CELTIC Project Number:	**C2012/2-5**
Project Title:	SDN Concept in Generalized Mobile Network Architectures – SIGMONA
Confidentiality:	PU / RE / CO[1]

Document Identifier:	D3.3
Document Title:	**Virtual network mobility management architectures**
Authors:	Michael BOC
Participants:	See authors section
Work Package:	WP3
Version:	Under BSCW version control
Date of last changes:	02.11.15
File Name:	D3.3.doc

Abstract: This document describes mobility management in SDN based virtual networks and summarizes the partner proposals. The document explains for each use-case the proposed solution, architecture and expected benefits and possible integration scenarios.

Keywords: software-defined mobile networks, mobility management, traffic management, resource management, mobile security, challenges software-defined networks, network function virtualization

Disclaimer:

Document History:	Document created

---

[1] Dissemination level:
PU = Public
RE = Distribution to a group specified by the consortium
CO = Confidential, only allowed for members of the consortium

Table of Contents

Authors

Partner	Name	Phone / Fax / e-mail
Budapest University of Techn. and Econ., Mobile Innovation Centre		
	Zoltán Faigl	
		Phone: +36 1 463 2500
		e-mail: zfaigl@mik.bme.hu
	László Bokor	
		Phone: +36 1 463 2048
		e-mail: bokorl@hit.bme.hu
University of Oulu, Center of Wireless Communications		
	Ijaz Ahmad	Phone : +358 453301160
		e-mail : iahmad@ee.oulu.fi
	Suneth Namal	
		Phone:
		e-mail: gkarunar@ee.oulu.fi
CEA		
	Mohamed LABRAOUI	
		Phone:
		e-mail: mohamed.labraoui@cea.fr
	Michael BOC	
		Phone:
		e-mail: Michael.boc@cea.fr

Executive Summary

This document contains the proposed technology solutions and descriptions of introduced functions/interfaces of our use cases in the context of virtual network mobility management architectures and software defined mobile networks (SDMN). Since OpenFlow is the widely used SDN standard, the solutions proposed in this document discuss OpenFlow based mobility management techniques for heterogenous (3GPP/non-3GPP) access environments. This document provides mobility management system models, technological components and their functionality, and logical and systematic workflow of the proposed systems. The sysems are also analyzed using mathematical modeling and simulation/emulation environments such as INET/OMNET++. Moreover, SDN based core network has been proposed, implemented and evaluated for combining technologically different access techniques such as WiFi, LTE, and cognitive radio access networks. The main use cases for mobility management contributed by the SIGMONA project partners extended in this document are:

- Extension of HIP-based DMM solutions for scalable and secure mobility management in SDMNs by BME-MIK
- OpenFlow-based mobility management for heterogeneous SDMNs by BME-MIK
- SDN core for Mobility management and heterogeniety by CWC
- Proxy MIPv6 in SDMNs by CEA

List of abbreviations

3GPP	Third Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ANDSF	Access Network Discovery and Selection Function
AR	Access Router
ARP	Allocation and Retention Priority
API	Application Programming Interface
CAPEX	Capital Expenditure
CN	Correspondent Node
DHCP	Dynamic Host Configuration Protocol
DL	Downlink
DMM	Distributed Mobility Management
E-E	End-to-End
EPC	Evolved Packet Core
EPS	Evolved Packet System
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
GW	Gateway
HA	Home Agent (MIP context), Host association (HIP context)
HIP	Host Identity Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IPv4 / IPv6	Internet Protocol version 4 / version 6
LAN	Local Area Network
LMA	Local Mobility Anchor
LTE	Long Term Evolution
MIH	Media Independent Handovers
MIIS	Media Independent Information Service
MIP	Mobile IP
MAC	Medium Access Control
MAG	Mobility Anchor Gateway
MM	Mobility Management
MN	Mobile Node
MNO	Mobile Network Operator
MR	Mobile Router
NE	Network Element
NEMO	Network Mobility
NFV	Network Function Virtualization
OF	OpenFlow
ONF	Open Network Foundation
OPEX	Operational Expenditure
PCC	Policy Control and Charging
PCRF	Policy and Charging Rules Function
PMIP	Proxy Mobile IP
PoA	Point-of-Access
QoE	Quality of Experience
QoS	Quality of service
RAN	Radio Access Network

SDN	Software-Defined Network
SDMN	Software-Defined Mobile Network
SLA	Service-Level Agreement
UE	User Equipment
UFA	Ultra Flat Architecture
UDR	User Data Repository
UL	Uplink
VLAN	Virtual LAN
VMNO	Virtual Mobile Network Operator
VM	Virtual Machine
VNP	Virtual Network Provider
VirtMobOpt	Virtualization-based Optimisation of Mobile Networks
WARP	Wireless Open-Access Research Platform

1. Introduction

Software defined networking provides a very flexible way to process IP packets and flows. It decouples the control and forwarding function of traditional IP appliance. IP mobility support has been specified by IETF. There is currently not much discussion regarding the mobility support in SDN network. This document discusses the motivation, problem and potential solution of the mobility support in SDN network. IP mobility support has been specified in IETF for years. Both share the similar idea that it introduces an anchoring point to maintain the mapping of the home address and routing address of the mobile node. It uses tunnel to encapsulate the user traffic so that the application layer is not aware of the mobility event.

Mobility management plays a key role in dense environments, where mobiles can easily undergo several handovers during the same connectivity session. Therefore, there is a need to define novel mobility management mechanisms that are both distributed and offered dynamically. They should be distributed in order to avoid any network bottleneck or single point of failure, and to provide better reliability. They should be activated/ deactivated dynamically as needed, in order to globally reduce their signaling load and to increase the achieved performances. Regarding Mobility Management, current IP mobility management solutions pose the following problems, which are exacerbated in dense networks: sub-optimal routing, since typical solutions rely on a central entity to forward packets to the current location of the terminal, hence providing paths that are generally longer than the direct path between the terminal and its communication peer; scalability problems, because existing mobile networks have to be dimensioned to support all the traffic traversing the central anchors and reliability, since centralized solutions share the problem of being more prone to reliability problems, as the central entity is potentially a single point of failure.

Therefore, the SIGMONA project investigates software-defined networking (SDN), network functions virtualization (NFV) and cloud-service paradigms, which are expected to enable less human intervention in network operation, better utilization of resources, hence less CAPEX and OPEX, and more flexibility to fulfil new demands. New demands mean new business models for radio access network, transport network and/or core network sharing, e.g., by virtualization of certain network functions of the mobile network, or by selling whole slices of the mobile network by virtual network providers (VNP) to virtual mobile network operators (VMNO).

The objective of this deliverble is to describe the preliminary solutions for mobility management in virtual networks. Integration and optimization of mobility management within the SDN based mobile network architectures needs that we analyse possible use cases, scenarios, and basic assumptions of these areas in Software-Defined Mobile Networks (SDMNs). Section 2 summarizes the challanges and research objectives of the proposed solutions. Section 3 presents the preliminary version of proposed technological solutions, including the definition of the goals, the proposed functions, targeted benefits and preliminary results. Section 4 addresses the required architectural modifications for the proposed techniques for mobility management in SDMN.

2. Challenges and Research Objectives

This section summarizes the research objectives and main research challenges of mobility management topics. The following topics are addressed in the document:

- Extension of HIP-based DMM solutions for scalable and secure mobility management in SDMNs by BME-MIK
- OpenFlow-based mobility management for heterogeneous SDMNs by BME-MIK
- SDN core for Mobility management and heterogeniety by CWC
- Proxy MIPv6 in SDMNs by CEA

2.1 Media independent OpenFlow-based mobility management for heterogeneous SDMNs

Purely SDN-based mobility management solutions provide mobility transparency to higher layers even without applying additional tunneling. Advanced mobility scenarios and fine-grained mobility management is to be supported within heterogeneous (3GPP/non-3GPP) access environments using a clean, OF-only mobility management solution.

2.1.1 Analysis of challenges

- SDMN integrated ANDSF – OpenFlow solution for 3GPP-compliant handover optimization in heterogeneous SDMN access environments.
    - ANDSF is specified in the 3GPP standards to provide information to assist non-3GPP access network selection and to facilitate efficient inter-technology handovers in 3GPP/non-3GPP context. However ANDSF is also a promising tool for SDMNs, very little previous work is available on the topic of ANDSF and OpenFlow integration. Support of flexible and adaptive network controlled, OF-driven IP flow mobility, location-based optimization of access nework discovery and smart access selection for UEs in SDMN environments are the main challenges to be solved within this topic.
- SDMN integrated IEEE 802.21 MIH – OpenFlow solution for obtaining link information and controlling link behavior, in an access-independent manner for 3GPP/non-3GPP heterogeneous SDMN access environments.
    - 802.21 MIH provides an extended set of features for handover optimization compared to ANDSF. There are published works that rely on the extreme flexibility of SDN mechanisms in order to desing and develop OpenFlow extensions for control and management of wireless links through Media Independent Handover mechanisms. However, none of the published efforts managed to map the proposed OF based MIH-aware mobility management schemes into the LTE/EPC protocols. Mapping IEEE 802.21 primitives together with the integrated OpenFlow-based mobility management extensions to LTE/EPC will offer new perspectives to network designers for enhancing future softwarized mobile networks (SDMNs).

2.1.2 Research directions

In order to address the described challenges above, the following research directions are planned to follow:

- Define an appropriate integration scheme of ANDSF and/or IEEE 802.21 MIH with OpenFlow-based mobility technologies.
- Investigate the deployment possibilities of such an integrated mobility management scheme as an SDN service:
    - Study different OpenFlow-based mobility management alternatives and the underlying procedures and architecture components.
    - Find key procedures for proactive and/or predictive mobility management schemes based on OpenFlow.
- Evaluate the performance of the ANDSF and/or IEEE 802.21 MIH integrated OpenFlow-based proactive scheme compared to standard mobility management solutions using INET/OMNeT++ based simulation models.

2.2 Extension of UFA HIP DMM solution for scalable and secure mobility management in SDMNs

By the integration of centralized and/or distributed anchors (of post-DMM solutions) with the SDN forwarding functions QoS/QoE driven mobility management and support of complex mobility scenarios will be supported.

2.2.1 Analysis of challenges

• What are the performance gains of delegation-based HIP signaling scheme over the E-E HIP signaling scheme in distributed mobile network environment. Several engineering questions must be answered regarding the optimal settings of HIP lifetime parameters and number of GWs in E-E HIP and UFA HIP. During the adjustment of those parameters, the main objective is to keep the signaling overhead low.

  - An interesting question is, e.g., the influence of setting a HIP parameter, called unused association lifetime (UAL), on the overhead. UAL gives the length of idle communication period between two HIP end-nodes after which the protocol deletes the HIP HA and IPsec SA pair. Higher UAL results in longer SA periods, hence lower SA establishment (HIP Base Exchange, BEX) rate, but increased number of IPsec SA and HIP HA entries that must be stored in the memory of HIP nodes. Moreover, it increases the overall rekeying rate in the system. Rekeyings are triggered at constant periods during the lifetime of SAs. Both BEX and rekeying contain the computationally demanding ephemeral Diffie-Hellman (DH) key exchange procedure, therefore reduction of their rate is essential.
  - Another interesting question is related to the delegation of signaling rights. How many levels of delegations, i.e., propagation of signaling rights from delegate to delegate, should be enabled, and how should the original delegator set the expiration time of such an authorization certificate (i.e., the delegation lifetime TDEL). These parameters influence the average length of delegation certificate-chains conveyed together with public-key signatures in mandated update procedures, hence influence the load of the network elements and transport network.

• How SDMN addressing conventions and options can support deployment of UFA HIP GW entities in a loosely coupled SDMN – UFA HIP integration scenario.

  - Loosely coupled integration of SDN technologies and the UFA HIP scheme considers scenarios where OF switches are not HIP-aware, they only provide a transparent transport service for HIP signaling and user plane messages. However, it is an open question how SDMN addressing techniques, conventions and options will emerge, and whether these SDMN addressing schemes will support deployment of UFA HIP (and any post DMM solution), or modifications are required for such integration.

• What are the performance characteristics of HIP-capable OF switch based UFA HIP operation in a tightly coupled SDMN – UFA HIP integration scenario.

Tightly coupled integration of SDN and UFA HIP considers HIP-aware OF switches where HIP (and its UFA HIP extensions) are operating as a novel and flexible secure control and user plane with advanced DMM solution for SDMNs. The most important question in this scheme is the applicability: what are the performance limitations of such a novel, HIP-based SDN mobility architecture.

2.2.2 Research directions

In order to address the described challenges above, the following research directions are planned to follow:

- Define an appropriate integration scheme UFA HIP and OpenFlow-based network management technologies.
- Investigate the deployment possibilities of such an integrated UFA HIP architecture.
- Evaluate the performance of the OpenFlow extended UFA HIP architecture compared to standard mobility management solutions using analytical models.

2.3 Proxy MIPv6 in SDMNs

This topic investigates the concrete evolution of the standardized Proxy Mobile IPv6 (PMIPv6) mobility management protocol for SDN-NFV architectures.

2.3.1 Analysis of challenges

- Addressing of the specification details for efficient integration of PMIPv6 into an SDMN infrastructure (e.g.: integration with service chaining, coordination with the orchestrator, etc.)
- Handling of the SDN control plane.
- Emulation of network functions (Mobile Access Gateway – MAG, Local Mobility Anchor - LMA).
- Relocation of the LMA function on a per user basis.

2.3.2 Research directions

Aiming to address the evoked challenges above, CEA established the following research plan:

- Define a strategy to provide a comprehensive configuration capability of infrastructure devices:
  - The capacity of supporting generic and specific wired and wireless network interfaces such as Ethernet, WiFi, 3G/4G, Bluetooth.
  - The capacity of configuring how devices behave according to their direct neighbors.
  - The capacity to gain multi-hops information about the network performance, e.g., the current round trip time (RTT) with a specific target device.
  - Finally, the capacity to control the configuration of other local applications and libraries such as Open vSwitch (OpenFlow application).
- Investigate about the deployment of PMIPv6 protocol as an SDN service:
  - Study in detail the PMIPv6 protocol and the underlying procedures involved.
  - Isolate key mechanisms to better emulate them in an alternative PMIPv6 service based on SDN.
- Study the deployment considerations and our experiences on both platforms: native PMIPv6 and SDN-based PMIPv6 service.
- Evaluate the performance of PMIPv6 as an SDN service compared to a native implementation.

2.4 SDN core for mobility management and heterogeniety

This topic investigates the use of SDN in the core of LTE and future mobile networks (5G). We have conducted research on the applicability of SDN control plane for integrating various access technologies through the centralized SDN-based core network. Our contribution focuses in the development of an OpenFlow testbed and feasibility study of integrating systems across SDN core.

2.4.1 Analysis of challenges

To meet the growing traffic demands, new access technologies must be used having mobility enabled between them. We suggest a common control platform for all the access technologies. Hence, in this work work we address the challenges in integrating multiple access technologies. We use the concepts of cognitive networking to enable heterogeneity in wireless networks. SDN brings dynamism in networking by introducing programmability and global visibility of the current network state. The centralized controller redirects the network resources at run-time to adapt to the changing user and business needs. Cognitive networks have a high potential to deliver the benefits of controller redirection of resources in SDN. However, this needs cognition process to be mapped to the SDN control plane. With this, the controller gets access to current network statistics which enables correct provisioning of the available network resources.

2.4.2 Research directions

Aiming to address the above challenges and to fulfil the needs of heterogeneity we define the following the research directions.

- Define access technologies that can be used to fulfil the growing traffic needs
- Define the control plane for integrating different access technologies
- Investigate the controller placement in the LTE core network
- Enable cognition in multiple access technologies
- Map congnition in the SDN domain through the logically centralized control plane
- Perform evaluation of the proposed integration, mapping of SDN and cognition in the control plane
- Investigate the deployment of the proposed architecture in real world networks.

3. Proposed Technologies for Mobility Management in SDN

3.1 Media independent OpenFlow-based mobility management for heterogeneous SDMNs

3.1.1 Definition

Purely SDN-based mobility management solutions provide mobility transparency to higher layers even without applying additional tunneling. Advanced mobility scenarios and fine-grained mobility management is to be supported within heterogeneous (3GPP/non-3GPP) access environments using a clean, OF-only mobility management solution.

3.1.2 Proposed solution and architecture

3.1.2.1 OpenFlow support for mobility management in SDMN

One of the most straightforward use-case of OpenFlow is traffic steering and path management that have received tremendous attention within the SDN community. Tools of smart traffic steering can be applied for advanced load balancing, load sharing, content filtering, policy control and enforcement, error recovery and redundancy, and in general, any application which involves traffic flow operations and control. Putting all of these potential SDN applications into the context of mobile and wireless networks, we gather additional set of potential use-cases like traffic offloading and roaming support, content adaptation (e.g., adaptive streaming solutions), and mobile traffic optimization.

After leaving the voice dominated mobile era, the traffic became a lot more unpredictable: the transition to broadband mobile Internet and the traffic explosion in the audio and video streaming domain resulted in a set of new requirements where the excessive bandwidth demand puts barriers in profitable network operation. More efficient paradigm is needed to be adopted in mobile Internet architectures helping to scale capacity, ensure optimal use of resources, and to support service differentiation to maximize revenues.

In this SDN use-case, OpenFlow enables mobile Internet traffic to be dynamically and adaptively moved and removed in the mobile network based on a number of possible trigger criteria, such as

- individual or aggregate flow rate (such as per application or per user aggregation),
- aggregate flows number on a particular port or link,
- flow duration,
- number of UEs per cell,
- available bandwidth,
- IP address,
- type of application,
- device utilization rate, etc.

All of these criteria can be defined either by the user or by the mobile operator. For example, the operator could measure network conditions and decide to offload mobile traffic in case of need. As a user-centric alternative, subscribers could opt in for such mobility management functions based on their preferred parameters and pre-defined policies, like: 1) voice calls should never be offloaded, 2) FTP download traffic should always be offloaded to Wi-Fi, etc.

In a more advanced use-case, it could be envisioned that users travel in a multi-access radio environment simultaneously connecting to multiple RANs (perhaps even ones of their choice). Network parameters such as congestion, quality of service (QoS), quality of experience (QoE) are measured, and triggering factors (e.g., a flow rate threshold) are set and changed dynamically by the mobile operator. For example, “if the flow is an FTP download, and the flow rate exceeds 100 Kbps, hand over the flow from LTE to Wi-Fi.” As the example shows, distinct criteria and thresholds could be applied for different applications and therefore different flow types running on the same UE, or on the terminals of different subscribers. Of course thresholds could be based on the widest range of possible criteria like user/flow profile, location, service plan, etc.

Within this scenario, offloading means moving traffic from a 3GPP RAN (e.g., cellular, small cells, femtocell, etc.) to a Wi-Fi access based on operator-centric decision and execution mechanisms. The vertical handover process must be completely seamless (i.e., no data loss or connectivity problem should occur, IP address should be preserved or at least the address change should be seamlessly handled, etc.) to maintain the user QoE. Offloading can also be used in reverse direction, when e.g., congestion on the Wi-

Fi network triggers to choose a set of mobile users to be moved to another Wi-Fi acces or to a cellular mobile data connection. In a programmable and softwarized environment, we could enable intelligent choices and offer a wide range of new services. The SDN controller could interact with network information server entities such as ANDSF (Access Network Discovery and Selection Function) or 802.21 MIH MIIS (Media Independent Information Service) for discovering and select the most appropriate wireless network to the UE/UEs or even individual application flows, and execute the Wi-Fi offload. This use-case requires the mobile network controller (probably residing in the MME) to cooperate with the ANDSF/MIIS, and also the framework must provide dynamic information on the actual connectivity details of UEs in all the available RANs.

Mobile traffic management and advanced offloading have become hot topics with the explosion of mobile data traffic because it enables operators to optimize resources, and improve QoS/QoE for high bandwidth mobile multimedia applications and services. However, not only offloading is the only interesting application: wireless link aggregation is also a promising sub scenario, which makes UEs able to bundle available wireless connections and bandwidth to increase aggregated link capacity for UEs. This also requires multi-access mobile devices handling different overlapping RANs at the same time.

3.1.2.2 System Model and introduction to technological components for efficient LTE/EPC mobility management in SDMN

Nowadays, the spreading and development of multi-access mobile devices together with the proliferation of different radio access technologies make possible to users to actively benefit from the advances of heterogeneous and overlapping wireless networks. This work-in-progress solution will provide natural extensions of OpenFlow and OpenFlow based technologies in order to provide efficient LTE/EPC mobility management in heterogeneous software defined mobile networks. In order to support multi-RAN environments and provide proactive behaviour, IEEE 802.21 Media Independent Handover (MIH) protocol and Access Network Discovery and Selection Function (ANDSF) will be integrated within this SDN-based technique.

ANDSF was introduced in 3GPP Rel-8 in order to assist to the UEs to discover wireless access networks and provide routing policies, rules and discovery information to facilitate the appropriate network selection for the mobile devices [3][4]. ANDSF also enables advanced traffic steering that adapts to the QoS/QoE and the actual traffic conditions of the controlled 3GPP network.

IEEE 802.21 MIH standard specifies a unified framework for proactive handover control in heterogeneous architectures (i.e., 802.3, 802.11, 802.16, 3G networks) [5]. It supports event and command service (ES, CS) mainly used for local and remote link-layer event monitoring, and information service (IS) collecting static information on access networks. The previous services enable network and MN-controlled handover decisions, i.e., target L2 Point of Access (PoA) selection. The standard defines procedures for PoA resource availability checks, resource reservation, and release. The handover execution protocols and decision algorithms are outside the scope of the standard. Point of Services (PoS) are network elements that communicate directly with the MN, and can assist in handover decision.

Although modern and recently released UEs usually have more than two mobile network interfaces, there is only a small chance nowadays to simultaneously use them. In order to achieve such an advanced function, the above summarized MIH/ANDSF solutions are used to guarantee the quality of applications and services when mobile terminals move from one RAN to another in a heterogeneous wireless network environment. By leveraging IEEE 802.21 MIH / ANDSF capabilities, operators of LTE/EPC networks may provide 1) abstraction of link layers, 2) acquisition of network and link layer information and 3) control and management of link layers. If we assume that MN-CN communication paths are established and maintained using SDN flows management capabilities, then a natural integration of SDN paradigm would result in a novel architecture. Our proposal is aimed at complementing SDN operations with 802.21 MIH and ANDSF information provision and handover control and optimization procedures. Moreover, processing of some parameters/information from 802.21 MIH and ANDSF may require functions in OpenFlow that are currently not supported: the extension and integration of these technologies stand in the middle of our activities.

The above introduced proliferation of overlapping heterogeneous access technologies, and multi-interface devices together with the current trends of increasing mobile broadband traffic volume and dynamicity motivated researches on novel networking solutions, encouraging developments in the area of mobile cloud computing, programmable network solutions and Network Function Virtualization even for multi-access environments. The technology, which provides the necessary toolset to support and widely integrate such innovative solutions, is the adaptation of the SDN paradigm into mobile Internet architectures. SDN techniques bring advanced control, configuration and management capabilities to the legacy network fabric by introducing OpenFlow-based mechanisms and operation. However there are some ongoing works regarding the application of OpenFlow in mobile networks, none of the existing SDN solutions consider link conditions and other cross-layer information when integrating OpenFlow controlling mechanisms into the handover management framework of 3GPP-based evolved mobile internet architectures. Mobility management optimization based on cross-layer information provision is crucial in the future's multi-access wireless setups when potentially a plethora of wireless technologies are simultaneously available.

Our proposal relies on the flexibility of OpenFlow-based SDN mechanisms and the cross-layer information provision and decision supporting capabilities of ANDSF and IEEE 802.21 MIH. Using OpenFlow protocol extensions, a centralized SDN controller entity controls and manages radio access links through an appropriate set of ANDSF and MIH mechanisms for obtaining access quality information, controlling link behavior, helping access network selection and supporting decisions in a flow-aware (i.e., fine grained) and access technology independent manner. With the help of OpenFlow we implement a generalized control of different access networks relying on central and network-centric optimization possibilities and offering a flexible toolset to network developers/operators for deep integration of adaptivity into the mobility management framework on the system level.

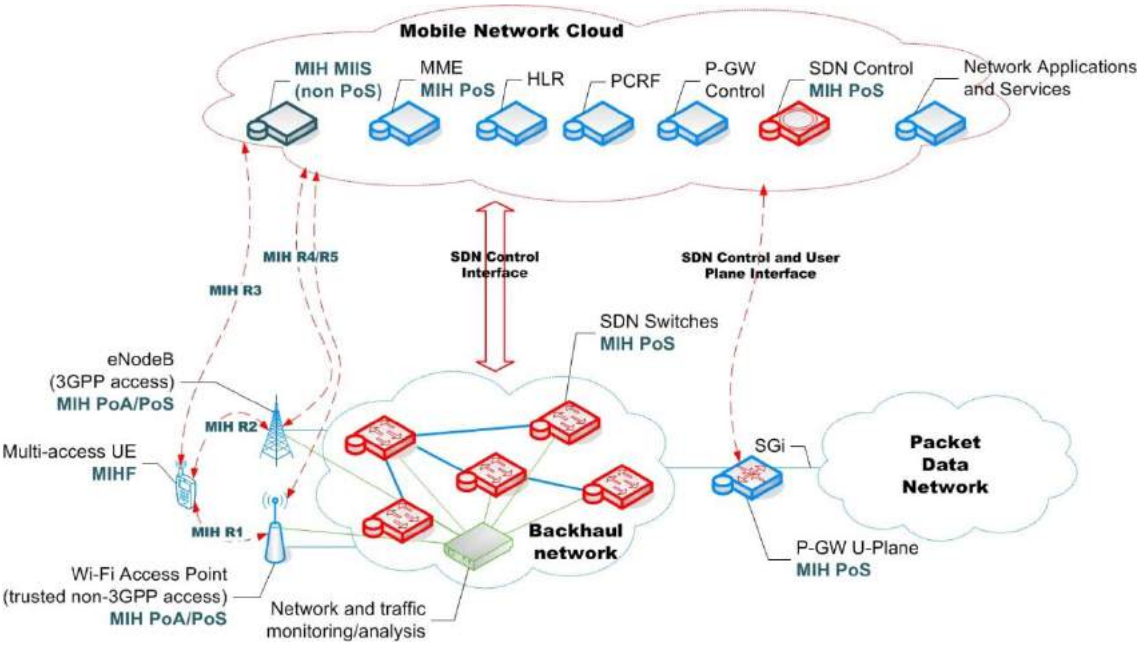


Figure 1: SDN-based mobility management for programmable LTE/EPC architectures (IEEE 802.21+OF use-case).

In the IEEE 802.21 MIH + OF use-case we define an SDMN mobility management framework (Figure 1) and signaling scheme where MIH is used to optimize handovers among different access networks and OpenFlow configures networking nodes to proactively establish and manage communication paths for user level data flows. In this way, OpenFlow mechanisms became aware of mobility related cross-layer information and will rely on these data to select candidate access networks, optimize network resources during handover events, and configure flow paths in a proactive and highly dynamic manner according to the actual connectivity options. The impact of inter-technology handovers on the user flows can be minimized, the user and data planes can be splitted, optimal transmission routes can be continuously maintained and also flow-level decisions can be made and executed to support efficient offloading situations.

In the ANDSF + OF use-case we define an ANDSF-based SDMN mobility management framework (Figure 2) with the appropriate signaling mechanisms in order to provide intelligent access network discovery and selection during handovers in heterogeneous access environments. Using ANDSF, 3GPP compatible policy based network selection and traffic steering is achievable: deriving dynamic policies and distributing them to the ANDSF client implemented in the UEs makes SDMN operators able to initiate enhanced traffic offloading to Wi-Fi from cellular networks, ensure higher level QoS and more efficient resource utilization. Decision complexity of such flow-based vertical handover decisison arises from the fact that the network, application, and user context data and policies are available at different segments of the network (i.e., UE, RAN, backhaul, core, etc.). Therefore OpenFlow-based, centralized control scheme is to be applied in this use-case where the controller commands SDN switches to proactively modify flow tables and such manage user communication paths in the network during handover events. Extension of ANDSF mechanisms is needed because existing ANDSF standards do not take into account any dynamic network and UE conditions like load or congestion.

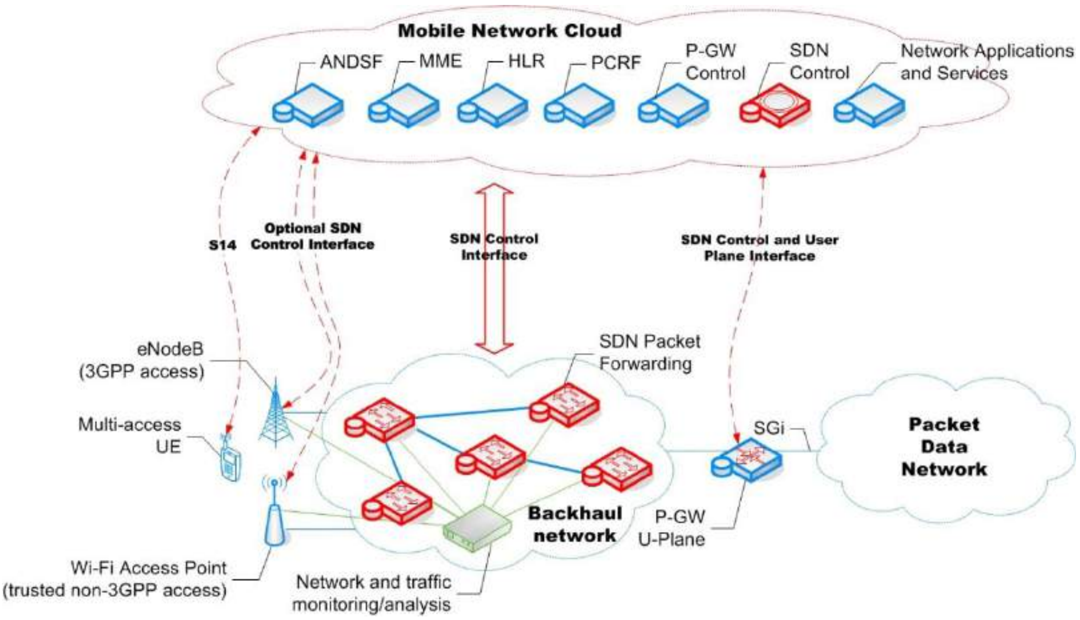


Figure 2: SDN-based mobility management for programmable and virtualized LTE/EPC architectures (ANDSF+OF use-case).

3.1.2.2.1 Components

3.1.2.2.1.1 Extended SDN controller

The SDN controller in our proposal is coupled with 802.21 MIH / ANDSF enforcer functions in order to 1) manage and control mobility independently of the access technology, 2) optimize path based on the wireless and wired link conditions, 3) provide seamless, flow-level handovers and 4) eliminate tunneling overhead to reduce network overhead.

3.1.2.2.1.2 SDN switch

Will be triggered by mobility detection, mobile/ network initiated handover mechanisms (e.g., using continuous QoE/QoS evaluation) and triggers SDN flow re-establishment by sending signalling messages to the SDN controller through the OpenFlow API. Also must provide advanced interfaces to control and manage link layer behaviour in regard of any handover management procedures.

3.1.2.2.1.3 UE

UEs provide interfaces to control, query and manage access links using IEEE 802.21 MIH / ANDSF support.

3.1.2.2.1.4 Network and Link Information Provision

Network information server instances (ANDSF / MIH MIIS) provide static information about the access environment. PCRF handles authorization and policy decision considering that a mobility event will occur in the future. Proactive operation is naturally supported and dynamic information can also be applied during the decision procedures.

3.1.2.2.2 Basic reactive SDN mobility management

Figure 3 introduces the basic operation of our proposed reactive SDN mobility management technique relying on the functions of OpenFlow. The mobile client attaches to a novel Wi-Fi access point. The ongoing transmission from the mobile client initiates a flow table lookup in the forwarding switch of the new access point. Due to the mobility event inconsistency will occur in the flow table, which will initiate the OpenFlow signalling communication between the concerned forwarding entity and the controller. The controller sends a flow mod message to all the switches on the concerned data path after it successfully calculated the new route for the communication flow.

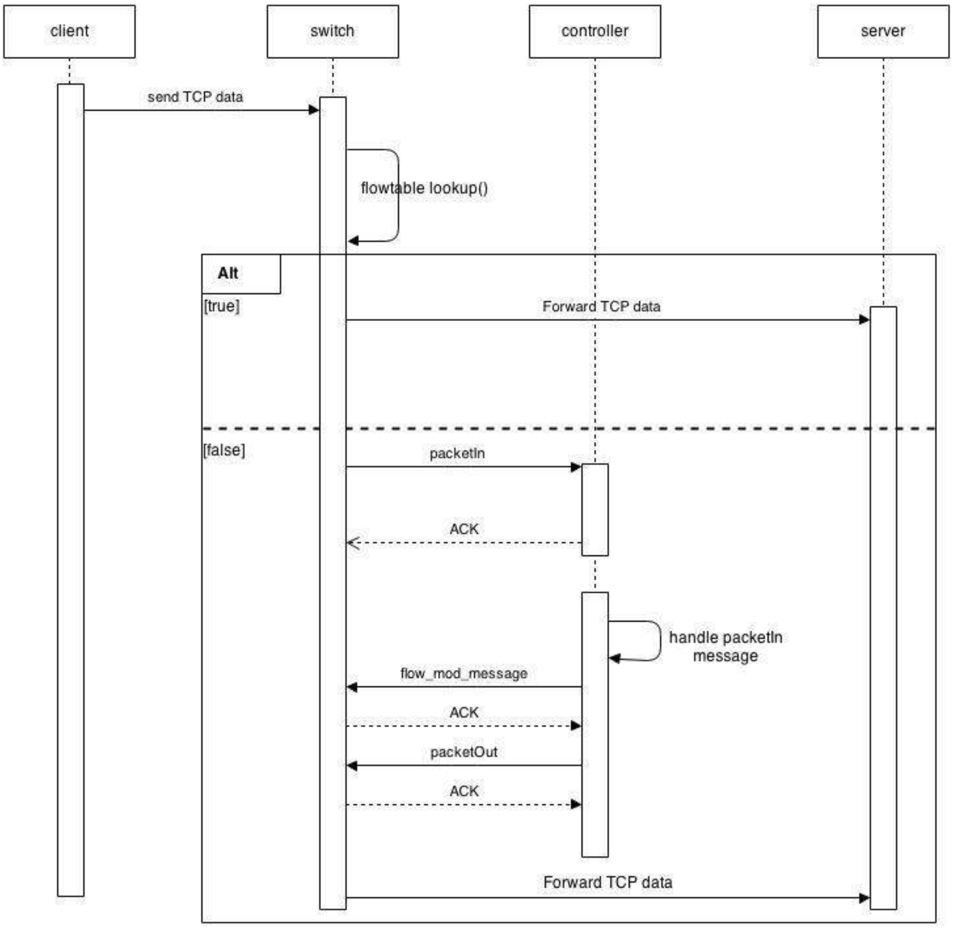


Figure 3: Flowchart of the proposed reactive SDN mobility management.

3.1.2.2.3 Proactive, network-based, media independent SDN mobility management

NFV/SDN technologies introduce enormous flexibility to support and deploy novel mobility management schemes. Only a few existing articles have analyzed the potential of such a dynamically manageable architecture in addressing the problems of future's heterogeneous wireless setups. Guimaraes et al. [24] enhance OpenFlow (OF) with IEEE 802.21 Media Independent Handover (MIH) [5] **Error! Reference source not found.** capabilities to optimize link connectivity establishment in SDNs, but they do not consider complete 3GPP or ETSI NFV architectures in their work. Although 3GPP has already standardized techniques for vertical mobility, these procedures are limited in using cross-layer information for mobility optimization and far from being media independent, proactive and seamless. In [26] Knaesel et al. propose a MIH-enabled Evolved Packet System (EPS) to provide seamless handovers between 3GPP and non-3GPP wireless technologies. However, their proposal does not consider programmable and virtualized mobile architectures.

Our concept maps IEEE 802.21 MIH functions into the proposed NFV architecture and integrates an OF-based handover execution scheme, in order to provide efficient, proactive, fine grained, QoS/QoE-aware and seamless mobility management. MIH is used to optimize handovers among different access networks, while OF configures network resources to proactively establish and manage communication paths for user level data flows. As Figure 4 shows, the proposed SDN controller extensions gather mobility related cross-layer information by relying on dynamic information exchange using MIH event/command services, static data of MIIS (Media Independent Information Service) and the policy control of PCRF (all implemented as VNFs in the telco cloud). Note that MME should also became MIH-aware, as handover (HO) signaling and decision procedures of 3GPP technologies are mainly handled by this node. Based on MIH protocol messages and also using data from other SDN controller modules, the architecture manages resources during handover events, and configures flow paths in a proactive and highly dynamic manner according to the actual connectivity options. Using this concept, the impact of inter-technology handovers on the user flows can be minimized, the user and data planes of mobility management can be separated, optimal transmission routes can be continuously maintained and flow-level decisions can be made and executed to support efficient offloading situations.

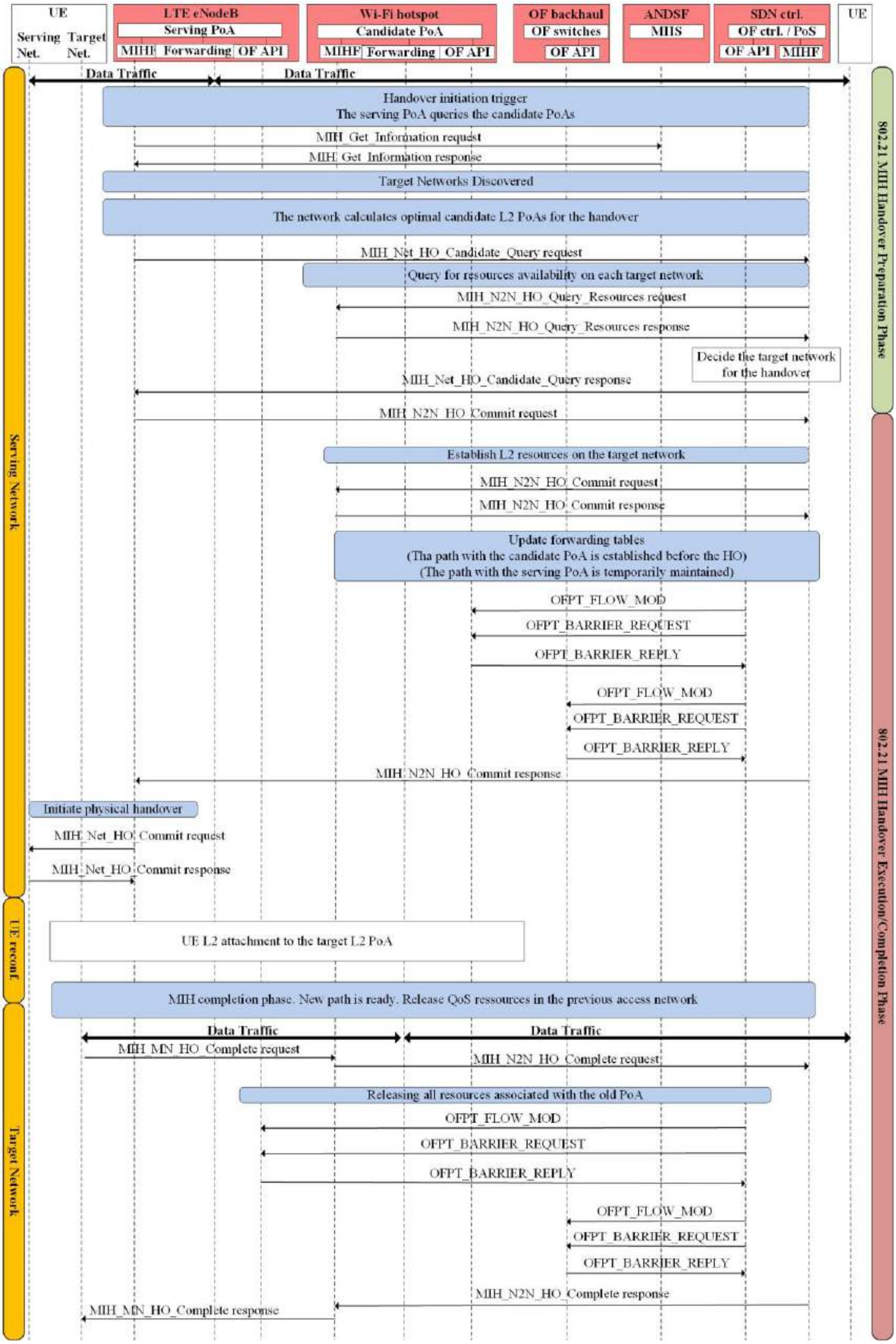


Figure 4: Flowchart of the proposed proactive, network-based SDN mobility management.

3.1.3 Main benefits, results & comparisons

3.1.3.1 Results of OpenFlow-based reactive/proactive mobility management

3.1.3.1.1 INET/OMNeT++ models of the proposed solution

As the OpenFlow standard model is not able to handle mobility events, we had to extend it in order to react on the handover situations and efficiently handle them. The goal is to make the forwarding entities connected to the APs to be able to handle mobile node attachments and detachments and create a controller application, which will dynamically reconfigure the overall SDN network in case of need in order to maintain ongoing connections and communication sessions of the mobile terminals [6]. In order to develop our own model, we have used the existing OfOMNeT solution [8] as a basis of our work.

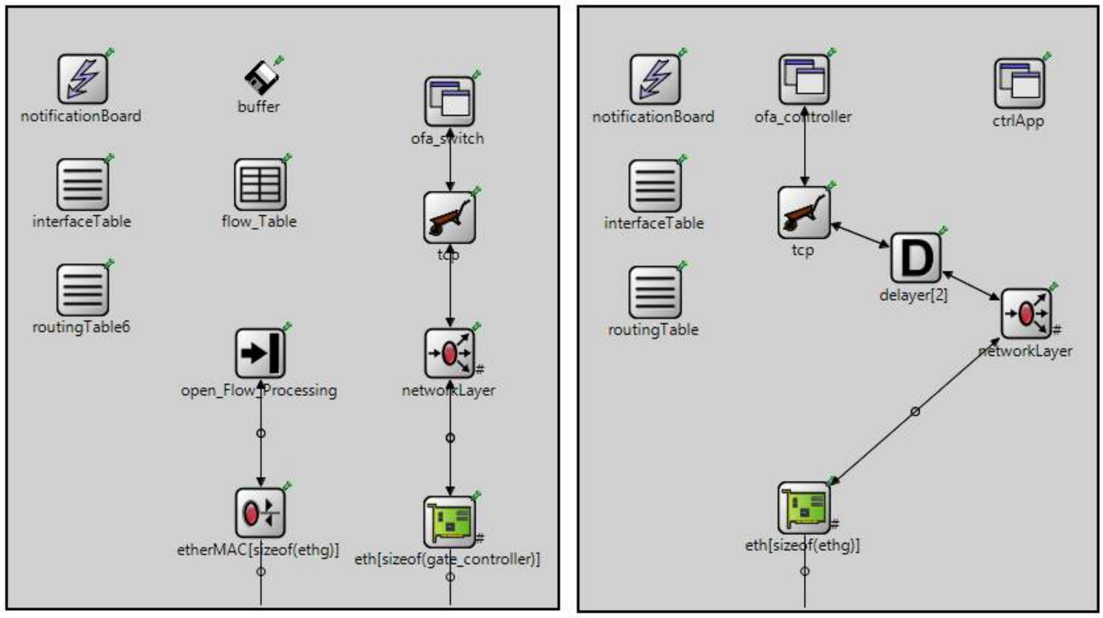


Figure 5: SDN controller (left) and switch (right) structure.

Figure 5, left side shows the SDN switch structure in our model and the simple way how the data and control planes are separated. OpenFlow switches are connected over the data plane, switches and the controller are communicating over the control plane. Every OpenFlow switch contains a flow table which is used as a database supporting packet forwarding decisions. There is also a buffer storing packets in a temporal manner. An interface and a routing table is also integrated to help MAC layer and network layer operations, respectively.

Right after the OpenFlow controller (Figure 5, right side) boots, it builds the control plane channels with all the switches in the SDN domain, collects all the required information and parameters for the forwarding entities, and generally, commands forwarding entities, sets and manages their forwarding rules. Switches boot with empty flow table and controller commands will help them to build and continuously modify the local forwarding databases. When a data packet arrives, it will go into the processing module of the switch (open_Flow_Processing). This module will decide how to forward the packet based on the flow table and the packet information. If there is a match in the flow table for the actual packet, the command in the forwarding entry will be executed. If there is no match, the processing module will send a message to the ofa_switch module. In default, the switch will send a packet-in message towards the controller. That SDN control datagram could contain the whole or a part of the unknown packet. The switch gets the answer in the form of a flow-mod message from the controller, which will be used to create/modify flow table entry or entries. Packet-out messages can also arrive over the control plane in this model, helping switches to decide which port to be used for the packet transmission.

The controller side of this operation starts with the examination of the arriving packet-in message: it will check the wheter there is a need for flow table entry creation/modification or some processing is to be executed on the received packet. If a flow table entry is to be created/modified, a flow-mod message will be sent by the controller towards the switch/switches. The controller can also send a packet-out message, containing the datagram received in the packet-in message (or its ID number), and the forwarding port definition to be used by the switch for the packet transmission. Messages arriving in the controller will be processed by the ofa_controller module, which uses signaling messages to communicate with the ctrlApp module. The ctrlApp module is the controller application, the decision engine of the SDN controller. Hub, switch, forwarding and handover manager applications are currently available in our extensions. We have also added a Delayer module into the switches in order to make the model able to set artificial forwarding delays.

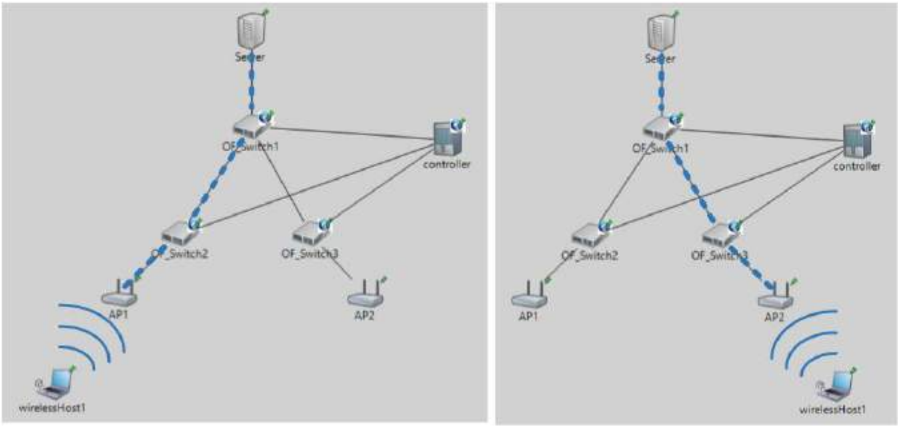


Figure 6: Handover situation in the model.

3.1.3.1.2 The implemented handover management subroutine

We have designed a special controller application to uniqly manage connection of individual mobile terminals and also to modify flow tables in the concerned switches when a handovier situation occurs (Figure 6). The operation is the following. The controller discovers the network topology right after the startup using the OMNeT++ cTopology class, and stores the connection parameters of every nodes in the architecture. The Wi-Fi access points continuously inform their OpenFlow switches about the connecting/disconnecting mobile devices. This requires a special cross-layer signaling between the switch and the access point that is implemented as an extension of the INET's Notification Board, but will be represented as IEEE 802.21 signalling in the complete model. In case of a discovered Wi-Fi level handover event, switches send a message to the controller containing the parameters of the concerning mobile device. The controller assigns the received information to the access point in its database. Then this updated information will be used to create/modify flow table entries in all the switches on the concerned data path.

This controller application decides based on MAC addresses of the mobiles, but these OpenFlow devices are able to rely on any other networking parameters such as IP addresses or Host Identity Tags, etc. Thanks to the application structure, it is very easy to adjust.

When the controller receives a message pertaining an unknown destination node, it will perform a graph search algorithm on the network topology stored in its database. As a results of Dijkstra's algorithm the controller will gather the shortest path towards that destination and also all the concerned switches on that path. Using this information the controller will update all the flow tables in the switches on the path, ensuring the appropriate and optimal flow level data transmission in the network.

3.1.3.2 Results

3.1.3.2.1 Basic reactive SDN mobility management

Our evaluation performed in the above introduced environment focused on the user experience during SDN-managed mobility events. We applied the E-model: a standardized way to measure the perceptual QoS metric that corresponds to the intelligibility and clarity of the speech, as perceived by the listener. It is also called as the ITU G.107 computational model.

The framework applied to our simulation experiments is the INET/OMNeT++ [28] with the OFOMNET OpenFlow model extension [27]. In order to provide a highly configurable, extensible, and adequate model for HIP-based schemes, we extended our previous IPv6-based Host Identity Protocol simulation framework called HIPSim++. The model is built on the top of the 1.99.3 version of INET which is an extension and TCP/IP model collection of the component based, modular OMNeT++ 4.2 discrete event simulation environment. The different scenarios and sub-scenarios were defined by using the OMNeT++ NED language (for topology description) and the omnetpp.ini configuration file (for parameter setup and definition of different simulation runs).

Results are summarized in the figure below where the E-model-based VoIP MOS values are depicted in the boxplots in the function of the number of HOs during the voice over IP transmission. One or two handovers in such a short interval will not result in serious decrease of the quality of user experience: the applied SDN mobility solution performs well. However, if more frequent handovers are likely to happen, the reactive operation scheme must be replaced by a proactive solution. This is what we are working on at the moment.

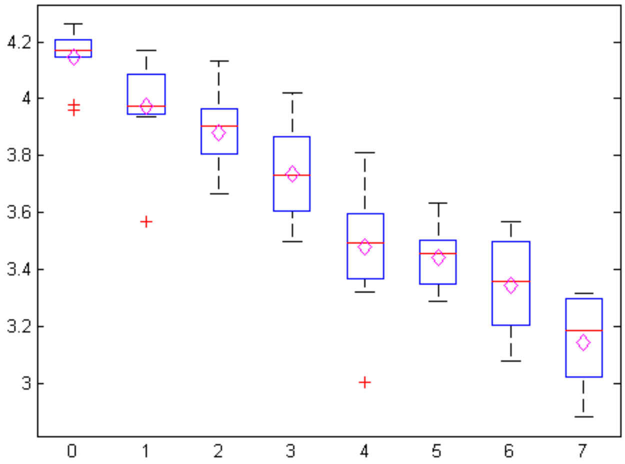


Figure 7: VoIP Mean Opinion Score reference number [MOS] boxplots with different number of handovers during the VoIP session [# of HOs]

3.1.3.2.2 Proactive, network-based, media independent SDN mobility management

The main scenario for the performance evaluation of the proposed OpenFlow-based SDMN mobility management focuses on highlighting the advances of the proactive behavior in a simulated network environment. The applied framework for the experiments is the same as above: INET/OMNeT++ with the OFOMNET OpenFlow model extension. In order to provide a highly configurable, extensible, and adequate model for other legacy and advanced mobility management schemes, we have integrated OFOMNET into our IPv6-based Host Identity Protocol simulation framework called HIPSim++ [29]. The model is built on the top of the 1.99.3 version of INET which is an extension and TCP/IP model collection of the component-based, modular OMNeT++ 4.2 discrete event simulation environment.

The mobile terminal moves between different wireless access points (APs) in the simulated mobile architecture. Therefore, it loses and builds up radio connections during the evaluation period according to the movement path and access coverages. The OpenFlow-based mobility management handles handover events based on the MIH supported cross-layer messages received by the controller from the mobiles, APs and the OF switches: the SDN forwarding plane configures flow paths in a proactive, network-based and highly dynamic manner according to the actual connectivity and other context information.

In our simulations legacy Mobile IPv6 is used as the comparison base and several key performance indicators are applied, like Handover Latency, UDP Packet Loss, different objective quality of experience metrics and TCP throughput. Figure 8 depicts the results in regard of the latter: TCP throughput proportion of the two protocols under analysis (i.e., standard MIPv6 without Routing Optimization and our proposed solution) in a one minute communication session between the mobile node and its correspondent entity performed at different handover frequencies from 0 to 7 and different extra delay values between the access and core network entities (0–90ms). The gain of our proactive, MIH-based solution is eye-catching especially when the circumstances are deteriorating (i.e., the number of handovers is increasing). In case of the highest handover frequency, the proactive OpenFlow-based SDMN solution shows more than 350% increase in TCP throughput. Moreover, the average gain of our advanced MIH-integrated SDN mobility scheme is above 100%.

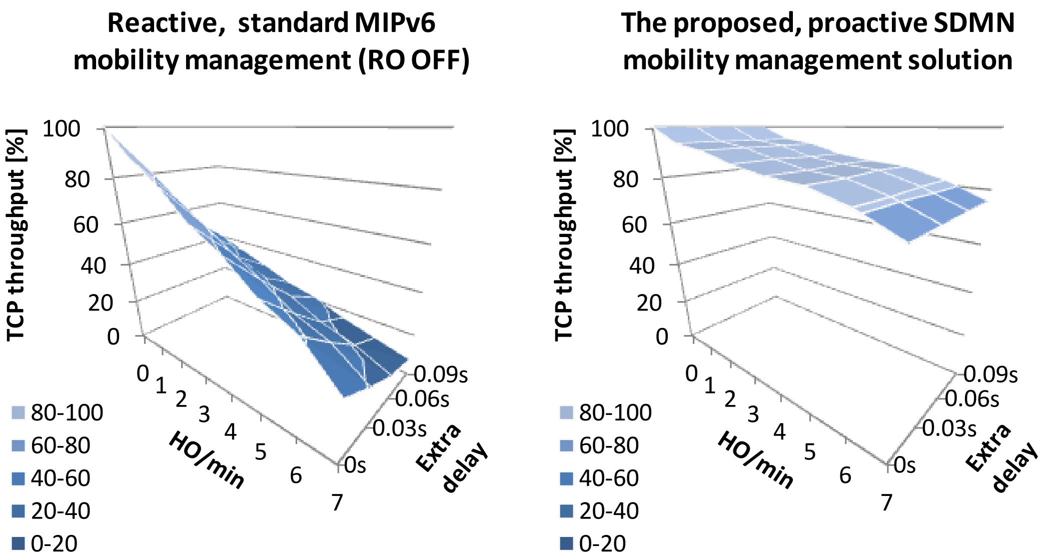


Figure 8: Standard MIPv6-like (left) vs. proactive OpenFlow-based (right) mobility management performance

3.1.4 Recognized issues and future work

A challenge still to be addressed here is the orchestration of multiple runs of optimization, i.e., how to harmonize decisions of HO events triggered by link going down indications due to user mobility with HO events triggered by other control mechanisms such as network initiated smart offloading algorithms.

3.2 Extension of UFA HIP DMM solution for scalable and secure mobility management in SDMNs

3.2.1 Definition

By the integration of centralized and/or distributed anchors (of post-DMM solutions) with the SDN forwarding functions, QoS/QoE driven mobility management and support of complex mobility scenarios will be supported.

3.2.2 Proposed solution and architecture

3.2.2.1 Modeling the signaling overhead in Host-Identity Protocol-based secure mobile architectures

We have introduced an analytical model for the analysis of overheads of HIP-based procedures in distributed mobile network architectures with the objective of determining the performance gains of delegation-based scheme, Ultra Flat Architecture (UFA) HIP, compared to the E-E HIP scheme.

The details of the modeling have been described in. Based on our model, we performed a detailed performance analysis of E-E and UFA HIP mobility schemes, those results are detailed in [2].

3.2.2.2 Coupling UFA HIP and OF-based SDMN technologies

In this use-case (Figure 9), UFA HIP managed HIP/IPsec tunneling is applied to replace standardized tunneling options between GWs and between UEs and GWs. However, for 3GPP and trusted non-3GPP access, OpenFlow-based path management would be required between the PoA and the first UFA GW. The solution supports seamless inter- and intra-GW handovers due to proactive OpenFlow and HIP mobility, multihoming and UFA-HIP based inter-GW mobility service. The proposal could be used by HIP-enabled UEs running any HIP-enabled or legacy network applications. The SDN-aware UFA HIP enables secure integration of untrusted non-3GPP access networks into the SDMN system, supports coexistence of IPv4 and IPv6 network segments and integrates signalling delegation based distributed UFA-HIP into SDMNs as an efficient Loc/ID splitter, security and mobility management option. In general, the proposal is a natural evolution step of UFA-HIP by:

- Inheriting benefits of efficient and secure signalling delegation of UFA-HIP systems in SDMN architectures
- Providing optimal UFA-GW selection based on SDN operation

- Ensuring optimal path selection between source and target UFA-GW nodes during and after handover events
- Efficiently supporting different UFA-HIP deployment scenarios
- Helping deployment by creating more lightweight UFA-GWs

The benefits of this use case come forward also in heterogenous access networks

- IEEE 802.21 MIH is integrated
- Proactive vertical handover management is naturally supported

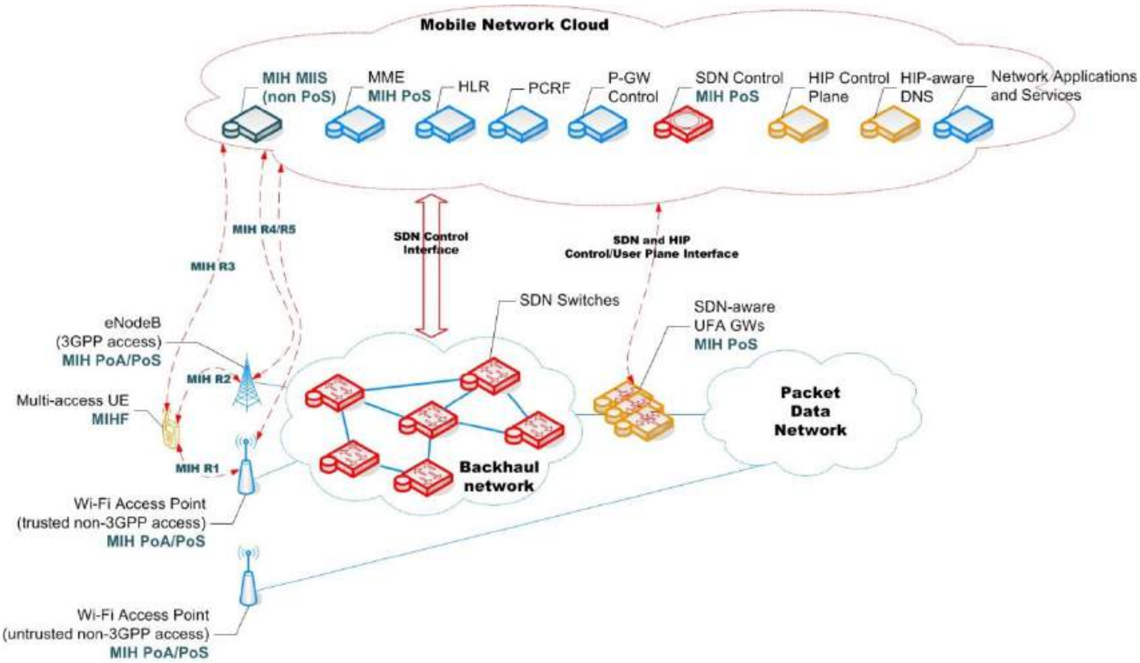


Figure 9: SDN-aware UFA HIP mobility management framework

3.2.3 Main benefits, results & comparisons

3.2.3.1 Results in coupling UFA HIP and OF-based SDMN technologies

In order to extensively evaluate our post-DMM solutions, to compare different performance metrics and to help the design, we have created a generic analytical model based on mathematical apparatus.

3.2.3.1.1 Generic network topology of the modified and updated analytical model

The proposed analytical model is built on a generic network topology model depicted in Figure 10.

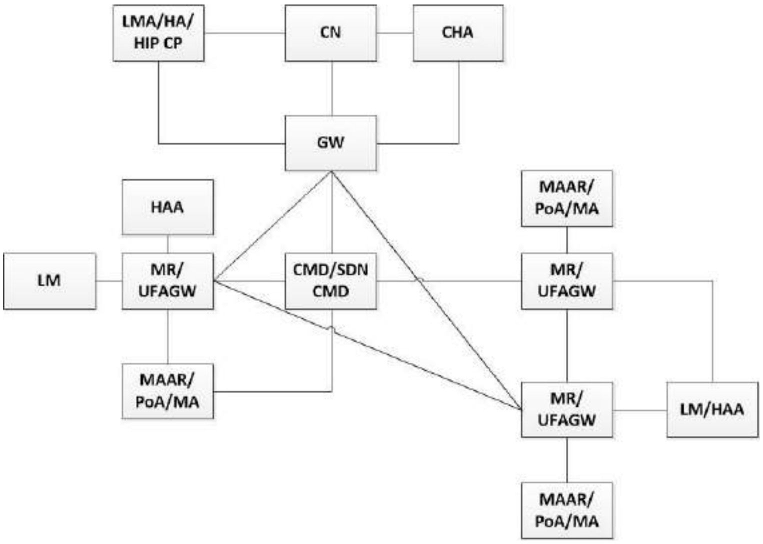


Figure 10: Generic network topology model for distributed, flat and SDN based mobility management schemes

Currently this model contains the building blocks of MIPv6, MIPv6 RO, MIPv6 ERO, Corresponding Network Homing, MIPv6 and PMIPv6 based DMM (partially and fully distributed cases), UFA-HIP, UFA-PMIP, and OpenFlow/OpenRoads. Therefore mobility scenarios of these protocols can be evaluated [7].

3.2.3.1.2 Details of the modified and updated analytical model

Based on the generic network topology model and the taxonomy of Table 1 the following system of equations were created to form our modified and updated analytical model [7].

Table 1: Location update taxonomy

τ1	delay of wireless links
τ2	delay of wired links
s	number of retransmissions
Ttimer	timer
k	number of signaling messges
dXY	number of hops between X and Y nodes
TL2	L2 handover latency
ϕ	extra delay
aX	processing capability of anchor node X
ΔT	movement detection latency
F	switch configuration time

3.2.3.2 Formulation of Layer 3 location update delay cost of different mobility management mechanisms

3.2.3.2.1 MIPv6:

$Location\ Update = BUBA + TL2 + \varphi$ where

$BUBA = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWHA * \tau 2) + k * aHA + s * Ttimer$

3.2.3.2.2 MIPv6 RO:

$Location\ Update = HoTIHoT + CoTICoT + BUBA + CNBUBA + TL2 + \varphi$ where

$CNBUBA = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCN * \tau 2) + s * Ttimer$ and

$$HoTIHoT = \Delta T + 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWHA * \tau 2 + k * dHACN * \tau 2) + k * aHA + s * Ttimer$$

and

$$CoTICoT = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCN * \tau 2) + s * Ttimer$$

3.2.3.2.3 MIPv6 ERO:

$\text{Location Update} = HoTIHoT + BUBA + earlyBUBA + totalBUBA + TL2 + \varphi$ where

$earlyBUBA = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCN * \tau 2) + s * Ttimer$ and

$teljesBUBA = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCN * \tau 2) + s * Ttimer$

3.2.3.2.4 Corresponding Network Homing:

$\text{Location Update} = HoTIHoT + CoTICoT + BUBA + TL2 + \varphi$ where

$$HoTIHoT = \Delta T + 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCHA * \tau 2 + k * dCHACN * \tau 2) + k * aCHA + s * Ttimer$$

and

$$BUBA = 2 * (k * dMNMA * \tau 1 + k * dMAGW * \tau 2 + k * dGWCHA * \tau 2) + k * aCHA + s * Ttimer$$

3.2.3.2.5 MIPv6 DMM:

$\text{Location Update} = \text{DMM MIPv6 Security Delay} + (MR2LM + MR1LM)/2 + TL2 + \varphi$ where

$MR1LM = k * dujMRregiMR * \tau 2 + k * dujMRLM * \tau 2 + k * aregiMR + k * aLM + s * Ttimer$ and

$$MR2LM = k * dujMRregiMR * \tau 2 + k * dujMRregiLM * \tau 2 + k * dujMRujLM * \tau 2 + k * aregiMR + k * aregiLM + k * ujLM + s * Ttimer$$

$\text{DMM MIPv6 Security Delay} = HoTIHoT + CoTICoT$

3.2.3.2.6 PMIPv6 DMM (partially distributed with SDN technologies):

$$\text{Location Update} = \Delta T + TL2 + \text{PMIP Security Delay} + \varphi + 2 * (k * \tau 2 * dSMAARCMD + k * \tau 2 * dPMAARCMD) + 2 * k * aCMD + k * aPMAAR + k * aSMAAR + s * Ttimer + dMNSMAAR * \tau 1 * k$$

or

$$\text{Location Update} = \Delta T + TL2 + \text{PMIP Security Delay} + \varphi + + k * \tau 2 * dSMAARCMD + k * \tau 2 * dPMAARCMD + k * \tau 2 * dSMAARPMAAR + k * \tau 2 * dPMAARCMD + 2 * k * aCMD + k * aPMAAR + k * aSMAAR + s * Ttimer + dMNSMAAR * \tau 1 * k$$

or

$$\text{Location Update} = \Delta T + TL2 + \text{PMIP Security Delay} + \varphi + + k * \tau 2 * dSMAARCMD + k * \tau 2 * dPMAARCMD + k * \tau 2 * dSMAARPMAAR + k * \tau 2 * dPMAARCMD + 2 * k * aCMD + k * aPMAAR + k * aSMAAR + s * Ttimer + dMNSMAAR * \tau 1 * k$$

depending on the actual role of the controller (i.e., PBU/PBA relay, MAAR locator, MAAR proxy, respectively), where

$\text{PMIP Security Delay} = HoTIHoT + CoTICoT$

3.2.3.2.7 PMIPv6 DMM (fully distributed scheme):

$$\text{Location Update} = \Delta T + TL2 + \text{PMIP Security Delay} + \varphi + 2 * (k * \tau 2 * dSMAARPMAAAR) + 2 * k * aCMD + k * aPMAAR + k * aSMAAR + s * Ttimer + 2 * (dMNSMAAR * \tau 1 * k)$$

where

$\text{PMIP Security Delay} = HoTIHoT + CoTICoT$

3.2.3.2.8 UFA-HIP:

$\text{Location Update} = \text{UFA\_HIP Security Delay} + \Delta T + TL2 + MNUFAGW + UFAGWHIP + \varphi$ where

$UFAGWHIP = 4 * (k * dUFAGWGW * \tau 2 + k * dGWHIP * \tau 2) + 2 * k * aHIP + s * Ttimer$ and

$MNUFAGW = 2 * (k * dMNPOA * \tau 1 + k * dPOAUFAGW * \tau 2) + k * aUFAGW + s * Ttimer$ and

$\text{UFA\_HIP Security Delay} = HoTIHoT + CoTICoT$

3.2.3.2.9 UFA-PMIP:

$\text{Location Update} = \text{UFA\_PMIP Security Delay} + \Delta T + TL2 + MNUFAGW + UFAGWLMA + \varphi$ where

$UFAGWLMA = 2 * (k * dUFAGWGW * \tau 2 + k * dGWLMA * \tau 2) + 2 * k * aLMA + s * Ttimer$ and

$MNUFAGW = 2 * (k * dMNPOA * \tau 1 + k * dPOAUFAGW * \tau 2) + k * aUFAGW + s * Ttimer$ and

$\text{UFA\_PMIP Security Delay} = HoTIHoT + CoTICoT$

3.2.3.2.10 OpenFlow/OpenRoads:

$$Location\ Update = OpenFlow\ \ Security\ Delay + \Delta T + k * \tau1 * dMNMAAR + k * \tau2 * dMAARCMD + F + \varphi$$

where

$$OpenFlow\ \ Security\ Delay = HoTIHoT + CoTICoT$$

3.2.3.3 Formulation of user plane communication cost of different mobility management mechanisms

3.2.3.3.1 MIPv6:

$$Message\ cost = h * dMNMA * \tau1 + h * \ dMAGW * \tau2 + h * dGWHA * \tau2 + h * dHACN * \tau2 + IE$$

3.2.3.3.2 MIPv6 RO:

$$Message\ cost = h * dMNMA * \tau1 + h * \ dMAGW * \tau2 + h * dGWCN * \tau2$$

3.2.3.3.3 MIPv6 ERO:

$$Message\ cost = token * (h * dMNMA * \tau1 + h * \ dMAGW * \tau2 + h * dGWCN * \tau2)$$

3.2.3.3.4 Corresponding Network Homing:

$$Message\ cost = h * dMNMA * \tau1 + h * \ dMAGW * \tau2 + h * dGWCHA * \tau2 + h * dCHACN * \tau2 + IE$$

3.2.3.3.5 MIPv6 DMM:

$$Message\ cost = h * dMNMA * \tau1 + h * dMAMR * \tau2 + h * \ dMRGW * \tau2 + h * dGWCHA * \tau2$$

3.2.3.3.6 PMIPv6 DMM (partially distributed with SDN technologies):

$$Message\ cost = h * dMNMAAR * \tau1 + h * \ dMAARGW * \tau2 + h * dGWCN * \tau2$$

3.2.3.3.7 PMIPv6 DMM (fully distributed scheme):

$$Message\ cost = h * dMNMAAR * \tau1 + h * \ dMAARGW * \tau2 + h * dGWCN * \tau2$$

3.2.3.3.8 UFA-HIP:

$$Message\ cost = h * dMNPOA * \tau1 + h * \ dPOAUFAGW * \tau2 + h * dUFAGWGW * \tau2 + h * dGWCN * \tau2$$

3.2.3.3.9 UFA-PMIP:

$$Message\ cost = h * dMNPOA * \tau1 + h * \ dPOAUFAGW * \tau2 + h * dUFAGWGW * \tau2 + h * dGWLMA * \tau2 + h * dLMACN * \tau2 + IE$$

3.2.3.3.10 OpenFlow/OpenRoads:

$$Message\ cost = h * dMNMA * \tau1 + h * \ dMAGW * \tau2 + h * dGWCN * \tau2$$

3.2.3.4 Forumulation of anchor scalability characteristics of different mobility management mechanisms

Table 2: Scalability parameters.

J, J'	number of signaling messages
M	number of mobile terminals
A	tunneling cost
H	number of user plane messages
D, D'	number of anchor nodes
F, F'	processing rate of anchor nodes
V	variable
E	number of domain changes
S	switch configuration time
C	variable

3.2.3.4.1 MIPv6:

$$Scalability = (F * D)/(E * J + M * H * A)$$

3.2.3.4.2 MIPv6 RO:

$$Scalability = (F * D)/(E * J + M * H * A * C)$$

3.2.3.4.3 MIPv6 ERO:

$$Scalability = (F * D)/(E * J * V + E * J' * (1 - V) + M * H * A * C)$$

3.2.3.4.4 Corresponding Network Homing:

$$Scalability = (F * D)/(E * J + M * H * A)$$

3.2.3.4.5 MIPv6 DMM:

$$Scalability = (F * D)/(E * J * V + E * J' * (1 - V) + M * H)$$

3.2.3.4.6 PMIPv6 DMM (partially distributed with SDN technologies):

$$Scalability = \min(F * D, F' * D') / (E * J) + (F' * D') / (M * H)$$

3.2.3.4.7 PMIPv6 DMM (fully distributed scheme):

$$Scalability = (F * D)/(E * J + H * M)$$

3.2.3.4.8 UFA-HIP:

$$Scalability = \min(F * D, F' * D') / (E * J) + (F' * D') / (M * H)$$

3.2.3.4.9 UFA-PMIP:

$$Scalability = \min(F * D, F' * D') / (E * J) + \min(F * D, F' * D') / (M * H * A)$$

3.2.3.4.10 OpenFlow/OpenRoads:

$$Scalability = (F * D)/(E * J + S)$$

3.2.3.5 Results

Our results are focusing on the scalability issues. In a scalable mobile architecture, the service quality must not be degraded when the number of mobile nodes or mobility events increases or the volume of mobile traffic growths. Scalability highly depends on the signalling and in case of centralized or partly centralized schemes the user plane load of anchor terminals. Therefore this KPI is a quotient of the anchor capacity and the actual computational load on the anchor node.

From this point of view, legacy MIPv6 performs the worst. It is not suprising as the centralized Home Agent manages all the user plane packets of every subscriber. The HA handles also the signalling plane, and deals with the computational overhead of encapsulation-decapsulation (which is costlier then simple packet forwarding) for every user plane datagram. The graph depicts that scalability highly depends on the number of mobile terminals handled by one anchor node instance in the network.

MIPv6 RO and ERO perform better than CNet Homing and legacy MIPv6 thanks to the optimal routes created during their routing optimization phase to be executed after every handover event. These optimal routes leave Home Agent out from the packet transmission path such decreasing the load on the anchor node of the protocols. However, both protocols still rely on the HA functionalities even if the amount of time the HA plays as a user plane anchor for a particular subscriber is significantly decresead. It means that before finishing the RO/ERO procedures, the HA is still a user plane anchor entity. Of course ERO shows a slightly better performance thanks to the application of CGAs and token based enhanced route optimization.

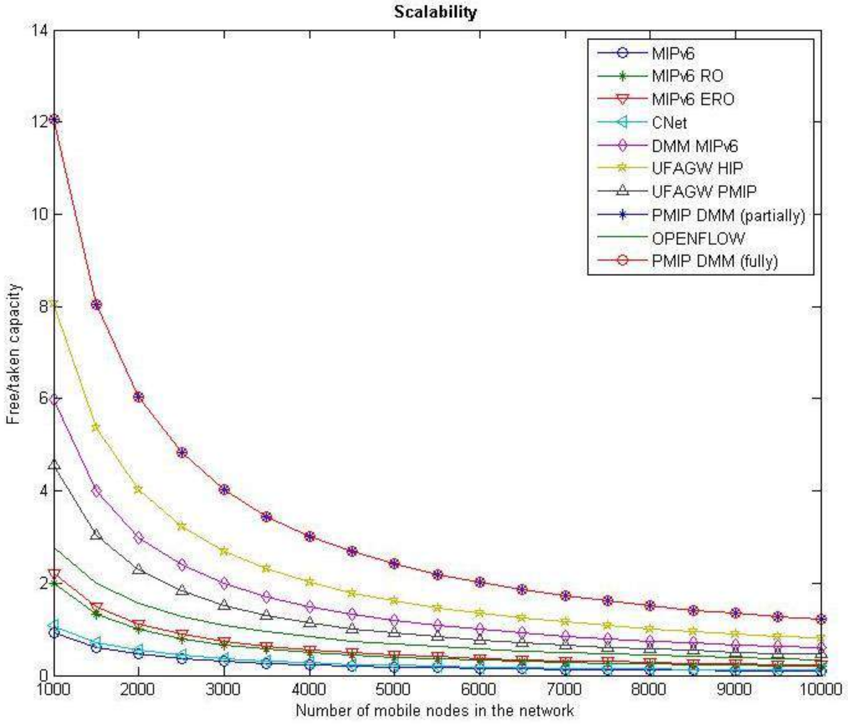


Figure 11: Scalability analysis of distributed, flat and SDN based mobility management schemes

DMM MIPv6 uses distributed anchors: HA functions of a domain are splitted into smaller pieces. Therefore not one centralized anchor will be loaded with the signalling and forwarding tasks, but the appropriate parts of this distributed scheme. However, if high number of mobile nodes joins to one MR and these MNs also create serious traffic, then the performance of that networking segment will be decreased. Of course this performance remission will harm only a particular networking segment, and not all the subscribers as in case of CNet and legacy MIPv6 solutions. Also DMM MIPv6 creates optimal routes between the communication partners and also eliminates tunnels from the data path.

The efficiency of UFA HIP is also provided by the distributed anchor nodes: load is distributed between the UFA GW entities, while the central HIP controller only deals with signalling packets. This is the main difference between UFA HIP and UFA PMIP from the scalability point of view: in the PMIP case also the user plane communication traverses the central LME node. This, and the applied bi-directional tunnels make UFA PMIP perform worst compared to the HIP scenario.

3.2.4 Recognized issues and future work

Since the beginning of this research many novel OpenFlow based mobility techniques and schemes designed for SDMNs were published (e.g., [30], [31], [32]), meaning that the list of modelled and examined solutions in our comparison would require a significant extension. Such an expansion could cause the modification and update of the general topology and formulas used in our model.

3.3 Proxy MIPv6 in SDMNs

3.3.1 Definition

Software-Defined Networking (SDN) enables programmatic exploitation and orchestration of a network infrastructure and resources [9]. Typically, a SDN system provides a required set of actions/behaviors to allow emulation of complex network protocols over generic unspecialized network equipments. For instance, a generic router could be used to act as a Mobile Management Entity (MME) or a Packet Gateway (P-GW) without having the appropriate software running natively [10][11].

Our objective is to design the PMIPv6 evolution that takes full advantage of the SDN-NFV concept. This use case targets to cover the challenges and objectives defined in Section 2.3. This use case requires an SDN southbound protocol able to provide low-level link technology information such as Received Signal Strength Indicator, wireless channel frequency, information about neighbouring wireless access-points. To that end specification of SDN software for the controller and on infrastructure devices is ongoing. Furtheremore, implementation of an NFV architecture relying on the OpenStack Cloud infrastructure is on-going as well with the necessary orchestration capabilities.

Providing comprehensive configuration capability of infrastructure devices is of paramount importance in SDN [12]. Indeed, deploying a network service requires accurate knowledge of the devices state and capabilities through the southbound API. To that end, we present NEON API, a southbound protocol aiming to overcome the current issues: (1) relying exclusively on the OpenFlow protocol limits our device management capacity to routing and flow handling. Furthermore, OpenFlow support of wireless specific ports statistics is inexistant. (2) Southbound management protocols, such as NetConf or ForCES [13][14] ensure network interfaces configuration. However, they do not provide API for live low-level elementary network configuration functions. Finally, dynamic devices and interfaces discovery are not supported.

In the following subsection, we introduce NEON API, our new SDN southbound protocol that we designed to enable fast deployment of services for dynamic infrastructure. Furthermore, we decompose the PMIPv6 procedures to explain how to achieve the same protocol behavior using an SDN service. And that, using the capabilities offered by NEON.

3.3.2 Proposed solution and architecture

3.3.2.1 NEON SDN southbound protocol

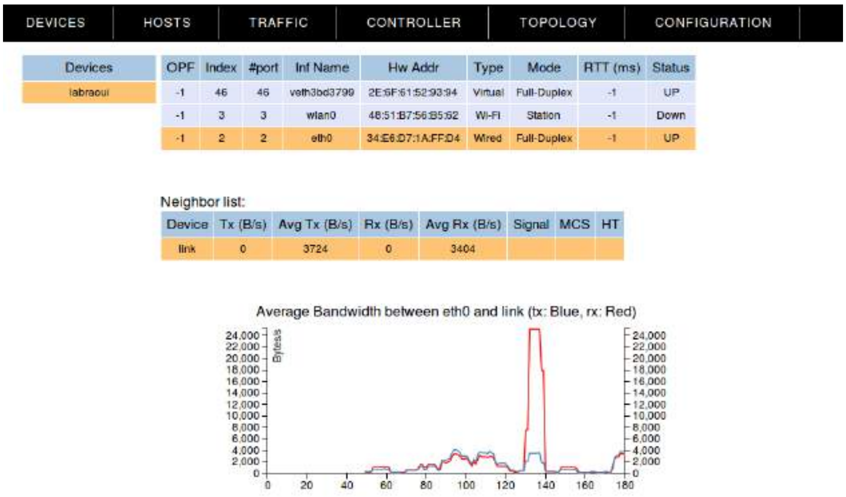
Deployment of network services over an SDN system requires a set of abilities that span configuration of SDN devices, management of flows, and performance monitoring. Such network services should be able to get an updated view of the network topology, to analyze devices capabilities, and setup devices configuration: it makes use of the infrastructure as a service (IaaS) paradigm. If most of the attention is today on OpenFlow [15], we may note that it is only focused on traffic and routing tables management. Considering this, we propose NEON, a structured but yet flexible southbound protocol that aims at ensuring extended, dynamic, and resilient remote device configuration capabilities.

3.3.2.1.1 NEON design principles

NEON is designed around four key requirements: (1) the capacity of supporting generic and specific wired and wireless network interfaces such as Ethernet, WiFi, 3G/4G, Bluetooth. The aim is to cover a wide range of interfaces and devices from powerful servers to users' terminals. (2) The capacity of configuring how devices behave according to their direct neighbors. For instance defining how local network services should forge ICMP and/or ND control packets. (3) The capacity to gain multi-hops information about the network performance, e.g., the current round trip time (RTT) with a specific target device. (4) Finally, the capacity to control the configuration of other local applications and libraries such as Open vSwitch (OpenFlow application) [14]. To support these requirements, the NEON protocol has been organized around three categories of messages.

Statistics:

This category of messages uses a binary format to query and return interfaces statistics. Such messages are targeted for regular polling from the controller to maintain a coherent and complete view of the network topology. Statistics cover specific information about the interfaces status. For instance, the operation mode (station, access-point, ad-hoc, etc.), the capabilities (mesh mode capable, etc.), the BSSID, the neighboring MAC addresses, Tx/Rx count, if the interface is managed by OpenFlow, etc.



Devices	OPF	Index	#port	Inf Name	Hw Addr	Type	Mode	RTT (ms)	Status
labraoui	-1	46	46	veth3bd3799	2E:5F:61:52:93:94	Virtual	Full-Duplex	-1	UP
	-1	3	3	wlan0	48:51:B7:56:B5:62	Wi-Fi	Station	-1	Down
	-1	2	2	eth0	34:E6:D7:1A:FF:D4	Wired	Full-Duplex	-1	UP

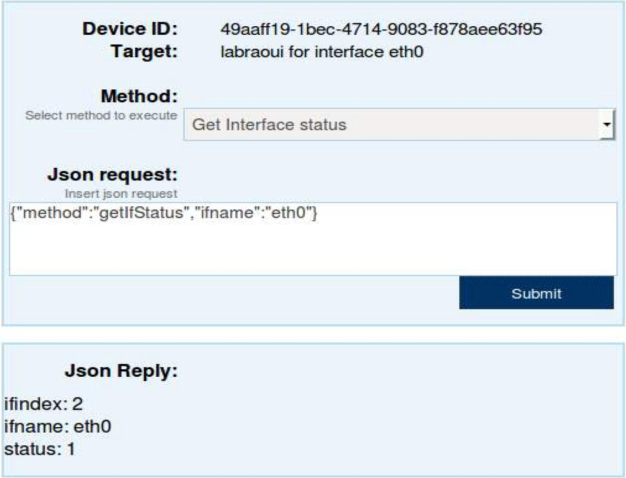
Neighbor list:

Device	Tx (B/s)	Avg Tx (B/s)	Rx (B/s)	Avg Rx (B/s)	Signal	MCS	HT
link	0	3724	0	3404			

Figure 12: Statistics covering specific information about the interfaces status

Commands:

This category of messages uses a JSON format packet to send commands (actions and requests), acknowledgments, and device information. Such messages may be initiated directly by software services or from the controller to setup the desired configuration or to get specific information. For example, to request WPA association or to configure router advertisements. The JSON format allow us to rely on the use of structured and flexible key/value pairs, it is lightweight compared to the XML format, and easily handled by computer programs.



Device ID: 49aaff19-1bec-4714-9083-f878aee63f95
Target: labraoui for interface eth0

Method:
Select method to execute
Get Interface status

Json request:
Insert json request

```
{"method":"getIfStatus","ifname":"eth0"}
```

Json Reply:

```
ifindex: 2
ifname: eth0
status: 1
```

Figure 13 : extended, dynamic, and resilient remote device configuration capabilities

Events:

Some specific events occurring at the device level may require immediate attention of the controller and services, e.g., a wired Ethernet cable unplugged or a new WiFi USB dongle plugged. Devices send “event” packets to the controller detailing the reason of the event. In the same way, the controller may catch some specific events such as the disconnection of a network device. The controller might not know how to handle all type of events but specific software services could. In all cases, events are sent to software services.

3.3.2.1.2 Communication model

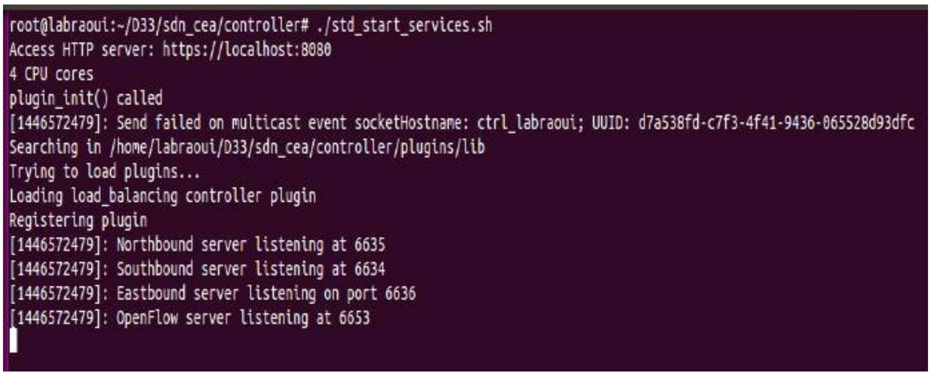
Considering NEON as the main southbound protocol, the communication model that we consider in this document is structured as follow (see Figure 15 for a global overview):

SDN devices:

Devices are configured to start the user space NEON application at boot-up. They establish a NEON connection to the controller which allows adding and removing infrastructure devices on the fly. The application regularly scans the local configuration, available network interfaces and handles NEON statistic/command request packets. After command validation, the application is able to configure and start other applications such as Open vSwitch. Each device generates a unique identifier (UUID) on 128 bits and UUID collision is prevented by the controller.

SDN Controller:

The controller maintains NEON connections with all SDN devices through regular statistics polling (heartbeat mechanism). Publishing groups are created by the controller to deliver different classes of event packets to services (subscribers). The northbound API relies on the command packets format (JSON). When the target of the command is a specific infrastructure device, the controller validates the service credentials and transfers the command packet to the right device.



```
root@labraoui:~/D33/sdn_cea/controller# ./std_start_services.sh
Access HTTP server: https://localhost:8080
4 CPU cores
plugin_init() called
[1446572479]: Send failed on multicast event socketHostname: ctrl_labraoui; UUID: d7a538fd-c7f3-4f41-9436-065528d93dfc
Searching in /home/labraoui/D33/sdn_cea/controller/plugins/lib
Trying to load plugins...
Loading load_balancing controller plugin
Registering plugin
[1446572479]: Northbound server listening at 6635
[1446572479]: Southbound server listening at 6634
[1446572479]: Eastbound server listening on port 6636
[1446572479]: OpenFlow server listening at 6653
```

Figure 14 : screenshot of the controller at launching moment

Services:

Services are connected to the controller through a TCP/IP connection and use the northbound API to query information about active network devices. They receive events and use the defined NEON protocol action messages to configure and get specific information about devices using the device's UUID as a target. However, those packets are intercepted, filtered, and validated by the controller.

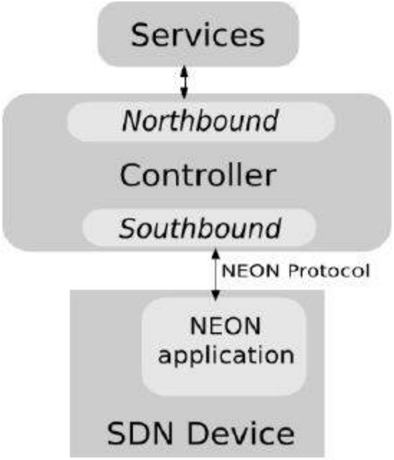


Figure 15 : Overview of NEON SDN reference architecture.

3.3.2.1.3 *The web page*

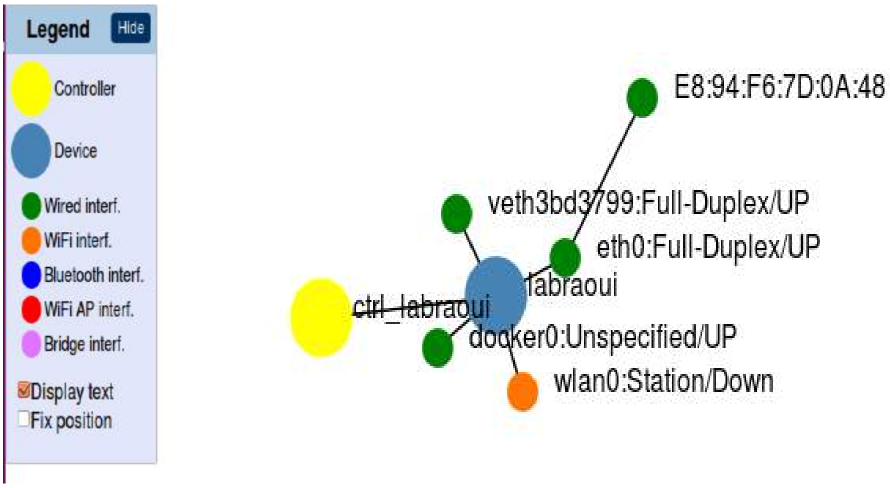


Figure 16: Graphic representation of the network using the web page

The web page shows the different available network functions and network functions currently undertaken with their IP address, status (in progress, stopped, turned off, in start-progress) and the information that they have recovered to their manager. It is possible from this page, remove or restart these network functions. It is also possible to instantiate new functions from the functions of sets available. The web page uses to display the informations a catalog created by the orchestrator that lists the available functions and those currently available. To perform the user actions it communicates directly with the json server.

Among the services offered by the web page, we can see in Figure 14 the possibility to visualise dynamically the overall view of the network including the different hardware characteristics of each device.

3.3.2.2 PMIPv6 as a SDN service

This subsection describes the deployment of PMIPv6 protocol as an SDN service. After briefly presenting the basics of PMIPv6 protocol, we detail the underlying procedures. The objective is to isolate key mechanisms to better emulate them in an alternative PMIPv6 service based on SDN using NEON.

3.3.2.2.1 PMIPv6 standard overview

PMIPv6 [15] is a network-based mobility management protocol that handles unchanged UEs mobility in IPv6 networks with no required interaction. Two functional elements constitute the PMIPv6 domain (see Figure 17). The Local Mobility Anchor (LMA) function is located on a core-network gateway. It registers the current point of attachment of all nodes and is the topological routing anchor point of the IPv6 address prefixes (i.e., Home Network Prefixes, HNPs) that it assigns to UEs. The Mobile Access Gateway (MAG) function is located on the infrastructure access routers.

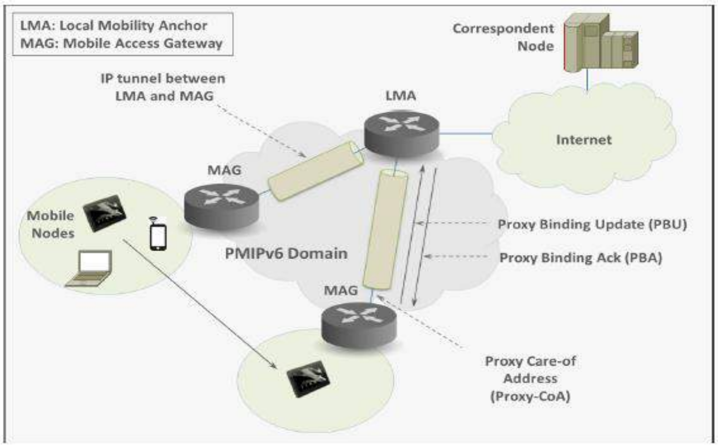


Figure 17 : The PMIPv6 network architecture

The MAG is the IP level point of attachment for mobile nodes and is responsible of informing the LMA about the presence of mobile nodes. It also advertises IPv6 prefixes to mobile nodes. To deliver mobile nodes traffic, the LMA and the MAG (on which the node is currently attached to), use an IPv6-in-IPv6 tunnel. This is to prevent incorrect routing by intermediate routers. As a result, UEs IPv6 addresses may not be topologically correct.

At node attachment, a Proxy Binding Update (PBU) signaling message is sent from theMAG to the LMA. The PBU provides node identification information, i.e. Mobile Node Identifier (MNID). Upon reception, the LMA checks if the node was already registered to another MAG. The LMA establishes an IPv6-in-IPv6 tunnel towards the new MAG, updates the routing tables to divert traffic towards the node through the tunnel. The LMA eventually sends configuration information (mainly the HNP) to be advertised by the MAG to the node in a Proxy Binding Acknowledgement (PBA) signaling message.

At PBA reception, the MAG adapts routing rules accordingly, i.e., data packets from the node IPv6 prefix must pass through the MAG-to-LMA tunnel. The HNP assigned by the LMA is advertised on the link between the MAG and the mobile node. By advertising the same HNP to the mobile node after each attachment, the mobile node keeps the same IPv6 address after handovers and is able to maintain on-going communications seamlessly. PBU and PBA messages may contain several mobility options and flags that have not been described in this section.

3.3.2.2.2 Adaptation to the SDN framework

As described, PMIPv6 control plane consists in two signaling messages: the PBU and PBA. These messages trigger routing and configuration subroutines at both MAG and LMA, in order to handle nodes traffic. This subsection isolates the set of elementary actions that MAG and LMA perform in the control plane, the required information, and sketches how a PMIPv6 SDN service could perform accordingly.

PBU generation by the MAG:

A PBU is generated at node attachment and its main content is a node identifier (MNID). There are several means to detect the attachment of a node to a wireless access-point and we will consider them in the context of an IEEE 802.11 interface. For instance, one may monitor an LLC (Logical Link Control -

Layer 2 IEEE 802.2) association message, the reception of MAC sublayer management entity (MLME) messages (e.g., the MLME ASSOCIATE.request), or the reception of a ND Router Solicitation message. In all cases, those messages provide an identifier for the node which is its MAC address.

When considering NEON, these attachment messages trigger an event reported to the controller. The event message could indicate if this is an association, re-association, or a disassociation as well as the MAC address of the node.

Received by the controller, the event message will be automatically published to the PMIPv6 service.

PBU reception by the LMA:

The LMA associates the MNID (the MAC address in our case) with the routing state in a single client specific database record. Basically, the MNID is linked to the client HNP and the current point of attachment (MAG). The database record is either created (in case of first association), altered (in case of re-association), or deleted (in case of disassociation). The record creation is performed along with the HNP allocation. In NEON, these actions are handled by the PMIPv6 service (which hosts the clients' database) as a set of elementary commands. First, the event type sent by the device triggers a creation/update/deletion of the record entry that identifies the client. The device behaving as the LMA is sent two commands: one to establish a tunnel towards the current MAG; a second to divert HNP-destined traffic through the tunnel.

PBA reception by the MAG:

The MAG uses the HNP within the PBA to configure the client and handle its traffic. Namely, (1) a source specific route entry points towards an IPv6-in-IPv6 MAG-to-LMA tunnel for the traffic originating from the HNP, and (2) periodic RA of HNP to the client. In practice, source-based routing could be achieved using a rule that states that traffic originating from HNP goes to the LMA.

In NEON, such a procedure could be performed by the help of the PMIPv6 service supervision. The device that detects the event and triggers the emulated PBU-PBA exchange is now commanded to set a rule to handle HNP source-based routing. The PMIPv6 service also orders the serving MAG device to set up the local RA daemon in order to announce the HNP.

In case of a handover, the previous MAG is also ordered to uninstall the client configuration. Basically, the latter device would delete the client-related routes and rules. The RA agent would also stop announcing the HNP of this client.

3.3.2.2.3 PMIPV6 illustration

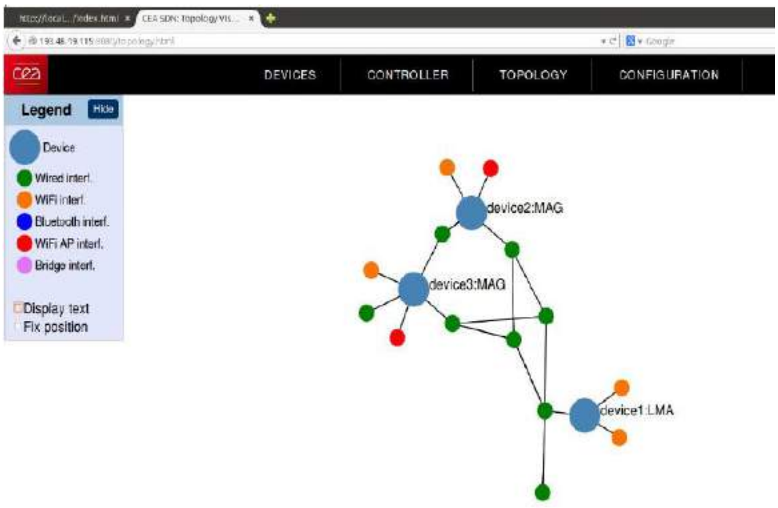


Figure 18: PMIPV6 service deployement

One of the devices is selected as LMA (Local Mobility Anchor) that is the equipement in charge of associating an IP adress to each device connected. Others are chosen as MAG (mobile Access Gateway) to serve as an access point to which terminals can then connect. We can see in Figure 18 the MAG devices playing the role of access points.

With the virtualization of the PMIPv6 procedure, the service deployment is then automated: just launch the creation of the PMIPv6 VNF in order that each device becomes configured after a few seconds.

3.3.3 Main benefits, results & comparisons

We evaluated the performance of PMIPv6 as an SDN service compared to a native implementation.

The objective of this performance analysis is to discriminate whether the service implementation of the control plane has a significant impact on the client data plane. The following performance results are subject to wireless specific conditions. It is noteworthy to emphasize that the initial objective of this evaluation is to show the transparency of both deployments to the end client, proving our assumptions.

To that end, our testbed setup includes a core infrastructure composed of 4 laptops and one client. In the native deployment, referred as “PMIPv6-native”, 1 laptop serves as an LMA, while the 3 others run MAG functions. In the PMIPv6 SDN service deployment, referred as “PMIPv6-service”, all laptops are configured to run the NEON application and protocol. The controller and the PMIPv6 service are located on the laptop serving as the LMA. The client is programmed to perform periodic handovers between configured MAGs every 10 to 15 seconds. Even though the SDN elements are colocated on the physical machine hosting the LMA, we assume that a more distributed architecture should ensure low latency of the control plane. This part aim to expose the responsiveness and flexibility of an SDN API compared to a native one, in this way the controller’s placement in an SDN topology is out of the scope.

Our data plane consists in a UDP dual iPerf traffic (with a target bandwidth fixed to 500 KBytes/s) generated between the client and the laptop serving as the LMA, acting as a destination.

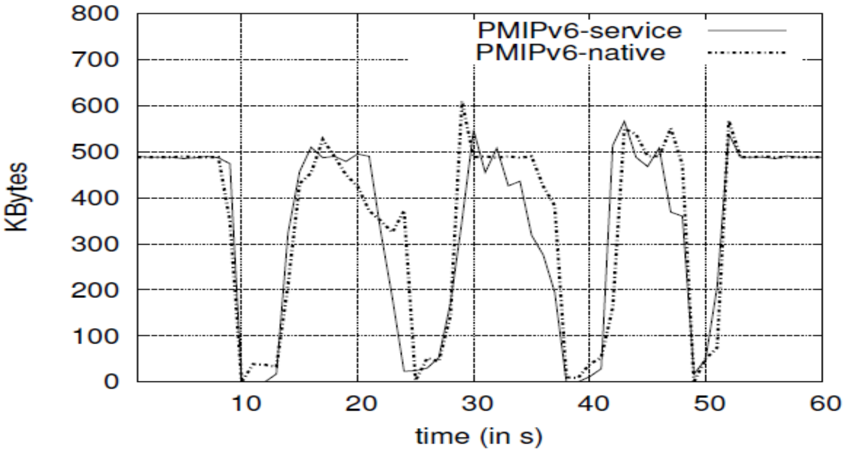


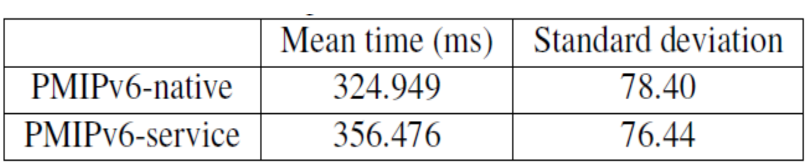
Figure 19 : Evolution of the effective bandwidth with time after successive client handovers in both native and SDN service deployments

3.3.3.1 Effective bandwidth:

Figure 19 presents the effective traffic bandwidth as received by the destination node in both native and SDN service deployments. One might observe consecutive drops in the bandwidth which reflects the di erent handovers planned for our evaluations. Recall that the target UDP bandwidth is 500 KBytes/s. The important result here is that PMIPv6-service achieves comparable performance than PMIPv6-native. Indeed the handover durations (around 2 seconds) and achieved bandwidth are quasi-similar.

Furthermore, the resulting average bandwidth for PMIPv6-native is 355 KBytes/s, whereas the PMIPv6-service achives 341 KBytes/s. With 4 handovers on a total duration of 60 seconds, the maximum average bandwidth is up to 433 KBytes/s. The values express a short difference between the 2 implementations, remembering that optimizations to reduce the latency could be considered.

Table 3 : Average and standard deviation time of client (re)configuration



	Mean time (ms)	Standard deviation
PMIPv6-native	324.949	78.40
PMIPv6-service	356.476	76.44

3.3.3.2 Configuration overhead:

Table 3 exposes the mean latency with standard deviations between the physical-level association to the MAG and the reception of the RA, in both native and SDN service deployments. Such metric reflects the overhead cost of the PMIPv6 protocol (in terms of time). Recall that PMIPv6-native operations only require the transmission of PBU and PBA messages, while our SDN service needs 8 elementary control messages to configure the LMA and MAG. With the native deployment the average latency is 325 ms and the standard deviation is 78. With PMIPv6-service the resulting average latency is the little bit higher at 356 ms with a standard deviation is 76. Clearly the gap in performance (+9.5% of time) is neglectable.

From those results one can consider that our PMIPv6 SDN service in conjunction with NEON achieves the same level of performance than the native approach.

3.3.4 Recognized issues and future work

Among the recognized issues in the context of our use case, the main one is the extra delay incurred by the position of the SDN controller. Indeed, whereas in current structures, all the network elements (LMA, MAG...) are placed to optimize the delay of proceedings, in SDN context, the controller can be instantiated anywhere in the network. Consequently, the flexibility offered by the SDN is unfortunately associated with an extra time in function of the distance between the controller and the mobile network elements.

Moreover, taking into account that the standard has not defined yet the requirements concerning the northbound protocol, we don't have yet a clear view of the northbound protocol implementation.

3.4 SDN core for mobility management and heterogeniety

3.4.1 Definition

SDN centralizes network intelligence and introduces programmatic tuning of network equipment at run time. Thus, the concepts of SDN can be exploited for deploying multiple radio access technologies to enable dynamism in future wireless networks. For example, WLAN is an essential part of the overall business model and users demand that it be available all the time, transparent, and provide the same benefits of the wired network. WLAN has a unique set of challenges that makes the need for centralized control with distributed intelligence. As users are constantly on the move, connectivity must be ubiquitous and the rules for connecting must transparently and dynamically change as they roam in the wireless environments. SDN enabled WLAN can abstract network services from the underlying physical resources, whereby implementing multi-tenancy network slices to have distinct policies and service profiles. Such services can readily be offered on a temporary basis, such as video feeds for a sporting or news event especially in offices or universities. Beyond WLAN, enabling other wireless networking methods helps multi-band users to roam seamlessly and communicate with different systems. Early last decade, there was a boom in cognitive networking. 5G proposes bring all different network technologies under the same umbrella for simplified management. Cognitive networking is an essential part in the 5th generation radio access networks as it could be used to respond the growing demand of mobile Internet data traffic by mechanisms for access to shared spectrum and improved usage of shared network resources [19]. SDN architecture is one of the best feasible frameworks for sharing the resources in a heterogeneous radio network.

3.4.2 Proposed solution and architecture

Our contribution lays in the development of an OpenFlow testbed and feasibility study of integrating systems across SDN core. Further, we present the results of seamless mobility leveraged by network virtualization in the OpenFlow domain. Finally, we present the needfulness of access control in wireless SDN.

We use the concepts of Cognitive networking to enable the convergence of differenet radio access technologies based on the availability of free resources and user requirements. Cognitive networks have a high potential to deliver the benefits of controller redirection of resources in SDN. However, this needs cognition process to be mapped to the SDN control plane. With this, the controller gets access to current network statistics which enables correct provisioning of the available network resources. Therefore, we present the Software Defined Cognitive Networking (SDCoN) framework to merge the concepts of SDN and cognitive networking for dynamic resource sharing in future wireless networks.

3.4.2.1 Architecture Components

In order to present the architectural framework and corresponding components of SDCoN, the cognitive networking and SDN architectures are mapped with respect to each other in Figure 20. In legacy cognitive networks, the end-to-end goals of a network are specified by users, applications or resources. A cognitive specification language is used as a medium to communicate those end-to-end goals to the cognition process in the cognition layer. Cognitive process is the actual learning and decision of appropriate response based on the observed network behavior [18]. The cognitive process receives the network information from the network status sensors or through network application programming interfaces (APIs) from configurable network elements called Software Adaptable Network (SAN) elements [18].

In SDCoN, a network operating system or the SDN controller maps the entire network to services and applications that are implemented on top of the control plane. The end-to-end goals are realized in the form of SDN applications. Similarly, the cognitive engine is implemented in the SDN application plane that receives the network status information from a cognitive process module in the controller. The cognitive process module is implemented in the control plane since adding a control plane functionality in SDN require writing a software-based logic in the same plane. The cognitive process obtains the network state information through a standard south-bound API from the infrastructure. Hence, the cognitive engine which implements the cognition and resource allocation process uses the cognitive module in the controller to deploy new configurations in the infrastructure plane.

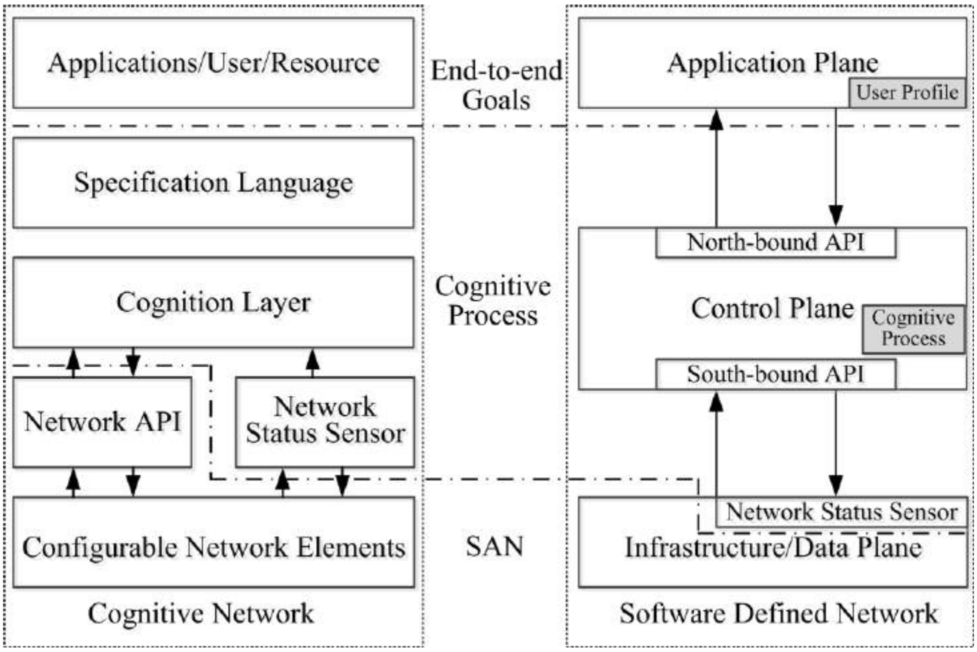


Figure 20: Cognitive network and corresponding SDN architecture

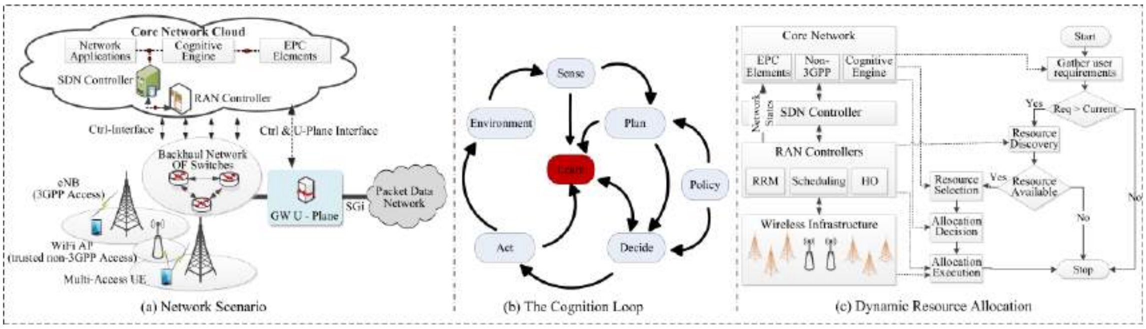
SAN [18] comprises of network elements which are tunable at run-time and provide action space for the cognitive process. SDN introduces programmable interfaces in networking elements to (re)program or adjust them at run-time from the logically centralized control plane. Making the forwarding devices simple e.g. the OF switches, the network is made more adaptable since the networking policies are imposed by the control plane software rather than by the hardware itself or its firmware. We envision reconfigurable eNodeBs (eNBs) or access points having only physical layer transmit-receive capabilities while the remaining control functionalities including base-band processing being moved to separate devices in the operator cloud as described in the following section.

Software defined radio (SDR) is a transceiver technology in which the communication functions are realized as programs running on a suitable processor allowing different levels of (re)configurations within the transceiver [20]. Cognitive radios (CRs) use SDRs to dynamically adjust to specific frequency bands at run-time based on the decisions from the cognitive engine. In SDCoN, CRs configure SDRs based on the instructions from the cognitive engine in the control plane.

In the SDCoN architecture, the SDN controller is extended to cooperate with the existing Evolved Packet Core (EPC) of LTE network to enable cognition in the network backhaul as well as the access technologies. The SDN controller is coupled with Access Network Discovery and Selection Function (ANDSF), Media Independent Handover (MIH) protocol, and Mobility Management Entity (MME) to enable proactive mobility between eNBs and other access technologies. ANDSF enables user equipment (UE) to discover various radio access networks and assists in their attachment, whereas, MIH supports proactive vertical handovers in heterogeneous environments. Furthermore, OF switches are used as the cell site routers to enable the controllers to retrieve physical resource information of each cell at run-time.

3.4.2.2 Architecture Framework

In this framework we lay down the architecture for dynamic configuration of radio terminals to select and operate in optimal wireless system among the available options. In this architecture, the decision making is centralized in the core network cloud with the help of SDN concepts for dynamic spectrum allocation, coordination among different radio access technologies (RATs), radio resource optimization and mobility management. The cognitive engine is implemented in the SDN application plane that receives information about the occupied and free spectrum in all the access technologies from the radio access network (RAN) controllers through the SDN controller. The architectural framework and cognitive resource allocation process is presented in Figure 21.



(a) Network Scenario (b) The Cognition Loop (c) Dynamic Resource Allocation

Figure 21: Software defined cognitive network framework.

In SDCoN, the SDN controller controls and synchronizes communication between the EPC elements, the RAN controller and cognitive engine as shown in Figure 21 (a). The RAN controller controls the monitoring and management of access networks, and forwards the physical resource information to the SDN controller through a standard interface (API). For example, the RAN controller polls the channel quality indicator (CQI) or received signal strength indicator (RSSI) values from each cell and provide that information to the SDN controller which in turn forwards it to the cognitive engine. This process of sensing the environment and learning constitutes the major part of the cognition loop (Figure 21 (b)). The cognition loop [18] is a typical operational duty cycle or feedback loop of cognitive systems.

The cognitive process of providing free resources to users based on the user QoS demands is shown in Figure 21 (c). If a user is transmitting or receiving data and meanwhile the signal strength falls below a certain threshold, then action (e.g. spectrum mobility or vertical handover) can be initiated. The cognitive engine gathers information about the available resources from the RAN controller, and user profile and QoS settings from the core network to plan according to the user and service provider policy. This constitutes plan, policy and learning of the cognition cycle. If resources (free frequency slots) are available, the cognitive engine forwards the allocation decision to the RAN controller via the SDN controller. Hence, the allocation execution is carried out in the infrastructure plane through the RAN controllers. This constitutes the decide and act steps of the cognition cycle [18].

As shown in Figure 21 (c), the RAN controller provides periodic updates to the SDN controller that is forwarded to the cognitive engine. Thus, resource reallocation can be carried out at run-time to enable dynamic spectrum access. Furthermore, in this approach allocation of more bandwidth in the backhaul network to particular users, eNB or access points is less complex since OF switches are used in the backhaul as well as in the cell sites. As the cognitive process in the control plane receives implementation configurations from the cognitive engine, the SDN controller modifies the flow rules in the OF switches simultaneously. Therefore, the deployment of cognitive networking does not require complex cross-layer designs for synchronizing the edge and backhaul network components.

3.4.3 Main benefits, results & comparisons

The measurements in these experiments are based on the SDCoN testbed which is developed with a focus on the possible integration of SDN and cognitive radio network. Hence, these results present the preliminary evaluation of the SDCoN architecture. We have performed two experiments, one for TCP throughput measurements using Iperf [13] to collect throughput statistics. Second, we used Session Initiation Protocol (SIP) [22] for testing voice sessions among the clients. The throughput measurements show the performance improvement by opportunistic spectrum usage aided by dynamic spectrum access and the SIP experiments portray cellular network like calls.

The implementation of cognitive WARP is rate limited which happen to drop off the excess packets. We measure the expected theoretical limit to prove the bandwidth limitation. WARP is using 10MHz bandwidth with Fast Fourier Transform (FFT) size of 64 and cyclic prefix of 16 samples. This means that one OFDM symbol takes altogether 80 samples and with 10MHz sampling, OFDM symbol duration is 8μs. Not all sub-carriers are used for data transmission, so 48 out of 64 sub-carriers carry data. We use QPSK (2 bits/sub-carrier) modulation and coding rate is 1 (no coding). This means, to transmit (48x2) bits, it would theoretically take 8μs, thus, the maximum theoretical throughput can be calculated as, 12Mbit/s. Since, resources are equally divided to uplink and downlink, maximum one way throughput is 6Mbit/s.

The throughput measurements in our experiments are taken between an OF WLAN client and a CRN client. First, we generated TCP traffic between the OF based WLAN and cognitive radio network on 1 and 8 frequency slots. Initially, a single frequency slot was used in experiments to see the performance improvement due to the cognition process using eight frequency slots. The average TCP throughput on a single frequency slot remained 1.15 MBps, whereas, using all eight frequency slots simultaneously, the average throughput increased to approximately 3.5 MBps as shown in Figure 22. The average round-trip-time (RTT) between these two clients remained well below 20ms as shown in Figure 23. The RTT is measured for a TCP packet and the corresponding ACK between the two clients.

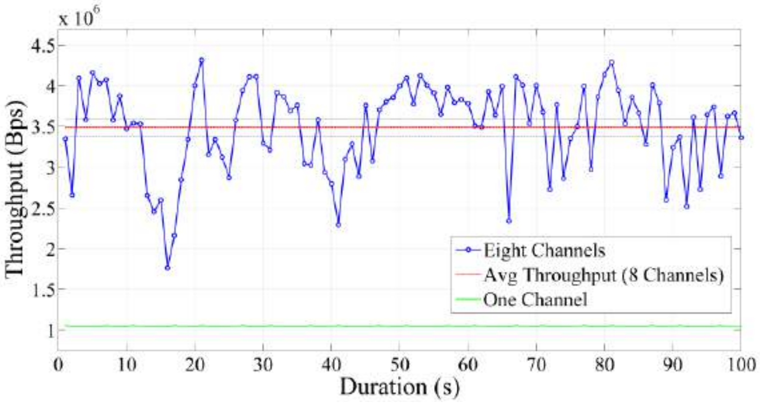


Figure 22: TCP throughput between OF and cognitive radio network

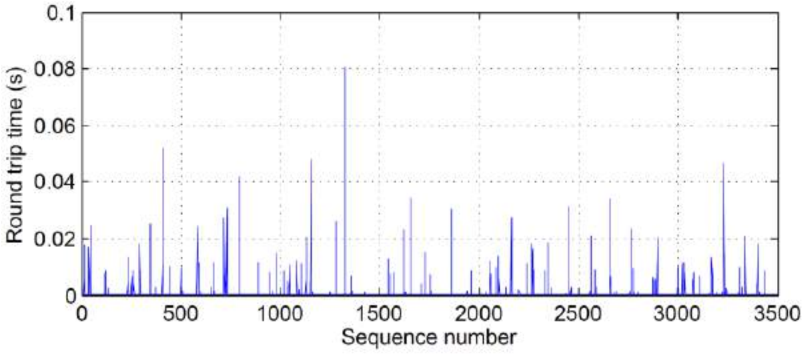


Figure 23: Round trip time between the OF and cognitive clients

SIP [22] is an application-layer signaling protocol used for creating, modifying and terminating sessions of VoIP calls, multimedia distribution, and multimedia conferences. We used SIPp [23] which is a free Open Source test traffic generator for SIP protocol. SIPp runs basic user agent scenarios with multiple calls that enable measuring complex call flows. We generated SIP voice traffic between the SDN wireless and cognitive clients. Changing the size of the contention window as a QoS parameter to avoid collision, we performed experiments having increment of 50 in call rate with each running for a duration of 60 seconds having different contention window sizes and burst lengths.

Note that as the size of the contention window increases, collision on multi-user channels decreases due to distribution of traffic into larger time space. Burst length is the time-duration of bursts in milliseconds during which the data channel scheduler handles the burst. Traffic burst occurs when an inter-arrival time of packets decreases. Hence, in the non-optimized setup, the contention window size remained 7 and the burst length remained 1.5. In voice-optimized setup the contention window size is 15 with burst length of 0. We reduced the burst length in order to avoid using resources on scheduling of packets.

We measured the call drop rate with respect to increasing the call rate with steps of 50 from 50 calls per second (cps) to 1000cps. Drop in call rate occurs due to a number of reasons including absence of radio coverage, cell overload, faulty handover and signal interference. Having addressed the rest of the possible reasons, call drops in our scenario occurred due to signal interference. With changes in QoS parameters, i.e., increase in contention window size and decrease in burst length, the drop in call rate occurred at 450cps as compared to call drops that start at 300cps for smaller contention window and higher burst lengths as shown in Figure 24.

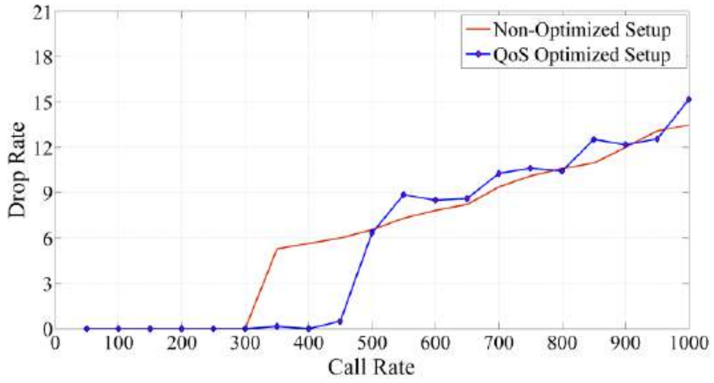


Figure 24: Call drop-rate for non-optimized and QoS-optimized settings

Next, we measured the response time (RT) and response time deviation (RTD) in the same setup. These measurements can be taken with the help of the SIPp commands, such as the INVITE and ACK messages [23]. Response time deviation is useful to evaluate the performance of a radio network for different delay sensitive services, such as voice calls and interactive video conferencing, etc. Response time and response time deviation increase with the number of call rates. It is visible from the Figure 25 that the QoS optimized setup has better results.

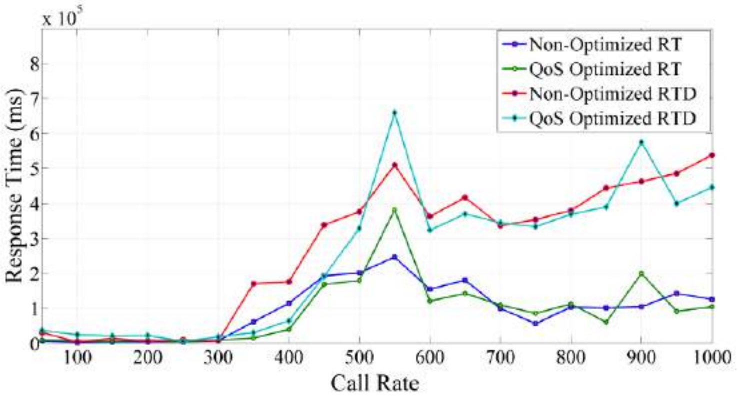


Figure 25: Response time (ms) measurements for different QoS settings

In the above setup, the cognitive platform is rate limited yet we can see the performance improvement due to the cognition process. The lower throughput shown in Figure 22 is attributed to frequency planning and scheduling which we aim to enhance it further in our future work. As it can be seen from the results, the number of successful calls and simplicity in QoS optimization from a centralized control plane shows the feasibility of SDN based cognitive wireless networks. An important result, however, is heterogeneous networking enabled by SDN where different access technologies are integrated for resource sharing by a centralized SDN control plane.

3.4.4 Seamless Mobility

To make smooth mobility between multiple access points and wireless interfaces, we configure the same SSID on all the interfaces. Therefore, for the evaluation, we configure a wireless client that forces the WLAN interface of the device to rejoin the network when another access point with same SSID of better signal strength is found in the same coverage region. We use OpenFlow enabled WLAN for the experiment purpose. To prove seamless handover between APs, the RTT between two hosts that are associated with different Aps is measured. RTT is a good example to observe the delay in handover when a client is on move. With Iperf and Wireshark, RTT and Jitter measurements with seamless mobility in OpenFlow enabled WLAN domain. The RTT measurements in Figure 26 are collected. During a time interval of 10 seconds, we perform two handovers. However, not having significantly high RTTs proves the transitions between the APs is seamless. The ITU G.114 specification recommends less than 150 millisecond one-way end-to-end delay for high-quality real-time traffic such as voice. Moreover, the TCP and UDP throughput during the transition process is shown in Figure 27.

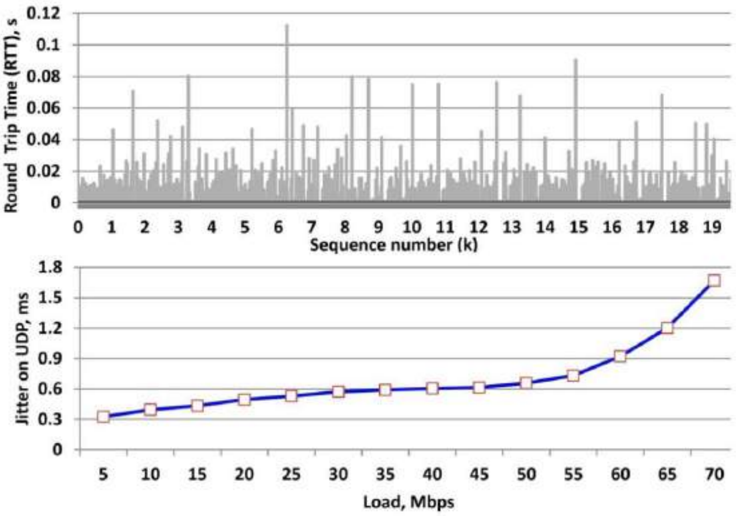


Figure 26: RTT and Jitter measurements with seamless mobility in OpenFlow enabled WLAN domain.

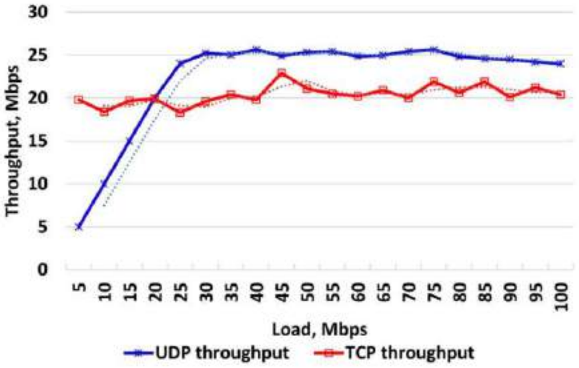


Figure 27: Throughput measurements with seamless mobility in OpenFlow enabled WLAN domain

3.4.5 Recognized issues and future work

SDN, however, has its own limitations of scalability and latency which can be solved by effectively delegating controller responsibilities to RAN or domain specific controllers in large heterogeneous networks. Moreover, virtualizing the cognitive engine and using a hierarchy of controllers will lead to highly scalable, robust and secure SDCoN architectures.

4. Architectural Modifications, Integration Points and Possible Performance Improvements

4.1 Media independent OpenFlow-based mobility management for heterogeneous SDMNs

4.1.1 Updates on architecture mapping

Figure 28 introduces the additional elements to integrate in the SIGMONA SDMN reference architecture, required by the proposed network-based mobility management. The scheme relies on the OpenFlow toolset and provides efficient vertical handover (VHO) support in a media independent manner aiming to support heterogeneous access environments. IEEE802.21 MIH / ANDSF is used for VHO preparation and completion and for the proactive operation of handover optimization and multi-access capabilities.

The SDN controller, the LTE and Wi-Fi access nodes are all coupled with the 802.21 MIH / ANDSF enforcer functions in order to manage and control mobility independently of the access technology, optimize path based on the wireless and wired link conditions, provide seamless flow-level handovers, eliminate the tunneling overhead and reduce the overhead. All of this can be achieved in a completely network-based and operator-centric way, leaving only minimal tasks on the UEs, namely providing interfaces to control, query and manage access links with the IEEE 802.21 / ANDSF functionality.

Network information server instances (ANDSF / MIH MIIS) provide static information about the access environment used by the mobility manager integrated within the SDN controller for handover decision.

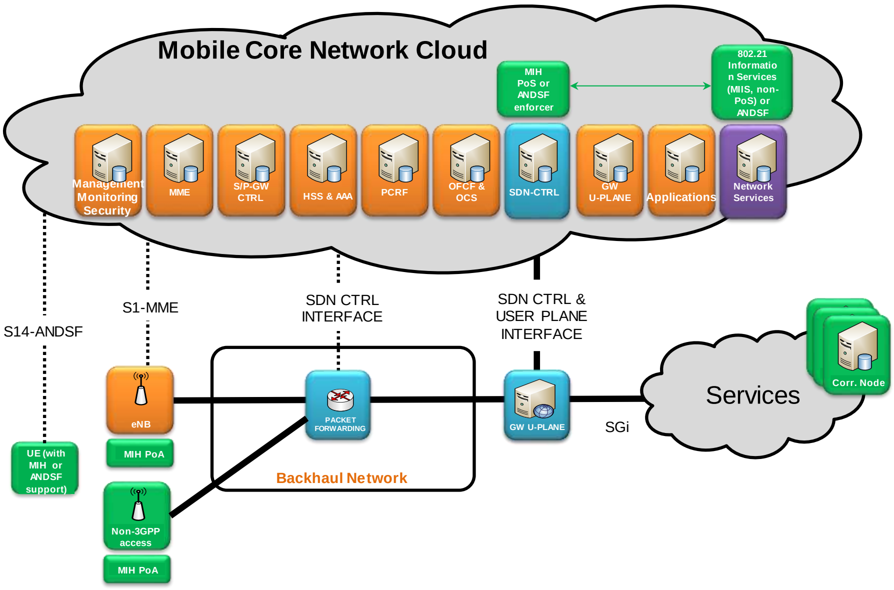


Figure 28: Main SDMN elements and their mapping in the proposed media independent network-based OpenFlow mobility management framework

The data plane is based on standard OpenFlow switches, while the control plane has multiple added elements as introduced above. It is important to notice that SDN controller integrates the mobility manager. Figure 29 shows the mapping of this framework into the ETSI NFV architecture as a result of multiple component integration utilizing SDN/NFV technologies. This mapping also identifies the main communication channels: 1) IEEE 802.21 MIH signaling between PoAs, PoSs, non-PoSs, etc., 2) southbound interface towards the SDN switches using OpenFlow API.

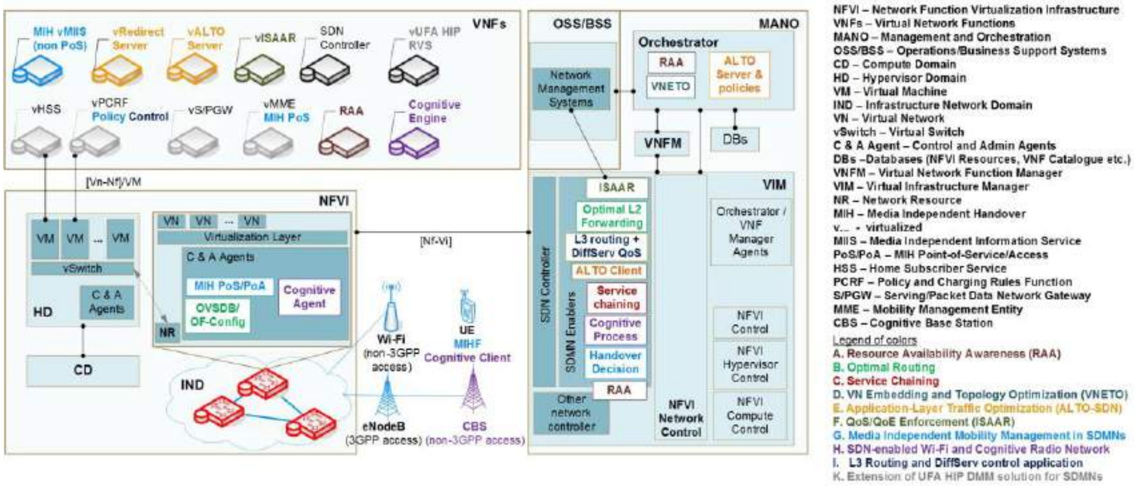


Figure 29: ETSI NFV mapping of the proposed SDMN media independent mobility management scheme (letter G. and light blue marks the elements/functions in question)

4.1.2 Detailed interface descriptions

The IEEE 802.21 MIH standard [5] **Error! Reference source not found.** introduces three main interfaces:

1. MIH_SAP: allows communication between the MIHF layer and higher layer MIHF users
2. MIH_LINK_SAP: allows communication between the MIHF layer and the lower layers of the protocol stack
3. MIH_NET_SAP: allows communication between remote MIHF entities

The southbound interface deals with the communication of the SDN informations required by:

1. forwarding
2. monitoring
3. resource management
4. statistics
5. network performance details

4.2 Extension of UFA HIP DMM solution for scalable and secure mobility management in SDMNs

4.2.1 Updates on architecture mapping

This research topic deals with the integration of Host Identity Protocol-based Ultra Flat Architecture (UFA HIP) concept [33] into SDMNs, in order to enable scalable mobility management. A natural evolution step of UFA HIP is to put UFA GWs into the mobile network cloud, and use SDN techniques for appropriate GW selection, optimal path selection between the source and target GWs during and after handover events, efficient support of different levels of GW distribution, more lightweight UFA GWs due to their virtualization on top of high-performance hosts. The benefit of integration of IEEE 802.21 MIH/ANDSF with the SDN controller supports and manages proactive handovers.

The main components needed by this mobility management solution are depicted on Figure 30: SDN supported HIP signaling overlay in the cloud, MIH Media Independent Information Service (MIIS) in the cloud, HIP-compatible UEs, CNs and UFA GWs distributed in the cloud. HIP and MIH signaling must be implemented between the GW nodes and the cloud. The mobility management is handled by the cloudified UFA HIP entitites and controlled by the SDN controller (UFA HIP based mobility management as a service) by the UFA HIP Coordination Agent placed as a Network Service in the SDN application layer. This agend handles all the coordintain functions and also manages IP address allocation and UFA HIP GW allocation as well.

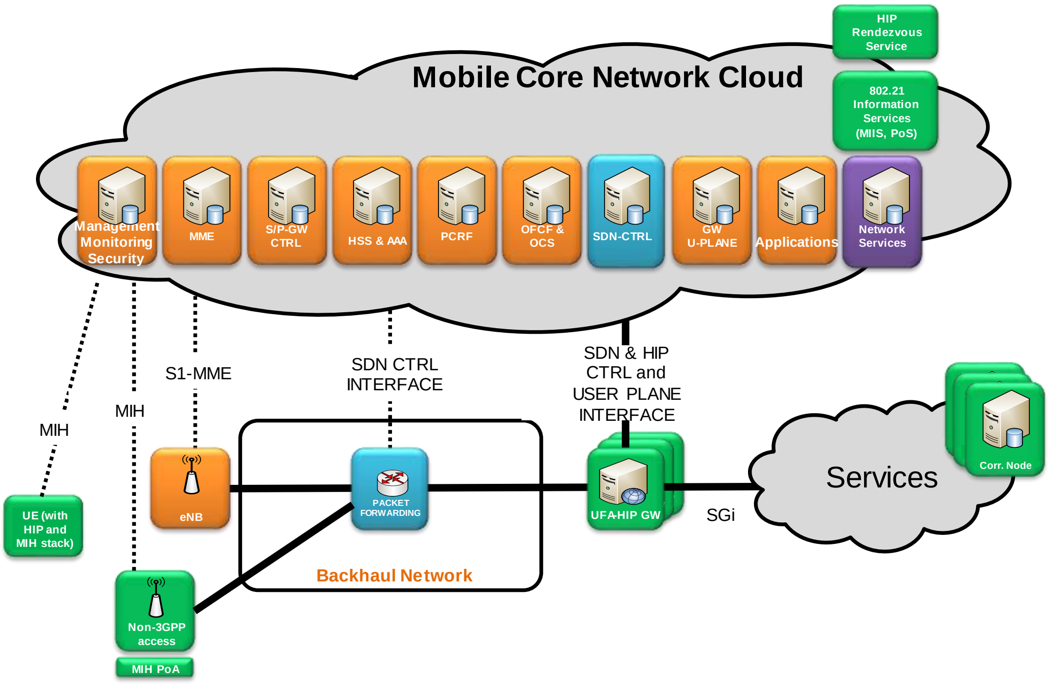


Figure 30: Main elements and their mapping in the UFA HIP SDMN integration scheme

The data plane is based on standard OpenFlow switches, while the control plane has multiple added elements as introduced above. It is important to notice that SDN controller integrates the mobility manager. Figure 31 shows the mapping of this framework into the ETSI NFV architecture as a result of multiple component integration utilizing SDN/NFV technologies. This mapping also identifies the main communication channels: 1) IEEE 802.21 MIH signaling between PoAs, PoSs, non-PoSs, etc., 2) southbound interface towards the SDN switches using OpenFlow API.

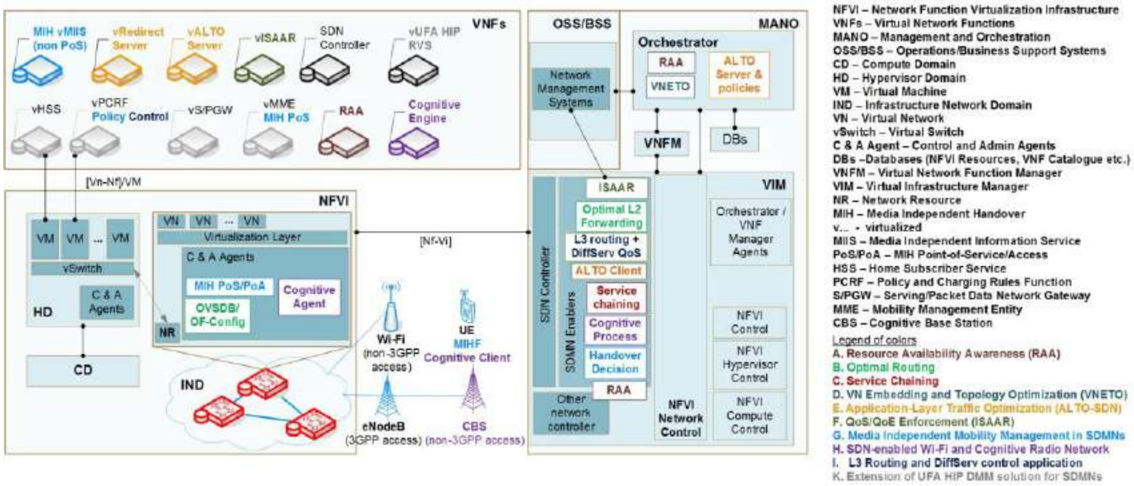


Figure 31: ETSI NFV mapping of the UFA HIP SDMN integration scheme (letter K. and grey marks the elements/functions in question)

4.2.2 Detailed interface descriptions

The IEEE 802.21 MIH standard [5] **Error! Reference source not found.** introduces three main interfaces:

1. MIH_SAP: allows communication between the MIHF layer and higher layer MIHF users
2. MIH_LINK_SAP: allows communication between the MIHF layer and the lower layers of the protocol stack
3. MIH_NET_SAP: allows communication between remote MIHF entities

The southbound interface deals with the communication of the SDN informations required by:

1. forwarding
2. monitoring
3. resource management
4. statistics
5. network performance details

4.3 Proxy MIPv6 in SDMNs

4.3.1 Updates on architecture mapping

Figure 32 presents the additional components to add in the SIGMONA reference architecture, required by the Proxy MIPv6 in SDMNs. Regarding the user plane (data plane), standard off-the-shelf SDN switches will be used with PMIPv6 protocol for data plane. The mobility management would be done using SDN functionality. Indeed, for the control plane, multiple services have to be added. In the SDN controller, we must primarily implement the handover management service. This latter are managed by the PMIP Coordinator Server placed on an upper layer (Network Services). In addition of its coordination task, it must manage other related services such as prefix allocation, IP address allocation and P/S-GW allocation. Communications between client-server applications are ensured through APIs.

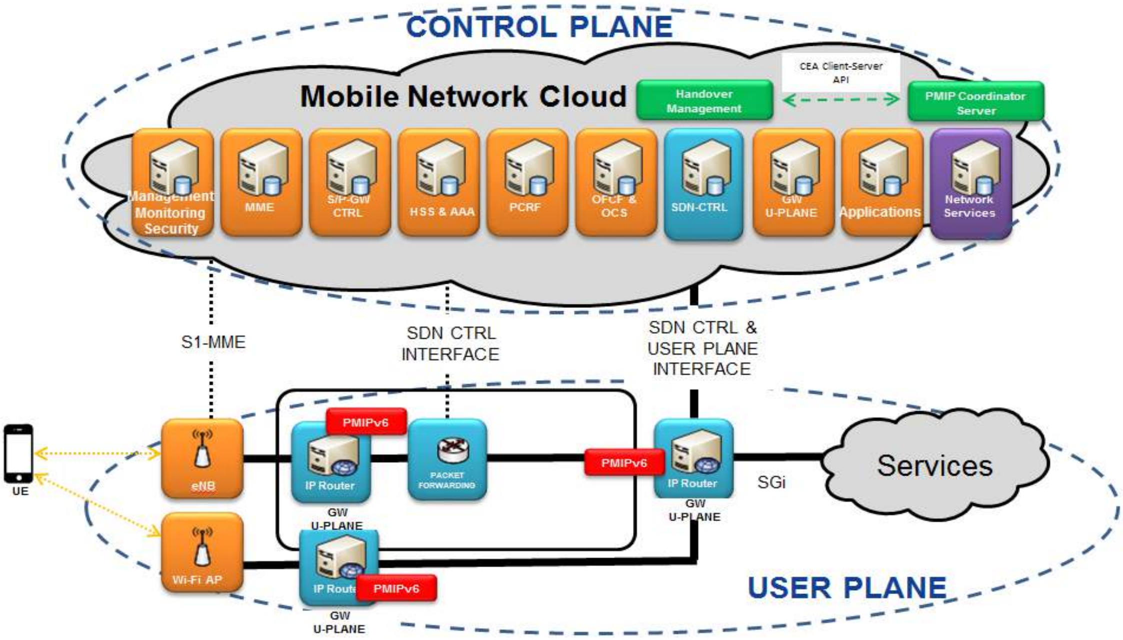


Figure 32 : Proxy MIPv6 in SDMNs (SIGMONA reference architecture)

A complete SDN- Proxy MIPv6 use case requires the following components in an SDN-layered reference architecture model, illustrated in Figure 33:

The main required functions in the SDN architecture layers are the following:

1. Infrastructure or datapath-layer: SDN switches, UEs, eNB and WiFi Access point
2. Controller-layer: handover management
3. SDN Application-layer: PMIP Coordinator Server
4. Management-layer: CEA Mobility Service Manager

These functions perform the following tasks:

- 1-2: Handover-related tasks
- 2-3: Coordinate and manage controller layer service according to the general structure described in 3.3.2.2
- 3-4: Resource status (monitoring) and inventory management unit to create chains

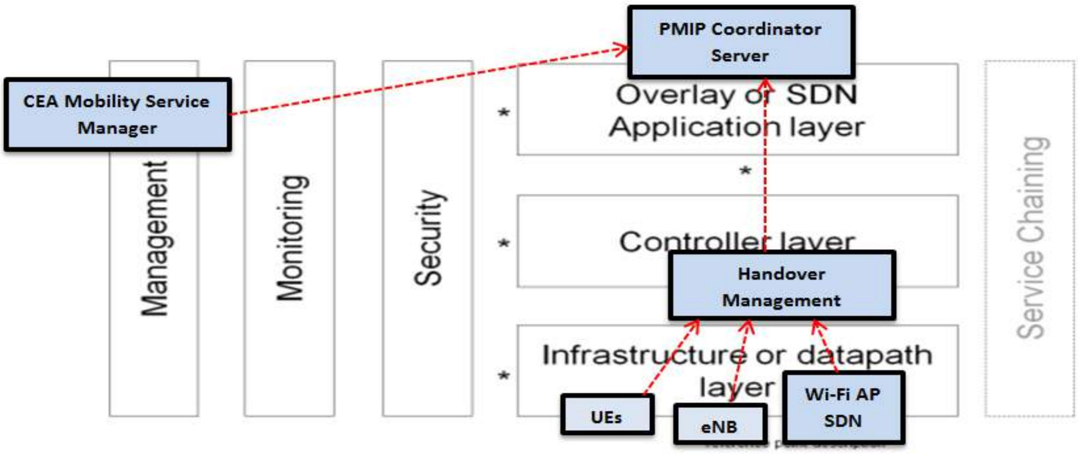


Figure 33: Proxy MIPv6 in SDMNs (SDN-layered reference model view).

Mapping to the ETSI NFV architectural framework:

Figure 28 includes the mapping of our research contribution into SDMN architecture. This mapping results in multiple components that utilize SDN/NFV as basic technology. The goal of such proposition is to ensure future proof solution taking into standardization guideline considerations such as the ETSI ISG NFV architectural framework.

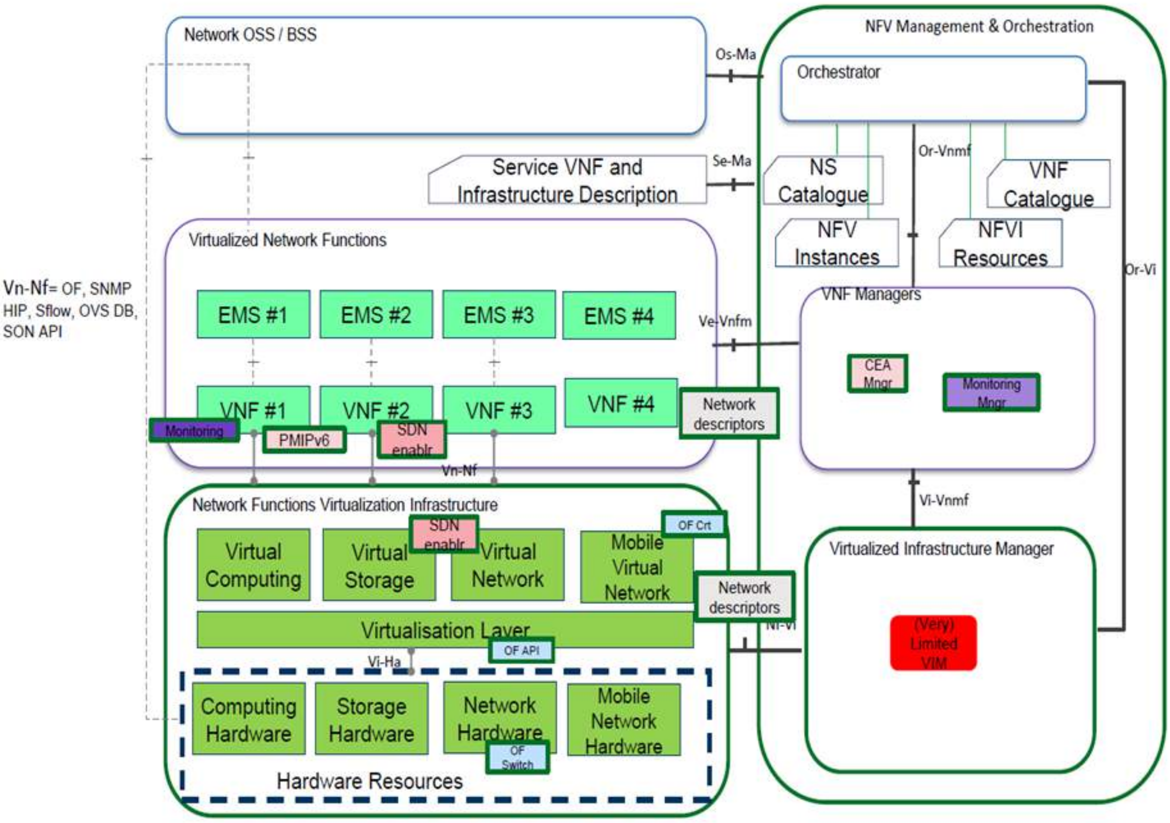


Figure 34- Proxy MIPv6 in SDMNs (ETSI NFV architectural framework)

Three communication channels have been identified:

- The northbound interface to provide communication to VNFm, orchestration services and Hypervisor network components,

- The East/west interface to provide communication between SDN controllers,

- The Southbound interface (OpenFlow) to communicate to SDN switches.

PMIPv6: This VNF investigates the concrete evolution of the standardized Proxy Mobile IPv6 (PMIPv6) mobility management protocol for SDN-NFV architectures. Our objective is to design the PMIPv6 evolution that takes full advantage of the SDN-NFV concept. This use case requires an SDN southbound protocol able to provide low-level link technology information such as Received Signal Strength Indicator, wireless channel frequency, information about neighbouring wireless access-points.

Note: Much more details and explanations can be found in the dedicated document (D1.2 from the WP1).

4.3.2 Detailed interface descriptions

The interface (southbound), in addition to the classical informations needed for optimal routing, flows monitoring, resources management and so on, must capture the following specific informations:

- Statistics covering specific information about the interfaces status. For instance, the operation mode (station, access-point, ad-hoc, etc.), the capabilities (mesh mode capable, etc.), the BSSID, the neighboring MAC addresses, Tx/Rx count, if the interface is managed by OpenFlow, etc.
- The capacity to gain multi-hops information about the network performance, e.g., the current round trip time (RTT) with a specific target device.
- The capacity of configuring how devices behave according to their direct neighbors. For instance defining how local network services should forge ICMP and/or ND control packets.
- The capacity to control the configuration of other local applications and libraries such as Open vSwitch (OpenFlow application).
- Some specific events occurring at the device level may require immediate attention of the controller and services, e.g., a wired Ethernet cable unplugged or a new WiFi USB dongle plugged. Devices send “event” packets to the controller detailing the reason of the event.

4.4 SDN core for mobility management and heterogeniety

4.4.1 Updates on architecture mapping

Figure 35 presents the architecture proposed in the SIGMONA reference architecture integrating different architectural components. Enabling cognitive networking from radio access network up to the SDN application layer, cognitive networking needs a cognitive engine, a cognitive agent in the data path and cognitive clients. The cognitive engine is the main decision making control element that can be implement as an SDN application or attached to the SDN orchestrator. In the architecture in Figure 35, the cognitive engine is placed in the orchestrator, the cognitive agent is placed in the data plane/virtual infrastructure. The cognitive agent works as the network status sensor and provides the network information to the cognitive engine in real time. The UE having the capabilities of cognitive clients to switch in frequency bands can use the variety of available access networks. In theory, cognitive clients scan spectra and can use unused spectrum whenever available. However, in our case the decision comes from the cognitive engine based on information not only from the cognitive clients but also from the cognitive agent in the data plane. Once an unused spectra is detected by the cognitive clients, the information is sent to the cognitive engine. The cognitive engine acquires network status information from the cognitive agent (e.g. available bandwidth). Henceforth, the cognitive engine allows the cognitive client to use the unused spectrum and at the same time allocates more bandwidth to that particular client in the backhaul network using the information from the cognitive agent in infrastructure layer.

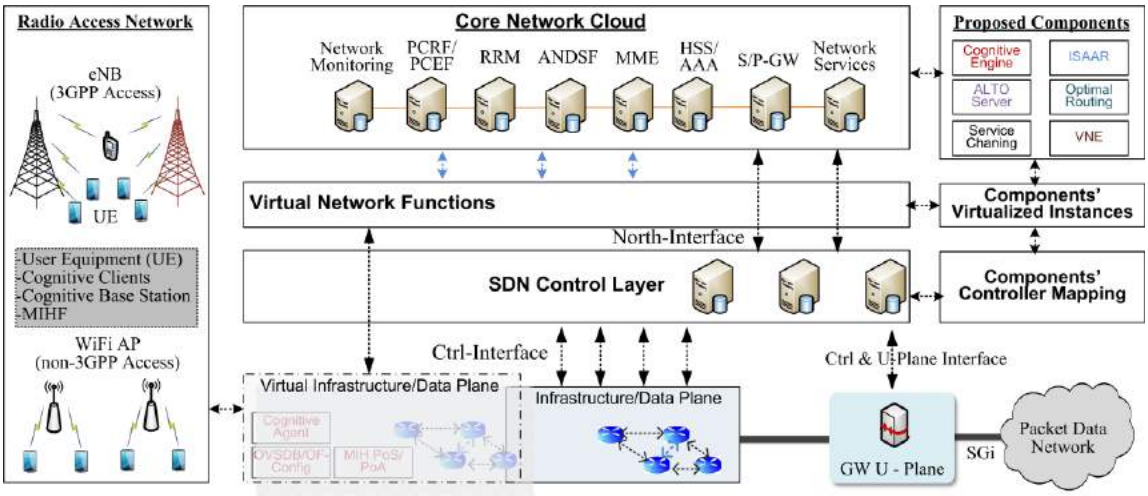


Figure 35: Architectural Mapping

4.4.2 Detailed interface descriptions

The necessary interfaces can be visualized from Figure 20 and Figure 21. In Figure 20 it is clear that the cognitive process if implemented in the SDN controller need to get information from the data plane and the EPC elements. The cognitive process needs to get the network status information from the data plane cognitive agent through the OpenFlow protocol. It can get the information about user profile from the EPC elements (such as HSS) through a north-bound API. Besides this generic SDN scenario, there are two other possible implementation architectures and might need different interfaces and methodologies. These are:

a. In a scenario shown in Figure 21, there is a hierarchy of controllers and the cognitive engine is implemented in the core network. In this case the SDN controller gets the network status information from the RAN controllers and provides that to the cognitive engine. This needs an API between the RAN and SDN controllers, where as a northbound API can be used between the cognitive engine and the SDN controller. Similarly, the cognitive engine needs its integration with the core network elements to get and act according to the user profiles for QoS settings, bandwidth allocation and user mobility.

b. For implementation of cognitive engine in the orchestrator as shown in Figure 35, the cognitive engine also needs to manage virtual instances of the cognitive agent in the infrastructure layer. This will require modifications to the OpenFlow protocol to support multiple and instantaneous communication between cognitive agent instances and the cognitive engine.

5. Conclusions

In the field of mobility management the following research questions are under investigation in the actual phase of the project. Current 3GPP networks make use of mobility management protocols, which are based on different IP tunnelling options (MIP, PMIP, GTP, HIP IPsec). The main research questions are the integration of centralized and/or distributed anchors with the SDN forwarding functions, host-based mobility management where the mobile node also deals with encapsulation/decapsulation duties, QoS/QoE driven mobility management and support of complex mobility scenarios (e.g., network mobility, flow mobility, macro- and micro-mobility, session mobility) in Software-Defined Mobile Networks.

Therefore we have started to elaborate an SDN-aware UFA HIP mobility framework with the appropriate proactive handover preparation, initiation and execution schemes. We are also concerned with purely SDN-based mobility management solutions. The advantages of such solutions are that mobility transparency can be provided to higher layers even without applying additional tunnelling. However, it is challenging, how complex mobility scenarios and additional features will be realized, such as identity-locator splitting or routing protocol independency of different network domains, which are already supported by current distributed mobility management solutions. We have designed the first versions of our OpenFlow based SDN mobility management framework for virtualized mobile networks. According to our preliminary evaluation results we can state the the direction is promising, the work must be continued. NFV/SDN technologies introduce enormous flexibility to support and deploy novel mobility management schemes. Only a few existing articles have analyzed the potential of such a dynamically manageable architecture in addressing the problems of future's heterogeneous wireless setups. Our concept maps IEEE 802.21 MIH functions into the proposed NFV architecture and integrates an OF-based handover execution scheme, in order to provide efficient, proactive, fine grained, QoS/QoE-aware and seamless mobility management. MIH is used to optimize handovers among different access networks, while OF configures network resources to proactively establish and manage communication paths for user level data flows. Our proposed SDN controller extensions gather mobility related cross-layer information by relying on dynamic information exchange using MIH event/command services, static data of MIIS and the policy control of PCRF (all implemented as VNFs in the telco cloud). Based on MIH protocol signaling, and also using data from other SDN controller modules, the architecture manages resources during handover events, and configures flow paths in a proactive and highly dynamic manner according to the actual connectivity options. Using this concept, the impact of inter-technology handovers on the user flows can be minimized, the user and data planes of mobility management can be splitted, optimal transmission routes can be continuously maintained, and flow-level decisions can be made and executed to support efficient offloading situations.

A challenge still to be addressed here is the orchestration of multiple runs of optimization, i.e., how to harmonize decisions of HO events triggered by link going down indications due to user mobility with HO events triggered by other control mech-anisms such as network initiated smart offloading algorithms.

Cognitive network aims at automating the network to react to changes in the network environment without sacrificing the QoS/QoE and minimal human intervention. The currently used networking technologies, however, are delimiting the autonomic notion of cognitive networking due to tight coupling of the control and data planes. Therefore, we proposed and demonstrated software defined cognitive networking along with its integration with OpenFlow WiFi which is presented in this document. The first experiments of inter-working of such heterogenous networks were carried out and the testbed will be further extended to implement the cognitive engine in the SDN control plane.

References

[1] Z. Faigl, M. Telek, „Modeling the signaling overhead in Host-Identity Protocol-based secure mobile architectures," Journal of Industrial Management and Optimization, AIMS, AIMS, vol. 11, no. 3, pp. 887-920. (2015) DOI: 10.3934/jimo.2015.11.887.

[2] Z. Faigl, Performance Analysis of Signalling Overhead in Host Identity Protocol-based Secure Mobile Networks: Ultra Flat Architecture or End-to-End Signalling?", Wireless Networks, Springer, 0.1007/s11276-014-0797-8, 2014.

[3] ETSI TS 124 312, "Access Network Discovery and Selection Function (ANDSF) Management Object (MO), 0.4, 2013.

[4] GPP TS 23.261. (2010). IP flow mobility and seamless wireless local area network (WLAN) offload, Stage 2, Release 10. 3GPP Technical Specification.

[5] IEEE, "IEEE Standard for Local and metropolitan area networks- Part 21: Media Independent Handover." IEEE Std 802.21-2008, 2009.

[6] B. Olteán, Scalable mobile internet architectures, MSc thesis, Research supervisor: L. Bokor, Budapest 2014.

[7] D. Paróczai, L. Bokor, "Centralizált, részben és teljesen elosztott, valamint programozható hálózatokhoz tervezett mobilitás menedzsment sémák analitikus elemzése", MIK research summary,https://mik.bme.hu/wp-content/uploads/2013/11/MIK_tanulmany_VirtMobOpt_mobilitas-kezeles_v2.pdf

[8] D. Klein and M. Jarschel. An OpenFlow extension for the OMNeT++ INET framework. In Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (SimuTools '13). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 322-329., 2013.

[9] K. Bakshi, Considerations for Software Defined Networking (SDN): Approaches and use cases, in: 2013 IEEE Aerospace Conference, 2013, pp. 1–9.

[10] IETF, Distributed Mobility Management (dmm) Working Group (http://datatracker.ietf.org/wg/dmm/).

[11] H. Ali-Ahmad, et al., An SDN-Based Network Architecture for Extremely Dense Wireless Networks, in: 2013 IEEE SDN for Future Networks and Services (SDN4FNS), 2013, pp. 1–7.

[12] S. Yeganeh, A. Tootoonchian, Y. Ganjali, On scalability of software-defined networking, IEEE Communications Magazine 51 (2) (2013) 136–141.

[13] A. Doria, et al., RFC5810 - Forwarding and Control Element Separation (ForCES) Protocol Specification, IETF.

[14] R. E. Enns, et al., RFC6241 - Network Configuration Protocol (NETCONF), IETF.

[15] S. Sezer, et al., Are we ready for SDN? Implementation challenges for software-defined networks, IEEE Communications Magazine 51 (7) (2013) 36–43.

[16] Open vSwitch, an Open Virtual Switch, Monitoring VM traffic using sFlow (http://openvswitch.org/support/config-cookbooks/sflow/).

[17] S. Gundavelli, et al., RFC5213 - Proxy Mobile IPv6, IETF.

[18] R. Thomas, L. DaSilva, and A. MacKenzie, "Cognitive networks," in New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. First IEEE International Symposium on, Nov 2005, pp. 352–360.

[19] C.-I. Badoi, N. Prasad, V. Croitoru, and R. Prasad, "5G based on cognitive radio," Wireless Personal Communications , vol. 57, no. 3, pp. 441–464, 2011.

[20] F. K. Jondral, "Software-defined radio: basics and evolution to cognitive radio," EURASIP Journal on Wireless Communications and Networking, vol. 2005, no. 3, pp. 275–283, 2005.

[21] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The tcp/udp bandwidth measurement tool," http://dast.nlanr.net/Projects, 2005.

[22] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler et al., "Sip: session initiation protocol," 2002.

[23] R. Gayraud and O. Jacques, "Sipp," SIPp Tool (http://sipp. sourceforge. net), 2004.

[24] Guimaraes, C.; Corujo, D.; Aguiar, R.L.; Silva, F.; Frosi, P., "Empowering software defined wireless Networks through Media Independent Handover management," In Proc. of IEEE GLOBECOM, 2013, pp. 2204-2209.

[25] A. De La Oliva, A. Banchs, I. Soto, T. Melia and A. Vidal, "An overview of IEEE 802.21: media-independent handover services," in IEEE Wireless Communications, vol. 15, no. 4, pp. 96-103, Aug. 2008. Doi: 10.1109/MWC.2008.4599227

[26] F. J. Knaesel, P. Neves, S. Sargento, "IEEE 802.21 MIH-enabled Evolved Packet System Architecture", LNICST-97, 2012, pp 61-75.

[27] D. Klein and M. Jarschel. 2013. An OpenFlow extension for the OMNeT++ INET framework. In Proceedings of SimuTools '13. ICST Brussels, Belgium, 322-329, 2013.

[28] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment, " in Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, ICST, Brussels, Belgium, pp. 60:1–60:10., 2008.

[29] L. Bokor, Sz. Nováczki, L. T. Zeke, G. Jeney: "Design and Evaluation of Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++", in the proceedings of the 12-th MSWIM Conference, pp. 124-133, Tenerife, Canary Islands, Spain, 26-30 October 2009.

[30] Van-Giang Nguyen and Younghan Kim: "Proposal and evaluation of SDN-based mobile packet core networks"EURASIP Journal on Wireless Communications and Networking (2015) 2015:172, DOI 10.1186/s13638-015-0395-1

[31] Pill-Won Park, Seong-Mun Kim, and Sung-Gi Min: "OpenFlow-Based Mobility Management Scheme and Data Structure f or the Mobility Service at Software Defined Networking", International Journal of Distributed Sensor Networks, Volume 2016, Article ID 3674192, 9 pages, http://dx.doi.org/10.1155/2016/3674192

[32] You Wang, Jun Bi, and Keyao Zhang. 2015. Design and Implementation of a Software-Defined Mobility Architecture for IP Networks. Mob. Netw. Appl. 20, 1 (February 2015), 40-52. DOI=http://dx.doi.org/10.1007/s11036-015-0579-2

[33] Z. Faigl, L. Bokor, P. M. Neves, K. Daoud, P. Herbelin: “Evaluation of two integrated signaling schemes for the Ultra Flat Architecture using SIP, IEEE 802.21, and HIP/PMIP protocols,” Computer Network – The International Journal of Computer and Telecommunications Networking, vol. 55, pp. 1560–1575, 2011.