State Space Analysis to Refactor the Mobile Core

Heikki Lindholm[⋆], Lirim Osmani[⋆], Hannu Flinck[†], Sasu Tarkoma[⋆], Ashwin Rao[⋆]
[⋆]University of Helsinki, [†]Nokia Networks

ABSTRACT

A state space analysis of the variables used by the network functions that serve the mobile core provides insights on the multiple roles portrayed by each network function. These insights can be leveraged to modularize and subsequently refactor the network functions that are expected to serve the fifth generation mobile networks (5G). In this paper, we group the variables according to their usage in each network function, then we identify the producers and consumers of these variable groups during the various events triggered to serve the mobile devices. We leverage on the producers and consumers of state changes to propose a publish-subscribe control plane in which the producers and consumers of state changes are the publishers and subscribers respectively.

1. INTRODUCTION

With the advent of the fifth generation cellular networks, mobile operators and equipment manufacturers are actively evaluating technologies to restructure the mobile core. Two technologies that offer a plethora of opportunities to restructure the core are Network Function Virtualization (NFV) and Software Defined Networks (SDNs). NFVs open opportunities to abstract and subsequently modularize the network functions, while SDNs offer solutions to efficiently configure and manage the connectivity between these modules.

One of the first steps towards modularizing the network functions is to separate the control plane from the data plane [2, 8]. The primary benefit of this separation is that the control plane components of the various network functions can coalesce into a logically centralized and physically distributed controller. This logically centralized controller not only serves the control messages but also programs the appropriate forwarding rules in the forwarding elements serving the data plane. To reap the benefits of a logically centralized controller, the modules of this controller must be placed at the appropriate physical locations in the network. This problem of physical distribution of the modules of the logically centralized control plane essentially boils down to the refactoring the control plane of the mobile core.

In this paper, we exemplify the refactoring of the control plane with the help of state space analysis. We begin our state space analysis by identifying the various roles of the network functions that serve the control plane of Long-Term Evolution (LTE) networks. For each role, we identify the relevant variables. We then group the variables according to the roles in which they are used, and leverage this grouping to refactor the control plane of the mobile core.

The key contributions of this paper are as follows:

1) We group the variables used by the network functions according to the roles in which they are used, and we use this grouping to abstract the various roles portrayed by the network functions. This abstraction, presented in §3.1, is essential to modularize the network functions and subsequently refactor the control plane.

2) We detail the usage of these variable groups during four frequently occurring events: a) the mobile device initiates a new session, b) the mobile device goes to sleep and becomes idle, c) the mobile device awakens and becomes active, and d) the mobile device moves from one location to another. We use this exercise, presented in §3.2, to identify not only the producers and consumers of the variable state changes, but also the network functions that are heavily used during these events.

3) We argue that identifying the producers and consumers of the state changes is useful to refactor the control plane of the mobile core. For example, we argue that it is beneficial to move the modules in the mobile core that are related to the device idle and device wake (scenarios (b) and (c) mentioned previously) to the radio network. In §4, we present a publish-subscribe control plane in which the producers and consumers of state changes are publishers and subscribers respectively.

AllThingsCellular '15, August 17, 2015, London, United Kingdom

DOI: http://dx.doi.org/10.1145/2785971.2785977

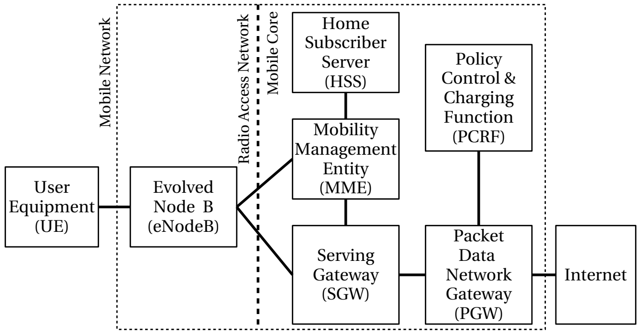

Figure 1: **Key network functions in LTE networks.** *The LTE network consists of the mobile core and the radio access network.*

The optimal placement of network functions can be addressed using different approaches. Basta *et al.* [4], provide a model to analyze the various scenarios in which the existing network functions can be placed. However, their model does not consider a logically centralized controller which includes the network functions other than the gateways that serve the mobile core. Moradi *et al.* [9], present a hierarchical control plane to distribute the control plane network functions. We build on their insights to refactor the modules in this logically centralized control plane and study the placement of individual control plane modules within this hierarchy. The objective of this work is to not propose any new set of functionalities, but to explore scenarios in which the existing network functions can benefit from becoming part of a logically centralized controller.

2. BACKGROUND AND MOTIVATION

We now present an overview of the key network functions in LTE networks and use this high-level view to motivate the need to refactor these network functions. We limit ourselves to the network functions in LTE networks because they are expected to be the cornerstone for future mobile networks.

As shown in Figure 1, the LTE mobile network consists of the radio access network and the mobile core. The radio network connects the User Equipment (UE) to the mobile core, which, in turn, connects the UE to external networks, such as the Internet. The radio access network and the mobile core contain the following primary network functions which ensure that the IP connectivity to the UE is provisioned, and the allocated resources comply to the subscription policy.

1) The *Evolved Node B (eNodeB)* is the bridge between the UE and the mobile core. The eNodeB can have multiple radio interfaces (cells), which it manages, and it has both control and user plane interfaces to the mobile core, thus serving both control and user plane traffic of the UE. The eNodeBs are also interconnected in clusters. The eNodeB or the cell serving the UE changes depending on the mobility of the user.

2) The *Mobility Management Entity (MME)* performs tasks related to mobility management, which includes authentication, authorization, and management of sessions initiated by the UE. The MME serves only the control plane messages.

3) The *Home Subscriber Server (HSS)* in its simplest form represents a database which stores the user and subscription information. Like the MME, the HSS only serves the control plane messages related to the UEs.

4) The *Service Gateway (SGW)* is the mobility anchor when the UE is on the move. The SGW also routes and forwards the UE data traffic to the selected Packet Data Network Gateway (PGW). The SGW thus serves the control plane messages and the data plane packets exchanged by the UE.

5) The *Packet Data Network Gateway (PGW)* is the gateway to external networks. Like the SGW, it serves the control plane by enforcing the policy and charging, and also the data plane by forwarding IP traffic.

6) The *Policy Control and Charging Rules Function (PCRF)* provides details on how UE sessions are treated by the PGW. Additionally, it may also contain information such as the IP addresses assigned to the UE.

A limitation of this architecture is that it relies on network functions that serve the control plane and also the data plane (the eNodeB, SGW, and PGW). Previous research works highlight this brittleness and detail the benefits of splitting the control plane and the data plane and using a logically centralized controller to manage the mobile core [3, 7, 8]. We argue that this logically centralized controller can be refactored to ensure that the key components of this controller are physically located where they are needed the most.

3. USER STATE IN NETWORK FUNCTIONS

When the User Equipment (UE) changes its status—for example, when the UE is powered on or off, or the UE goes to sleep or awakes, or the UE moves from one location to another—the UE and the network functions in the LTE network go through series of actions called *procedures*. Each UE session can be considered as a collection of the state which is stored in the UE and the network functions that serve the UE. Some of the state is persistent and provisioned at subscription time, some exists only for the duration of the UE session, and some of the non-provisioned state is cached for some time after the session ends for the next near future use. Some state variables are only used by a single entity, some a communicating pair of network functions, and others are duplicated in several network functions. Each procedure consists of a specified set of actions taken by the UE and the network functions, and each action accesses some subset of the network functions' state.

In this section, we present a state space analysis of the network functions. This state space analysis serves as a basis for refactoring the network functions. We begin our analysis by first grouping the variables into more general categories and then use the producer-consumer model to illustrate the roles the network functions take when accessing these variable groups.

3.1 Grouping of State Variables

We leverage on the functional description of the state variables used in the 3GPP specification [1] and manually categorize these variables into the following representative groups. We refer the reader to the 3GPP specification for the expansions of the abbreviations used.

1) **Device identification (Id)**. This group contains variables whose values are part of the primary keys in device or subscription lookups, such as the persistent subscriber identifier, IMSI, and GUTI (a temporary identifier used instead of IMSI over the radio link).

2) **Location.** This group contains variables that store the cell-level location information of the UE (ECGI) as well as the bounds of where to UE can go without reporting its location back to the network (TAI, Tracking Area List, *etc.*).

3) **Radio session state (Radio state)**. This group contains variables that maintain the radio link related information which is mainly used between the UE and the eNodeB, including the radio link security context (C-RNTI, DRB, AS security context, *etc.*).

4) **Control plane connection state (Control plane state)**. This group contains variables that maintain the identifiers for the user-specific control plane connections (S11/S5/S8/etc IPs and TEIDs) as well as the security context between the UE and the core network (NAS security context).

5) **User plane connection state (User plane state)**. This group is composed of bearer (a logical connection and related attributes) information (including the UE IP address, APN, per bearer tunnel endpoints (S1u/S5u IPs/TEIDs) in the core network, and per bearer QoS and policy parameters, such as GBR, MBR, and TFT) and global and per APN QoS and policy parameters (default bearer, APN-AMBR).

We would like to point out that we grouped the security context information between the UE and the radio access network under the Radio state category and the security context between the UE and the mobile core network under the Control plane state category. We group the security related information in this way because there is a natural split into radio access network and mobile core network sides.

Table 1 lists the components that create or access state in each of the groups after the UE has attached itself to the mobile network. We observe that the state from three groups—Location, Control Plane state, and User Plane state—are spread throughout the components. In the following, we select these three groups for a closer look.

3.2 Procedures Accessing State Information

The variable groups presented in Section 3.1 are updated during procedures to serve events such as UE mobility. We restrict our analysis to four frequently occurring procedures: UE attach, UE idle, UE awake, and UE mobility. For each procedure, we analyze the *producer-consumer* relationship of each of the network functions in relation to each group of state. For our analysis, a network function that creates or maintains a state variable and sends messages containing updated values of that variable is the *producer* of that state variable, while a network function that receives the messages and reads the updated value of the variable is the *consumer* of that state variable. This producer-consumer model serves as the basis for the publish-subscribe model that we present in the next section.

Network Function	**Id**	**Location**	**Radio State**	**Control-plane State**	**User-plane State**
UE	✓	✓	✓	-	✓
eNodeB	✓	✓	✓	✓	✓
MME	✓	✓	-	✓	✓
SGW	✓	✓	-	✓	✓
PGW	✓	✓	-	✓	✓
HSS	✓	-	-	✓	✓
PCRF	✓	✓	-	-	✓

Table 1: **State in LTE components after the Initial Attach procedure.** *Apart from the Id, the location, control-plane state and user-plane state are maintained by most of the network functions in the mobile core.*

We use the producer-consumer model to see (1) whether producers could be moved close to the consumers, (2) whether certain functions are only producers or consumers of variables or certain groups are related to some components and (3) whether some network functions are dominant players in specific changes in user state. In Table 2, we list the producer-consumer relationships for the device location, control plane connection state and user plane connection state, respectively, in the following network procedures.

1) **Initial Attach**. The UE initiates the attach procedure by issuing an attach request message containing its identification information (IMSI). This identification information is consumed by all the network functions in the mobile core. On receiving the identification information, the eNodeB acts as a producer of the UE's location information, which the MME and HSS, and later SGW, PGW and PCRF consume. The MME updates this location information with the cells to which the UE can contact, and this information is consumed by the UE. Furthermore, the radio state information is produced and consumed by the eNodeB and the UE, while the control plane connection state variables and the user plane state variables are produced and consumed by all the network functions in the mobile core, except HSS for the user plane case.

2) **Idle (S1 release [1])**. On UE inactivity, the eNodeB releases its radio resources and updates the location information. These updates made by the eNodeB

Variable	Network Function	Attach		Idle		Wake-up		Mobility	
		C	P	C	P	C	P	C	P
Location	UE	✓	-	✓	-	✓	-	✓	-
	eNodeB	-	✓	-	✓	-	✓	-	✓
	MME	✓	✓	✓	-	✓	✓	✓	✓
	SGW	✓	-	✓	-	✓	-	✓	-
	PGW	✓	-	✓	-	✓	-	✓	-
	HSS	-	-	-	-	-	-	-	-
	PCRF	✓	-	✓	-	✓	-	✓	-
Control Plane State	UE	-	-	-	-	-	-	-	-
	eNodeB	✓	✓	-	✓	✓	✓	✓	✓
	MME	✓	✓	✓	✓	✓	✓	✓	✓
	SGW	✓	✓	✓	-	-	-	-	-
	PGW	✓	✓	-	-	-	-	-	-
	HSS	✓	-	-	-	-	-	-	-
	PCRF	-	-	-	-	-	-	-	-
User Plane State	UE	✓	-	-	-	-	-	-	-
	eNodeB	✓	✓	-	✓	✓	✓	✓	✓
	MME	✓	✓	✓	-	✓	✓	✓	✓
	SGW	✓	✓	✓	-	✓	✓	✓	✓
	PGW	✓	✓	-	-	-	-	-	-
	HSS	-	✓	-	-	-	-	-	-
	PCRF	✓	✓	-	-	-	-	-	-

Table 2: **The Consumers (C) and Producers (P) of the Location, Control Plane State, and the User Plane State.** *The MME is the consumer of the Location state produced by the eNodeB during the Idle event, and the MME is both, the producer and the consumer of the Control Plane state during the Idle event.*

are consumed by the MME, SGW, PGW, and PCRF. The eNodeB also updates the control plane connection state which is consumed by the MME and SGW; the MME subsequently updates this information which is then consumed by the SGW. Similarly, the user plane state is updated by the eNodeB and these updates are consumed by the MME and SGW.

3) **Wake-up (Service request [1])**. The radio connection with an idle UE is re-established when either the UE initiates the wake-up because it has some traffic to send, or the network initiates the wake-up because the UE has to receive some traffic. If the location of the UE has changed, the location specific information is updated by the eNodeB and is consumed by the UE, MME, SGW, PGW and the PCRF. The UE and the eNodeB both produce and consume each other's radio link related security information, while the MME and eNodeB produce and consume the subsequent updates in the control plane connection state (for example, updates in the S1AP tunnel end-point variables). The eNodeB, MME, and SGW produce and consume the user plane state information such as the user plane tunnel end-points.

4) **Mobility**. The mobility of UE contains multiple cases which includes mobility to a network managed by another operator. We limit our current analysis to scenarios in which the UE moves to different locations but uses the same network operator (i.e. non-roaming). This type of mobility contains the following two sub-cases: a) the eNodeBs that serve the UE during the handover are connected to each other, or b) the eNodeBs are not connected to each other. Both sub-cases may, or may not, require location processing from the MME. In each of the two cases, the eNodeBs and (possibly) the MME update the location information of the UE which is consumed by the MME, SGW, PGW, and the PCRF. The changes in the control plane state and user plane state (the tunnel end-points) which are made by the eNodeB are consumed by the MME and the SGW.

In Table 2, we observe that much of the activity in idle, wake-up and mobility procedures centers around the eNodeB(s), the MME and the SGW. The MME is usually both a producer and a consumer of information and therefore the related functions from the MME could be candidates for simplifying the chain of information, *i.e.*, moving the functions either to the eNodeB or the SGW. Next, we present a preliminary refactoring based on these insights.

4. MODULARIZING AND REFACTORING THE NETWORK FUNCTIONS

In this section, we discuss approaches to modularize and subsequently refactor the network functions. In particular, we present a publish-subscribe control plane in which we consider producers and consumers of state changes as publishers and subscribers, respectively.

4.1 Approach to Refactor Network Functions

In Table 2, we observe that each network function serves multiple events, and while serving the events, modifies variables that belong to specific variable groups. We use this observation to present two approaches to modularize the network functions.

1) *Each network function contains a module for each variable group.* In this approach, each variable group is managed by a module responsible for that variable group. For example, the SGW would have a module that manages the Control plane state, a module for the User plane state, *etc.*, and similar modules would exist in other network functions. In Table 2, we observe that multiple variable groups are modified during each event. Therefore, moving a module (one variable group) from one physical location to another would incur additional signaling to update the other variable groups. This approach would therefore be counterproductive in reducing the control signals exchanged, one of the key driving factors for the refactoring exercise.

2) *Each network function contains a module for every event.* In this approach, each event is managed by a module responsible for that event. For example, the MME would have a module that manages the attach event, a module for the *Idle* event, *etc.*, and similar modules would exist in other network functions. The advantage of this approach is that modules from different network functions can be merged and placed in

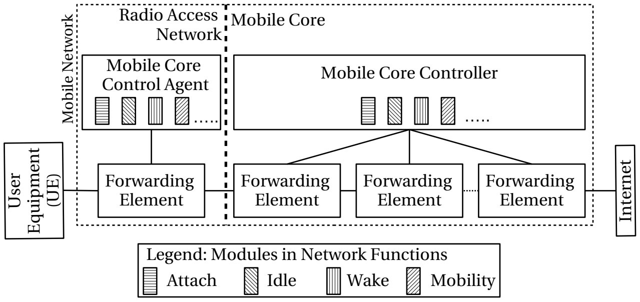

Figure 2: **The refactored Mobile Core with a publish-subscribe control plane.** *The logically centralized Mobile Core Controller publishes events on state changes within the Mobile Core. The Mobile Core Control Agent subscribes to these changes and publishes state changes in the Radio Access Network which are subscribed to by the Mobile Core Controller.*

the same physical location to reduce the control signals exchanged when serving the event. We now present an overview of the architecture based on this approach.

4.2 Publish-Subscribe Control Plane

One key requirement for the merging and subsequent refactoring of the modules is that the control plane (which serves control messages) be separated from the data (user) plane (which serves the data traffic). Previous works have shown that this separation is possible [2, 7, 8, 10]. A key benefit of separating the control and data planes is that the control messages can be served by a logically centralized controller which programs the forwarding elements that serve the data plane.

We envision that the logically centralized *Mobile Core Controller*, henceforth referred to as the *Controller*, is composed of modules for serving each event. In Figure 2, we show that this controller contains modules to serve the *Attach*, *Idle*, *Wake-up*, *Mobility*, and other events. The controller also programs the forwarding rules in the forwarding elements present in the mobile core. However, the events are not served solely by the controller present in the mobile core. Some of the events are served by an agent of this controller, the *Mobile Core Control Agent* which is located in the radio access network. This agent is a result of the merging of the modules from the eNodeB with some of the modules in various network functions of the Mobile Core.

The *Mobile Core Control Agent*, henceforth referred to as the *Agent*, serves as the end-point for the control messages originating from the UE, and a state proxy for the state information stored in the *Controller*. This ensures that the system requires no changes in the UE, and that some control messages can be served by the *Agent*. At a high level, the *Agent* is configured as a publisher for state changes related to the events associated with the change in UE state, while the *Controller* is configured as a subscriber to these state changes. Similarly, the *Controller* is configured as a publisher for policies governing the routing of the data traffic, while the *Agent* is configured as a publisher and subscriber to these policies. For example, the *Agent* caches and thus proxies the location information which includes the list of eNodeBs (or the other *Agents*) that can serve UE. As a consequence *Idle* and *Wake-up* events from the same UE which do not include mobility can be served by the *Agent*. However, the *Agent* cannot proxy all the information which is currently maintained in the current mobile core. For example, regulatory constraints may govern the physical location where user data such as subscription information is situated.

The publish-subscribe model requires signalling for subscriptions as well as the state updates. We are evaluating techniques to minimize this signalling overhead of the frequent events at the possible cost of slight increase in signalling for the less frequent events. As discussed below, we can perform subscriptions during the initial attach and then lessen the overhead for the more frequent *Idle* and *Wake-up* events.

4.3 Event Processing in the Refactored Core

During the *Attach* event, the *Agent* receives the control message from the UE and forwards it to the *Controller*. The *Agent* also explicitly subscribes to the UE related state information that can be proxied and is free from regulatory constraints, such as the location information. The *Controller* updates and subsequently publishes the state information of the UE which includes the Control Plane state, and the User Plane state, including security context. The *Agent* receives a copy of this state because it has subscribed to the changes.

During the *Idle* event, the *Agent* updates the state information and performs all the computation traditionally performed inside the mobile core. The *Controller* receives these updates because it has subscribed to these changes. For example, the Control plane and User plane state (deleted control plane eNodeB and downlink user plane tunnel end-points) and Location state (the UE's currently connected cell) are updated in the core. The *Controller* also reprograms the switches responsible for the data plane according to this UE status change.

During the *Wake-up* event, the *Agent* publishes that the UE is active again which is received by the *Controller*, which has subscribed to the change; or the *Controller* updates the *Agent* that there is new traffic for the UE. All of the *Agents* that the UE is allowed to connect to should be subscribed to updates on available traffic from the *Controller*. The *Controller* receives updates from the *Agent* about the changed Control plane and User plane state (new end-points) and also about the UE's Location. The *Controller* reprograms the data plane switches to re-establish the downstream to the UE.

During both the *Idle* and *Wake-up* events, routing tables in data plane switches need to be updated, and therefore at least one message between the RAN and the core is required. At the same time, much of the control plane signals of the traditional network are exchanged between the modules in the *Agent*.

During the *Mobility* event inside the same operator's network, the *Agents* in the neighboring eNodeBs and the *Controller* are subscribed to the UEs movements. If the eNodeBs between which the UE moves are in the same network, the *Agents* in the source and target eNodeBs can handle the processing and UE state information exchange between themselves first. The *Controller* is notified by the *Agent* in the target about the changes in Control plane, User plane, and Location state, and reprograms the data plane switches accordingly. If the eNodeBs are not directly connected, the *Agent* in the target eNodeB must subscribe to the user state first, commanded by the *Controller* and up-to-date state is then transferred from the *Controller*. The *Controller* also needs to setup temporary tunnels between the eNodeBs to transfer traffic when the UE is switching its radio link. Finally, the target *Agent* notifies the *Controller* about the changed Control plane, User plane, and Location state as in the first case.

5. DISCUSSION

In this paper, we first group the variables used by the network functions according to the roles in which they are used. We then treat the network functions as producers and consumers of variable groups during the attach, idle, wake-up, and mobility events related to the UE. This exercise allows us to abstract the actions performed by the network functions during the various procedures. In §3.2, we use this abstraction to isolate the network functions that are the primary producers and consumers of state changes during these events. For example, we observe that the eNodeB and the MME are the primary producers and consumers of state change information during the idle and wake-up events. In §4, we build on these insights and highlight the benefits of modularizing the network functions according to the events they serve. This approach to modularize network function allows the modules that serve the same event to be merged, thus making these modules available to be placed at appropriate physical locations. We show that by maintaining a state proxy in the radio access network, the computation performed during the idle and wake-up events can be offloaded to this state proxy. This state proxy performs these computations by subscribing to the state changes at the core and also by publishing the changes it makes in the radio access network to the core.

Our approach to address the problem of placement of network functions in the core is based on the initial insights from previous research works that advocate for a logically centralized control plane [3, 4, 7, 8, 9]. Kempf *et al.* [7], propose a hierarchical architecture for the cellular network, whereby a local controller managing the local switches communicates with a central controller. Moradi *et al.* [9] extend this approach and present a hierarchical architecture to reduce the path inflation and inter-region handovers. Furthermore, Jin *et al.* [6] have explored the benefits of moving some components of the controller to the base stations, while Gudipati *et al.* [5], present an SDN based centralized control plane architecture for radio access networks.

We are currently building a rudimentary simulator based on our abstractions, and we plan to simulate the behavior of the network functions during various events to evaluate the effectiveness of refactoring. Our producer-consumer model differs from request-reply model because in our model the requests are implicit and replies (publish events) can arrive at any time. Furthermore, timing of subscriptions is crucial to how the new signalling model will behave. We plan to evaluate these aspects of the producer-consumer model in our simulation framework. We also plan to extend this refactoring to the hierarchical control plane proposed by Moradi *et al.* [9]. We believe this exercise is crucial to fully quantify the impact of having a logically centralized controller and to optimally place the modules of this controller. Looking ahead, understanding network function placement and refactoring of the mobile core are important steps towards the design of the next generation (5G) network architecture.

6. REFERENCES

[1] 3GPP - The Mobile Broadband Standard. http://www.3gpp.org/.

[2] OpenEPC - Open Evolved Packet Core. www.openepc.net.

[3] Basta, A., Kellerer, W., Hoffmann, M., Hoffmann, K., and Schmidt, E.-D. A Virtual SDN-Enabled LTE EPC Architecture: A Case Study for S-/P-Gateways Functions. In *Proc. of IEEE SDN for Future Networks and Services* (2013).

[4] Basta, A., Kellerer, W., Hoffmann, M., Morper, H. J., and Hoffmann, K. Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem. In *Proc. of All Things Cellular* (2014).

[5] Gudipati, A., Perry, D., Li, L., and Katti, S. SoftRAN: Software defined radio access network. *Proc. of HotSDN* (2013).

[6] Jin, X., Li, L., Vanbever, L., and Rexford, J. SoftCell: Scalable and flexible cellular core network architecture. *Proc. of ACM CoNEXT* (2013).

[7] Kempf, J., Johansson, B., Pettersson, S., Luning, H., and Nilsson, T. Moving the mobile Evolved Packet Core to the cloud. In *Proc. of IEEE WiMob* (2012).

[8] Li, L. E., Mao, Z. M., and Rexford, J. Toward Software-Defined Cellular Networks. *2012 European Workshop on Software Defined Networking* (2012).

[9] Moradi, M., Wu, W., Li, L. E., and Mao, Z. M. SoftMoW: Recursive and Reconfigurable Cellular WAN Architecture. In *Proc. of ACM CoNEXT* (2014).

[10] Osmani, L., Lindholm, H., Chemmagate, B., Rao, A., Tarkoma, S., Heinonen, J., and Flinck, H. Building Blocks for an Elastic Mobile Core. In *Proc. of ACM CoNEXT Student Workshop* (2014).