A Refactoring Approach for Optimizing Mobile Networks

Matteo Pozza*, Ashwin Rao*, Armir Bujari†, Hannu Flinck‡, Claudio E. Palazzi†, and Sasu Tarkoma*,
*University of Helsinki †University of Padua ‡Nokia Bell Labs
{firstname.lastname}@{helsinki.fi,nokia-bell-labs.com}, {abujari,cpalazzi}@math.unipd.it

***Abstract*—Mobile networks are expected to serve a wide range of verticals, however the Long Term Evolution (LTE) network is optimized for basic mobile operator services only. The network functions serving LTE networks are largely implemented as dedicated single function devices that offer poor customization options. This intrinsic inflexibility makes current LTE networks unable to meet the requirements of future mobile networks. For example, LTE networks experience signaling storms because the signals exchanged by the network functions cannot be optimized according to the current usage pattern of mobile services. Modularizing these network functions would enable a refactoring of the LTE network, allowing operators to compose networks that adapt and evolve with the influx of verticals. In this article, we present a new approach for refactoring the network functions serving LTE networks which can be leveraged to compose a modular mobile network optimized for the verticals using its services. As an example, we demonstrate that deploying network functions at the edge significantly reduces the signals exchanged within a mobile network.**

***Index Terms*—refactor, mobile networks, 5G.**

I. Introduction

The next generation mobile networks are expected to serve a wide range of verticals, each of which is expected to arrive with a unique set of requirements from the control plane and data plane. For example, critical communication such as autonomous driving has different control plane and data plane requirements compared to wireless sensors that are expected to be deployed in a house. Therefore, mobile networks must be able to dynamically transform to meet these requirements.

This influx of verticals imposes serious challenges to current Long Term Evolution (LTE) networks [1]. The LTE network is optimized for mobile operator services, as it is the outcome of satisfying the requirements of a high-speed data network while being compatible with previous generation mobile networks. The network functions serving current LTE networks are largely implemented as dedicated single function devices, some of which serve both the control plane and the data plane [2], [3]. The convoluted design of these devices combined with their lack of customization options makes them expensive, difficult to upgrade, and nearly impossible to customize [1], [4]. Consequently, current LTE networks will not be able to serve a large number of verticals.

Modularizing the network functions will allow operators to refactor the network in order to meet the requirements coming from the verticals using its services. In this article, we detail our approach for refactoring the network functions serving LTE networks. Our refactoring approach begins by abstracting and modularizing the network functions of LTE networks. These modules can then be mapped to a desired number of physical devices depending on the demands from the mobile network. As an example, we show how our refactoring approach can be leveraged to mitigate the problem of signaling storms in LTE networks. In particular, we compare three possible ways in which the network can be refactored to address this problem. Our key contributions are as follows.

1) We analyze and abstract the tasks of the various network functions serving LTE networks. This gives us a list of high-level services provided by each network function. Moreover, this also gives us insights on the avenues for optimizing mobile networks to satisfy the requirements of the verticals that use their services.
2) We split the network functions in modules by identifying both the role and the requirements of each network function. These modules form the cornerstone for providing the aforementioned services.
3) We present how to meet the requirements coming from a specific vertical by mapping the modules to the physical devices present in the network. As an example, we show how our refactoring approach can be leveraged to address the signaling storm issue by decreasing the number of signals exchanged between the network functions. A decrease in the number of signals also has a direct impact on the latency experienced by the mobile devices.

We were motivated by the seminal works of Li *et al.* [3] and Gudipati *et al.* [5], which detail the benefits of bringing Software Defined Networking (SDN) to cellular networks. We also leverage the insights of previous works that propose a logically centralized control plane for mobile networks [6], [7], [8], [9]. Similarly, Hampel *et al.* [10] have also explored the benefits of splitting the control plane and data plane of the network functions. In §IV, our proposals are based on the insights from Moradi *et al.* [9] and Jin *et al.* [11], which advocate splitting the control plane and moving some of its components to the base stations.

The rest of the paper is organized as follows. In §II we discuss the inflexibility of the network functions in LTE networks. Then we discuss our approach to abstract and modularize the network functions in §III, followed by an application of this method in §IV and a discussion in §V.

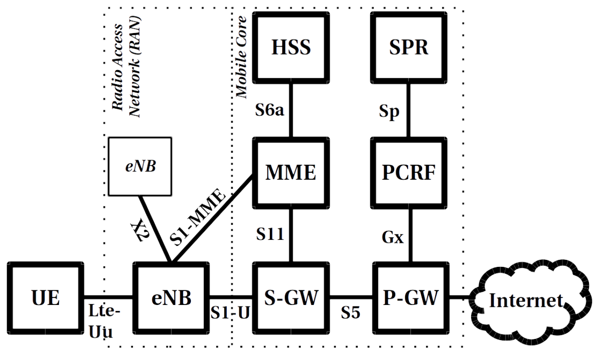


Fig. 1. **Key network functions and interfaces in an LTE Network.** *An LTE network consists of the Radio Access Network (RAN) and the Mobile Core. The eNB, S-GW, and P-GW act as a data bridge between UEs and the Internet. The MME takes care about the management of network events, as handovers, and the PCRF generates the QoS rules for the data flows. The HSS and SPR store subscriber-specific data.*

II. Background and Motivation

In this section, we first provide an overview of the current LTE architecture, and then we briefly discuss its shortcomings.

A. Overview of the LTE Architecture

As shown in Figure 1, current LTE networks consist of two sub networks: the Radio Access Network (RAN) and the Mobile Core [2]. The key network function in the RAN is the Evolved Node B (eNB). The eNBs use their Lte-Uu interface to communicate with mobile devices, henceforth referred to as User Equipment (UE). Furthermore, eNBs may use their X2 interface to communicate with some of the other eNBs in the LTE network during handovers. An eNB is also connected to two network functions of the Mobile Core: the Mobile Management Entity (MME) via the S1-MME interface, and the Serving Gateway (S-GW) via the S1-U interface. The key roles of the MME include ensuring smooth handovers and retrieving customer-specific values from the Home Subscriber Server (HSS), a database storing the users' subscription information. The S-GW acts as a mobility anchor during handovers and it is typically linked to more than one base station. It is also linked to the Packet Data Network Gateway (P-GW), the gateway to the external networks. The P-GW also assigns the IP addresses to the UEs and enforces the Quality of Service (QoS) rules received by the Policy and Charging Rules Function (PCRF). The PCRF uses the values retrieved by the Subscriber Profile Repository (SPR) in turn to generate the rules fed to the P-GW.

B. Shortcomings of the LTE Architecture

Current LTE networks are inflexible because their key network functions are implemented as dedicated single function devices. These devices are typically sold as vendor-specific black boxes that carry out a fixed set of tasks [12]. Furthermore, some devices serve both the control plane and the data plane [3], [4], resulting in a further loss of uniformity in the management of the network. In this context, mobile operators cannot deploy new services, test solutions for common problems such as handling flash crowds, or perform optimizations like power saving. This is a serious shortcoming given the wide range of verticals expected to use future mobile networks [1].

A glaring example of this inflexibility are signaling storms. This problem highlights the inability of the LTE networks to adapt to the different ways in which their services are used by UEs. For example, apps like WhatsApp periodically send heartbeat messages which make UEs frequently connecting and disconnecting from mobile network [13]. However, each time a UE changes its status (*e.g.* power on, go idle), the UE and the network functions serving the UE undergo a series of actions called *procedures*. In carrying out the procedures, the network functions internally exchange a large number of messages called *signals* to ensure a consistent state of the UEs they serve [14], [15]. In addition, the number of mobile devices has been increasing significantly in the past years, and it is expected to keep its growth in the years to come [16]. Mobile operators have limited opportunities to decrease the number of signals per procedure, and consequently they have to face an exponential growth in the signaling load, which leads to network congestion [17].

In the following, we present our approach to address this inflexibility. In particular, we show how the approach can be leveraged in order to reduce significantly the number of signals required to carry out the procedures triggered by UEs.

III. Abstract and Modularize the Network Functions

Our refactoring approach consists in three steps. First, we identify the roles of the network functions serving LTE networks. Then, we split each network function in modules, creating one module for each role of the network function. Moreover, for each module, we identify the requirements of a physical device instantiating that module. Finally, we show that we can change the mapping between physical devices and modules depending on the requirements from the network.

A. Leverage Abstractions to Identify Modules

In this first phase, we inspect the network functions in order to identify the high-level functions carried out by the whole LTE network. The Software-Defined Networking paradigm provides us two key abstractions for any communication system: a) the control and management plane and b) the data plane. Indeed, the network functions take care of the forwarding of the data between UEs and the Internet. At the same time, they also take care about the management of the network, as the creation of appropriate QoS rules and the handling of handovers. Nevertheless, we think that an LTE network carries out also some duties that go beyond control plane and data plane. In fact, an LTE network is in charge of storing subscriber-specific parameters that are used during generation of the QoS rules and security procedures. Therefore, an LTE network takes care also about the storage and provisioning of customers' information.

As shown in Figure 2, we use three layers to abstract an LTE network: a) the *Storage* layer for persistent data and database-like services, b) the *Control* layer for the management of the

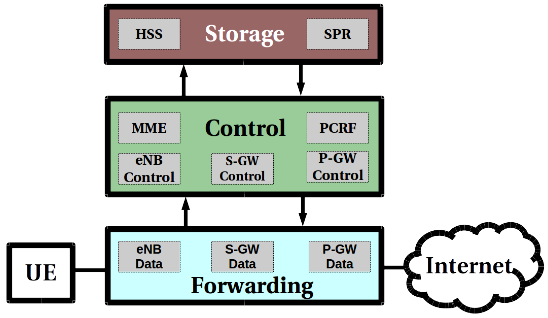


Fig. 2. **Three-layers abstraction.** *An LTE network can be abstracted into a Storage layer for database-like services, a Control layer for network management, and a Forwarding layer for data flows handling. The Storage layer and Control Layer match the control and management plane of LTE networks while the Forwarding layer corresponds to the data plane.*

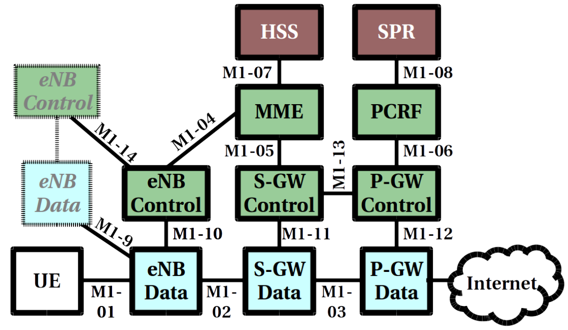


Fig. 3. **Modularized network functions.** *The eNB, S-GW, and P-GW are split as Control layer and Forwarding layer modules with interfaces added to connect these modules. This step allows us to study the interaction between the modules and explore the impact of combining them.*

forwarding elements, and c) the *Forwarding* layer for handling data flows according to the rules imposed by the *Control* layer. The *Forwarding* layer acts as the data bridge between the UE and the Internet. It also forwards the control plane traffic from the UE to the *Control* layer (*e.g.*, signals during authentication). The *Control* layer takes decisions based on the statistics coming from the data plane (*e.g.*, bandwidth, queue length in the gateways, *etc.*) and the policies generated using the information in the *Storage* layer. Consequently, it instructs the forwarding devices on how to serve the packets.

B. Abstract the Roles of Network Functions

The aim of this step is to map the network functions to the previously identified roles: *Storage*, *Control*, and *Forwarding*. For network functions that participate in more than one role, we split each network function in modules such that we get a module for each role. These splits are useful to untangle the intrinsic convolution of the current network functions.

As shown in Figure 2, the MME and PCRF serve the control plane because they take only network management decisions. Similarly, the HSS and SPR belong to the *Storage* layer because they provide subscribers' data. In contrast, the eNB, the S-GW, and the P-GW serve both the control plane and the data plane. We therefore split each one of these network functions into two modules, one for the *Forwarding* layer and the other for the *Control* layer. We also explicitly add a communication interface between these modules. The outcome of these splits is presented in Figure 3.

We now define the requirements associated with each module. The modules in the *Storage* layer require a database platform such as NoSQL or relational databases. Similarly, the *Control* layer modules need a computing platform for their software because all control tasks are software programs which can be executed on commodity hardware. In contrast, the *Forwarding* layer modules have more specific requirements. All of them need hardware for enforcing QoS rules received from the *Control* layer. Moreover, the eNB Data module requires a radio interface to communicate with UEs. Finally, S-GW Data and P-GW Data require switching/routing capabilities.

C. Map Modules to Physical Devices

The last step is the association between the identified modules and the physical devices we have. Indeed, once the requirements of each module have been defined, we just have to provide a hardware device (or a set of hardware devices) which is able to meet them. Furthermore, multiple modules can be instantiated on the same physical device. As an example, consider the *Control* layer modules. The only requirement of these modules is a computing platform. As a consequence, we can run all the *Control* layer modules in a single server, or we can run part of them in a cloud environment, or we can assign a dedicated physical device for each module. This decision is driven by the objectives for which the network is deployed. If we need to scale our control plane with the number of connected subscribers, then we map the *Control* layer modules to a cloud environment. If we need an emergency network for the organization of rescue operations instead, then we put all the modules in a single, portable device.

We argue that this association can be tailored to satisfy the specific needs coming from different use cases. In this way, we obtain the flexibility that is needed by mobile networks in order to meet the requirements of current and future verticals. For example, in the context of signaling storms, we can instantiate the modules with the aim of reducing the number of signals exchanged between the devices in the network.

IV. USE CASE: REDUCING THE SIGNALING LOAD

We now show a practical example of how to use our approach. We focus on the requirements that have recently arisen in LTE networks, *i.e.* the need of serving an increasing number of UEs and the ability to deal with short and frequent connections. Current LTE networks are unable to meet these requirements because they experience signaling storms, *i.e.* excessive signaling loads that compromise the network performance. Therefore, we leverage our approach to reduce the number of signals required by the procedures executed in LTE networks. A decrease in the number of signals also has a direct impact on the latency experienced by the mobile devices. We restrict our analysis to the procedures triggered by

TABLE I
TOTAL NUMBER OF SIGNALS EXCHANGED

	Total no. of signals per procedure					
Implementation	*IA*	*AtI*	*ItA (UE)*	*ItA (Net)*	*X2H*	*S1H*
LTE (Baseline)	35	6	13	17	15	22
Modularized Architecture	57	11	24	30	31	41
Split RAN & Core	26	8	15	19	20	20
Thin Edge	24	6	13	16	16	16
Intelligent Edge	17	3	10	12	12	12

Total number of signals during Initial Attach (IA), Active to Idle (AtI), Idle to Active (ItA), and Handover procedures. For the ItA procedure, we show the amounts for the UE-triggered and the network-triggered variants. Similarly, we describe the signaling cost for both X2 and S1 handover. The signals observed by existing LTE networks are the base line for our comparison. Details of these signals are presented in extended version of this paper [18].

the following events: a) the UE connects to an LTE network (*Initial Attach*), b) the UE goes idle (*Active to Idle*), c) an idle UE becomes active (*Idle to Active*), and d) the UE moves to another cell (*Handover*). The selected procedures either require a huge number of signals or they are frequently invoked [17].

Table I shows the total number of signals required by the current LTE architecture for each one of these procedures. We can see that *Initial Attach* is the most onerous procedure because it requires 35 signals. Indeed, this procedure needs to carry out several duties, such as establishing secure connections with the UE and setting the QoS parameters along the data path. For example, in Figure 5(a) we present the source and destination of the signals exchanged while setting the QoS parameters. The transitions between idle and active states require fewer signals because they just need to remove or restore previously computed state variables. Finally, S1 handover requires a high number of signals because it is invoked when the base stations are not directly linked. The details on the signals exchanged during each procedures are presented in extended version of this paper [18].

In order to reduce the number of signals required by these procedures, we present three examples of refactoring the network functions. In each example, we refactor the network functions by coalescing in different ways the modules identified in §III-A. *Coalescing in this context implies running the modules in the same physical machine.* The coalesced modules internally exchange signals with each other but these signals do not leave the physical machine, resulting in a decrease in the number of signals traversing the interfaces connecting these physical machines. The modules in the *Forwarding* layer, *i.e.* the *eNB Data*, the *S-GW Data*, and the *P-GW Data*, can be conceptualized as SDN switches and routers which can be dynamically programmed by the *Control* layer. Similarly, we can consider the modules of the *Control* layer as software modules which can be coalesced and instantiated as virtual network functions [19]. Finally, the modules of the *Storage* layer can be coalesced in a cloud database or a server providing database services. We use this exercise to explore the impact of composing network functions in specific ways.

A. Split RAN & Core

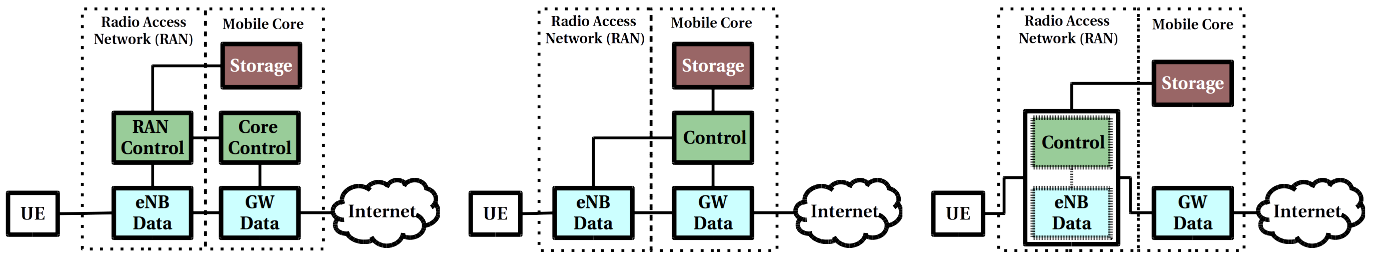
This example aims to reduce the number of signals by coalescing the modules but keeping the RAN independent from the Core. This example enables the analysis of topologies aimed for fault tolerance, such as linking auxiliary infrastructures (*e.g.* backup RAN and Core) to be used when experiencing issues caused by natural disasters.

As shown in Figure 4(a), we coalesce the *Control* layer modules in two entities: the RAN Control and the Core Control. The RAN Control is the product of coalescing the eNB Control, the MME, and the PCRF, *i.e.*, these modules are instantiated on the same physical machine in the RAN. Furthermore, having the MME and PCRF in the same physical machine requires only one interface to the Storage entity. Similarly, S-GW Control and P-GW Control are coalesced in the Core Control entity, and S-GW Data and P-GW Data are coalesced in the GW Data entity. We would like to point out that our approach is not completely new. In many LTE networks the S-GW and P-GW are largely implemented as a single box [12]; this example can also be seen as a step in using a legacy S-GW and P-GW in the Mobile Core while moving some of its control logic to the RAN. In this example, most of the decisions are taken in the RAN Control while the Core Control is only responsible for managing the *Forwarding* layer of the Mobile Core. Finally, the Storage entity comprises both HSS and SPR: this means that the network uses a single database for all the subscriptions' information.

As presented in Table I, we observe that the number of signals exchanged between the entities of this network during the Initial Attach decreases to 26. The main cause for this decrease is the generation of the policy rules directly in the RAN. Indeed, the PCRF module is now directly linked to the single Storage entity, and therefore it can retrieve all the needed information through a single access. For example, in Figure 5(b) we observe a fewer number of signals exchanged for sharing the QoS parameters compared to signals exchanged in the current LTE architecture presented in Figure 5(a). However, these gains come at a cost of increased signals for other procedures; we observe that this approach requires more signals compared to existing LTE networks during all the other procedures. These increases come from the indirection between the RAN Control and the GW Data: in the LTE architecture, the MME communicates directly with the S-GW, while here we have the Core Control in between.

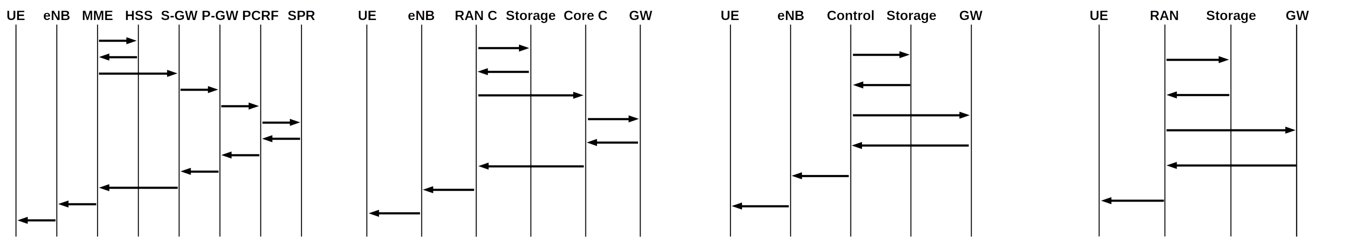
B. Thin Edge

This example is used to satisfy the requirement of having minimal resources in the RAN. Indeed, as shown in Figure 4(b), in this example all the modules in the *Control* layer are coalesced and instantiated on a single physical server in the Mobile Core. The proposal follows the idea of having dumb, software-defined base stations remotely instructed by a control plane which has been shifted to the cloud [9], [5]. The eNB Data serves the data plane of the RAN, while the GW Data serves the data plane of the Mobile Core.



(a) Split RAN & Core. (b) Thin Edge. (c) Intelligent Edge.

Fig. 4. **Examples of refactoring LTE networks.** *The Split RAN & Core aims to keep the RAN independent from the Core, such that we can set up a backup configuration in case of Core failure. The Thin Edge satisfies the requirement of having all the control logic in the Mobile Core, keeping the forwarding devices as-dumb-as-possible. The Intelligent Edge satisfies the requirement of having all the control logic in one physical box in the RAN.*



(a) LTE. (#signals = 12) (b) Split RAN & Core. (#sig. = 8) (c) Thin Edge. (#signals = 6) (d) Intelligent Edge. (#signals = 5)

Fig. 5. **Source (arrow tail) and destination (arrow head) of signals required to exchange QoS parameters during Initial Attach.** *Current LTE networks require a large number of signals because the QoS parameters are stored in the HSS and the SPR while the parameters are processed in the PCRF. Coalescing the HSS and the SPR into a single entity and simultaneously reducing the number of control entities decreases the number of signals required to complete the exchange of QoS parameters.*

As shown in Table I, this refactoring and coalescing of modules results in fewer signals compared to existing LTE networks. For example, during the *Initial Attach* a total of 24 signals are exchanged between the various components. The merging of the two control entities of the previous example is the primary reason behind the decrease in the number of signals. As an example, Figure 5(c) shows that the number of signals required in exchanging QoS parameters during *Initial Attach* is now smaller than both the previous case and the default case. Nevertheless, this example results in an increase in the signals when handling an equivalent of *X2 handovers*. Indeed having moved the control plane away from the base stations implies that they cannot use the X2 interface (interface connecting eNBs) anymore in order to configure the handover procedure; the X2 interface serves only data plane messages.

C. Intelligent Edge

This example is used to satisfy the requirement of having all the control logic in the RAN. As shown in Figure 4(c), the modules in a control plane are instantiated as virtual machines in a device running the eNB Data. This is done to make use of the free computation power which is available in the base stations. Indeed, Yousaf *et al.* [20] point out that up to 80% of the processing capacity of base stations is not used. Therefore, we leverage this capacity to carry out control plane tasks.

As shown in Table I, this refactoring and coalescing of modules results in fewer signals compared to existing LTE networks, and it also outperforms the example of the Thin Edge. For example, during the *Initial Attach* a total of 17 signals are exchanged between the various components. The main reason behind this saving is the merging of eNB Data and the whole control plane in the same physical device. In fact, LTE networks require UEs to tightly interact with the MME for many purposes, as authentication and exchanging of security and QoS parameters, and the eNB acts as a relay for these messages. Therefore, coalescing eNB with the Control entity results in a great reduction in the number of signals required, as shown in Figure 5(d). We would like to point out that shifting the entire control logic to RAN can pose administrative and domain issues for some operators. However, these issues are beyond the scope of this paper. Our objective for presenting these examples is to highlight the benefits of modularizing and refactoring the network functions.

D. Analysis and Discussion

We now compare the signaling load generated when these approaches are deployed to replace an existing LTE network. For this comparison, we combine our results from Table I with real frequencies of the considered procedures. Metsälä *et al.* [14] provide data for the control plane traffic in an LTE network. We use a subset of this data, and in Table II(a), we present the frequency of procedures per busy hour per subscriber per base station. Please note that LTE networks in different geographical regions will have different values for the observed frequencies.

TABLE II
IMPACT OF REFACTORING.

Dataset	IA	AtI	ItA (UE)	ItA (Net)	X2H	S1H
	Frequency of procedures					
Metsälä *et al.* [14]	0.5	34	19	15	8	0.2

(a) Frequency of procedures.

Implementation	IA	AtI	ItA (UE)	ItA (Net)	X2H	S1H
	Frequency of signals					
LTE (Baseline)	17.5	204	247	255	120	4.4
Split RAN & Core	13	272	285	285	160	4
Thin Edge	12	204	247	240	128	3.2
Intelligent Edge	8.5	102	190	180	96	2.4

(b) Frequency of signals considering Metsälä *et al.* [14].

Using the dataset of Metsälä et al. [14], we quantify the impact of the different examples on the signaling load, i.e., number of signals per busy hour per subscriber per base station for a given procedure.

For the given dataset, the *Intelligent Edge* approach outperforms the other approaches. Furthermore, the examined network has a small number of *Initial Attach* and a small number of *S1 handovers* compared to the X2 Handovers. As a consequence, the *Split RAN & Core* approach is not suitable for this network. Nevertheless, the *Split RAN & Core* approach is still beneficial in networks with a high frequency of *Initial Attach* events, a potential scenario in IoT context for example. Finally, we can see the *Thin Edge* approach performs poorly only when handling the equivalent of *X2 handovers*. Therefore, this approach is beneficial for network deployments which do not require base stations to be directly connected to each other.

Our approach of splitting the network functions into small modules is in line with the recent proposals made by 3GPP for the Next Generation System [21]. Indeed, the proposed solutions have native support for Software-Defined Networking and Network Functions Virtualization, which are also leveraged by our refactoring work.

V. CONCLUSION AND FUTURE WORK

Mobile networks are expected to meet the needs of current and future verticals. Each one of these verticals comes with different requirements, so mobile networks have to dynamically adapt depending on the deployment scenario. Unfortunately, the network functions serving current LTE networks are inherently inflexible, making them not suitable for the aforementioned aims. We therefore abstract the roles of the key network functions serving LTE networks. This empowers us with a vantage point to modularize the network functions. These modules can be leveraged in turn to refactor the LTE network in order to serve specific deployment scenarios. In particular, we have shown how a refactoring that moves the control plane close to the base stations reduces significantly the impact of signaling storms on mobile networks.

The issues that may arise in the transition from a legacy infrastructure to the proposed refactoring have still to be identified. For this reason, we are planning to implement these network modules as the next step in our research. We also aim to explore the replication of network modules in different physical devices for fault tolerance and scalability purposes.

To conclude, we believe that our work provides a key building block for composing modular mobile networks which can adapt to serve the verticals of future mobile networks.

ACKNOWLEDGEMENT

This work has been supported by the Nokia Center for Advanced Research (NCAR).

REFERENCES

[1] "Taking 5G to other industries - a view from the top," http://networks.nokia.com/news-events/insight-newsletter/articles/taking-5g-to-other-industries-a-view-from-the-top, 2015.

[2] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3rd Generation Partnership Project (3GPP), TS 23.401, Dec. 2015.

[3] L. E. Li, Z. M. Mao, and J. Rexford, "Toward Software-Defined Cellular Networks," in *Proc. of EWSDN*, 2012, pp. 7–12.

[4] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 107–115, Feb 2015.

[5] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "Softran: Software defined radio access network," in *Proc. of ACM HotSDN*, 2013, pp. 25–30.

[6] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "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, pp. 1–7.

[7] J. Kempf, B. Johansson, S. Pettersson, H. Luning, and T. Nilsson, "Moving the mobile Evolved Packet Core to the cloud," in *Proc. of IEEE WiMob*, 2012, pp. 784–791.

[8] H. Lindholm, L. Osmani, H. Flinck, S. Tarkoma, and A. Rao, "State Space Analysis to Refactor the Mobile Core," in *Proc. of ACM AllThingsCellular Workshop*, 2015, pp. 31–36.

[9] M. Moradi, W. Wu, L. E. Li, and Z. M. Mao, "SoftMoW: Recursive and Reconfigurable Cellular WAN Architecture," in *Proc. of ACM CoNEXT*, 2014, pp. 377–390.

[10] G. Hampel, M. Steiner, and T. Bu, "Applying Software-Defined Networking to the telecom domain," in *IEEE INFOCOM Workshops*, 2013, pp. 133–138.

[11] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *Proc. of ACM CoNEXT*, 2013, pp. 163–174.

[12] Alcatel-Lucent, "7750 Service Router - Mobile Gateway," https://www.alcatel-lucent.com/products/7750-service-router-mobile-gateway.

[13] Diametriq, "A storm is brewing – an lte signaling storm," http://www.diametriq.com/wp-content/uploads/downloads/2013/01/A-Storm-is-Brewing.pdf, 2012.

[14] E. M. Metsälä and J. Salmelin, *LTE Backhaul - Planning and Optimization*. Wiley, 2015.

[15] A. Banerjee, R. Mahindra, K. Sundaresan, S. Kasera, J. V. der Merwe, and S. Rangarajan, "Scaling the LTE Control-Plane for Future Mobile Access," in *Proc. of ACM CoNEXT*, 2015.

[16] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020 white paper," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, Feb 2016.

[17] Nokia, "Managing lte core network signaling traffic," 2013, https://insight.nokia.com/managing-lte-core-network-signaling-traffic.

[18] M. Pozza, "Solving signaling storms in lte networks: a software-defined cellular architecture," Master Thesis, University of Padua, http://tesi.cab.unipd.it/53297/1/tesi_Pozza.pdf, Jun 2016.

[19] ETSI, "Network Functions Virtualization - Introductory White Paper," https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf, 2014.

[20] F. Z. Yousaf, J. Lessmann, P. Loureiro, and S. Schmid, "SoftEPC - Dynamic instantiation of mobile core network entities for efficient resource utilization," in *Proc. of IEEE ICC*, 2013, pp. 3602–3606.

[21] 3GPP, "Study on Architecture for Next Generation System," 3rd Generation Partnership Project (3GPP), TS 23.799, Dec. 2016.