Near-Optimal Placement of Virtualized EPC Functions with Latency Bounds

David Dietrich[1(✉)], Chrysa Papagianni[2], Panagiotis Papadimitriou[3], and John S. Baras[2]

[1] Institute of Communications Technology, Leibniz Universität Hannover, Hanover, Germany
david.dietrich@ikt.uni-hannover.de
[2] Institute for Systems Research, University of Maryland, College Park, USA
{chrisap,baras}@isr.umd.edu
[3] Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
papadimitriou@uom.edu.gr

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), *i.e.,* the cellular network component residing between the radio access network and the Internet. 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.

1 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 [33] and Internet-of-Things (IoT). Control plane traffic is also expected to grow at more than 100% annually [7].

N. Sastry and S. Chakraborty (Eds.): COMSNETS 2017, LNCS 10340, pp. 200–222, 2017.
DOI: 10.1007/978-3-319-67235-9_13

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 [11]). In the EPC, operators also tend to deploy middleboxes for packet inspection and network address translation (NAT) [36]. In fact, the middlebox diversity tends to increase along with the number of offered services and the pressing need for faster service deployment.

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]) [1–5]. 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 on clients' premises, either by introducing new functionality in the network or by scaling existing network services. Besides cost reduction, NFV 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 [15,23]. 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 [28].

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). In this respect, KLEIN [28] decomposes the NF placement problem into region selection, DC selection and server selection within the assigned DC to reduce the problem complexity.

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). In our evaluation environment, we carefully inspected and took into consideration both

data plane and signalling traffic to account for all CPU and bandwidth required for NF placement.

This paper extends our previous work in [20], by providing additional evaluation results, a more elaborate problem description, further details on the EPC signaling model, as well as a more extensive related work discussion. The remainder of the paper is organized as follows. Section 2 describes the NF placement problem, while the corresponding request and network model are presented in Sect. 3. In Sect. 4, we introduce our MILP formulation, its relaxed variant (LP), and a heuristic algorithm for the NF placement. Section 5 presents our evaluation results and discusses the efficiency of the proposed NF placement methods. In Sect. 6 we provide an overview of related work, and finally in Sect. 7, we highlight our conclusions and discuss directions for future work.

2 Problem Description

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

2.1 Cellular Core Background

Overview. An LTE cellular network comprises the Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the cellular core, known as the Evolved Packet Core (EPC). The E-UTRAN mainly contains the base stations, termed as eNodeBs (eNBs), which provide radio access to the UEs. 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 user plane, S-GW and P-GW are used for data forwarding. 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 UE authentication and authorization, session establishment and mobility management. The eNBs are connected to MME and SGW by means of S1-MME and S1-U interfaces, respectively. The S-GW supports the S11 interface with the MME and S5/S8 interface with P-GW. The eNBs are also interconnected with each other via the X2 interface, mainly used for inter-eNB handover. The QoS level for each transmission path (termed as EPS bearer) between the UE and the P-GW is decided by the P-GW. When a UE is attached to the network, a default bearer is established supporting best-effort QoS. Additional bearers can be set up with different QoS levels. The EPS bearer is made up of the radio data bearer (*i.e.,* between UE and eNB), the S1 data bearer (*i.e.,* between eNB and S-GW) and the S5 data bearer (*i.e.,* between the SG-W and the P-GW). The GPRS tunneling protocol (GTP) is used for setting up the user plane data-paths between the eNB, S-GW and P-G. During handovers, the MME re-establishes the data-path between the S-GW and the new eNB.

LTE-EPC Signaling. 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. However, significant signaling load is considered to be generated by the *Service Request* and *X2 handover* [24,27]. In terms of signaling load on the S-GW, *Attach* and *S1 handover* procedures are the most expensive [6,30]. In our study, we are considering only the costly *Service Request* procedure related to data plane management and consequently also *Service Release*, both of which are explained hereafter.

At the highest layer of the Control Plane protocol stack (Non-Access Stratum - NAS) two signaling protocols are used between the UE and the MME; the EPS Mobility Management (EMM) protocol and the EPS Connection Management (ECM) protocol. The EMM is responsible for handling UE mobility, supporting functions for attaching/detaching the UE from the network and performing location updates in between (tracking area update). The ECM is used to handle signaling connections between the UE and the EPC.

Once a UE is registered/attached to the network (*EMM-REGISTERED*), it can be either in *ECM-CONNECTED* or *ECM-IDLE* state. In the *ECM-IDLE* state, the UE has no radio (Radio Resource Control–RRC) connection to the eNB or S1 connections to the EPC. If, at this time, new traffic is generated from the UE, or from the network to the UE, the UE moves to the *ECM-CONNECTED* state, where radio and S1 signaling connections are established. Following the service request, the radio and S1 bearers are established at the user-plane allowing the UE to receive or send traffic. Service requests can be triggered by a UE or by the network (UE-originated or Network-originated). Service Release is triggered by the eNB due to detected UE inactivity or UE-generated signaling connection release.

As shown in Fig. 1, when the UE has new traffic to send, or learns about the network's intent to send new traffic, it establishes an RRC connection and sends a *Service Request* to the eNB. The *Service Request* is forwarded to the MME from the eNB, through an *Initial UE Message*, leading to S1 connection establishment. We skip UE authentication initiated by the MME and NAS security setup between the UE and MME, as they are optional when UE context exists in the network. Upon receiving the *Service Request*, the MME sends an *Initial Context Setup Request* to the eNB that leads to setting up the data radio bearer with the UE and the SI bearer, leading to end-to-end user-plane traffic paths for the UL. Following the *Initial Context Setup Response*, exchanging the *Modify Bearer* messages, between the MME and SGW, leads to the DL S1 bearer setup. If the UE's cell or tracking area has been changed at the time of the request, the UL/DL S5 bearers are modified.

2.2 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 [28]. In this respect, we

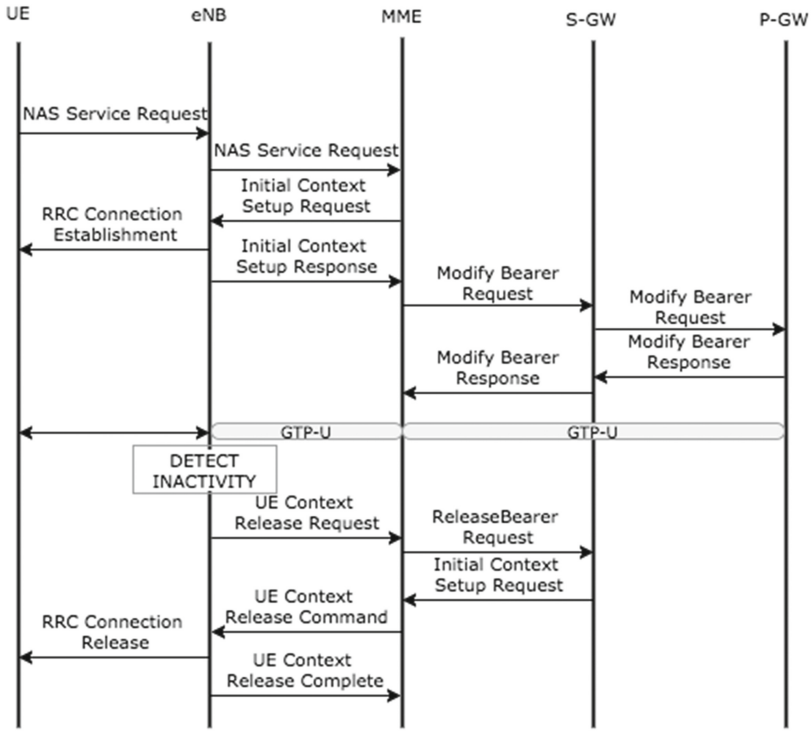

Fig. 1. Service Request/Release workflows.

consider sequences of EPC vNFs expressed as service chains. For instance, Fig. 2 illustrates a service chain consisting of datapath elements and a MME.

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 [8,22,29], so in this work we mainly focus on the NF placement problem.

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), where EPC elements can be deployed as vNFs. 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 [21]. 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 standard, 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 [28] and the UE is anchored to a single P-GW, leading to the service chain(s) per eNB depicted in Fig. 2. In this context, an exemplary NF placement is further illustrated in Fig. 3 (*e.g.,* P-GWs belonging to two different chains are assigned to the two distinct servers of an NFVI PoP). 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.

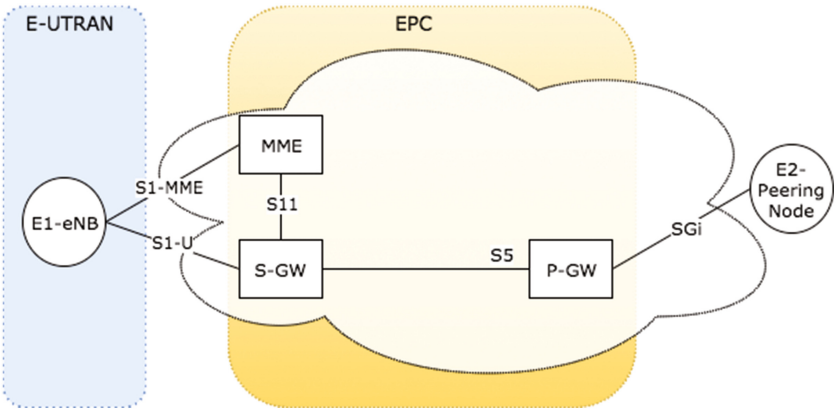

Fig. 2. EPC service chain.

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 [9,16,19,25,26]. 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 [32]. 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 (Sect. 4) fulfil the latency and resource requirements of the EPC elements.

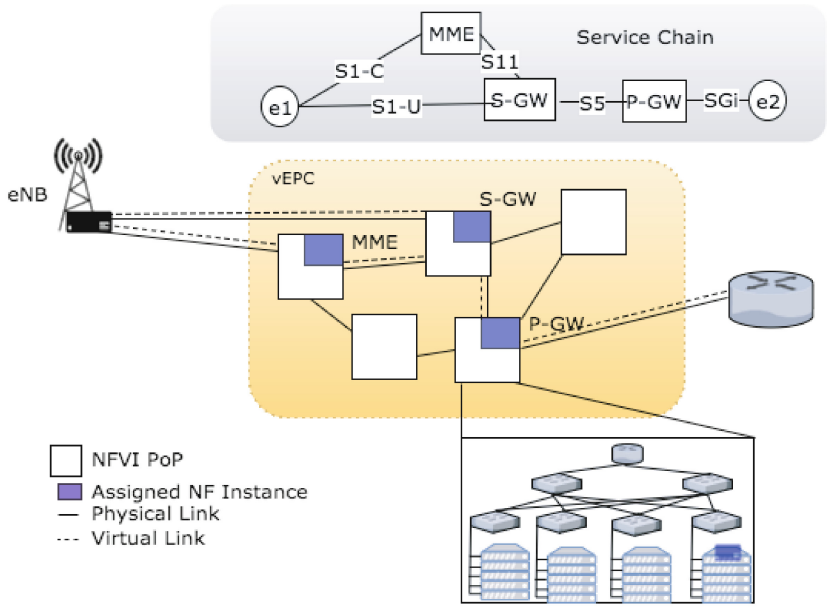

Fig. 3. Example of NF placement on the EPC.

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 [28] 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.

3 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 $d_{(i \to j),max}$ 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 1.

Table 1. 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 $Mbps$
$d_{(i \to j),max}$	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

4 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.

4.1 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

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

subject to:

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

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

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

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

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

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

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

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

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

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

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

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, the input variable 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 $\gamma_{u \in V_S}^{i=SGW}$, $\gamma_{u \in V_S}^{i=MME}$ to mark each the potential servers for the S-GW and MME.

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:

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

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

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

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. The right-hand side of these constraint formulations represents a delay threshold, whereas the left-hand side computes the actual delay between i and j by accumulating the delay over the assigned links. The latter is calculated with the aid of the boolean expression f_{uv}^*/g^*, which is 1 if the link uv is assigned and 0 otherwise. Finally, the conditions (9) and (10) express the domain constraints for the variables x_u^i (binary) and f_{uv}^{ij} (real).

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

4.2 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:

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

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:

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

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

We use a rounding algorithm to extract feasible solutions from the LP solutions that potentially contain non-boolean x_u^i. More specifically, the algorithm invokes a call to the LP solver and processes the set of feasible LP solutions iteratively. Each iteration includes the rounding of the x_u^i variables of the current solution and either the acceptance or the rejection of the request. If the rounded solution does not violate the capacity and delay constraints then the request is accepted; otherwise it is rejected. Algorithm 1 shows the pseudo code for the LP rounding.

Algorithm 1. NF placement with LP rounding

```
repeat
    {x_u^i, f_uv^ij} ← Solve_LP(..)
    FeasSol := true if solution for LP exists, false otherwise
    X ← {x_u^i | x_u^i ∉ {0,1}}
    if X ≠ ∅ then
        {i_fx, u_fx} ← argmax_{i∈V_F, u∈V_S} X
        if ( Σ_{i∈{V_F | x_{u_fx}^i = 1}} g^i + g^{i_fx} ≤ r_{u_fx} ) and ( Σ_{(i,j)∈E_F} Σ_{(u,v)∈E_S} (f_uv^ij / g^ij) d_uv ≤ d_max ) 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 = ∅) ∨ (FeasSol = false)
if FeasSol = true then
    return {x_u^i, f_uv^ij} {Accept request}
else
    ∀x_u^i := 0, ∀f_uv^ij := 0
    return {x_u^i, f_uv^ij} {Reject request}
end if
```

Our tests[3] show that both MILP and LP lead to server and link load balancing[4] (Figs. 4 and 5). 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. 6). 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. 7). Consequently, since the LP trades a small degree of optimality for a substantially lower runtime, we use this variant in our evaluations (Sect. 5).

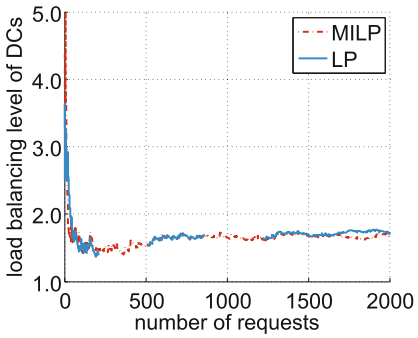

Fig. 4. DC load balancing level (based on server load).

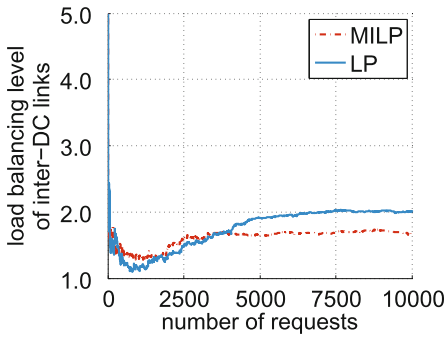

Fig. 5. Inter-DC link load balancing level.

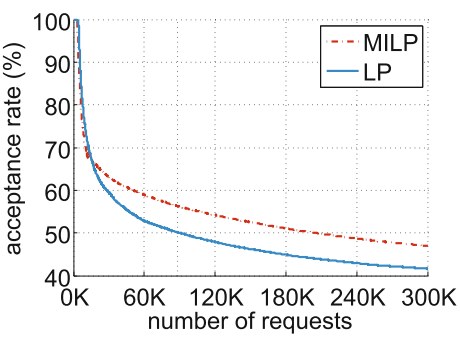

Fig. 6. Request acceptance rate.

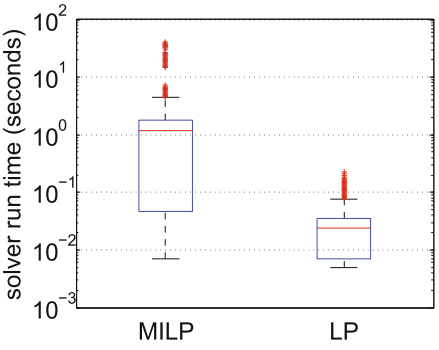

Fig. 7. Solver runtime.

[3] The tests were conducted on a 2 GHz AMD Opteron server (restricted to single core).
[4] See Sect. 5.1 for the definition of the load balancing level.

Algorithm 2. NF Placement - Greedy Algorithm (Baseline)

$DCs \leftarrow$DCs, ordered by distance from eNB
$NrByDCs \leftarrow DCs$ with $d_{(eNB \rightarrow DC)} \leq d_{(eNB \rightarrow DC),max}$
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 $d_{(eNB \rightarrow MME)} \leq d_{(eNB \rightarrow MME),max}$ **and** $d_{(MME \rightarrow SGW)} \leq d_{(MME \rightarrow SGW),max}$ **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 $d_{(SGW \rightarrow PGW)} \leq d_{(SGW \rightarrow PGW),max}$ **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

4.3 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 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.

Greedy algorithms are generally known to be time-efficient but sub-optimal. Based on our analysis, there is a substantial optimality gap between the MILP solutions and the solutions of the greedy algorithm. More precisely, the MILP yields approximately 100% better load balancing compared to the greedy variant. We use the greedy algorithm as a baseline in our evaluation in Sect. 5.

5 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 (Sect. 5.1), the evaluation metrics (Sect. 5.2), and the evaluation results (Sect. 5.3).

5.1 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 2 shows additional cellular core network parameters.

Radio Access Network. We rely on a multi-cell scenario for the RAN, similar to the one presented in [18] that was based on real statistics from a region in Paris. However, in our case, we consider varying user density ($\rho = U[385, 2308] UEs/\text{km}^2$), so that the number of active UEs per eNB ranges

from 500 to 3000 (Table 3). Considering uniform circular cells with an overlapping factor γ of 1.2, the required cell radius is $r = \gamma\sqrt{A_t/C\pi}$.

Table 2. Cellular core network parameters

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

Table 3. 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 4. 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

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 4 [17]. $Pr\{O\}$ 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 4). We derive the CPU demands for each NF from resource profiles, similar to [19]. Based on the session parameters of Table 4, 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 [32].

MME Signaling Load and Traffic. 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 [17,18] and 3GPP LTE/EPC signaling messages (Fig. 1) and their sizes provided in [31]. In this respect, applications are modeled 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-Connected*) and Idle (*ECM-Idle*) states, as described in Fig. 1. In other words, only *Service Request/Release* procedures are taken into account. The RRC inactivity timer defines the inactivity period required for the UE to switch to IDLE state. This timer is adjusted to 10 s, which is a widely used setting in cellular networks.

Based on the model provided in [17,18], the processing load at the MME for a *Service Request - Release* is given by:

$$L_{MME} = \beta\rho\, A_c C[M_{MME}^{UE-SR} P_{UE} + M_{MME}^{NET-SR}(1 - P_{UE}) + M_{MME}^{SRel}]$$

where A_c, C, ρ denote the cell area, number of eNBs, and the UE density, respectively. The number of messages M per case for the Service Request (SR) and Service Release ($SRel$), depending on where the session is originated (UE or NET), are given in Fig. 1. The same methodology is used for all SR involved control plane elements. In a similar manner, we estimate the UL/DL bandwidth demands, given the sizes of the various packets exchanged during the *Service Request*. For example, in the case of UL between eNB and MME, the bandwidth demand is given by:

$$T_{eNB-MME}^{UL} = \beta\, \rho\, A_c[P_{ICSR} + P_{CRQ} + P_{CRTE}]$$

where P_{ICSR}, P_{CRQ}, P_{CRTE} denote the packet size for the *Initial Context Setup Response*, the *UE Context Release Request*, and the *UE Context Release Complete*, respectively [31]. The same methodology is used for all control links.

5.2 Evaluation Metrics

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.

5.3 Evaluation Results

Initially, we discuss the load balancing in the cellular core, which is the main objective of our LP. Figure 8 shows the load balancing level among the DCs, based on server load. The LP achieves a significant improvement in the DC load balancing level (Fig. 8) compared to the baseline which corresponds to the common practice today. We note that LP's load balancing efficiency is achieved while complying with 3GPP. These constraints most of the times inhibit the partitioning of service chains across DCs. This is corroborated in Table 5, which shows the number of DCs used for the assignment of each service chain, on average. Relaxing the 3GPP-associated constraints is expected to yield even better load distribution across the DCs, as inter-DC service chaining partitioning will not be restricted.

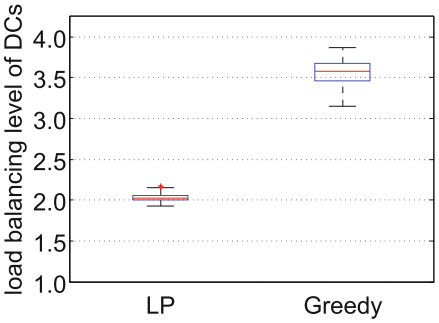

Fig. 8. DC load balancing level (based on server load).

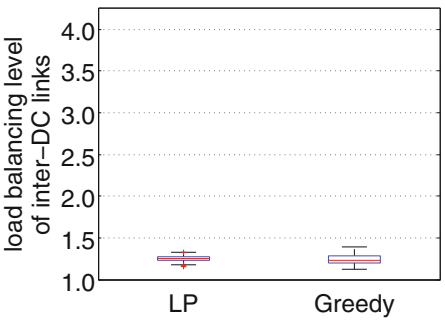

Fig. 9. Inter-DC link load balancing level.

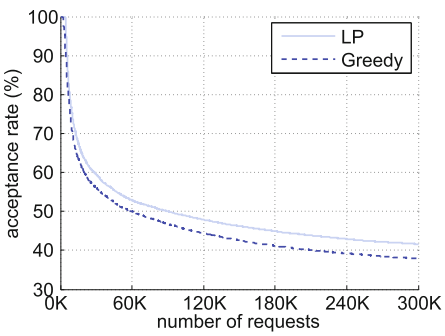

Fig. 10. Request acceptance rate.

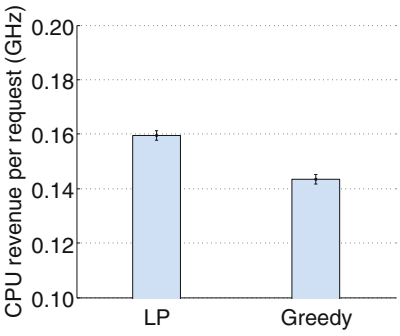

Fig. 11. Revenue from CPU per request.

Table 6 provides additional insights into the NF placement by the LP and the greedy algorithm. We observe that both NF placement methods minimize the

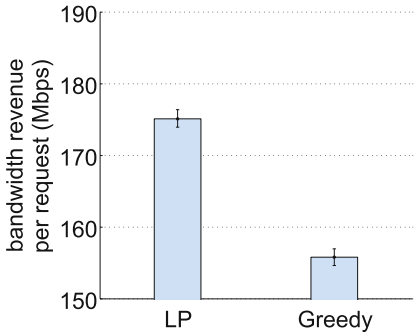

Fig. 12. Revenue from bandwidth per request.

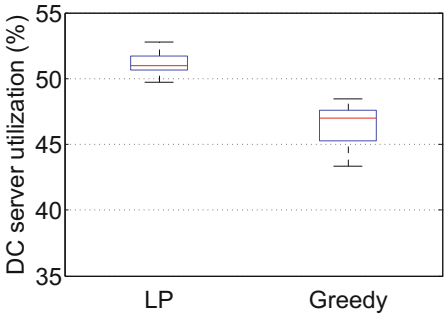

Fig. 13. Server utilization.

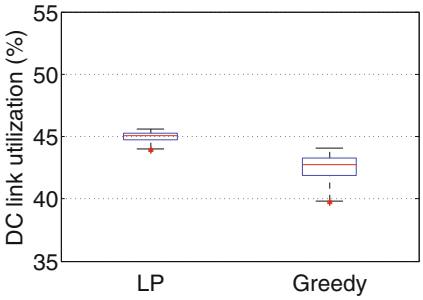

Fig. 14. Intra-DC link utilization.

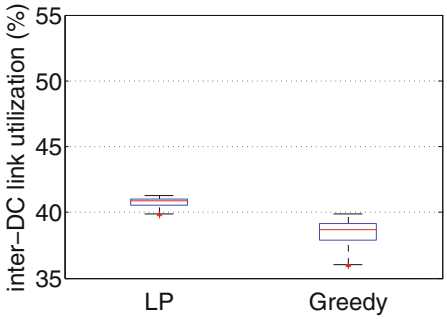

Fig. 15. Inter-DC link utilization.

number of servers assigned to each service chain (subject to capacity constraints), thus minimizing inter-rack traffic within DCs. In certain cases, both methods accomplish the co-location of all NFs in the same server, which reduces the provisioning cost for cellular network operators.

We further investigate load balancing across the links connecting the DCs. As shown in Fig. 9 both the LP and the greedy algorithm yield an equally high level of inter-DC link load balancing.

Figure 10 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,

Table 5. Number of DCs per service chain

Method	1 DC	2 DCs
LP	74.56%	25.95%
Greedy	77.48%	22.52%

Table 6. Number of servers per service chain

Method	1 server	2 servers	3 servers
LP	18.63%	81.27%	0.10%
Greedy	23.76%	76.18%	0.06%

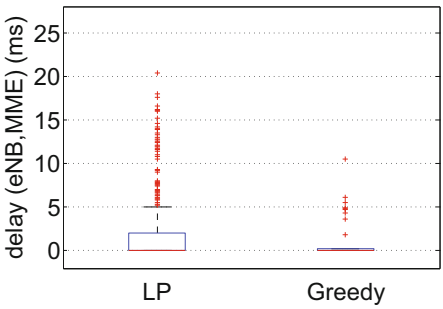

Fig. 16. Delay between eNB and MME.

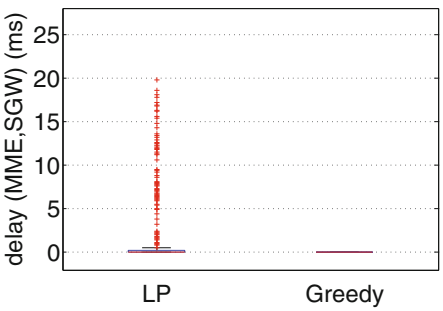

Fig. 17. Delay between S-GW and MME.

i.e., 12% more CPU and 13% more bandwidth demand per request relatively to the revenue generated by the baseline (Figs. 11 and 12). 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. 10), 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.

Figures 13, 14 and 15 depict the utilization level of the 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. 10). 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.

Finally, we investigate the delays incurred between communicating data and control plane EPC elements. In this respect, Figs. 16 and 17 illustrate the delay between the eNB and the MME, and the delay between the MME and S-GW, respectively. It can observed that delays are below the 50 ms threshold (mandated by 3GPP) for both NF placement methods. As expected, the greedy algorithm yields very low delays, since it strives to assigns all EPC vNFs close to the eNB. On the other hand, the LP exploits the delay budgets to achieve a more flexible

placement in order to achieve DC load balancing. The mean delays incurred with the LP are significantly lower than the 50 ms delay budget.

6 Related Work

In this section, we provide an overview of 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 [10,35,37]. In the same direction, additional approaches to the problem [14,34] 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, the placement of virtualized EPC control plane functions (*e.g.,* MME, HSS) along S/P-GWs has been also considered towards the instantiation of a 3GPP-compliant elastic cellular core [12,28]. Furthermore, latency bounds for control as well as data plane traffic have also been considered to match real-world deployments [12,28].

KLEIN [28] proposes an orchestration framework for a software-defined mobile core network. The corresponding NF placement problem is formulated as an ILP, with the goal to minimize the total resources (capacity per DC) allocated to the virtualized mobile core, subject to capacity constraints of the physical resources and delay budgets for each traffic class supported. KLEIN argues that the attempt to model latency constraints between the control and data plane functions yields quadratic constraints. As such, potential solutions include the co-location of control and data plane functions or a two step procedure, *i.e.,* control function placement followed by data function placement problem, with an additional constraint on the delay budgets between the MME and S-GW components. To deal with complexity and scalability issues, KLEIN decomposes the global optimization into a three-level hierarchy; (i) UE aggregates are assigned to specific regions (Region Selection Problem) solving the aforementioned ILP problem, assuming the regional collocation of control and data plane functions (ii) then these aggregates are further assigned to DCs (DC Selection Problem) running a two step placement procedure to break the quadratic dependency issue, and (ii) finally KLEIN solves the Server Selection Problem using an appropriate greedy heuristic.

In [12], authors also decompose the EPC network graph into a data-plane chain and several control-plane ones. The service chain endpoints are (i) the RAN traffic aggregation point (for a cluster of eNodeBs) and (ii) Internet Exchange Point(s). Their goal is to jointly embed them into the underlying virtualized infrastructure. The problem is formulated as an ILP and solved optimally for small problem instances. Authors have extended their work in [13], taking into account delay budgets. Specifically, they propose a MILP formulation for the joint embedding of individual 3GPP-compliant core network service chains, considering end-to-end data- and control-plane latency bounds.

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, tailored to specific traffic classes and service delivery (*e.g.,* with the addition of 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.

7 Conclusions and Future Work

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.

In future work, we plan to couple our NF placement methods with service chaining and NF state transfer for EPC-wide orchestration. We will further investigate techniques for scaling in/out existing NF instances (*e.g.,* based on our previous work in [15]) to provide better responses to evolving service demands.

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. SONATA Project. http://www.sonata-nfv.eu/
5. UNIFY Project. http://www.fp7-unify.eu/
6. 3GPP TS 24.301: 3GPP Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS). http://www.3gpp.org/DynaReport/24301.htm

7. Business Case for Juniper Networks Virtualized Mobile Control Gateway, White Paper, Juniper (2013)
8. Abujoda, A., Kouchaksaraei, H.R., Papadimitriou, P.: SDN-based source routing for scalable service chaining in datacenters. In: Mamatas, L., Matta, I., Papadimitriou, P., Koucheryavy, Y. (eds.) WWIC 2016. LNCS, vol. 9674, pp. 66–77. Springer, Cham (2016). doi:10.1007/978-3-319-33936-8_6
9. Abujoda, A., Papadimitriou, P.: MIDAS: middlebox discovery and selection for on-path flow processing. In: IEEE COMSNETS, Bangalore, India, January 2015
10. Bagaa, M., Taleb, T., Ksentini, A.: Service-aware network function placement for efficient traffic handling in carrier cloud. In: IEEE WCNC, Istanbul, Turkey, April 2014
11. Banerjee, A., et al.: Scaling the LTE control-plane for future mobile access. In: ACM CONEXT, Heidelberg, Germany, December 2015
12. Baumgartner, A., Reddy, V.S., Bauschert, T.: Mobile core network virtualization: a model for combined virtual core network function placement and topology optimization. In: IEEE NetSoft 2015, London, UK, April 2015
13. Baumgartner, A., Reddy, V.S., Bauschert, T.: Combined virtual mobile core network function placement and topology optimization with latency bounds. In: EWSDN 2015, Bilbao, Spain, September 2015
14. Basta, A., et al.: Applying NFV and SDN to LTE mobile core gateways, the functions placement problem. In: 4th Workshop on All Things Cellular, ACM SIGCOMM 2014, Chicago, US, August 2014
15. Cao, Z., Abujoda, A., Papadimitriou, P.: Distributed data deluge (D3): efficient state management for virtualized network functions. In: IEEE INFOCOM SWFAN, San Francisco, USA, April 2016
16. Cohen, R., Lewin-Eytan, L., Naor, J., Raz, D.: Near optimal placement of virtual network functions. In: IEEE INFOCOM, Hong Kong, China, April 2015
17. Diego, W., Hamchaoui, I., Lagrange, X.: The cost of QoS in LTE/EPC mobile networks evaluation of processing load. In: IEEE VTC, Boston, MA, USA (2015)
18. Diego, W., Hamchaoui, I., Lagrange, X.: Cost factor analysis of QoS in LTE/EPC mobile networks. In: IEEE CCNC, Las Vegas, USA, January 2016
19. Dietrich, D., Abujoda, A., Papadimitriou, P.: Network service embedding across multiple providers with nestor. In: IFIP Networking, Toulouse, France, May 2015
20. Dietrich, D., Papagianni, C., Papadimitriou, P., Baras, J.: Network function placement on virtualized cellular cores. In: IEEE COMSNETS, Bangalore, India, January 2017
21. Bari, M.F.: Data center network virtualization: a survey. IEEE Commun. Surv. Tutorials **15**(2), 909–928 (2013)
22. Fayazbakhsh, S., et al.: Enforcing network-wide policies in the presence of dynamic middlebox actions using flowtags. In: USENIX NSDI 2014, Seattle, USA, April 2014
23. Gember-Jacobson, A., et al.: OpenNF: enabling innovation in network function control. In: ACM SIGCOMM 2014, Chicago, USA, August 2014
24. Hirschman, B., et al.: High-performance evolved packet core signaling and bearer processing on general-purpose processors. IEEE Netw. **29**(3), 6–14 (2015)
25. Lukovszki, T., Schmid, S.: Online admission control and embedding of service chains. In: Scheideler, C. (ed.) Structural Information and Communication Complexity. LNCS, vol. 9439, pp. 104–118. Springer, Cham (2015). doi:10.1007/978-3-319-25258-2_8
26. Mehraghdam, S., Keller, M., Karl, H.: Specifying and placing chains of virtual network functions. In: IEEE CloudNet, Luxembourg, October 2014

27. Prados-Garzon, J., et al.: Latency evaluation of a virtualized MME. In: IEEE Wireless Days, Toulouse, France, March 2016
28. Qazi, Z., et al.: KLEIN: a minimally disruptive design for an elastic cellular core. In: ACM SOSR 2016, Santa Clara, USA, March 2016
29. Qazi, Z., et al.: SIMPLE-fying middlebox policy enforcement using SDN. In: ACM SIGCOMM 2013, Hong Kong, China, August 2013
30. Rajan, A.S., et al.: Understanding the bottlenecks in virtualizing cellular core network functions. In: IEEE LANMAN, Beijing, China, April 2015
31. Sama, M.R., Ben Hadj Said, S., Guillouard, K., Suciu, L.: Enabling network programmability in LTE/EPC architecture using OpenFlow. In: WiOpt 2014, Hammamet, Tunisia, May 2014
32. Savic, Z.: LTE Design and Deployment Strategies - CISCO. http://tinyurl.com/lj2erpg
33. Shafiq, M.Z., Ji, L., Liu, A.X., Pang, J., Wang, J.: A first look at cellular machine-to-machine traffic: large scale measurement and characterization. In: ACM SIGMETRICS, London, UK, June 2012
34. Taleb, T., Bagaa, M., Ksentini, A.: User mobility-aware virtual network function placement for virtual 5G network infrastructure. In: IEEE ICC 2025, London, UK, June 2015
35. Taleb, T., Ksentini, A.: Gateway relocation avoidance-aware network function placement in carrier cloud. In: ACM MSWiM, Barcelona, Spain, November 2013
36. Wang, Z., et al.: An untold story of middleboxes in cellular networks. In: ACM SIGCOMM 2011, Toronto, Canada, August 2011
37. Yousaf, F., et al.: SoftEPC: dynamic instantiation of mobile core network entities for efficient resource utilization. In: IEEE ICC, Budapest, Hungary, June 2013