Contreras, L. M., Cominardi, L., Qian, H. & Bernardos, C. J. (2016). Software-Defined Mobility Management: Architecture Proposal and Future Directions. *Mobile Networks and Applications, 21*(2), 226–236.

DOI: 10.1007/s11036-015-0663-7

Software-Defined Mobility Management: Architecture Proposal and Future Directions

Luis M. Contreras[1], *Telefónica*, luismiguel.contrerasmurillo@telefonica.com
Luca Cominardi, *IMDEA Networks Institute* and *Universidad Carlos III de Madrid*, luca.cominardi@imdea.org
Haiyang Qian, *China Mobile USA Research Center*, haiyangqian@chinamobile.com
Carlos J. Bernardos, *Universidad Carlos III de Madrid*, cjbc@it.uc3m.es

Abstract

A common characteristic for all of the uses in 5G wireless networks is the ubiquity and the almost permanent connection to the mobile network to get access to external applications. This really imposes a challenge in the signaling procedures provided to get track of the user and to guarantee session continuity. The mobility management mechanisms will play a central role in the 5G networks because of the always-on connectivity demand. This article presents a software defined approach to mobility management procedures addressing the present challenges and proposing some future directions for a more efficient service provision and a better usage of the network resources. The feasibility of such a Software-Defined Mobility Management architecture is assessed in a specific test-bed.

Keywords

SDN; Mobility Management; OpenFlow;

1 Introduction

The 5G wireless networks are expected to support the needs of an hyper-connected society which is continuously demanding very high data rate access, requiring a wider coverage, and offering an increasing number of almost permanently connected devices.

Signaling load is one of the most problematic aspects in existing mobile core networks [1], [2]. On one hand, the number of mobile devices supported by the network (which is the final responsible for their reachability as they move) is ever increasing. On the other hand, the need to increase both capacity and coverage is fueling the deployment of smaller cells that in turn produce more signaling because of more frequent handover events. Figure 1 shows the evolution of the peak CPU load of a Serving GPRS Support Node (SGSN) observed in a real operation as a function of the number of maximum weekly Simultaneous Attached Users (SAUs) along 23 weeks. As can be seen in Fig.1 the processing needs increase with the number of users almost lineally, then imposing limits to the number of supported users being handled by dedicated boxes because of their constrained resources. This will happen even if no traffic is generated by this users. The main limitation today resides in the control part and not in the data plane, which has enough throughput capacity for traffic forwarding, then driving to a constant upgrade of the mobile core network elements.

Specifically, the Mobility Management (MM) is a task that imposes serious constraints to the processing capacity of the existing mobile core network elements. Even if a mobile terminal is not active in a communication, the network should manage its mobility and provide the network resources for facilitating such communication when it is set up.

---

[1] Corresponding author:
Luis M. Contreras
Distrito Telefónica, Edificio Sur 3, Planta 3
28050 Madrid, Spain
T: +34 91 482 2785 / M: +34 608 650 415
luismiguel.contrerasmurillo@telefonica.com

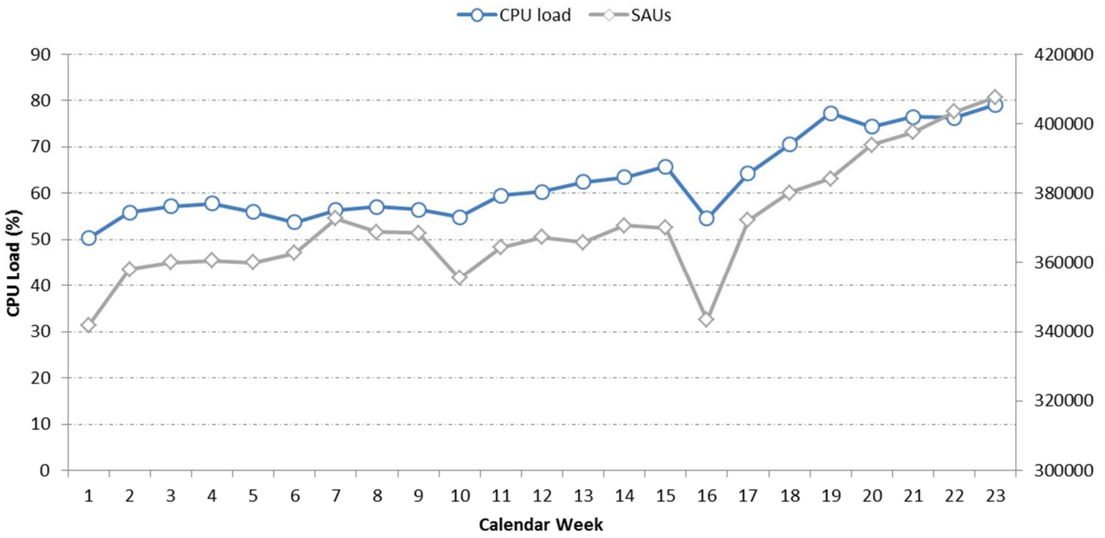


Figure 1. Peak CPU load versus maximum weekly SAUs in a SGSN

In this situation it is reasonable to look for decoupling the control plane from the data plane in mobile core networks. In fact, mobile protocols already imply such conceptual separation. However the common incarnation of the mobile core nodes today is the vertical integration of both control and data plane functions in purpose-built, monolithic hardware.

The Software Defined Networking (SDN) paradigm [3] proposes a truly separation of the control plane from the data plane in network devices, logically centralizing the control functions in such a way that they can be implemented on commodity computing platforms, more powerful in terms of processing capacity, and then, in a more scalable way. In this way, situations like the one reflected in Fig. 1 can be overcome just adding more external and even distributed computing capabilities in a flexible manner.

The communication between the centralized controller entity and the merely traffic forwarding devices could be supported by what is called southbound interface protocols. One of the candidate protocols for it is OpenFlow, which is being standardized by the Open Network Foundation (ONF). A Mobile Packet Core (MPC) architecture based on SDN has been already proposed in [4] describing the evolution of the existing MPC architecture towards a new architecture with the separation of control and user plane by using a programmable infrastructure. This article is focused on Mobility Management procedures in a solution as the one described in [4], providing insight on technical feasibility and performance evaluation. A more general review of advances in SDN propositions for mobile and wireless networks can be found in [5].

The structure of the paper is as follows. Section 2 describe current mobility management protocols in both 3GPP and IETF standardization bodies. Section 3 introduces the usage of SDN for mobility management. In Section 4 an experimental evaluation of the proposed architecture is provided. Future directions for SDN based mobility management are proposed in Section 5. Finally, Section 6 summarises the conclusions of the paper.

2 Existing Mobility Management Protocols

End-users require maintaining their data connectivity while changing the point of attachment to the network. This feature is enabled by the mobility management (MM) protocols that ensure the reachability of the mobile terminal. In deep surveys of MM protocols can be found in [6],[7].

Mobility management protocols can be categorized as host-based and network-based MM protocols. In the former case the IP-capable mobile terminals are aware of their IP layer mobility and have to do operations in order to maintain their ongoing communication sessions. In contrast, in the latter case, the functionality to detect and manage the terminal movement resides in the network, and the terminals only perform the standard IP operations. From an operator point of view, this kind of mobility management approach is advantageous because it permits to support mobility without relying on functionality and specific configurations in the mobile node.

In this article, we focus the discussion on the network-based mobility management protocols.

2.1 Mobility Management in 3GPP/LTE

The Evolved Packet Core (EPC) in LTE [8], as depicted in Fig. 2, is in charge of providing IP connectivity and session continuity to the mobile terminal or User Equipment (UE) as moves around. To provide such IP connectivity, the EPC extends an overlay tunnel from the eNB in the access to the PGW deep in the core, to establish connectivity between a mobile terminal to the target external network (e.g., Internet). Those tunnels can be based on different transport protocols along the different EPC interfaces. In legacy mobile systems, the GPRS Tunneling Protocol (GTP) [9] has been deployed to provide IP data connectivity from the core network up to the mobile terminal. GTP enables the overlay tunneling that traverses the core entities, and it carries some other useful information such as policy control and QoS.

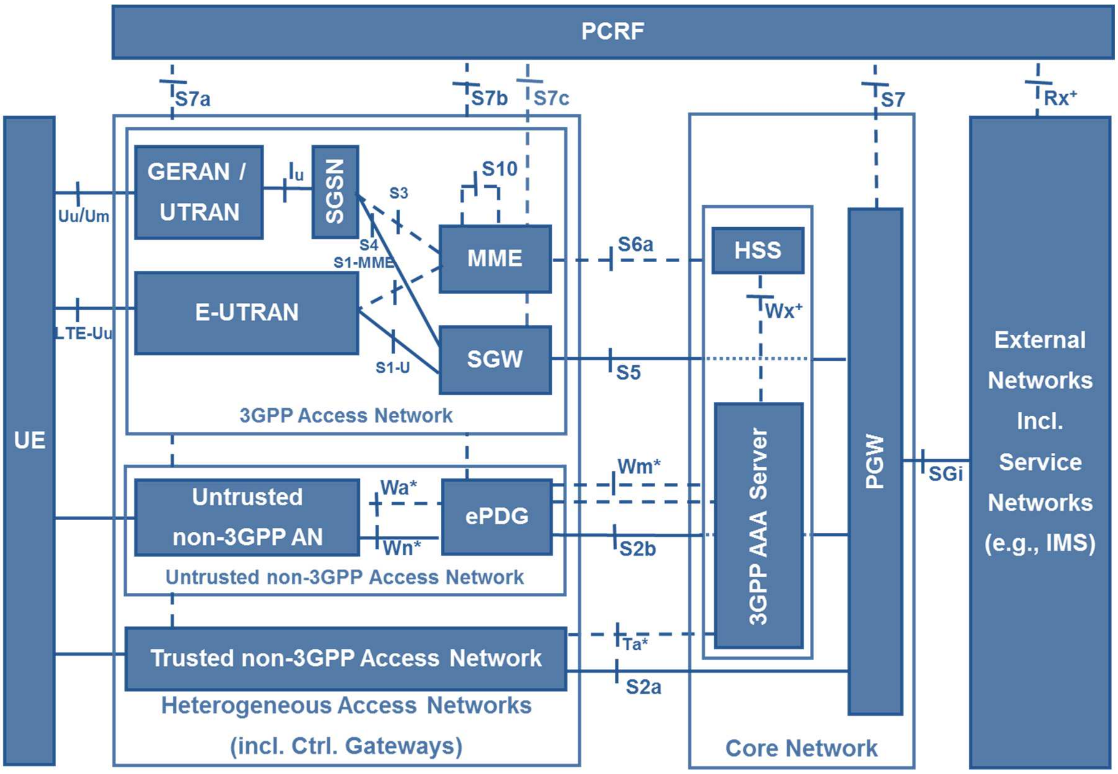


Figure 2. LTE Evolved Packet Core [8]

The final objective of managing the mobility of the terminal is to guarantee that the IP connectivity is maintained independently of the access point of the user. As a consequence of that, the overlay tunnel has to be re-built to follow the terminal movement as the end user moves.

The overlay tunnel which transports the terminal connection from the access to the core is known as bearer. There are different types of bearers, namely default and dedicated bearers. The GTP components for user plane (GTP-U) and control plane (GTP-C) protocols use certain mechanisms for setting up, modifying, and tearing down the EPS bearers and specifying the corresponding QoS levels. The default bearer is set up by the packet core as soon as the user device registers, being the basic connectivity artifact. This bearer provides an always-on connectivity to the Internet. In addition to that, a Guaranteed Bit Rate (GBR) bearer ensures a binary rate. GBR bearers are suitable for delay-sensitive traffic such as voice. A non-GBR bearer usually is used for non-real time traffic such as web. The dedicated bearers are set up on demand for specific needs and could have specific requirements in terms of QoS, for instance. A user device can receive more dedicated bearers (up to 11) which share a single IP address. Both types of bearers have to be managed as the terminal moves changing its points of attachment.

GTP-U protocol maintains a mapping from each bearer to a bidirectional tunnel. Each tunnel has two end-points or tunnel end-point identifiers (TEIDs), one for uplink traffic, and the other for downlink. These tunnels are established by GTP-C signaling messages, and are stored at both end-points. One of the responsibilities of the PGW is to assign each incoming flow to the correct EPS bearer. To achieve this, each EPS bearer contains a traffic flow template (TFT). Each template contains a set of filters. The filters matches with flows assigned to the bearer. The PGW then looks up the corresponding GTP-U tunnel, adds certain headers, looks up the IP address of the SGW of user device, and then forwards the packet to the SGW. When the packet arrives at the SGW, the SGW opens the GTP-U header and reads its TEID. Using this information, it identifies the corresponding EPS bearer, and looks up the destination eNB and the TEID. It then establishes another tunnel to forward the packet to the correct eNB.

A base station is connected to the packet core network by means of the S1 interface. This interface usually has two components: the S1-U carries traffic for the SGWs, whereas the S1-MME interface carries signaling messages for Mobility Management Entities (MMEs).

While the S1 interface uses exclusively GTP as transport protocol, S5/S8 interfaces opens the possibility of using some alternative, like Proxy Mobile IPv6 (PMIPv6) [10]. Both GTP and PMIPv6 are network-based mobility management solutions because the functionality to handle the movement of mobile terminals resides in the network, and the terminals only perform the standard IP operations. The selection of GTP or PMIPv6 on S5/S8 interface has no impact from the UE point of view.

In PMIPv6, mobility support is provided by some specific network entities, namely Mobile Access Gateway (MAG) and Local Mobility Anchor (LMA). The MAG takes care of the mobility signaling on behalf of the MNs attached to its links, tracking the mobile nodes as they move, while the LMA stores all the routing information needed to reach the MNs in the PMIPv6 domain by associating each mobile node with the MAG that the MN is using. A tunnel between the LMA and the MAG allows the transfer of traffic from and to the MN. Using PMIPv6, the MN can move across a PMIPv6 domain changing its access link, while keeping its IP address. In EPC architecture the SGW incorporates the MAG functionality and the PGW plays the role of LMA [10].

The S5/S8 interface identifies the logical connection between SGW and PGW in the EPC, as shown in Figure 3. The S5 interface is present in a real deployment when the SGW and the PGW are deployed separately in the operator network. Additionally, there are scenarios where S5 have to be implemented even when consolidated SGW and PGW nodes are deployed. For instance, this would be the case of a mobile terminal maintaining two open IP sessions through distinct Access Point Names (APNs) that are not configured in the same PGWs.

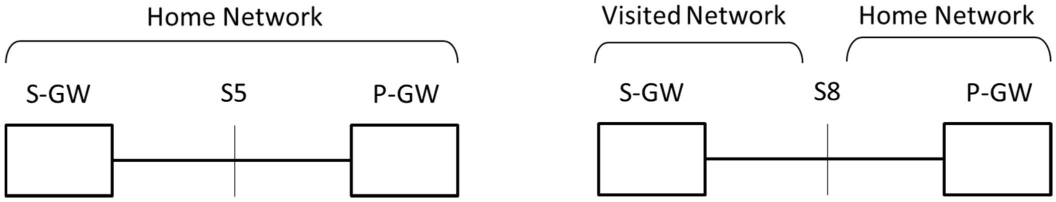


Figure 3. LTE Roaming Interface

The S8 interface is always required for roaming scenarios. In that situation, the S8 interface connects SGW and PGW nodes from different administrative domains, allowing for the users to roam from a home to a visited network. When a terminal is roaming in a visited network the connectivity to external networks (e.g., the Internet) is gained through its home network [11].

There are two potential alternatives for S5/S8 interface: GTP- and PMIPv6-based deployments. Each of these alternatives has its own specific mechanisms for controlling the bearers and for delivering the traffic between the SGW and the PGW.

There is a one-to-one mapping between S1 and S5/S8 bearers. In the case of using GTP each bearer is extended between the terminal and the PGW. On the contrary, when PMIPv6 is in place, the bearers are defined only between the mobile terminal and the SGW. All the packet flows are later on carried over a single IP-in-IP tunnel between SGW and PGW.

2.2 New Mobility Management Proposals in IETF

In centralized mobility management schemes the address of the UE is anchored at home network, so traffic is required to always traverse the central anchor. This indicated that paths between the UEs and its communicating peers would be unnecessarily prolonged. In addition, centralized mobility management imposes scalability challenges because the centralized anchor requires having enough processing capabilities to be able to handle all UEs' traffic.

The IETF is working on mobility management protocols that can benefit from the introduction of SDN in these environments. Apart from the previously referred PMIPv6 protocol, the Distributed Mobility Management (DMM) group is modifying existing approaches to support a distributed anchoring model. In network-based mobility management, there are two kinds of solutions: one is fully distributed and the other is partially distributed. In the fully distributed approach, each access router acts as both a mobile access gateway (MAG) and a local mobility anchor (LMA). In partially distributed approach, the data and control planes are separated. But only data plane is distributed. The partially distributed approach is similar to the mobility management scheme used in 3GPP.

3 Software-Defined Mobility Management

The generic application of the SDN paradigm to wireless and mobile networks has been described in [12],[13],[14]. Thanks to the programmable capabilities offered by SDN, the control of the mobile network might become much more flexible and efficient than the conventional mobile networks today. Specifically, the usage of SDN in this area is expected to simplify the mobility management procedures at both control and user plane.

Some of the present challenges of mobile networks in general [14] and MM protocols [6] in particular can be benefitted by the logically centralized control approach provided by SDN. One of them has been already mentioned, and it is the scalability of the control plane itself. By decoupling the control plane from the forwarding element the signaling workload can be offloaded from the forwarding devices as implemented today by consolidating the control in a (logically centralized) single entity. This entity can be easily upgraded in terms of computing resources to support the ever increasing demand of signaling traffic in 5G networks. For instance, the CPU load increase shown in Fig. 1 could be alleviated by adding computation recources (even distributed across the network) to the processing of the signaling related to the attachment of new users to the network. This facility provides better ways of dimensioning core network elements by adjusting the resources to the real demand in a flexible way instead of adding new complete full boxes for preventing CPU saturation.

On the other hand, regarding the data plane, the SDN approach can improve the operation of the network by simplifying the traffic delivery methods directing the traffic flows to and from the mobile terminal as result of the forwarding rules in the network nodes which receive instructions from the SDN controller which maintains a complete view of the communication end-to-end (from the terminal to the gateway, and even beyond).

For the different mobility management methods described before it can be possible to explore common procedures deployed internally in the SDN controller to be applied for all the cases, facilitating the interworking of them, and complemented by the specific functional modules per method. Furthermore, it can be completed by moving the remaining particularities to the edge in the case of the user plane aspects (e.g., frame formats).

The centralization of the control functions permit to maintain a centric knowledge at the control plane regard to the mapping of identifier and locator of the IP address assigned to the UEs. As mentioned later in the article, this fact renders flexibility for efficiently routing the traffic to and from the terminal during handover, allowing for optimizations from the network usage point of view.

Finally, since the control plane is not more tightly bound to the data plane, it becomes easier and more cost efficient to distribute the points of management to avoid relying on centrally deployed anchors for the management of mobility at the data plane. This distributed schema can be achieved, for instance, by means of a hierarchical deployment of SDN controllers.

3.1 Review of existing proposals

In the literature a number of proposals can be found devising the usage of software defined methods for mobile networks, including the MM procedures.

SDN is proposed in [15] as the mean for processing the tunneling procedures of the mobility management protocols. The mobility management function is considered to be an external application accessing an SDN controller via an API. The movement of the terminal, as detected by the mobility related entities (e.g., MAG in PMIPv6) generates some update in the registers that identify the new point of attachment of the terminal. Those registers are kept by the mobility management application external to the controller. Immediately, this controller acts on the forwarding plane by updating the forwarding rules of the underlying nodes according to the terminal mobility. In [16] the authors propose a general mobility management framework where an SDN controller combines in an integral manner a number of functions related to the mobility management, such as tunnel handling or mobility management decision.

In [17] the control plane functions of MME, SGW and PGW are moved and enclosed to a central controller while maintaining the data plane unaltered but running in simplistic forwarding elements. The controller yet offers the standard external interfaces for easy integration with legacy systems. A similar approach is taken in [18] but limiting the centralization of control functionalities to the ones residing in MME and SGW, interacting with an OpenFlow controller via APIs.

The MobileFlow architecture described in [19] supports the mobility management function also on top of a controller. There new network elements are implemented for handling the data plane, called MobileFlow Forwarding Engines, being directly instructed by the MobileFlow Controller, which communicates with different applications such as the mobility management one. The forwarding engines implement more sophisticated functionality that the one in a mere OpenFlow-based switch (such as layer-3 tunneling), yet keeping some control plane capabilities on it. These engines, however, are connected through a simple OpenFlow-enabled switch network, where the traffic forwarding is controlled by a separated OpenFlow controller.

The work in [20] presents two different controller models (single controller and hierarchical controller models) where the SDN approach is integrated with conventional mobility control plane, resulting in a partially-separated SDN architecture. In this way, the SDN-based control complements and extends conventional control plane facilitating advanced features.

Reference [21] analyses the handover process in an SDN-based environment and proposes some algorithms to improve the handover process in terms of optimal forwarding, minimization of flow entry population for supporting the handover. The results show promising performance of SDN-based solutions. Similarly [22] and [23] present two alternative analysis of an OpenFlow-based PMIPv6 operation, including analytical formulation of the handover latency improvement.

3.2 Software-Defined Mobility Management in the scope of LTE

This section presents an architectural approach based on the work being carried out by the Wireless and Mobile Working Group (WMWG) of the ONF [4] for introducing SDN capabilities in the core nodes of mobile networks. The main target is to define extensions to the OpenFlow protocol for accomplishing S5/S8 functionality without introducing changes to the rest of the standard 3GPP defined interfaces, ensuring compatibility with deployed infrastructure and investment protection.

The EPC natively separates control plane and user plane protocols and procedures in current 3GPP standards for both GTP- and PMIPv6-based deployments. The idea proposed in this article is to implement the mobility management procedures as an application on top of an SDN controller, separating such function from the current SGW and PGW, keeping the existing devices to merely forwarding traffic and framing. The MM procedures are then located on a centralized entity, in a Mobility Management Control (MMC) functional block, in charge of managing the terminal mobility in the network. The Figure 4 schematically shows the idea behind this case. Additional control procedures could be deployed on top of the controller, which are out of the scope of this paper. Those other functions will be more or less integrated with the MMC to accomplish more complex control functions in an orchestrated manner. Some potential scenarios for that integration are presented at the end of the paper.

The controller is expected to use a standard protocol as southbound interface for populating the forwarding tables aligned with the mobility management process. This centralized control approach ensures consistency of the rules to be configured in each device due to the fact of maintaining an end-to-end visibility of the resources.

In more sophisticated scenarios, the controller could instruct the traffic forwarding devices according to the execution of the some algorithms containing more complex service delivery logic, enriching the mobility management process. The same southbound interface would be used for this task.

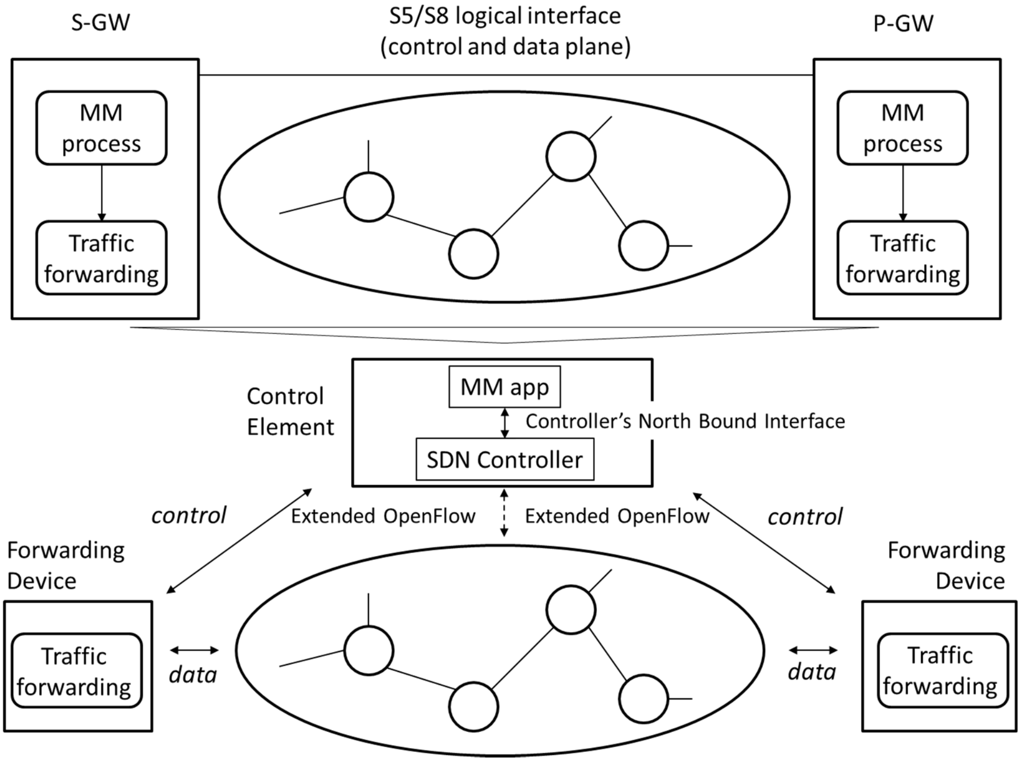


Figure 4. SDN approach on S5/S8 interface

An additional complexity faced by network operators is the possibility of interfacing with other operator's networks that could have implemented a distinct technology on the S8 interface. This could be the case of an operator implementing a GTP-based solution willing to connect to other operator implementing a PMIPv6-based solution. The interworking between PMIPv6 and GTP is not supported by current specifications. Also in this scenario the SDN approach is promising as more differences can be hidden in the black-box idealization of an SDN controller with some adaptors to the specifics of each mechanism.

In this case two different cases can be in place. In the first case, one network supports MM procedures on legacy monolithic elements while the other implements such capability in an SDN controller. In this situation, the SDN controller can mediate with the legacy device by adapting the signaling to the corresponding protocol implemented by the other network, either GTP or PMIPv6.

In the second case both networks follow an SDN approach for their corresponding MM procedures. Consequently a SDN controller interconnection is needed to accomplish the mobility management of the roaming terminal. One option for that interconnection could be the usage of existing standard protocols as defined today for S8 interface. If this is the case, the controllers then are not conscious about the SDN capabilities of its counterpart, then, as before, the SDN controller adapts the signaling to one of the S8 interface protocols in a predefined manner.

On the other hand, if these MM capabilities can be signaled among the controllers, the other option for the interconnection is to have a tailored and more simplistic inter-controller communication for interchanging the mobility management information, not being restricted to the GTP or PMIPv6 syntax and logic.

By running the MM application on generic computing resources the scale of the system increase and the production costs decrease.

4 Experimental evaluation

In order to support experimentally our Software-Defined Mobility Management architecture, as well as the feasibility of the example use cases described in Section 3, we deployed an SDN test-bed based on off-the-shelf hardware running GNU/Linux operating system.

4.1 Test-bed

The test-bed comprehends 14 switches and a network controller interconnected through Ethernet (IEEE 802.3). Moreover, 1 switch exposes IP gateway capabilities and 6 switches offer IEEE 802.11b/g connectivity to the UEs. As our SDN architecture does not devise any intervention on the UE, its hardware and software requirements are simply an IEEE 802.11b/g interface, and a standard IP stack. Employing IEEE 802.3 and 802.11 as link layer technologies does not affect the implementation nor the evaluation of our architecture since the Mobility Management module (MM) works on an abstract network view according to ONF guidelines [24]. Clearly, the adoption of a different link layer technology may require some changes in the controller module in charge of building the network model. All the switches run Open vSwitch 2.3.0 which provides an OpenFlow 1.3 interface. The connection between Open vSwitch and the SDN controller is performed out-of-band through standard OpenFlow mechanisms.

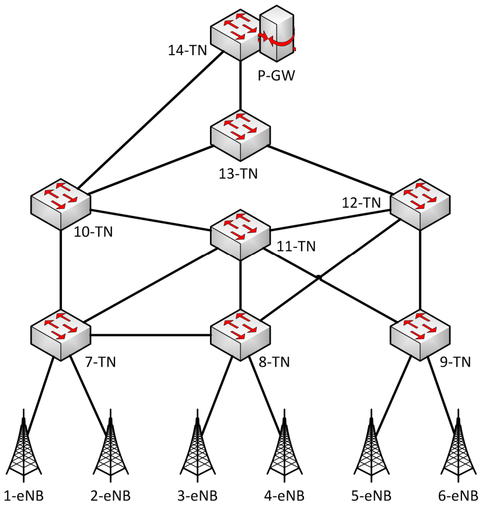


Figure 5. SDN test-bed topology

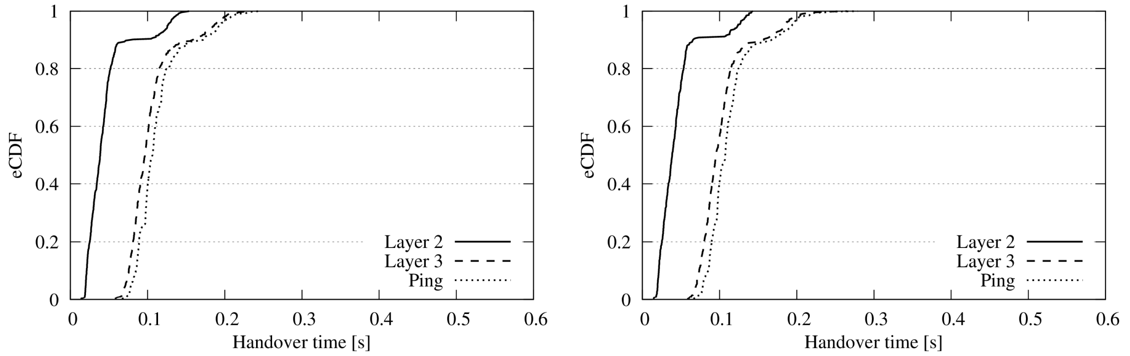
With the purpose of providing a reactive framework, we adopt an event-driven communication paradigm. This paradigm increases responsiveness compared to asynchronous communication, being this aspect fundamental in an environment like a mobile network. In addition, an event-driven architecture can be used as a complement of a service-oriented architecture because services can be activated by triggers fired upon incoming events. The network controller runs Ryu as component-based SDN framework whose APIs natively support such paradigm allowing a simpler prototyping of our architecture. The whole architecture has been implemented using Python and a single network controller is employed.

4.2 Evaluation

With the purpose of evaluating the performance of our implementation, we configured the network topology shown in Figure 5 on our test-bed. We focus on the handover delay (i.e., the time an UE does not have connectivity as a result of a change of point of attachment), by analyzing the components that affect more the overall latency. For this analysis, a node external to the test-bed generates *ping* traffic destined to the UE every *1ms*. The handover delay analysis can be analyzed looking at 3 different aspects: (*i*) the Layer 2 handover: time elapsed since the old radio link is torn down until the new one is established; (*ii*) the Layer 3 configuration: time required by the UE to obtain network layer connectivity (including the Layer 2 handover); and (*iii*) the IP flow recovery: time interval during which an IP flow is interrupted due to the handover (including both the Link 2 and the Layer 3 configuration). The main component of Layer 3 configuration time is given by the processing at the network controller. With the aim to evaluate the scalability of the network controller, 100 Simultaneous Attached Users perform a handover following a Poisson process with different rate λ. It is worth noting that the SAUs are emulated UEs. Therefore, the Layer 2 handover time of the physical UE is not affected by any other UE contending the same radio channel.

(a) User arrival rate λ = 6

(b) User arrival rate λ = 12



(c) User arrival rate λ = 30

(d) User arrival rate λ = 60

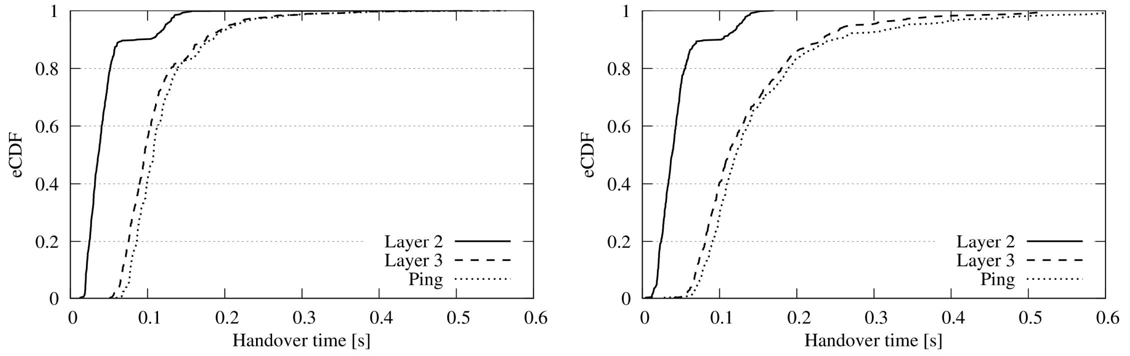


Figure 6. UE handover time eCDF for different handover rate λ at network controller

	λ=6			λ=12			λ=30			λ=60		
	L2	L3	Ping	L2	L3	Ping	L2	L3	Ping	L2	L3	Ping
Mean	0.045	0.104	0.113	0.045	0.104	0.113	0.45	0.110	0.120	0.45	0.136	0.151
Std. dev.	0.029	0.032	0.032	0.29	0.032	0.032	0.029	0.054	0.054	0.029	0.075	0.094
95% pctl	0.125	0.184	0.193	0.125	0.186	0.193	0.125	0.206	0.215	0.125	0.264	0.336

Table 1. UE handover time results (in seconds) for different handover rate λ at network controller

Figure 6 depicts the experimental Cumulative Distribution Function (eCDF) of the handover time for different λ as experienced by the UE, while Table 1 reports the numerical values of obtained results with the most significant statistical properties. As it can be noticed, the Layer 2 is independent of λ, on the contrary the time required to obtain Layer 3 configuration increases for higher λ values. This is due to the higher number of operations run in the network controller because of greater handover rate. Additional tests were performed with the purpose of identifying the main components of the increased processed time as explained in the following.

The network controller relies on a multi-thread implementation in order to take advantage of modern multi-core CPUs whereas each thread accomplishes a specific task. In particular, one thread manages the connection with the MME, a second thread is in charge to select the best P-GW for the UE upon an handover, while a third thread takes care of communicating the IP parameters to the UE (i.e. sending a Router Advirtisement for IPv6 connections). Two additional threads are in charge of computing and configuring the paths within the network.

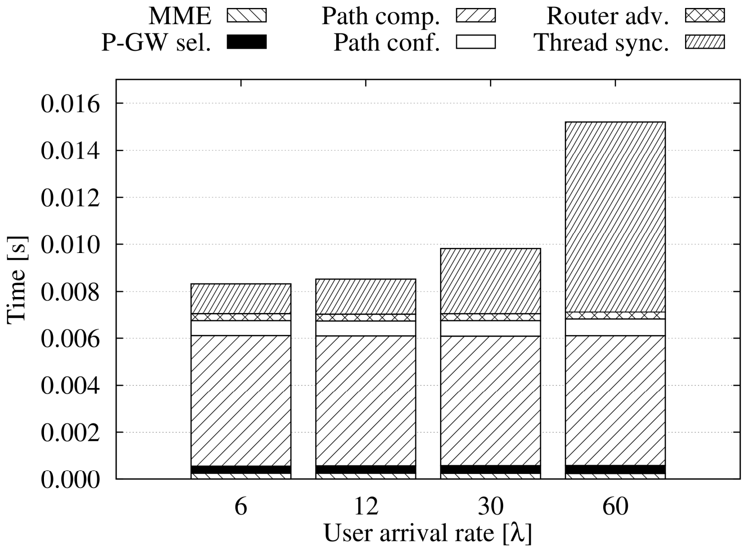


Figure 7. Processing time for different handover rate λ at network controller

Figure 7 depicts the processing and synchronization time for each thread for different λ. As it can be seen, the processing time is indepent of the rate λ. This is due to the fact that, according to our implementation, the operations required for managing a single handover do not depend on the numbers of handovers occurring simultaneously. On the contrary, the synchronization time increases with λ. A higher synchronization time means that the message queue system is getting more and more congested. Two potential causes of such behavior are the following: (i) the message queue system does not scale adequatly or, (ii) the processing time is greater than the handover rate. Our tests showed a CPU occupation growth according to λ, in particular, the CPU was 100% busy for λ=60. This means that the network controller is always backlogged and it is not able to serve all the handover requests in time. Indeed, this occurs when the processing is not fast enough compared to the arrival rate leading to an increase in the overall handover time experienced by the UEs. In order to leverage the problem, the network controller can be easily extended in a distributed fashion following the nowadays mature techinques for high-performance distributed systems [25].

5 Future Directions

The proposed Software Defined MM solution for 5G wireless networks considers a smooth integration with legacy mobile network core by offloading the signaling procedures towards SDN-oriented platforms. However, the SDN-based solution can be extended to deal with some other issues present today in current networks. Several aspects can be further investigated, with some future directions are provided here.

- Signaling and data plane simplification: as described in [2] (and additional references on it) the conventional tunneling protocols as GTP impose a severe load and burden to existing core control nodes as well as data plane. As an incremental step regarding the signaling offloading described above, new ways of handling the mobility of the terminals can be studied for simplifying the tracking of the users. The S1 interface makes use of GTP as basic transport tunneling protocol, then a simplification of the S1 interface signaling procedure can consider the usage of a more efficient traffic forwarding mechanisms instead of GTP tunneling. One of possible solution is to decouple the dual functionality of the IP address as both identifier and locator. The concept of SDN is an enabler for this solution because its capability to define rules for different header fields in the packet at the same time when processing packets handling individual flows. Acting on different fileds to implement the forwarding of the packets permits more sophisticated behaviour. The overhead from the dataplane component of GTP (GTP-U) could be even removed. On the other hand, the contro messages of GTP (GTP-C) could be directed and terminated directly on the SDN-based control elements.
- SGW/PGW Traffic engineering in the scope of MM: once the mobile node sets up a tunnel to an APN in a PGW, the session is hooked to such PGW during the session lifetime. However this can lead to an inefficient usage of network resources as the terminal moves because of the path needed to reach the PGW from the changing point of attachment. A centralized controller can manage a complete view of the network topology, and then, optimization algorithms can be run to improve the usage of network resources by re-routing certain traffic or even by redirecting the tunnel termination to PGWs closer to the terminal. This has further implications such as the transfer of the user states between core nodes. The SDN approach can make this possible since all the user information is managed from a central controller. Similarly, in DMM scenarios optimization on anchor selection can be implemented by associating flows to closest mobility entities as the mobile terminal is moving.
- Full service composition and elasticity: new trends in network communications, such as Network Functions Virtualization (NFV) [26], will change the way of providing services to the mobile end users. Basically the NFV approach facilitates the instantiation of network functions in the network on generic computing facilities in such a way that these functions can be easily scale in and out according to network needs. Functions like SGW or PGW lay on the scope of NFV. This elasticity in the way of creating network functions across the network (mainly in data centers spread in different parts of the access and backbone) requires from agile and automated control functionalities to reconcile the traffic demand with the changing service topology. In this circumstance, SDN-capable mobility management mechanisms can integrate with NFV control mechanisms (potentially SDN oriented, as well) to compose the service and to connect the end users to the required functions in an efficient way.

6 Conclusions

SDN is changing the way of designing services and, in consequence, the way of operating the networks. The new 5G wireless networks are expected to manage applications with very different requirements in terms of traffic, ranging from HD video (characterized by a very high rate) to Internet-of-Things traffic (with low rate and not very demanding in terms of latency but high scalability). While this impact on the data plane, a common characteristic for all of the uses in 5G is the ubiquity and the almost permanent connection to the mobile network to get access to the content repositories. This really imposes a challenge in the signaling procedures provided to get track of the user and to guarantee session continuity. The mobility management mechanisms will play a central role in the 5G networks because of the always-on demand. SDN-based approaches can alleviate the MM operation and allow further improvements to the whole network control by (i) more computational capabilities can be at disposal of the MM protocols without the constraint of the monolithic box approach nowadays, and (ii) integrating and orchestrate the mobility management procedures with other control mechanisms in the network for a more efficient service provision and a better usage of the network resources. An architecture proposal is presented for co-existence with current 3GPP LTE deployments.

7 References

[1] Nokia Siemens Networks, “Signaling is growing 50% faster than data traffic”, white paper available at http://www.nokiasiemensnetworks.com/sites/default/files/document/signaling_whitepaper_online_version_final.pdf, 2012.

[2] F. Metzger, A. Rafetseder, P. Romirer-Maierhofer, K. Tutschku, “Exploratory Analysis of a GGSN’s PDP Context Signaling Load”, Hindawi Journal of Computer Networks and Communications, vol. 2014, Article ID 526231, 13 pages, 2014. doi:10.1155/2014/526231.

[3] Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks”, white paper available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, 2012.

[4] M.R. Sama, L.M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, H. Ni, “Software-Defined Control of Virtualized Mobile Packet Core”, in IEEE Communications Magazine, Vol. 53, No. 2, pp. 107-115, 2015.

[5] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, A. V. Vasilakos. “Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey”, in ACM/Springer Mobile Networks and Applications, vol. 20, no. 1, pp. 4–18, 2015.

[6] I. Al-Surmi, M.Othman, B.M. Ali, “Mobility Management for IP-based Next Generation Mobile Networks: Review, Challenge and Perspective”, Journal of Network and Computer Applications vol. 35, pp. 295-315, 2012.

[7] R. Bolla, M. Repetto, “A Comprehensive Tutorial for Mobility Management in Data Networks”, IEEE Communications Surveys and Tutorials, vol. 16, No. 2, pp. 812-833, Second Quarter, 2014.

[8] 3GPP TS 23.401, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”, V13.3.0, June, 2015.

[9] 3GPP TS 29.060, “GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface”, V13.1.0, June, 2015.

[10] 3GPP TS 29.275, “Proxy Mobile IPv6 (PMIPv6) based Mobility and Tunnelling protocols”, V13.2.0, March, 2015.

[11] GSMA Association, “LTE and EPC Roaming Guidelines”, version 10.0, July 2013.

[12] L.E. Li, Z.M. Mao, J. Rexford, “Toward Software-Defined Cellular Networks”, Proceedings EWSDN, 2012.

[13] C.J. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L.M. Contreras, H. Jin, J.C. Zúñiga, "An architecture for software defined wireless networking," IEEE Wireless Communications, vol.21, no.3, pp.52,61, June 2014.

[14] C. Chaudet, Y. Haddad, "Wireless Software Defined Networks: Challenges and opportunities," Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on , vol., no., pp.1,5, 21-23 Oct. 2013.

[15] D. Liu, H. Deng, “Mobility Support in Software Defined Networking”, draft-liu-sdn-mobility-00 (work in progress), January 2014.

[16] X. Wei, J. Yin, D. Ni, K. Xue, S. Chen, “Software Defined Network based General Mobility Management Framework”, draft-wei-sdnrg-framework-mobility-sdn-00 (work in progress), July 2014.

[17] G. Hampel, M. Steiner, T. Bu, “Applying Software-Defined Networking to the Telecom Domain”, Proceedings of the 16th IEEE Global Internet Symposium in conjunction with IEEE Infocom, Turin, Italy, April 2013.

[18] S. B. H. Said, M. R.Sama, K.Guillouard, L.Suciu, G. Simon, X. Lagrange, J.-M. Bonnin, "New Control Plane in 3GPP LTE/EPC Architecturefor On-Demand Connectivity Service", Proceedings of the 2nd IEEE International Conference on Cloud Networking (CloudNet), 2013.

[19] K. Pentikousis, Y. Wang, W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks", IEEE Communications Magazine, pp. 44-53, July 2013.

[20] S. Jeon, C. Guimaraes, R.L. Aguiar, "SDN-Based Mobile Networking for Cellular Operators", Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture (MobiArch), September 2014.

[21] Y. Wang, J. Bi, "A Solution for IP Mobility Support in Software Defined Networks", in Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN), 2014.

[22] S.M. Raza, D.S. Kim, H. Choo, "Leveraging PMIPv6 with SDN" in Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication (ICUIMC), 2014.

[23] S.-M. Kim, H.-Y. Choi, P.-W. Park, S.-G. Min, Y.-H. Han, "OpenFlow-based Proxy Mobile IPv6 over Software Defined Network (SDN)", in Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), 2014.

[24] Open Networking Foundation, "SDN Architecture Overview", Version 1.1, ONF TR-504, November, 2014, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf

[25] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair, "Distributed Systems: Concepts and Design (5th Edition)", Addison-Wesley, 2011, ISBN 0-132-14301-1.

[26] "Network Functions Virtualisation – An Introduction, Benefits, Enablers, Challenges, and Call for Action", white paper available at http://portal.etsi.org/NFV/NFV_White_Paper.pdf, 2012