Software Defined Networking to Support IP Address Mobility in Future LTE Network

Morteza Karimzadeh[1], Luca Valtulina[1], Hans van den Berg[1,2] and Aiko Pras[1]
[1]University of Twente, [2]TNO, The Netherlands

Marco Liebsch
NEC Laboratories Europe, Germany

Tarik Taleb
Aalto University, Finland

***Abstract*—The existing LTE network architecture dose not scale well to increasing demands due to its highly centralized and hierarchical composition. In this paper we discuss the major modifications required in the current LTE network to realize a decentralized LTE architecture. Next, we develop two IP address mobility support schemes for this architecture. The proposed solutions can handle traffic redirecting and seamless IP address continuity for the nodes moving among the distributed anchor points in a resource efficient manner. Our approaches are based on the SDN (*Software Defined Networking*) paradigm which is also one of the most important candidate technologies to realize 5G mobile networks. We extend the NS3-LENA simulation software to implement a decentralized LTE network as well as the proposed IP mobility support schemes. The evaluation results show that the proposed solutions efficiently fulfill the functionality and performance requirements (*e.g.,* latency and packet loss) related to mobility management.**

I. Introduction

Cisco forecasted that the worldwide mobile data traffic will be increased more than 8-fold between 2015 and 2020 [1]. Coping with such a demand in the current mobile networks is neither economically nor technically viable. The RAN (*Radio Access Network*) cannot be easily extended due to spectrum limitations. Furthermore, the core of mobile networks is highly centralized, introducing scalability and reliability issues.

Mobile network operators augment RAN capacity by improving spectrum utilization in several ways, *e.g.,* deployment of small cells, and exploiting multi-carrier and multi-radio access approaches [2]. The major challenge related to the core networks (standardized by 3GPP, IETF) is due to the fact that a few high level network elements, entitled *anchor points*, handle both the *Data plane* and the *Control plane*. Such centralization makes the network prone to several limitations, *e.g.,* sub-optimal routing, low scalability, signaling overhead, and lack of granularity on services [3, 4]. The straightforward and short-term solution to cope with the core networks issue may consist of operators investment to upgrade the resources. This approach is technically feasible. However, network operators always stand for the cost-effective and long-term solutions. Traffic offloading is an alternative approach to mitigate the traffic impact to limited resources in the core networks. That can be achieved by placing small-scale anchor points in the proximity of the access network to handle Mobile Nodes (MNs) connections and traffic locally [5]. This essentially leads to a decentralized network architecture (Fig. 1). Relocation of the mobile devices' edge anchor points helps maintaining efficient routes for MNs' connections. However, it demands additional mechanisms to maintain the MNs' ongoing data sessions, by enabling IP address continuity to the mobile devices and steering the data packets towards the new anchor points.

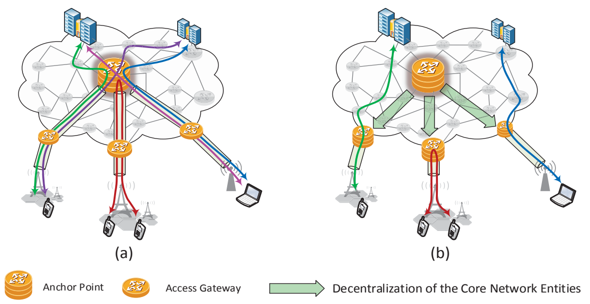


Fig. 1: A general view of the current (a) and decentralized (b) mobile network architectures.

Mobile networks have achieved a high-level of acceptance and become the major Internet access approaches. LTE (*Long Term Evolution*) network is expected to be the leading mobile networking technology in the coming 5-10 years. While it supports 47% (from 3.7 Exabytes) of mobile traffic nowadays, it is estimated to handle about 72% (from 30.6 Exabytes) of the worldwide mobile data traffic by 2020 [1]. Therefore, in this paper we pay special attention to an architecture for realizing a decentralized LTE network for the 3GPP access and introduce two novel solutions to support IP address and traffic continuity for the MNs in such an architecture.

Objectives and strategies for IP address mobility have been comprehensively discussed in *e.g.,* [6, 7]. Due to space limitation, we refer to our previous research [8] for more references and as the related works. There we analyzed and compared several key techniques, and summarized that the NAT (*Network Address Translation*) mechanism and SDN (*Software Defined Networking*) paradigm are as the promising enablers to handle MNs' traffic continuity in a mobile network with distributed anchor points. In our previous work [9], we proposed a NAT-based approach, utilizing the *Identifier-Locator* split concept, to support the MNs' IP address and traffic session continuity in a decentralized LTE network. In this paper, we develop two new approaches to handle the above mentioned issues, relying on SDN which is also one of the leading candidate technologies to efficiently address the demands of the future mobile networks (5G) [10]. Our main contributions in this paper are summarized as follows:

- We discuss the major modifications required in the current LTE network to realize a decentralized LTE network and to support the MNs' mobility in such an architecture.
- We develop two SDN-based mechanisms to enable the MNs' IP address and data traffic continuity in a decen-

978-1-5090-5856-3/17/$31.00 ©2017 IEEE

tralized LTE network.

- We extend the NS3-LENA simulation environment to support the implementation of a decentralized LTE network as well as the proposed solutions.
- Using the new implemented LTE architecture, we verify the function and evaluate performance of the developed mechanisms.

The remainder of this paper is organized as follows: § II provides concisely the necessary background about the current LTE network and its mobility management solutions. It also describes the main modifications required to realize a decentralized LTE network and handle MNs' mobility in this architecture. § III presents our proposed solutions and details their functions and components. § IV describes the implementation of the LTE's new architecture and developed IP mobility support solutions in the NS3-LENA. § V defines the performance metrics and presents the obtained simulation results. Finally, the paper ends up with conclusion at § VI.

II. LTE Network

This section gives a brief background about the current LTE system (§ II-A) and its mobility management mechanisms for the 3GPP access (§ II-B). It also describes an approach to realize a decentralized LTE network deployment (§ II-C) and the required modifications to support the MNs' mobility in this architecture (§ II-D). These are essential for understanding the problem statement being addressed in this paper.

A. Current LTE Architecture

The existing LTE network architecture is hierarchical and defines the EPS (*Evolved Packet System*) consisting of E-UTRAN (*Evolved Universal Terrestrial RAN*) and EPC (*Evolved Packet Core*). The E-UTRAN consists of a network of radio base stations (eNodeB – *evolved Node B*), that provides radio connectivity to the MNs. The EPC is a multi-access IP-based network that uses a common core network for the 3GPP and non-3GPP radio access, and fixed access.

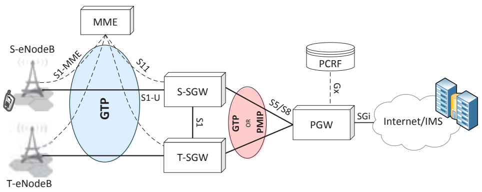


Fig. 2: Current LTE network architecture for the 3GPP access.

The EPC consists of four main elements (Fig. 2), that allow for the convergence of packet-based services [11]: The PGW (*Packet Data Network Gateway*) connects the EPC to external IP networks. It also carries out other functions, *e.g.,* IP address allocation, mobility management, and policy enforcement, *e.g.,* for QoS and charging. The SGW (*Serving Gateway*) provides data paths between the eNodeBs and PGW. It also handles the MNs' mobility between the local eNodeBs. The MME (*Mobility Management Entity*) controls the MNs in accessing to the LTE network. It also supports roaming and handover procedures. The PCRF (*Policy and Charging Rule Function*) determines QoS policies and charging rules to the PGW (if GTP is used) and SGWs (if PMIP is used).

B. Mobility Management in the Current LTE Network

A mobility management mechanism supports a set of procedures to enable seamless IP address and traffic session continuity for moving MNs within the network. In the current LTE for the 3GPP access, mobility management is based on either GTP (*GPRS Tunneling Protocol*) or PMIP (*Proxy Mobile*) protocols, where the PGW acts as a central mobility anchor point. When an MN connects to an eNodeB, its traffic is encapsulated in a GTP tunnel between the eNodeB and SGW and in another GTP (or PMIP) tunnel between the SGW and PGW. When the MN performs a handover between two eNodeBs to keep the ongoing IP flow(s) active, the new S1-U and S5/S8 tunnels are established between the EPC and E-UTRAN entities (Fig. 2), depending on whether the target eNodeB is served or not by the same SGW. The procedure described above shows that the existing data plane and mobility management procedure are highly hierarchical, demanding management of several tunnels between the PGW and the MNs. In a large LTE network, the PGW needs to maintain a considerable number (*e.g.,* a range of millions for a nationwide network) of per-user tunneling data, which may cause scalability and performance issues.

LIPA (*Local IP Access*) and SIPTO (*Selected IP Traffic Offload*) have been introduced in the 3GPP Release 10 to alleviate data traffic load on the LTE's core network. However, they are limited in supporting mobility functions only for the local MNs [12].

C. Decentralized LTE Architecture

The hierarchy of the data and control planes in the current LTE network can be eliminated by co-locating the SGW and PGW functions into a single entity, entitled as *S/PGW*. Accordingly, the S/PGWs are distributed closer to the edge network and can handle the MNs' connection functions, data traffic, and mobility locally. This approach basically leads to a decentralized LTE architecture and effectively reduces the load on the core network entities (Fig. 3). In the current LTE, MNs rarely change their attached PGW. If this happens (*e.g.,* during inter-operator roaming) a *PDN Disconnection* procedure will be triggered by the network for the IP flow(s) initiated at the previous PGW. Next, the new PGW anchors the MNs and serves the re-initiated traffic. This implies a disruption in the MNs' ongoing traffic, since the MNs' IP address(s) is not maintained but a new one is assigned. Following a decentralized architecture, relocation of the MNs' mobility anchors (S/PWG) will happen far more often. In this case, two layers of mobility management are needed in order to handle the MNs' IP traffic continuity during a handover with a S/PGW relocation: (i) within the EPC network (between the S/PGWs and eNodeBs); and (ii) outside the EPC network (between the S/PGWs and data networks), which hosts the MNs' corresponding services.

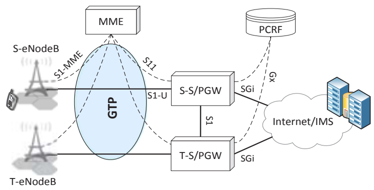


Fig. 3: Decentralization of the LTE core network architecture.

The following section describes the required modifications in the current EPC data and control planes for supporting the MNs' mobility within the EPC network of a decentralized LTE architecture. The mobility support outside the EPC network is based on our developed mechanisms, presented in § III.

D. Mobility Management within the EPC of a Decentralized LTE Architecture

IP address continuity, upon changing the attached PGW, is not supported in the current 3GPP's LTE standard. This is due to the fact that there is neither a signaling nor data forwarding scheme available between two different PGWs. Following a decentralized architecture, the existing control messages and traffic forwarding mechanisms, used during a SGW relocation can be revised to use the same IP address in different S/PGWs. This modification should enable the following functions: (***I***) the target S/PGW must be informed to implement a GTP bearer for the moving MN without requiring a new IP address allocation. This bearer is used to keep active the MN's flow(s) after handover; (***II***) the source S/PGW must be informed when the IP address used by the moving MN can be released. During an MN's attach procedure three concatenated segments of bearers are set up to forward packets end-to-end between the MN and the data network, *e.g.,* Internet or IMS (Fig. 4). The S1 and S5/S8 bearers use the GTP protocol to identify the individual connections between two nodes. A TEID[1] (*Tunnel Endpoint Identifier*) is assigned to each GTP bearer allowing the nodes to determine to which specific bearer a particular packet belongs. Each EPS Bearer is associated with one TFT (*Traffic Flow Template*), defining the filtering rules to differentiate data packets. In a decentralized EPC, the S5/S8 bearer is unnecessary and direct mapping of downlink traffic from the data network to an S1 bearer (DL-TFT→S1-TEID) can be performed in the S/PGW. A procedure to adjust the S1-TEID in a X2-based

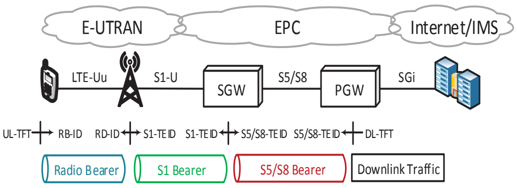


Fig. 4: EPS bearers in the current LTE network.

handover with SGW relocation[2] is specified in § 5.5.1.1.3 of [13]. We considered it as a baseline and defined a new handover mechanism to cope with the issues described in (***I***)[3]. We modified the *Create Session Request/Response* and *Modify Bearer Request/Response* messages between the MME and target S/PGW (which is replaced as PGW) to create a new DL-TFT in the target S/PGW. These changes handle the migration of an EPS bearer to a newly established data forwarding plane for the moving MN. To realize the function described in (***II***)[3], we modified the *PDN disconnection* procedure specified in § 5.10.3 of [13]. This procedure allows the MN to request for disconnection from the network. By receiving a *PDN Disconnection Request (LBI)* message, the MME exchanges the *Delete Session Request/Response* messages with the target S/PGW to inform the list of bearers to be released for a particular PDN connection. After the procedure of step 10 in § 5.10.3 of [13], the MME exchanges other *Delete Session Request/Response* messages with the source S/PGW to inquire it to drop the corresponding bearer context from its list and release the associated MN's IP address(es).

[1]For the PMIP, a GRE (*Generic Routing Encapsulation*) key is used as a identifier.

[2]The S1-based handover can also be used to handle MN's mobility during a SGW relocation (§ 5.5.1.2.2 of [13]). As we used the X2-based for development of our solutions, in this paper we ignore to present the required modifications for the S1-based approach.

[3]Due to space limitation, we only discuss the messages modified for our purpose, and refer the readers to § 5.5.1.1.3 and 5.10.3 of [13] for more information.

III. Proposed IP Mobility Support Approaches

This section describes the functional approach, architecture, components, and control messages of our proposed solutions to handle the MNs' mobility outside the EPC network, during a S/PGW relocation in a decentralized LTE architecture.

A. Functional Approach

In a decentralized LTE network, during a handover procedure with a S/PGW relocation, the MN's traffic forwarding and IP address continuity between two eNodeBs can be managed by the mechanism explained in § II-D. In this section, we present the functional approach for two new mechanisms to handle the MN's mobility support requirements in the transport network (outside the EPC network). The proposed schemes rely on SDN, offering a logical centralized control model and making networks more manageable and adaptable, which is ideally suited for the mobile networks.

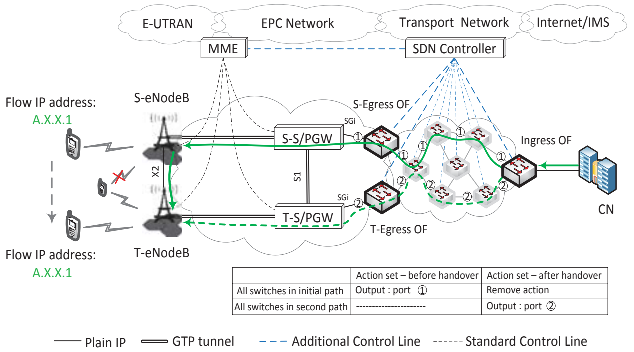
OpenFlow is the most common protocol used in SDN [14] and enables accessing the forwarding plane of the OpenFlow-enabled switches. Using it, the switches can be (re)configured according to the requirements of the network services. The OpenFlow-based switches may work in two modes: the *OpenFlow-only* or *OpenFlow-hybrid*. In the first mode the SDN Controller makes the forwarding decision for every packet in the network and the incoming data traffic always goes through the switch's flow-tables pipeline. However, the OpenFlow-hybrid switches support both the OpenFlow operations and the normal routing (L3) and switching (L2) functions. Therefore, in the latter case the SDN Controller may only make the forwarding decision for certain packets or flows depending on the application requirements, and steering of the remaining traffic can be handled by the conventional routing (switching) operations. Based on the operational modes and the capabilities offered by the OpenFlow switches, we develop two solutions to support the MNs' mobility outside the EPC in a decentralized LTE network. The solutions are different in terms of whether the transport network is deployed using the OpenFlow-only or OpenFlow-hybrid switches. The first approach (§ III-B1) is based on updating the flow's table(s) of the switches, without any IP address translation or modification in the flow's packets. The second solution (§ III-B2) relies on the modification of the headers of the flow-specific packets on the switches. These procedures are easily feasible using the set of features offered by the OpenFlow protocol.

B. Architecture

In a decentralized network architecture, the MN's IP address is anchored at a distributed anchor point and may need to be (temporarily) maintained to keep the ongoing sessions active, when the anchor point is relocated. This may change the MN's

IP address from a *routable* (topologically correct) into a *non-routable* (topologically incorrect) address at the new anchor point. In this condition, the transport network needs to steer the MN's downlink traffic to the new anchor point. Fig. 5 and 6 show the architecture of proposed approaches to handle the above mentioned aspects in a decentralized LTE network.

1) ***Transport Network with the OpenFlow-Only Switches****:* Fig. 5 shows the implementation architecture for the first solution, where the transport network is set up using the OpenFlow-only switches. In this scenario the SDN Controller has one-way connections to all the switches and manages them to redirect the MNs' traffic.



	Action set – before handover	Action set – after handover
All switches in initial path	Output : port ①	Remove action
All switches in second path	--------------------	Output : port ②

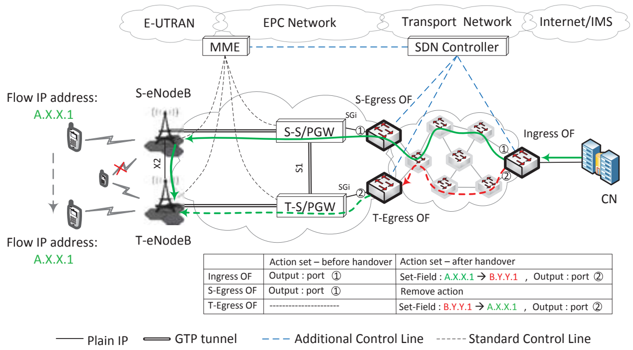
Fig. 5: SDN-based traffic redirection using OpenFlow-Only switches.

Due to the lack of L3 routing functionality in the transport network switches, the MNs' downlink paths towards the S/PGWs need to be set up by the SDN Controller, through updating the flow tables in the switches. Assume that an MN attaches to the source eNodeB and gets an IP address allocated from the source S/PGW (*e.g.,* A.X.X.1), to inquire some data from the CN. The MME notifies the SDN Controller about this. Accordingly, the SDN Controller sends a *Modify-State* message [14] to all the switches to add an *Output* action to set up a specific route for the MN's flow(s) towards it's current S/PGW. This action is added to the set of actions for the flow(s) belonging to address (A.X.X.1), and specifies the port on the switch(es) that the packets must be routed through (*e.g.,* the ports with number ①, Fig. 5). Upon receiving the MN's downlink packets in the transport network (*i.e.,* Ingress OF), the switch looks up the IP header of the packets to find a match in its flow tables (using the *<ingress port, source IP address, destination IP address>* information). Following the *Output* action added by the SDN Controller, the appropriate interface (*i.e.,* port ①) is selected as the output port to send the received packets. The same procedure is repeated at the other switches of the transport network to forward the MN's downlink traffic to the source S/PGW (the green solid-line between the Ingress OF and source S/PGW).

As previously described during handover procedure, the MN's downlink traffic forwarding between the source and target eNodeBs is handled by the procedure presented in § II-D (the green solid-line between two eNodeBs).

When the MN is handed over to the target eNodeB, in transport network the *Output* actions of the OpenFlow switches are proactively modified to forward the MN's traffic to the correct target S/PGW. To accomplish this the SDN Controller sends a *Modify-State* message to the appropriate switches in order to add/modify the action set for the desired flow(s) accordingly (*i.e.,* choosing interface ② as the output port). Next, a similar procedure as described for setting up the MN's downlink initial path (before handover), is fulfilled in the switches to forward the MN's traffic to the new position (the green dashed-line between the Ingress OF and target S/PGW). Furthermore, a *Modify-State* message is also sent by the SDN Controller to the switches located on the MN's downlink initial path, to remove the previously added flow(s) from their flowtables. When the MN terminates the flow(s), the corresponding entries must be removed from the switches' flow tables.

2) ***Transport Network with the OpenFlow-Hybrid Switches****:* In this solution, (re)direct of the MN's downlink traffic is realized by the SDN Controller, managing only the Ingress and Egress switches of the transport network. Traffic forwarding between the Ingress and Egress points in the core, is done via L3 routing in the switches (which can also be replaced with normal IP routers), instead of flow forwarding (Fig. 6).



	Action set – before handover	Action set – after handover
Ingress OF	Output : port ①	Set-Field : A.X.X.1 → B.Y.Y.1 , Output : port ②
S-Egress OF	Output : port ①	Remove action
T-Egress OF	--------------------	Set-Field : B.Y.Y.1 → A.X.X.1 , Output : port ②

Fig. 6: SDN-based traffic redirection using OpenFlow-Hybrid switches.

In this scenario, the SDN Controller sends a *Modify-State* message to the Ingress and source Egress switches, upon being informed by the MME about the newly attached MN to the source S/PGW. Accordingly, an *Output* action is added to the flow tables of the Ingress and source Egress switches. This action specifies the appropriate port (*e.g.,* the ports with number ①, Fig. 6) to forward incoming downlink traffic to the relevant switch(es) in the transport network (from the Ingress switch) and the correct S/PGW (from the source Egress switch). Within the transport network, the MN's downlink traffic forwarding is fulfilled via an L3 routing scheme (the green solid-line between the Ingress OF and source S/PGW).

During the MN's handover procedure, traffic forwarding between two eNodeBs is handled by the same mechanism as explained in the previous section.

After completing the handover, the target S/PGW allocates an IP address (e.g., *B.Y.Y.1*) from its address pool to the MN (§ II-D). This address is not advertised to the MN and is only used to steer the MN's downlink traffic within the transport network. For this, first the SDN Controller is notified by the MME about the MN's primary IP address allocated by the source S/PGW (*i.e.,* A.X.X.1) and its new IP address allocated by the target S/PGW (*i.e.,* B.Y.Y.1). Next, the SDN Controller uses this information to construct the *Modify-State* messages and sends them to the Ingress and target Egress switches in order to add/modify the *Output* and *Set-Field* actions [14] in the flow's set of actions. The *Set-Field* action in the Ingress switch, specifies the change from the MN's old IP address

(A.X.X.1) → new one (B.Y.Y.1) in the incoming packets. The packets are than forwarded to the transport network through the proper port (*e.g.,* port number ②) specified by the SDN Controller (using the *Output* action). Afterwards, within the transport network data forwarding towards the target Egress switch is fulfilled via a normal IP routing procedure (the red solid-line between the Ingress OF and target Egress OF). Upon receiving the packets, the target Egress switch uses the *Set-Field* instruction to reverse the modified packets' header to the original one (translating back the IP: A.X.X.1 → B.Y.Y.1). Next, the incoming data packets are forwarded to the correct target S/PGW via the port determined by the SDN Controller (*e.g.,* port number ②, Fig. 6), as the Egress switch may connect to more than one S/PGW (the green dashed-line between the target Egress OF and target S/PGW).

In both approaches (III-B1 and III-B2), upon arrival of the data packets at the target S/PGW, they are processed and encapsulated into a GTP tunnel and then forwarded to the target eNodeB to be delivered to the MN, via the LTE air interface.

During a handover procedure the MME has the knowledge about the IP address of the active flow(s) after handover and the new IP address allocated by the target S/PGW. Therefore, in both approaches, we choose it to signal the SDN Controller about the required information.

An MN may have multiple IP addresses (or multiple active flows per address). In such a case, in order to decrease number of the flows' redirection paths, they can be initiated based on the IP (or MAC) address of SGi interface of the MN's current S/PGW. For this, the flow matching in the Egress switches can be performed, in order to distinguish the packets belonging to different flows, based on a combination of the source (CN) IP address and the transport layer ports. In all the other switches flow matching can be done only based on the destination (SGi) IP address of the incoming packets. Detailed information about the flow matching can be found at § 5.3 of [14].

C. Control Messages

According to our proposed schemes, two steps of signaling must be performed to create/remove traffic redirection paths in the transport network (Fig. 7). This section describes briefly the messages used for these procedures as follows.

1) Messages from MME → SDN Controller: The MME uses two messages to signal the SDN Controller. *Create Path* message is used to set up a new downlink path, and *Remove Path* message is used to remove the previously established downlink path in the transport network. These messages may carry the IP address of the MN's assigned by the source S/PGW (during the PDN connection procedure), the IP address of the MN's active flow(s) after handover and/or the new IP address allocated by the target S/PGW (during the handover and after the PDN disconnection procedure). In different procedures, the MME may receive this information via the *Create Session Response*, *Modify Bearer Response*, or *PDN Disconnection request* messages (Fig. 7).

2) Messages from SDN Controller → OpenFlow Switches: SDN Controller uses a set of *Modify-State* messages to add/modify *Set-Field* and/or *Output* action(s) in the flow's set of actions the corresponding switches. The action(s) and the switches to be messaged depend on the implementation of the transport network. A *Modify-State* message may carry the MN's old and/or new IP addresses and/or the ID of the output interface in the switches (Fig. 5 and Fig. 6).

D. Control Messages Flow Diagram

This section describes the flow of control messages (Fig. 7) exchanged between the EPC and SDN components during different procedures as follows:

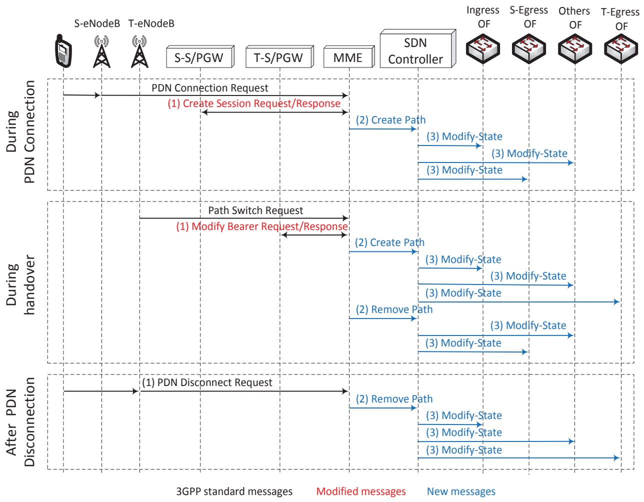


Fig. 7: Control messages flow diagram of the SDN-based solutions.

1) During PDN Connection Procedure: After receiving a *PDN Connection Request* message, the MME sends a *Create Session Request* to the source S/PGW and asks to establish a new bearer to the attached MN. Via (1) the source S/PGW replies the *Create Session Response* message and notifies the MME about the assigned IP address to the MN. (2) the MME looks at the MN's context and retrieves the allocated IP address, to construct a *Create Path* message. Accordingly, the message is sent to the SDN Controller. (3) the SDN Controller uses this information to create the *Modify-State* messages and sends them to the relevant switches in the transport network. Upon receiving the *Modify-State* messages, the switches add new action(s) to the corresponding flow's set of actions in order to establish the initial path for the MN's downlink traffic.

2) During Handover Procedure: Via (1) the target S/PGW sends the *Modify Bearer Response* message to notify the MME about the newly assigned IP address to the MN (§ II-D). (2) the MME looks also at the MN's context and retrieves the IP address of the MN's active flow(s) after handover. Next, using the new and/or old IP addresses, it makes a *Create Path* (or *Remove Path*) message and sends it to the SDN Controller. (3) the SDN Controller uses this information to construct the *Modify-State* messages and sends them accordingly to the relevant switches in the transport network. Upon receiving the *Modify-State* messages, the switches add new action(s) to the corresponding flow's set of actions in order to set up the MN's downlink traffic redirection path (or to remove the previously established downlink path) in the transport network.

3) After PDN Disconnection Procedure: Via (1) the MN sends a *PDN Disconnect Request* message including the LBI (*Linked EPS Bearer ID*) to the MME and asks for disconnection from the network (§ II-D). (2) the MME extracts the current and previously assigned IP addresses of the MN from the MN's context and LBI. A *Remove Path* message is created using this information, and then it is sent to the SDN Controller. (3) The SDN Controller uses this data to create the *Modify-State* messages and then forwards them to the already signaled OpenFlow switches. When the switches receive this message, they look up at the related flow's tables for an entry whose data matches with the received information. If any entry is found, the action(s) (specified in the received message) are removed from the set of actions of the terminated flow.

IV. Set up of the Simulation Study

This section presents the evaluation scenario, implementation of the components, and parameters setting in the simulation environment. We set up the simulation environment as realistic as possible to get reliable results, as described below.

A. Evaluation Scenario

The logical network topology of the evaluation scenarios are the ones shown in Fig. 5 and Fig. 6. During 20 second simulation, 120 MNs attach to both eNodeBs and generate the E-UTRAN traffic according to Table II. 30% of MNs from the source eNodeB, running VoIP application move to the target eNodeB at different times between 10.51 to 12.96 seconds. The X2-based (within the EPC) and the SDN-based solutions (outside the EPS) forward the MNs' downlink traffic to the target position.

B. Implementation of the Scenarios in the NS3-LENA

To evaluate the proposed solutions, we used the NS3-LENA simulations, since there is no decentralized LTE system deployment that could be used to study and verify the discussed MN's mobility support approaches. We extended the NS3-LENA environment to implement the following components, needed for the evaluation. The source codes to implement all of the modules can be found in [15].

1) Decentralized LTE Network: Several modifications are needed in the existing version of the NS3-LENA to implement a decentralized LTE network: *(i) Instantiation of Multiple S/PGWs:* we enhanced the *EpcHelper* module of the NS3-LENA to implement two standalone EPS subsystems that use independent S/PGWs with different pool of IP addresses. The S/PGWs serve the separate eNodeBs, but use a shared MME; *(ii) Implementation of X2-based Handover Procedure with S/PGW Relocation:* we modified the currently deployed X2-based handover procedure of the NS3-LENA to support relocation of S/PGW during the MNs handover and to realize the functions described in § II-D.

2) Transport Network: The transport network topology is according to a small part of the EBONE (one of the European ISPs) network, covering The Netherlands, north-east of Belgium and north-west of Germany. To implement it we used a map provided by the Rocketfuel project [16].

3) SDN Functions: We used the OFSID (*OpenFlow Software Implementation Distribution*) library, already presented in the NS3 stable releases to implement the SDN functions. Using the *OpenFlowSwitchNetDevice* and *OpenFlowSwitchHelper* entities of it, OpenFlow switch capabilities can be added to a node defined in the NS3 environment. The SDN Controller functions can be added under the *ofi namespace* using *ofi::Controller*. More Information can be found in [17].

4) Control Messaging: To implement the control messaging procedures described in § III-D, the *UdpSockets* are set up between the MME, the SDN Controller, and the OpenFlow switches entities.

5) The MNs Movement: In NS3-LENA, the MN's handover time is based on a pre-defined schedule, set up in the simulation. Therefore, no MNs' movement is needed to trigger the handover procedures. The MNs are only placed in positions at the same distances from the eNodeBs and a distribution of dwell time is used to trigger the handovers. We used the *Fluid-Flow* mobility model [18] to derive the average dwell time of the MNs in each S/PGW. As a S/PGW relocation most likely happens for the MNs that are on a highway, we used the *Free Speed Distributions* model [19] to compute the velocity of the MNs. We used a *Normal Distribution* with the *Mean = 32.1 m/s* and *Standard Deviation = 4.33* [19]. For the sake of simplicity we assumed that the MNs move in a straight road between the S/PGWs.

C. Simulation Parameters

This section presents the parameters setting in the simulation environment.

1) E-UTRAN Setting: Table I summarizes values of the configured parameters in the RAN. The values are based on the LTE release-8 specifications, implemented in the NS3-LENA.

TABLE I: The E-UTRAN parameters.

Parameters	Value
Uplink and Downlink bandwidth	5 MHz
Source eNodeB uplink / downlink	2535 / 2655 MHz
Target eNodeB uplink / downlink	2540 / 2650 MHz
Transmission mode	MIMO 2x2
MN / eNodeB transmission power & noise figure	26 / 49 dBm & 5 dB
Cell radius / distance	2 / 4 Km

2) E-UTRAN Traffic: We used the *traffic mix* model (Table II) specified in [20] to generate the RAN traffic. VoIP is selected as the traffic generated by the moving MNs. Other types of traffic are used to generate the RAN background traffic by the fixed MNs attached to the both eNodeBs.

TABLE II: The E-UTRAN traffic model.

Application	Traffic	Percentage of MN
VoIP	Real-time	30%
FTP	Best effort	10%
HTTP	Interactive	20%
Video	Streaming	20%
Gaming	Interactive real-time	20%

3) EPC and Transport Networks Setting: The values of parameters set up for the EPC and transport networks are shown in Table III. The size of buffer for the entities in both networks is set to $(\frac{C \times RTT}{10})$ [21]. Where C and RTT denote the link speed and the *Round Trip Time* ($= 250ms$ [21]) of the flows in the network, respectively.

TABLE III: The EPC and transport network parameters.

Parameters	Value
Transmission technology	Ethernet
MTU size	1.5 KB
Transport / EPC network links data rate	10 / 1 Gbps
Queue scheme	Drop-tail
Transport / EPC network nodes buffer size	31.250 / 3.125 MB
Radius of a tracking area (TA) of each S/PGW	50 KM

4) The EPC and Transport Networks Traffic: We used the PPBP (*Poisson Pareto Burst Process*) model [22] to generate a realistic Internet traffic in the wired networks. 80% is chosen as the maximum level of link utilization for the both networks. The rest of available capacity is used to transfer the VoIP traffic and also to keep as the safety capacity.

V. Performance Metrics, Results, and Discussion

In this section, using the simulations specified above, we evaluate the seamlessness of the proposed schemes. For this, we define the following performance measures for the MNs performing a handover with S/PGW relocation.

A. Average Latency of Data Packet Delivery Before and After Handover

Fig. 8a presents for different approaches, the average latency of the data packets received by the moving MNs via the alternative downlink paths (before and after handover). The graphs clearly demonstrate that none of the proposed solutions have significant impact on the latency of the data packets, redirected to the MNs after a handover. Slightly better result is observed for the scenario with the OpenFlow-Only switches, compared to the one with OpenFlow-Hybrid switches. This is due to the fact that, in the former solution only one action (*Output*) is added to the flow's set of actions in the switches. This accordingly imposes lower processing latency on the incoming flows, compared to the second approach, where two actions (*Output* and *Set-Field*) must be added to the switches.

The placement of the SDN Controller in the transport network can be critical, as it must receive within reasonable time the MN's mobility information from the MME in order to decide for the switches and setting up the traffic redirection paths. Therefore, we also studied the impact of it's position in the network architecture (§ III-B). Given that the MME uses the transport network to signal the SDN Controller, the obtained results show that no significant impact ($\leq$ 3 ms) is noted in the average latency after handover, where the distance between the SDN Controller and MME is changed from 1 to 7 hops. In the simulation scenarios, the Egress OF switches are placed in the fixed positions with one hop distance to the S/PGWs. Positioning of the Ingress OF switch mainly depends on the topology (and size) of the transport network and should be as close as possible to the Internet (*e.g.,* on the transport network's Internet PoPs (*Point-of-Presences*)). The obtained results show that after handover, the average latency of data packet delivery has a little variation ($\leq$ 2 ms), where the distance between the Ingress and Egress OF switches is changed from 1 to 7 hops. This accordingly may slightly affect the throughput of the proposed solutions.

B. CDFs of Latency of the MNs' Downlink Data Packets

Fig. 8b presents the CDFs (*Cumulative Distribution Functions*) of latency of the first data packets delivered to the moving MNs after completion of a handover, using the proposed approaches. We select the first data packet of the MNs, as it is the most delayed packet after the handover procedure. This is because its delivery time is directly influenced by the time required to establish the traffic forwarding paths in the transport network. The results show that both approaches are fast enough to easily meet the maximum allowed one-way latency of 150 ms [23] as a threshold for the VoIP application. It is also observed that the approach with the OpenFlow-Only switches outperforms the one with OpenFlow-Hybrid switches. As mentioned before, this is because in the former one less number of actions must be applied to the incoming flows during the traffic redirection procedure, compared to the second approach. Fig. 8b also shows, for both scenarios, the CDFs of latency of the data packets forwarded from the source eNodeB to the target eNodeB, through the X2 path during the MNs' handover. The graph shows that both proposed solutions significantly outperform the X2-path traffic redirection. This implies that in the case of using a mobility prediction scheme [24] the traffic redirection paths in the SDN-based solutions can be set up a priori and hence avoid the usage of the X2 path data forwarding.

C. Packet Loss Ratio in the Redirected Downlink Data Traffic

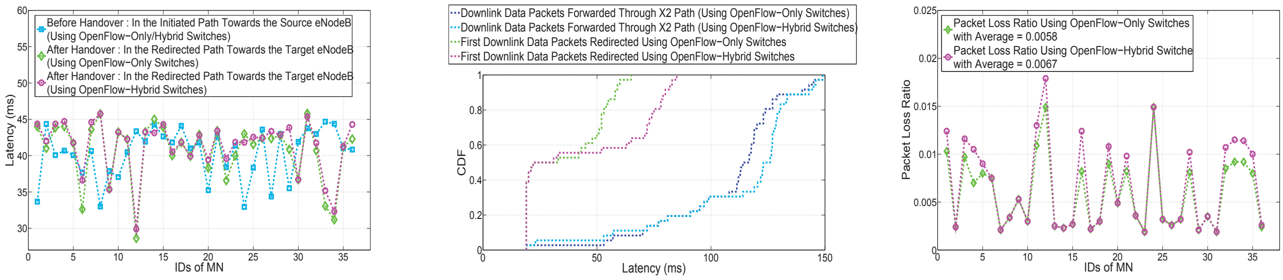
The X2 path is used to forward the MNs' downlink data during handover procedure from previous to the new eNodeB, and also after the handover when the traffic redirection paths of the SDN-based solutions are not yet set up. By using the X2 path for 10 ms after handover procedure (by setting *Delete Session Timer* = 10 ms in the MME [13]), no loss is present in the MNs' data packets. It implies that 10 ms is sufficient time, for both schemes, to establish the traffic forwarding paths in the transport network. Mobile network operators may not prefer to use the X2 path after the MNs' handover (setting *Delete Session Timer* = 0). In this case, the moving MN's session is removed from the source S/PGW, once the S1 bearer is initiated at the target S/PGW. This may result in some packet loss for the MNs moving to a new S/PGW (Fig. 8c). The results show that, the solution with OpenFlow-Only switches is more reliable than the one with OpenFlow-Hybrid switches. This is due to the fact that, the former scheme is faster than the second one in setting up the traffic redirection paths and the X2 path traffic forwarding takes shorter during the handover procedure. This implies that the solution with OpenFlow-Only switches provides a lower packet loss ratio while the X2 path is not even used during the the MNs' handover.

D. Control Messaging Overhead of the Proposed Solutions

In the proposed approaches, several messages must be exchanged between the MME and SDN entities in order to set up a traffic redirection path (Fig. 7). Assuming that only the messages from the SDN Controller towards the OpenFlow switches are sent through the transport network, Fig. 9 shows the overhead of the control messages of the proposed solutions on the network (while distance between the Ingress and Egress OF switches = 6 hops), in terms of the number of MNs handed over to a new S/PGW. It is observed that the solution with OpenFlow-Hybrid switches outperforms the one with OpenFlow-Only switches and has lower impact on the network load. In this figure, we also compare our approaches with the overhead impact of the existing centralized solutions (the GTP [11] and PMIP [13] protocols in the current LTE network), where a SGW relocation happens during the MNs handover.

VI. Conclusion

In this paper, we have developed two novel schemes to support IP address and traffic continuity for the MN's active flow(s) after a handover with S/PGW relocation in a LTE decentralized architecture. Our solutions are based on SDN,



(a) Average latency of the MNs' downlink data packets delivery before and after handover procedures.

(b) CDFs of latency of the MNs' downlink data packets via the X2 and proposed schemes traffic redirection paths.

(c) The MNs' downlink data packets loss ratio without using the X2-path traffic redirection.

Fig. 8: Evaluation results of the SDN-based traffic redirection solutions in a decentralized LTE network.

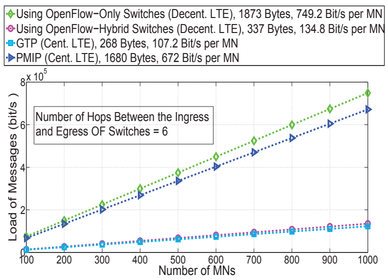


Fig. 9: Overhead of the control messages.

which is also a promising technology for the 5G mobile network. The proposed solutions are different based on using the OpenFlow-only or OpenFlow-hybrid switches for the implementation of the transport network. In the former approach, all the switches in the transport network must be controlled by the SDN Controller to set up traffic redirection towards the MN's current anchor points. In the latter scheme, only the Ingress and Egress switches need to managed by the SDN Controller and the rest can be even replaced by the normal L3 routers. Therefore, implementation of the solution with OpenFlow-Hybrid switches is easily feasible with a trivial overhead and complexity. Detailed simulations show that both proposed solutions are fast enough in setting up the traffic redirection path considering the maximum allowed delay threshold for real-time applications (*e.g.,* VoIP). The solution with OpenFlow-Only switches outperforms the one with OpenFlow-Hybrid switches in terms of the delivery latency and loss ratio of the downlink data packets, but provide higher overhead of the control messages on the network. Considering a scenario with the MME relocation and utilizing more than one SDN Controller within the network are other research areas of interest, as future work.

References

[1] Cisco Visual Networking Index, Global Mobile Data Traffic Forecast Update, 2015−2020, February 2016.

[2] Dealing with Density: The Move to Small-Cell Architectures. In *RUCKUS Wireless White paper*, 2014.

[3] H. A. Chan, H. Yokota, J. Xie, P. Seite, and D. Liu. Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues. *JCM*, 2011.

[4] Z. F. L. Bokor and S. Imre. Flat architectures: Towards scalable future internet mobility. In *The Future Internet*. Springer, 2011.

[5] T. Taleb, K. Samdanis, and F. Filali. Towards supporting highly mobile nodes in decentralized mobile operator networks. In *IEEE ICC*, 2012.

[6] J. C. Ziga, C. J. Bernardos, A. de la Oliva, and A. Reznik. Distributed mobility management: a standards landscape. *IEEE Communications Magazine*, 2013.

[7] Fabio Giust. Distributed Mobility Management for a Flat Architecture in 5G Mobile Networks: Solutions, Analysis and Experimental Validation. *Universidad Carlos III de Madrid, Spain*, 2016.

[8] M. Karimzadeh, L. Valtulina, and G. Karagiannis. Applying SDN/OpenFlow in Virtualized LTE to support DMM. In *4th CLOSER*, 2014.

[9] M. Karimzadeh, L. Valtulina, A. Pras, H. Van den Berg, R.O. Schmidt, M. Liebsch, and T. Taleb. Double-NAT Based Mobility Management for Future LTE Networks. In *IEEE WCNC*, 2017.

[10] 5G PPP Architecture Working Group. View on 5G Architecture . In *EuCNC*. 5G-PPP, 2016.

[11] 3GPP.TS.23.402. Universal Mobile Telecommunications System (UMTS); LTE. 2015.

[12] 3GPP SR 23.859. Technical Specification Group Services; LIPA Mobility and SIPTO at the Local Network, Release 11 and 12. 2012.

[13] 3GPP.TS.23.401. General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access. 2011.

[14] OpenFlow Switch Specification Version 1.5.0. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow.

[15] https://github.com/MortezaKarimzadeh/SDN_Mobility_LTE.

[16] Rocketfuel: An ISP Topology Mapping Engine. http://research.cs.washington.edu/networking/rocketfuel/.

[17] OpenFlow switch support. https://www.nsnam.org/docs/models/html/openflow-switch.html.

[18] Radhika Ranjan Roy. *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer, 2010.

[19] S. P. Hoogendoorn. Vehicle-Type and LaneSpecific Free Speed Distributions on Motorways: A Novel Estimation Approach Using Censored Observations. In *Journal of the Transportation Research Board*, 2006.

[20] R. Irmer. Next Generation Mobile Networks Radio Access Performance Evaluation Methodology. In *NGMN Alliance White paper*, 2008.

[21] Y. Ganjali and N. McKeown. Update on Buffer Sizing in Internet Routers. *ACM SIGCOMM*, 2006.

[22] D. Ammar, T. Begin, and I.G. Lassous. A New Tool for Generating Realistic Internet Traffic in NS-3. In *ICST (SIMUTools)*, 2011.

[23] http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf.

[24] M. Karimzadeh, Z. Zhao, L. Hendriks, R. de O. Schmidt, S. la Fleur, H. van den Berg, A. Pras, and T. Braun. Mobility and Bandwidth Prediction as a Service in Virtualized LTE Systems. In *4th IEEE CloudNet*, 2015.