Network Function Placement on Virtualized Cellular Cores

David Dietrich[§], Chrysa Papagianni[†], Panagiotis Papadimitriou[§], and John S. Baras[†]

[§]Institute of Communications Technology, Leibniz Universität Hannover, Germany
{david.dietrich,panagiotis.papadimitriou}@ikt.uni-hannover.de
[†]Institute for Systems Research, University of Maryland, USA
{chrisap,baras}@isr.umd.edu

***Abstract*—The proliferation of mobiles devices, application sprawl, and the ever-increasing data volume generates significant stress on cellular networks and particularly on the cellular core, also known as the Evolved Packet Core (EPC). This is further exacerbated by the deployment of hardware appliances for the implementation of a wide range of network functions (*e.g.,* gateways, mobility management, firewalls, network address translation), hindering any opportunity for elastic provisioning, and eventually leading to high operational costs and a significant degree of load imbalance across the EPC.**

Network Function Virtualization (NFV) has been seen a promising solution in order to enable elasticity in the cellular core. Applying NFV to the EPC raises the need for network function (NF) placement, which in turn entails significant challenges, due to the stringent delay budgets among cellular core components and the coexistence of communicating data and control plane elements. To address these challenges, we present a linear programming (LP) formulation for the computation of NF placements that strikes a balance between optimality and time complexity. Our evaluation results show that the LP achieves significantly better load balancing, request acceptance rate, and resource utilization compared to a greedy algorithm that performs NF placement inline with carriers' common practice today.

I. Introduction

Cellular networks have been facing a significant growth both in terms of coverage and capacity in order to cope with increasing traffic volumes. The latter stems from the proliferation of mobile devices and the increasing application diversity. This trend is expected to continue in the future with the rise of Machine-to-Machine (M2M) communications [31] and Internet-of-Things (IoT). Control plane traffic is also expected to grow at more than 100% annually [6].

The ever-growing data volume raises the need for more elasticity in terms of network function (NF) deployment. In particular, the cellular core, *i.e,* the cellular network components residing between the radio access network and the Internet - also known as the Evolved Packet Core (EPC), provides data (*e.g,* Serving and Packet Data Network Gateways) and control plane functions (*e.g,* mobility management and signaling [10]). In the EPC, operators also tend to deploy middleboxes for packet inspection and network address translation (NAT) [34].

Network Function Virtualization (NFV) has been seen as a promising solution to cope with the increasing stress on the cellular core. NFV promotes the consolidation of NFs on platforms built of commodity servers components, deployed in virtualized network infrastructures (*i.e,* datacenters [DCs]) [1], [2], [3], [4]. As such, NFV provides a great opportunity for the reduction of investment and operational costs, as it obviates the need to acquire, deploy, and operate specialized equipment. Furthermore, it allows for elastic provisioning, which can lead to the rapid instantiation of new services and enhanced response to evolving demands via virtualized NF instance scale-out [21], [14]. In the EPC, NFV can mitigate the problem of load imbalance across the DCs, as operators tend to utilize middleboxes in DCs close to base stations [26].

Leveraging on NFV towards an elastic cellular core poses significant challenges in terms of NF placement. First, NF placement should be optimized jointly for load balancing and latency, since there are stringent delay budgets among communicating data and control plane elements, such as the eNodeB (eNB), the Serving Gateway (S-GW), the Packet Data Network Gateway (P-GW), and the Mobility Management Entity (MME). Second, NF placement should be scalable with a large number of User Equipment (UE) and DCs. This will allow for rapid NF placement decisions in reaction to sudden changes in the traffic load (*e.g.,* flash crowds).

Since there is full visibility across all DCs in the cellular core, we seek to provide a single-stage scalable solver for the EPC NF placement problem. To this end, we initially present a mixed-integer linear programming (MILP) formulation for the computation of near-optimal NF assignments onto the cellular core at a single stage. To reduce the time complexity of the MILP, we employ relaxation and rounding techniques, transforming the initial MILP into a linear program (LP) that trades a small degree of optimality for fast retrievable NF placements. Our evaluation results show that the proposed LP yields significant gains in terms of load balancing, request acceptance rate, and resource utilization compared to a greedy algorithm at which EPC elements are assigned to DCs in proximity to the eNB (*i.e.,* which is a common practice today).

The remainder of the paper is organized as follows. Sec-

978-1-5090-4250-0/17/$31.00 ©2017 IEEE

tion II describes the NF placement problem, while the corresponding request and network model are presented in Section III. In Section IV, we introduce our MILP formulation, its relaxed variant (LP), and a heuristic algorithm for the NF placement. Section V presents our evaluation results, whereas Section VI discusses related work. Finally, in Section VII, we highlight our conclusions.

II. Problem Description

In this section, we provide background on cellular networks and elaborate on the problem of NF placement on the EPC.

A. Mobile Core Background

An LTE cellular network comprises the Radio Access Network (RAN) and the cellular core, known as the EPC. The RAN mainly contains the base stations (termed as eNodeBs) which provide radio access to the UE. The RAN and EPC are typically connected via an optical transport network.

The EPC consists of a range of data and control plane elements, responsible for routing, session establishment, mobility management, and billing (among others). In the data plane, the S-GW acts as a mobility anchor, whereas the P-GW routes cellular traffic to the Internet. The S-GW interacts with the MME which, in turn, is responsible for session establishment and mobility management. After a UE has been attached to an eNB, the MME authenticates the UE and subsequently selects a S-GW. These steps are repeated every time the UE is switched from *idle* to *active* mode (*i.e.,* a UE is switched to *idle* mode after a period of inactivity). Finally, a data-path across the eNB, S-GW and P-GW is established with the setup of a tunnel using the GPRS tunnelling protocol (GTP). During handovers, the MME re-establishes the data-path between the S-GW and the new eNB.

In addition, carriers tend to deploy middleboxes on the EPC, such as NATs, firewalls, and proxies. Certain classes of cellular data traffic may be routed to other specialized middleboxes (*e.g.,* HTTP traffic may be forwarded to a web cache, while video traffic may have to traverse a transcoder). Currently, carriers handle cellular data traffic per traffic class, *e.g.,* data, voice, video (instead of per-flow), adhering to the 3GPP standard. All the aforementioned data and control plane functions are deployed using network appliances built of specialized hardware.

B. Problem Description

Following the recent trends on EPC virtualization, we consider the deployment of its main elements as virtualized NFs (vNFs) in DCs. This creates opportunities for elasticity in resource provisioning and better load balancing, avoiding traffic and processing overload at DCs close to base stations [26]. In this respect, we consider sequences of EPC vNFs expressed as service chains.

EPC virtualization essentially requires the placement of vNFs on servers and orchestration for service chaining, *i.e.,* routing cellular data traffic through a set of vNFs, as prescribed in the service chain. Service chaining in DCs has been addressed by recent work [27], [20], [7], so in this work we mainly focus on the NF placement problem.

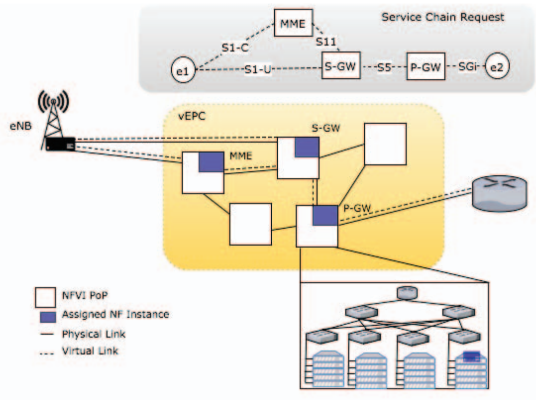

Fig. 1: Example of NF placement on the EPC.

In this respect, we consider a mobile operator's network, consisting of NFV Infrastructure (NFVI) and the RAN. The NFVI is composed of NFVI Points of Presence (PoPs). These could extend to the operator's WAN infrastructure, including local or regional PoPs for small or larger-scale NFVI deployments. The NFVI PoP is essentially a DC, therefore we consider a 2-level hierarchical network topology, although any common DC topology could be used for each site depending on the processing and bandwidth demands [19]. On each NFVI PoP, one or more NFs can be dynamically instantiated on demand for a requested service chain.

The problem at hand is to move the EPC's individual components (*i.e.,* MME, S/P-GW, middleboxes) that are traditionally deployed on specialized hardware to the operator's NFVI in order to support efficiently the operator's RAN, adhering to delay budgets between the individual control and data plane components. Therefore, the objective is to efficiently map the corresponding vNF forwarding graph(s), creating on demand an elastic EPC environment, optimized jointly for load balancing and latency.

In order to provide compatibility with 3GPP standards, specific constraints are taken into consideration for NF placement, *e.g.,* a single S-GW is attached to a UE at any point in time. We also assume that the traffic of a single eNB is routed to a single S-GW [26] and the UE is anchored to a single P-GW. In this context, an exemplary NF placement is further illustrated in Fig. 1. Such placements provide the number and location of NFVI PoPs that will provision the vNFs as well as the servers where these vNFs will be deployed, and the physical paths that data (*i.e.,* GTP) and control traffic will traverse.

Along these lines, NF placement on the EPC entails the following challenges:

Coordinated placement of data and control plane elements. NF placement has been recently tackled for the migration of middleboxes from enterprise networks to virtualized DCs [15], [24], [23], [18], [8]. However, proposed methods optimize the placement only of data plane functions for specific objectives, such as minimization of embedding footprint or load balancing. In contrast, EPC requires a coupling between data plane and control plane functions (*e.g.,* S-GW and MME). This has led to the specification of communication delay budgets between EPC elements [30]. These delay constraints should be taken into account in the NF placement, raising the need for NF assignments optimized jointly for load balancing and latency. In this respect, our NF placement methods (Section IV) fulfil the latency and resource requirements of the EPC elements.

Time complexity. The NF placement problem can be formulated as an integer linear program which yields high complexity and solver runtime, especially for a large number of UEs and DCs. KLEIN [26] copes with this complexity by decomposing the problem into region, DC, and server selection. This brings some benefits in terms of orchestration (*e.g.,* the server selection method can be invoked for intra-DC optimizations). Our work aims at global optimization of the NF placement, given the network-wide view on the cellular core. In this respect, we derive a LP formulation to reduce the time complexity.

III. Request and Network Model

In the following, we introduce models for the service chains and the cellular core network.

Request Model. We use a directed graph $G_F = (V_F, E_F)$ to express a service chain request. The set of vertices V_F include all virtualized EPC elements, such as S-GW, P-GW, MME, as well as any NFs (*e.g.,* NAT, firewall) that the traffic has to traverse. Each vertex in the graph is associated with a computing demand g^i, which we estimate based on the inbound traffic rate and the resource profile of the EPC element (*i.e.,* CPU cycles / packet). The edges are denoted by $(i,j) \in E_F$ while their bandwidth demands are expressed by g^{ij}. Each request is associated with a maximum delay $d_{(i\rightarrow j),max}$ over the virtual links eNB → MME, MME→S-GW, and S-GW→P-GW.

Network Model. We specify the cellular core network topology as an undirected graph $G_S = (V_S, E_S)$, where V_S represents the set of all nodes (*i.e.,* routers, servers, gateways, end-points). We further use $V_{servers} \subset V_S$ to explicitly express the servers in a DC. The delay incurred to a flow when assigned to a graph edge $(u,v) \in E_S$ is denoted by d_{uv}. Furthermore, nodes and links are associated with their residual capacity, denoted by r_u and r_{uv}, respectively. Their maximum capacity is given by $r_{u,max}$ and $r_{uv,max}$. A list of all notations is given in Table I.

IV. NF Placement Methods

In this section, we present our NF placement methods: (i) a MILP formulation for retrieving optimal mapping solutions, (ii) a scalable LP model that is used in conjunction with a rounding algorithm for retrieving near-optimal solutions in polynomial time, and (iii) a greedy algorithm as baseline.

TABLE I: Notations in the network model and the MILP/LP.

Symbol	Description
g^i	computing demand of NF i in *GHz*
g^{ij}	bandwidth demand of edge (i,j) in *Mbps*
$d_{(i\rightarrow j),max}$	maximum delay of the virtual link (i,j) in *ms*
d_{uv}	delay of the link (u,v) in *ms*
r_u	residual capacity of server u in *GHz*
$r_{u,max}$	maximum capacity of server u in *GHz*
r_{uv}	residual capacity of link (u,v) in *Mbps*
$r_{uv,max}$	maximum capacity of link (u,v) in *Mbps*
x_u^i	assignment of NF i to DC or server u
f_{uv}^{ij}	amount of bandwidth assigned to link (u,v) for NF graph edge (i,j) in *Mbps*
ε	helper variable in the MILP/LP objective function
γ_u^i	feasibility indicator of the mapping of NF i to server u
λ_{links}	link load balancing factor
$\lambda_{servers}$	server load balancing factor
Φ	link-to-node balancing factor in the MILP/LP objective function

A. MILP Formulation

In our MILP formulation, we use the binary variable x_u^i to express the assignment of NF i to the EPC node u. The real variable f_{uv}^{ij} expresses the amount of bandwidth assigned to link (u,v) for NF graph edge (i,j). The MILP is formulated as follows:

Minimize

$$\sum_{i\in V_F}\sum_{u\in V_S}\left(1-\frac{r_u}{r_{u,max}}\right) g^i x_u^i \gamma_u^i + \Phi \sum_{(i,j)\in E_F}\sum_{\substack{(u,v)\in E_S\\(u\neq v)}}\left(1-\frac{r_{uv}}{r_{uv,max}}+\varepsilon\right) f_{uv}^{ij} \quad (1)$$

subject to:

$$\sum_{u\in V_S} x_u^i = 1 \quad \forall i\in V_F \quad (2)$$

$$\sum_{\substack{v\in V_S\\(u\neq v)}} (f_{uv}^{ij} - f_{vu}^{ij}) = g^{ij}(x_u^i - x_u^j)$$
$$i\neq j, \forall (i,j)\in E_F, \forall u\in V_S \quad (3)$$

$$\sum_{i\in V_F} g^i x_u^i \leq r_u \quad \forall u\in V_S \quad (4)$$

$$\sum_{(i,j)\in E_F} f_{uv}^{ij} \leq r_{uv} \quad \forall (u,v)\in E_S \quad (5)$$

$$\sum_{(u,v)\in E_S} \frac{f_{uv}^{eNB,MME}}{g^{eNB,MME}}\, d_{uv} \leq d_{(eNB\rightarrow MME),max} \quad (6)$$

$$\sum_{(u,v)\in E_S} \frac{f_{uv}^{MME,SGW}}{g^{MME,SGW}}\, d_{uv} \leq d_{(MME\rightarrow SGW),max} \quad (7)$$

$$\sum_{\substack{(i,j)\in\\ \{\{SGW,NF_1\},\\ \{NF_1,NF_2\},\cdots\\ \{NF_n,PGW\}\}}} \sum_{(u,v)\in E_S} \frac{f_{uv}^{ij}}{g^{ij}}\, d_{uv} \ \le d_{(SGW\to PGW),max} \tag{8}$$

$$x_u^i \in \{0,1\} \quad \forall i \in V_F, \forall u \in V_S \tag{9}$$

$$f_{uv}^{ij} \ge 0 \quad \forall (u,v) \in E_S, \forall (i,j) \in E_F \tag{10}$$

The objective of the MILP is load balancing as expressed by the objective function (1). The first term of this function represents the amount of CPU resources multiplied by the utilization of each assigned server[1]. This term is minimized, if servers with lower utilization are preferred. Similarly, the second term of the objective function expresses the accumulated bandwidth assigned to EPC links multiplied by the corresponding link utilization. By minimizing the right-hand term, the number of assigned links is minimized while less loaded links are preferred. We further use a very small offset ε to avoid unnecessary use of zero-utilized links as they would otherwise result in multiplication by zero in the objective function[2]. In the first term, γ_u^i is used to avoid infeasible NF/server combinations. γ_u^i is infinite if the mapping $i \leftrightarrow u$ is already known to be infeasible, otherwise it is set to 1. For instance, we adjust the corresponding feasibility indicators $\gamma_{u\in V_S}^{i=SGW}$, $\gamma_{u\in V_S}^{i=MME}$ accordingly to assign a single instance of a S-GW and a MME per UE, as mandated by the 3GPP standard.

Furthermore, we introduce the link-to-node balancing factor Φ. $\Phi \gg 1$ yields solutions aiming at link load balancing while $\Phi \ll 1$ balances the load among the servers. We adjust Φ to strike a balance between node and link load balancing as follows:

$$\Phi = \frac{\lambda_{links}}{\lambda_{servers}} \cdot \frac{\sum\limits_{i\in V_F} g^i}{\sum\limits_{(i,j)\in E_F} g^{ij}} \tag{11}$$

$$\lambda_{servers} = \frac{max\,\{1 - \frac{r_u}{r_{u,max}} \mid u \in V_{servers}\}}{\frac{1}{|V_{servers}|} \sum\limits_{u\in V_{servers}} (1 - \frac{r_u}{r_{u,max}})} \tag{12}$$

$$\lambda_{links} = \frac{max\,\{1 - \frac{r_{uv}}{r_{uv,max}} \mid (u,v) \in E_S\}}{\frac{1}{|E_S|} \sum\limits_{(u,v)\in E_S} (1 - \frac{r_{uv}}{r_{uv,max}})} \tag{13}$$

Φ essentially depends on the current load balancing factors for the servers $\lambda_{servers}$ (12) and the links λ_{links} (13). The right-hand term of (11) is used for the normalization of CPU and bandwidth units.

Next, we explain the constraints of the MILP. Constraint (2) ensures that each NF $i \in V_F$ is mapped exactly to one server. Constraint (3) enforces flow conservation, *i.e.,* the sum of all inbound and outbound traffic in switches, routers, and servers that do not host NFs should be zero. More precisely, this condition ensures that for a given pair of assigned nodes i, j (*i.e.,* NFs or end-points), there is a path in the network graph where the edge (i, j) has been mapped. The constraints (4) and (5) ensure that the allocated computing and bandwidth resources do not exceed the residual capacities of servers and links, respectively. The constraints (6)–(8) ensure that the delays eNB→MME, MME→S-GW, and S-GW→P-GW do not exceed predefined bounds. Finally, the conditions (9) and (10) express the domain constraints for the variables x_u^i (binary) and f_{uv}^{ij} (real).

B. LP Relaxation and Rounding Algorithm

In the following, we describe a transformation of the above MILP to an LP model by relaxing the integer domain constraint of x_u^i:

$$x_u^i \in \{0,1\} \to x_u^i \ge 0 \quad \forall i \in V_F, \forall u \in V_S \tag{14}$$

The LP model can yield solutions with $x_u^i \notin \{0,1\}$ in which the boolean characteristic of x_u^i is not considered, thus constraints (2), (3), and (4) could be omitted. Therefore, we introduce an upper bound to the variables. The final domain constraints that replace (9) and (10) are as follows:

$$0 \le x_u^i \le 1 \quad \forall i \in V_F, \forall u \in V_S \tag{15}$$

$$0 \le f_{uv}^{ij} \le g^{ij} \quad \forall (u,v) \in E_S, \forall (i,j) \in E_F \tag{16}$$

The remaining solutions that contain any non-boolean x_u^i are processed by a rounding algorithm. In particular, the algorithm invokes a call to an LP solver and processes the set of feasible LP solutions iteratively. Each iteration includes the rounding of the current solution and either the acceptance or the rejection of the request. If the capacity and delay constraints still hold after the rounding, then the request is accepted; otherwise it is rejected and removed from the solution space. Algorithm 1 shows the pseudo code for the LP rounding.

Our tests[3] show that both MILP and LP lead to server and link load balancing[4] (Figs. 2 and 3). However, the optimality gap between MILP and LP is larger in terms of link load balancing, since our rounding approach results in the acceptance of requests with higher CPU demand and lower bandwidth demand compared to the requests accepted by the MILP. In particular, the LP generates in the long run 95% and 92% of the CPU and bandwidth revenue compared to the MILP. At the same time, the request acceptance rate of the LP is lower (Fig. 4). On the other hand, the LP is known to yield substantially lower time complexity. This is corroborated by our tests, *i.e.,* the solver runtime of the LP is up to two magnitudes lower than the MILP solver runtime (Fig. 5).

C. Greedy Algorithm

In addition, we have developed a greedy algorithm, which is shown in Algorithm 2. This algorithm assigns the NFs to the DC located most proximately to the eNB. In the case of lack of resources in proximate DCs, the algorithm

[1] The relative sever utilization is deducted from their residual capacities in the term $1 - \frac{r_u}{r_{u,max}}$. The same applies for the link utilization.

[2] We set $\varepsilon = 10^{-10}$ in our simulations.

[3] The tests were conducted on a 2 GHz AMD Opteron server (restricted to single core).

[4] See Section V-A for the definition of the load balancing level.

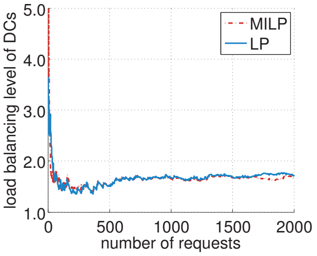

Fig. 2: DC load balancing level (based on server load).

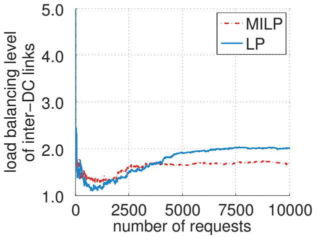

Fig. 3: Inter-DC link load balancing level.

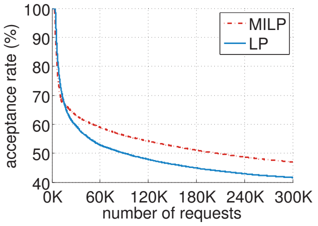

Fig. 4: Request acceptance rate.

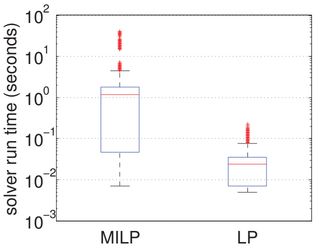

Fig. 5: Solver runtime.

Algorithm 1 NF placement with LP rounding

```
repeat
    $\{x_u^i, f_{uv}^{ij}\} \leftarrow$ Solve_LP(..)
    $FeasSol := true$ if solution for LP exists, $false$ otherwise
    $X \leftarrow \{x_u^i \mid x_u^i \notin \{0,1\}\}$
    if $X \neq \emptyset$ then
        $\{i_{fx}, u_{fx}\} \leftarrow argmax_{\{i \in V_F, u \in V_S\}} X$
        if $\left( \sum_{i \in \{V_F | x^i_{u_{fx}} = 1\}} g^i + g^{i_{fx}} \leq r_{u_{fx}} \right)$ and $\left( \sum_{(i,j) \in E_F} \sum_{(u,v) \in E_S} \frac{f_{uv}^{ij}}{g^{ij}} d_{uv} \leq d_{max} \right)$ then
            Add_LP_Constraint("$x_{u_{fx}}^{i_{fx}} = 1$")
        else
            Add_LP_Constraint("$x_{u_{fx}}^{i_{fx}} = 0$")
        end if
    end if
until $(X = \emptyset) \vee (FeasSol = false)$
if $FeasSol = true$ then
    return $\{x_u^i, f_{uv}^{ij}\}$ {Accept request}
else
    $\forall x_u^i := 0, \forall f_{uv}^{ij} := 0$
    return $\{x_u^i, f_{uv}^{ij}\}$ {Reject request}
end if
```

seeks placements on other DCs, subject to delay budgets and capacity constraints. For the mapping of NFs to the servers of each DC, the algorithm calls the routine *MapToDcServer*, which strives to co-locate the NFs in order to save link capacity and reduce delays among the assigned NFs. More specifically, the algorithm uses a list of servers of the DC, ordered by decreasing residual CPU capacity, and maps all the NFs to the first server. If the capacity of the first server is not sufficient, the remaining NFs will be mapped to the next servers in the list. Similar to the LP, the greedy algorithm allocates a single S-GW and MME per UE. We use the greedy algorithm as a baseline in our evaluation in Section V.

V. Evaluation

In this section, we assess the efficiency of our NF placement methods on virtualized EPC. To this end, we compare the efficiency between:

- The **LP** that aims at achieving load-balancing across the EPC,
- The **greedy** algorithm that maps NFs to the DC which is most proximate to the associated eNB, similar to what we consider a common practice today.

In the following, we discuss the evaluation environment (Section V-A) and the evaluation results (Section V-B).

A. Evaluation Environment

We have implemented an evaluation environment in C/C++, including a service chain generator and a cellular core network topology generator. We use CPLEX for our MILP/LP models. In the following we provide further details about our evaluation setup.

Cellular Core Network. We have generated a PoP-level cellular core network topology, spanning 10 homogeneous NFVI PoPs. Each PoP is essentially a micro-DC with a two-level fat-tree network topology. Table II shows additional cellular core network parameters.

UEs. We simulate an area with a varying user density ($\rho = U[385, 2308] UEs/km^2$), so that the number of active UEs per eNB ranges from 500 to 3000 (Table III). The total number of eNBs is 5000. Considering uniform circular cells with an overlapping factor γ of 1.2, the required cell radius is $r = \gamma\sqrt{A_t/C\pi}$.

Traffic Classes. Based on 3GPP, traffic is classified into three types, *i.e.,* voice, media streaming, and background traffic, with their busy-hour parameters shown in Table IV [16]. $Pr\{O\}$ is the probability that a session of a specific application type is originated by a UE.

Service Chains. We generate vNF-forwarding graphs per traffic class based on service chain templates. In particular, each service chain contains the main EPC elements (*i.e.,* S/P-GW, MME) and a set of security and application-specific NFs depending on the traffic class (see table IV). We derive the CPU demands for each NF from resource profiles, similar to [18]. Based on the session parameters of Table IV, we generate service chain requests that express a periodic update of active sessions (UEs). Upon its generation, each service chain request is embedded replacing the existing chain that handles the traffic of the same class.

Delay Budgets. The delay budgets among the communicating EPC elements (*i.e.,* eNB-MME, MME-SGW, SGW-PGW) are set to 50 ms, inline with [30].

Algorithm 2 NF Placement - Greedy Algorithm (Baseline)

```
$DCs \leftarrow$DCs, ordered by distance from eNB
$NrByDCs \leftarrow DCs$ with $d_{(eNB \to DC)} \leq d_{(eNB \to DC),max}$
if $SGW$ and $MME$ exist already for the UE group then
    MapToServer ($SGW$, currently used server)
    MapToServer ($MME$, currently used server)
else
    $DC :=$ first DC in $NrByDCs$
    while $SGW$ is not mapped to any server do
        if MapToDcServer ($SGW$, $DC$) not successful then
            $DC :=$ next DC in $NearByDCs$
            if $DC = \emptyset$ then
                return $\forall \{x_u^i, f_{uv}^{ij}\} := 0$ {Reject request}
            end if
        end if
        Update $x_u^{SGW}$
    end while
    while $MME$ is not mapped to any server do
        if MapToDcServer ($MME$, $DC$) not successful then
            $DC :=$ next DC in $NearByDCs$
            if $DC = \emptyset$ then
                return $\forall \{x_u^i, f_{uv}^{ij}\} := 0$ {Reject request}
            end if
        end if
        Update $x_u^{MME}$
    end while
end if
Compute least-delay paths:
-from $eNB$ to $Server(MME)$ and $Server(SGW)$
-and from $Server(MME)$ to $Server(SGW)$
if $d_{(eNB \to MME)} \leq d_{(eNB \to MME),max}$ and $d_{(MME \to SGW)} \leq d_{(MME \to SGW),max}$ then
    Update $f_{uv}^{ij}$
else
    return $\forall \{x_u^i, f_{uv}^{ij}\} := 0$ {Reject request}
end if
Reorder $DCs$ such that $DC(1) = DC$ of $SGW$
$DC :=$ first DC in $DCs$
for $NFi = \{NF_1, NF_2, \cdots, NF_n, PGW\}$ do
    $prevDC := DC$
    while MapToDcServer ($NFi$, $DC$) not successful do
        $DC :=$ next DC in $DCs$
    end while
    if $DC = \emptyset$ then
        return $\forall \{x_u^i, f_{uv}^{ij}\} := 0$ {Reject request}
    else
        Update $x_u^i$
    end if
end for
Compute least-delay path from $SGW$ to $PGW$
if $d_{(SGW \to PGW)} \leq d_{(SGW \to PGW),max}$ then
    Update $f_{uv}^{ij}$
    return $\{x_u^i, f_{uv}^{ij}\}$ {Accept request}
else
    return $\forall \{x_u^i, f_{uv}^{ij}\} := 0$ {Reject request}
end if
```

Signaling Load. Signaling procedures in LTE allow the control plane to manage the data flow between the UE and the P-GW, as well as UE mobility. Each procedure implies processing and exchange of signaling messages between the control plane entities. The most signaling load is considered to be generated by the *Service Request* and *X2 handover* [22], [25]. In terms of signaling load on the S-GW, *Attach* and *S1 handover* procedures are the most expensive [5], [28].

We quantify the processing load and the uplink/downlink traffic generated by LTE/EPC data management procedures, using the aforementioned traffic profile based on the analysis provided in [16], [17] and 3GPP LTE/EPC data plane management messages and their sizes [29]. Therefore, applications are modelled as ON-OFF state machines, while we assume that each UE is registered in the LTE/EPC network (EMM-registered) and alternates between Connected (ECM-/RRC-Connected) and Idle (ECM-/RRC-Idle) states. In other words, only *Service Request/Release* procedures are taken into consideration. The RRC inactivity timer defines the inactivity period required for the UE to switch to IDLE state. This timer is set to 10 sec.

We use the following metrics for the evaluation of NF placement efficiency:

- **Load Balancing Level (LBL)** is defined as the maximum over the average load. We report the LBL for DCs (based on server load) and for inter-DC links. Lower values of LBL represent better load balancing, while $LBL = 1$ designates optimal load balancing.
- **Request Acceptance Rate** is the number of successfully embedded service chain requests over the total number of requests.
- **Revenue per Request** is the amount of CPU and bandwidth units specified in the request.
- **Resource Utilization** is the amount of CPU and bandwidth units allocated for the embedded requests.

B. Evaluation Results

Initially, we discuss the load balancing in the cellular core, which is the main objective of our LP. Figs. 6 and 7 show the load balancing level among the DCs (based on server load) and the inter-DC links, respectively. The LP achieves a significant improvement in the DC load balancing level (Fig. 6) compared to the baseline which corresponds to the common practice today. Across inter-DC links, both the LP and the greedy algorithm yield an equally high level of load balancing (Fig. 7).

Fig. 8 illustrates the request acceptance rate of the LP and the greedy algorithm. The optimized NF placement of the LP leads to notable gains in terms of acceptance rate. More precisely, at steady state the LP variant accepts 11% more requests which are further associated with higher resource demands, *i.e.,* 12% more CPU and 13% more bandwidth demand per request relatively to the revenue generated by the

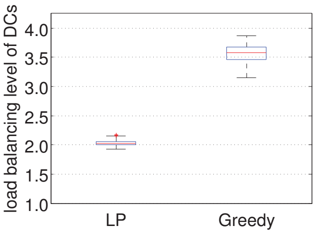

Fig. 6: DC load balancing level (based on server load).

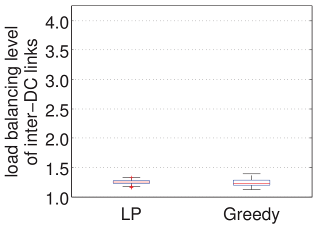

Fig. 7: Inter-DC link load balancing level.

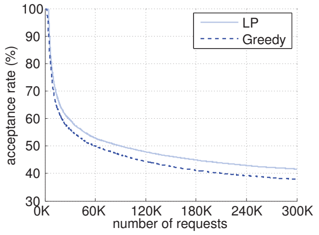

Fig. 8: Request acceptance rate.

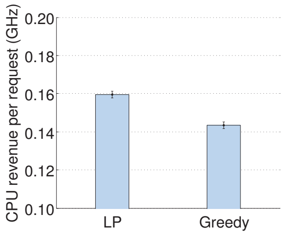

Fig. 9: Revenue from CPU per request.

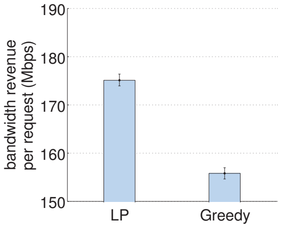

Fig. 10: Revenue from bandwidth per request.

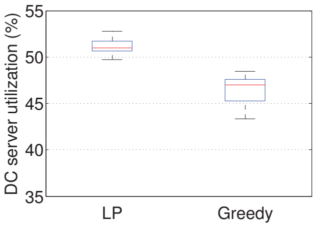

Fig. 11: Server utilization.

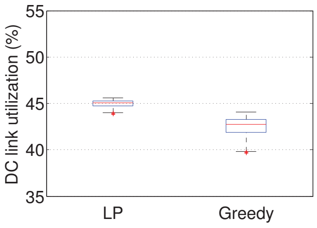

Fig. 12: Intra-DC link utilization.

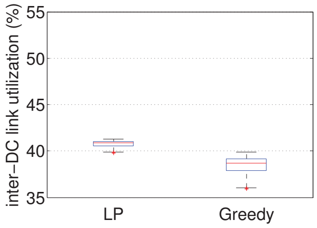

Fig. 13: Inter-DC link utilization.

TABLE II: Cellular Core Network Parameters

NFVI PoPs	10
Servers per DC	20 in 2 racks
Server Capacity	16 · 2 GHz
ToR-to-Server link capacity	4 Gbps
Inter-rack link capacity	16 Gbps
Inter-DC link capacity	100 Gbps

TABLE III: User Modeling Parameters

Area Size (A_t)	4500 km^2
Total Number of eNBs in the area (C)	5000
Active UEs per eNB	500 . . . 3000

TABLE IV: Session Parameters

Application Type (and NFs)	Arrival Rate (1/hour)	Duration (seconds)	Nominal Rate (Kbps)	Pr(0)
Voice (FW, NAT, Echo cancellation)	0.67	180	12.65	0.5
Streaming (FW, NAT, Transcoder)	5	180	256	1
Background traffic (FW, NAT)	40	10	550	0.8

baseline (Figs. 9–10). A high request acceptance rate is crucial for a carrier, since he can increase his revenue by fulfilling QoS requirements of a larger number of UEs. The ability to accommodate and process larger volumes of data traffic can also lead to higher revenues for carriers that lease network slices to Mobile Virtual Network Operators (MVNO).

Since a fraction of requests are rejected even with the LP (Fig. 8), we investigate the potential reasons that lead to these rejections. Our logs indicate that delay budgets and 3GPP's requirement of a single instance of S-GW and MME per UE rarely lead to rejections; instead, they merely restrict the solution space. In fact, the main reason for the request rejections is the inability to meet CPU or bandwidth requirements within highly utilized DCs.

Figs. 11 – 13 depict the utilization level of the DC servers, the intra-DC links, and the inter-DC links, respectively. The higher utilization levels achieved by the LP stem from the higher request acceptance rate (Fig. 8). Essentially, our optimized NF placement allows a carrier to utilize his resources more effectively accommodating larger volumes of traffic. Furthermore, in the case of cellular network slicing and leasing, the carrier will be able to monetize much more efficiently his infrastructure.

VI. Related Work

In this section, we briefly discuss related work on NF placement for virtualized EPC. NF placement has been tackled for the migration of LTE mobile core gateways (S-GW and/or P-GW) to DCs [35], [33], [9]. In the same direction, additional approaches to the problem [13], [32] take into consideration data-plane delay constraints. However, the proposed methods optimize the placement only of data-plane functions for design or operational objectives (*e.g.,* minimizing the EPC resource provisioning cost or GW relocations, load balancing).

Recently, NF placement of EPC control plane components (*e.g.,* MME, HSS) along S/P-GWs has been also considered towards a 3GPP-compliant elastic cellular core [11], [26]. Moreover, in a real-world deployment latency bounds for control traffic should be also considered, especially in the case that MME and mobile core gateways are not co-located. In the case of the KLEIN architecture [26], control and data plane functions are placed in three steps, taking the delay budgets between MME and S/P-GW into account. Authors in [12] propose a MILP formulation for the joint embedding of 3GPP-compliant core network service chains, considering end-to-end data- and control-plane latency bounds. However, they also decompose the EPC network graph into a data-plane chain and several control-plane ones, where delay bounds depend solely on the chain type (control or data). Furthermore, authors in [12] consider eNB clusters feeding these core service chains in order to deal with scalability issues, as opposed to KLEIN [26] and our work.

In our proposed solution (i) we investigate the embedding of service chains, containing both data and control-plane EPC elements as well as service-specific NFs and (ii) we consider delay budgets among individual EPC components, based on LTE design and deployment strategies provided by vendors. Our main aim is global optimization, therefore we derive the NF placement in a single step given the fact that virtual EPC providers (potentially telecom operators) will have a network-wide view on the cellular core.

VII. Conclusions

In this paper, we tackled the challenging problem of NF placement onto the cellular core. In this respect, we introduced a MILP and its relaxed variant for NF placement optimization, subject to capacity constraints, delay budgets between EPC components, and 3GPP-related restrictions. We further presented a greedy algorithm that strives to map NFs proximately to eNBs, inline with carriers' common practice.

We set up a realistic evaluation environment after a careful inspection of a wide range of cellular core network settings as well as signaling load and UE session models. According to our evaluation results, the proposed LP mitigates the load imbalance problem in today's cellular networks, spreading the load more evenly across the EPC's DCs, while maintaining compliance with the 3GPP standard. This leads to notable gains in terms of request acceptance and resource utilization, enabling the carrier to better monetize his infrastructure. Compared to the MILP, the LP exhibits substantially lower time complexity and solver runtime. As such, the LP can enable reprovisioning at lower timescales and thus better responses to traffic load variations. A small penalty is paid by the LP in terms of inter-DC link load balancing, whereas the DC load balancing level is similar for both variants.

VIII. Acknowledgments

This work was partially supported by the EU FP7 T-NOVA Project (619520).

References

[1] ETSI Network Function Virtualization, http://www.etsi.org/technologies-clusters/technologies/nfv

[2] OPNFV, https://www.opnfv.org/

[3] T-NOVA Project, http://www.t-nova.eu/

[4] UNIFY Project, http://www.fp7-unify.eu/

[5] 3GPP TS 24.301: 3GPP Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS). http://www.3gpp.org/DynaReport/24301.htm

[6] Business Case for Juniper Networks Virtualized Mobile Control Gateway, White Paper, Juniper 2013.

[7] A. Abujoda, H. R. Kouchaksaraei, and P. Papadimitriou, SDN-Based Source Routing for Scalable Service Chaining in Datacenters, IFIP WWIC, Thessaloniki, Greece, May 2016

[8] A. Abujoda and P. Papadimitriou, MIDAS: Middlebox Discovery and Selection for On-Path Flow Processing, IEEE COMSNETS, Bangalore, India, January 2015.

[9] M. Bagaa, T. Taleb, and A. Ksentini, Service-Aware Network Function Placement for Efficient Traffic Handling in Carrier Cloud, IEEE WCNC, Istanbul, Turkey, April 2014.

[10] A. Banerjee et al., Scaling the LTE Control-Plane for Future Mobile Access, ACM CONEXT, Heidelberg, Germany, December 2015.

[11] A. Baumgartner, V.S. Reddy, and T., Bauschert, Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization, IEEE NetSoft 2015, London, UK, April 2015.

[12] A. Baumgartner, V.S. Reddy, and T. Bauschert, Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds, EWSDN 2015, Bilbao, Spain, September 2015.

[13] A. Basta et al., Applying NFV and SDN to LTE mobile core gateways, the functions placement problem, 4th Workshop on All things cellular, ACM SIGCOMM 2014, Chicago, US, August 2014.

[14] Z. Cao, A. Abujoda, and P. Papadimitriou, Distributed Data Deluge (D3): Efficient State Management for Virtualized Network Functions, IEEE INFOCOM SWFAN, San Francisco, USA, April 2016

[15] R. Cohen et al., Near Optimal Placement of Virtual Network Functions, IEEE INFOCOM, Hong Kong, China, April 2015.

[16] W. Diego, I. Hamchaoui, and X. Lagrange, The Cost of QoS in LTE/EPC Mobile Networks Evaluation of Processing Load, IEEE VTC, Boston, MA, USA, 2015.

[17] W. Diego, I. Hamchaoui, and X. Lagrange, Cost factor analysis of QoS in LTE/EPC mobile networks, IEEE CCNC, Las Vegas, USA, Jan. 2016.

[18] D. Dietrich, A. Abujoda, and P. Papadimitriou, Network Service Embedding Across Multiple Providers with Nestor, IFIP Networking, Toulouse, France, May 2015.

[19] M. Faizul Bari et al., Data center network virtualization: A survey, IEEE Communications Surveys and Tutorials, 15(2), 2013, pp. 909-928.

[20] S. Fayazbakhsh et al., Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions using FlowTags, USENIX NSDI 2014, Seattle, USA, April 2014.

[21] A. Gember-Jacobson et al., OpenNF: Enabling Innovation in Network Function Control, ACM SIGCOMM 2014, Chicago, USA, August 2014.

[22] B. Hirschman et al. High-performance evolved packet core signaling and bearer processing on general-purpose processors, IEEE Network, 29(3), pp. 6-14, 2015.

[23] T. Lukovszki and S. Schmid, Online admission control and embedding of service chains, SIROCCO, Montserrat, Spain, July 2015.

[24] S. Mehraghdam, M. Keller, and H. Karl, Specifying and placing chains of virtual network functions, IEEE CloudNet, Luxembourg, Oct. 2014.

[25] J. Prados-Garzon et al., Latency evaluation of a virtualized MME, IEEE Wireless Days, Toulouse, France, March 2016

[26] Z. Qazi et al., KLEIN: A Minimally Disruptive Design for an Elastic Cellular Core, ACM SOSR 2016, Santa Clara, USA, March 2016.

[27] Z. Qazi et al, SIMPLE-fying middlebox policy enforcement using SDN, ACM SIGCOMM 2013, Hong Kong, China, August 2013.

[28] A.S. Rajan et al., Understanding the bottlenecks in virtualizing cellular core network functions, IEEE LANMAN, Beijing, China, April 2015.

[29] M. R. Sama et al, Enabling network programmability in LTE/EPC architecture using OpenFlow, WiOpt 2014, Hammamet, Tunisia, May 2014.

[30] Z. Savic, LTE Design and Deployment Strategies - CISCO. http://tinyurl.com/lj2erpg.

[31] M. Z. Shafiq et al., A First look at cellular machine-to-machine traffic: Large scale measurement and characterization, ACM SIGMETRICS, London, UK, June 2012.

[32] T. Taleb, M. Bagaa, and A. Ksentini, User mobility-aware virtual network function placement for virtual 5G network infrastructure, IEEE ICC 2025, London, UK, June 2015.

[33] T. Taleb and A. Ksentini, Gateway relocation avoidance-aware network function placement in carrier cloud, ACM MSWiM, Barcelona, Spain, November 2013.

[34] Z. Wang et al., An Untold Story of Middleboxes in Cellular Networks, ACM SIGCOMM 2011, Toronto, Canada, August 2011.

[35] F. Yousaf et al., SoftEPCDynamic instantiation of mobile core network entities for efficient resource utilization, IEEE ICC, Budapest, Hungary, June 2013.