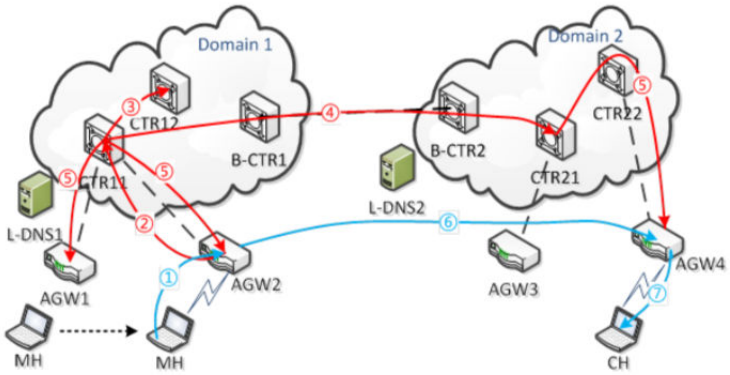

Figure 3.8 Handover and location update [89]

responsible for detecting the mobility of MHs and updating the routing table in the anchor points directly. The handover operation is illustrated in Figure 3.8. Multiple controllers are used for supporting distributed mobility management. They maintain the mobility related information and re-configure the routing tables in access gateways during handover.

The mobiles in SaDMM can connect to the traditional mobiles by the responsibility of legacy proxy server and extended DNS server. The extended DNS server adds the records of the mapping information between the Fully Quality Domain Name (FQDN) and each MH-ID.

– **Advantages and Limitations**

  SaDMM uses the benefits of SDN to detect and update the location of mobiles, and also update routing in access gateway directly without messages exchange. Thus SaDMM reduces the handover delay. It also designs to support the MH authorization with AAA server. However, because of its distributed design, the paging and connection establishment functions are complex [89]. It also needs to implement all network devices to support MH-ID because of the IP addresses of source and destination were replaced.

- **WiFi Context**

3.1.4.2 *SDN-based*

You Wang and Jun Bi [90] [91] [92] proposed SDN-architecture to enhance Mobile IP network using OpenFlow. The SDN-based approach is a network-based mobility management. It uses OpenFlow messages for managing the routing path of all packets

by mapping the HoA of the mobile to its CoA. When mobile attaches a switch, this switch assigns to it a CoA and sends a Binding Update message to the controller. In this approach, CoA is the IP address of the mobile's first-hop switch. Then, the controller knows the CoA of the mobile and adds it in Binding Cache. When the correspondent node connects to mobile, the mobile's first-hop switch downloads Binding Cache and rewrites the original destination addresses (containing HoA) in each packet with the CoA, and then it forwards the packet to the mobile node.

– **Advantages and Limitations**

  Even though SDN-based provides mobility management with SDN concept, it rewrites all mobile's packets on the network side. This can imply that the switches and the controllers suffer heavy loads and increasing complexity as a large number of mobiles.

3.1.4.3 *SDN-based DMM*

In 2014, SDN-based DMM [83] approach was proposed to apply SDN for distributed mobility management in 5G networks. Two main components were proposed in this approach. One is Network Controller (NC), and other is a Distributed Mobility Management Gateway (DMM-GW). Both NC and DMM-GWs support OpenFlow protocols. The NC is responsible for configuration of the forwarding rules on access routers (DMM-GWs) with OpenFlow messages. The DMM-GW plays the role of anchor.

The SDN-based DMM architecture and operations are illustrated in Figure 3.9. The OpenFlow messages are used to configure the forwarding rules in both registration and handover procedure. After MN handover, the MN packet header will be translated to new IPv6 address at the previous DMM-GW and will be translated to old IP address at the new DMM-GW for providing MN mobility.

– **Advantages and Limitations**

  SDN-based DMM approach uses IP translation that rewrites IP header from an old IP address to new IP address and vice versa. This can avoid the tunnel establishment, leading to eliminating the overhead problem. However, the header translation process may make the DMM-GWs suffer heavy load. Moreover, the delay may increase in case of MN moved far away from the first DMM-GW, because the MN packets are always routed to the first DMM-GW and redirected to new DMM-GW.

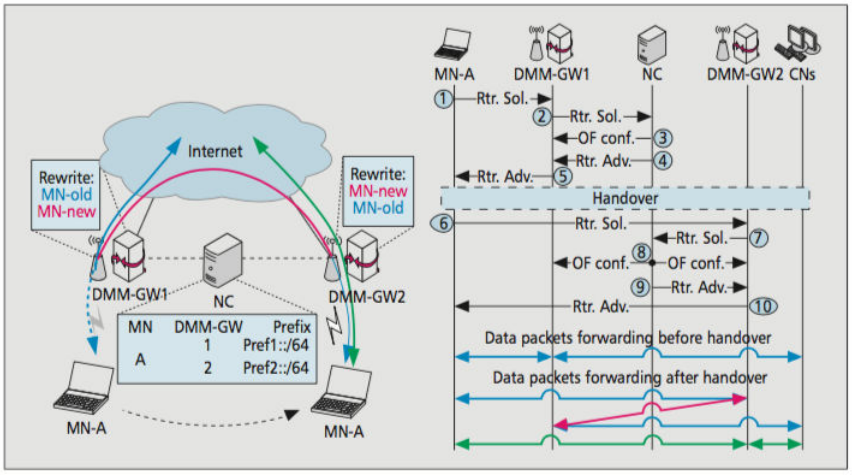

Figure 3.9 SDN-based DMM: architecture and operations [83].

3.1.4.4 *Distributed Mobility Management solution based on SDN architecture (S-DMM)*

More recently, in 2016, T. Nguyen et al proposed a Distributed Mobility Management (DMM) solution based on SDN architecture called S-DMM [93]. It implemented DMM as a service on top of a controller.

S-DMM helps to reduce the complexity of the tunnel and flow management mechanisms in the conventional DMM solution. It also mitigates the tunneling overhead and sub-optimal routing.

S-DMM has proposed two possible handover operations: ST-DMM mode and SO-DMM mode. ST-DMM still costs the tunneling overhead during MN handover. In contrast, SO-DMM can avoid the tunnel establishment by route optimization. However, SO-DMM mode is not implemented, and only ST-DMM is considered by evaluating in experimental.

– **Advantages and Limitations**

  The proposed scheme is a centralized model that is advantageously compared to a fully distributed scheme and a partially distributed scheme. But ST-DMM still has the overhead cost by tunnel establishment. While this work focuses on DMM problematic, when we focus on PMIP, this work is on the same guideline as those we present in this document.

TABLE 3.5 Number of Signaling Comparison in MN Registration period.

	SDN-based	**SDN-based DMM**	**S-DMM**
	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)
IP Address Configuration			
Router Solicitation (RS)	1	2	1
Router Advertisement (RA)	1	2	1
Movement Detection			
OF_PACKET_IN	2	-	2
OF_FLOW_MOD	2	2	2
OF_PACKET_OUT	1	-	2
Tunnel Establishment			
IP-in-IP Tunnel	No	No	No
Total	**7**	**6**	**8**

3.1.5 Analysis of Stand-alone Architecture

We compare the literature solutions in WiFi context by enumerating the signaling in both registration and handover period as shown in TABLE 3.5 and TABLE 3.6. For TABLE 3.6 we detail two S-DMM versions: tunnel (ST-DMM) and optimized (SO-DMM).

The results in TABLE 3.5 show that SDN-based DMM approach takes less signaling than SDN-based and S-DMM in MN registration period. Because SDN-based and S-DMM used the first packet of the flow to update the mobile flow tables on all OpenFlow switches (reactive flow insertion). Therefore, these process requires more signaling. While SDN-based DMM updates the mobile flow tables after mobile attached at the gateway (proactive flow insertion). Even if registration is not so frequent, it may be still improved. In Chapter 3, our proposal will improve this step.

Considering the handover period in TABLE 3.6, the SO-DMM approach costs higher signaling than ST-DMM, SDN-based, and SDN-based DMM. Because SO-DMM approach avoids to establish the tunnel in ST-DMM mode, so it needs to inform the mobile flow tables to OpenFlow switch in the correspondent network. The SDN-based approach takes less signaling. It rewrites the mobile packet's header. Thus, it requires only two OpenFlow messages to inform the mobile binding cache entries at new and old switches.

In conclusion, a centralized based solution has less signaling during the handover period but more signaling during the registration period.

TABLE 3.6 Number of Signaling Comparison in MN Handover period.

	SDN-based	**SDN-based DMM**	**S-DMM**	
			ST-DMM	**SO-DMM**
	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)	Quantity (Messages)
IP Address Configuration				
Router Solicitation (RS)	1	2	1	1
Router Advertisement (RA)	1	2	1	1
Movement Detection				
OF_PACKET_IN	1	-	1	1
OF_FLOW_MOD	2	2	2	3
OF_PACKET_OUT	-	-	1	1
Tunnel Establishment				
IP-in-IP Tunnel	No	No	Yes	No
Total	**5**	**6**	**6**	**7**

3.2 ICN and Mobility management

Mobility support in ICN architectures [94] [95] [96] can be considered in two categories based on their roles: receiver mobility and source mobility. Both mobilities have the different requirements and involve different mechanisms in case of handover.

3.2.1 Receiver Mobility

Providing that mobility is the mobile device changing its physical location after roaming, then host mobility is quite simple in ICN network. Because the concept of ICN uses a content name which is independent of the physical location. A content is named when it is created, no matter where or from who. A named content can be republished or replicated everywhere, but the name does not change. When mobile receiving a content moves to attach new network, it can re-express the interest packet to acquire the desired content from a new network without disrupted connectivity [97].

Even though ICN natively supports the receiver mobility, it still requires the approaches to improve efficiency in the real-time applications by such as SIP-based scheme [98], and pre-fetching and caching [99]. SIP-based scheme [98] reduces the new name prefix configuration delay by allocating the name prefix before the handover. Pre-fetching and caching [99] proposed a new Control Interest Packet for informing the current and new routers after a link layer trigger is detected. This solution minimizes some packet loss during handover and reduces the handover latency.

3.2.2 Source Mobility

One challenge issue in ICN architecture is a source mobility which is similar to mobility in IP networks but with different fundamental concepts. When the source moves, the content is moved, its content prefix is changed. This impacts the communication because the content cannot be accessed by hosts during source handover. This problem perhaps can be solved by the high benefits of ICN that are in-network caching, replication and multisource [100] [101]. The other packet loss problem occurs in a situation that the content is being forwarded from source to destination while the source is moving to another network. Then, the content cannot be forwarded to the right destination, it is similar to the Mobile IP problem. To address this problem, there is a need for additional mechanisms to update the name-based routing [102] [103] and also it requires that the routing protocol advertises the new content prefix.

3.3 Conclusion

In this chapter, we have explained two possible ways for Software-Defined Mobility Management (SDMM). The first way was proposed to integrate the legacy mobility signaling with the signaling of the new architecture (SDMM with legacy PMIPv6). The second way is stand-alone SDN mobility management. The approaches in these areas have been described and analyzed regarding signaling cost. The comparisons have shown that the SDMM with legacy PMIPv6 approaches have costed high signaling than the SDMM without legacy mobility protocols approaches.

From the literature analysis, we confirm our basic work [104] that a new approach may be defined that would be replaced PMIP by a centralized approach to decrease the overhead cost and to decrease the operation control. Next, we are going to propose a new SDMM approach which called SDN-Mobility.

Chapter 4

Proposal for Software-Defined Mobility Management (SDMM)

Contents

This chapter gives our first proposal for Software-Defined Mobility Management (SDMM) is called SDN-Mobility [104] approach. It is designed based on SDN concept to easily manage all network devices by using a centralized controller and programmability.

The first section presents the proposed SDN-Mobility proposal that explains SDN-Mobility architecture and operation. The second section gives the SDN-Mobility, and PMIPv6 comparison and obviously shows the SDN ability to provide mobility for MN by using the wired experimental simulation. The SDN-Mobility in WiFi networks evaluation will be presented in the next section.

4.1 SDN-Mobility Proposal

SDN-Mobility in this thesis means that the SDN network can provide mobility services based-on SDN concept. In fact, the mobility protocols may run in SDN network, but the operation of mobility protocols is not relevant to SDN components, it has been done individually. Thus, there are two ways to achieve this aim. The first way needs to modify the mobility protocol for co-operation with SDN signaling and the second way needs to find the new method based on SDN signaling to provide mobility service without the legacy mobility protocol. The second way is the one that is proposed in this thesis.

4.1.1 SDN-Mobility Architecture

The SDN Mobility architecture has two main functional entities, the controller and the Access Router (AR) as shown in Figure 4.1 in the red label.

4.1.1.1 Controller

An OpenFlow controller is located in the same network as ARs. Its duty is to be responsible for the flow table to all AR in SDN Mobility network.

4.1.1.2 Access Routers (ARs)

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

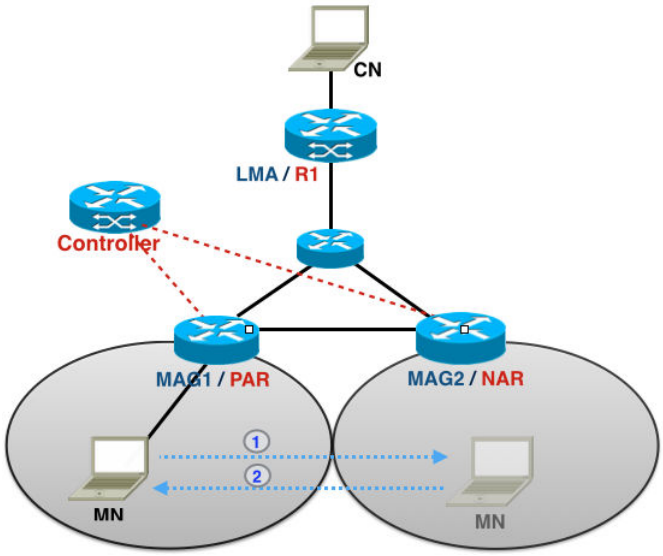

Figure 4.1 PMIPv6/SDN Mobility Architecture

4.1.2 SDN-Mobility Operation

For SDN-Mobility, we consider the situation where the MN moves from the current location to another in the local network, and the MN will receive the same IPv6 prefix home address like as PMIPv6 method.

In a general SDN network, when the MN moves to the new location and attaches to a new Access Router (NAR), the routing path in the Access Routers (ARs) is not updated. So, the packets flow continue to be forwarded to the Previous Access Router (PAR). This problem is being solved by modifying the function of the response of the MN status and using the SDN concept to update the controller with the MN status and the routing path of MN. We did this solution in this thesis. Our approach, SDN-Mobility uses OpenFlow protocol for communication between the controller and OpenFlow switches. The OpenFlow messages are sent to exchange the information between the controller and every ARs for mobility management in SDN network.

The SDN Mobility operation can be separated in two procedures: MN registration and MN handover. Both procedures transmit OpenFlow messages for notifying MN event and for updating the routing path. The SDN Mobility signaling illustrated in Fig. 4.2, looks similar as PMIPv6 one.

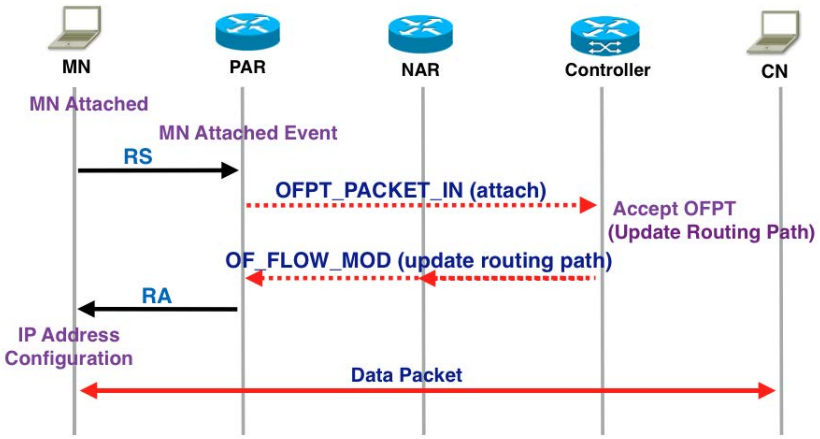

Figure 4.2 SDN Mobility Signaling

4.1.2.1 MN Registration

First, when the MN enters in the SDN Mobility network, PAR detects the attached event and sends the OFPT_PACKET_IN message to its controller for informing this event. After the controller received and processed that message, it sends the OF_FLOW_MOD message to all ARs for adding the routing flow of MN. Then, CN directly communicates with MN. Note that, we assume that the controller has the network policy allowing the MN to access network.

4.1.2.2 MN Handover

When the MN is detached from PAR, PAR sends OFPT_PACKET_IN message to its controller to inform it of this event. The controller immediately sends the OF_FLOW_MOD message to all ARs for deleting the routing flow of MN. At the same time, the MN moves to the new location, attaches to the NAR, NAR informs of the attached event to its controller by sending OFPT_PACKET_IN message. After the controller received and processed that message, it sends the OF_FLOW_MOD message to all ARs for adding the routing flow of MN. Then, the connection of MN and CN continues.

4.2 Interest of the Proposal

The caching mechanism is proposed for handover management in the LTE cellular network. Controlled by the Serving Gateway (SGW) which has a central view. The data are stored on the source E-UTRAN Node B (eNB) when the Radio Link Control protocol (RLC) detects the losses of data. Then, the source eNB transfers the data to the target eNB that will store the data until SGW received signaling from the target eNB, indicating that user data plane has been switching to the target eNB.

Unlike this context, we have enrolled in a network without connection signaling, without data link level 2 able to numbered the frames and therefore without buffering mechanism defined for mobility management.

We are going to show that it is possible to simply achieve handover by the mechanism of WiFi/IP technology. We do not use complex LTE signaling either, but only SDN control.

SDN-Mobility and PMIPv6 are network-based mobility approaches that do not require the signaling from MN which is suitable to be implemented for localized mobility network. SDN-Mobility operates an IPv6 autoconfiguration same as PMIPv6 for obtaining an IPv6 address. It uses the OpenFlow message for immediately updating the routing path, leading to avoid IP-in-IP tunneling establishment. But PMIPv6 approach uses the PMIPv6 messages to setup IP-in-IP Tunnel for communication between MN and CN. They are very easy and smooth to be in the real network. However, both methods need to add extra functions at the routers or switches for the mobility management. The problem of both methods is the packet loss during the MN handover. The size of each message in PMIPv6 and SDN-Mobility is represented in TABLE 4.1. The comparison of PMIPv6 and SDN-Mobility can be illustrated in TABLE 4.2 and TABLE 4.3.

Considering the number of exchanged messages in PMIPv6 and SDN-Mobility as shown in TABLE 4.4, the signaling cost of SDN-Mobility is almost the same as PMIPv6 during MN registration and MN handover.

Although PMIPv6 and SDN-Mobility can be implemented to support mobility, PMIPv6 uses IP-in-IP tunneling technique which makes some overhead cost that depends on the number of transferred packets. Thus, if the data size is large, it will make more IP-in-IP overhead, leading to increasing the packet transmission time. SDN-Mobility can provide packet forwarding directly without tunneling establishment by immediately updating the routing flow as the MN changes the point of attachment.

Note that our solution outperforms also the literature solutions (see TABLE 3.5 and TABLE 3.6 in Chapter 3).

TABLE 4.1 PMIPv6 and SDN-Mobility Messages Summary

Message	**Size (Bytes)**
IPv6 Address Configuration	
Router Solicitation (RS)	16
Router Advertisement (RA)	56
PMIPv6	
Proxy Binding Update (PBU)	84
Proxy Binding Acknowledgement (PBA)	76
SDN-Mobility	
OFPT_PACKET_IN (attached/detached)	156
OF_FLOW_MOD (update routing path)	120

TABLE 4.2 PMIPv6 and SDN-Mobility summarization

Approach	**PMIPv6 Operation**	**OpenFlow Operation**
PMIPv6	-notifying the attachment of MN - forwarding the HNP - add/delete/update BCE - setup the tunnel	
SDN-Mobility		- notifying the attachment of MN - setting up the routing path

TABLE 4.3 PMIPv6 and SDN-Mobility Advantages and Limitations

Approach	**Advantages**	**Requirement and Limitations**
PMIPv6	- network based mobility - decreases the long handover latency of MN problem in MIPv6.	- localized mobility management. - IP-in-IP Tunneling overhead. - packets loss during MN handover. - PMIPv6 implementation requirement.
SDN-Mobility	- network based mobility - avoids IP-in-IP tunneling establishment. - decreases the handover latency of MN problem.	- localized mobility management. - packets loss during MN handover. - OpenFlow implementation requirement.

TABLE 4.4 The Number of Signaling Comparison in PMIPv6 and SDN-Mobility in MN Registration Period and Handover Period.

	MN Registration Period		**MN Handover Period**	
	PMIPv6 (Messages)	**SDN-Mobility** (Messages)	**PMIPv6** (Messages)	**SDN-Mobility** (Messages)
IP Address Configuration				
Router Solicitation (RS)	1	1	1	1
Router Advertisement (RA)	1	1	1	1
Binding Address and Tunnel				
PBU	1	-	1	-
PBA	1	-	1	-
Movement Detection				
OF_PACKET_IN	-	1	-	1
OF_FLOW_MOD	-	2	-	2
OF_PACKET_OUT	-	-	-	-
Tunneling Establishment				
IP-in-IP Tunnel	Yes	No	Yes	No
Total	**4**	**5**	**6**	**5**

4.2.1 Experimental Comparison: SDN-Mobility vs PMIPv6

From the previous chapter, we conclude that a new approach may be defined to replace PMIPv6. Also in the previous section in this chapter, we proposed the SDN-Mobility concept. Thus, an objective of this section is to justify that SDN can be used to provide the mobility of MN without the implementation of the legacy Mobility protocol. We set up an experimental network to compare the performance of SDN-Mobility and PMIPv6, in Figure 4.1. Labels represent usual notation in PMIPv6 and SDN for each component.

We use Mininet [105] to generate topology. The MN connects to switches by using the wire channel. We update the source code to enable the MN to attach/detach with the switch in Mininet. It acts like a MN movement by hard handover scheme.

We compiled the kernel and installed UMIP mobility patch [106] [107] for PMIPv6 and uses RYU [39] [40] controller for SDN network with a set of parameters as shown in TABLE 4.5. Iperf [108] tool generates UDP and TCP traffics and performs the performance measurement. We also use Wireshark [109] to capture TCP traffic and use TShark [110] to classify the data for performance analysis.

TABLE 4.5 Simulation Parameters of PMIPv6 and SDN-Mobility

Parameter	Setting	
	PMIPv6	**SDN-Mobility**
Simulation Tool	Mininet 2.1.0p2	Mininet 2.1.0p2
Mobility Patch	PMIPv6-v0.4.1	-
Bandwidth on edge	10Mbps	10Mbps
All Link delay	$0.5x10^{-6}$ s	$0.5x10^{-6}$ s
Controller	-	Ryu 3.18
OpenFlow Message	-	v1.3.0
Testing Tool	Iperf v2.0.5	Iperf v2.0.5
UDP Datagram	1450 Byte	1450 Byte

4.2.2 Performance Analysis

This section shows the experimental result and analysis. After we setup the experimental topology as shown in Figure 4.1 which is described in the previous section. We use the same scenario for testing PMIPv6 and SDN-Mobility. Two scenarios were proposed for measuring performance on the different transport protocols (UDP and TCP).

4.2.2.1 Scenario 1: UDP

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

- **UDP Throughput**

  We ran Iperf server and Iperf client at CN and MN. We generate UDP traffic from MN to CN for 50 seconds and report the result every 0.5 seconds. In this scenario, MN moves two times. Five seconds after the simulation start, MN moves to the other attachment and will move back to a home network at 20 seconds later. The result of UDP throughput of PMIPv6 and SDN-Mobility can be illustrated in Figure 4.3.

  Considering the result in Figure 4.3, in y-axis, the result shows that the UDP throughput of SDN-Mobility is higher than PMIPv6 about 1 Mbps that is caused by the tunneling overhead of PMIPv6. The UDP throughput of both methods significantly dropped when the MN changed the point of attachment to the other access router after second 5.0 and reached 0 Mbps in second 5.5 for both methods. Then, the UDP throughput increases when the MN already attaches again and obtains an IPv6 address. The UDP throughput of SDN-Mobility began to increase at second 6.0 and second 7.0 in PMIPv6.

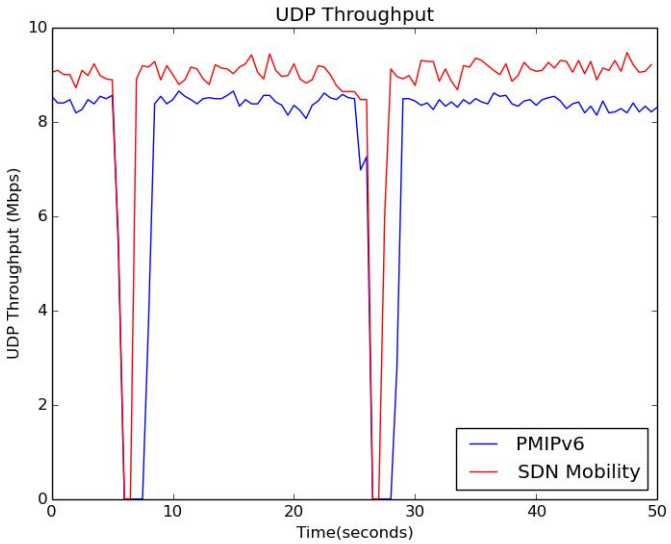

Figure 4.3 PMIPv6 and SDN Mobility UDP Throughput

This difference in times is due to a handover latency which is about 1.0 second for SDN-Mobility and about 2.0 seconds for PMIPv6.

- **Packet Loss**

In another experiment, we ran Iperf server and Iperf client at CN and MN. We generate UDP traffic from MN to CN for 50 seconds and report the result every 0.5 seconds. We did two sets of experiments: the first with one MN handover and the second with two MN handovers. For each set, we repeat simulation for 20 times and average the results. The number of packet loss can be shown in Figure 4.4.

For one MN handover experiment, MN moves only one time. MN moves to the other attachment 5 seconds after the simulation start. MN will move back to the previous attachment at 20 seconds later for two MN handovers experiment.

Considering the percentage of packet loss in Figure 4.4, the percentage of packet loss of PMIPv6 is 9.82% and 15.85% for one MN handover and two MN handovers. The packet loss is 3.85% and 7.0% for SDN-Mobility in one MN handover and two MN handovers experiment. As this result shows SDN-Mobility gives a lower percentage of packet loss compared to PMIPv6 which is about twice.

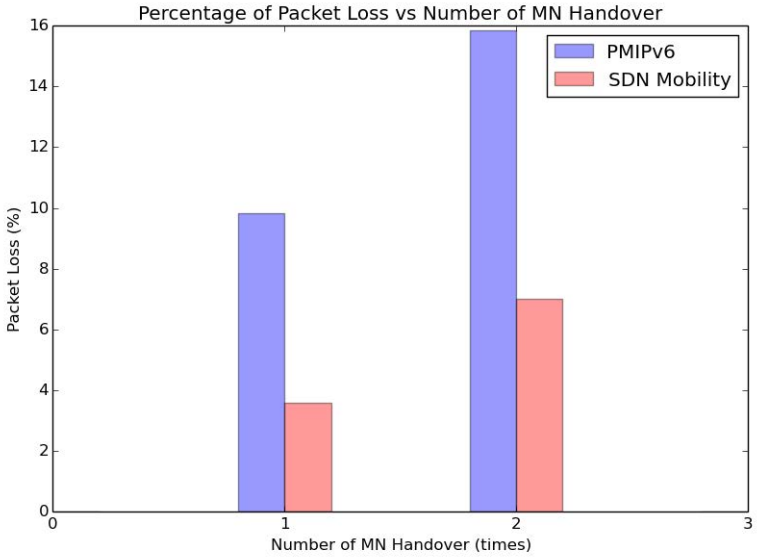

Figure 4.4 Percentage of Packet Loss versus Number of MN Handover

4.2.2.2 Scenario 2: TCP

In TCP scenario, we did an experiment quite similar to UDP throughput measurement experiment. We generated a TCP traffic by running Iperf between CN and MN for 50 seconds. During simulation times, MN moves to other attachment at second 5 and will move back to the previous attachment at second 25. Figure 4.5 shows TCP sequence of PMIPv6 and SDN-Mobility, from this we can learn that during handover time, TCP packets cannot be sent to the CN. So the TCP sequence number is held and will be counted after the MN connection restores. This result shows that SDN-Mobility took a shorter handover delay than PMIPv6.

4.2.3 Summary

The experimental results of the wired experiment have supported our proposal that SDN-Mobility can be implemented for Mobility Management like PMIPv6 without Mobility protocol implementation. Moreover, SDN-Mobility is better than PMIPv6 in two advantages: provides higher throughput and faster management.

Moreover, the wired experimental results precisely show the handover losses of both PMIPv6 and SDN-Mobility. For the real experiments, the mobiles will move to the other attachment with the wireless connection. Therefore to evaluate the impact of the mobility on the loss rate in the SDN wireless network, we are going to consider WiFi environment in the next section.

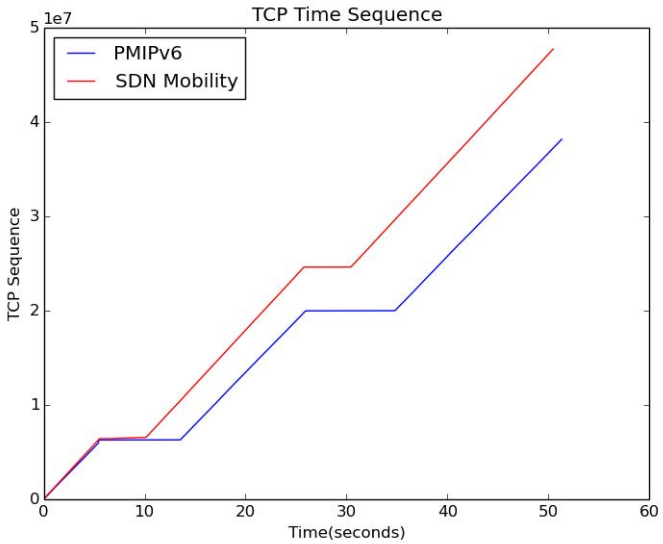

Figure 4.5 PMIPv6 and SDN-Mobility TCP Time Sequence

4.3 Performance Evaluation in WiFi Environment

We explained that SDN can support mobility management without mobility protocol, but it has some packets loss during MN handover. Now we focus on SDN-Mobility in WiFi networks and evaluate the number of losses. The first subsection gives the WiFi concept and IP configuration for our experiment. The second subsection presents the experimental network simulation. The experimental results are presented in the next section. The summary is shown in the final section.

4.3.1 WiFi Connection and IP Configuration

The experimental results in the previous section verified that SDN can provide mobility service to MN. We used Mininet to generate the network experiment of PMIPv6 and SDN-Mobility. However, the MN had hard handover because since the time we started this thesis the limitation of Mininet does not support modeling of wireless channel and mobility. On the other hand, a simple simulator like NS3 has limited support for software-defined controllers and does not fully implement the handover process.

Even now, Mininet supports wireless (Mininet-WiFi) [111] by using basic Linux TC tools to emulate the wireless channel with setting link parameters such as packet loss, delay, and

channel bandwidth. But Mininet-WiFi doesn't completely support a fundamental primitive in the wireless network such as wireless rate adaptation.

Thus, we proposed an experimental simulation where we have patched OpenNet [112] to link Mininet and NS3 to use both Mininet's advantage of controller compatibility and NS3's ability in the wireless/mobility modeling.

When the MN moves in the wireless network, first of all, layer 2 process operates. The WiFi AP sends periodically a beacon frame which contains all the information about the network such as timestamp, Beacon interval, capability information, and SSID. MN receives the beacon and chooses the nearest AP by using the scan function. After that, MN sends an association request to the AP for allocating the resources and synchronizing with a radio NIC. After receiving the association request, the AP considers associating with the NIC, and if accepted, it reserves memory space and establishes an association ID for the NIC. After that AP sends the association response frame to the MN. The layer 2 connection will be established when the MN receives this frame. The MN sends the RS message to request the home network prefix from the router. The IPv6 address of the MN is generated when the MN receives the network prefix from RA and do the autoconfiguration procedure.

In the roaming process of OpenNet, the MN decides to change the AP and channel after a beacon loss. The MN will scan an available AP to connect based on the Signal to Noise Ratio (SNR) by sending a probe request frame. The AP is chosen from candidate APs which has the highest SNR. The signaling of WiFi connection and IP configuration is illustrated in Figure 4.6.

4.3.2 The Experimental Network Emulation

We use OpenNet to generate a simple network topology which is illustrated in Figure 4.7. There are three main components for SDN-Mobility; Controller, OpenFlow Switch (OFSW) and OpenFlow Switch-Access Point (OFSW-AP).

We installed the controller to support OpenFlow 1.3.0 and uses RYU 3.18 API for the management of the OpenFlow messages. We add an IPv6 condition at the controller for the acceptation of IPv6 packets in the SDN network. When there is an IPv6 packet through the OFSW or OFSW-AP, it sends OpenFlow message to request the routing path from the controller. After receiving, the controller flush IPv6 rule to all OFSW in its SDN network.

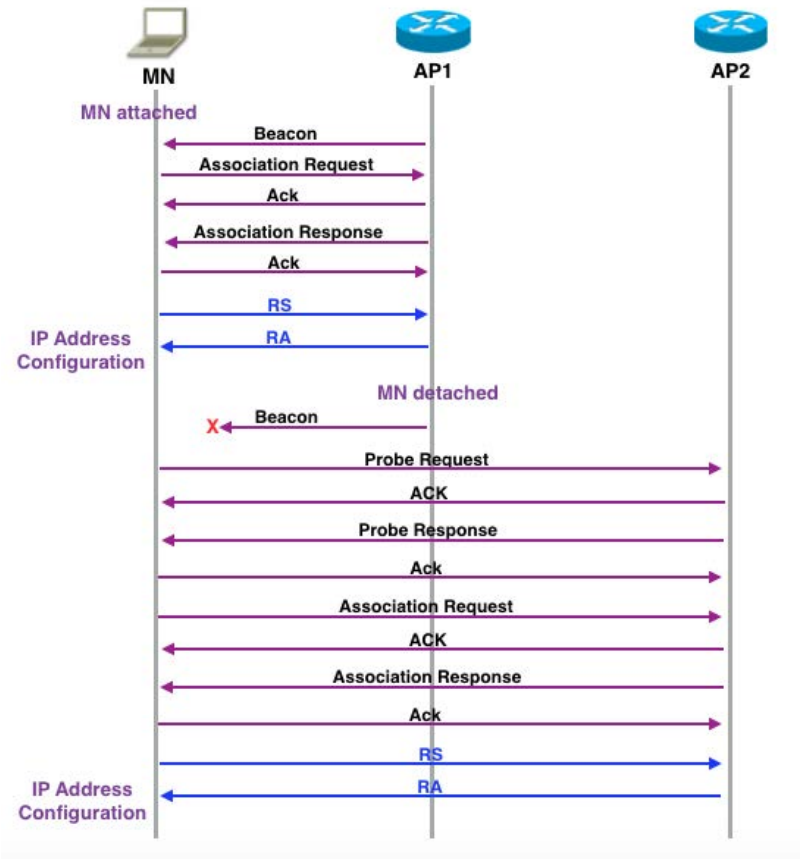

Figure 4.6 WiFi Connection and IP Configuration

The OpenFlow Switch (OFSW) was configured to support OpenFlow 1.3.0 and connects with the controller. For OpenFlow Switch – Access Point (OFSW-AP) was extended from OFSW and added NS3's ability in the wireless/mobility modeling. The OFSW-AP was generated to be an access point which was set to support the IEEE802.11g standard. The software installation and parameter configuration of each component are presented in TABLE 4.6.

The Corresponding Node (CN), the Mobile Node (MN) and the Router1 (R1) are based on Linux Ubuntu14.04.1 and were generated from the Mininet tool. The MN was configured to support mobility by using NS3 modeling. The R1 was configured to be Linux router by using Linux router command. It uses the router advertisement daemon for IPv6 (Radvd) to periodically advertise the home network prefix to the MN and uses the static routing table to route all packets to the right destination.

TABLE 4.6 Simulation Parameters

Device/Tool	**Software and Parameter Setting**
Emulation Tool	OpenNet (Mininet 2.2.1 + NS3)
Bandwidth on edge	100 Mbps
Controller	- RYU 3.18 - OpenFlow Message v1.3
OFSW	- Open vSwitch 2.3.1 - OpenFlow Message v1.3
OFSW-AP1	- Open vSwitch 2.3.1 - OpenFlow Message v1.3 - SSID = sdn - WiFi Channel 1 - ARF WiFi Management - IEEE802.11g
OFSW-AP2	- Open vSwitch 2.3.1 - OpenFlow Message v1.3 - SSID = sdn - WiFi Channel 6 - ARF WiFi Management - IEEE802.11g
R1	- Linux Ubuntu14.04.1 LTS - Linux Router - radvd v1.9.1
MN	- Linux Ubuntu14.04.1 LTS - Linux node (Mininet) - WiFi Station node (NS3) - Iperf v2.05 - VLC v2.16 - Firefox browser v57.0
CN	- Linux Ubuntu14.04.1 LTS - Linux node (Mininet) - CSMA node (NS3) - Iperf v2.05 - VLC v2.16 - Apache2 HTML5
Testing Tool	- Iperf v2.0.5 - VLC v2.1.6 - Firefox browser v57.0
UDP Datagram	1400 Byte

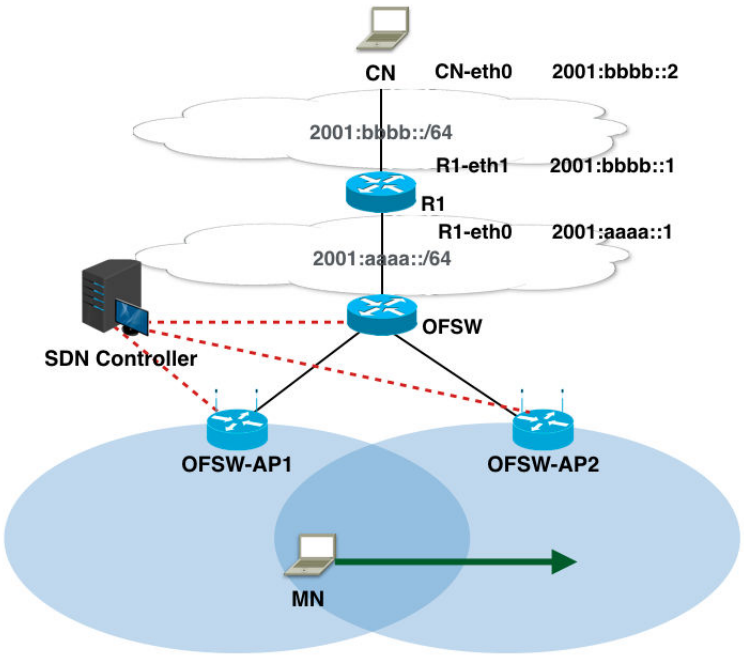

Figure 4.7 The Experimental Network Topology

4.3.3 Performance Analysis

To comprehend the characteristics of WiFi networks and the impact of loss rate, we did three different scenarios: UDP traffic on WiFi Fix Node, MN Roaming, and Dynamic Adaptive Streaming over HTTP (DASH). Each scenario was configured based-on a simple network topology as explained in the previous subsection.

4.3.3.1 Scenario 1 : WiFi Fix Node

The objective of the first experimental scenario is to learn the characteristics of the wireless network as the function of the distance between MN and OFSW-AP, and the data rate effects to the data throughput and the data packets loss.

We set up the experimental network topology, as shown in the Figure 4.7, and fixed the MN position of each experiment. The distance between MN and OFSW-AP is varying from 5 meters to 170 meters. We use Iperf to generate UDP traffic from MN to CN. The UDP datagram size is 1400 Bytes, the data rate is 0.5 Mbps, and the data packet size is 4 MByte. After Iperf reported the result, we vary the data rate and change the MN position. The UDP throughput and the number of packets loss can be illustrated in Figure 4.8 and Figure 4.9.

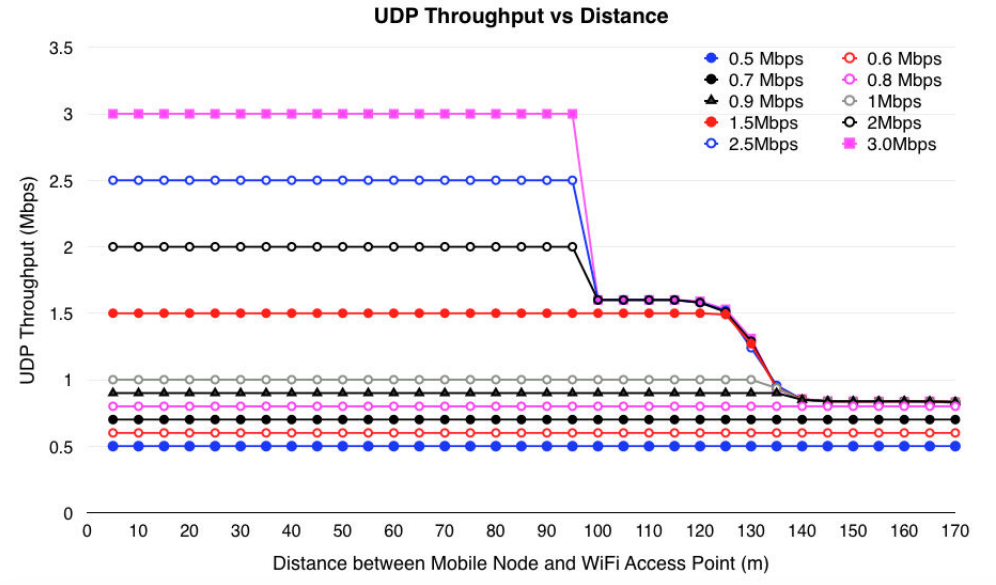

Figure 4.8 UDP Throughput vs Distance

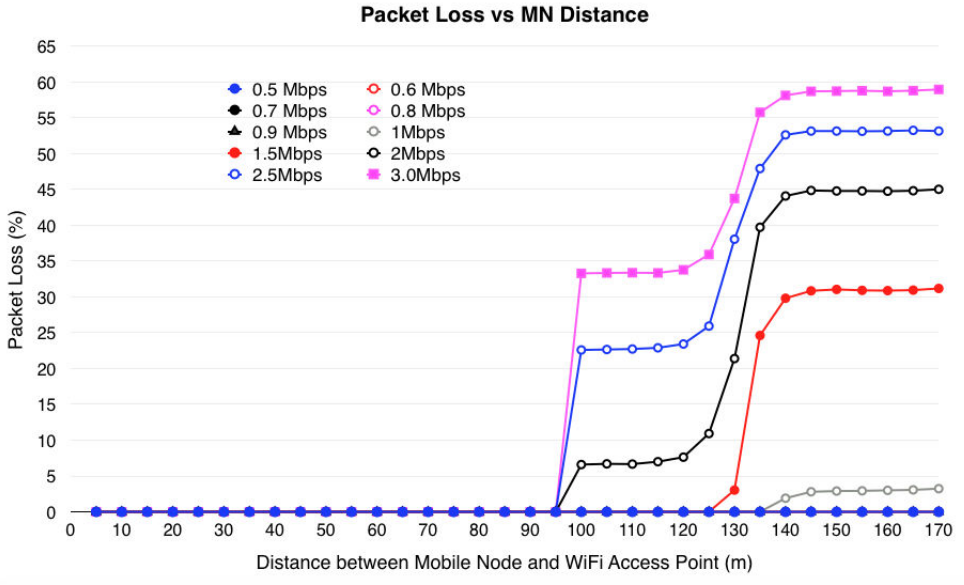

Figure 4.9 Packet Loss vs MN Distance

Considering the graph in Figure 4.8, UDP throughput decreases to 1.6 Mbps (at the application layer, excluding the overhead) as the distance raises above 95 meters. When the distance is more than 120 meters, this affects all traffics that have data rate more than 1 Mbps. For the traffic with a data rate of 1 Mbps and 0.9 Mbps, UDP throughput dropped when the distance reaches 135 meters. This result leads to the conclusion that when the MN is close to the OFSW-AP it will be able to send the obvious data at higher rates than when MN is farther.

The graph in Figure 4.9 illustrates the result of the packet loss as the function of the distance from the AP. It confirms the previous findings. The packet loss increases with the distance and with the data rate. These results are due to the WiFi rate adaptation.

4.3.3.2 Scenario 2 : MN Roaming

In the following, we focus on the packets loss during handover between two WIFI access points. We study the impact of the mobility during handover duration on the loss rate. In order to evaluate the number of lost packets while MN handover, we set up an MN roaming scenario (it is a situation where the MN is sending data to the CN while moving and changing its point of attachment as shown in Figure 4.7).

We use Iperf to generate UDP traffic by sending the data where the size is 4 MBytes, and the UDP datagram of each packet is set to 1400 Bytes. MN starts to transmit the data at the same position as 161 meters from OFSW-AP1. We vary the data rate from 0.5 Mbps to 1Mbps and vary the velocity of MN from 0 to 5.5 m/s. The obtained performance while varying the velocity of the Mobile Node can be illustrated in Figure 4.10.

Considering static MN (velocity= 0 m/s), the loss equals zero for all traffics that use data rate lower than and equal to 0.9 Mbps. For data rate equals 1 Mbps, the percentage of packet loss is about three percent, which is compliant with the result of the previous scenario. Because while MN lives in a weak signal area, OFSW-AP decreases the bandwidth capacity lower than the sending rate of the MN, leading to packets loss. We refer to WiFiAdapLoss in this case. Considering the data rate is lower than 0.9 Mbps, the percentage of loss is constant for all MN speeds. This loss referred to, handover loss, is due to MN handover. The handover loss is directly proportional to the transmitted data that can be explained by the following formula.

Handover Loss = Handover Delay x Data Rate

Considering the data rate is higher than or equal to 0.9 Mbps, the percentage of loss is not constant for all MN speeds. The number of packets loss will significantly increase when the velocity of the MN is less. The following formula can explain that.

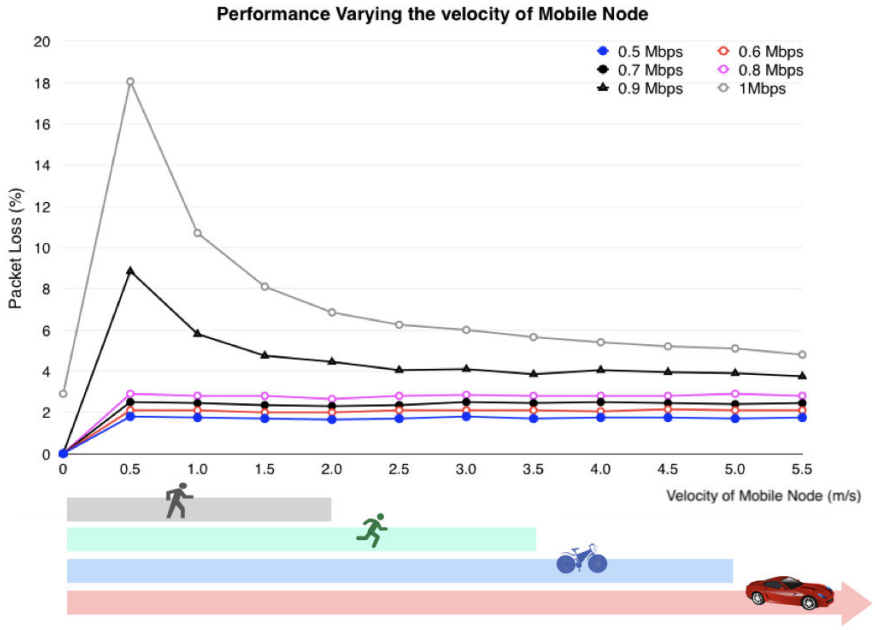

Figure 4.10 Impact of the velocity on the packet loss during handover.

$$\text{Total of Loss} = \text{Handover Loss} + \text{WiFiAdapLoss} + \text{BufDataLoss}$$

The handover loss is the number of losses during MN handover which comes from handover delay multiplied by the data rate. WiFiAdapLoss is the number of loss when the MN is in the weak signal area, and the WiFi management decreased the data rate of the MN as shown the experimental results in the previous scenario.

BufDataLoss is the number of packets loss which is buffered in the previous AP before the MN move to attach new the AP. Because while the MN is moving in the weak signal area, the AP decreases the data rate by buffering some packets in the AP. This implies that if the MN stays a long time in the weak signal area, more data packets will be buffered in the AP. Then, the MN moves to the other AP, all old buffered packets in the previous AP are lost, leading to the number of packets loss obtained for low-speed MN more than in case of faster MN. Because this latter spend a long time in a weak signal area. To overcome these loss issues, caching techniques can be used to buffer data for the handover duration. In the next chapter, we will explain our caching implementation. In the same direction used by CEOVDS (Cross Site Evaluation of an Open flow assisted Video on Demand Distribution) project [113], we used the software capacity of SDN to cache data during the handover.

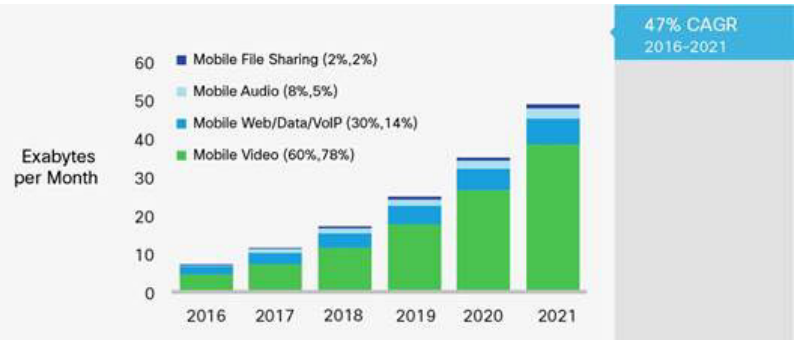

Figure 4.11 Mobile consumer Internet traffic in exabytes per month (Cisco VNI-2017) [114]

4.3.3.3 Scenario 3 : Dynamic Adaptive Streaming over HTTP (DASH)

The mobile consumer Internet traffic in exabytes per month [Cisco VNI-2017] [114] is illustrated in Figure 4.11. The mobile video will grow at a CAGR of 54 percent between 2016 and 2021, higher than the overall average mobile traffic CAGR of 47 percent. Many video providers such as YouTube and NetFlix are facing a challenge to keep a high quality of experience (QoE) for their subscribers. Dynamic Streaming over HTTP (DASH) approach for video streaming, also known as MPEG-DASH has been chosen for video streaming in both YouTube and NetFlix. MPEG-DASH is an adaptive bitrate streaming technique where the client automatically adapts the video bitrate by requesting the most suitable video segment from DASH server underlying the current network conditions and buffer occupancy.

From the above mentioned, leads us to study this scenario for two main purposes, the first is to investigate DASH approach for video streaming. The second purpose is to learn the impact of mobility on quality of experience (QoE).

We experimented by setting up a simple network topology as in Figure 4.7. The CN is a web server or DASH server, which is installed apache2 to support HTML5, and contains the Big Buck Bunny video files from ITEC-DASH project [115] that are encoded with twenty different bitrates in five resolutions and divided into smaller sized segments as shown in TABLE 4.7.

The DASH clients or mobile nodes will request the video via Firefox web browser which does not support MPEG-DASH natively. Thus, the website in web server needs JavaScript library (dash.all.debug.js) [116] to allow the mobile nodes to play DASH streaming. This Javascript will be downloaded to the mobile nodes side when the mobile nodes request the video. This script enables the DASH client to request the video bitrate depending on the current network parameters that the requested video bitrate will be degraded when the

TABLE 4.7 The Quality of Video

Resolution x Definition	Bitrate	Packets/Second	Video Quality
320 x 240	45kbps	3	Low
	88kbps	7	
	127kbps	10	
480 x 360	177 kbps	15	
	217kbps	18	
	253kbps	21	
	317kbps	27	
	369kbps	31	
854 x 480	503 kbps	43	Medium
	569kbps	48	
1280 x 720	771 kbps	66	
	987kbps	84	
	1.2Mbps	102	
	1.4Mbps	119	
1920 x 1080	2.1 Mbps	179	High
	2.4 Mbps	205	
	2.9 Mbps	248	
	3.2 Mbps	273	
	3.5 Mbps	299	
	3.8 Mbps	325	

available bandwidth of DASH client decreases and the buffer occupancy decreases, and vice versa.

To see the impact of mobility on QoE, we emulated two sub-experiments: fixed node and moving node. A mobile node requests the Big Bug Bunny video from DASH server via Firefox browser in both experiments. The position of a mobile node in a fixed node experiment is fixed at 100 meters from AP1. For moving node experiment, the velocity and position of the mobile node are set equal to 1.0 m/s and at 100 meters from AP1. The mobile node requests the video while moving from AP1 to AP2 and AP3.

Figure 4.12 shows (a) HTTP request, (b) requested video bitrate, and (c) buffer occupancy. By considering the HTTP request (Figure 4.12 (a)), both fixed node and moving node use HTTP GET message to request each suitable video segment that is relevant to the requested video bitrate as shown in Figure 4.12 (b). The overall requested video bitrate of a moving node is less than a fixed node and is frequency adapted. Since the higher video bitrate refers to higher video quality, then these results affect the moving node's QoE is worse than the fixed node's QoE.

The number of stalls and the number of quality switches are the important factors for the smoothness of the video playback. The mobile node expects to smooth playback video. From the results in Figure 4.13 has shown that the number of stalls and quality switches of moving

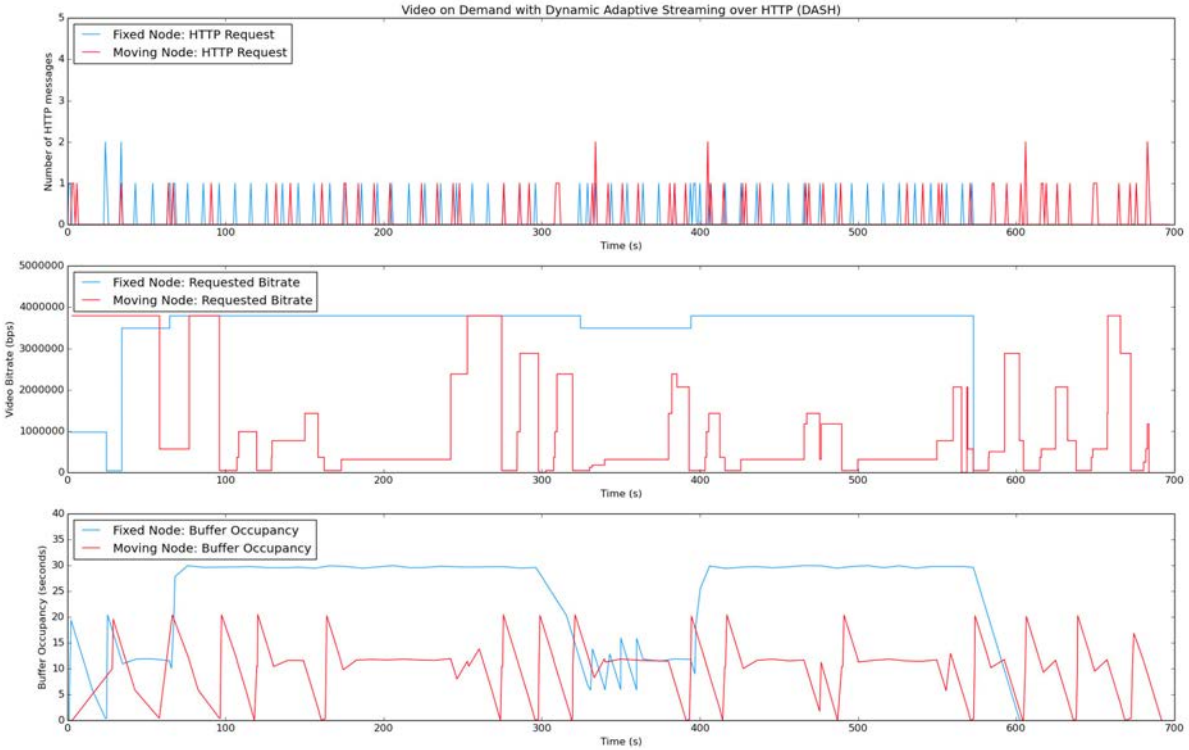

Figure 4.12 Video on Demand with Dynamic Adaptive Streaming over HTTP (DASH)

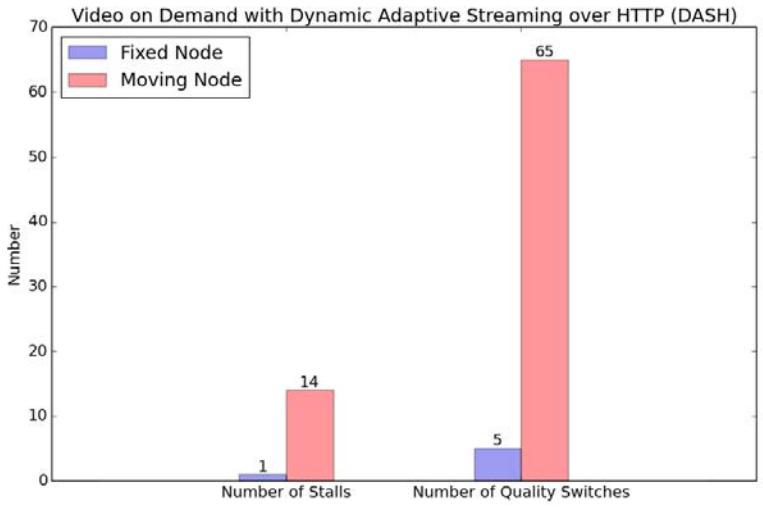

Figure 4.13 Number of Stalls and Number of Quality Switches

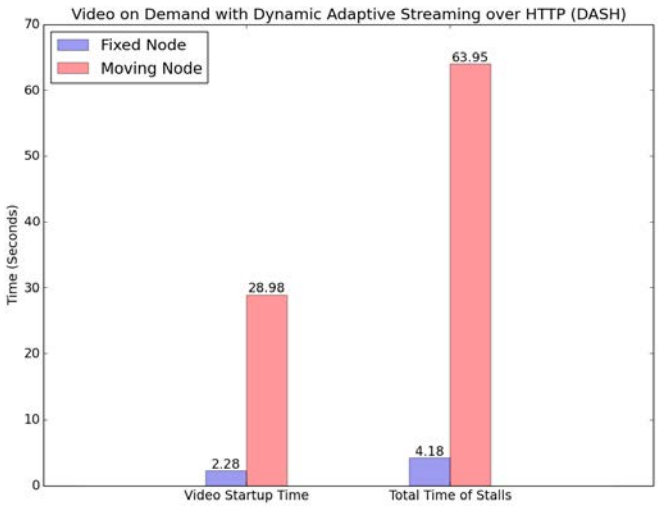

Figure 4.14 Video Startup Time and Total Time of Stalls

node are higher than fixed node about 13 times and 60 times. The fixed node's QoE is better than the moving node's QoE.

The video startup time is one factor which affects QoE. The DASH player will start playback after buffer occupancy reached the threshold value. Figure 4.14 shows the video startup time of both experiments, and the results show that a fixed node takes shorter the video startup time than a moving node around 26 seconds, leading to a fixed node's QoE better than a moving node's QoE.

These results indicated that the mobility impacts on QoE, even if the DASH client automatically adapts the video bitrate and buffers the video segments.

4.3.4 Summary

We emulated the SDN-Mobility on the wireless network by using OpenNet to connect Mininet and NS3. OpenNet uses Mininet for OpenFlow signaling procedure and uses NS3 for wireless/mobility procedure.

From the experimental results, it can be concluded that the loss is due to two causes. First, when the data rate is lower than the threshold (here 0.9 Mbps), the loss value depends on the rate of data transmission. The number of loss increases with the data rate. The second case, when the data rate is higher or equals the threshold, the loss value depends on two factors, the data rate of transmission and the velocity of the MN. The velocity of MN parameter refers to the amount of time that MN stays in a weak signal area. If the velocity of MN is fast, MN

will take a short time to stay in the weak signal area. But if the velocity of MN is slow, MN will take a long time.

Also, DASH approach has been tested in SDN-Mobility wireless network. The loss was not considered because DASH client requests each video segment via HTTP GET method and the requested video segments are transmitted to the client via TCP protocol. Thus, the video segments will be retransmitted in case of loss. However, the other affecting factors such as the requested video bitrate, buffer occupancy, number of stalls, number of quality switches, video startup time, and total time of stalls are considered to compare the QoE of a moving node and a fixed node. These results confirm that the mobility impacts on QoE, even if DASH adapts the video bitrate and buffers the video segment at the DASH client side.

4.4 Conclusion

We proposed an original network-managed mobility solution, in SDN in a stand-alone architecture, with centralized management. SDN-Mobility can provide mobility management without mobility protocol implementation with OpenFlow protocol. This approach makes it possible to reduce the signaling concerning the PMIPv6 standard solution, as well as solutions from the literature.

We have shown the feasibility of our approach by implementing it on a wired network with Mininet emulator, and then we have analyzed its behavior regarding the loss in a more realistic wireless environment.

The proposed solution is subject to different types of losses that may be detrimental to certain applications. Note that, the wireless environment we considered is a WiFi connection, nevertheless the behavior in an LTE (Long-Term Evolution) type cellular network should be similar as there would be losses due to handover, rate adaptation and buffering at the access points.

Our results of experimentation highlight the importance of movement speed on the losses and therefore the interest of contextualizing our study with representative application scenarios.

From the evaluation, we conclude the interest of two contexts of study that are an application with terrestrial displacement and an application with vehicular displacement.

Proposal evaluation indicated both PMIPv6 and SDN-Mobility still suffer a lot of packets loss in the MN handover period. Then, the question is whether this loss is a problem that must be solved. An answer element is provided by the literature, which indicates that for

basic Internet applications, the number of lost packets is a QoS parameter that is strongly correlated to the user experience in case of standard Internet applications [117]. We will seek to confirm this result in the next chapter.

Beyond the traditional applications, we have also been interested in new applications over HTTP as adaptive video (DASH) and have evaluated the QoS parameters specific to the buffering and rate adaptation. We find that again mobility introduces degradation.

In the next chapter, we explore the effect of QoS impairments on the user's perception of the service, the QoE, and propose to remedy this.

Chapter 5

Proposal Improvement with Caching Policy

Contents

In this chapter, we experiment the impact of the loss of data on a representative application of the new (compare to last decade) uses, the adaptive video, and propose to solve the problem by a new caching mechanism. We deal with the specification aspect: reactive or proactive, the architecture: centralized or decentralized as well as the implementation of the mechanism with cache event and policy aspects, before the handover by prediction or after the handover.

5.1 Need of Improvement

5.1.1 Loss Problem

In the previous chapter, we have shown that our approach, SDN-Mobility uses SDN concept to provide mobility management like as PMIPv6 without any software implementation at the mobile node. Moreover, SDN-Mobility gained more advantages than PMIPv6 regarding overhead and handover latency, leading to a reduced number of packet loss for SDN-Mobility compared to PMIPv6. However, both PMIPv6 and SDN-Mobility still suffer from packet loss during handover. This represents an issue for the future networks because the loss value increases with the data rates, leading to low performance.

For PMIPv6, we can improve the loss problem by using FMIPv6 which uses a caching mechanism for decreasing the number of packet loss, but the process of FMIPv6 needs some MN signaling while the MN attached/detached its point of attachment. It requires the mobility software at the MN, which is not flexible and not convenient to use in the real network. Also, FMIPv6 uses tunneling technique for forwarding cached contents, making more packet overhead.

With the leverage of SDN is centralized network management. SDN contributes to the creation of network automation that enables a policy-based decision. From these distinctive points of SDN, we propose a feasible solution with caching policy to decrease the number of packet loss and overhead without the MN software requirement. Caching policy cooperates with SDN controller for automatic buffering of the data during the handover. The cache operation will be explained in the next section.

Due to the velocity related to the mobility model. We defined the low velocity such as walking, or running to pedestrian mobility model as shown in Figure 4.10. For vehicular mobility model is a higher velocity such as biking, or transport vehicle.

5.1.2 Impact of Packet Loss on the Perceived Video Quality

Today, video streaming is one of the most popular services on the Internet. The increasing demand for video streaming has meant video constitutes a large portion of the total data traffic on the Internet [114]. Therefore, we chose the video streaming to see the impact of packet loss on the perceived video quality.

We experimented and generated two WiFi Mobile Nodes. One is a moving node (MN) and the other one is in a fixed position (FN). To compare the number of losses and the video quality of both MNs. We used a Big Buck Bunny [118] video to stream which was published under the Creative Common Attribution 3.0 license. It is allowed to freely reuse and distribute this content, also commercially, for as long as we provide a proper attribution.

We set the velocity of moving node (MN) equal to 1.0 m/s and set the position of FN at 100 meters from AP. Both MNs run VLC media player [119] for receiving the content from a streaming server. A streaming server streams the video to MN and FN by using a Transport Protocol for Real-time Transport Protocol (RTP) [120] over IPv6 unicast address [121]. The results are illustrated in the Figure 5.1. The number of the lost video frame for a moving node is higher than for fixed node, leading to a better video quality for fixed node compared to the moving node. This is because WiFiAdapLoss that we have explained in the previous chapter.

Considering the video scene of each mobile node, the scene of a fixed mobile node (FN) is delayed around 2 seconds compared with a server streaming. As for moving node (MN) have delayed around 10 seconds. Because when MN is moving far away from the AP, the latency time is increasing. However, in the real situation, the delayed scene is not effective for the user's perceiving. The user expects to receive the scene continuously and with good video quality. Because the video quality is related to the number of packet loss. This leads us to improve SDN-Mobility approach for decreasing the packet loss during mobility and MN handover.

Moreover, we have compared the video quality between the source (streaming server) and the receiver (mobile nodes) with Video Quality Measurement Tool (VQMT) [122]. We did three experiments to analyze the factors for packet loss. First, we generate one wireless node which is at the fixed position (FN). The second experiment, one moving node (MN) is generated. The last one, one fixed node and one mobile node. In all experiments, nodes receive video from a streaming server. A received video is recorded at each node and is compared frame by frame with the original quality by VQMT tool. We got the results in the different quality models such as Peak Signal to Noise Ratio (PSNR), Peak Signal-to-Noise Ratio taking into account Contrast Sensitivity Function (CSF) (PSNR-HVS), Peak Signal-to-Noise Ratio taking into account Contrast Sensitivity Function (CSF) and between-coefficient contrast masking of DCT basis functions (PSNR-HVS-M), Visual Information Fidelity pixel domain version (VIFP), Structural Similarity (SSIM), and Multi-Scale Structural Similarity (MS-SSIM) [123]. All results show the same trend. We chose the results of MS-SSIM metric to present because MS-SSIM model has performed equally well or better than the other quality models [124] [125]. The MS-SSIM results of each experiment are shown in Figure

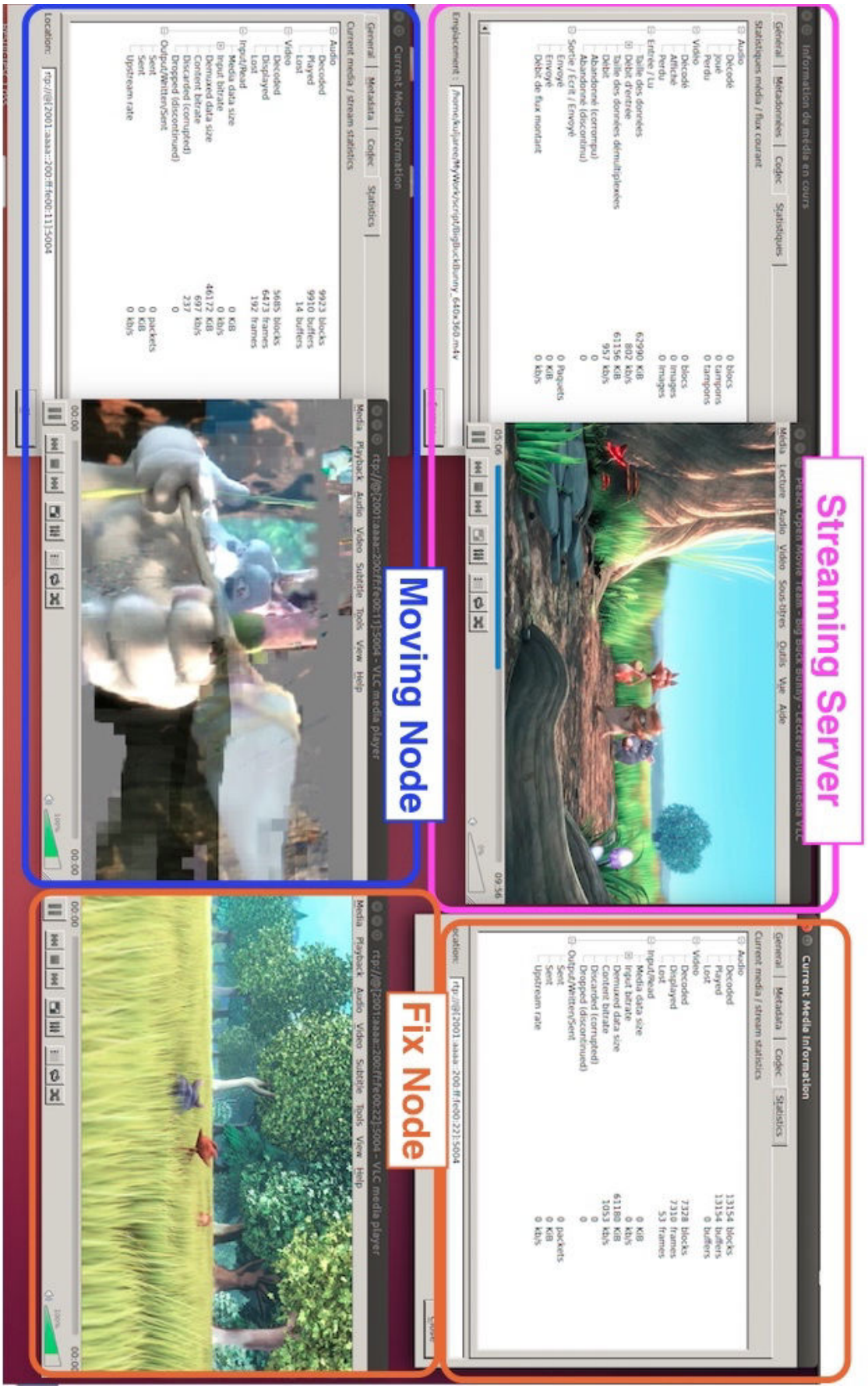

Figure 5.1 IPv6 Unicast Live Streaming

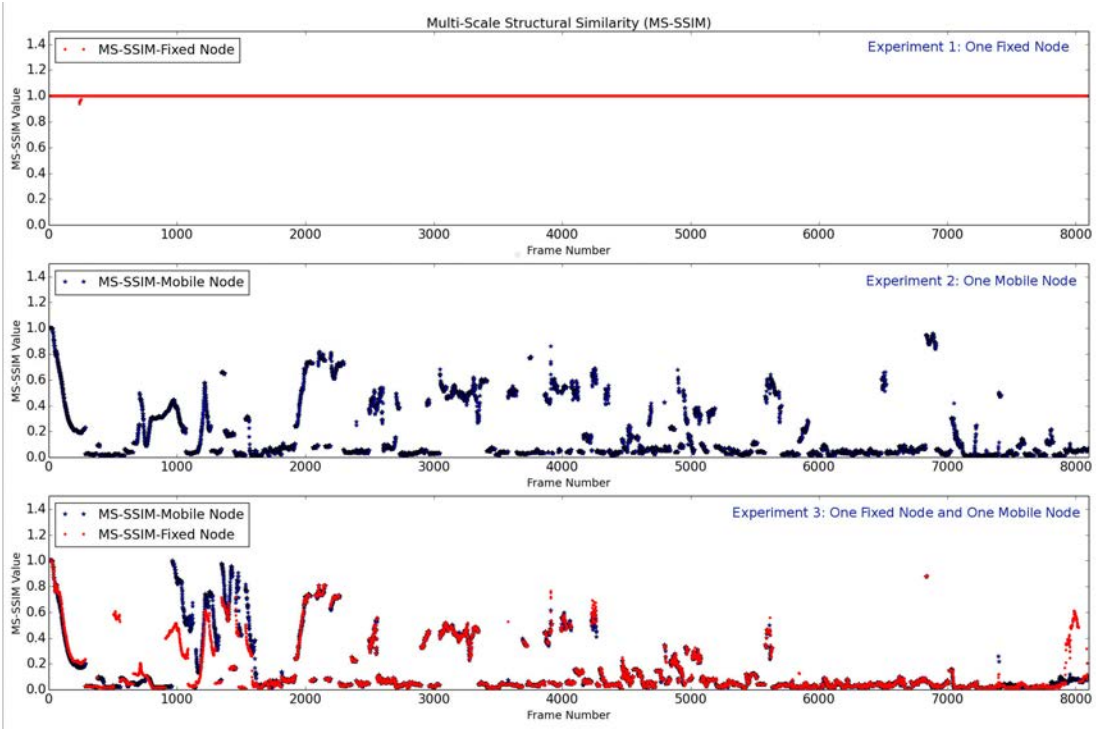

Figure 5.2 Multi-Scale Structural Similarity (MS-SSIM). a) Experiment 1, b) Experiment 2, c) Experiment 3

5.2. Note that, the interpretation of MS-SSIM results is 1 for equal frames and higher values are better.

Considering the result of the first experiment, the received video of a fixed node is quite the same as the original video. For the moving node, the quality of the received video is not good compared with the original video and the received video of a fixed node in the first experiment. From these results, make us know that the mobility of node is one factor which impacts to the quality of a received video. The results of the third experiment shown that the moving node effect to the quality of received video of a fixed node. Because the concept of WiFi transmission is a sharing of media that the Access Point will block the media for each node until the successful transfer of a data packet. So when moving node occupies the channel media, it will block the opportunity of a fixed node. Moreover, the latency time of moving node is large when it moves far away from the access point, this effects the quality of received video of a fixed node.

From the results of this section, we comprehended the MN handover loss and WiFi adaptation loss impact the quality of a received video. Considering these issues, we proposed to uses a caching scheme for decreasing the loss, refers to improving the quality of experience.

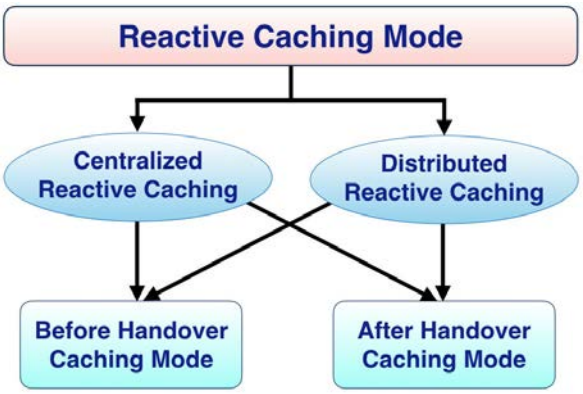

Figure 5.3 Reactive Caching Approach

5.2 Caching Scheme

Caching is a mechanism for storing data. It can be classified into two broad categories: proactive caching and reactive caching. The proactive caching approach is one caching method that guarantees no packet loss during MN handover; the data packets will be stored at all the time, whether the MN stays or moves from a network. Reactive Caching works only while the MN had the movement event that is an interesting point to improve by decreasing the number of packet loss and saving the storage. Thus, we adopt a reactive caching policy with an SDN approach for IP mobility management in the network. The additional caching operation is managed by the SDN controller based on the caching policy.

In this thesis, we propose a possible approach based on a reactive caching approach, a caching policy is a policy which caches/buffers the packets while MN had the movement event and handover, not all time. A reactive caching approach can be divided into two major architectures based upon where data is cached, and each one has two modes as shown in Figure 5.3.

We defined the switch events that can occur. There are three events: MN attached event, MN detached event, and predictive MN movement event. The MN attached and detached the event occurs when MN attaches/detaches in SDN Mobility network. These events are triggered at layer 2 using SYSLOG messages that are exchanged between the Access Point and OpenFlow switch likes as OAI-PMIPv6 [107]. The predictive MN movement event occurs while the MN's traffics are forwarded through OpenFlow AP-SW, and the data rate is decreased to 1 Mbps or lower according to the WiFi adaptation. This can imply that MN lives in the weak signal area where located near the border of the WiFi network. It has a higher probability of changing AP-SW.

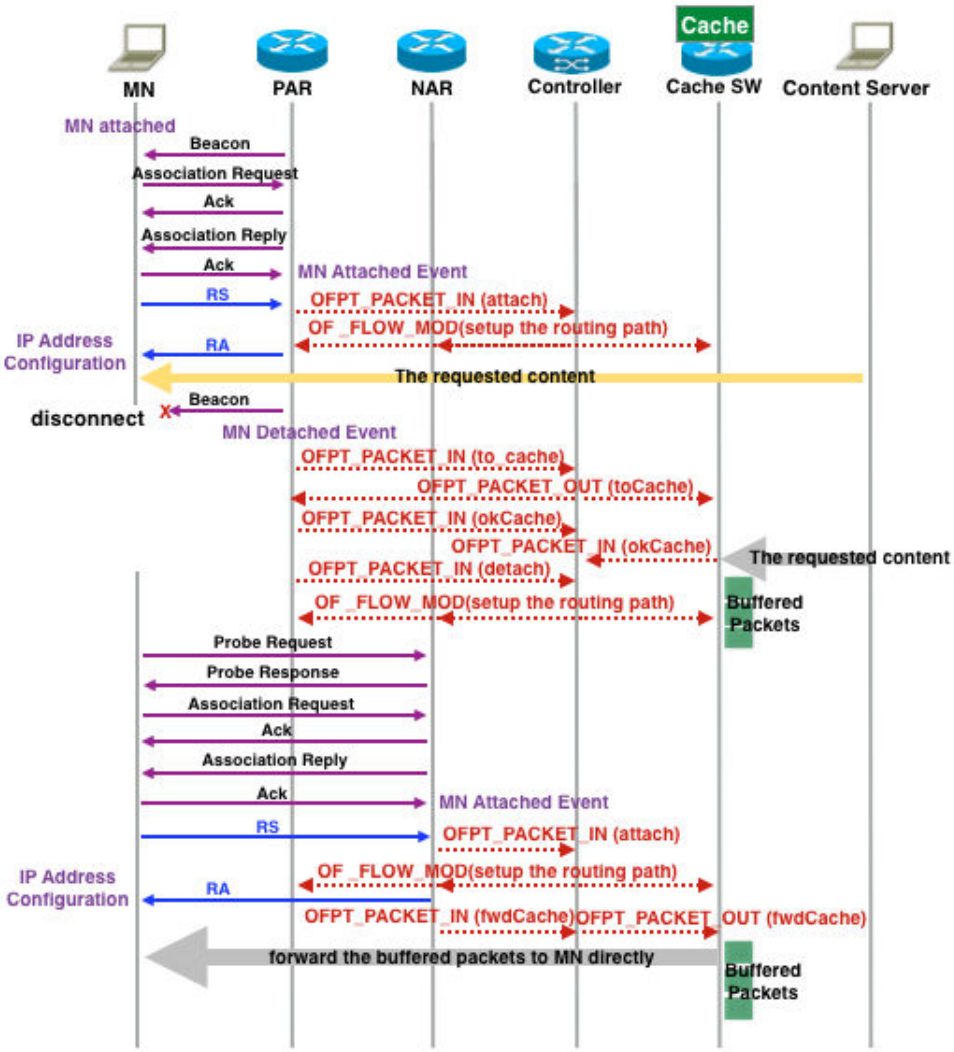

Figure 5.4 After Handover Caching Mode of Centralized Caching.

5.2.1 Centralized Reactive Caching

In centralized reactive caching architecture, the requested contents are stored at the cache switch (Cache SW) which is located in the SDN network. The Cache SW is selected by the controller, depending on the caching policy. There are two modes, operating in either after handover mode or before handover mode, depending upon the switch events.

5.2.1.1 After Handover Caching Mode

The after handover caching mode operates in the case is the Cache SW did not store the contents before MN handover. The signaling of this mode is illustrated in Figure 5.4. When MN disconnected from PAR, PAR informed the detached event to the controller and the routing path will be deleted. Then, when the controller knows that the MN has changed the point of attachment, it will check the cache status of the MN. Therefore the controller knows that the requested contents were not stored in the Cache SW before, it sends OFPT_PACKET_IN (toCache) immediately to PAR and Cache SW for starting the caching process. After that, the data contents will be stored at the Cache SW. While the MN moves to a new network, NAR handles the attachment event and sends the OFPT_PACKET_IN (attach) to its controller for informing the information. The controller sends OF_FLOW_MOD to all ARs in SDN network for updating the routing path of MN. After that NAR requests the cached contents by sending OFPT_PACKET_IN (fwdCache) to its controller. The controller receives the request and sends OFPT_PACKET_OUT (fwdCache) to Cache SW. Then, the Cache SW forwards the cached contents to MN directly.

5.2.1.2 Before Handover Caching Mode

Before handover caching mode requests and caches the content items before the occurrence of the handover process. This makes no packets loss in this mode. The signaling of this mode illustrated in Figure 5.5, will be started when PAR handled the predictive MN movement event. When the MN moves in the weak signal area, the AP decreases the data rate of MN to 1Mbps or lower from WiFi adaptative algorithm. This event is a trigger to cache. PAR will send OFPT_PACKET_IN (toCache) message to the controller for inquiring to cache policy. If the information of MN matched with the caching policy in the controller. The controller will send OFPT_PACKET_OUT (toCache) to PAR and Cache SW. Then, PAR received OFPT_PACKET_OUT (toCache), it caches and forwards the requested data to Cache SW. The PAR and NAR also send the OFPT_PACKET_IN (okCache) to its controller for cache confirmation. These data contents will be stored at Cache SW.

During the handover period, PAR handled the detachment event of MN and sends OFPT_ PACKET_IN (MN-detached) to the controller. The controller sends the OF _FLOW_MOD for deleting the old routing path of MN. After that NAR handles the MN attachment event, NAR sends OFPT_PACKET_IN (attach) to the controller for updating routing path and also sends immediately OFPT_PACKET_IN (fwdCache) for requesting the cached content items. Then, the controller received the request and sends OFPT_PACKET_OUT (fwdCache) to Cache SW for forwarding the requested contents in its storage to the MN directly.

Note that, in the worse case, MN have disconnected from PAR but cache trigger is not enabled, PAR automatically performs the after handover caching mode.

5.2.2 Distributed Reactive Caching

The functions of the distributed reactive caching are quite similar to those of the centralized reactive caching, but the location to cache the packet's data is different. The caching place of distributed reactive caching is PAR that can operate in two modes: either after handover mode or before handover mode, depending on when the controller detects the detachment event of MN.

5.2.2.1 After Handover Caching Mode

The after handover caching mode works in the case when contents of MN were not stored before MN has handover, leading to some packets loss during handover. When the MN detaches from PAR, the PAR senses this event and sends OpenFlow messages to its controller for caching and deleting the routing flow of MN. The requested data is stored at PAR and PAR sends the OFPT_PACKET_IN(okCache) to its controller as illustrated in Figure 5.6.

While MN attaches at the new network, NAR sends the OFPT_PACKET_IN (fwdCache) to its controller immediately. If the MN-ID matches in the caching policy, the contents of the MN are available. Then, the controller sends OFPT_PACKET_OUT(fwdCache) to PAR for forwarding the stored contents to the MN directly.

5.2.2.2 Before Handover Caching Mode

Before handover caching mode operates to cache the contents before the MN is disconnected from PAR, there is no packet loss in this mode. When the MN moves in the weak signal area, the AP decreases the data rate of MN to 1Mbps or lower from WiFi adaptative algorithm. This event is a trigger to cache. When this event occurs, PAR sends the requested message OFPT_PACKET_IN (toCache) to its controller. The controller verifies the MN information, if a match in the caching policy, the controller will send OFPT_PACKET_OUT (toCache) to PAR for storing the packet's data at PAR. PAR receives this messages, caches the data and also sends the OFPT_PACKET_IN (okCache) to the controller for confirming that PAR has already cached them.

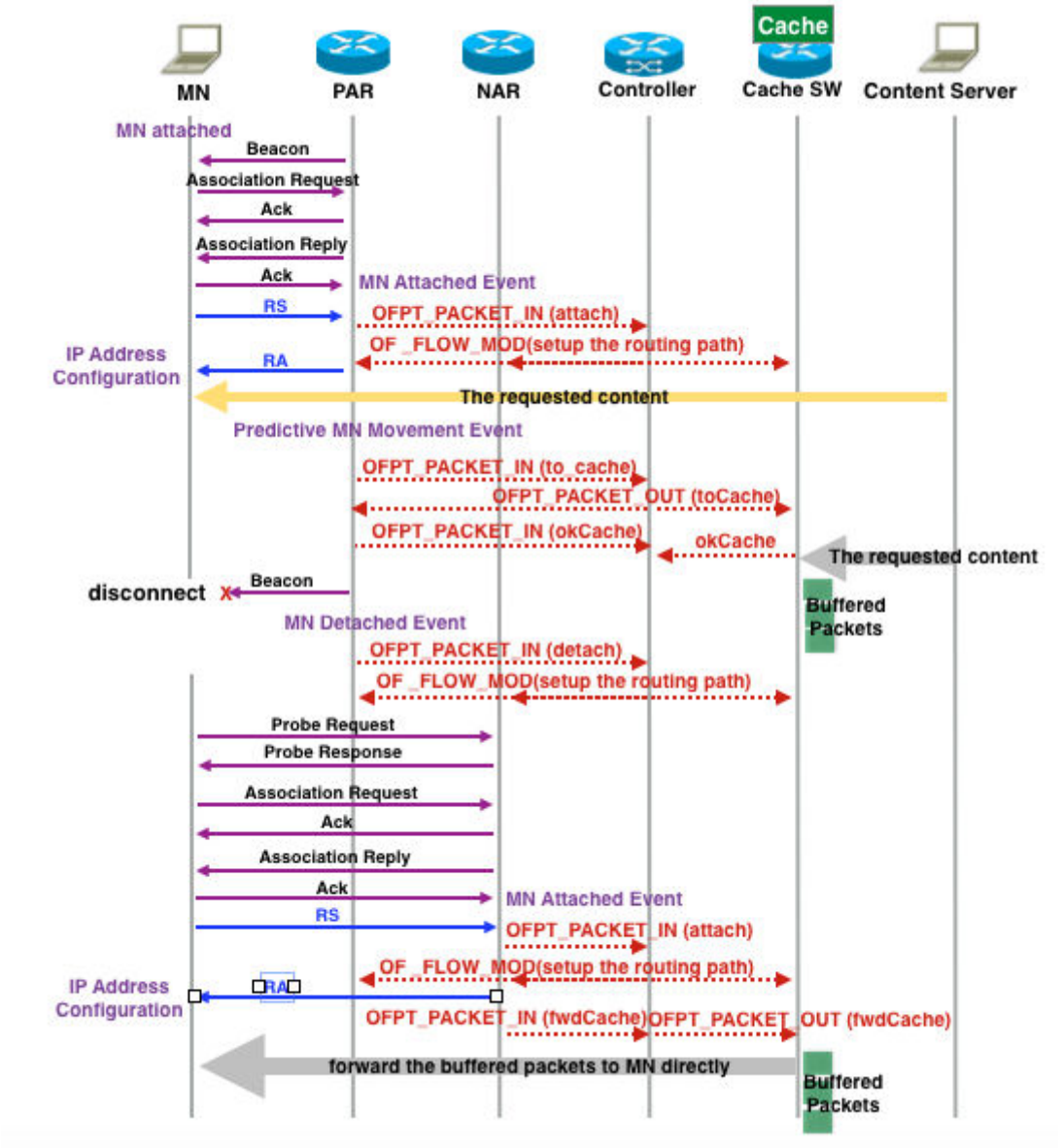

Figure 5.5 Before Handover Caching Mode of Centralized Caching.

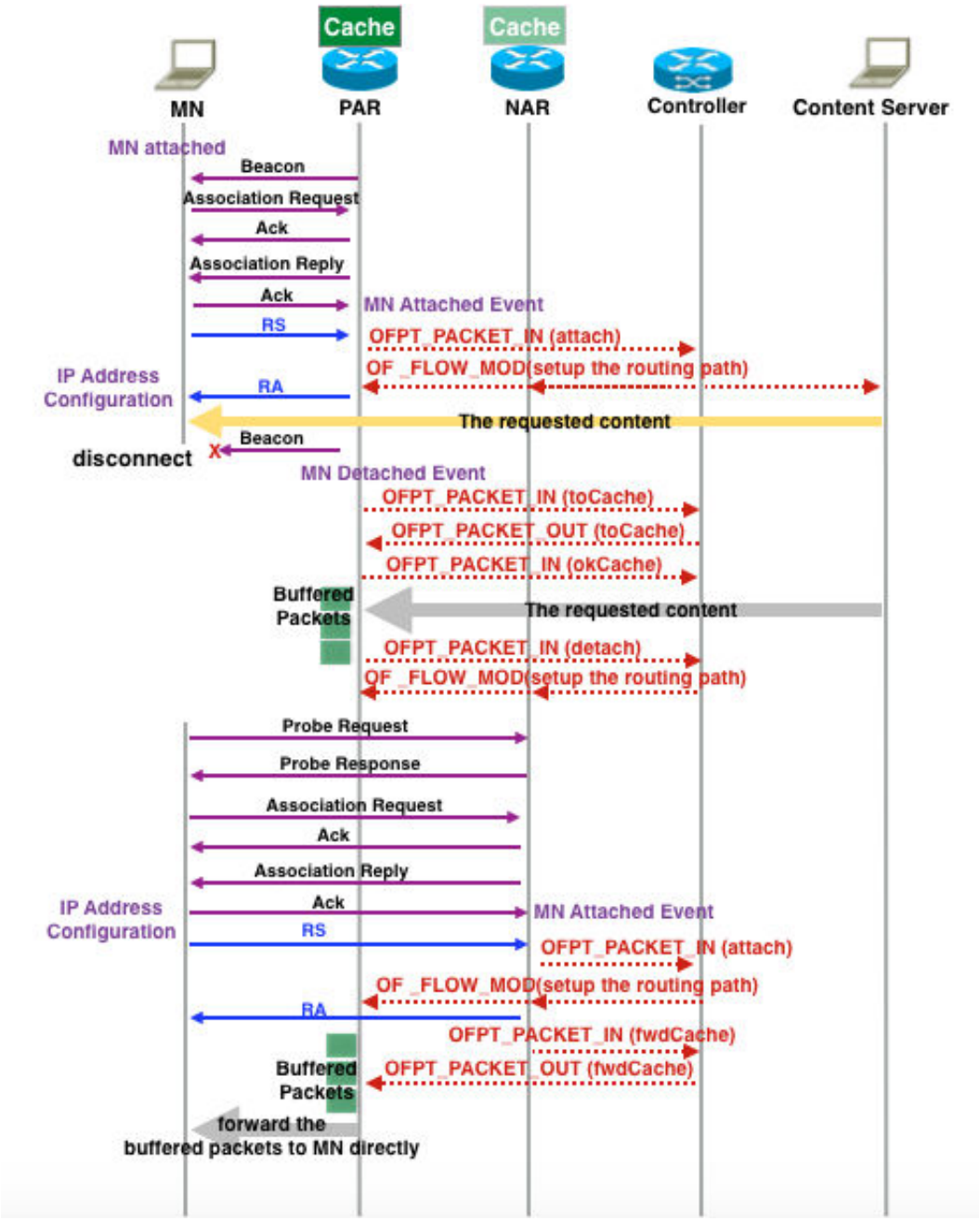

Figure 5.6 After Handover Caching Mode of Distributed Caching.

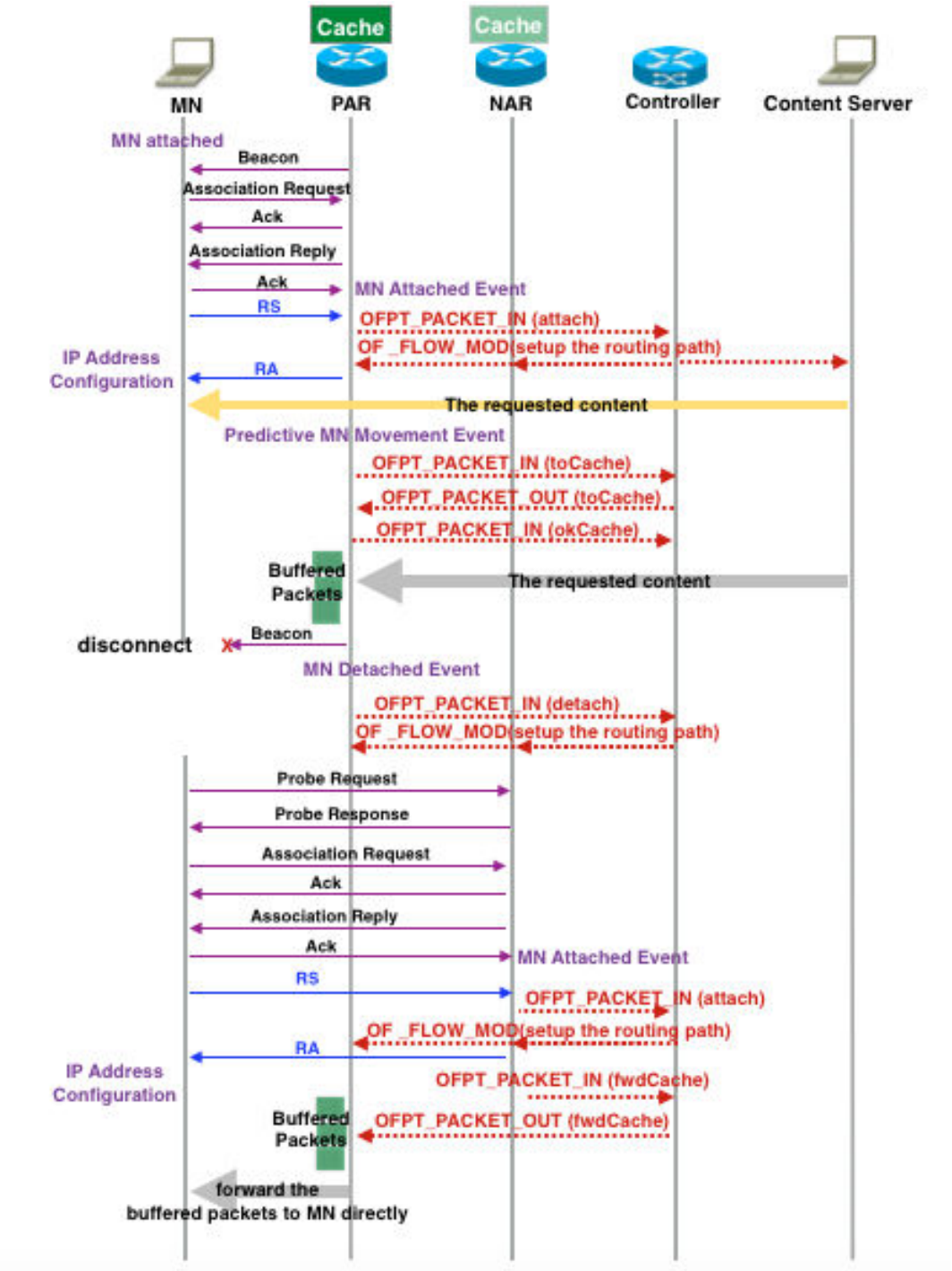

Figure 5.7 Before Handover Caching Mode of Distributed Caching.

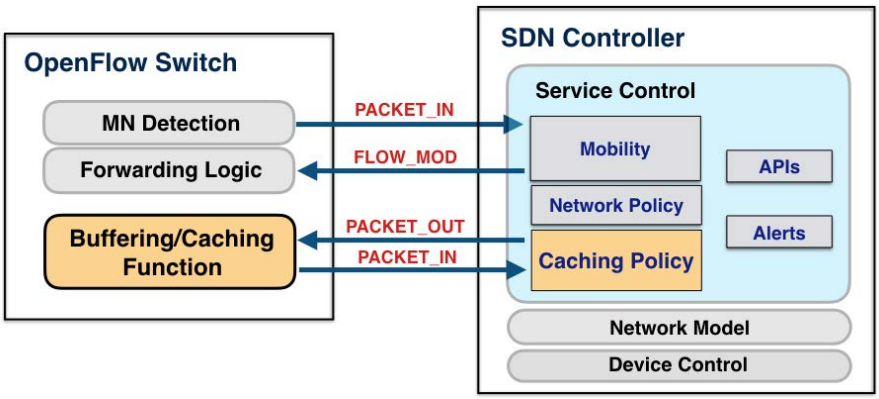

Figure 5.8 Cooperation between OpenFlow switch and SDN Controller.

During handover, PAR senses the MN detached event. Then, the routing path of MN will be deleted during handover duration and a new routing path will be added after the MN is attached at the NAR as shown in Figure 5.7. After that NAR sends OFPT_PACKET_IN (fwd-Cache) requested message to its controller for requesting the contents of MN. The controller verifies the data, and if there is the cached data of MN, it will send the OPPT_PACKET_OUT (fwdCache) to PAR for forwarding the data packets to MN directly.

Note that, in the worse case, MN have disconnected from PAR, but cache trigger is not enabled, PAR automatically performs the after handover caching mode.

5.2.3 Cache Operation

5.2.3.1 Caching Algorithm

To achieve the caching data during MN handover, the caching algorithms are required to be implemented in two parts: on the OpenFlow switches and the SDN controller. The perspective of cooperation between OpenFlow switch and SDN controller is illustrated in Figure 5.8. The SDN controller uses OpenFlow messages to get the network information from OpenFlow switch and to control the network services such as mobility or/and caching function.

The buffering/caching function is enabled and is managed by the SDN controller through PACKET_OUT message. The PACKET_IN message is used to inform the caching events to its SDN controller. The caching algorithm flowchart of OpenFlow switch and SDN controller are presented in Figure 5.9 and in Figure 5.10.

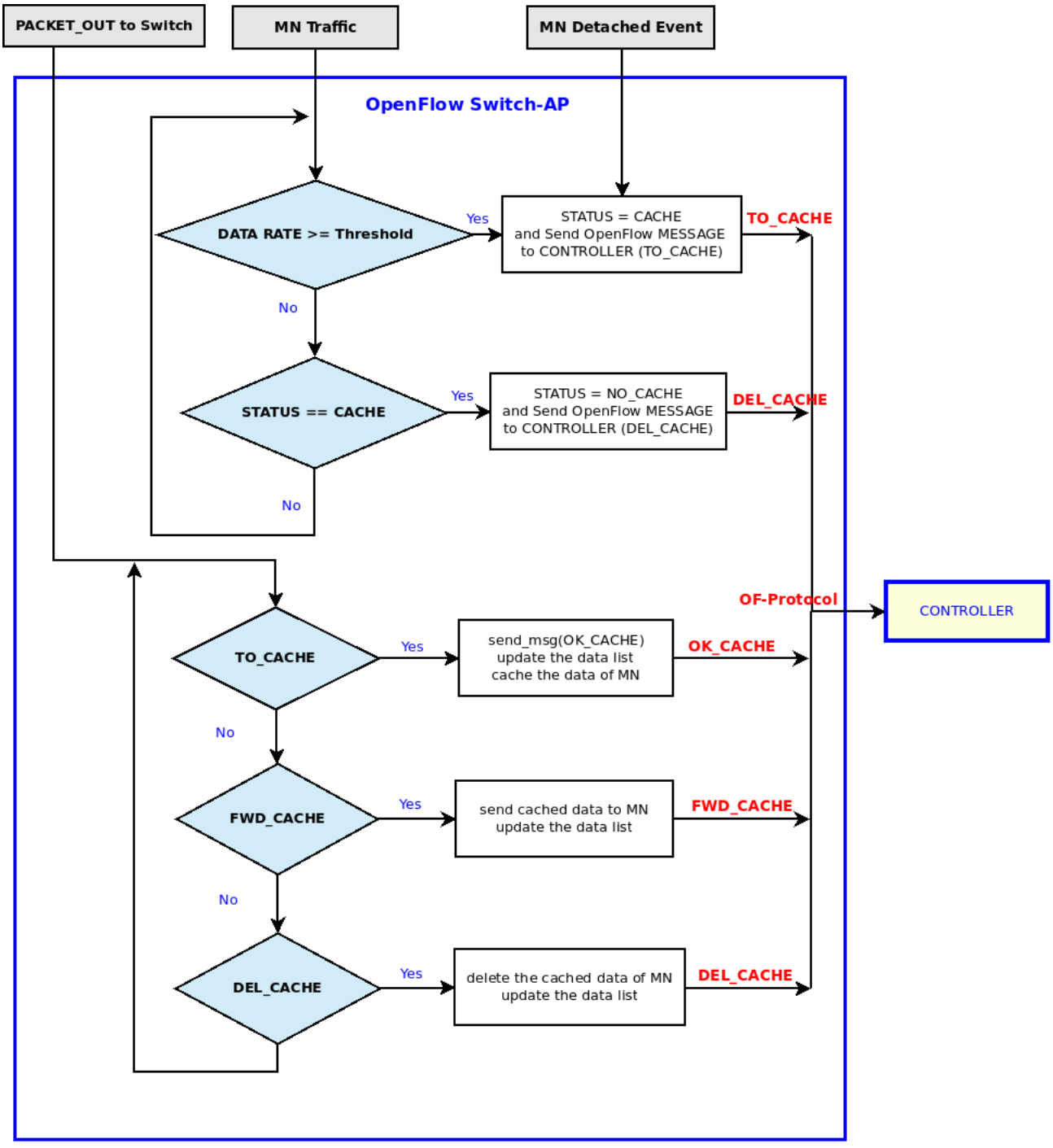

Figure 5.9 Caching Algorithm Flowchart at OpenFlow Switch.

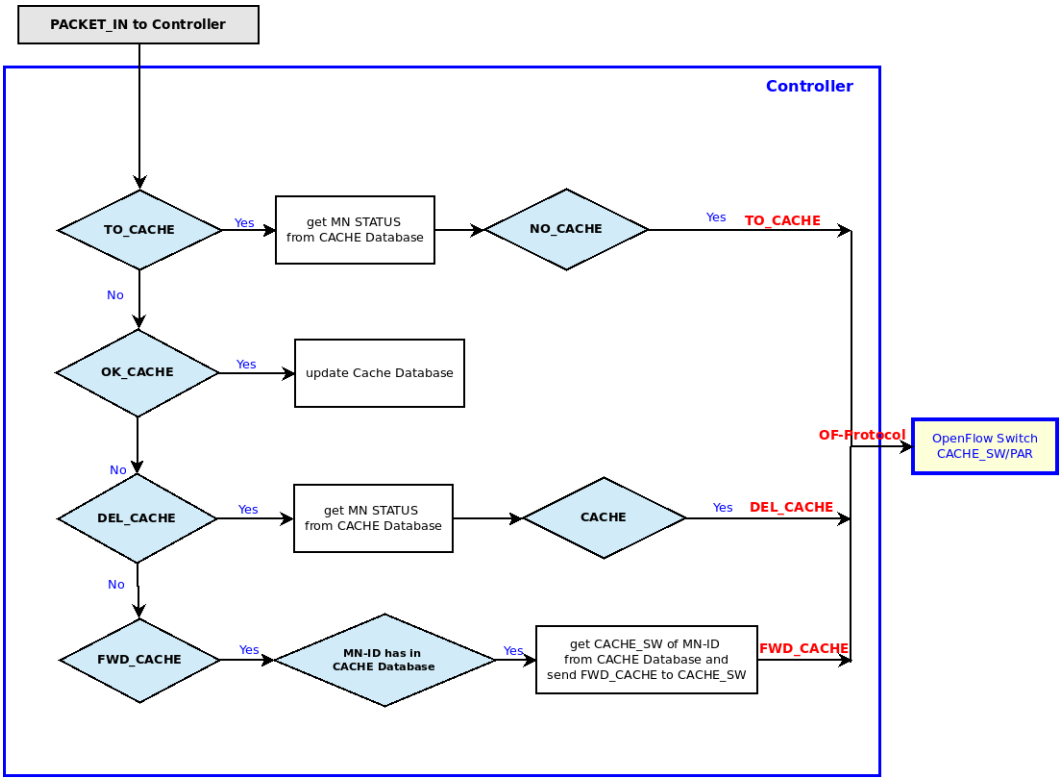

Figure 5.10 Caching Algorithm Flowchart at Controller

An OpenFlow switch predicts the MN handover event by using the data rate of MN traffic. If the data rate value is below or equals the threshold value, this means that MN may move to the other access router. The cache trigger will be enabled and CacheSW will start to cache the data depending on the caching policy.

PACKET_IN response part is an event to verify the OpenFlow PACKET_IN messages. We defined three new PACKET_IN reason fields: To_CACHE, FWD_CACHE, and DEL_CACHE. TO_CACHE is used for updating and caching the content data. FWD_CACHE is used for forwarding the cached content data to the destination. DEL_CACHE is used to delete the cached data in CacheSW.

- **Cache Trigger**

  We have simulated a simple wireless topology, by NS3, to determine the characteristics of WiFi regarding throughput and distances by varying the data rate and the mobile node position. The results, in Figure 5.11, show that when the mobile moves far away from an access point, its throughput capability is reduced. So this contrast is used for enabling cache trigger. We define the **"good signal area"**, as a zone that allows the received data rate of the mobile node, to be equal to the transmitted data rate.

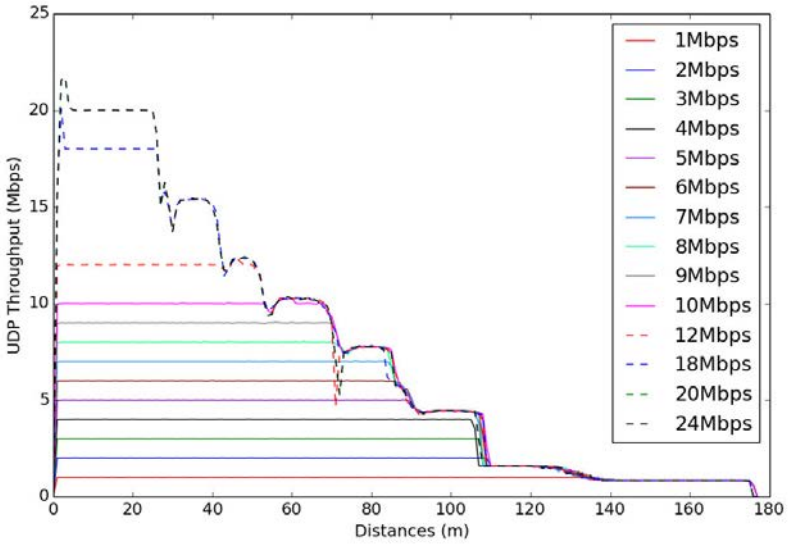

Figure 5.11 Throughput vs Distances.

- **Caching Policies**

  Once the architecture is established, the problem is to define the proper caching parameters and especially the caching duration. We consider that the quality of the network defines the duration of caching. We postulate that when the mobile moves such as the quality of its link to the access point are deteriorated the access point has an interest in keeping the information rather than in transmitting to it. Mobile will receive its cached data upon a better access quality event. More precisely, we propose two caching policies: **ON-OFF** and **Adaptive** caching policy. Both policies will start to cache the packets after a cache trigger is enabled. In the ON-OFF caching policy, the packets will be cached and non-forwarded while the mobile node stays in a low signal area. But the Adaptive caching policy still forwards the packets to the mobile node with an adaptive rate and will stop to forward the packets during the handover. The adaptive forward rate can be calculated from the physical rate of the mobile node position multiplied by the proportion of useful throughput [126] which varies according to the total number of active nodes associated with that access point.

  Caching policies are more precisely described in the next chapter.

5.2.3.2 Cache Implementation: NS3

We have created a new class which is named **"CacheApWifiMac"** class, to add to the access points capability to support caching operation as illustrated in Figure 5.12. CacheApWifiMac class inherits from ApWifiMac class of NS3 WiFi module. The packet classification and

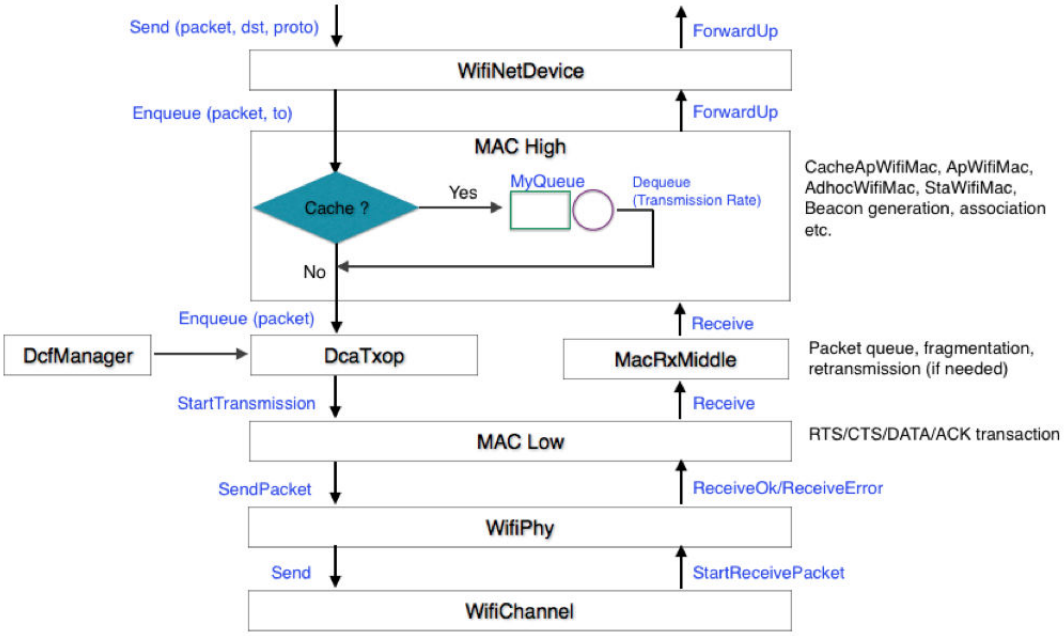

Figure 5.12 CacheApWifiMac Class

caching operation have been implemented in this class. These operations will be done at the MAC level of the WiFi model in the access points. Therefore, when the packets of mobile nodes are forwarded to the MAC high level, they will be classified specifically for mobiles based on caching policy and will be enqueued to a specific queue of each mobile. The packets will be dequeued and forwarded down to MAC low level for transmission while the forward state of the mobile is true.

5.3 Conclusion

We have shown the interest of SDN-Mobility improvement for video application both for UDP and adaptive HTTP. For this last, the buffering mechanism at the user level is not able to mask perfectly the mobility impairment; also we have proposed to improve our proposal by a network caching mechanism that could be used by user video application and even by other applications.

Among two modes of caching operation, reactive and proactive, we have chosen a reactive approach which able to decrease the network storage. Because in a react the data are stored when mobile moves while the data are stored at all time in a proactive mode. Besides, we have chosen to start the caching before the network disconnection by predicting the node

disconnection, based on the signal measurement. If not possible caching, cache trigger will be enabled after the disconnection.

The main idea of our caching is to make an adaptation mechanism that manages packet transmission at the network access point, depending on the network state. In fact, caching is quite a buffering.

Specifically, we specified caching triggers, as well as setting policies for caching duration. The latter is related to the quality of the network that we measure by the state of the signal. We proposed two kinds of processes, either the transmission is stopped in case of bad network state and restarted in case of good quality, it is the ON-OFF policy, or it is just slowed down and increased depending upon the network quality, it is the adaptive policy.

In the next chapter, we are going to evaluate the performance of our proposal onto two use cases: pedestrian and vehicular.

Chapter 6

Evaluation of SDN-Mobility with Caching Policy

Contents

In this chapter, the SDN-Mobility with caching policy which was presented in the previous chapter will be applied for different mobility patterns (pedestrian and vehicular) and will be evaluated through the performance analysis in two aspects: user and operator.

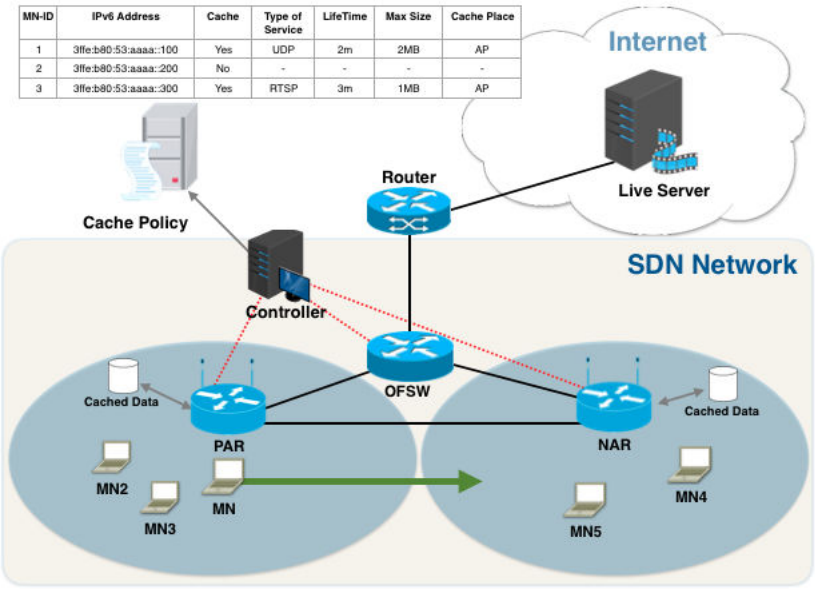

Figure 6.1 Pedestrian Context: SDN-Mobility with Cache Policy Architecture.

6.1 Caching Operation in Pedestrian Context

In this first part, we are going to evaluate the performance of the SDN-Mobility with caching on a use case which is a pedestrian displacement context (Figure 4.10 in Chapter 5) with various scenarios of network topology (overlapping or not overlapping access point).

6.1.1 Local Network Architecture

First of all, let us expose our management of mobile users by SDN architecture. It is similar to the explained architecture in Chapter 4. There are three main components: the controller, the Access Router (AR), and the Caching Policy as illustrated in Figure 6.1.

6.1.1.1 SDN Controller

It is an OpenFlow controller which locates in the same network as ARs. It has to be responsible for the flow table to all ARs in the network and to organize the cache events based on the caching policy.

In addition, to manage the flow table to all access routers, the controller manages the cache mechanism according to the defined policy. To allow the caching management to support

TABLE 6.1 The cached information in SDN Controller

Parameter	Description
MN-ID	Mobile Node identification.
SrcAddr	An IPv6 source address.
DstAddr	An IPv6 destination address.
CacheStatus	Specifies status of cache, YES means the content is cached.
ToS	Specifies type of traffic.
PAR	Specifies the previous access router.
CurAR	Specifies the current access router.
CachedPlace1	Specifies the place which stored the content.
CachedPlace2	Specifies the place which stored the content. In case, MN moved and attached more than one AR.

TABLE 6.2 The parameter of caching rules

Parameter	Description
MN-ID	Mobile Node identification.
MN-IPv6	An IPv6 address of MN.
CacheOp	Specifies YES value, cache requirement for that MN.
ToS	Specifies type of traffic which requires to cache.
ExpTime	Specifies the expiration of cached data.
MaxSize	Specifies maximum size of cached content.

moving nodes, the SDN controller needs to record the cached information of each mobile node as shown in TABLE 6.1.

6.1.1.2 Caching Policy

A caching policy defines rules for caching the requested content of each MN. It can be installed either on the SDN controller or the other servers. SDN controller can get the rule of each mobile node from the caching policy through Application Programming Interface (API).

SDN controller uses a caching policy to determine which mobile requires to cache the requested content. Caching parameters, that are shown in TABLE 6.2, will be sent to PAR or NAR after the mobile attached for caching configuration of the mobile.

6.1.2 Basic Operation

The overall functioning of our architecture is as follows. Mobile Node (MN) process can be separated into two procedures: MN registration and MN handover. The OpenFlow messages

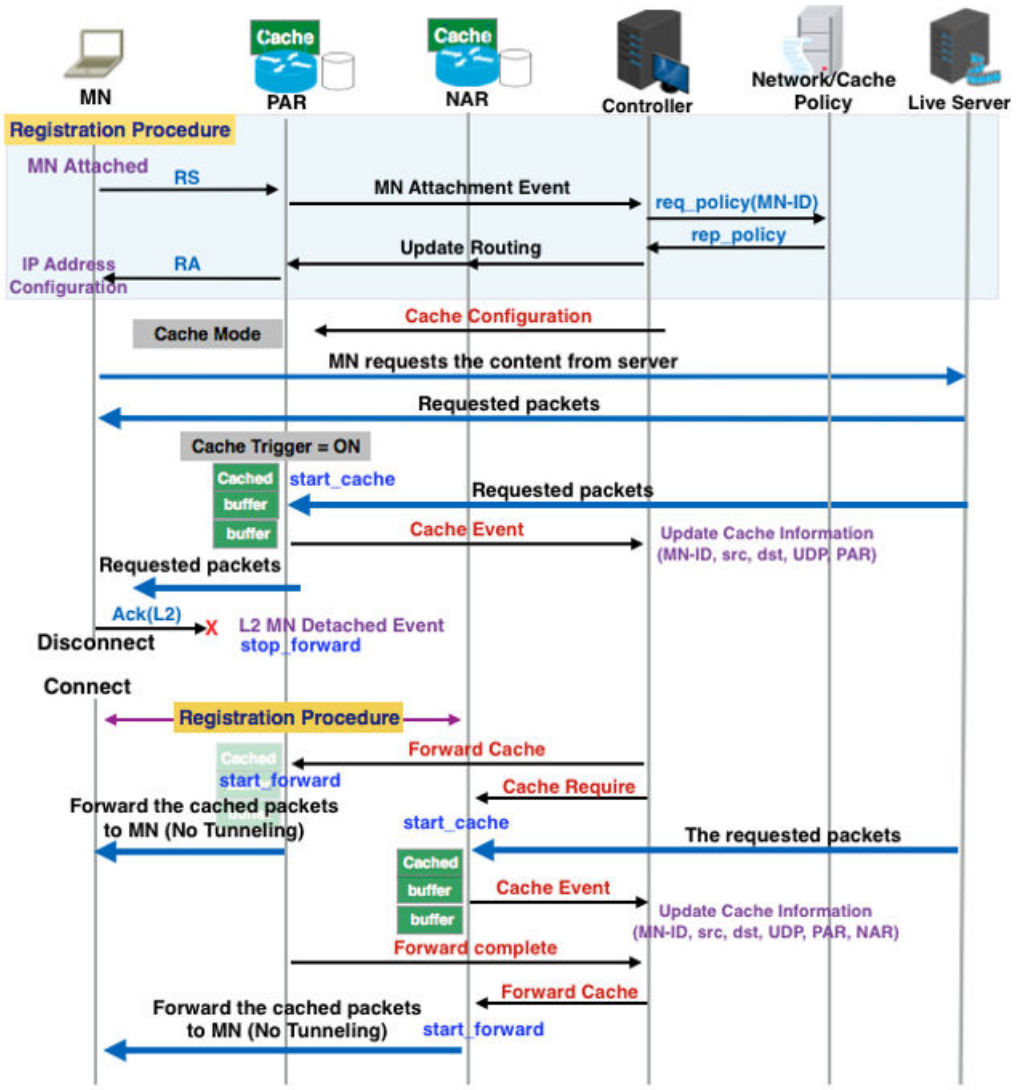

Figure 6.2 SDN-Mobility with Caching Policy Signaling (Pedestrian Context).

are exchanged between the controller and ARs for notifying of the MN arrival, for updating the routing path, and for additional caching options. The SDN Mobility with cache policy signaling is illustrated in Figure 6.2.

6.1.2.1 MN Registration

When the MN enters the network, PAR detects the attached event and sends an OpenFlow message to its controller for informing this event. Once the controller receives and processes that message, it requests the caching policy of the MN from the cache policy server. After that, the SDN controller sends the OpenFlow messages to all ARs for adding the routing flow

of the MN and sends an OpenFlow message to the PAR for enabling the caching mode of the MN. Then, the Correspondent Node directly communicates with the MN. Note that, we assume that the controller has the network policy allowing the MN to access the network and a caching rule of the MN.

If the MN is communicating and moving simultaneously far away from the PAR, the ability of received data rate of the MN changes. If the difference of data rate is higher than a given threshold, the cache trigger is enabled. For the Adaptive caching policy, the PAR starts to cache the packets of the MN and adapts the forwarding rate to send the cached packets to the MN. The PAR caches and stops to forward the packets of the MN in the ON-OFF caching policy. At the same time, the PAR sends an OpenFlow message to its SDN controller for informing the data of the MN has been cached at PAR. Then, SDN controller updates its cached information.

6.1.2.2 MN Handover

For Adaptive caching policy, when the MN detaches from the PAR, the PAR stops to forward the packets of the MN. After the MN completely registers with the NAR and received a forward cache command from the controller, the PAR forwards all cached packets to the MN directly.

For the ON-OFF caching policy, the PAR will forward the cached packets when the MN moves in the good signal area. Meanwhile, the SDN controller sends an OpenFlow message to the NAR for caching and stops forwarding. The new packets of the MN are cached at the NAR. After the PAR has finished forwarding all cached packets, it informs the controller of this event. The controller updates the cached information and advertises to the NAR to directly forward the new cached packets to the MN. Note that, the MN attached/detached event can be done in layer two by using IEEE 802.21 Media-independent Handover (MIH) or OpenFlow-specified.

6.1.3 The Experimental Network Topology

We set up an experimental network with cache policy that consists of the main components as illustrated in Figure 6.1. Two network architectures have been considered: overlapping network (Net1) and no-overlapping network (Net2). In the overlapping network, the distance between PAR and NAR is 260 meters and is 355 meters for the no-overlapping network. These architectures have been chosen because, after MN handover, MN lives in the different signal areas and takes different handover delay. So, we can study how it impacts: 1) the

User Quality of Service (U-QoS), determined by transmission time, loss and jitter and 2) the Operator Quality of Service (O-QoS), determined by channel occupancy time and fairness index.

The percentage of packet loss in the wireless network depends on the velocity of the MN and the traffic data rate of the MN. From [104], the number of packet loss will significantly increase when the velocity of the MN is less, and the traffic data rate is higher or equal than 0.9 Mbps, due to WiFi loss and handover loss. Meanwhile, only handover loss appears at a data rate less than 0.9 Mbps. To experiment a lossy network context, we focus on the movement speed for walking, MN speed equal to 1 meter per second, and use data traffic higher than or equal 1 Mbps.

6.1.4 Performance Analysis

This section shows the experimental results and analysis. The experimental topology was set up as described in the previous section. In both networks, we measured the performances in two cases; single MN traffic and multiple MNs traffic. The same configured parameters are set in each network for measuring the performance with different caching policies. Each simulation with No-cache, ON-OFF caching policy, and Adaptive caching policy has run ten times. The results are obtained by averaging.

6.1.4.1 Case 1: Single traffic MN

With this first traffic case, we show the usefulness of the caching mechanism. We are not only interest in the quality of service which is provided to the mobile user but also to the impact of the scheme at the network level, a mechanism that would increase the bandwidth occupancy would cause a problem of network efficiency since the network supports several devices.

In both overlapping and non-overlapping network. The MN is generated and set to the position (0,0) from the AP. The mobility model of the MN is constant velocity at speed 1 meter per second. The MN receives a CBR UDP traffic that is generated from a live streaming server for 200 seconds. The UDP datagram size is set to 1400 MBytes.

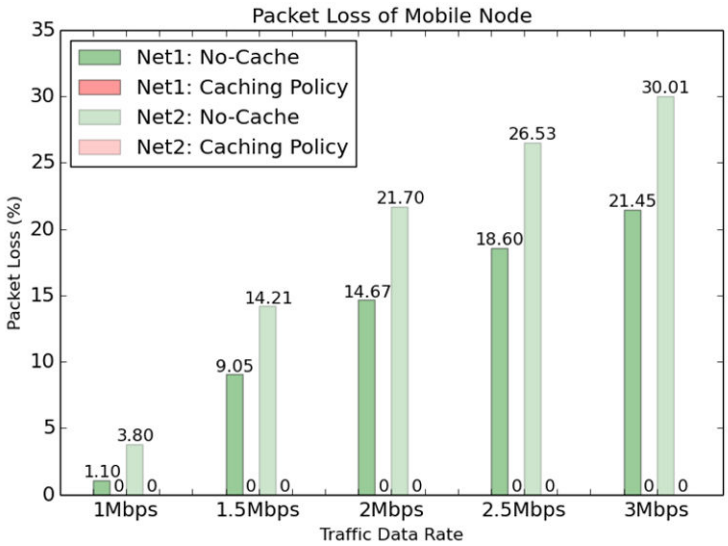

Figure 6.3 Percentage of Packet Loss in Single Traffic MN case.

User Quality of Service (U-QoS)

We evaluate the percentage of loss and the transmission time.

- ***Packet Loss***

  The results, as shown in Figure 6.3, indicate that both ON-OFF and Adaptive caching policy eliminates the packet loss at all data traffic rates. Note that, as expected the percentage of packet loss with No-cache in Net2 is higher than in Net1 because Net2 takes handover time more than Net1 and MN spends a long time in the weak signal area.

- ***Transmission Time***

  The transmission time to the MN is illustrated in Figure 6.4. The results show that the transmission time of the ON-OFF policy is highest in both networks. About 30% and 62.5% higher than No-cache policy in Net1 and Net2. These results are caused as the ON-OFF policy, is forwarding the packets only in situations where the MN is living in the good signal area. Then, the transmission time of the ON-OFF caching policy is depending on the network topology and moving pattern of MN. The Adaptive caching policy takes a higher transmission time than No-cache policy (about 4.5% and 5.85% in Net1 and Net2). Indeed the Adaptive caching policy degrades the forwarding data rate while the MN lives in the weak signal area for avoiding packet loss.

  From the results, we can conclude that U-QoS is improved by caching, there is no loss. The cost regarding delay depends on the caching policy. The Adaptive caching policy seems preferable.

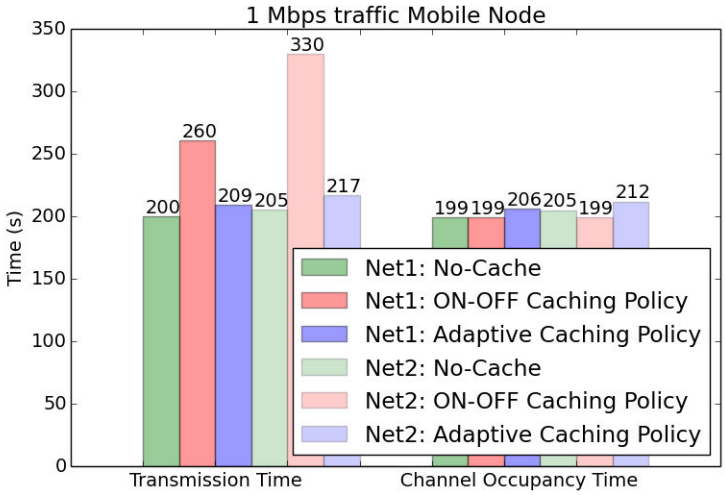

Figure 6.4 Transmission Time and Occupancy Time of Single Traffic MN.

Operator Quality of Service (O-QoS)

- ***Channel Occupancy Time***

  Considering the channel occupancy time of each network as shown in Figure 6.4, the occupancy time of the ON-OFF caching policy in Net1 is equal with No-cache policy but less than in Net2 about 2.9%. In this network, the MN lives in the weak signal area (the ability of a received data rate less than the transmitted rate) after the handover. Then, some packets are cached in the access point, leading to cost more delay than in No-cache policy. But the ON-OFF caching policy forwards packets only in the good signal area, (the transmitted data rate equals the received data rate). Therefore the ON-OFF caching policy doesn't cache packets in the access point, leading to the same occupancy time in both networks.

  For the Adaptive caching policy, it takes a higher occupancy time than the No-cache policy (about 3.5% and 3.4% in Net1 and Net2). Indeed, the Adaptive caching policy degrades the forwarding data rate depending on the MN position, the packets are cached. These cached packets will cost more in terms of the occupancy time.

  To conclude, with caching, packets are transmitted more rapidly, and more packets are transmitted (there is no loss) resulting in quite the same occupancy time for caching and non-caching method. There is no O-QoS degradation.

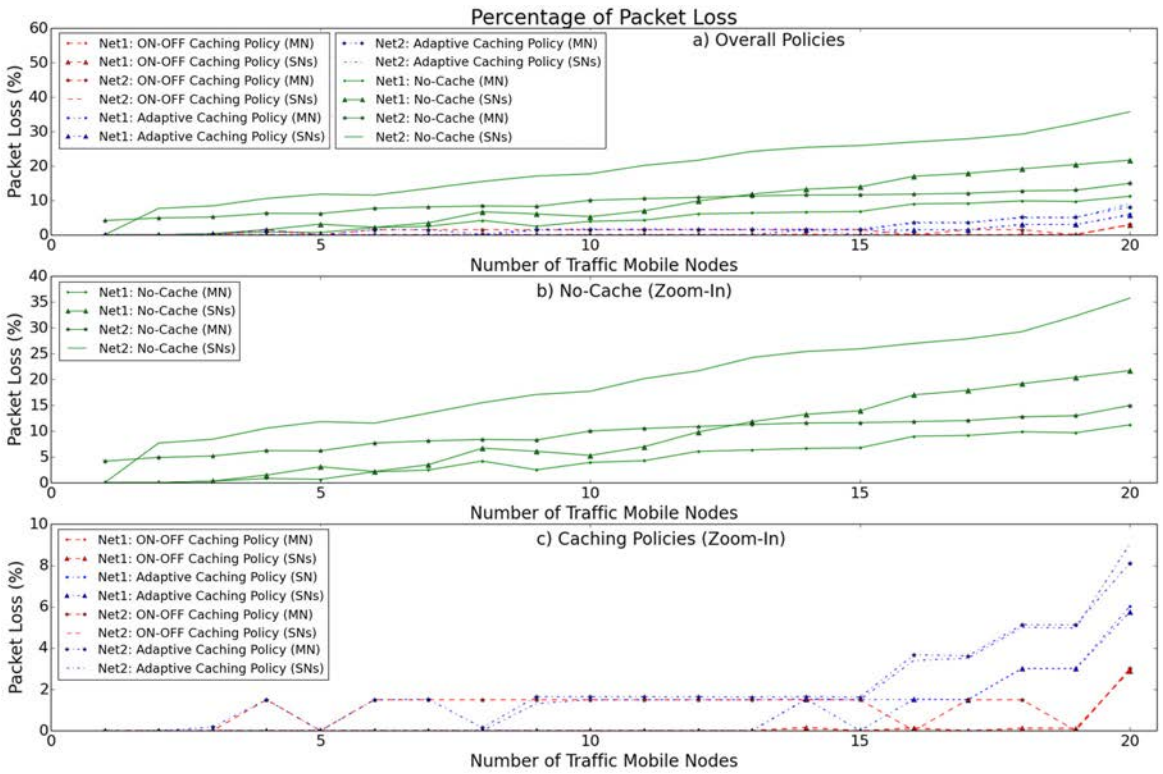

Figure 6.5 Percentage of Packet Loss in Multiple Traffic MNs.

6.1.4.2 Case 2: Multiple Traffic Nodes

We increase the number of traffic nodes to degrade the network quality and to consider the impact of the proposed policies over other nodes. The UDP traffic rate of each node is set equally. The MN is generated and set at a position (0,0) from AP; its mobility model is constant velocity, 1 meter per second. The other nodes are stationary nodes. Their positions are set in the good signal area, away from the AP about 1 meter to 5 meters.

User Quality of Service (U-QoS)

- ***Packet Loss***

  The percentage of packet loss is shown in the Figure 6.5. Considering No-cache policy, in conformance to the result of [126], the packet loss of stationary nodes significantly increases when the number of traffic nodes is high on both networks. Because, while the MN is moving far away from the AP, the MN may encounter bad transmission conditions, leading to the possibility of high packet loss. When the AP transmits a packet to the MN if the AP does not receive the layer two acknowledgments from the MN, the AP will retransmit the lost packet. Therefore, the traffic of the MN occupies

the radio channel more than the other nodes, leading to the performance anomaly: the other nodes lose more packets, because of retransmission. But for both Adaptive and ON-OFF caching policies this problem does not occur because they stop forwarding the packets before the MN handover, leading to gain a fewer loss that caused by WiFi loss of the MN movement.

- ***Packet Interval Time and Transmission Time***

  For visibility reason, we present results for two nodes. We obtained similar results with more nodes. The MN starts to move from the initial position (0.0) far from the AP and will approximately handover at second 175. The packet interval time of the MN and stationary nodes of both networks are shown in Figure 6.6 and Figure 6.7. For, the first period, second 0 to the second 25, the interval packet of each policy is close to one sent by the source (CN). Then, the interval packet is spread when the MN is moving further away from the AP, caused by the possibility of high packet loss from the MN movement and the distance of the MN. The AP retransmits the MN packets, and the packets of stationary nodes are cached at the AP, for sending after the channel becomes idle.

  During the handover, the AP should serve the stationary nodes, but the results of No-cache policy, in both networks have shown that fewer packets have been sent to stationary nodes. Because the AP tries to retransmit the MN packets, blocking the traffic of stationary nodes, it leads to unfairness. For Adaptive caching policy and ON-OFF caching policy, the AP serves only the traffic of stationary nodes, because both policies do not forward the MN packets in this duration.

  After MN attached the new AP (NAR), No-cache policy and Adaptive caching policy continue to send the MN packets, but ON-OFF caching policy starts to forward in Net1 and will forward later, when the MN lives in the good signal area, for Net2.

  Considering the packet interval of the stationary nodes, after the MN handover, the results of the ON-OFF policy are close to those sent by the CN. From this results, we see that the MN movement affects the traffic of the other nodes while the MN is moving in the weak signal area.

  The packet interval time of Adaptive caching policy is quite similar to the No-cache policy one in both networks. But the transmission time of the MN with the Adaptive caching policy is slightly higher than with the No-cache policy about 4.5% and 6.5% in Net1 and Net2.

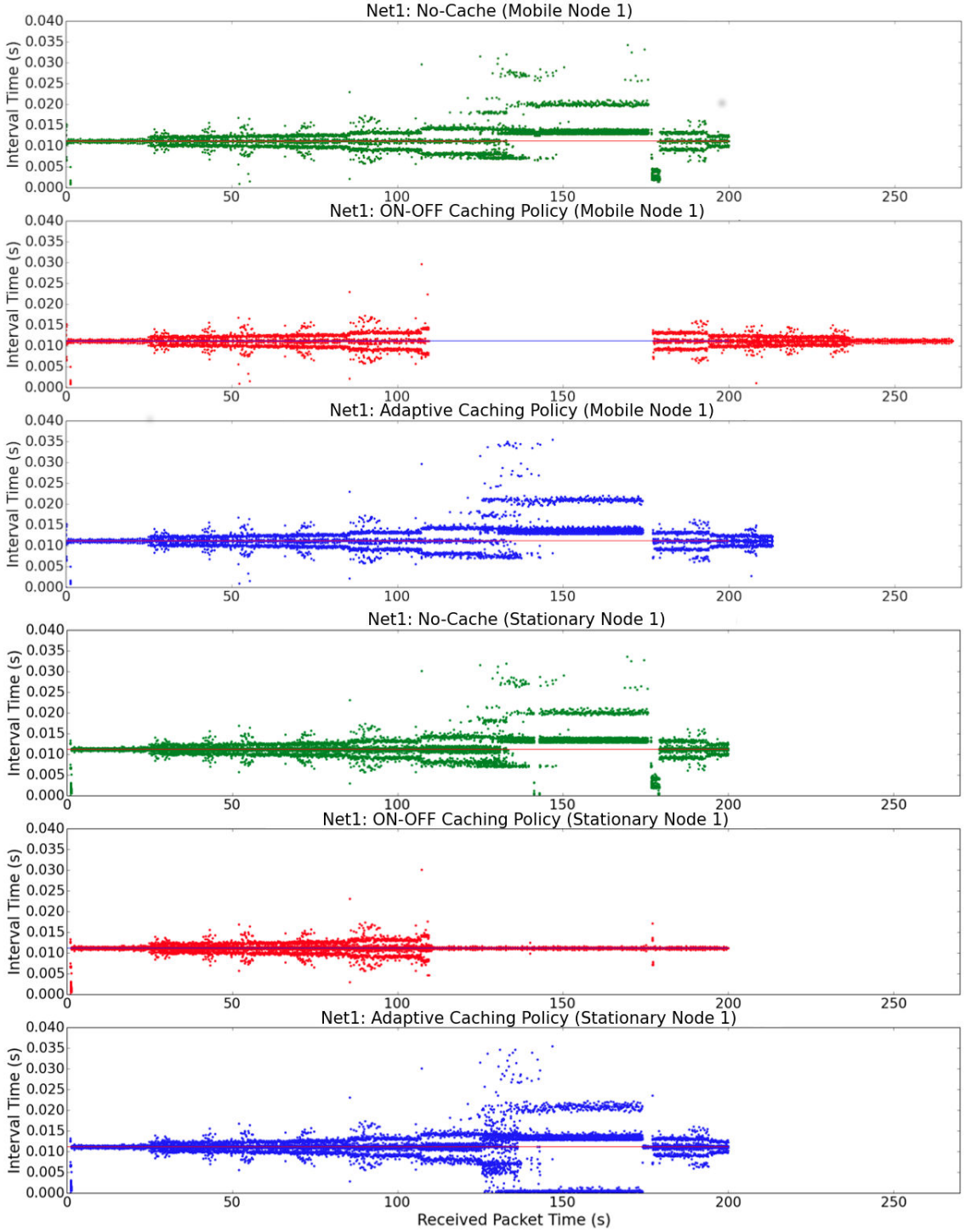

Figure 6.6 Interval Time of Mobile Node and Stationary Node in Net1.

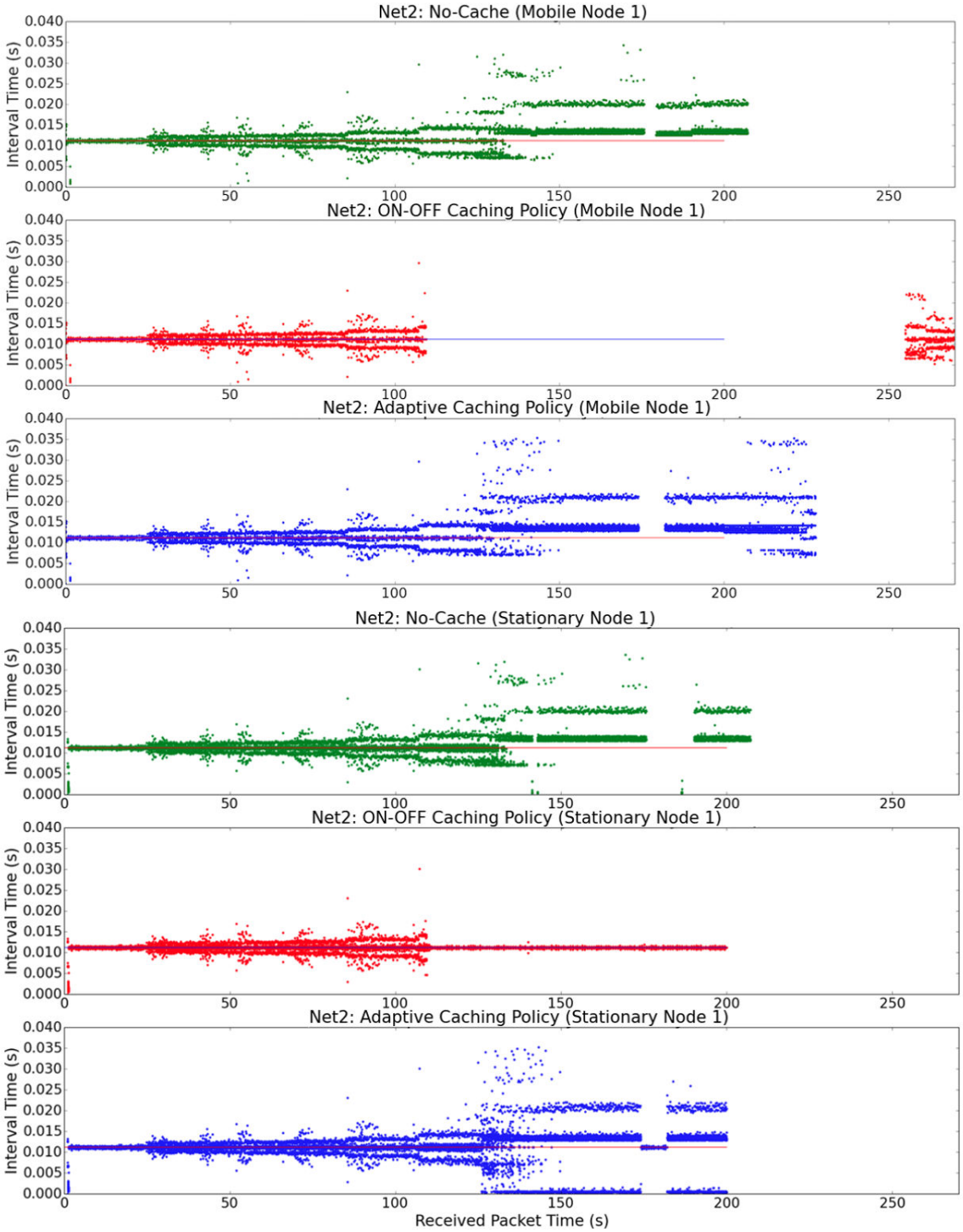

Figure 6.7 Interval Time of Mobile Node and Stationary Node in Net2.

Considering the packet interval time, the ON-OFF caching policy is better than the No-cache policy and the Adaptive caching policy for both MN and stationary nodes. But the ON-OFF caching policy is worse regarding transmission time of the MN. This cost is higher than No-cache policy about 35% and 72.5% in Net1 and Net2. The transmission time of the MN in the two nodes case is higher than the one MN case about 3.8%, due to the WiFi bandwidth sharing.

Operator Quality of Service (O-QoS)

- ***Bandwidth Fairness***

  We examine the sharing of the bandwidth between several users (1 to 20 users) by using Jain's fairness index on bandwidth occupancy as illustrated in Figure 6.8. The results of Adaptive caching policy are better than No-cache policy ones whatever the number of traffic nodes. But the ON-OFF caching policy gives high bandwidth fairness as the number of traffic nodes is high. Because of the ON-OFF caching policy sends the packets only where the MN lives in the good signal area, it takes more transmission time, then the average of bandwidth is low.

We can synthesize the results presented in this section by stating that the caching time which depends on the quality of the network and the Adaptive caching policy affects the quality of the user service but also that of the operator insofar as the service may be unfair. A significant caching time but with a fast transmission, as implemented in the ON-OFF policy, improves the operator quality to the detriment of the user quality. A shorter caching time but with a higher transmission rate, as proposed by the Adaptive caching policy, gives an opposite effect.

6.1.5 Summary

In this section, we have proposed to use the cache policy with the SDN-Mobility approach for providing IP mobility management in pedestrian context. It can eliminate or minimize some packet loss in SDN mobility network. Two caching policies have been proposed, the ON-OFF caching policy and the Adaptive caching policy. Their performance analysis has been measured regarding the percentage of packet loss, transmission time, channel occupancy time and bandwidth fairness. According to the obtained results, we can conclude that both proposed cache policies can improve SDN-Mobility regarding packet loss. Even though, the ON-OFF caching policy is worse concerning transmission time but good regarding the channel occupancy time. Then, the ON-OFF caching policy is suitable to be used with

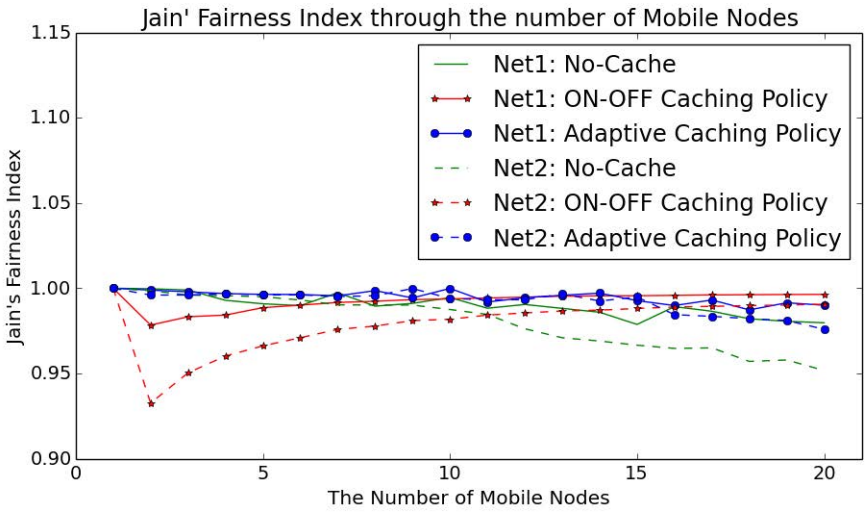

Figure 6.8 Jain's Fairness Index through the number of traffic nodes.

the applications which are non-sensitive to the delay such as file transfer or some media applications which download all contents before display. For delay-sensitive applications, an Adaptive caching policy is properly used. Even though, the transmission time of the Adaptive caching policy is a bit higher than No-cache policy. But this weakness will not affect for actual usage because the general media display applications buffer the video before display.

6.2 Caching Operation in Vehicular Context

Connected vehicle (CV) is one area of particular interest to investors from the automotive industry perspective that companies are pouring budget into the Internet of Things (IoT) connected vehicles. The first, the connected vehicles were designed for a safety service, and only voice is enabled to contact the emergency call center when an accident occurs. After the capacities of connected vehicles have been increased, for the localization with Global Positioning System (GPS) and the ability to transfer voice and data at the same time by supporting 4G LTE. The connected vehicles are driven based on the roadmap from integrated navigation which supports online services to respond to a driver for intelligent routing, traffic alerts, fuel stations, existing attractions, etc. Moreover, many infotainment services are provided for the passengers since WiFi hotspot can be enabled in connected vehicles. While the passengers are enjoying the entertaining program along with their trip, it is possible that the connected vehicles may change to the other point of attachment, leading to a possible disconnection and disturbed services that makes the passengers unhappy. In this section, we address this problem by proposing a new possible approach for providing

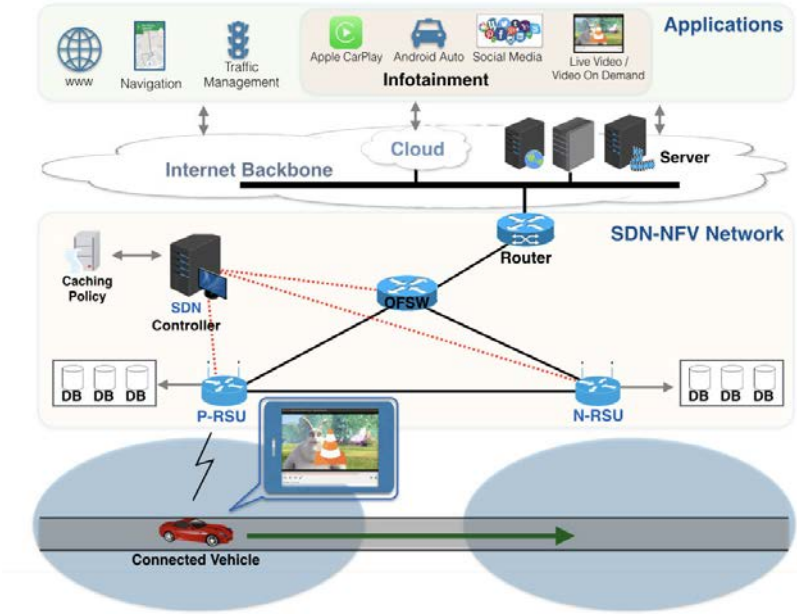

Figure 6.9 The SDNVMM Architecture for Connected Vehicles.

mobility management in vehicular networks. Because SDN-Mobility approach has gained high performance and is also efficiently and flexibly deployable in the real network. Therefore we propose to extend SDN-Mobility approach and to apply the ON-OFF caching policy to minimize service interruption in vehicular networks that is called SDN Vehicle Mobility Management (SDNVMM).

6.2.1 Global Architecture

Figure 6.9 illustrated the SDNVMM architecture for connected vehicles. It can be separated into three main layers: Application, Internet backbone, and access infrastructure. The proposed access infrastructure is similar to the pedestrian context (previous section) that consists of three main components: Road Side Units (RSUs), SDN controller and Caching Policy.

6.2.1.1 Application Layer

Application layer provides the connected services such as intelligent navigation, traffic management, the infotainment service, etc.

6.2.1.2 Backbone Layer

The interconnection of many large networks is called Internet backbone. Many network service providers (NSPs) are connected in Internet backbone to provide the Internet for their customers. Several servers are located on the Internet backbone such as web servers, mail servers, media servers, etc.

6.2.1.3 Infrastructure Layer

- **Road Side Unit (RSU)**

  Road Side Units (RSUs) are installed on the roadside to provide vehicles connectivity. The RSUs can be equipped with the wireless transceiver operating on either Dedicated Short-Range Communication (DSRC) or WiFi [127].

  The general access infrastructure for connected vehicles consists of eNBs and RSUs. The eNBs provide LTE communication that covers all city areas. The link communication between connected vehicles and eNBs is more stable than link communication with RSUs. Then, the LTE connectivity is recommended for the time-sensitive services such as safety services, intelligent navigation, and voice data. Video data or streaming data is recommended to be transferred through WiFi link because WiFi provides high data rate communication [128].

  In this thesis, we focus on the infotainment application such as video on demand and live streaming video. So RSUs based on WiFi technology and support the OpenFlow protocol. They also support caching function and coordinate with OpenFlow command messages with the SDN controller. In the following, we distinguish the Previous Road Side Unit (P-RSU) from the New Road Side Unit (N-RSU).

- **SDN Controller**

  The SDN controller is an essential component which supports the OpenFlow protocol. It is located on the same network as RSUs and acts as the brain of the vehicular system that can automatically and dynamically manage and control the network components in the vehicular network. Because the SDN controller can monitor the information of connected vehicles such as origin, destination, the road path, vehicle position. It also knows the network information such as the load of each RSU, the bandwidth usage of each RSU, the location of RSU, etc.

In this context, the main function of the SDN controller can be divided into two parts according to the purpose of the task: routing path management and cache management.

Routing Path Management:

The SDN controller automatically updates the routing path when the connected vehicle drives in SDN vehicular network and when the connected vehicle changes to another RSU network.

Cache Management:

For cache management, the SDN controller uses the OpenFlow protocol to enable the cache mode, configure and manage the cached data on the RSUs. These processes show the ability of SDN to easily and flexibly enable the network function services.

After the cache rules are configured in the RSUs, RSUs can decide to cache the CV data by themselves under caching policy. This is an edge computing which can minimize the number of signaling and share the load with SDN controller.

- **Caching Policy**

  The cache Policy contains the defined rules for caching the requested content of each connected vehicle. It can be installed either on the SDN controller or the other servers. SDN controller can get the rule of each connected vehicle from caching policy through Application Programming Interface (API).

  For the vehicular network due to previous research, we propose to use an ON-OFF caching policy which only transfers the data while the connected vehicle lives in the good signal area. Even though the experimental results of ON-OFF caching policy with walk mobility model are worst regarding total transmission time, the ON-OFF caching policy gains a better jitter delay. In this thesis, we focus on in-vehicle communication, and infotainment services are considered that's why an ON-OFF caching policy is chosen.

6.2.2 Intelligent Operation

The SDNVMM operation can be separated in two procedures: registration procedure and pre-uploading data procedure as illustrates in Figure 6.10.

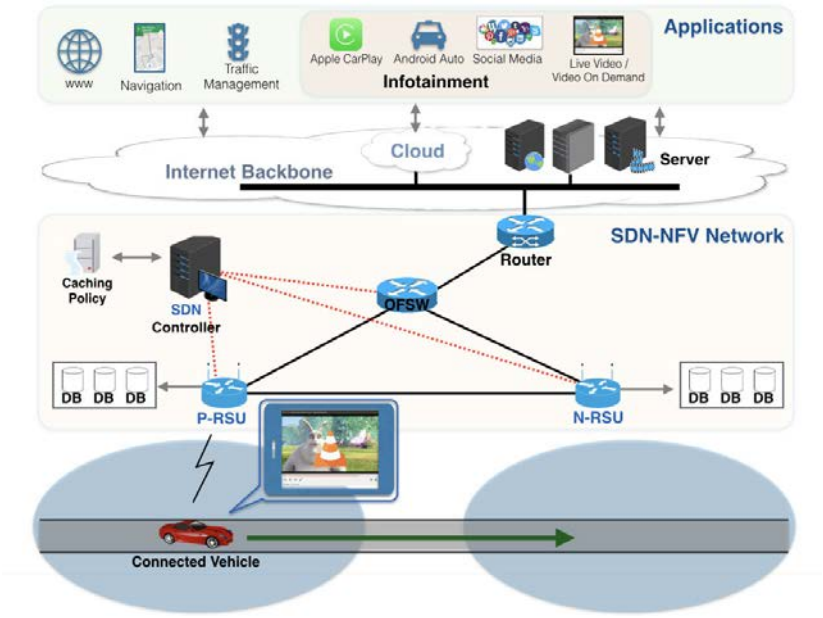

Figure 6.10 The SDNVMM Operation

6.2.2.1 Registration Procedure

When the engine of the Connected Vehicle (CV) is started, the connected vehicle requires an IPv6 address by sending the Router Solicitation (RS) message to the closest RSU (P-RSU). After that, the P-RSU informs the CV attachment event to the SDN controller to authenticate the CV permission with the network/cache policy. In this case, assume that CV has been defined in the network policy. Then, the CV routing path will be added in all RSUs, and the caching requirement of the CV will be configured at the P-RSU by the SDN controller. Once the connected vehicle receives the Router Advertisement (RA) message, an IPv6 address is assigned by IPv6 auto-configuration. The connected vehicle can access the Internet to update its information or to access the infotainment services.

While the connected vehicle drives on the road far away from P-RSU, the bandwidth capacity of CV is changed, leading to enabling the cache trigger. The P-RSU starts to cache the CV data under the caching policy and informs the caching event to the SDN controller. In this thesis, we apply an ON-OFF caching policy that will cache and stop to forward the CV data after the cached trigger is enabled.

6.2.2.2 Pre-uploading Data Procedure

Concerning the data transfer, we propose to use the predictability of the vehicular movement to anticipate the loading of the user data. As for FMIP solution, the idea is to accelerate the handover by preparing the handover.

The data pre-uploading procedure occurs after the caching trigger is enabled. Meanwhile, the connected vehicle keeps driving on the roadmap with the GPS navigation. The CV position can be known by GPS location, and the SDN controller localizes each RSU. Therefore the SDN controller can predict the next RSU with which the connected vehicle will attach. Then, the routing path of the CV and the cached requirement can be updated at the N-RSU before the connected vehicle handover. Also, the cached data at the P-RSU will be offloaded to N-RSU before the connected vehicle handover.

Then, the connected vehicle attaches to the new network; the CV data is continually transferred directly from N-RSU.

6.2.3 The Experimental Network Topology

In a real situation, from origin to destination, it is possible that a vehicle will be driven through the road path without Internet connectivity. Maybe some RSUs or APs are broken or fixed. So for user aspect, we propose three experimental network topologies depending on the network coverage: overlapping network coverage (Net1), closed network coverage (Net2), and no-overlapping network coverage (Net3).

Figure 6.11 illustrates the roadmap which is used for the testing approach in this context. Each vehicle drives from the same origin to same destination (from our laboratory in the ENSEEIHT to François-Verdier metro station). As shown the vehicle direction is plotted on a red line. Three networks have been chosen for two reasons. The first reason, all configurations are possible to meet in the real situation, Net1 and Net2 fit normal situation with overlapping network coverage and closed network coverage. For Net3 will be met in the worst case that the connected vehicle drives in a way that does not have RSUs or in case that some RSUs are broken or is fixed. The second reason is that the connected vehicle will live in the different signal area after its handover and takes different handover delay.

We consider the operator aspect, to decrease the operating cost. The experimental network topology is set up like as Net3 in Figure 6.11 by varying the distance between the WiFi network coverages from 10 to 120 meters. The simulation runs one connected vehicle from

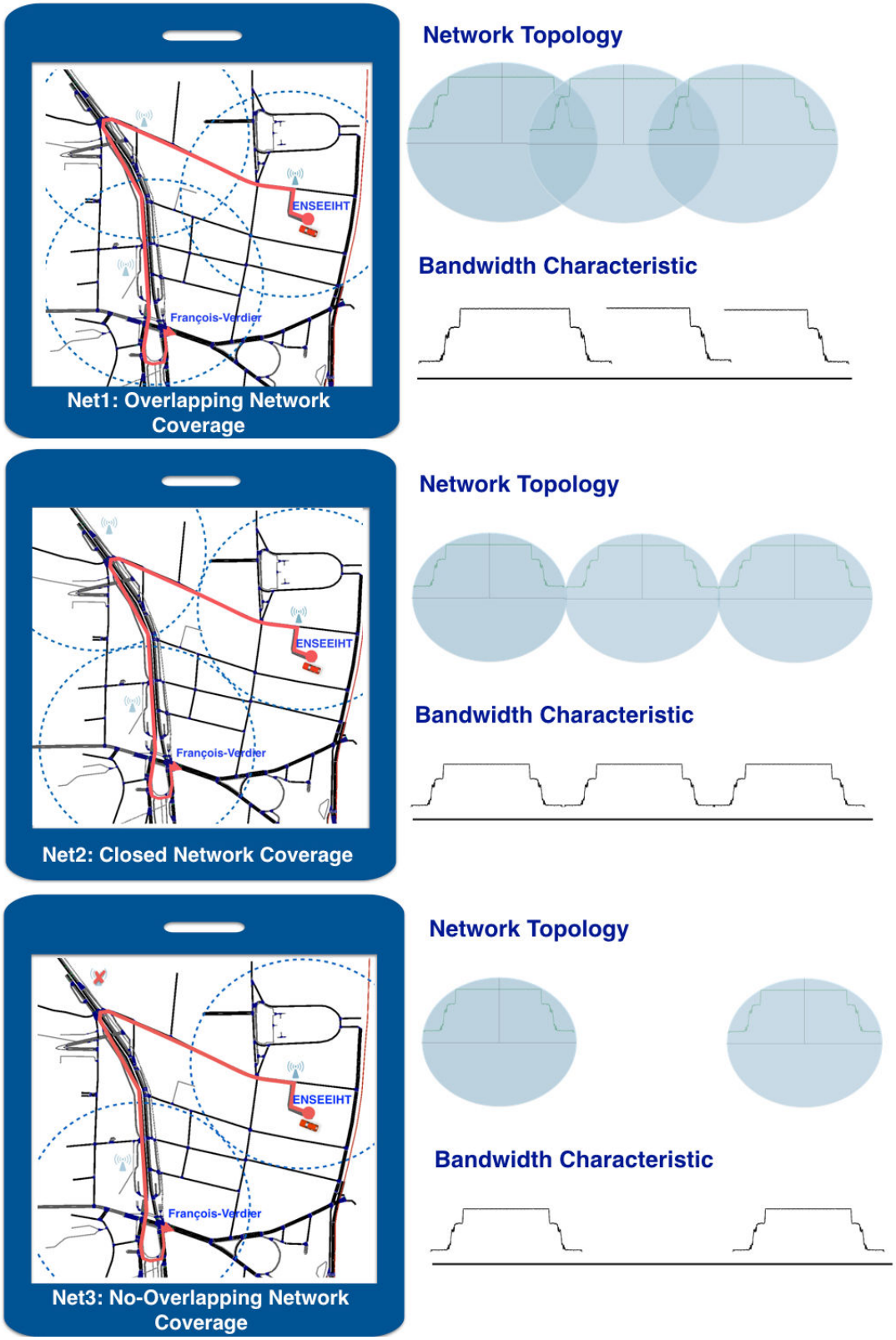

Figure 6.11 The experimental network topologies

ENSEEIHT to François-Verdier requesting a video with a constant data rate (1Mbps) during 200 seconds.

6.2.4 Performance Analysis

In this section, we focus on in-vehicle communication with the infotainment services such as live streaming video based on WiFi technology, while considering the two aspects: user and operator. For the user aspect, they always require to receive a good quality of video and to playback the video without interruption. So the quality of service regarding the loss and transmission delay will be considered for user aspect.

For operator aspect, they prefer to serve their users with the high user experience (UX) and also prefer to decrease the operating expense (Opex). To achieve this, we vary the distance between the network coverages (cells) to find the suitable range that can extend the network coverage by decreasing the number of RSUs.

The experimental network topology is simulated by NS3 that can be separated into two main topologies with each purpose (user and operator). Each connected vehicle will move depending on the mobility trace which is generated by Simulation of Urban Mobility (SUMO) [129].

6.2.4.1 User Quality of Service

The simulation has been done by setting the experimental topologies as shown in Figure 3. Each Net is tested with the same scenario that every 5 seconds, the connected vehicle departs from the same origin to same destination (ENSEEIHT to François-Verdier) and requests the same video with a video duration about 200 seconds and a constant data rate at 1Mbps. The experimental results show mainly two criteria: the number of loss and the transmission time.

- **The Number of Loss**

  Figure 6.12 illustrates the average percentage of packet loss through the varying number of vehicles in the vehicular network. For No-Cache policy, the number of packet loss is significantly increased when the number of connected vehicles is large. For ten connected vehicles with No-Cache policy, Net1, Net2, and Net3 suffer from the about 38%, 49%, and 65% loss. But the ON-OFF caching policy minimizes the loss in Net3 and prevents the loss in Net1 and Net2.

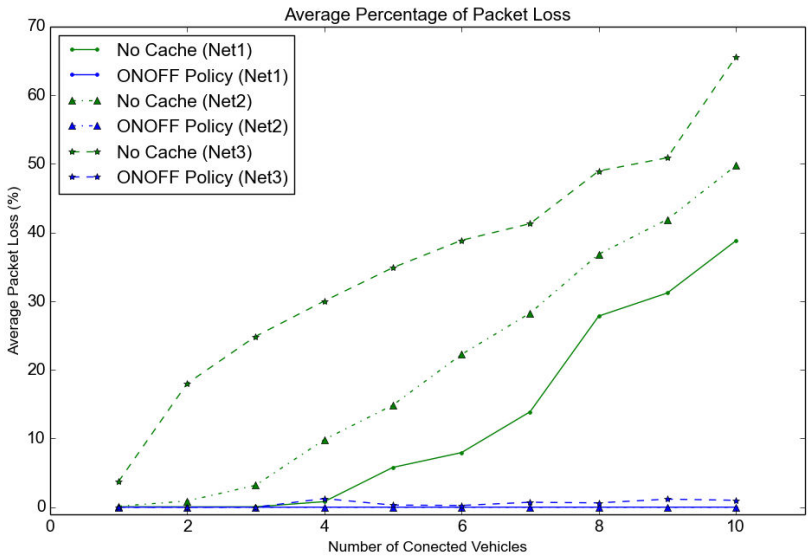

Figure 6.12 The average percentage of packet loss.

Reducing the number of loss gives better video quality. From the experimental results, we can conclude that ON-OFF caching policy provides the benefit with regarding loss, leading to the better user experience.

- **Transmission Time** The total transmission time is illustrated in Figure 6.13. The perspective results showed that the total transmission time of ON-OFF policy is highest than No-Cache policy about 43%, 21% and 14% in Net3, Net2, and Net1, respectively. The results are caused, as the ON-OFF policy, is forwarding the packets only in situations where the connected vehicle is living in the good signal area. Therefore the transmission time of the ON-OFF caching policy is depending on the network topology and moving pattern of the connected vehicle.

  Even though the ON-OFF policy minimizes the loss in all Nets, it gives the highest transmission time. This may pause the video playback. However, this problem can be addressed by configuring the display application to buffer the video before playback.

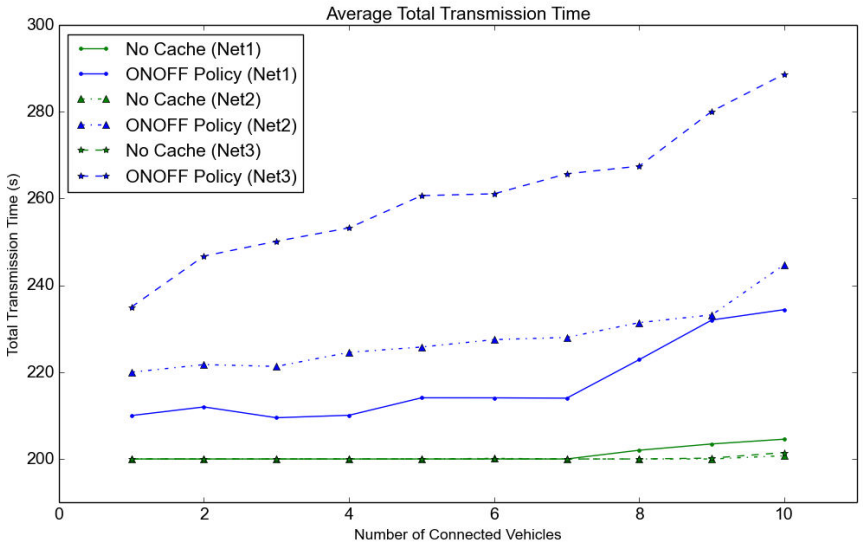

Figure 6.13 The average transmission time.

6.2.4.2 Operator Quality of Service

- **The number of packet loss**

  Figure 6.14 illustrates the average percentage of packet loss for different distances between the WiFi network coverages. The ON-OFF policy prevents the packet loss at all distances. The number of loss of No-Cache policy will be increased when the distance increases.

- **Transmission Time**

  The total transmission time through the varying of the distance between the WiFi network coverages is illustrated in Figure 6.15. The ON-OFF policy takes higher transmission time than No-Cache policy because of the ON-OFF policy caches the data during the interruption of the Internet connection for the connected vehicle and transfers them to the latter once connected again to a new RSU network. This limitation is due to the handover latency plus the total transmission time of the cached data.

  For instance, at a distance reaching 120 meters, the transmission time of ON-OFF policy is higher than No-Cache policy about 27% or 53 seconds.

  Considering the number of RSUs with the coverage range, for overlapping network, we can see that 3 RSUs are required for 800 meters. But when analyzing non-overlapping

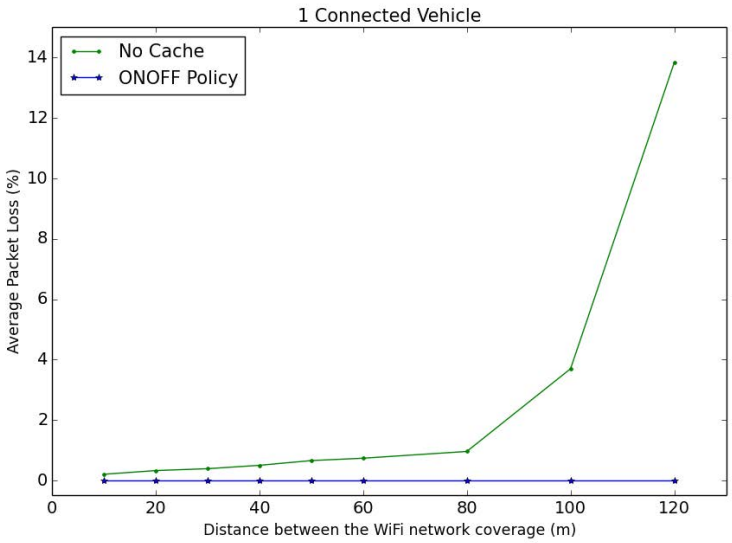

Figure 6.14 The percentage of packet loss through the varying of the distance between the WiFi network coverages.

configuration (with a gap of 120 meters), we can cover a range of 800 meters with only 2 RSUs. Consequently, the network coverage can be extended more than 34% compared to the overlapping network.

However, the problem of video interruption may not occur in this case that the playback application can be configured to buffer before playback video more than 53 seconds (VLC can be configured the to buffer up to 60 seconds).

6.2.5 Summary

This section has proposed an alternative approach to provide mobility management in a vehicular network. The proposed approach uses the ability of SDN to handle the vehicle mobility and to enable the cache function for reducing the packet loss. The ON-OFF caching policy has been proposed to apply in SDN vehicular network. The performance analysis has been measured regarding the percentage of loss and the total transmission time. From the obtained results, we can conclude that an ON-OFF caching policy can improve the quality of services regarding packet loss, but it is worse concerning total transmission time. But this weakness would not affect the user experience because the connected vehicles can be equipped with large storage. So the data can be held in the storage before is forwarded to the passengers in the vehicle for smooth video playback.

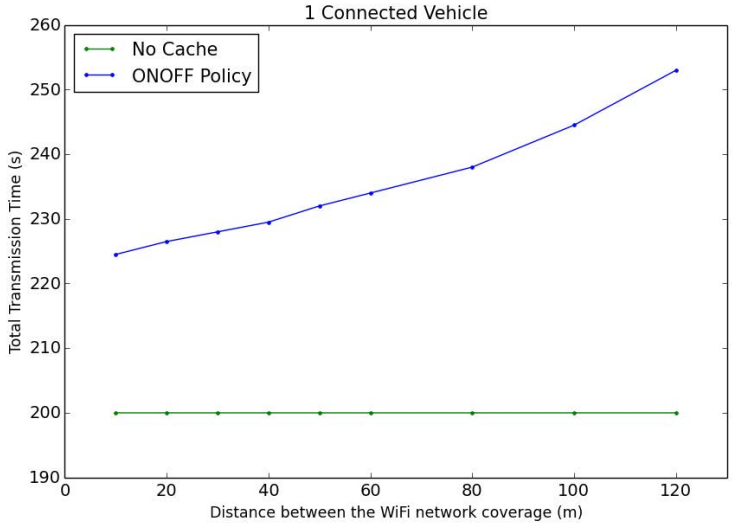

Figure 6.15 The total transmission time through the varying of the distance between the WiFi network coverages.

We have also considered the operator aspect; the results can be used to adjust the distance between RSUs. The operator can choose a suitable distance to locate the RSUs and can reduce the number of RSUs to decrease the operating cost without affecting the user experience.

6.3 Conclusion

We have evaluated by simulation the caching mechanism specified in the previous chapter. The evaluations are conducted in 2 different contexts: pedestrian mobility and vehicular mobility context.

In the pedestrian context, we evaluated two caching policies: ON-OFF and Adaptive. The obtained performance analysis indicates that two policies improve the results compared to a functioning without caching. The interest of each one must be understood from the applications which use them. Thus, the ON-OFF policy is more suited to non-time-sensitive applications while the Adaptive policy is suitable for time-sensitive applications.

In the vehicular context, we have introduced in the mobility management architecture, an anticipation mechanism relying on the predictability of the movement to prepare the handover. By predicting the new access point, the cached data is transferred, according to ON-OFF or Adaptive policy, to the new access point before the handover is effective. The obtained results are similar to those of the terrestrial context.

It can be concluded that Adaptive caching policy is suitable for the delay sensitive applications such as live streaming video. But the ON-OFF caching policy is appropriately with the delay non-sensitive applications such as file transfer and Video-on-Demand.

We achieved some proof of concept that the network software implementation possibly makes the proposed caching mechanisms. We have shown their interest, and further work would concern their exploitation, considering the caching function as an NFV deployable on-demand, depending upon the topology, in one or more instance functions of the caching policies and the traffic data rate.

To continue our study, we extend the network context to a heterogeneous environment. We have so far considered a homogeneous IP architecture, delivering content. Considering ICN architectures are a promising future alternative. In the next chapter, we interested in mobility management in heterogeneous IP and ICN architecture.

Chapter 7

Mobility Management in Heterogeneous Network Architectures

Contents

This chapter gives how to apply our SDN-Mobility with caching policy to be the alternate solutions to manage mobility services in heterogeneous networks with SDN. The first section, we introduce the heterogeneous network architectures and explain the mobility problems. After our designed architectures and approaches are presented.

7.1 Introduction

Information Centric Network (ICN) [8] [9] [130] is a proposed future Internet architecture that changes the network communication model from end-to-end packet delivery (IP networks)

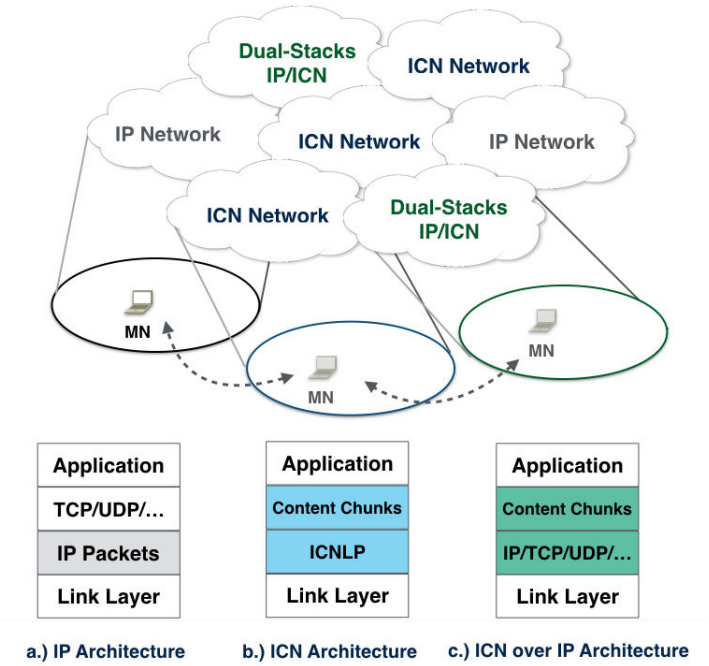

Figure 7.1 Overall Internet during Transition Period and Network Architectures

to named data retrieval. ICN architecture supports the location-independent content name, in-network caching and name-based routing. The MNs send the interests to fetch data with the name. Many approaches were proposed for ICN architectures such as Data-Oriented Network Architecture (DONA) [131], Publish-Subscribe Internet Technology (PURSUIT) [132] [133] [134], Network of Information (NetInf) [135], Content-Centric Networking (CCN) [136] and Named Data Network (NDN) [137]. All of these approaches were designed to the same purposes that provide access to content and services with name and independence of original content location but differ regarding implementation.

The migration of IP to ICN cannot happen overnight. Instead, there will be a period of transition when both network architectures are in use over the same infrastructure on the Internet. The transition is not easy to accomplish because of incompatibility of IP and ICN architecture. An IP architecture uses IP addresses to identify where the data is located but ICN named to communicate for what is data they want. The main building blocks of the ICN architecture are named content chunks, in contrast to the IP architecture's fundamental unit of communication, which is an end-to-end channel between two end endpoints identified by IP addresses as shown in Figure 7.1. Thus, the existing IP architecture must be extended to support ICN. Also, current software applications and the routing systems need to be upgraded to support ICN. Because the traditional IP applications data structure embed IP addresses and the routing systems are IP-based forwarding.

The ICNs are being deployed in many networks for content delivery. There are two feasible ways to deploy ICN architecture. The first way, ICN is deployed over the legacy IP networks. A few works have proposed the solutions to deploy ICN architecture in current IP architecture and vice versa. M. Vahlenkamp et al [138] proposed a mechanism to deploy ICN protocols in IP networks via the assistance of the SDN by packet header rewriting in combination with a single IP prefix to initially contact the ICN network.

The other way is a native ICN architecture that can be run directly on Ethernet links without having to configure IP addresses and without the extra layer of processing, leading to minimizing overhead. This architecture requires the fragmentation of ICN messages to packets in case ICN messages size is larger than Maximum Transfer Units (MTU) such as A Link Protocol for NDN (NDNLP) [139]. There are a few works in cellular context proposed to deploy native ICN architecture. [140] and [141] proposed to embedded native ICN into 3GPP protocol stack for optimizing 4G/LTE mobile networks.

7.2 Mobility Problems in Heterogeneous Network Architectures

During the transition period, the network architecture on the Internet will be a mixture that may support native IP networks, dual IP and ICN networks, or native ICN networks. This heterogeneous network architecture does not effect on fixed node communication. Because a fixed node uses either IP stack or ICN stack depending on the used existing network for communicating with the destination, with the same network stack.

With the ability to move, the mobile devices can move anywhere. It is possible that while an IP mobile node is receiving the content data, it moves from IP networks to future networks (ICN), or vice versa. The effect of this is that when an IP mobile node has changed its point of attachment, it does not obtain a new IP address from the ICN network, leading to its routing path cannot update. This makes packets to its old address will be sent to its old IP network connection point by the routing system, leading to mobile node's connection loss after the handover. On the other hand, when an ICN mobile node handover to the IP network, it expresses an interest packet for requesting the content data. But IP network devices do not understand the ICN packet format. These interest packets will be discarded from the attached access point, leading to the ICN connection loss after handover into the IP network.

These handover in heterogeneous network architectures affect mobile communications that the mobile connectivity will be broken.

To maintain the connectivity of mobile nodes in the heterogeneous network architectures, a possible mobile management approach is required. Therefore in this chapter, we propose a solution to solve this problem. The proposed solution is used to enable mobility support for mobility while the mobile nodes are communicating and to permit handover into a different network architecture. We apply our SDN-Mobility with caching policy which has presented in Chapter 5 and Chapter 6 to provide mobility services in heterogeneous network architectures.

7.3 Mobility Management Approach in Heterogeneous Network Architectures

Three different network architectures may be present during the deployment of ICN: native IP networks, dual stacks networks (IP and ICN), and native ICN networks. The mobile nodes possibly move into any networks.

Considering a mobile node moving from IP/ICN networks to dual-stacks networks, the mobile node can continue its connectivity in a new network with the same network stack like as IP to IP and ICN to ICN. This situation does not have any problem due to the SDN-Mobility approach can be used to provide mobility in IP to IP and IP to dual-stacks networks situation. Furthermore, the mobility natively supported for ICN to ICN and ICN to dual-stacks networks situation.

For the communications between the different network architectures, they cannot communicate directly. So in this chapter, we focus on the case where mobile nodes handover from native IP networks to native ICN networks and vice versa.

7.3.1 Mobility Management in Heterogeneous Architecture

The general architecture of mobility management in heterogeneous networks is illustrated in Figure 7.2. For ICN networks, all devices support ICN stack and are able to cache the data. The network devices in IP networks support SDN protocol.

To provide the mobility services in both situations (ICN-to-IP and IP-to-ICN), we proposed to locate the SDN controller at the border network for managing the mobility signaling in heterogeneous networks. The border router requires to support dual-stacks. For ICN network devices, at least the access points in ICN networks require to support SDN protocol.

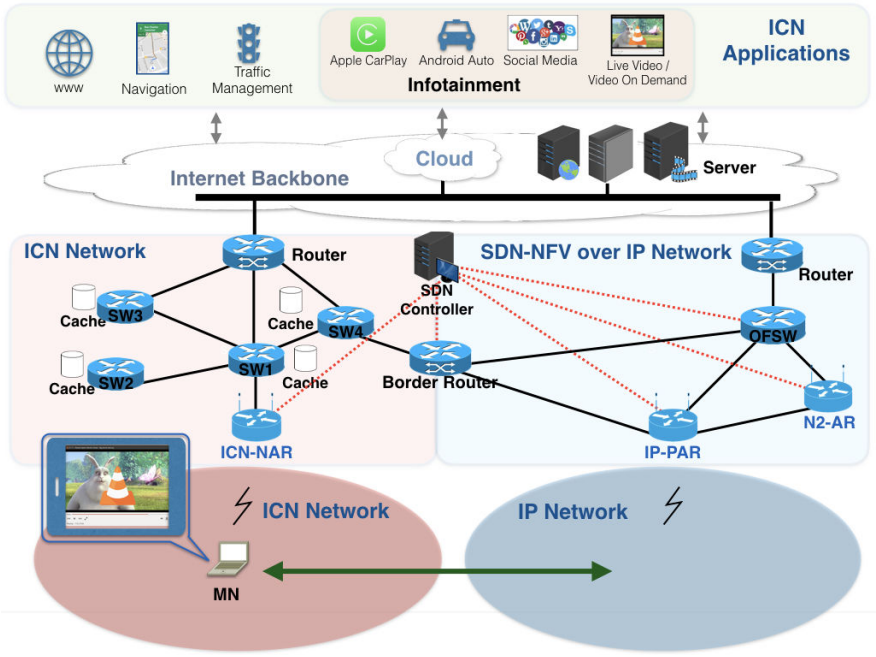

Figure 7.2 Mobility Management in Heterogeneous Architecture

The general responsibilities of the SDN controller are to detect the mobile handover event and to manage the mobile services. The operation will vary depending on the situation where the mobile nodes move from and to.

In ICN-to-IP situation, the SDN controller sends an OpenFlow message to the border router for requesting and caching the mobile data from ICN networks on behalf of the mobile node.

For IP-to-ICN situation, the SDN controller operates depending on the mobile node network stack. If the mobile node supports only IP stack, SDN controller has to manage the border router to caching the mobile data and to establish ICN tunnel to the access point with which mobile is attached. In case the mobile node supports dual-stacks, SDN controller is responsible for generating the name of the mobile node and managing the border router to cache and forward the mobile data.

7.3.2 Mobility Management in Heterogeneous Operation

The mobility management operation in heterogeneous networks can be classified in two situations: ICN-to-IP and IP-to-ICN.

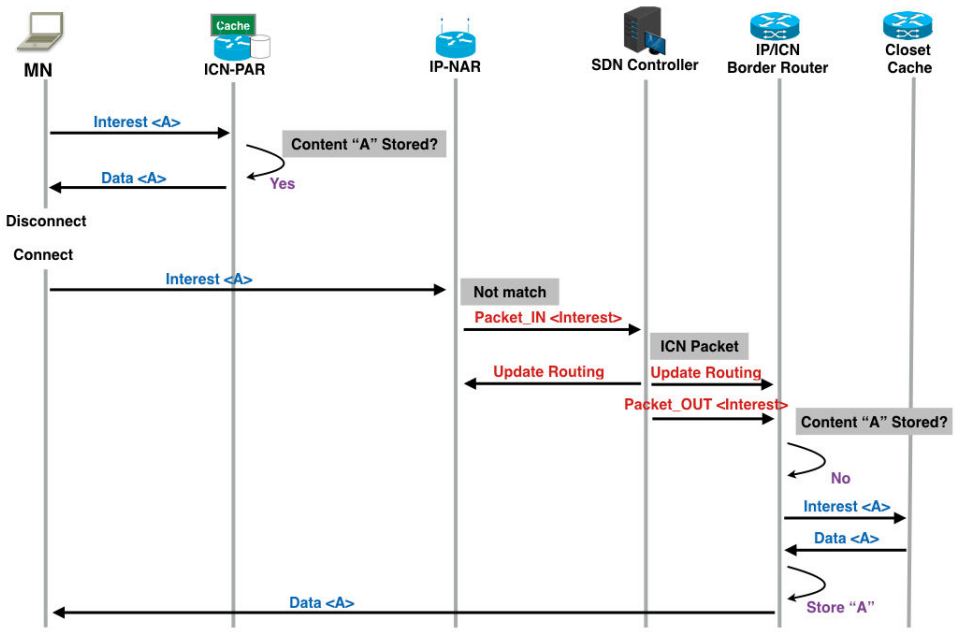

Figure 7.3 ICN-to-IP Mobility Management Operation

7.3.2.1 ICN-to-IP situation

As illustrated in Figure 7.3, the first mobile node lives in ICN network. The mobile node requests the chunk data with the interest packet. When the mobile node handover to the native IP network, it still sends an interest packet for requesting the chunk data. The access point (IP-NAR) receives an interest packet and processes for forwarding. But it does not know this packet, then it copies this interest packet and sends to SDN controller. This message makes the SDN controller knowing the mobile node has moved from the ICN network. So the SDN controller updates the routing path for the mobile node and sends an OpenFlow message out for managing the IP/ICN border router to request the chunk data on behalf of the mobile node. The border router firstly finds the requested chunk in its cache. If that chunk data does not have, the border router will request to the other ICN devices and will cache that chunk data before forwards to the mobile node directly. In case the border router already has the requested chunk data in its cache place, it sends the requested chunk data to the mobile node without requesting from the other places.

7.3.2.2 IP-to-ICN situation

For the mobility management operation in case the mobile node moves from IP to ICN networks, we consider in two sub-cases depending on the mobile node network stack.

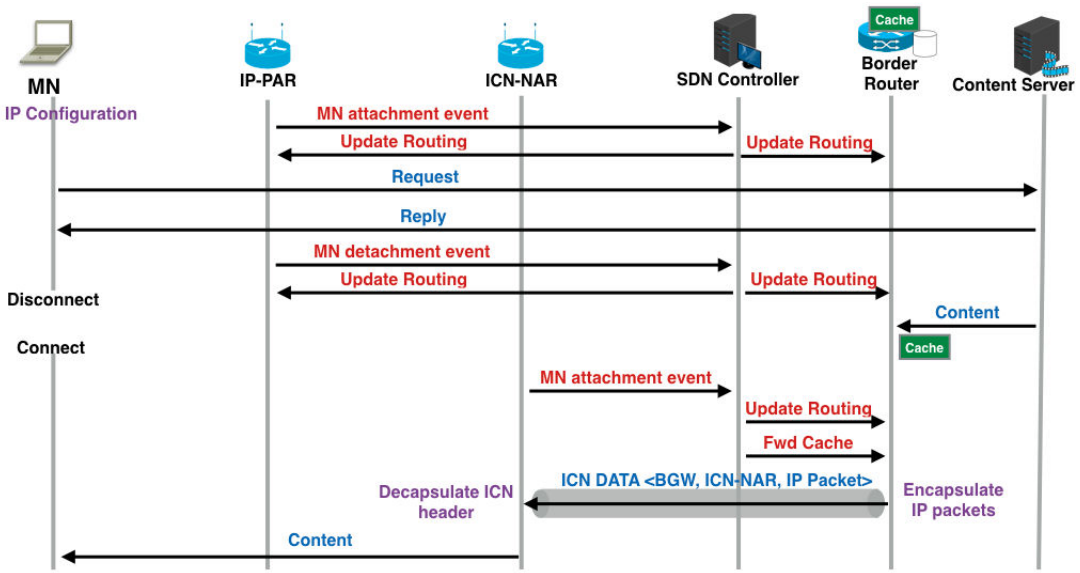

Figure 7.4 IP-to-ICN Mobility Management Operation: IP Mobile Node and only access points support SDN.

- **Mobile Node supports only IP stack**

  In case the mobile node supports only IP stack, the mobile node requests the data from the content server with normal IP packets. When the mobile node detaches from the access point (IP-PAR), the SDN controller updates the routing path and manages the border router to cache the requested data of the mobile node.

  Then, the mobile node attaches at the new access point (ICN-NAR) in the ICN network, the operation can be performed in two cases depending on the network devices in ICN support SDN.

  The first case is only access points in ICN network and the border router support SDN as illustrated in Figure 7.4. When the mobile node attaches at the new access point (ICN-NAR) in ICN network, an ICN-NAR informs the MN attached event to SDN controller. After that, the routing path of the mobile node will be updated through OpenFlow message from the SDN controller. The SDN controller also manages the border router to forward the cached data to the mobile node through the tunnel. The tunnel will be established between the border router and ICN-NAR. Thus, the mobile data (entire IP packet) is encapsulated at the border router and is decapsulated at ICN-NAR before is forwarded an original IP packet to the mobile node.

  The second case is all network devices in ICN support SDN as illustrated in Figure 7.5. When the mobile node attached at the ICN-NAR in ICN network, an ICN-NAR

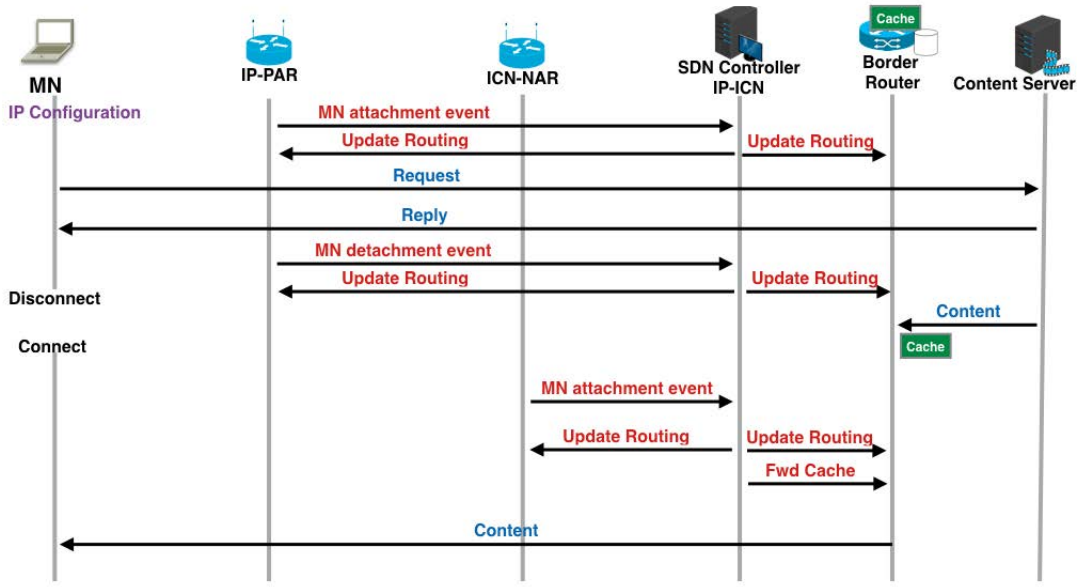

Figure 7.5 IP-to-ICN Mobility Management Operation: IP Mobile Node and all network devices support SDN.

informs the mobile attachment event to SDN controller. Then, SDN controller updates the routing path at all network devices in the ICN network and sends an OpenFlow packet to manage the border router for forwarding the cached data to the mobile node directly via IP packets.

- **Dual-Stacks Mobile Node**

  When the dual-stack mobile node entrances in IP networks, the mobile node automatically uses IP stack to communicate with normal IP packets. The requested data will be cached at the border router after the mobile node detached from the access point (IP-PAR).

  Then, the mobile node changes its attachment to the ICN network, the current access point (ICN-NAR) informs the mobile attachment event to the SDN controller. The SDN controller gives the name to the mobile node. Note that, the ICN stack will be used automatically. Then, the routing path of the mobile node is updated. After the border router received the Fwd_Cache command from the SDN controller, the border router forwards the cached data to the mobile node via ICN messages.

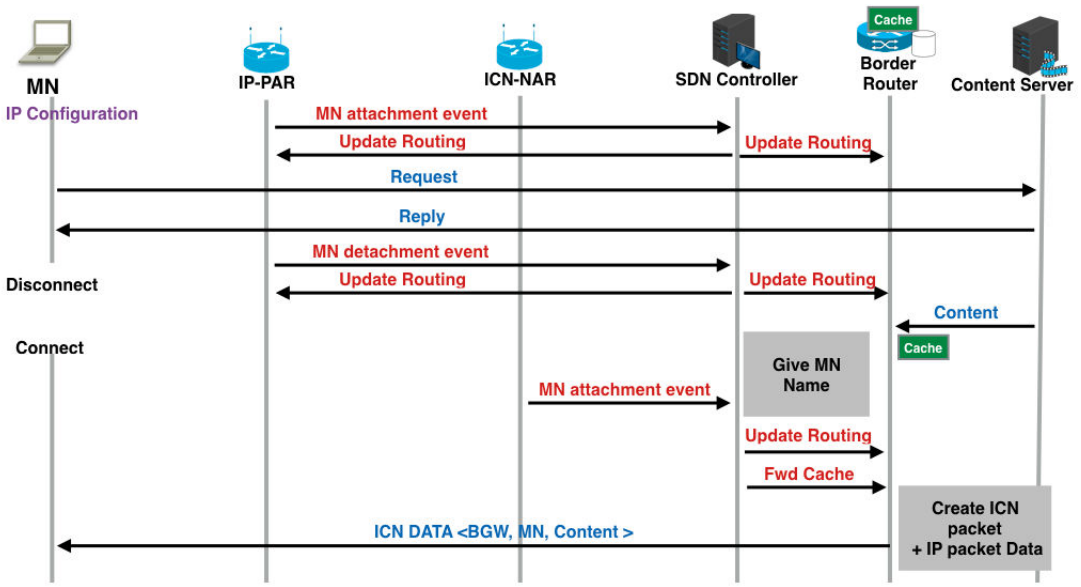

Figure 7.6 IP-to-ICN Mobility Management Operation: Dual-stacks Mobile Node.

7.4 Discussion

The summary of each approach for mobility management in the heterogeneous network is presented in TABLE 7.1. The ICN mobile nodes can still communicate with the others when it moves into the different IP networks via original ICN packets by upgrading the border router to support both IP and ICN stacks. This also uses the ability of SDN-Mobility to update the routing path and uses the advantage of SDN to manage the border router for requesting the mobile data from ICN networks. The ICN-to-IP situation still gains the benefit of network caching in ICN networks.

In IP-to-ICN situation, the tunneling overhead can be avoided in case of all network devices in ICN network support SDN. The original IP packets can be forwarded to the mobile node directly through dynamically update the routing path with SDN-Mobility. The tunneling overhead also can be avoided in case of the mobile node network stack, and applications support both IP and ICN. In this case, the SDN controller requires more function to mapping the name and IP address of the mobile node, and the border router requires the operation for generating ICN packets.

TABLE 7.1 Summary of Mobility Management Approaches in Heterogeneous Networks

Handover	**MN Stack**	**After Handover**		**Requirements**
		via Packets	Tunnel	
ICN-to-IP	ICN or Dual-Stacks	ICN	No	- Border router supports dual-stacks.
IP-to-ICN	IP	ICN	Yes	- Border router supports dual-stacks. - Only ICN-NARs support SDN.
	IP	IP	No	- Border router supports dual-stacks. - All ICN network devices support SDN.
	Dual-Stacks	ICN	No	- Border router supports dual-stacks. - The applications support dual-stacks.

7.5 Conclusion

We have extended the study of mobility to a heterogeneous architecture with IP and ICN networks. We also proposed conceptual management approaches in several situations according to the mobile protocol stack, IP or ICN, and the integration level of the SDN, on a single access point or overall access points. Our mobility management proposal inspires the conceptual approach.

We have shown that SDN-Mobility with caching policy can be applied for providing mobility services in heterogeneous network architecture. The proposed approaches still inherit the distinctive points of SDN-Mobility that are automatic routing update, without the legacy IP mobility protocol implementation, and no need of tunnel to forward the data in case of network devices support SDN protocol. In addition, it gains the benefit of network caching in ICN for requesting the closest content data, even if the handover of the mobile node into the IP network.

Chapter 8

Conclusion and Perspectives

Contents

8.1 Conclusion

This thesis began as an examination of Mobility IPv6 and Software-Defined Network (SDN) issues, hoping to find the ways to improve the mobility management in SDN network. During that examination, it was noticed that we can use the SDN concept to manage the mobility operation without mobility protocol implementation. We have proposed SDN-Mobility to replace the existing IP mobility protocol, called PMIPv6. Their performance has been measured in terms of UDP throughput, TCP sequence and percentage of packet loss. From the results analysis, we conclude that the SDN-Mobility can be used for mobility management like as PMIPv6, but without the legacy IP mobility protocol implementation. Moreover, as SDN routing is directly managed by the centralized controller, the well-known direct routing problem is resolved, there is no need of tunnel to forward the data in case of localization change (only need to change the routing).

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

packet overhead, decreases the handover latency time, and mitigates the percentage of packet loss.

Based on the impact of packet loss on the perceived video quality through VQMT tool in Chapter 5, we proposed to minimize the number of loss with caching scheme. We have proposed two caching policies: Adaptive and ON-OFF caching policy. Their performance analysis has been measured regarding the percentage of packet loss, transmission time, channel occupancy time and bandwidth fairness. According to the obtained results, we can conclude that both proposed cache policies can improve SDN-Mobility regarding packet loss. Even though, the ON-OFF caching policy is worse concerning transmission time but good regarding the channel occupancy time. Then, the ON-OFF caching policy is suitable to be used with the applications which are non-sensitive to the delay such as file transfer or some media applications which download all contents before display. For delay-sensitive applications, an Adaptive caching policy is properly used. Even though, the transmission time of the Adaptive caching policy is a bit higher than No-cache policy. But this weakness will not affect actual usage because the general media display applications buffer the video before display. It would be interesting to evaluate our proposed schemes and compare them to the caching mechanism proposed by adaptive HTTP that will be described in the Perspectives section.

In this thesis, we also proposed an alternative approach to provide mobility management in a vehicular network which is extended SDN-Mobility with caching policy. The ON-OFF caching policy has been chosen for caching MN's data. The ability of SDN is used to handle the vehicle mobility, to enable the cache function, and to upload the cached data to the new network before the handover. The performance analysis has been measured regarding the percentage of loss and the total transmission time. From the obtained results, we conclude that an ON-OFF caching policy improves the quality of services regarding packet loss, but it is worse concerning total transmission time. But this weakness will not affect the user experience because the connected vehicles can be equipped with large storage. So the data can be held in the storage before is forwarded to the passengers in the vehicle for smooth video playback. We have also considered the operator aspect; the results can be used to adjust the distance between RSUs. The operator can choose a suitable distance to locate the RSUs and can reduce the number of RSUs to decrease the operating cost without affecting the user experience.

In conclusion, we showed the feasibility and the interest of the SDN technology in mobility management, which simplifies the signaling compared to the solutions of the IP mobility type but also compared to the cellular solutions of the LTE type. SDN also opens the door to more dynamic management, with adaptive policies such as caching that we proposed.

8.2 Perspectives

This thesis proposed SDN-Mobility that can enable mobility for MN in the SDN network and also obviously shown the packet loss problem during MN handover. This thesis also proposed a new approach to improve this loss problem with caching policy. It would be interesting to know what happens if the mobile nodes request the video from adaptive HTTP server (DASH) in SDN-Mobility with caching policy network. In this case, we can consider that DASH is one application which mobile node uses. It just changes the way to request and receive video data. In regular operation of DASH, the mobile nodes request the video via HTTP and the video data will be transmitted over TCP. So if packet loss, it will retransmit. Then, in this case, the loss does not affect the perceived video quality. But for smooth playback, the requested quality of the video will be changed depending on the currently available bandwidth and network conditions. When the handover of the mobile node to the other attachment, our proposed approach (SDN-Mobility with caching policy) still works without affectation in case of handover latency less than TCP timeout value. For example, in case an Adaptive caching policy is applied, the previous access point will cache the requested video and will transfer to the mobile node after handover. During mobile node handover, it continues playback video from the retained cache. So if handover latency is large, the cache is not sufficiency for playback. The video will freeze and continue playback after the handover. But if the retained cache is enough, the video is smoothly played back.

We have a few suggestions for improving the strategy of adaptive requested video bitrate in MPEG-DASH with the ability of SDN. From our study in WiFi characteristic (Chapter 4), when the mobile node moves far away from the access point, its capable bandwidth is less. This can imply that the requested video bitrate of DASH mobile node before its handover will be adapted to lowest bitrate. The current DASH client strategy, after DASH mobile node handover, it starts to request the next video segment with the previous bitrate before its disconnect (low bitrate), whether the mobile node handover into a good signal area or a weak signal area. This is an interesting point which we suggest to improve with SDN for increasing quality of experience.

Appendix A

Technological Background

A.1 Basic PMIPv6 Operation Messages

- **Solicitation (RS) Message**

  Router Solicitation (RS) is ICMP router message which can be used to request routers to generate Router Advertisement message immediately.

- **Advertisement (RA) Message**

  Router Advertisement (RA) is ICMP router message which is periodically multicasted by a router or sent in response to a Router Solicitation message from a host.

- **Binding Update (PBU)**

  Proxy Binding Update (PBU) is a PMIPv6 signaling packet sent by MAG to LMA to indicate a new mobile attachment. Main fields are: MN-ID, MAG address and handover indicator to signal if the MN-attachment is a new one or handover from previous point.

- **Proxy Binding Acknowledgement (PBA)**

  Proxy Binding Acknowledgement acknowledge PBU messages sent by LMA to MAG. PBA contains the MN-ID, the MAG address and the prefix assigned to the mobile.

TABLE A.1 The existing mobility protocols classification.

Approaches	**Protocols**	**Strategy**
Routing-based	Columbia [14], 1991 Connexion [16], 2004	Broadcast-based
	Mobility Support Using Multicast in IP (MSM-IP) [142], 1997 Cellular IP [17], 1999 Handoff-Aware Wireless Access Internet Terminal Independent Mobile IP (TIMIP) [19], 2001 Infrastructure (HAWAII) [18], 2002	Host-based Path
Mapping-based	Virtual Internet Protocol (VIP) [143], 1991 Loose Source Routing (LSR) [144], 1993 End-to-End (E2E) [145], 2000 Mobile Stream Control Transmission Protocol (M-SCTP) [146], 2002 Mobile IPv4 (MIPv4) [1], 2002 Mobile IPv6 (MIPv6) [2], 2004 Network Mobility (NEMO) [147], 2005 GLOBAL HAHA [148], 2006 Identifier-Locator Network Protocol (ILNPv6) [149], 2007 Hierarchical Mobile IP (HMIP) [4], 2008 Host Identify Protocol (HIP) [150], 2008 Fast Handover for Mobile IPv6 (FMIPv6) [3], 2009 Back to My Mac (BTMM) [151], 2009 LISP-Mobility [15], 2013	Host-based
	Proxy Mobile IP (PMIP) [5], 2008	Network-based
Combination	Wide-Area IP Network Mobility (WINMO) [152], 2008	Host-based

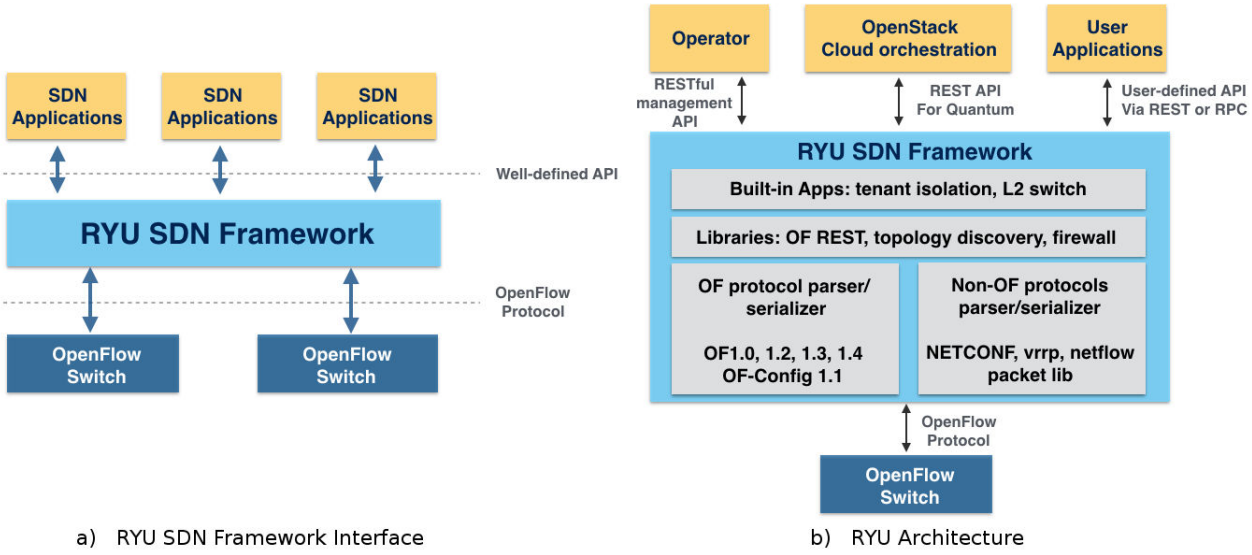

Figure A.1 RYU Architecture

A.2 Overview of the Mobility Approaches

A.2.1 RYU SDN Framework

A.3 Experiments

TABLE A.2 shows tools and programming language of each experiment in this thesis that can be classified into two main network topologies: wired and WiFi. Mininet version 2.1.0p2 is used to generate wired network topology before run PMIPv6 or SDN-Mobility for providing mobility to nodes. Because a regular Mininet uses for generating OpenFlow switches in the wired network, it doesn't have routers and mobility function. Thus, we generated the routers by the normal hosts in Mininet and configure them to be Linux routers. We also created the attach/detach function for hard handover of nodes. RYU SDN controller runs the simple_switch_13_my.py which is extended from example code (simple_switch_13.py) to update the routing paths and support IPv6 traffic. These source codes are written in Python language.

For SDN-Mobility in WiFi network, we used OpenNet to generate the wired network with Mininet library and generate wireless access point with an NS3 library. Even through OpenNet uses Python language, but the way to generate topology is quite different from the general Mininet (previous code). This because of their programming syntax different.

TABLE A.2 Overall Tools and Programming of Experiments Implementation.

Experiments	Emulator/ Simulator	Programming Language	Additional Methods/ Classes
Wired			
PMIPv6	Mininet 2.1.0p2 PMIPv6-v0.4.1	Python	- detach/attach SW function
SDN-Mobility	Mininet 2.1.0p2	Python	- detach/attach SW function - routing update function
WiFi			
SDN-Mobility	OpenNet (Mininet+NS3)	Python	- routing update function
Caching Policy			
- Pedestrian	NS3 v3.22	C++	- CacheApWifiMac
- Vehicular	NS3 v3.22 and SUMO v0.30.0	C++	- CacheApWifiMac - read mobility trace

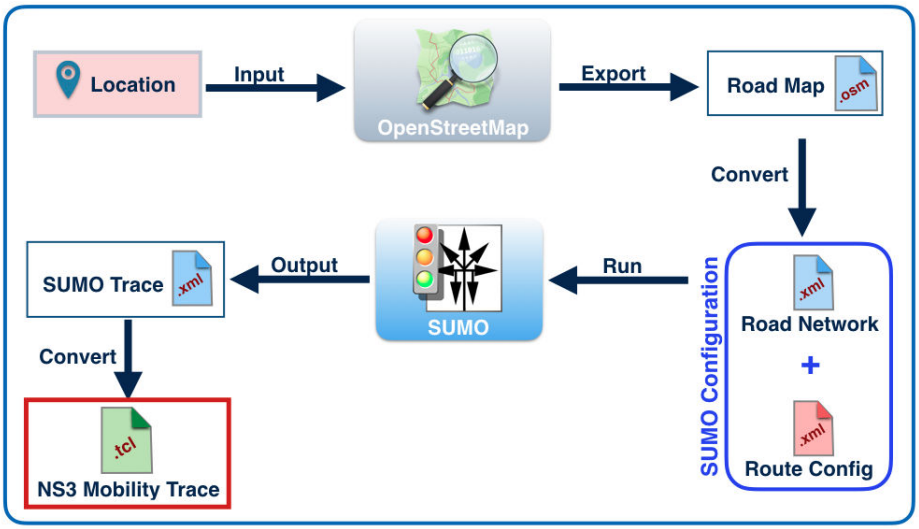

Figure A.2 Creating Process of NS3 Mobility Trace

We created a new class "CacheApWifiMac" which inherits from class "ApWifiMac" in wifi-module of NS3 to able the access points to cache/buffer packets when CacheTrigger is enabled. These source codes are written in C++ language about 800 lines. Class CacheApWifiMac is used in both pedestrian and vehicular experiment, but in vehicular it cooperates with NS3 mobility trace. The creating process of NS3 mobility trace shows in Figure A.2.

A.3.1 Source Codes

Listing A.1 PMIPv6/SDN-Mobilily Hard Handover Source Code (Mininet)

```
#!/usr/bin/python

"""
Script modified by KULJAREE - Visual Network Description (SDN version)
It setup the network topology. Also there is a mobility function for moving
    Node

We move a host from s1 (MAG1) to s2 (MAG2),
and then back to s1.
Gotchas:
1. The interfaces are not renamed; this
   means that s1-eth1 will show up on other
   switches.
2. The reference controller doesn't support
   mobility, so we need to flush the switch
   flow tables.
3. The port numbers reported by the switch
   may not match the actual OpenFlow port
   numbers used by OVS.
Good luck!

"""
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSSwitch,
    OVSKernelSwitch, OVSLegacyKernelSwitch, UserSwitch
from mininet.topo import LinearTopo
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink
from mininet.util import dumpNetConnections
from time import sleep
```

```
import os

def printConnections( switches ):
      "Compactly print connected nodes to each switch"
      for sw in switches:
         print '%s:' % sw,
         for intf in sw.intfList():
                 link = intf.link
                 if link:
                    intfs = [ link.intf1, link.intf2 ]
                    if intfs[ 0 ].node != sw:
                          intfs.reverse()
                    local, remote = intfs
                    print remote.node,
         print

class RYUBridge( Controller ):
   "Custom Controller class to invoke my RYU simple_switch_13"
   def start( self ):
      "Start RYU learning switch"
      self.ryu = '%s/ryu/bin/ryu-manager' % os.environ[ 'HOME' ]
      self.cmd( self.ryu, 'simple_switch_13_my.py &' )
   def stop( self ):
      "Stop RYU"
      self.cmd( 'kill %' + self.ryu )

controllers = { 'ryubridge': RYUBridge}

def topology():
      "Create a network."
      net = Mininet( controller=RYUBridge, link=TCLink,
          switch=MobilitySwitch )

   print " --------------- KULJAREE TEST --------"

   print " *** Create Controller ***"
   #Add Controller
   c1 = net.addController( 'c1', controller=RYUBridge, ip='::1', port=6633 )

      print " **** Create Switchs **** "
   #Add Switchs
```

```
   s1 = net.addSwitch( 's1',switch=UserSwitch, mac='00:00:00:11:00:00' )
s2 = net.addSwitch( 's2', switch=UserSwitch, mac='00:00:00:22:00:00' )
s3 = net.addSwitch( 's3', switch=UserSwitch, mac='00:00:00:33:00:00' )

# Create Linux Router
LMA1 = net.addHost( 'LMA1', mac='00:00:00:00:22:00')
MAG1 = net.addHost( 'MAG1', mac='00:00:00:00:33:00')
MAG2 = net.addHost( 'MAG2', mac='00:00:00:00:44:00')

print " **** Create Hosts **** "
#Add Hosts
MN = net.addHost('MN', mac = '00:00:00:00:00:11')
CN = net.addHost('CN', mac = '00:00:00:00:00:22')
h2 = net.addHost('h2', mac = '00:00:00:00:00:33')

#Create Links
   print "*** Creating links"
   net.addLink( s1, LMA1, 1, 0, bw=10 )
   net.addLink( s1, MAG1, 2, 0, bw=10 )
net.addLink( s1, MAG2, 3, 0, bw=10 )
net.addLink( LMA1, CN, 1, 0, bw=10 )
net.addLink( MAG1, s2, 1, 1, bw=10 )
net.addLink( s2, MN, 2, 0, bw=10 )
net.addLink( MAG2, s3, 1, 1, bw=10 )
net.addLink( s3, h2, 2, 0, bw=10 )

   print "*** Starting network"
net.build()
   s1.start( [c1] )
s2.start( [c1] )
s3.start( [c1] )
   c1.start()

#Fix  MAG MAC Address
print MAG1.cmd( 'ifconfig MAG1-eth1 hw ether 00:11:22:33:44:55')

#config IPv6 Address
print LMA1.cmd( 'ifconfig LMA1-eth0 inet6 add 2001:100::1/64' )
print LMA1.cmd( 'ifconfig LMA1-eth1 inet6 add 2001:2::1/64' )
print MAG1.cmd( 'ifconfig MAG1-eth0 inet6 add 2001:100::2/64' )
print MAG1.cmd( 'ifconfig MAG1-eth1 inet6 add 2001:1::1/64' )
print MAG2.cmd( 'ifconfig MAG2-eth0 inet6 add 2001:100::3/64' )
```

```
print MAG2.cmd( 'ifconfig MAG2-eth1 inet6 add 2001:1::2/64' )
print CN.cmd( 'ifconfig CN-eth0 inet6 add 2001:2::2/64' )

print CN.cmd( 'ip -6 route add ::/0 via 2001:2::1' )
print LMA1.cmd( 'sysctl -w net.ipv6.conf.all.forwarding=1' )
print MAG1.cmd( 'sysctl -w net.ipv6.conf.all.forwarding=1' )
print MAG2.cmd( 'sysctl -w net.ipv6.conf.all.forwarding=1' )

# Add rounting
print LMA1.cmd( 'ip -6 route add ::/0 via 2001:100::2' )
print MAG1.cmd( 'ip -6 route add ::/0 via 2001:100::1' )
print MAG2.cmd( 'ip -6 route add ::/0 via 2001:100::1' )

print
print '* Starting network:'
printConnections( net.switches )
print '*Identifying switch interface for MN'
MN, s2 = net.get( 'MN', 's2' )

hintf, sintf = MN.connectionsTo( s2 )[0]
last = s2

print "*** Running CLI"
   CLI( net ) #for start PMIPv6 Service /SDN-Mobility Service

# Simple test of mobility
print

for s in 3, 2, 3, 2, 3: #sw2-->sw3-->sw2

        next = net['s%d' % s ]
      print '* Moving', sintf, 'from', last, 'to', next
      last.detach( sintf )
      last.delIntf( sintf )
      next.attach( sintf )
      next.addIntf( sintf )
      sintf.node = next
      print '* Clearing out old flows'
      for sw in net.switches:
              sw.dpctl( 'del-flows' )
      print '* New network:'
```

```
      print MN.cmd( 'ifconfig MN-eth0 down' )
      print MN.cmd( 'ifconfig MN-eth0 up' )

         printConnections( net.switches )
      last = next
      #end moving code
      CLI ( net )

   CLI (net)
      print "*** Stopping network"
      net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    topology()

topos = { 'mytopo': ( lambda: topology() ) }
```

Listing A.2 SDN-Mobilily in WiFi network source code (OpenNet)

```
#!/usr/bin/python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
               CN
                |
                | 100 Mbps
                |
               R1
                |
               OFSW
                |
        -------------------
        |                 |
     (OFSW+AP)         (OFSW+AP)

[MN]
"""
import sys
import os
```

```
import mininet.net
import mininet.node
import mininet.cli
import mininet.log
import mininet.ns3

from mininet.net import Mininet, MininetWithControlNet
from mininet.node import Controller, RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import Link, TCLink
from mininet.ns3 import *

import ns.core
import ns.network
import ns.wifi
import ns.csma
import ns.wimax
import ns.uan
import ns.netanim
import ns.point_to_point
import ns.stats

from mininet.opennet import *

class RYUBridge( Controller ):
        "Custom Controller class to invoke RYU"
        def start( self ):
                "Start RYU learning switch"
                self.cmd( 'cd /home/kuljaree/ryu/bin/ && ./ryu-manager
                    --verbose
                    /home/kuljaree/ryu/ryu/app/simple_switch_13_my.py &' )
        def stop( self ):
                "Stop RYU"
                self.cmd( 'kill %' + self.ryu )

controllers = { 'ryubridge': RYUBridge}

MNxStart = 0.0     # x position of MN
MNvStart = 1.0     # velocity of MN
```

```
nodes = [ { 'name': 'MN', 'type': 'host', 'mac': '00:00:00:00:00:11','ip':
    '10.10.10.2', 'position': (MNxStart, 160.0, 0.0), 'velocity': (MNvStart,
    0, 0) },
          { 'name': 's1', 'type': 'switch', 'position': (100.0, 150.0, 0.0) },
          { 'name': 's2', 'type': 'switch', 'position': (0.0, 160.0, 0.0) },
          { 'name': 's3', 'type': 'switch', 'position': (200.0, 160.0 , 0.0)
              },
          { 'name': 'R1', 'type': 'host', 'ip': '10.10.10.1', 'position':
              (100.0, 100.0, 0.0) },
          { 'name': 'CN', 'type': 'host', 'ip': '20.20.20.2', 'position':
              (100.0, 0.0, 0.0) },
     { 'name': 'MN2', 'type': 'host', 'mac': '00:00:00:00:00:22','ip':
         '10.10.10.3', 'position': (10, 170.0, 0.0)},
        ]

wifiintfs = [ {'nodename': 'MN', 'type': 'sta', 'channel': 1, 'ssid':
    'ssid'},
              {'nodename': 's2', 'type': 'ap', 'channel': 1, 'ssid': 'ssid'},
              {'nodename': 's3', 'type': 'ap', 'channel': 6, 'ssid': 'ssid'},
          {'nodename': 'MN2', 'type': 'sta', 'channel': 1, 'ssid': 'ssid'},
            ]

csmalinks = [ {'nodename1': 's1', 'nodename2': 's2'},
              {'nodename1': 's1', 'nodename2': 's3'},
              {'nodename1': 's1', 'nodename2': 'R1'},
              {'nodename1': 'R1', 'nodename2': 'CN'},
            ]

def getWifiNode( wifinode, name ):
    for n in wifinode:
        if n.name == name:
            return n
    return None

def WifiNet():

    "Create an Wifi network and add nodes to it."

    net = Mininet()

    info( '*** Adding controller\n' )
```

```
#    net.addController( 'c0', controller=RemoteController, ip='127.0.0.1',
    port=6633 )
    net.addController( 'c0', controller=RYUBridge, ip='::1', port=6633 )


    wifi = WifiSegment(standard = ns.wifi.WIFI_PHY_STANDARD_80211g)
    wifinodes = []

    for n in nodes:
        nodename = n.get('name', None)
        nodetype = n.get('type', None)
        nodemob = n.get('mobility', None)
        nodepos = n.get('position', None)
        nodevel = n.get('velocity', None)
        nodeip = n.get('ip', None)
   nodemac = n.get('mac', None)
        if nodetype is 'host':
            addfunc = net.addHost
            color = (0, 0, 255)
        elif nodetype is 'switch':
            addfunc = net.addSwitch
            color = (255, 0, 255)
        else:
            addfunc = None
        if nodename is None or addfunc is None:
            continue
        node = addfunc (nodename,mac=nodemac, ip=nodeip)
        mininet.ns3.setMobilityModel (node, nodemob)
        if nodepos is not None:
            mininet.ns3.setPosition (node, nodepos[0], nodepos[1], nodepos[2])
        if nodevel is not None:
            mininet.ns3.setVelocity (node, nodevel[0], nodevel[1], nodevel[2])
        wifinodes.append (node)

    for wi in wifiintfs:
        winodename = wi.get('nodename', None)
        witype = wi.get('type', None)
        wichannel = wi.get('channel', None)
        wissid = wi.get('ssid', None)
        wiip = wi.get('ip', None)
        if witype is 'sta':
            addfunc = wifi.addSta
```

```
        elif witype is 'ap':
            addfunc = wifi.addAp
        else:
            addfunc = None
        if winodename is None or addfunc is None or wichannel is None:
            continue
        node = getWifiNode (wifinodes, winodename)
        tb = addfunc (node, wichannel, wissid)
    for cl in csmalinks:
        clnodename1 = cl.get('nodename1', None)
        clnodename2 = cl.get('nodename2', None)
        if clnodename1 is None or clnodename2 is None:
            continue
        clnode1 = getWifiNode (wifinodes, clnodename1)
        clnode2 = getWifiNode (wifinodes, clnodename2)
        if clnode1 is None or clnode2 is None:
            continue
        CSMALink( clnode1, clnode2, DataRate="100Mbps")

    rv = os.path.isdir("/tmp/pcap")
    if rv is False:
        os.mkdir("/tmp/pcap")
    ns.wifi.YansWifiPhyHelper().Default().EnablePcapAll("/tmp/pcap/wifi")
    ns.csma.CsmaHelper().EnablePcapAll("/tmp/pcap/csma")

    rv = os.path.isdir("/tmp/xml")
    if rv is False:
        os.mkdir("/tmp/xml")
    anim = ns.netanim.AnimationInterface("/tmp/xml/wifi-wifi-simple.xml")
    anim.EnablePacketMetadata (True)

    for n in nodes:
        anim.UpdateNodeDescription (node.nsNode,
            nodename+'-'+str(node.nsNode.GetId()))
        anim.UpdateNodeColor (node.nsNode, color[0], color[1], color[2])

    info( '*** Starting network\n' )
    net.start()
    mininet.ns3.start()

    # Configure Switch to use OpenFlows version 1.3
    print wifinodes[1].cmd( 'ovs-vsctl set bridge s1 protocols=OpenFlow13' )
```

```
    print wifinodes[2].cmd( 'ovs-vsctl set bridge s2 protocols=OpenFlow13' )
    print wifinodes[3].cmd( 'ovs-vsctl set bridge s3 protocols=OpenFlow13' )
    print wifinodes[4].cmd( 'ifconfig R1-eth0 inet6 add 2001:aaaa::1/64' )
    print wifinodes[4].cmd( 'ifconfig R1-eth1 inet6 add 2001:bbbb::1/64' )
    print wifinodes[4].cmd( 'sudo radvd -C radvd1.conf' )
    print wifinodes[4].cmd( 'sysctl -w net.ipv6.conf.all.forwarding=1' )
    print wifinodes[5].cmd( 'ifconfig CN-eth0 inet6 add 2001:bbbb::2/64' )
    print wifinodes[5].cmd( 'ip -6 route add ::/0 via 2001:bbbb::1' )

    CLI( net )

    # Close Mininet and NS3
    info( '*** Stopping network\n' )
    mininet.ns3.stop()
    info( '*** mininet.ns3.stop()\n' )
    mininet.ns3.clear()
    info( '*** mininet.ns3.clear()\n' )
    net.stop()
    info( '*** net.stop()\n' )

if __name__ == '__main__':
    setLogLevel( 'info' )
    WifiNet()
    #sys.exit(0)
```

Publications

- K. Tantayakul, R. Dhaou and B. Paillassa, "**Mobility Management with Caching Policy over SDN Architecture**," presented at *The 3rd IEEE Conference on Network Functions Virtualization and Software Defined Networking (IEEE NFV-SDN 2017)*, Berlin, Germany, November, 2017.
- K. Tantayakul, R. Dhaou, B. Paillassa and W. Panichpattanakul, "**Experimental Analysis in SDN Open Source**," presented at *2017 International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (IEEE ECTI-CON 2017)*, Phuket, Thailand, pp. 337-340, June, 2017.
- K. Tantayakul, R. Dhaou and B. Paillassa, "**Impact of SDN on Mobility Management**," presented at *The 30th IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA-2016)*, Crans-Montana, Switzerland, pp. 260-265, March, 2016.

Bibliography

[1] C. Perkins, "IP Mobility Support for IPv4," *RFC 3344*, August 2002.

[2] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," *RFC3775*, June 2004.

[3] R. Koodli, "Mobile IPv6 Fast Handovers," *RFC 5568*, July 2009.

[4] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," *RFC 5380*, October 2008.

[5] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," *RFC 5213*, Auguest 2008.

[6] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks (ONF White Paper)," *ONF*, April 2012.

[7] C. Alaettinoglu, "Software Defined Networking," *PACKET DESIGN*, 2013.

[8] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information centric network: Research Challenges and Opportunities," *JNCA*, February, 2015.

[9] D. Kutscher, S. Eum, I. Psaras, D. Corujo, D. Saucez, T. Schmidt and M. Waehlisch, "ICN Research Challenges," *Internet-Draft*, March, 2016.

[10] Open Networking Foundation, "OpenFlow Switch Specification Version 1.2 (Wire Protocol 0x03)," December 2011.

[11] Open Networking Foundation, "OpenFlow Switch Specification Version 1.3 (Wire Protocol 0x04)," June 2012.

[12] Open Networking Foundation, "OpenFlow-enabled Transport SDN (ONF Solution Brief)," May 2014.

[13] Z. Zhu, R. Wakikawa, and L. Zhang, "A Survey of Mobility Support in the Internet," *IETF RFC 6301*, July 2011.

[14] J. Ioannidis, D. Duchamp, and G. Maguire, "IP-based Protocols for Mobile Internetworking," *ACM SIGCOMM CCR*, 1991.

[15] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol(LISP)," *RFC 6830*, January 2013.

[16] L. Andrew, "A Border Gateway Protocol 4 (BGP-4)," *Boeing White Paper*, 2006.

[17] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility," *ACM SIGCOMM*, 1999.

[18] R. Ramjee, K. Varadhan, and L. Salgarelli, "HAWII: A Domain-based Approach for supporting Mobility in Wide-area Wireless Networks," *IEEE/ACM Transactions on Networking*, 2002.

[19] A. Grilo, P. Estrela, M. Nunes, "Terminal Independent Mobility for IP (TIMIP)," *IEEE Communications Magazine*, 2001.

[20] C. E. Perkins ,and D. B. Johnson, "Route Optimization for Mobile IP," *Cluster Computing 1*, pp. 161–176, 1998.

[21] D. Farinacci, D. Lewis, D. Meyer, and C. White, "LISP Mobile Node," *Internet-Draft*, December 2016.

[22] J. Kempf (Ed.), "Goals for Network-Based Localized Mobility Management(NETLMM)," *RFC 4831*, April 2007.

[23] R. Hinden, and S. Deering, "IP version 6 Addressing Architecture," *RFC 4291*, February 2006.

[24] S. Thomson, T. Narten, and T. Jinmei, "Neighbor Discovery for IP version 6 (IPv6)," *RFC 4862*, September 2007.

[25] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," *RFC 4861*, September 2007.

[26] M. Liebsch, S. Jeong, and Q. Wu, "Proxy Mobile IPv6 (PMIPv6) Localized Routing Problem Statement," *RFC 6279*, June 2011.

[27] S. Krishnan, R. Koodli, P. Loureiro, Q. Wu, and A. Dutta, "Localized Routing for Proxy Mobile IPv6," *RFC 6705*, September 201.

[28] M. Chiosi et al., "Network Functions Virtualisation: Introduction, Benefits, Enablers, Challenges Call for Action," *SDN, and OpenFlow World Congress*, pp. 1–16, October, 2012.

[29] R. Guerzoni, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. Introductory white paper," *Proc. SDN OpenFlow World Congr.*, pp. 1–16, June, 2012.

[30] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun.Mag*, no. 53, pp. 90–97, February, 2015.

[31] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Communication Magazine*, pp. 114–119, February 2013.

[32] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," *RFC 6241*, June 2011.

[33] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4," *RFC 4271*, 2006.

[34] B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," *RFC 7047*, December 2013.

[35] B. Niven-Jenkins, D. Brungard, M. Betts, and S. Ueno, "Requirements of an MPLS Transport Profile," *RFC 5654*, September, 2009.

[36] C. Thaenchaikun, G. Jakllari, and B. Paillassa, "Mitigate the load sharing of segment routing for SDN green traffic engineering," *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2016)*, pp. 1–6, Octobre 2016.

[37] K. Tantayakul, R. Dhaou, and B. Paillassa, "Experimental Analysis in SDN Open Source," *IEEE ECTI-CON 2017*, pp. 337–340, June 2017.

[38] POX wiki, "POX," *Available: https://openflow.stanford.edu/display/ONL/POX+Wiki.*

[39] RYU, "Ryu 3.18 documentation: WELCOME TO RYU THE NETWORK OPERATING SYSTEM (NOS)," *Available: http://ryu.readthedocs.org/en/latest/index.html.*

[40] RYU, "Build SDN Agilely: Component-Based Software Defined Networking Framework," *Available: https://osrg.github.io/ryu/index.html.*

[41] Pyretic, "Pyretic," *Available: http://http://frenetic-lang.org/pyretic/.*

[42] Trema, "Trema," *Available: https://trema.github.io/trema/.*

[43] Floodlight, "Floodlight," *Available: http://www.projectfloodlight.org/floodlight/.*

[44] ONOS, "," *Available: http://onosproject.org.*

[45] OpenDayLight, "OpenDayLight Controller," *Available: http://www.opendaylight.org/.*

[46] A. Shalimov, D. Zuikov, D. Zimarina, V.Pashkov, and R.Smeliansky, "Advanced Study of SDN/OpenFlow Controllers," *CEE-SECR '13*, October 2013.

[47] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, "Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers," *WCCAIS 2014*, October 2014.

[48] K. Kaur, S. Kaur, and V. Gupta, "Perfomance Analysis of Phyton Based OpenFlow Controllers," *EEECOS 2016*, June 2016.

[49] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, "Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers," *WCCAIS*, 2014.

[50] M. Ehrgott and X. Gandibleux, "Multiple Criteria Optimization State of the Art Annotated Bibliographic Surveys," *Kluwer Academic Publishers*, 2003.

[51] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceeding of the IEEE*, no. 103, pp. 14–76, January 2015.

[52] J. Postel, "Transmission Control Protocol," *RFC 793*, September 1981.

[53] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Communications Magazine*, no. 5, pp. 136–141, 2013.

[54] J. Hu, C. Lin, X. Li, and J. Huang, "Scalability of Control Planes for Software Defined Networks:Modeling and Evaluation," *IEEE IWQoS*, May 2014.

[55] K. He, J. Khalid, A. Gember-Jacobson, S. Das, C. Prakash, A. Akella, L. E. Li, and M. Thottan, "Measuring Control Plane Latency in SDN-enabled Switches," *ACM SOSR*, June, 2015.

[56] K. He, J. Khalid, S. Das, A. Gember-Jacobson, C. Prakash, A. Akella, L. E. Li, and M. Thottan, "Latency in Software Defined Networks: Measurements and Mitigation Techniques," *SIGMETRICS'15*, June 2015.

[57] T. Benson, A. Akella, and D. A. Maltz

[58] A. Tootoonchian and Y. Ganjali, "Hyperflow: a distributed control plane for openflow," *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, 2010.

[59] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama et al, "Onix: A distributed control platform for large-scale production networks," *OSDI*, pp. 1–6, 2010.

[60] S. H. Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," *Proceedings of the first workshop on Hot topics in software defined networks*, pp. 19–24, 2012.

[61] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable Flow-Based Networking with DIFANE," *Proc. ACM SIGCOMM 2010 Conf*, pp. 351–362, 2010.

[62] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: scaling flow management for high performance networks," *ACM SIGCOMM*, no. 41, pp. 254–265, 2011.

[63] S. Song, H. Park, B-Y. Choi, T. Choi, and H. Zhu, "Control Path Management Framework for Enhancing Software-Defined Network (SDN) Reliability," *IEEE Transaction on Network and Service Management*, no. 14, pp. 302–316, June 2017.

[64] J. Jung, E. Sit, H. Balakrishnan, and R. Morris , "DNS performance and the effectiveness of caching," *IEEE/ACM Transactions on Networking*, no. 10, October 2002.

[65] S. N. Bhatti and R. Atkinson, "Reducing DNS Caching," *INFOCOM WKSHPS*, April 2011.

[66] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye, "Analyzing the Performance of an Anycast CDN," *IMC'15*, pp. 351–357, October 2015.

[67] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, M. Varvello, "From content delivery today to information centric networking," *Comput. Netw 57*, no. 16, pp. 3116–3127, 2013.

[68] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Journal Computer Networks*, no. 57, pp. 3128–3141, August 2013.

[69] S. Arianfar and P. Nikander, "Packet-level caching for information-centric networking," *ACM SIGCOMM Worksop*, 2010.

[70] G. Carofiglio, M. Gallo, and L. Muscariello, "Bandwidth and storage sharing performance in information centric networking," *ICN*, 2011.

[71] V. Jacobson, D. K. Smetters, J. D. Thornton, et al, "Networtking named content," *CoNEXT'09*, 2009.

[72] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MutiCache: an incrementally deployable overlay architecture for information centric networking," *IEEE INFOCOM*, 2010.

[73] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: an overlay architecture for information centric networking," *Computer Networs*, no. 55, pp. 936–947, 2011.

[74] Z. Li and G. Simon, "Time-shifted TV in content centric networks: the case for cooperative in-network caching," *Proceedings of ICC*, 2011.

[75] N. Kumar, S. Zeadally, and J. J. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Commun. Mag.*, no. 54, pp. 60–66, October, 2016.

[76] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys Tutorials*, no. 19, pp. 1628–1656, March 2017.

[77] N. A. Jagadeesan and B. Krishnamachari, "Software-Defined Networking Paradigms in Wireless Networks: A Survey," *ACM Computing Survey*, no. 47, January 2015.

[78] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: scalable and flexible cellular core network architecture," *ACM conference on Emerging networking experiments and technologies(CoNEXT)*, pp. 163–174, 2013.

[79] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine*, no. 55, pp. 94–100, July 2017.

[80] H. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen, "Requirements for Distributed Mobility Management," *RFC 7333*, August 2014.

[81] S. Gundavelli and S. Jeon, "DMM Deployment Models and Architectural Considerations," *Internet-Draft*, November 2017.

[82] D. Liu, JC. Zuniga, P. Seite, H. Chan, and CJ. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis," *RFC 7429*, January 2015.

[83] F. Giust, L. Cominardi, and C. J. Bernardos, "Distributed Mobility Management for Future 5G Networks: Overview and Analysis of Existing Approaches," *IEEE Communications Magazines*, January 2015.

[84] 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)*, April 2013.

[85] C. J. Bernardos, A. De La Oliva, and F. Giust, "A PMIPv6-Based Solution for Distributed Mobility Management," *IETF Draft, draft-bernardos-dmm-pmip-03*, January 2014.

[86] S. Jeon, C. Guimaraes, 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.

[87] S. Kim, H. Choi, P. Park, S. Min, and Y. Han, "OpenFlow-based Proxy Mobile IPv6 over Software Defined Network (SDN)," *CCNC 2014*, pp. 119–125, January 2014.

[88] S. M. Raza, D. S. Kim, and H. Choo, "Leveraging PMIPv6 with SDN," *IMCOM (ICUIMC) '14*, January 2014.

[89] Y. Li, H. Wang, M. Liu, B. Zhang, and H. Mao, "Software Defined Networking for Distributed Mobility Management," *GC Wkshps*, pp. 885–889, December 2013.

[90] Y. Wang, and J. Bi, "A Solution for IP Mobility Support in Software Defined Networks," *ICCCN 2014*, pp. 1–8, August 2014.

[91] Y. Wang, J. Bi, and K. Zhang, "Design and Implementation of a Software-Defined Mobility Architecture for IP Networks," *Mobile Netw Appl*, February 2015.

[92] Y. Wang, and J. Bi, "Software-Defined Mobility Support in IP Networks," *Computer Journal*, no. 59, pp. 159–177, 2016.

[93] T. Nguyen, C. Bonnet, and J. harri, "SDN-Based Distributed Mobility Management for 5G Networks," *IEEE WCNC 2016*, April 2016.

[94] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A Survey of Mobility in Information-Centric Networks: Challenges and Research Directions," *NoM'12*, pp. 1–6, June, 2012.

[95] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, "A Survey of Mobility in Information-Centric Networks," *ACM COMM*, no. 12, pp. 90–98, December, 2013.

[96] C. Anastasiades, T. Braun, and V. A. Siris, "Information-Centric Networking in Mobile and Opportunistic Networks," *Wireless Networking for Moving Objects*, pp. 14–30, 2014.

[97] J. Wang, R. Wakikawa, and L. Zhang, "DMND: Collecting data from mobiles using named data," *Proc. IEEE Vehicular Networking Conference (VNC)*, 2010.

[98] , "SIP-based Real-Time Traffic Mobility Support Scheme in Name Data Networking," *JNW7*, no. 6, pp. 918–925, 2012.

[99] Y.Rao, H. Zhou, D. Gao, H. Luo, and Y. Liu, "Proactive Caching for Enhancing User-Side Mobility Support in Named Data Networking," *IMIS*, pp. 37–42, 2013.

[100] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. Siris, and G. C. Polyzos, "Caching and mobility support in a publish-subscribe Internet architecture," *IEEE Communications Magazine*, no. 7, pp. 52–58, 2012.

[101] Y. Thomas, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, "Multisource and Multipath File Transfers through Publish-Subscribe Internetworking," *ACM ICN'13*, pp. 43–44, August, 2013.

[102] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting seamless mobility in named data networking," *IEEE FutureNet V*, June, 2012.

[103] X. Vasilakos, V. A. Siris, G. C. Poyzos, and M. Pomonis , "Proactive selective neighbor caching for enhancing mobility support in information-centric networks," *Proceedings of ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, pp. 61–66, August, 2012.

[104] K. Tantayakul, R. Dhaou, and B. Paillassa, "Impact of SDN on Mobility Management," *IEEE IANA 2016*, pp. 260–265, March 2016.

[105] "Mininet: An Instant Virtual Network on your Laptop (or other PC)," Available: http://mininet.org.

[106] "UMIP: Mobilie IPv6, and NEMO for Linux," Available: http://umip.org.

[107] "EURECOM Sophia Antipolis: OPENAIRINTERFACE PROXY MOBILE IPV6 (OAI PMIPV6)," Available: http://openairinterface.eurecom.fr/openairinterface-proxy-mobile-ipv6-oai-pmipv6.

[108] Iperf, "The Network Bandwidth Measurement Tool," *Available: https://iperf.fr*.

[109] Wireshark, "Wireshark," *Available: https://www.wireshark.org/learnWS*.

[110] tshark, "Dump and Analysis Network Traffic," *Available: https://www.wireshark.org/doc/man-pages/tshark.html*.

[111] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, C. E. Rothenberg, "Mininet-WiFi: Emulating Software-Defined Wireless Networks," *IEEE CNSM*, pp. 384–389, November 2015.

[112] C. Min-Cheng, C. Chien, H. Jun-Xian, K. Ted, Y. Li-Hsing, and T. Chien-Chao, "OpenNet: A Simulator for Software-Defined Wireless Local Area Network," *IEEE WCNC '14*, April 2014.

[113] T. G. O. C. D. OCC-DS2.1, "Design of a Multisite Open-flow-assited Video-on-Demand Distribution architecture (CEOVDS)," *Available: https://www.wireshark.org/doc/man-pages/tshark.html*, May 2014.

[114] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021," *White Paper*, March 2017.

[115] ITEC, "ITEC-Dynamic Adaptive Streaming over HTTP," *Available: http://www-itec.uni-klu.ac.at/dash/*.

[116] Dash-Industry-Forum, " ," *Available: https://github.com/Dash-Industry-Forum/dash.js*.

[117] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A Generic Quantitative Relationship between Quality of Experience and Quality of Service ," *IEEE Network*, pp. 36–41, April, 2010.

[118] "Big Buck Bunny," Available: https://peach.blender.org.

[119] "VideoLAN ORGANIZATION: A project and a non-profit organization, composed of volunteers, developing and promoting free, open-source multimedia solutions," Available: https://www.videolan.org/vlc/index.html.

[120] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC 3550*, July 2003.

[121] R. Hinden, S. Deering, and E. Nordmark, "IPv6 Global Unicast Address Format," *RFC 3587*, August 2003.

[122] Multimedia Signal Processing Group MMSPG, "VQMT: Video Quality Measurement Tool," *Available: https://mmspg.epfl.ch/vqmt*.

[123] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, no. 2, p. 1398–1402, 2004.

[124] J. Søgaard, L. Krasula, M. Shahid, D. Temel, and K. Brunnström, "Applicability of Existing Objective Metrics of Perceptual Quality for Adaptive Video Streaming," *Electronic Imaging*, no. 13, pp. 1–7, 2016.

[125] R. Dosselmann and X. Yang, "A comprehensive assessment of the structural similarity index," *Signal, Image and Video Processing*, no. 5, pp. 81–91, 2009.

[126] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *INFOCOM 2003*, March 2003.

[127] T. H. Luan, X. Ling and X. Shen, "Mac in motion: impact of mobility on the mac of drive-thru internet," *IEEE Trans. Mobile Comput*, no. 11, pp. 305–319, February 2012.

[128] C. Chen, T. H. Luan, X. Guan, N. Lu, and Y. Liu, "CONNECTED VEHICULAR TRANSPORTATION: Data Analytics and Traffic-Dependent Networking," *IEEE VEHICULAR TECHNOLOGY MAGAZINE*, pp. 42–54, September 2017.

[129] SUMO, "Simulation of Urban MObility," *Aviable: http://www.sumo.dlr.de/userdoc/Downloads.html*.

[130] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine 50*, pp. 26–36, 2012.

[131] T. Koponen, M. Chun B-G, A. Kim KH, S. Shenker, and T. Stoica, "A data-oriented (and beyond) network architecture," *SIGCOMM*, no. 37, pp. 181–192, 2007.

[132] "FP7 PURSUIT project (Online)," Available: http://www.fp7-pursuit.eu/PursuitWeb/.

[133] K. V. D. Lagutin and S. Tarkoma, "Publish/Subscribe for Internet: PSIRP Perspective," *Towards the Future Internet Emerging Trends from European Research*, no. 4, pp. 75–84, 2010.

[134] D. T. N. Fotiou, P. Nikander and G. C. Polyzos, "Developing Information Networking Further: From PSIRP to PURSUIT," *BROADNETS 2010*, pp. 1–13, October, 2010.

[135] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network Information (NetInf)- An information-centric networking architecture," *j.comcom*, no. 36, pp. 721–735, January, 2013.

[136] V. Jacobson, DK. Smetters, JD. Thornton, MF. Plass, NH. Briggs, and BL. Braynard, "Networking named content," *CoNEXT'09*, pp. 1–12, 2009.

[137] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Name data Networking," *SIGCOMM*, no. 44, pp. 66–73, 2009.

[138] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling ICN in IP Networks Using SDN," *IEEE International Conference on Network Protocols(ICNP)*, October, 2013.

[139] J. Shi and B. Zhang, "NDNLP: A Link Protocol for NDN," *NDN: Technical Report NDN006*, July, 2012.

[140] P. Suthar, M. Stolic, and A. Jangam, "Native ICN Deployment in LTE Networks," *Vehicular Technology Conference (VTC-Fall)*, no. 86, September, 2017.

[141] P. Suthar, M. Stolic, and A. Jangam, " Native Deployment of ICN in LTE, 4G Mobile Networks," *Internet Draft*, September, 2017.

[142] J. Mysore, and V. Bharghavan, "A New Multicast-based Architecture for Internet Host Mobility," *ACM Mobicom*, 1997.

[143] F. Teraoka, Y. Yokote, and M. Tokoro, "A Network Architecture Providing Host Migration Transparency," *ACM SIGCOMM CCR*, 1991.

[144] P. Bhagwat, and C. Perkins, "A Mobile Networking System Based on Internet Protocol (IP)," *Mobile and Location Independent Computing Symposium*, 1993.

[145] A. Snoeren, and H. Balakrishnan, "An End-to-End Approach to Host Mobility," *ACM Mobicom*, 2000.

[146] W. Xing, H. Karl, and A. Wolisz, "M-SCTP: Design and Prototypical Implementation of An End-to-End Mobility Concept," *5th Intl. Workshop on the Internet Challenge*, 2002.

[147] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," *RFC 3963*, January 2005.

[148] R. Wakikawa, G. Valadon, and J. Murai, "Migraing Home Agents Towards Internet-scale Mobility Deployment," *ACM CoNEXT*, 2006.

[149] R. Atkinson, S. Bhatti, and S. Hailes , "A Proposal for Unifying Mobility with Multi-Homing, NAT, and Security," *MobiWAC*, 2007.

[150] R. Moskowitz, P. Nikander, P. Jokela, and T. henderson, "Host Identity Protocol," *RFC 5201*, April 2008.

[151] S. Cheshire, Z. Zhu, R. Wakikawa, and L. Zhang, "Understanding Apple's Back to My Mac (BTMM) Service," *RFC 6281*, June 2011.

[152] X. Hu, L. Li, Z. Mao, and Y. Yang, "Wide-Area IP Network Mobility," *IEEE INFOCOM*, 2008.