Virtual-Mobile-Core Placement for Metro Network

Abhishek Gupta[*], Massimo Tornatore[*‡], Brigitte Jaumard[†], and Biswanath Mukherjee[*]

[*]University of California, Davis, USA [†]Concordia University, Canada [‡]Politecnico di Milano, Italy

Email: [*]{abgupta,mtornatore,bmukherjee}@ucdavis.edu [†]bjaumard@cse.concordia.ca [‡]massimo.tornatore@polimi.it

Abstract—Traditional highly-centralized mobile core networks (e.g., Evolved Packet Core (EPC)) need to be constantly upgraded both in their network functions and backhaul links, to meet increasing traffic demands. Network Function Virtualization (NFV) is being investigated as a potential cost-effective solution for this upgrade. A virtual mobile core (here, virtual EPC, vEPC) provides deployment flexibility and scalability while reducing costs, network-resource consumption and application delay. Moreover, a distributed deployment of vEPC is essential for emerging paradigms like Multi-Access Edge Computing (MEC). In this work, we show that significant reduction in network-resource consumption can be achieved as a result of optimal placement of vEPC functions in metro area. Further, we show that not all vEPC functions need to be distributed. In our study, for the first time, we account for vEPC interactions in both data and control planes (Non-Access Stratum (NAS) signaling procedure Service Chains (SCs) with application latency requirements) using a detailed mathematical model.

I. INTRODUCTION

Mobile network operators have to manage increasing traffic from bandwidth-intensive applications. This creates challenges for the Evolved Packet Core (EPC) (i.e., the current state-of-the-art mobile core network) where complex dependencies among various EPC elements lead to scalability issues as traffic demand increases and results in frequent upgrades of costly proprietary hardware.

Network Function Virtualization (NFV) virtualizes network functions (called Virtual Network Functions, VNFs) and runs them on commodity hardware. Also, EPC functions can be virtualized into VNFs to provide cost savings, flexibility, and ease of deployment. We refer to EPC implemented using VNFs as virtual EPC (vEPC). In order to understand possible VNF inter-dependencies and interactions in vEPC, we must discuss what a traditional EPC is and how it functions.

Evolved Packet Core (EPC) is an end-to-end IP-based mobile core network infrastructure. Fig. 1 shows a high-level functional view of an EPC, its functional entities and interfaces. Delineation between control and data planes is an important aspect of EPC and will be discussed further. Mobility Management Element (MME), Policy and Charging Rules Function (PCRF), and Home Subscriber Server (HSS) are EPC control plane elements. Serving Gateway (SGW) and Packet Data Network Gateway (PGW) are EPC data plane elements but also have tightly coupled control plane functions. In this work, we consider vEPC to have VNFs for MME, PCRF, HSS, SGW and PGW.

User Equipment (UE) connects to the Internet (or in general to an external network) through EPC. The data path (bearer)
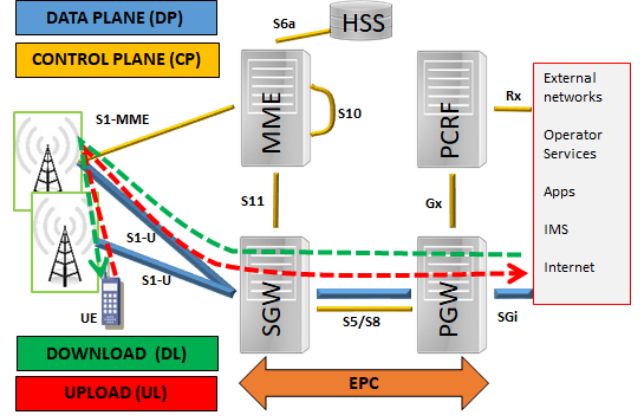

Fig. 1: Evolved Packet Core (EPC).

to the Internet for a UE is setup as a result of control plane signaling procedures called Non-Access Stratum (NAS) procedures. NAS is a layer-3 protocol for mobile networks, responsible for management and modification of bearers. The types of EPC functions involved in a NAS procedure depend on the UE signaling event. Thus, there are different NAS procedures, such as Attach, Dedicated bearer setup, X2-based handover, S1-based handover etc. as detailed in [1]. NAS procedures form a critical part of control plane signaling. A slight increase (e.g.,1%) in the control plane signaling can result in a significant decrease (70%) in data plane capacity [2]. This happens because SGW while handling all data plane traffic is also involved in 33% [2] of all control plane transactions. A highly-loaded SGW can become a bottleneck which affects data path throughput and increases control plane latency.

Figure 2(a) shows the NAS Attach Procedure, where all EPC functions are involved in control signaling. Here, signaling proceeds to next function only after it has been processed by current function. So, control plane signaling can be characterized as a chained sequence of interactions between EPC functions. We refer to this chained sequence of interactions in the control plane as a Control Service Chain (CSC)[1]. Fig. 2(b) shows the corresponding CSC for attach procedure.

NAS attach procedure completion results in a data path

---

[1]The term "Service Chain"(SC)[3] is usually used for value-added services (Firewall, Video Optimization etc.). We use "Service Chain" here since an ordered sequence of functions is similar to chained sequence of interactions between functions.

(a) Simplified NAS Attach Procedure[1]

(b) Attach Control Service Chain (CSC)

(c) Data Service Chains (DSCs)
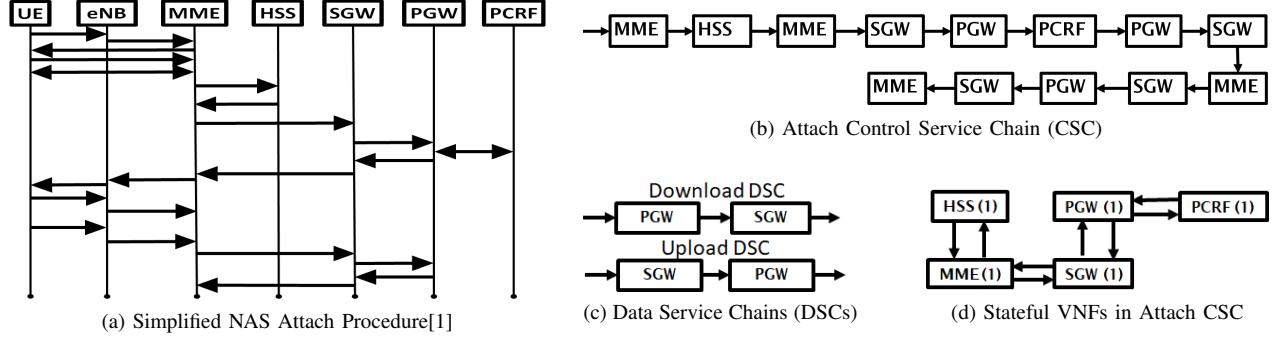
(d) Stateful VNFs in Attach CSC

Fig. 2: NAS Procedure, control and data Service Chains (SCs)

(bearer) for UE to Internet. Note that control-plane latency requirements for each NAS procedure depends on whether a default or dedicated bearer is setup. The UE upload traffic reaches eNodeB and is directed to SGW. SGW is the mobility anchor for eNodeB's, and hence, upload traffic has to traverse SGW first, then PGW to reach Internet as shown in Fig. 1. If a UE downloads data, SGW and PGW are traversed in reverse order. Fig. 2(c) shows the Data Service Chains (DSCs) for upload and download.

An important aspect of EPC functions is that they are stateful. They need to maintain session information, i.e., maintain information regarding UE and bearer state and internal database states, which might be different between instances of the same function. For example, there could be two VNF instances for MME, where the first MME (MME(1)) holds session information for a different set of UE's from the second MME (MME(2)). For NAS Attach procedure shown in Fig. 2(a) the EPC function interactions happen with the same instance of EPC functions as seen in Fig. 2(d) (1 indicating a single instance). The required number of instances of a function will depend on the NAS procedure. This is an important distinction between EPC SCs and traditional (e.g., value-added) SCs where VNFs can be stateless. In this work, we solve vEPC placement with stateful VNFs unlike previous works (e.g., our work in [4][5]) for traditional SCs. We account for stateful VNFs by tracking the VNF instance in our model.

In this paper, we propose an Integer Linear Program (ILP) to reduce bandwidth consumption in metro core while accounting for control and data plane interactions of UEs and satisfying application latency requirements through the optimal placement of vEPC function replicas. This model allows us to show that having distributed vEPC replicas reduces bandwidth consumption and not all vEPC functions need to be distributed.

The rest of this study is organized as follows. Section II describes our network architecture. Section III summarizes existing literature on vEPC placement problem and remarks on the novelty of this study. Section IV provides details on problem definition, input parameters and ILP model. Section V and VI details illustrative examples and conclusion.

II. NETWORK ARCHITECTURE

We assume a metro-core mesh topology connected to 2 aggregation rings to be our network architecture as shown in Fig. 4(a). Traffic from UEs is aggregated at Traffic Aggregation Points (TAPs). Aggregated UE traffic forms traffic flows that originate at a TAP and terminate at an application gateway or vice-versa. Application gateways in our architecture are peering points of the mobile core with other networks. For example, video traffic will go to application gateway that peers with the network that has the video content.

If all applications require traffic to be routed to mobile core to reach desired application gateway, network-resource consumption will increase with traffic demand. To avoid routing to mobile core, we need to host applications closest to the edge, i.e., in aggregation rings. So, in Fig. 4(a) we include two Multi-Access Edge Computing (MEC)[2] [6] nodes for hosting applications.

III. RELATED WORKS

Only few studies already exist for the vEPC-placement problem. Ref. [7] was the first to address the vEPC placement problem. Author's model vEPC functions as service chains but do not account for NAS procedures in control plane and upload and download in data plane, modeling service chaining explicitly, stateful nature of vEPC elements and application latency. Ref. [8] proposes combined network function and vEPC placement while accounting for vEPC function mapping to eNodeB. A mathematical model and heuristic are proposed, however this work also has same set of limitations as Ref. [7].

Other works, while not solving vEPC placement directly, provide useful insights. Ref. [2] emulates vEPC functions and shows that SGW is a bottleneck in EPC, and proposes a better functional design for vEPC. We utilize this insight in our model as stated in Section I. Ref. [9] demonstrates virtual SGW and PGW placement using a simulation framework, but does not provide details of the VNF-placement algorithm. Ref. [10] focuses on minimizing SGW relocations without considering control and data plane interactions of other EPC network functions.

[2]MEC is an emerging paradigm which aims to reduce network-resource consumption and improve application latency by deploying cloud-based IT services at network edge.
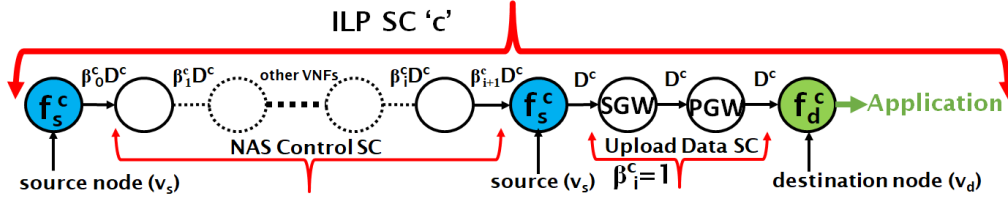
Fig. 3: Application flow requesting upload with NAS procedure

To the best of our knowledge, this is the first attempt to account for control and data plane interactions and application latency for optimal placement of vEPC replicas in a metro network.

IV. PROBLEM DESCRIPTION

A mobile network operator requires an EPC for connecting user equipment (UE) to Internet. To provide services, EPC network functions have to engage in control signaling (chained requests or Control SC (CSC)) for data path (Data SC (DSC)) setup. We develop a Integer Linear Programming (ILP) model for optimal placement of vEPC functions which accounts for latency requirements of applications, control and data plane interactions and statefulness of EPC VNFs, number of VNF replicas available and nodes allowed to host VNFs.

A. Problem Statement

Given a network topology, capacity of links, a set of network nodes with NFV support (NFV nodes), compute resources at NFV nodes, number of NFV nodes used, aggregated traffic flows using a NAS procedure and requesting an application, and latency requirements for application and control signaling, we determine the placement of vEPC VNFs and traffic routing to minimize network-resource (bandwidth) consumption.

B. Input Parameters

G	Physical topology of backbone network
	$G = (V, L)$ with V: node set and L: link set
$\mathcal{SD}$	Set of source-destination (v_s, v_d) pairs
V^{NFV}	$\subseteq V$ Set of nodes that can host VNFs (NFV nodes)
n^{CORE}	Number of CPU cores present in a NFV node
F	Set of VNFs, indexed by f
R_f	Maximum number of replicas of VNF f
n_f^{CORE}	Number of CPU cores per Gbps for function f
C	Set of chains, indexed by c
n_c	Number of VNFs in SC c
χ_f^c	Number of instances of VNF f required by c
$\mathcal{SD}_c$	Source-destination (v_s, v_d) pair for SC c
f_s^c	VNF for v_s in SC c
f_d^c	VNF for v_d in SC c
D^c	Traffic demand for SC c
$\sigma_i(c)$	ID of ith VNF in SC c,
	where $f_{\sigma_i(c)} \in F$
T_f^{ci}	VNF ID (f) of the ith VNF in SC c

$\theta_i(c)$	Instance used for ith VNF with ID f in SC c,
	where $\theta_i(c) \in \{1, 2, \ldots, \chi_f^c\}$
β_i^c	Fraction of D^c between i and $(i+1)$th VNF
	for SC c
$\Delta_\ell^{\text{PROG}}$	Propagation latency for link ℓ
Δ_f^{PROC}	Processing latency for function f
L_c	Latency requirement for SC c

C. Variables

x_{vi}^{cj}	1 if instance j of ith function f_i of c is located
	in $v \in V^{\text{NFV}}$ (responsible for stateful VNFs)
	1 if ith function $f_i \in \{f_s^c, f_d^c\}$ is located in $v \in \mathcal{SD}_c$
	0 otherwise
x_{vf}	1 if function f is located in $v \in V^{NFV}$; 0 otherwise
$y_{i\ell}^c$	1 if ℓ is on the path from location of f_i
	to location of f_{i+1}, 0 otherwise.

D. Problem Modeling

We consider aggregated traffic at Traffic Aggregation Points (TAPs) in G. For upload v_s is a TAP and v_d is an application gateway (opposite for download).

We consider that application flows request upload or download for an application and may or may not require a NAS procedure. To simplify formulation, we define a SC c in our ILP as consisting of (v_s, v_d) pairs, NAS Control SC, Upload/Download Data SC and application as shown in Fig. 3. Our ILP considers SC c as a sequence of VNFs. This also includes (v_s, v_d) which are source-destination nodes and are not VNFs. We create functions f_s^c and f_d^c to represent source v_s and destination v_d respectively. VNFs f_s^c and f_d^c have the only requirement of being located at v_s and v_d respectively.

Fig. 3 shows an application flow for upload with NAS procedure. Here, control traffic originates from VNF f_s^c (which represents source node v_s) and traverses NAS CSC. After the traversal, control traffic reaches f_s^c to notify that data path is setup. Since application traffic requests upload, the upload DSC is to be traversed on data path to destination (here, application gateway). Total latency requirement L_c for SC c is "control plane latency + application latency". When c has no NAS procedure, L_c equals application latency.

The Integer Linear Program (ILP) formulation of the vEPC placement problem is detailed in the following.

Objective: Minimize bandwidth consumed:

$$\sum_{c \in C} \sum_{\ell \in L} \sum_{i=1}^{n_c-1} D^c \beta_i^c y_{i\ell}^c \tag{1}$$

(a) Network topology [9]

(b) Bandwidth vs. number of vEPC replicas

(c) Bandwidth vs. number of VNF replicas

Fig. 4: Simulation topology and results

Eq.(1) gives total network-resource consumed to route SCs.

$$x_{vi}^{cj} = 1 \qquad c \in C, v \in \mathcal{SD}_c, j = \theta_i(c)$$
$$i \in \{1, 2, \ldots, n_c\} : f_i \in \{f_s^c, f_d^c\} \qquad (2)$$

$$x_{vi}^{cj} = 0 \qquad c \in C, v \in \mathcal{SD}_c : v \notin V^{\text{NFV}}, j = \theta_i(c),$$
$$i \in \{1, 2, \ldots, n_c\} : f_i \notin \{f_s^c, f_d^c\} \qquad (3)$$

$$\sum_{v \in V^{\text{NFV}}} x_{vi}^{cj} = 1 \quad c \in C, j = \theta_i(c),$$
$$i \in \{1, 2, \ldots, n_c\} : f_i \notin \{f_s^c, f_d^c\} \qquad (4)$$

Eq. (2) enforces VNFs (f_s^c, f_d^c) be placed on (v_s, v_d) respectively while Eq. (3) avoids placing vEPC VNFs on (v_s, v_d) if they are not NFV nodes. Eq. (4) ensures that all vEPC VNFs in c are placed at exactly one node.

The next constraints are flow conservation ones and are responsible for explicit service chaining.

$$\sum_{\ell \in \omega^+(v)} y_{i\ell}^c - \sum_{\ell \in \omega^-(v)} y_{i\ell}^c = x_{vi}^{cj} - x_{v,i+1}^{cj}$$
$$c \in C, v \in \{V^{\text{NFV}} \cup \mathcal{SD}_c\}, j = \theta_i(c),$$
$$i = 1, 2, \ldots, n_c - 1 \qquad (5)$$

$$\sum_{\ell \in \omega^+(v)} y_{i\ell}^c - \sum_{\ell \in \omega^-(v)} y_{i\ell}^c = 0$$
$$c \in C, v \in V \setminus \{V^{\text{NFV}} \cup \mathcal{SD}_c\},$$
$$i = 1, 2, \ldots, n_c - 1 \qquad (6)$$

$$Mx_{vf} \geq \sum_{c \in C : f \in c} \sum_{\substack{i \in \{1, 2, \ldots, n_c\}: \\ f_i = f, j = \theta_i(c)}} x_{vi}^{cj} \geq x_{vf}$$
$$v \in V^{\text{NFV}}, f \in F \qquad (7)$$

$$\sum_{v \in V^{\text{NFV}}} x_{vf} \leq R_f \qquad f \in F \qquad (8)$$

Eqs. (7) and (8) keep track of VNF replicas.

$$\sum_{c \in C} D^c \sum_{i=1, j=\theta_i(c)}^{n_c} \beta_{i-1}^c x_{vi}^{cj} \mathrm{T}_{f:f_i \notin \{f_s^c, f_d^c\}}^{ci} n_f^{\text{CORE}} \leq n^{\text{CORE}}$$
$$v \in V^{\text{NFV}} \qquad (9)$$

$$\sum_{c \in C} D^c \sum_{i=1}^{n_c - 1} \beta_i^c y_{i\ell}^c \leq \text{CAP}_\ell \qquad \ell \in L \qquad (10)$$

$$\sum_{\ell \in L} \sum_{i=1}^{n_c - 1} \Delta_l^{propag} y_{i\ell}^c + \sum_{i=1}^{n_c} \beta_{i-1}^c D^c \Delta_f^{proc} \mathrm{T}_{f:f_i \notin \{f_s^c, f_d^c\}}^{ci}$$
$$\leq L_c \qquad c \in C. \qquad (11)$$

Eq. (9) bounds total CPU resource consumption by VNFs at a node. Eq. (10) constrains total bandwidth consumed in routing SCs at a link. Eq. (11) enforces that the total latency incurred in propagation across links and processing by VNFs for each SC does not exceed latency bounds.

Application	Traffic %
Progressive video (buffered streaming)	71.19%
Video conferencing	4.56%
VoIP	1.50%
Media downloads	13.3%
Non-real-time application (web,email)	9.45%

TABLE I: Application traffic [11]

NAS procedure	Flows	Bearer	Latency (ms)
Attach	10	Default	500
Dedicated bearer request	45	Dedicated	250
X2-based handover	5	Default	500
S1-based handover	10	Default	500

TABLE II: Traffic flows with NAS procedure requirements [11]

V. Illustrative Numerical Examples

We run our simulations on the 19 node topology shown in Fig. 4(a). The application gateways shown in Fig. 4(a) are selected based on Table I and are shown in red, green nodes are TAPS and black nodes are switches. We add Multi-Access Edge Computing (MEC) sites at node 12 and 8 for progressive video. This decision is taken as progressive video has highest traffic load, which MEC can reduce. Table II shows the number of application traffic flows for NAS procedures (aggregated from 1000 to 5000 UEs). These traffic flows are associated with a upload or download DSC. We split the application traffic in Table I over the flows shown in Table II and 50 flows with no NAS procedure requirements (i.e., only upload/download). We consider the upload to download traffic ratio as 1:4 [12]. Latency requirements for applications are taken from [13] and for NAS procedures depends on bearer type, as seen in Table II.

Total traffic is 224 Gb [11]. All nodes are allowed to host VNFs. Each node is allocated 2400 CPUs. Link capacity is set to 60 Gbps. Control plane traffic is 5% of data plane traffic i.e. $B_i^c = 0.05$ for control plane signaling. For data path $B_i^c = 1.0$ since all traffic is data traffic. All links are considered optical fibers of length 50km. Processing latency is 132 μs per Gbps [14] for each VNF. All VNFs require 2 CPUs per Gbps of throughput [15].

We run 10 iterations of our optimization model and plot mean results. Fig. 4(b) shows reduction in bandwidth consumption as the number of allowed vEPC replicas (Eq. (8)) are increased. We find the largest reduction occurs from 1 to 2 vEPC replicas. This happens because vEPC gets distributed across aggregation rings which reduces route length to the core, thereby resulting in reduction of bandwidth consumption. It should be noted that MEC is a significant factor in bandwidth usage reduction since applications become available at the edge. vEPC without MEC would always require routing to the core and reduction in bandwidth consumption would not be significant. We find that as number of replicas are increased beyond 4, reduction is not significant as we reach almost optimal bandwidth consumption by 4 replicas. Having 2 vEPC replicas reasons to be the best tradeoff between bandwidth

reduction and the overhead of deploying more replicas and making more nodes NFV compatible.

Figure 4(c) shows bandwidth consumption for different number of VNF replicas. Here, all VNFs have 2 replicas unless specified on the Y-axis. This tells us that having 1 replica of MME, HSS and PCRF has a negligible effect on bandwidth consumption in comparison to having 2 replicas of each (All=2). Hence, we only need to deploy 2 replicas of PGW and SGW to achieve the same bandwidth reduction as 2 vEPC replicas in Fig. 4(b). This demonstrates that we do not need to replicate all VNFs to achieve bandwidth reduction.

VI. Conclusion

We introduce the problem of vEPC placement while accounting for VNF interactions in control plane and data plane and application latency. We develop an Integer Linear Program (ILP) for placement of EPC VNFs and route traffic along service chains. We demonstrate that there is reduction in network-resource consumption with increase in number of VNF replicas. Further, we show that not all EPC VNFs need to be replicated and that SGW and PGW replication gives almost the same network resource consumption as replicating all vEPC VNFs.

References

[1] 3GPP, "3GPP TS 24.301: 3GPP Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)." [Online]. Available: http://www.3gpp.org/DynaReport/24301.htm.

[2] A. S. Rajan et al., "Understanding the bottlenecks in virtualizing cellular core network functions," in *IEEE Workshop on LANMAN*, 2015.

[3] IETF, "Network service chaining problem statement," https://tools.ietf.org/html/draft-quinn-nsc-problem-statement-00, 2013.

[4] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, "Service Chain (SC) Mapping with Multiple SC Instances in a Wide Area Network," in *IEEE GLOBECOM 2017*, Dec 2017, pp. 1–6.

[5] A. Gupta, B. Jaumard, M. Tornatore and B. Mukherjee, "A Scalable Approach for Service Chain (SC) Mapping with Multiple SC Instances in a Wide-Area Network," *IEEE Journal on Selected Areas in Communications*, 2018.

[6] ETSI, "Mobile-Edge Computing - Introductory Technial White Paper." [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[7] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *IEEE NetSoft 2015*.

[8] D. Dietrich, C. Papagianni, P. Papadimitriou, and J. S. Baras, "Network function placement on virtualized cellular cores," in *COMSNETS 2017*.

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

[10] T. Taleb and A. Ksentini, "Gateway Relocation Avoidance-aware Network Function Placement in Carrier Cloud," in *Proceedings of the 16th ACM MSWiM '13*.

[11] B. H. et al., "High-performance evolved packet core signaling and bearer processing on general-purpose processors," *IEEE Network*, vol. 29, no. 3, pp. 6–14, 2015.

[12] ITU, "Report ITU-R M.2370-0: IMT traffic estimates for the years 2020 to 2030." [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2370-2015-PDF-E.pdf

[13] 3GPP, "3GPP TR23.401 V8.1.0: General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access."

[14] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," in *ACM Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges 2014*, pp. 33–38.

[15] Brocade, "Brocade vEPC." [Online]. Available: https://www.brocade.com/content/dam/common/documents/content-types/datasheet/brocade-vepc-ds.pdf