Self-X for Multi-Tenant Cloud Enabled Small Cells

J. Pérez-Romero
UPC
Barcelona, Spain
jorperez@tsc.upc.edu

C. Costa
FBK Create-Net
Trento, Italy
ccosta@fbk.eu

B. A. Abubakar, E. Panaousis,
H. Mouratidis
University of Brighton
Brighton, UK
B.Abubakar2@brighton.ac.uk

K. M. Nasr, S.Vahid, K. Moessner
Institute for Communication Systems,
5G Innovation Centre, University of Surrey
Guildford, GU2 7XH, UK
k.nasr@surrey.ac.uk

Abstract—**The combination of Small Cells as a Service (SCaaS) together with Network Function Virtualization (NFV) and Mobile Edge Computing (MEC) technologies, as considered by the SESAME project, provides a promising framework for dealing with the challenging requirements of mobile broadband service provisioning in multi-tenant dense scenarios. Under this context, this paper focuses on the development of Self-X functions to support the autonomic management of multi-tenant cloud enabled small cells in these scenarios. After presenting the architecture for supporting these functions in the SESAME project, the paper presents three different examples of Self-X functions, dealing with cross-layer TCP optimization during handovers, on-demand virtualized resource allocation and mobility load balancing.**

Keywords—Small Cells; Self-X; NFV; Multi-tenancy; MEC

I. INTRODUCTION

Small Cell deployment is considered as a means to effectively increase the available capacity in high traffic areas as well as locations with poor coverage conditions from outdoor macrocells and therefore it will become a fundamental component to deal with the stringent traffic requirements of Mobile BroadBand (MBB) services in future systems, particularly in localised areas of high user density (e.g. stadiums, malls, etc.). Given the characteristics of these scenarios, dedicated operator deployments are impractical and, as a result, the use of neutral host models, such as Small Cells as a Service (SCaaS) [1], becomes an attractive solution. In this approach, a third party deploys and operates a small cell network which is shared by multiple mobile network operators (MNOs), also denoted as "tenants", to provide services to their customers, thus resulting in benefits in terms of both CAPEX and OPEX reduction.

Another relevant trend in the deployment of future wireless networks relies on the introduction of softwarisation and virtualisation technologies in the mobile network. Network Function Virtualization (NFV) [2] refers to the software implementation of network functions running on general purpose computing/storage resources, providing an inherent flexibility to modify network configurations in real time, and facilitates infrastructure sharing between tenants through the use of multiple virtual network running on the same infrastructure. The introduction of NFV technologies and the availability of cloud-computing capabilities in the Radio Access Network (RAN) enable the provision of Mobile Edge Computing (MEC) services [3], which facilitate the reduction in service latencies and the quick introduction of new services.

Relying on the abovementioned concepts, the Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME) project [4] focuses on the provision of SCaaS under multi-tenancy, exploiting the benefits of NFV and MEC. For that purpose, it proposes the Cloud-Enabled Small Cell (CESC) concept, a new multi-operator enabled Small Cell (SC) that integrates a virtualized execution platform for executing novel applications and services inside the access network infrastructure.

The efficient management of the resulting multi-tenant cloud enabled small cell network and the complexity of the highly dense environments where these networks are envisaged to be deployed require the introduction of Self-Organizing Networks (SON) functionalities, also denoted as Self-X functions. They include a set of functions for reducing or even removing the need for manual network optimization tasks, so that operating costs can be reduced as well as revenue can be protected by minimizing human errors [6]. Besides, the development of Self-X functions can benefit as well from the virtualized execution platform provided by the CESCs and by the use of NFV and MEC technologies.

Under the above context, this paper intends to present the framework for the development of Self-X functions in a multi-tenant cloud-enabled RAN considered in the SESAME project and to particularize it with different exemplary functions. The paper is organized as follows. Section II summarizes the SESAME architecture from the perspective of Self-X functions. Sections III, IV and V present three different Self-X use cases dealing with, respectively, cross-layer TCP optimization during handover, on-demand virtualized resource allocation and Mobility Load Balancing (MLB). Finally, Section VI concludes the paper.

II. ARCHITECTURE FOR SUPPORTING SELF-X IN CLOUD ENABLED MULTI-TENANT SMALL CELLS

The architecture of SESAME that addresses the evolution of the SC concept through the NFV, SON and MEC paradigms is presented in [7] and depicted in Fig. 1. In general terms, SESAME scenarios assume a certain venue (e.g. a mall, a stadium, an enterprise, etc.) where a Small Cell Network Operator (SCNO) is the SCaaS provider that has deployed a number of CESCs that provide wireless access to end users of different operators, denoted as Virtual Small Cell Network Operators (VSCNOs), according to specific Service Level Agreements (SLAs). RAN sharing in SESAME focuses on the Multi-Operator Core Network (MOCN) model [5], where the core networks of the VSCNOs are connected to the SCNO's RAN.

The main components of the architecture of Fig. 1 are: (i) the CESC, which consists of complete MOCN-enabled SC composed by a physical SC unit, i.e. the SC Physical Network Function (PNF) and a micro-server; (ii) the Light DC, which results from the physical aggregation of the micro-servers of multiple CESCs in a cluster, thus constituting the virtualised execution infrastructure; (iii) the CESC Manager (CESCM), which is the central service management component that integrates the 3GPP network management elements, i.e. the Element Management System (EMS), and the functional blocks for the management and orchestration of virtualised networks, i.e. the Network Function Virtualisation Orchestrator (NFVO) and the Virtualized Network Function Manager (VNFM), which rely on the Virtualized Infrastructure Manager (VIM) for managing the virtual resources of the Light DC. The reader is referred to [7] and references therein for a detailed description of all these blocks.

Self-X functions will tune global operational settings of the SC (e.g., transmit power, channel bandwidth, electrical antenna tilt) as well as specific parameters corresponding to Radio Resource Management (RRM) functions (e.g., admission control threshold, handover offsets, packet scheduling weights, etc.). As shown in Fig. 1, the PNF EMS and SC EMS include the centralised self-x functions (cSON) and the centralised components of the hybrid SON functions. In turn, the dSON functions - or the decentralised components of the hybrid functions - reside at the CESCs. Concerning the dSON functions, they can be implemented as a PNF or, if proper open control interfaces with the element (e.g. the RRM function) controlled by the self-x function are established, they can also be implemented as VNFs running at the Light DC. Based on the architecture of Fig. 1, the next sub-sections present specific examples of Self-X functions.

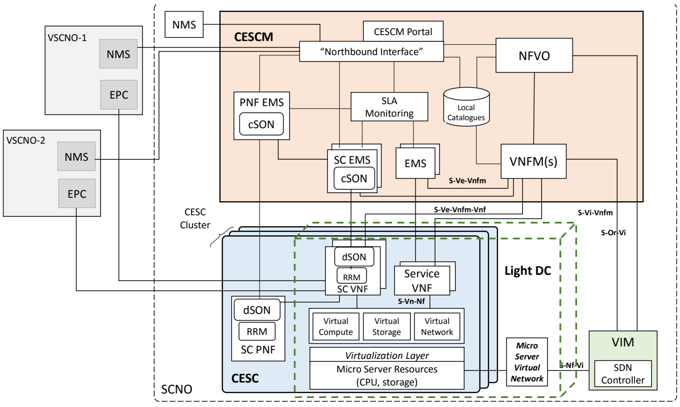

Fig. 1. SESAME architecture

III. Self-X service for cross-layer TCP optimization

Self-X functionalities targeting TCP optimization may be offered as a service to tenants whose contents are made available in the Light DC (e.g. a caching video server), timely detecting events that might deteriorate TCP performance (such as low channel quality or handovers) and autonomously taking action to mitigate them. TCP is commonly adopted in video streaming services in combination with Dynamic Adaptive Streaming over HTTP (DASH) [8]. Such applications require both low delay and jitter and high throughput. TCP was designed in wired networks to relieve network congestion as the main cause of packet loss, and it does not consider that radio links are prone to transmission errors or can be affected by mobility [9]. Referring to Fig. 1, we propose to employ a type of Self-X function to aid TCP in Cloud-Enabled Small Cells. When fast system reaction is required, we envisage a dSON function that trespass the typical boundaries of Self-X in order to autonomously tweak TCP parameters at the server side, taking into account radio signal propagation, traffic and UEs mobility.

A. *Example of solution for Handover optimization*

Typically, when transmitting TCP traffic, lossless handover (HO) is used in combination with RLC Acknowledge Mode (AM). In this type of HO both the untransmitted PDCP SDUs (Service Data Units) in the serving eNodeB and those in the PDPC retransmission queue are forwarded to the target eNodeB, resulting in increased delays. This approach supports high throughput but, unfortunately, the price to be paid is an increased delay. This may be favourable to applications such as FTP and HTTP, but for DASH it becomes critical factor, causing undesirable effects on the user's QoE (e.g. video freezing). In small cells deployments, where handovers may occur frequently, this can become a serious issue. TCP congestion control algorithms are designed to continuously estimate the available bandwidth, but this process is slow and it takes several round trip times (RTT) for the TCP to adjust to the correct transmission rate. Typically, the congestion window (*cwnd*) is adjusted to regulate the packet transmission rate (the smaller *cwnd*, the lower the transmission rate). Sudden changes of connectivity, typical of handovers, are difficult to address. Although some TCP versions are designed for being robust against bursts of packets lost, cross-layers approaches can explicitly address TCP performance during handovers. It was shown that information related to handovers (e.g. explicit handover notification) fed back to the video server can mitigate the effects of mobility since the sender is able to correctly interpret the cause of packet loss [10].

In this work, we present initial results on a cross-layering type of Self-X function that aids TCP during lossless HO. We propose to use an algorithm as part of the dSON to make a prediction of the occurrence of an HO event, based on UEs' signal measurements. This estimate is then fed to the TCP server which adaptively sets the TCP re-transmission time out (RTO) to a smaller value immediately before the HO takes effect. In TCP, the RTO is calculated dynamically based on RTT statistic values. During the HO the delay of the packets suddenly increases, and so does also the RTO. Typically, retransmission timeouts occur, causing the sender to enter slow-start, drastically decreasing its *cwnd* thus slowing down the transmission. In our adaptive approach, instead of waiting for a long RTO, this happens immediately before an HO. The amount of data in the PDCP queues to be forwarded to the target eNodeB decreases, while the *cwnd* starts growing shortly after the HO starts, thus enabling a faster recovery compared with the default TCP mechanism.

We used ns-3 for simulating the behaviour of a simple empiric algorithm for HO prediction based on the difference between the reference signal received power (RSRP) of the serving CESC and its best neighbour: when the difference between RSRPs is at least a margin Δ for a time-to-trigger (TTT), a HO is predicted to occur, and the RTO is set to a pre-defined value. A counter is increased whenever Pa-Pb is less than or equal to 1dB plus HO Hysteresis, i.e when the handover is likely to happen. Otherwise the counter is set to zero i.e. the handover is less likely to happen. A guard range of 1dB was added to the HO hysteresis against ping-pong handovers. The values for the margin Δ and the new RTO are set empirically. Taking a reference UE travelling across multiple CESCs, the changes of the *cwnd* size over time are shown in Fig. 2. The figure compares adaptive and default methods and shows that our adaptive method exhibits faster recovery time after the HO, enabling the TCP to send a higher number of packets, which ultimately avoids problems such as the video to freeze.

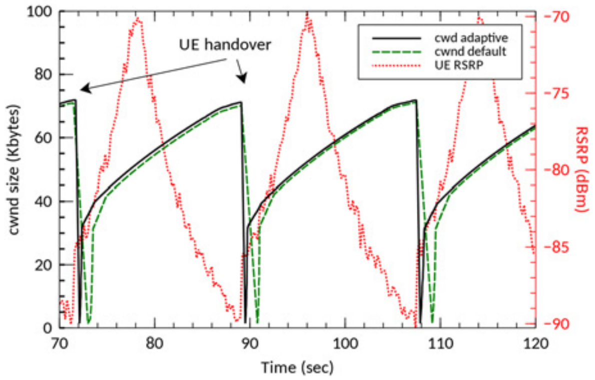

Fig. 2. Results obtained using a simple empiric algorithm: *cwnd* adaptation.

In the case presented above the prediction was implemented as a simple empiric algorithm, but, thanks to the SON approach, it could be improved taking into account additional contextual information that considers more advanced mobility management approaches.

IV. On-Demand Virtual Resource Allocation Model for Multi-Tenant Environment

In multi-tenant virtualised computing and network environments, like the ones proposed in SESAME, dynamic changes in resources' and applications' quality requirements of different tenants can lead to poor resource utilisation when using existing static resource allocation mechanisms. Therefore, for the dynamic requests for resources, such as computing, storage and communication, a dynamic resource allocation mechanism is proposed that will allocate virtual resources to different users on-demand. A well-designed on-demand resource allocation algorithm may optimise the resource allocation by minimising the waste of resources and ensure good quality of service. The SESAME self-optimising resource allocation function for virtualised Small Cells is designed to consider per-tenant resource allocations. It aims at allocating resources to individual tenants based upon an agreed SLA and the available CESC resources. For the virtual system, an upper threshold, in terms of granted resources, should be set for each tenant based on its SLA and service demands. The resource allocation algorithm, in the NFV, should be continuously aware of the total available virtual resources and the tenants' threshold in order to allocate, at least, a minimum required amount resources to a tenant. Fig. 3 illustrates the resource allocation model considered here.

The resources are allocated to individual tenants based on: (i) their subscription determined by SLA; (ii) the available small cell resources; (iii) the location and traffic load.

A tenant should at-least have the resources indicated on his SLA (called *fair resources*) at all times. A *'reserved resource'* is part of small cell resources that are reserved for immediately service delivery when another tenant is using more than its *fair resources*. For example, tenant X has 40% of the resources of each small cell after the *reserved resources* are deducted. This means that tenant X has 40% of the entire resources in SESAME system (in all the small cells) after deducting *reserved resources*. However, to optimise the resource utilisation, a tenant can have more than his allocated *fair resources* in a particular small cell if the following conditions are satisfied:

1. A small cell has ideal available resources.
2. The requesting tenant has available resources in other small cell(s).
3. The other tenant is not using his resources at that point on time.

A. Assumptions and possible scenarios

For a better understanding, let's assume that we have tenants X, Y,…. Let all resources including CPU, memory and storage collectively be referred to as '*resources'*. Let's assume that there are n small cells installed in the entire SESAME system in locations A_1, A_2…A_n. We also assume that 5% of all small cells' resources are '*reserved*' to serve instant service request while a tenant's fair resource is used by another tenant. Let's assume that, based on the SLA, tenant X has 40% of the remaining 95% of all resources and tenant Y has the remaining 60% of the resources.

The possible scenarios are described as follows:

1) For an ideal small cell in a location A_1, when a request is received from tenant X in that location, and the required resources to serve that request are less than or equal to the tenant X's *fair resources* (40%), the request should be served immediately.

2) If tenant X requested more than its *fair resources* (40%) in location A_1, the request should also be served as long as tenant Y is not using part/all his resources (that means there is some part of Y resources that are not utilised) and tenant X has available resources in the other small cells.

3) However, if tenant Y requests part/all its *fair resources*, the request should be served. If the request can be served only using the *reserved resources*, then it should be served immediately and the *reserved resources* used should be recovered from tenant X by migrating part of its services to another available small cell.

4) If tenant Y's request can not be served using only the *reserved resources*, then the *reserved resources* is to be used to start servicing the request and some of the tenant X's services are to be migrated to another available small cell so as to recover tenant Y's *fair resources* and *reserved resources*.

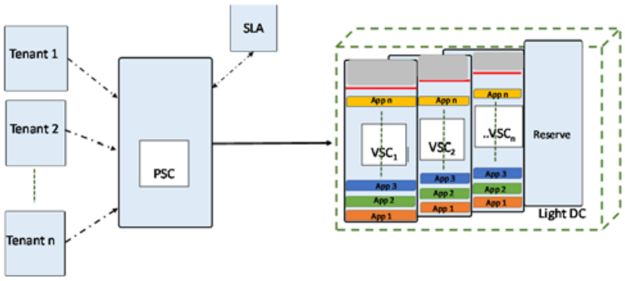

Fig. 3. SESAME Resource Allocation Model

B. Model Formulation

We denote by T = {1, 2,..., n} the set of all tenants. We also express by C the capacity (i.e. all resources) of a small cell and by $L < C$ the safety margin, which is a capacity reserved to serve tenants in case their SLA capacity can not be served by the small cell. There can be different ways of computing L but this is out of the scope of our work here.

We denote by using the random variable $Rx \in [0, 1]$ the proportion of capacity allocated to tenant x. Intuitively, $\sum_{Lx \in T} Rx = 1$. For the sake of our analysis, we assume two tenants x and y. If tenant x consumes (C − L) Rx resources of a small cell (i.e. x spends no more than what they are allowed to by their SLA), then any amount, within the SLA resource, that y will require it can be served. However, if x demands more resources than the ones permitted by his SLA, then it may be the case that the demand of y may not be able to be served without having part of x's capacity being migrated.

We can have the following scenarios taken place. In a case, where x consumes all C − L resources, should a tenant y demands capacity $d_y < L$, then y can be immediately served because the safety margin is adequate to satisfy his demand. As each small cell must have at least L resources reserved for future demands (e.g. we may introduce a third tenant who may request to receive his minimum resources as determined by their SLA). In that case, we must migrate some of x's workload, denoted by Mx, to another small cell, where:

$$Mx := L - d_y .$$

In this way the safety margin is maintained.

In the previous scenario we assume that $d_y < L$; if $d_y > L$ and $d_y > (C - L)Ry$, tenant y seeks to receive more resources than determined by their SLA. In that case, given that tenant x already spends more than $(C - L)Rx$, the system must migrate both tenants' extra workload to another small cell. If we denote tenant's x current workload, in the investigated small cell, by Px, we will have that $Mx = Px - (C - L)Rx$. In that case, y can now use the allocated, by their SLA, resources $(C - L)Ry$ in this small cell and migrate the remaining to another small cell, i.e. $My = d_y - (C - L)Ry$.

C. Results

To evaluate the performance of the model, a static resource allocation model is compared with the on-demand resource allocation model. For example if the SLA of a tenant x says $Rx = 0.4$, then tenant x is allowed to use up to 40% of the resources of any small cell for the static resource allocation model, thus tenant x is not allow to have more than 40% of the resource ($Rx \leq 0.4$). If there are only two tenants, (x, y), then $Ry \leq 0.6$. However, in the dynamic resource allocation model tenant x can have up to 90% resources in a given small cell after the reserve resource (L) is deducted, thus, $Rx + Ry \leq 0.95$. However, $Ry = 1 - Rx + L$. In the static model, random numbers were generated from 0 to 0.4 and from 0 to 0.6 to represent Rx and Ry respectively. While for the on-demand model random numbers were generated from 0 to 0.95 to represent Rx and Ry must not be greater than 0.95, thus, $Rx + Ry \leq 0.95$. The result as illustrated in Fig. 4 shows that the on-demand allocation model has on average about 90% resource utilisation while the static allocation model has on average about 75% utilization. Thus, the on-demand resource allocation model has outperformed the static resource allocation model. In this scenario, there might be some delay in the service delivery. However, since the migration process is done using high-speed backbone connection, the delay cost may be insignificant compared to the cost of the underutilised resources.

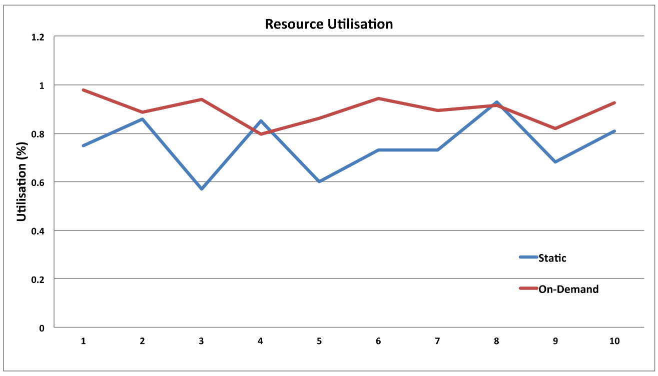

Fig. 4. Resource Utilisation

V. New Approaches to MLB and the User Association Problem

MLB is a self optimisation functionality that intelligently spreads users across system resources to ensure a target QoS and improve edge users throughput. MLB is typically triggered in response to local instances of overload. This reactive approach enables overloaded cells to redirect a percentage of their load to neighbouring less loaded cells hence alleviating congestion problems.

Traditionally, all users use the same set of handover parameters (e.g. hysteresis margin and time to trigger). Moreover, mobility and interference are normally treated separately. Ideally, a pro-active approach to MLB is needed for MLB offloading taking into account interrelated factors such as interference, load, speed and including an enhancement to small cell discovery and user association.

MLB can make use of Cell Range Expansion (CRE), which is achieved by either cell coverage or mobility parameters adjustments. CRE increases the downlink coverage footprint of a low power cell by adding a positive bias value. Offloaded users may experience unfavourable channel from biased cells and strong interference from unbiased higher power cells. CRE forces alternate cell selection without considering loading or resource allocation in the corresponding cell. Advanced MLB makes use of CRE together with the Almost Blank Subframes (ABS) feature. ABS is a time domain technique, which improves the overall throughput of the off-loaded users by sacrificing the throughput of unbiased cells. Given an ABS ratio (i.e. a ratio of blank over total subframes), a user may select a cell with maximum ABS ratio. CRE and ABS are classified as distributed cell association schemes.

Re-association of a user to a cell other than the one offering the largest signal strength as is sometimes implemented by traditional MLB approaches described above, often leads to reduced desired signal level and an increase in interference level which results in an overall network performance degradation.

Multi cell load balancing in dense small cell deployments can help reduce blocking probability and improve network performance. Such action makes use of clustering of cells and Automatic Neighbour Relationship (ANR) which in turn ensures that resources are appropriately allocated to groups of similar cells and the frequency of invocation of other SON algorithms is reduced, thereby minimising conflicts.

Examples of MLB clustering and cell selection approaches are illustrated in Fig. 5.

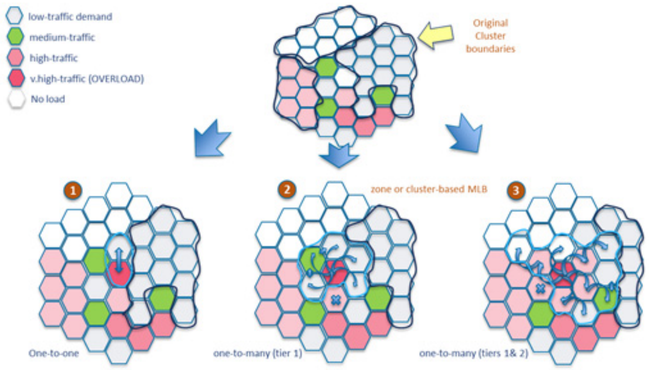

Fig. 5. Examples of clustering and cell selection approaches

The main modelling approaches for user association rely on utility modelling [11]-[15]. Examples of utility functions include spectrum efficiency, energy efficiency, QoS, outage and fairness. These approaches include game theory, combinatorial optimisation and stochastic geometry. Future work will concentrate on developing a new optimisation technique well suited to MLB in dense small cell deployments and taking into account multi-tenancy.

VI. Conclusions

This paper has addressed the development of Self-X functions for multi-tenant cloud enabled small cells scenarios. In these scenarios, the availability of a virtualized execution platform at the RAN facilitates additional capabilities for these Self-X functions. In this respect, after discussing the general SESAME architecture and how it incorporates the Self-X functionalities, the paper has elaborated on three different use cases. First, a dSON function for optimizing TCP performance during Handover has been presented. It is based on HO predictions used to set the TCP retransmission timeouts. Second, considering the specificities of the multi-tenant virtualized environment, the paper has presented a dynamic on-demand resource allocation model for assigning the small cell resources to the involved tenants. Finally, the paper has elaborated on the MLB functionality and how it can be enhanced through the joint use of ABS and CRE features.

Acknowledgment

This work has been supported by the EU funded H2020 5G-PPP project SESAME under the grant agreement no. 671596.

References

[1] Real Wireless Ltd “An assessment of the value of small cell services to operators. Based on Virgin Media trials”, October, 2012, http://www.realwireless.biz/small-cells-as-a-service-trials-report/

[2] ETSI GS NFV-MAN 001 (V1.1.1): "Network Function Virtualisation (NFV); Management and Orchestration".

[3] Yun Chao Huet al. “Mobile Edge Computing A key technology towards 5G”, ETSI White Paper No.11, September, 2015.

[4] Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME), http://www.sesame-h2020-5g-ppp.eu/

[5] 3GPP TS 23.251 v13.1.0, “Network Sharing; Architecture and functional description (Release 13)”, March, 2015.

[6] J. Ramiro, K. Hamied, *Self-Organizing Networks. Self-planning, self-optimization and self-healing for GSM, UMTS and LTE*, John Wiley & Sons, 2012.

[7] I. Giannoulakis (editor), “SESAME Final Architecture and PoC Assessment KPIs”, Deliverable D2.5 of SESAME project, December, 2016.

[8] Information technology - dynamic adaptive streaming over HTTP (DASH) - part 1: Media presentation description and segment formats, ISO/IEC MPEG standard, May 2014. ISO/IEC 23009-1:2014.

[9] Nguyen, Binh, et al. "Towards understanding TCP performance on LTE/EPC mobile networks." Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges. ACM, 2014.

[10] Izumikawa, Haruki, Ichiro Yamaguchi, and Jiro Katto. "An efficient TCP with explicit handover notification for mobile networks." Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE. Vol. 2. IEEE, 2004.

[11] D.Liu et al, “ User Association in 5G Networks: A Survey and an Outlook”, IEEE communications Surveys & Tutorials, Vol.18, No. 2, Second Quarter 2016

[12] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, “Energy efficient context-aware user association for outdoor small cell heterogeneous networks,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2014, pp. 1614–1619.

[13] S. Corroy, L. Falconetti, and R. Mathar, “Dynamic cell association for downlink sum rate maximization in multi-cell heterogeneous networks,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2012, pp. 2457–2461.

[14] H. Zhou, S. Mao, and P. Agrawal, “Approximation algorithms for cell association and scheduling in femtocell networks,” IEEE Trans. Emerging Topics Comput., vol. 3, no. 3, pp. 432–443, Sep. 2015.

[15] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, “Cell association and interference coordination in heterogeneous LTE-A cellular networks,” IEEE J. Sel. Areas Commun., vol. 28, no. 9, pp. 1479–1489, Dec. 2010.