Network Protocols and Algorithms
ISSN 1943-3581
2018, Vol. 10, No. 2

Experimental Evaluation of a SDN-DMM Architecture

Rodrygo T. Córdova, Paulo R. L. Gondim

Electrical Engineering Department – University of Brasilia

70297-400 – Brasilia, Distrito Federal (Brasil)

E-mail: rodrygo.cordova@rnp.br; pgondim@unb.br

Jaime Lloret, Jose Miguel Jimenez

Integrated Management Coastal Research Institute, Polytechnic University of Valencia.

C/ Paranimf nº 1, Grao de Gandía – Gandía, Valencia (Spain)

E-mail: jlloret@dcom.upv.es; jojiher@dcom.upv.es

Received: March 24, 2018 Accepted: June 4, 2018 Published: June 29, 2018

DOI: 10.5296/npa.v10i2.13193 URL: https://doi.org/10.5296/npa.v10i2.13193

Abstract

Mobility management has become a great challenge due to the exponential growth in the number of devices that can connect to home or visited networks, and the need for providing seamless mobility in future generation networks. SDN-DMM (Software Defined Network Architecture for Distributed Mobility Management) architecture has been proposed [11], allowing to separate control and data planes, for the distributed mobility management through bidirectional IP flows. This article reports on aspects related to the implementation of SDN-DMM, conducted with metrics as packet loss, throughput and handover latency, considered in a comparison involving traditional routing and SDN-DMM. The results show the SDN approach not only provides the intrinsic benefits of SDN in comparison with traditional architectures, but also deals with the distributed mode of mobility management in heterogeneous access networks in a simplified and efficient way.

Keywords: Software-Defined Networking (SDN); OpenFlow; Distributed Mobility Management (DMM); IP Mobility Management; Mobile IP networks**.**

1. Introduction

Mobile communication networks have become the main method of access to the Internet, which has significantly increased the number of mobile devices connected to the global network [1]

Once the services offered by network operators tend to involve solutions completely based on IP protocol [2] and the communication sessions must be maintained along the user´s movement through networks, the IP mobility management is a factor of extreme impact for communication networks [3]. This aspect is essential for a great number of applications, including video streaming, and the evaluation of network parameters such as bandwidth, delay and packet loss [4] is of great importance for assessing the QoE of the end-users.

The IETF standards of IP mobility management, as MIPv6 [5] and PMIPv6 [6], depend on the central units that manage and act on both control plane and data plane of the network. Once they are based on the traditional routing of IP packets, they involve some challenges, such as sub-optimal routing, low scalability, processing overload in the central units and low granularity of the mobility management service.

An alternative for dealing with the intrinsic problems of centralization and the associated costs involves a concept, called distributed mobility management [7], whose main characteristic is the clear separation among the actions performed in both control plane and data plane. The data plane functions are distributed along the equipment on the network edge, towards approximating the user´s mobility agent through the implementation of a flatter network approach [8]. The traffic forwarded to a mobile node does not need to cross a central point in the network, where it is treated by the agent nearest the user.

DMM approach allows the optimal use of network resources primarily due to the approximation between data traffic delivery and the point of attachment of the user, this way prevent exceeding the available core network capacity. A mobile architecture with fewer hierarchical levels, a flatter network, could benefit the quality of experience, traffic offloading and CDN (Content Distribution Network) mechanisms, once that the network works best for direct communications among peers in the same geographical area [9].

Depending on the distribution level of the control plane [9, 10], the DMM solutions can be categorized into two types, namely partially distributed – the data plane is completely distributed in the network, however, the control plane is centralized in the control points in the network - and completely distributed – both data plane and control plane are distributed along the network (there are no central control points). Figure 1 shows the DMM solutions for both, partially and fully distributed DMM solutions.

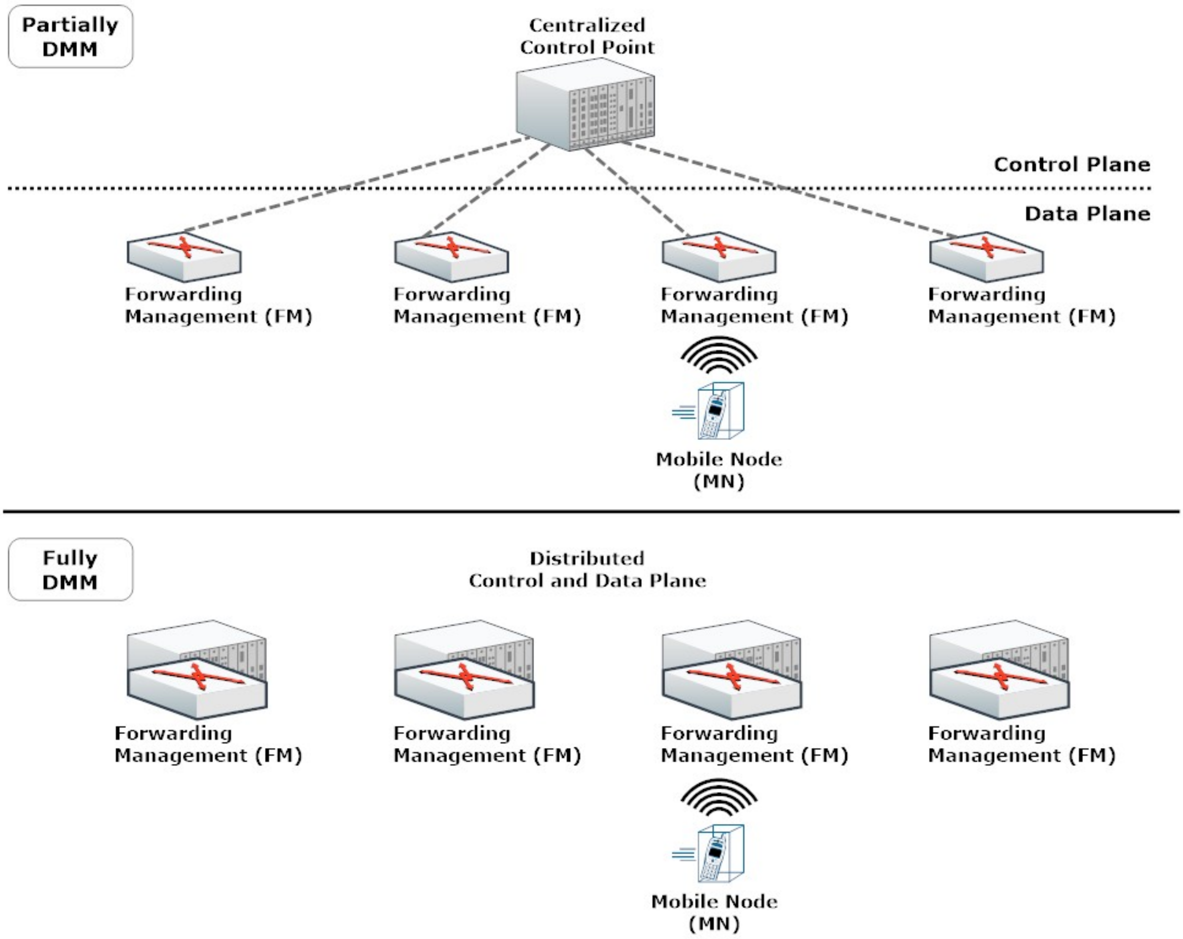

Figure 1 – DMM solutions: partially and fully distributed

With the forwarding management distributed in the edge of the network, closer to the mobile node, the DMM can address problems as non-optimal routing and lack of scalability. The non-optimal routing is resolved because the traffic do not need to be routed via centralized anchor, that increasing the end-to-end delay, the traffic is forwarding by the nearest FM agent closer to the user. The scalability issue, where setting up tunnels through a central anchor and maintaining mobility context for each MN usually requires more concentrated resources in a centralized design [9], is handled by each FM serving the MN that are connect to them, therefore, do not exist a single device that have to process all tunnels within the network, this function is distributed along the FMs.

On the other hand, software defined networking (SDN) represents an emerging trend, redefining the landscape for telecommunication operators and Internet service providers (ISP). This trend has commonly been implemented by the use of the OpenFlow protocol and specific rules, that are defined in a controller, with northbound and southbound layers.

In the traditional routing architecture, the control and data plane residing in each internetwork device in the network, as routers and switches. Even with the separation of the control and data planes inside the device, where the operation between the planes is segmented, there is no concept of separation between the planes for the network as a whole.

New services, deployments, control plane operations, need to be run or set up in each control plane at each network device. For example, to create a specific route between two points of the network, it is necessary to insert this information in each control plane of the devices that are involved in this route.

The SDN paradigm promotes the clear separation of the control plane and data plane for the entire network infrastructure, where the control plane is decoupled of each internetwork device to a central point in the network. The networks devices are responsible only by the data plane, to forwarding traffic, just providing hardware capability, while all the intelligence and control of the network are centralized in one point named controller. This central point of control is responsible for the network management and has full knowledge of all network devices, protocols and operations.

In the example mentioned before, for implementing a specific route between two points of the network with SDN paradigm, it is necessary only to inform the controller about the route, then automatically occurs the controller communication with each network device involved and adjust of the data plane to implement the route. The figure 2 shows the SDN architecture with the separation between the control and data planes, and the communication between the controller and the internetwork devices using OpenFlow protocol to the management of traffic flows in the networks.

This article reports on the implementation of a proposal, called SDN-DMM (Software Defined Network Architecture for Distributed Mobility Management) [11], in a real experimentation environment. The proposal is based on the SDN network paradigm, for the mobility management distributed through bidirectional IP flows.

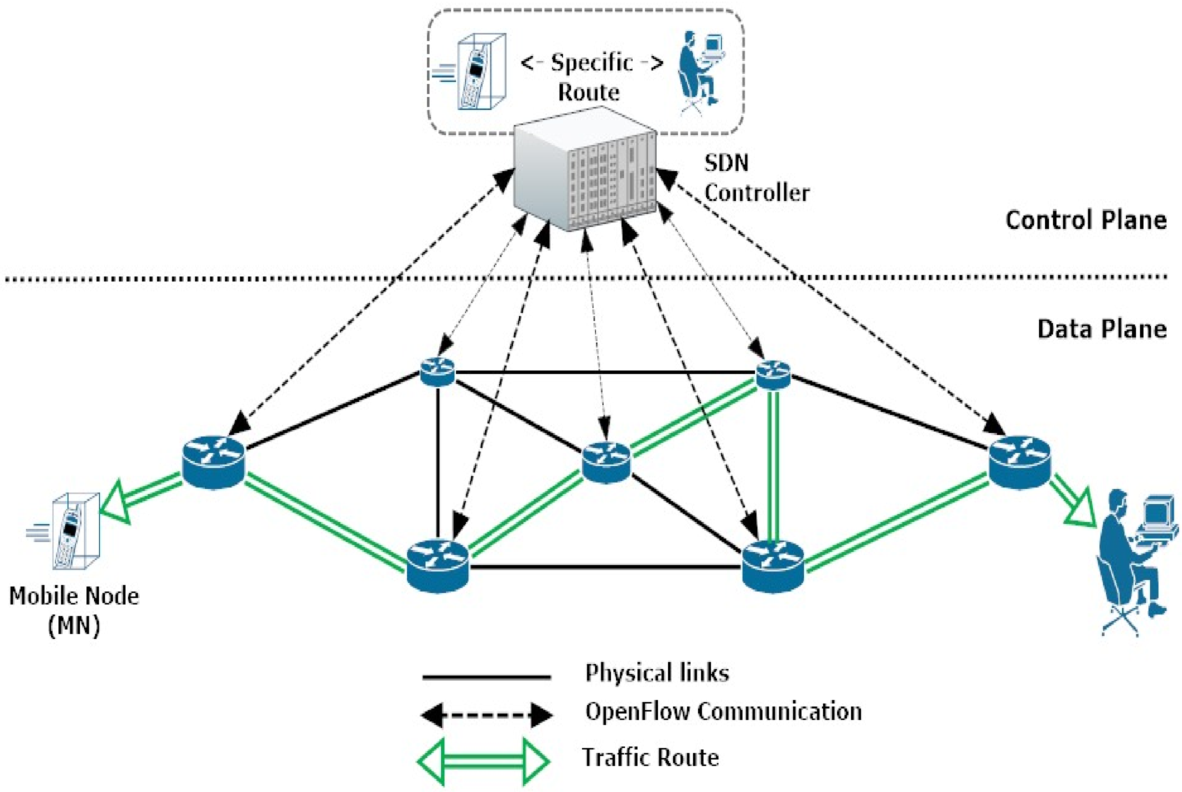

Figure 2 – SDN architecture

The remainder of the paper is organized as follows: Section 2 describes the SDN-DMM proposal and the implementation environment; Section 3 addresses the scenarios implemented and the results; finally, Section 4 provides the conclusions and suggests some future work.

2. Implementation of SDN-DMM

This section introduces the SDN-DMM proposal, which uses the SDN approach for the management of the distributed mobility with IP flows, and its implementation in a real experimentation scenario through equipment and market software. It describes the architecture, topology, scenarios and metrics analyzed.

SDN-DMM is a partially and network-based distributed DMM proposal. It is based on the SDN architecture, where the IP mobility is supported by the network infrastructure through the automatic adjustment of IP flows in the data plane performed by the SDN controller with OpenFlow protocols, according to the handover conducted with the user terminal assistance.

It uses an abstraction of the northbound SDN layer, called *Intent*, which informs the SDN controller on certain communication flows among hosts to be implemented in the network. Through a general view of the network topology, the controller creates and adjusts OpenFlow rules according to the mobility scenario. The topology information is continuously sent to controller, for creating and fixing the data plane of the network with OpenFlow rules to maintain the communication, as indicated by the Intent. This operational cycle is represented in the figure 3.

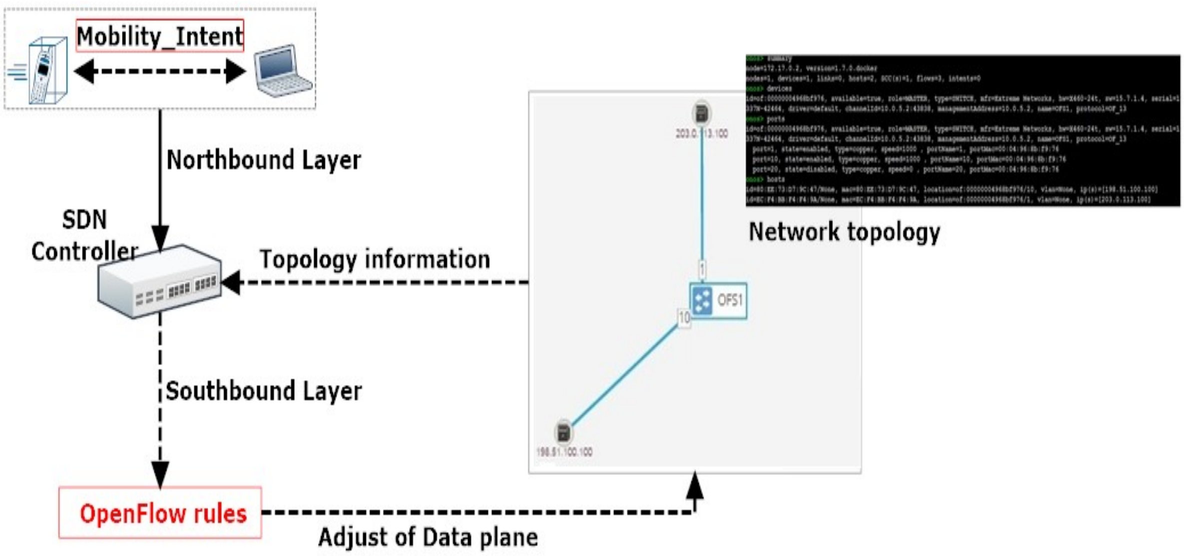

Figure 3 – SDN-DMM operation cycle

The Intent application create OpenFlow rules based on the current topology information to implement specifics IP flows in the data plane for permit the full communication between hosts during their mobility. The main OpenFlow messages related with the mobility process are PacketIn (type 10), FlowRemoved (type 11), PortStatus (type 12) and FlowMod (type 14).

To implement the IP flows in the data plane, the FlowMod message type with the Add command is used. It defines some criteria to realize matching with the traffic and identifies an unidirectional IP flow; for example, a flow can be identified by packets that match a specific source and destination MAC address. This OpenFlow message also defines a specific action to be applied in the identified flow, for example, forwarding the flow to a determined switch port. Therefore, it is necessary at least two FlowMod messages to create a bidirectional IP flow in the data plane for effective communication between the hosts.

When a topology change occurs, depending on how it occurs, a PortStatus or a PacketIn message is sent to the controller to inform about the change. If this change affect the previous communication established in the data plane, the controller sends new FlowMod messages to install or delete flows in the data plane for adjust the paths and keep the communication active. A FlowRemoved message can also be sent to the controller to inform that an old flow that is no more in use, because of topology change, is being removed from the flow table. The figure 4 shows the main message exchange diagram of the process to establish a bidirectional IP flow between two hosts and adjust the flows after a mobility occurs.

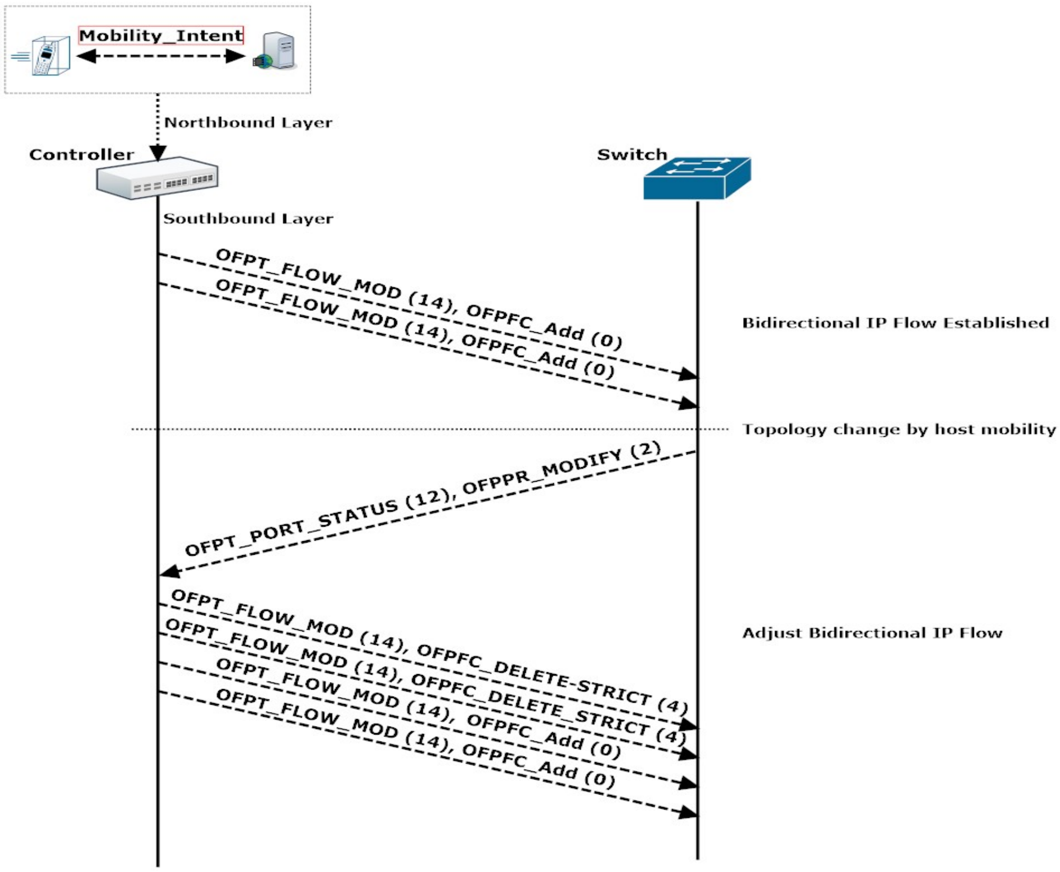

Figure 4 – OpenFlow message exchange diagram for the mobility process

Therefore, the data plane of the network can forward the traffic directly to the mobile node at any network point where it is connected. Figure 5 displays the process, described as follows:

1. Mobile node MN have an active communication with the correspondent node CN when performs a handover from NB in the 3G Network A to eNB in the 4G Network B and maintains its original IP address;

2. Switch SB identifies the presence of the mobile node with an IP different from its address scope and informs the event to controller C1;

3. Controller C1 detects a topology change in relation to mobile node MN, recalculates a new path and sends new OpenFlow rules to adjust the communication links used.

In the sequence, the new bidirectional IP flow is established in the network, which enables the packets addressed to mobile node MN to be correctly forwarded according to the IP flow, rather than the traditional routing. Therefore, the general routing and the other communications in the network are not affected.

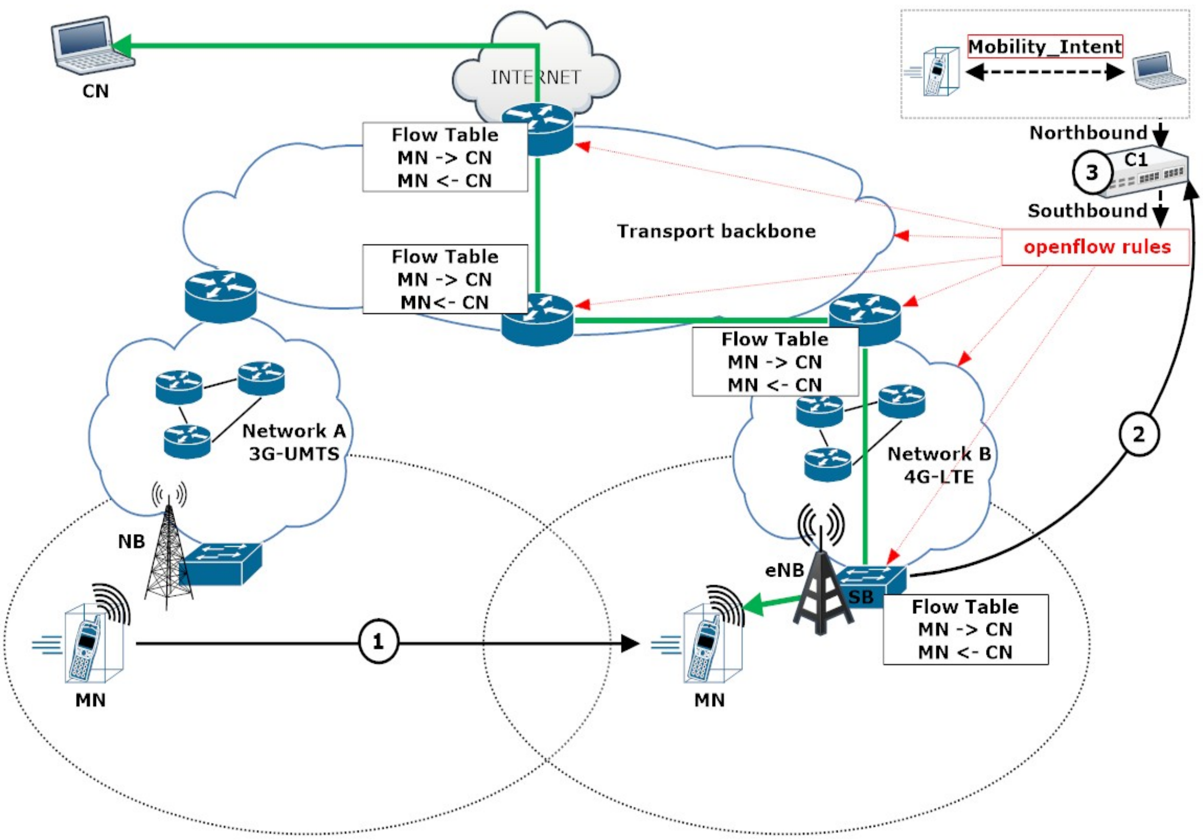

Figure 5 – SDN-DMM proposal for the IP mobility management

The implementation consists in the use of SDN-DMM for enabling the continuity of the IP session in the mobility management with an SDN controller adjusting the data plane of the network through OpenFlow protocol, when a mobile node performs a handover from a network A to a network B, with active connections with a server.

The controller used for the implementation was Open Network Operating System

(ONOS). It was implemented in the container modality through Docker software in Ubuntu Server operational system. The following equipment was used with its respective functions:

- 01 OpenFlow Switch Extreme X460-24t, called OFS1;
- 02 Access Points Cisco WRT160N, called AP-A and AP-B;
- 03 Notebooks VAIO Fit15F, called Mobile Node (MN1), Media Server (MD1) and Controller SDN (C1).

The network metrics selected to evaluate the performance of the proposed architecture were:

1. Throughput of UDP;
2. Throughput of TCP;
3. Loss of UDP packets;
4. Loss of ICMP packets;
5. RTT latency with ICMP;
6. Handover latency.

Metrics 1 through 5 were analyzed for a performance comparison between the forwarding of packets with traditional routing and bidirectional IP flow with OpenFlow.

All metrics were analyzed for two handover scenarios:

- SDN-DMM WITH SWITCHING BY MANUAL HANDOVER

For a direct verification of the impact of SDN-DMM on the network metrics analyzed, hosts MN1 and MD1 are directly connected to OpenFlow switch through UTP cables, where the handover of host MN1 from network A to network B is manually performed through its disconnection from the port associated with network A and connection with the port associated with network B in the OpenFlow switch, while maintaining active sessions with MD1. This scenario aims at the evaluation of the impact of SDN-DMM without the handover performed in the radio part of the network.

- SDN-DMM WITH AUTOMATIC HANDOVER THROUGH IAPP

For the verification of the proposal behavior in a real handover scenario, i.e., implemented with the handover performed in the radio part of the network, MN1 connects to Access Point AP-A and performs handover for Access Point AP-B, while maintaining an active session with host MD1. The results reflect a better approximation of the complete implementation of the proposal for the mobility management, and the Inter-Access Point Protocol (IAPP) is used.

Regarding software, operational systems and main protocols used for the evaluation of the network metrics selected, a free software (iPerf) was used for the generation of TCP/UDP traffic for measurements of the throughput and loss of packets parameters. The latter was also analyzed through the generation of messages with ICMP protocol, also utilized for the evaluation of RTT latency.

Another free software (Wireshark) was used for the capture of the traffic generated for measurements of the handover latency through time markings of packets received by the MD server, where ICMP packets were also used for measurements of the metrics.

The general architecture, used as a basis for the implementation of the environments and scenarios described, is composed of five main networks and their elements, as described below and shown in Figure 6:

- Internet network: MD1 server. Traffic destined to external networks;
- Client A network: host MN1 and access point AP-A. Traffic destined to clients of network A;
- Client B network: access point AP-B. Traffic destined to clients of network B;
- Management network: controller C1 and switch OFS1. Traffic of SDN control and for the management of switch;
- Out-of-band access network (OOB): remote terminal and controller C1. Management access to controller C1.

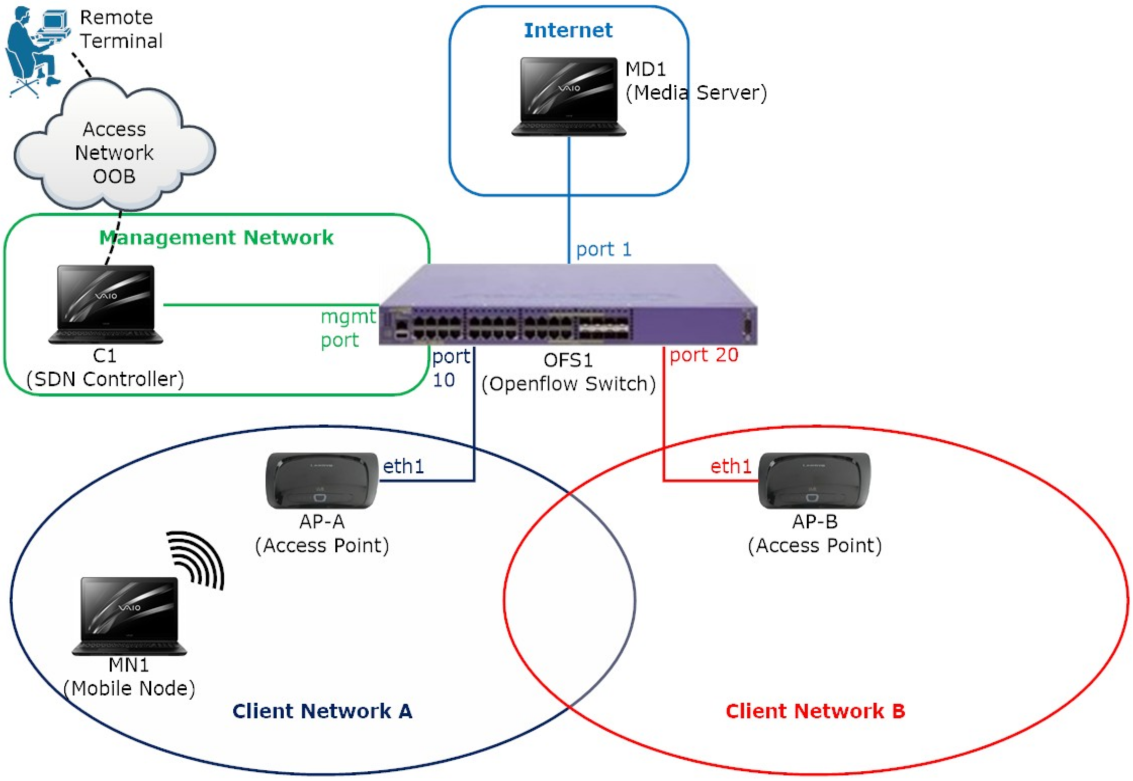

Figure 6 – Testbed for SDN-DMM Implementation

3. Tests and Results

This section reports the tests conducted and the works regarding the real implementation of SDN-DMM and the analysis of the network metrics of throughput, loss of packets, RTT latency and handover latency, according to the descriptions in Section 2. Below are the environments and scenarios implemented:

- Traditional routing without handover;
- SDN-DMM without handover;
- SDN-DMM with handover by manual switching;
- SDN-DMM with automatic handover by IAPP protocol [12].

3.1 Traditional routing without handover

In this scenario, communication between hosts MN1 and MD1 was implemented by IP traditional routing and their direct connection through UTP cables to switch OFS1 in their original networks, according to the network topology shown in Figure 2.

No SDN mechanism or mobility scenario was used. The communication was performed by the routing of packets among the networks based on the switch OFS1 routing table.

Such a scenario aims at evaluating the performance of UDP throughput, TCP throughput, loss of UDP packets, loss of ICMP packets, and RTT latency for the obtaining of data for a comparative analysis of the metrics with SDN environment.

After communication had been established, iPerf software generated UDP traffic at a 100 Mbps rate, for the evaluation of the throughput and loss of packets, and TCP traffic for evaluating the maximum throughput in a 5-minute interval. It run in the client mode, in host MN1, and in the server mode, in host MD1. The results are shown in Table 1.

Table 1 – Results for the traditional routing scenario.

Test	Tool/Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	100 Mbps
TCP throughput	Iperf/TCP	300 seconds	523 Mbps
Loss of UDP packets	Iperf/UDP	300 seconds	0,052%
Loss of ICMP packets	Ping/ICMP	300 seconds	0%
RTT latency	Ping/ICMP	300 seconds	<1 ms

3.2 SDN-DMM without handover

In an SDN-DMM approach, the communication between MN1 and MD1 hosts was implemented through bidirectional IP flows defined by controller C1 through the use of Intents.

The hosts were directly connected to switch OFS1 by UTP cables in their original networks, similarly to the traditional routing scenario, and conducted the same tests to obtain data on the UDP throughput, TCP throughput, loss of UDP packets, loss of ICMP packets and RTT latency and compare the performance with the traditional routing environment. The results are shown in Table 2.

Table 2 – Results for SDN-DMM scenario.

Test	Tool/ Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	100 Mbps
TCP throughput	Iperf/TCP	300 seconds	522 Mbps
Loss of UDP packets	Iperf/UDP	300 seconds	0,11%
Loss of ICMP packets	Ping/ICMP	300 seconds	0%
RTT latency	Ping/ICMP	300 seconds	<1 ms

3.3 SDN-DMM with handover by manual switching

From the implementation of scenario SDN-DMM with no previous handover and with the hosts directly connected to switch OFS1, the handover of host MN1 was manually performed through the switching between the port of the switch associated with network A and the port of the switch associated with network B during tests for the collection of UDP throughput, TCP throughput, loss of UDP packets, loss of ICMP packets and RTT latency in the host handover.

The handover latency metric was also analyzed through both TCP packets generated by iPerf and ICMP packets. The same parameters used for software iPerf were utilized and Figure 7 shows the throughput and loss of UDP packets. The handover was performed in the 50 to 60-seconds interval in the test, in which packets were lost and the throughput was momentarily reduced. Both packets and rate returned to their state of normality after handover. At the end, a 99 Mbps average UDP throughput and a 0,96% loss of packets were obtained.

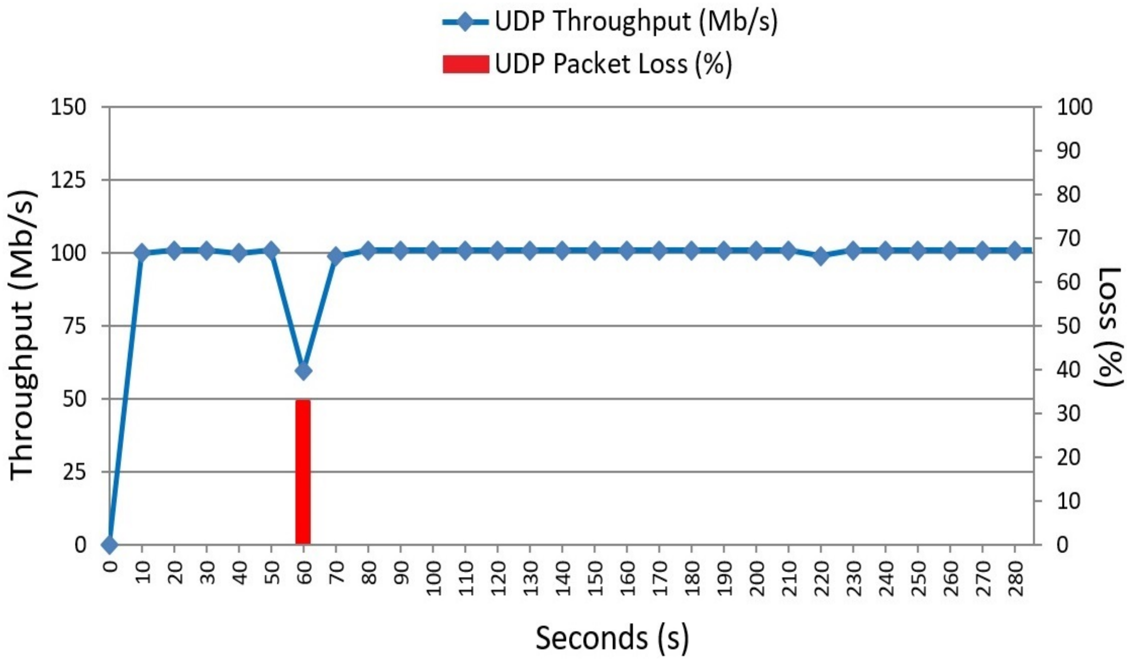

Figure 7 – Test of throughput and loss of UDP packets for SDN-DMM with handover by manual switching.

Figure 8 shows the handover was performed in the 50 to 60-seconds interval, in which the TCP throughput was momentarily reduced; the final average was 418 Mbps. Figure 9 shows a 1% loss of packets and average RTT latency lower than 1ms, with ICMP.

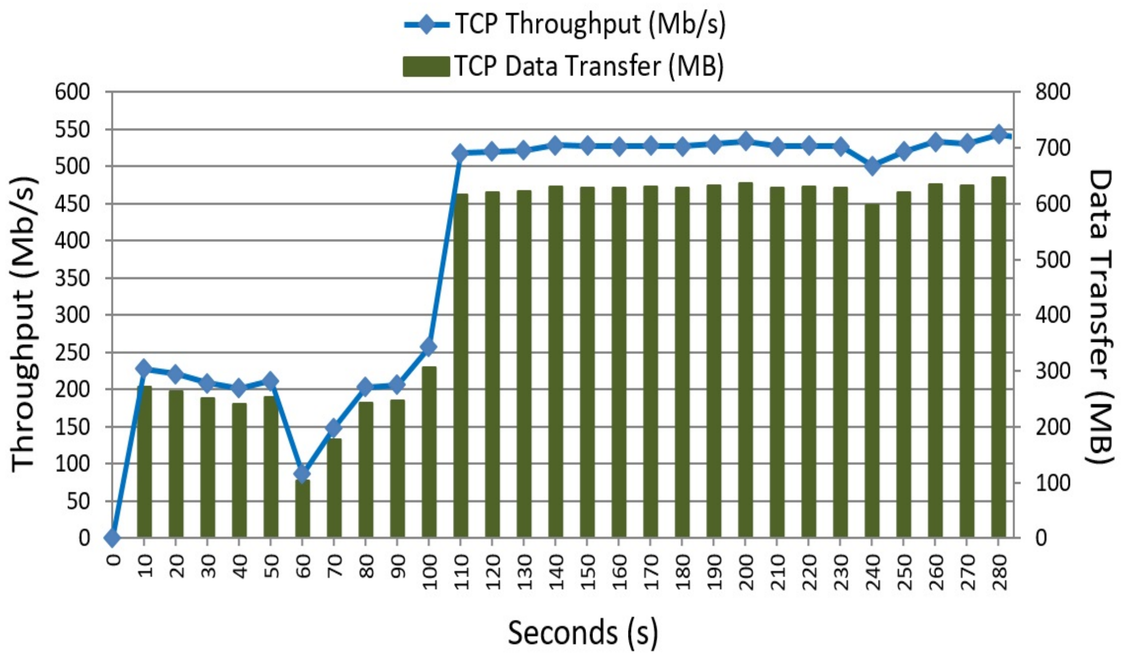

Figure 8 – Test of TCP maximum rate and data transfer for SDN-DMM with manual handover by switching.

```
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
.
<lines omitted>
.
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
General failure.
General failure.
Request timed out.
Request timed out.
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
.
<lines omitted>
.
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
reply from 203.0.113.100: bytes=32 time<1ms TTL=128
Ping statistics for 203.0.113.100:
    Packeys: Sent = 288, Received = 284, Lost = 4 <1% loss>,
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figure 9 – Loss of packets and RTT latency with ICMP for SDN-DMM with manual handover by switching.

Figure 9 shows the handover is performed at the moment the General Failure message occurs, due to the physical disconnection the host from switch port of network A. After reconnect the host to the new switch port of network B, the Request timed out message indicate that the data plane is being adjust for support the host mobility, and finally, the Reply message indicate the end of the mobility process, where the data plane is update with the new IP flows, where the communication between the host is restored.

For the analysis of handover latency, during the test of TCP throughput, the packets received were collected in the server and the intervals between the last pack received when host MN1 had been disconnected and the first after the handover were compared. Figure 10 shows the result for TCP handover latency.

```
        |Time      |Source       |Destination   |Protocol|Length|Info
1014823 54.840643 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014824 54.840657 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014825 54.840666 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014826 54.840676 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014827 54.840685 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014842 59.357945 203.0.113.100 198.51.100.1.. TCP          66 [TCP Dup ACK 1014827#1]
1014846 59.358448 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014849 59.358591 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014852 59.358842 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
1014855 59.359084 203.0.113.100 198.51.100.1.. TCP          54 5001 -> 53178 [ACK]
```

Figure 10 - TCP handover latency for SDN-DMM with manual handover by switching.

Figure 10 shows an approximately 4,517-seconds latency between the TCP packets received by MD1 during handover performed by MN1. An approximately 10,67-seconds ICMP handover latency is displayed in Figure 11.

```
        |Time       |Source          |Destination
143      23.374744   198.51.100.100   203.0.113.100
146      24.390991   198.51.100.100   203.0.113.100
149      25.407102   198.51.100.100   203.0.113.100
154      26.427938   198.51.100.100   203.0.113.100
157      27.445060   198.51.100.100   203.0.113.100
257      38.115483   198.51.100.100   203.0.113.100
292      39.132151   198.51.100.100   203.0.113.100
304      40.148671   198.51.100.100   203.0.113.100
309      41.166448   198.51.100.100   203.0.113.100
321      42.181282   198.51.100.100   203.0.113.100
```

Figure 11 - ICMP handover latency for SDN-DMM with handover by manual switching.

Table 3 shows the results for scenario SDN-DMM with handover by manual switching.

Table 3 – Results for scenario SDN-DMM with handover by manual switching

Metric	Tool/Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	99 Mbps
TCP throughput	Iperf/TCP	300 seconds	418 Mbps
Loss of UDP packets	Iperf/UDP	300 seconds	0,96%
Loss of TCP packets	Ping/ICMP	300 seconds	1%
RTT latency	Ping/ICMP	300 seconds	<1 ms
Handover latency	Iperf/TCP	300 seconds	4,517 seconds
Handover latency	Ping/ICMP	300 seconds	10,67 seconds

3.4 SDN-DMM with automatic handover by IAPP[12]

This scenario involves the implementation of the topology of Figure 2, with two wireless networks in which host MN1 will perform a handover maintaining communication with MD1.

For an automatic handover with no additional protocols, the access points were configured with the same SSID and authentication password; band 2,4 GHz with channel 6 were used for access point AP-A, responsible for connectivity with network A, whereas channel 11 was utilized for access point B, responsible for connectivity with network B. They were arranged according to the topology for a superposition of coverage areas that enables host MN1 to perform handover adequately, selecting the access point with the most intense signal for a certain moment.

Once the environment was wireless and, therefore, subject to several interferences, the tests were conducted in three stages for the obtaining of results that enable a consistent analysis of the impact of SDN-DMM on the network metrics:

1. Stage 1 – MN1 connected to network A;
2. Stage 2 – MN1 connected to network B;
3. Stage 3 – Handover of MN1.

Tables 4 and 5 show the results for stages 1 and 2, respectively.

Table 4 – Results for scenario SDN-DMM with automatic handover – MN1 connected to network A.

Metric	Tool/Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	88,9 Mbps
TCP throughput	Iperf/TCP	300 seconds	84 Mbps
Loss of UDP packets	Iperf/UDP	300 seconds	0%
Loss of ICMP packets	Ping/ICMP	300 seconds	0%
RTT latency	Ping/ICMP	300 seconds	2 ms

Network Protocols and Algorithms
ISSN 1943-3581
2018, Vol. 10, No. 2

Table 5 – Results for scenario SDN-DMM with automatic handover a– MN1 connected to network B.

Metric	Tool/Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	25,4 Mbps
TCP throughput	Iperf/TCP	300 seconds	10,4 Mbps
Loss of UDP packets	Iperf/UDP	300 seconds	0,3%
Loss of ICMP packets	Ping/ICMP	300 seconds	0%
RTT latency	Ping/ICMP	300 seconds	15 ms

Figure 12 displays the Wi-Fi connection state of host MN1 during its handover from network A to network B in the third stage of the handover between the wireless networks, using InSSIDer tool. It also shows an increase in the signal intensity in relation to the time of access point AP-B in comparison to access point AP-A during its movement. Handover was performed with the ongoing tests of UDP thrpughput, TCP throughput, loss of UDP packets, loss of ICMP packets and RTT latency.

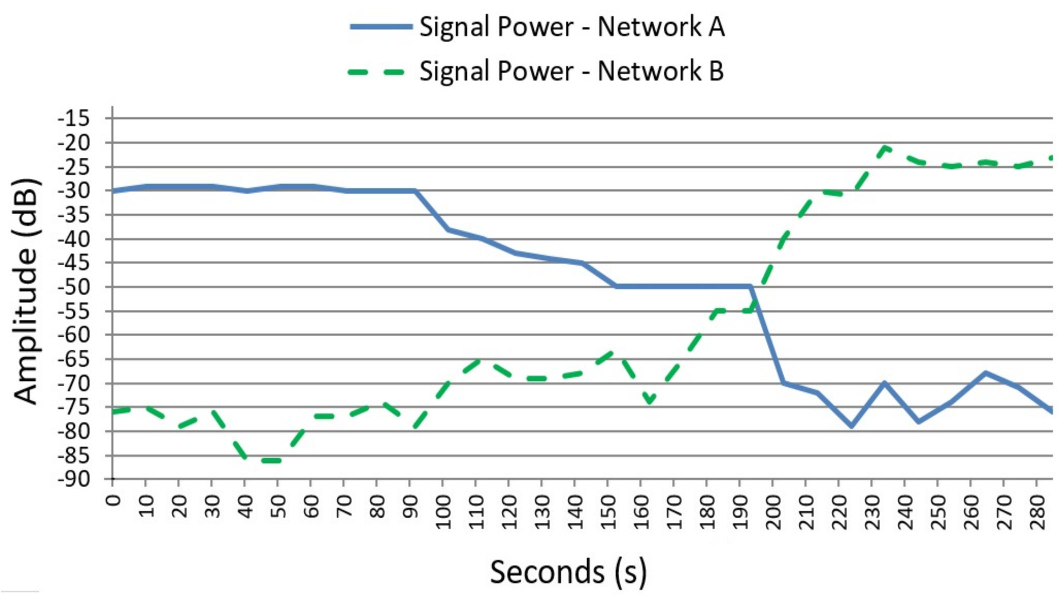

Figure 12 – Variation of RSSI during handover between networks A and B

The abscissa axis in Figure 12 shows an increasing time relation, i.e., during the MN1 movement, the AP-A signal intensity decreases in function of the increase in the AP-B signal. When the AP-B signal is sufficiently higher than that of AP-A, MN1 performs handover for network B.

During the TCP and ICMP throughput, the packets received were collected in the server and the intervals between the last pack received prior to host MN1 performing handover of network A and the first pack received soon after MN1 had connected to network B were compared, for the analysis of handover latency. Figure 13 shows the result for TCP handover latency.

	Time	Source	Destination	Protocol	Length	Info
603072	74.270288	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603073	74.270565	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603074	74.270611	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603075	74.270630	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603076	74.270651	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603088	78.844312	203.0.113.100	198.51.100.100	TCP	1514	[TCP Spurious Rettra
603090	78.962652	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603091	78.968679	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603092	78.968887	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603093	78.968967	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]
603094	78.969061	203.0.113.100	198.51.100.100	TCP	1514	53928 -> 5201 [ACK]

Figure 13 - TCP handover latency for SDN-DMM with automatic handover by IAPP

Figure 13 shows an approximately 4,573-seconds latency between the TCP packets received by MD1 during the handover performed by MN1. The ICMP handover latency shown in Figure 14 was approximately 5,893 seconds.

	Time	Source	Destination
386	159.426374	198.51.100.100	203.0.113.100
388	160.416258	198.51.100.100	203.0.113.100
392	161.416328	198.51.100.100	203.0.113.100
394	162.417606	198.51.100.100	203.0.113.100
396	163.433690	198.51.100.100	203.0.113.100
402	169.427373	198.51.100.100	203.0.113.100
404	170.327373	198.51.100.100	203.0.113.100
408	171.329384	198.51.100.100	203.0.113.100
410	172.328496	198.51.100.100	203.0.113.100
412	173.326193	198.51.100.100	203.0.113.100

Figure 14 - ICMP handover latency for SDN-DMM with automatic handover by IAPP.

Table 6 shows the results of the tests conducted prior to the handover and finished after it with the same software and parameters previously used.

Table 6 – Results for the SDN-DMM scenario with handover by manual switching

Metric	Tool/Protocol	Time of execution	Average result
UDP throughput	Iperf/UDP	300 seconds	63,3 Mbps
TCP throughput	Iperf/TCP	300 seconds	34,8 Mbps
Loss of packets	Iperf/UDP	300 seconds	0,82%
Loss of packets	Ping/ICMP	300 seconds	0%
RTT latency	Ping/ICMP	300 seconds	10 ms
Handover latency	Iperf/TCP	300 seconds	4,573 seconds
Handover latency	Ping/ICMP	300 seconds	5,893 seconds

The SDN environment exerts no negative impacts on the network metrics in comparison with traditional routing. The results showed the same performance for the forwarding of packets based on both IP destiny address and bidirectional IP flow used in the SDN-DMM. Table 7 shows the results of the two scenarios.

Table 7 – Comparison of metrics between Traditional Routing and SDN-DMM.

Metric	Traditional Routing	SDN-DMM
UDP throughput	100 Mbps	100 Mbps
TCP throughput	523 Mbps	522 Mbps
Loss of packets	0,052%	0,11%
Loss of packets	0%	0%
RTT latency	<1 ms	<1 ms

According to Tables 7 and 8 and the results of the experiments, SDN-DMM causes no performance loss when the mobile node is connected to a foreign network. When the mobile node performs a handover for a new network, the values obtained for the metrics analyzed are the same of those for a native host to the network, according to the tests conducted in the handover scenario with SDN-DMM and Table 8.

Table 8 – SDN-DMM metrics with manual and automatic handover

Metric	SDN-DMM with manual handover by switching	SDN-DMM with automatic handover by IAPP
UDP throughput	99 Mbps	63,3 Mbps
TCP throughput	418 Mbps	34,8 Mbps
Loss of packets (UDP)	0,96%	0,82%
Loss of packets (ICMP)	1%	0%
RTT latency	<1 ms	10 ms
Handover latency (TCP)	4,517 seconds	4,573 seconds
Handover latency (ICMP)	10,67 seconds	5,893 seconds

The values obtained for the *handover* latency metrics (TCP) for manual handover by switching and automatic handover by wireless are due to the balancing between the factors that hamper their measurements in each scenario.

In the manual handover, the host disconnection from the switch port cause the down state of the NIC`s host, where the host reconnection to the new switch port start a process where is need to wait a time to the NIC and switch port go to active state again for forwarding traffic. So this affects the measurement of the metric of the handover latency in the SDN-DMM approach.

On the other hand, for the automatic handover, the interference of the wireless environment with no specific protocol for handover between access points hampers the measurements of the handover latency of the proposal, although the transition between the networks is smoother.

Another important factor to be considered in the analysis of the values obtained for the handover latency is its measurement through the tools used, which hampers the results. When the capture of the packets in the host is activated, part of the capacity of both processing and network is directed to the capture action, which decreases the performance of the metrics being measured. Therefore, better values than those observed are expected for a production scenario.

4. Conclusions and Future Work

SDN represents a big trend in ICT, based on the decoupling of the network control and forwarding functions, allowing the programmability of network control, the abstraction of infrastructure for applications and services, as well as the ability to scale network resources.

The implementation of SDN-DMM in a real experimentation environment and the tests conducted show the approach used for the distributed mobility management based on an SDN architecture meets the communication requirements and causes no harm to the network metrics in comparison to the traditional routing scenario.

The performance for the throughput when the *mobile node* is in mobility matches the cost of packets delivery analyzed in the analytical modelling [11], which shows the proposal always performs delivery in an optimized way and reduces costs in relation to the other approaches analyzed. The loss of packets and handover latency obtained in the tests can be considered coherent with the results of the analytical modeling, according to the low number of messages necessary in the control plane for mobility through the readjustment of the data plane.

Future works involve strategies for applying SDN-DMM for traffic offloading, aiming to produce a better Quality of Experience (QoE) for mobile terminal users in a scenario of heterogeneous radio access technologies, taking into account algorithms for adequate resource allocation. Moreover, the integration with Media Independent Handover (IEEE 802.21 standard) will be discussed, aiming to ensure seamless service continuation during transition between different wireless networks.

Another focus of research involves the evaluation of SDN-DMM for multimedia traffic, in special aspects related to QoE related to video streaming applications ([13], [14]).

Future work also involve aspects related to possible threats and vulnerabilities of SDN-DMM architecture, as well as measures to protect against them, aiming to assure confidentiality, integrity and availability of data. Moreover, computational trust models involving the different entities of the architecture should be discussed, considering context-awareness and nodes with different resources capabilities.

Acknowledgements

This work has been supported by:

- FAP-DF (“Fundação de Apoio `a Pesquisa do Distrito Federal ”) – Brazil;

- “Ministerio de Economía y Competitividad ”/Spain , through the “Convocatoria 2017: Proyectos I+D+I - Programa Estatal de Investigación, Desarrollo e Innovación, convocatoria excelencia” (Project TIN2017 -84802 -C2 - 1 -P).

References

[1] Ericsson, “Ericsson Mobility Report, 2016”. [Online]. Available: https: //www.ericsson.com/res/docs/2016/ericsson - mobility - report - 2016.pdf. Last access: february 18, 2018.

[2] Karimzadeh, M; Valtulina, L; Karagiannis, G. “Applying SDN/OpenFlow in virtualized LTE to support Distributed Mobility Management (DMM)”. In: Proceedings of the 4th International Conference on Cloud Computing and Services Science, CLOSER 2014, 3-5 Barcelona, Spain. 86. SciTePress, Abril 2014. https:// doi.org/10.5220/0004946106390644

[3] Bernardos C., Oliva A., Serrano P., et al. (2014). “An Architecture for Software Defined Wireless Networking”. In IEEE Wireless Communications, vol. 21, no.3, pag 52-61. https://doi.org/10.1109/MWC.2014.6845049

[4] Cánovas, A., Taha, M., Lloret, J., Tomás, J. “Smart resource allocation for improving QoE in IP Multimedia Subsystems”, Journal of Network and Computer Applications, Volume 104, 15 February 2018, Pages 107-116. https://doi.org/10.1016/j.jnca.2017.12.020

[5] Perkins, C., Johnson, D., & Arkko, J. (2011). “Mobility Support in IPv6”. RFC 6275.

[6] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K.,&Patil, B. (2008). “Proxy Mobile IPv6”. RFC 5213.

[7] Chan,H.A.,Yokota,H., Xie, J., Seite, P., & Liu,D. (2011). “Distributed and dynamic mobility management in mobile internet: Current approaches and issues”. Journal of Communications, 6(1), 4–15. https:// doi.org/10.4304/jcm.6.1.4-15

[8] Giust, F; Bernardos, C; Olivia, A. “HDMM: deploying client and network-based distributed mobility management”. Telecommunications Systems, Volume 59 Issue 2, Pages 247-270, Juny 2015. https://doi.org/10.1007/s11235-014-9959-3

[9] Chan, H. (2014). “Requirements for Distributed Mobility Management”. RFC 7333.

[10] Yokota, H., Seite, P., Demaria, E., & Cao, Z. (2010). “Use case scenarios for Distributed Mobility Management”. Internet-Draft (work in progress), draft-yokota-dmm-scenario-00.txt.

[11] Cordova, R. “SDN – DMM: software-defined networking for distributed mobility management on heterogeneous mobile IP networks”. Master’s Degree Dissertation. University of Brasilia, 2016.

[12] IETF. "IEEE trial-use recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distribution systems supporting ieee 802.11 operation," IEEE Std 802.11F-2003, 2003.

[13] Jimenez, J.M., Martínez, J.O.R., Rego, A., Dilendra, A., Lloret,J. “Study of multimedia delivery over software defined networks”, Network Protocols and Algorithms 7 (4), 37-62. 2015. https:// doi.org/10.5296/npa.v7i4.8794

[14] Taha, M., Garcia, L. Jimenez, J.M., Lloret, J. “SDN-based throughput allocation in wireless networks for heterogeneous adaptive video streaming applications”, The 13th International Wireless Communications and Mobile Computing Conference (IWCMC 2017), June 26-30, 2017, Valencia, Spain. https://doi.org/10.1109/IWCMC.2017.7986416.

Copyright Disclaimer