A Power Efficient and Robust Virtual Network Functions Placement Problem

Antonio Marotta and Andreas Kassler
University of Karlstad
Universitetsgatan 2, 65188 Karlstad, Sweden
Email: antonio.marotta, andreas.kassler {@kau.se}

***Abstract*—Reducing the CAPEX and OPEX is a major concern for Telecom Operators (TOs): to this extent, Network Function Virtualization (NFV) has been considered a key aspect to virtualize network functions and push them to the NFV Infrastructure. Virtual Network Functions (VNFs) can be deployed as a set of components running on several cooperating Virtual Machines (VMs) inside modern data centers. As a consequence, it becomes crucial for network operators to minimize the power consumption of their NFV infrastructure, by using the minimum set of physical servers and networking equipment subject to the constraints that VNFs impose on the infrastructure in terms of compute, memory, disk and network resources requirements. In this work, we present a joint resources and flow routing assignment problem for VNFs placement, with the objective of minimizing both the power consumption of the servers and switches needed to deploy the overall virtualized infrastructure and the routing graph. In contrast to many existing works assuming perfect knowledge on input parameters, such as VNFs CPU demands, which is difficult to predict, we propose a novel mathematical model based on the Robust Optimization (RO) theory to deal with data uncertainty. Our numerical evaluation focuses on a specific use-case, that is the deployment of a virtualized Evolved Packet Core (vEPC), namely the core for next generation mobile networks. We demonstrate that with our model, a vEPC operator can trade-off between two important aspects: the power consumption minimization on one side, and the protection from severe deviations of the input parameters on the other (e.g. the resources requirements).**

***Index Terms*—Network Function Virtualization, VNF Placement Problem, Evolved Packet Core, Mixed Integer Optimization, Robust Optimization.**

I. INTRODUCTION

Telecom Operators (TOs) currently use a combination of vendor specific hardware and software to implement their network functions such as Load-balancers, Firewalls, Mobility Management Entity and so on. With the advent of Cloud Computing, those functions are increasingly virtualized, leading to the concept of Network Function Virtualization (NFV) [1]. This is a promising technology which leverages the benefits of virtualization in the telecommunication realm. The purpose is to offer telecommunication features as a service, so that mobile network operators can take advantages of the cloud paradigm. Indeed by deploying Virtualized Network Functions (VNFs) on Virtual Machines (VMs), operators are allowed to run telecommunication services inside virtualized data centers on commodity hardware with the advantage of reducing capital expenses. More importantly, by changing VMs resources dynamically (e.g. by adding more compute or memory resources), the VNFs may be scaled according to the load or new VMs can be dynamically spawned on demand, and this significantly simplifies the VFN operation and management.

In general a Virtual Network Infrastructure (VNI) consists of different VNFs interconnected by well-defined interfaces and thus creating a VNF forwarding graph (e.g. service chain). A VNF itself consists of different components (VNFCs), each one executing a clear task. VNF instances need to be deployed on VMs in the NFV infrastructure which is composed of compute, storage and network resources. The virtualization layer maps the virtual resources to the physical servers, disks and network nodes. One of the main aspects of virtualization is the resources consolidation, since more VMs may reside on the same physical server. However, the more VMs are hosted on the same physical machine, the higher the potential for contention and, thus, the possibility of SLA violations. On the other hand, deploying each VNFC on a different server may result in high energy consumption, which may be unnecessary during low load. From a NFV infrastructure point of view, it is obvious to pursue the objective of saving as much energy as possible, in order to reduce the electricity costs and the CO_2 footprint, leading towards the concept of Green NFV Infrastructure operation. In order to maximize the energy efficiency, TOs need to place as many VNFs as possible on the smallest set of physical servers, by trying to guarantee the expected quality of service to the final users. In the literature, several optimization models have been proposed to solve different versions of the VNFs placement problem. A common assumption of those models is that input data is precisely known beforehand, which is very difficult to achieve in practice. For instance, it is hard to predict how much CPU a VNF will require or how much data VNF A will send towards VNF B during its execution time. Unfortunately, the presence of uncertain data may produce solutions to optimization problems that are useless in practice [2], [3]. This is due to the fact that small deviations in input data values may usually lead to situations where an optimal solution, previously found, is not even feasible any more. By taking into account variable CPU demands, if during runtime some VNF components allocated on the same server require more CPU than the expected one, contention may occur leading to severe service degradation. Consequently, we need to develop models that cope with data uncertainty by applying e.g. Stochastic Programming or Robust Optimization (RO).

In this paper, we present an optimization model based on

978-0-9883045-1-2/16 $31.00 © 2016 ITC

RO to solve the joint resources and flow routing assignment problem when designing a VNI, with the objective of minimizing both the power consumption of the servers and switches needed to deploy each VNF under uncertainty. In particular we are considering uncertainty on the VNFC resource demands, since it is likely that the CPU utilization may vary according to the VNF processing load. We model the service chain as a set of communicating VNFCs, with traffic demands and latency constraints. Our model allows a TO to protect against parameter deviations, by avoiding overloading situations that are a result of aggressive VNF consolidation but, at the same time, incurring in the so-called *price of robustness* [4]. Our numerical results demonstrate that it is possible to achieve a trade-off between the power consumption and the protection from deviations in the CPU utilization that may lead to low and unexpected QoS levels. By taking into account more severe and unlikely deviations of VNFCs requirements, the model will provide a higher protection but also a higher power consumption. Alternatively, a more risky placement will offer less protection at a lower power consumption. Our model can be used by a VNI operator to balance between the two conflicting goals, according to the sensitiveness of the VNF to resources contention. In our numerical evaluation we consider a virtual Evolved Packet Core as use-case, namely the cornerstone of next generation mobile networks.

The rest of the paper is structured as follows: in section II the related work on the VNFs placement problem is discussed. In section III, the RO theory is introduced; while in section IV the problem formulation is described, alongside with the robust model. In section V the use case for the experimental evaluation is illustrated and some numerical results for the vEPC case are presented. The last section concludes the paper and lists our future work.

II. Related Work

The VNFs placement is a well-studied problem in the literature due to its importance for Telecom Operators. In [5], the authors present an optimization model for the embedding of Virtual Mobile Core Networks. They face the problem of resource allocation for a core network service chain which is intended as a combination of VNFs that the user or control plane related traffic needs to traverse. In their formulation latency is nicely modelled as a combination of processing, packet queuing and propagation delay, where the first two variables depend on the traffic utilization of the node the VNF is placed on, while the last one is a function of the path length. Authors show numerical results of the model on a real network topology. However, in their assumption input parameters are precisely known in advance and they do not study the impact of such uncertainty. [6] presents an Integer Linear Programming (ILP) model for VNF orchestration. The problem consists in finding the number of necessary VNFs and allocating them in order to minimize the total network related cost and the resources fragmentation. The model is optimally solved, even if a dynamic programming based heuristic is also used for large instances. In [7] an interesting approach is applied to the aim of reducing the energy consumption in telecommunication networks, by consolidating the available resources. A game theoretic based procedure is applied to drive the resource consolidation and achieve a good trade-off between energy efficiency and network resiliency. The work in [8] investigates the benefits of the application of two approaches to the telecommunication networks: the network functions virtualization on one hand, and the Software Defined Networking paradigm for functions decomposition on the other. A model to solve the VNFs placement is proposed, whose objective is to minimize the total network load overhead, by taking into consideration several parameters, such as the data plane delay and the SDN control overhead. In [9], two constraint-based heuristics are applied for the deployment of a virtualized infrastructure providing Evolved Packet Core services. The authors discuss all the involved parameters to determine the components needed to run the VNF and they show the results in terms of average number of used CPU cores and aggregate throughput for two placement strategies.

In all those models and algorithms, perfect knowledge on all the parameters is assumed and optimal values are computed based on such precise input assumptions. However, if input parameters later vary the optimal solution previously found may be totally infeasible, by making those approaches impractical. We address those deficiencies in our approach by modelling the VNFs placement problem using Robust Optimization. We also consider the network topology in the model, and we model the collaborating VNFCs as a service chain. By varying the protection level, the model is able to output different solutions that trade off between the total power consumption of physical resources and protection against SLA violations (for instance due to resource contention or link overloading). The theory of RO has been already applied in a different context successfully. For example, in Chaisiri et al. [10], authors present a robust cloud resource provisioning approach (RCRP) that considers fluctuation in resource demands and cloud providers resource price. In [11] RO theory is applied to a well-known problem, namely the Virtual Machines Consolidation. [12] present an approach to mitigate network jammers in the wireless context based on a very interesting extension of RO that allows to model multi-band robustness. Also in [13] the RO theory is applied for solving the Virtual Network Embedding (VNE) problem on a physical network substrate. The objective is to maximize the revenue that comes from the embedding of virtual nodes and links with a constraint on the capacity budget. In order to solve large instances, they propose a very interesting two phase heuristic which is based on Γ-robustness to deal with capacity requests variability.

III. Introduction to Robust Optimization Theory

Robust Optimization [2], [3], [4] tries to mitigate problems that arise in optimization under input data uncertainty or deterministic variability. One difference to stochastic optimization is that the probability distribution of the parameters uncertainty is not known beforehand: instead, it is specified by a so-called *uncertainty set*, representing the parameters space which the problem is optimized over. According to [3] we can write an uncertain linear optimization problem as:

$$\begin{aligned} &\min\ \mathbf{c^T x} \\ &s.t.\ \ \mathbf{Ax} \leq \mathbf{b}\ \ \mathbf{x} \in X \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ is the vector of decision variables and X is a deterministic polyhedron. The uncertain parameters may assume arbitrary values from a given uncertainty set $\mathbf{U}$. The aim is to find the minimum cost solutions $\mathbf{x}^*$ among all feasible ones for any possible realization of the unknown coefficients. In other words, the constraints need to be satisfied for all the possible values out of the given uncertainty set $\mathbf{U}$. The problem can be translated into the robust counterpart as in [14]:

$$\begin{aligned} &\min\ \ \mathbf{c^T x} \\ &s.t.\ \ \ \mathbf{Ax} \leq \mathbf{b},\ \ \ \forall \mathbf{a_1} \in \mathbf{U_1}, ..., \mathbf{a_m} \in \mathbf{U_m},\ \ \mathbf{x} \in X \end{aligned} \tag{2}$$

where $\mathbf{a_i}$ is the i-th row of the uncertain matrix $\mathbf{A}$, taking values from the uncertainty set $\mathbf{U_i} \subseteq \mathbb{R}^{\mathbf{n}}$. We call a solution robust feasible if it satisfies all the uncertain constraints $\mathbf{a_i^T x} \leq b_i \forall \mathbf{a_i} \in \mathbf{U_i}$ and any optimal solution of (2) is called a robust optimal solution. The robust counterpart (2) has typically infinitely many constraints and provides solutions that are worse than the ones provided by the original (non-robust) problem, since RO tries to mitigate the effects of uncertainty.

An important aspect of the robust optimization is the definition of the uncertainty set. Bertsimas and Sim [4] considers an uncertainty set that allows to specify a sort of uncertainty budget $\Gamma \geq 0$. For a given uncertain matrix $\mathbf{A} = (a_{ij})$ we can assume that each coefficient a_{ij} has a nominal value $\bar{a}_{ij}$ and a possible symmetric maximum deviation $\hat{a}_{ij} \geq 0$, thus lying in the interval $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$. We allow that at most Γ_i coefficients of row i may deviate from their nominal value and Γ_i denotes the budget of uncertainty of constraint i. Then, we can define the robust uncertainty set as all coefficients for which the sum of the relative deviations from the nominal values is at most Γ_i. More formally, given the parameter a_{ij}, we define $z_{ij} = (a_{ij} - \bar{a}_{ij})/\hat{a}_{ij}$ and we require that

$$\sum_{j=1}^{n} |z_{ij}| \leq \Gamma_i\ \ \forall i. \tag{3}$$

Γ_i defines the maximum number of parameters, whose values deviate towards their maximum value. When $\Gamma_i = 0$, all the parameters are at their nominal values and the solution is not protected against any uncertainty. On the other hand, if $\Gamma_i = n$, the i-th constraint is fully protected against uncertainty, leading to the most conservative solution. Any trade-off in between is possible: by tuning Γ_i, we can now obtain more robust solutions characterized by higher Γ_i and leading to worse objective function values, but, at the same time, protecting from more parameter deviations. Or otherwise we can accept more opportunistic solutions with a lower Γ_i, leading to better objective values, but also to a higher risk. This uncertainty model allows to compute an upper bound to the probability of constraints violation as shown in [4]. Consequently, a robust solution remains feasibly with a high probability and Γ_i controls the trade-off between the constraint violation probability and the impact on the objective function.

IV. Problem Formulation

Our main goal is to derive a mathematical optimization model for the robust VNFs placement problem, where several parameters are not known precisely. The objective is to minimize the overall energy consumption of the VNI in terms of power needed to drive the physical servers, switches and other networking equipment. This will be achieved by placing the VNFs on the smallest set of physical servers and powering down unused switches, switch-ports and physical servers. An important aspect is that too many VNFs should not be allocated on the same server as we need to take into account the uncertainty in demand fluctuations of the components, in terms of compute resources. In order to derive such a model, we need to look into the power consumption models for the physical infrastructure in a data center.

A. Server and Switch Power Models

Most part of the whole VNI power consumption is due to physical servers that run the VNFCs. The servers power consumption depends on several factors such as CPU load, memory, cache states, and so on. As stressed out by several papers in the literature, the most influential subsystem on the server's power consumption is the CPU. When a server is powered on but not experiencing any load, it consumes P_j^{idle}. Hence, the power consumption can be simplified as:

$$P_j(t) = P_j^{idle} + (P_j^{max} - P_j^{idle}) \cdot used_j^{CPU}(t) \tag{4}$$

where $used_j^{CPU}$ is the usage (in percentage) of the processor (value between 0 and 1). At each time instant the power consumption is linearly increasing with the CPU utilization, due to the running tasks on the physical machine. If the physical server does not run any VNF, we power down that server.

In our work we also model the energy consumption of the network that connects the physical servers running the set of VNFs. As a consequence, we need to model the switches power consumption, which can be expressed as [15]:

$$P = P_{ch} + n_c \cdot P_c + \sum_{r=1}^{max} n_p^r \cdot P_p^r \cdot u_p \tag{5}$$

The power consumption of a switch is given by:

- a static power consumption related to the chassis, namely the frame for mounting the circuit components, (P_{ch}), and a number (n_c) of line cards providing the network interfaces.
- the power consumption by a number (n_p^r) of powered on ports operating at a specific rate and characterized by a total utilization u_p.

We simplify (5) to obtain the switch power consumption as:

$$P = P^{static} + \sum_{p \in Q_{act}} P_p \tag{6}$$

where P^{static} is the static power component and the sum involves the power of all the switch active ports. We assume that if a port is not carrying any traffic, we can power down it. Finally, if all the ports are powered down, we can power down the whole switch.

B. Problem Definition

As introduced in I, a VNI can consist of different cooperating VNFs, whose functionality is accomplished through different VNFCs, each one with a well defined interface. We are considering a group of Service Chains (SCs) to model a single VNF: each service chain is a group of VMs that communicate among each other, by exchanging a certain amount of traffic. In order to provide service guarantees, we associate a certain maximum tolerable latency for each service chain. In order to illustrate this concept, assume a generic VNF which is composed of two service chains as it follows:

$$VNF_1 = \{sc_1, sc_2\} \tag{7}$$

$$SC_1 = \{vm_1, vm_2, vm_3\} \tag{8}$$

$$SC_2 = \{vm_4, vm_5\} \tag{9}$$

$$Demands = \{(vm_1, vm_2), (vm_2, vm_3), (vm_4, vm_5)\} \tag{10}$$

$$Latencies = \{(vm_1, vm_3), (vm_4, vm_5)\} \tag{11}$$

The first SC contains three VMs and it is characterized by two traffic demands; the second one is composed by two VMs and has a single traffic demand. For each service chain we define the maximum latency which is the latency for packets sent by the first VM and forwarded to the last one in the same chain. In our example, the first SC has one traffic demand between VM_1 and VM_2 and another between VM_2 and VM_3, but the latency constraint is applied to any combination of paths the traffic originating in VM_1 can be sent over in order to reach VM_3. Consequently, we need to find the set of links in the communication graph that fulfil the latency constraint given by the service chain.

We suppose that the cloud environment consists in a number of j physical machines, v VNFs to deploy, s service chains and m VMs where to run the different VNF components. Each physical server can be connected to a particular node in the network topology, consisting of n nodes (e.g. switches). The static allocation of each server to a network node in the topology is given by the binary matrix al. A node in the topology is considered as a switch with a specific power model and a set of links, each of which has a power consumption, an expected latency and a maximum bandwidth. Given these assumptions, we tackle the following **Power Efficient Robust VNFs Placement Problem**:

Given the amount of resources available at each server, the amount of CPU and RAM requirements for the VNF components, the CPU uncertainty model and the power profile of the servers and switches, the problem consists in finding the placement for the VNFs and jointly the traffic flows routing that minimize the total power consumption due to the active servers, switches and links.

The objective function of the problem is:

$$f = \sum_{j \in J} Pj + \sum_{n \in N} Pn \tag{12}$$

The placement must guarantee that the used resources on each physical server do not exceed the available amount, the traffic on the links should not be greater than the available bandwidth and the latency constraint for each service chain is respected.

C. Robust VNFs Placement Model

The problem has been modelled as an optimization model with robustness constraints on resources requirements and traffic demands. All the input parameters and the decision variables are summarized in Table I. The overall model is shown in Table II.

	Input parameters:
a_{ij}	is the amount of resource i available at server j
sc_{sm}	is 1 if the VNFC m belongs to the service chain s
r_{im}	is the amount of resource i requested by VNFC m
$\Delta r_{i,m}$	is the max variation in the usage of resource i by VNFC m
$P_{idle,n}$	is the static power consumption of node n
$P_{idle,j}$	is the idle power consumption of server j
$P_{max,j}$	is the maximum power consumption of server j
$e.(s, d, b, pw, lat)$	is the source, destination, max bandwidth, power consumption and latency of link e
d_{m_1,m_2}	is the traffic demand between m_1 and m_2
lat_s	is the maximum latency which can be tolerated by service chain s
al_{jn}	represents which network node n the server j is connected to
Γ	is the protection level from parameters deviation
	Decision variables:
x_{jm}	is 1 if VNFC m is allocated to server j
$actSr_j$	is 1 if server j is active, 0 otherwise
$UncReq_{\Delta r_{i,m}}$	is the uncertain usage of resource i by VNFC m
$P_{f,j}$	is the power consumption of server j
$P_{f,n}$	is the power consumption of node n
$f_e^{m_1,m_2}$	is the flow demand d_{m_1,m_2} on the link e
$h_e^{m_1,m_2}$	is 1 if the link e is carrying the demand d_{m_1,m_2}
H_e	is 1 if the link e is used for any traffic
F_e	is the total traffic on the link e
$actSw_n$	is 1 if node n is active, 0 otherwise

Table I: Model Parameters

Each server is connected to a network node: this is specified through the binary variable al_{jn}. A server is active if and only if it is hosting at least one component belonging to a VNF after the placement. The binary decision variable indicating if the server hosts any VNFCs is $actSr_j$ (1 means the server is up, 0 the server is shut-off). Similarly, we use a binary decision variable, $actSw_n$, to describe that a network switch is active, meaning at least one of its links is actually carrying traffic.

The Optimization Model

$$\min f = \sum_{j=1}^{J} P_{f,j} + \sum_{n=1}^{N} P_{f,n} \quad (13)$$

s.t.

$$util_{ij} = \sum_{m}^{M} x_{jm} \cdot (r_{im} + UncReq_{\Delta r_{i,m}}) \quad \forall j, \forall i \quad (14)$$

$$\sum_{m}^{M} \left| \frac{UncReq_{\Delta r_{i,m}}}{\Delta r_{i,m}} \right| \leq \Gamma \quad (15)$$

$$P_{f,j} = P_{idle,j} \cdot actSr_j + (P_{max,j} - P_{idle,j}) \cdot \frac{util_{ij}}{a_{ij}} \quad \forall j, i = CPU \quad (16)$$

$$\sum_{j}^{J} x_{jm} = 1 \quad \forall m \quad (17)$$

$$actSr_j \leq \sum_{m}^{M} x_{jm} \quad \forall j \quad (18)$$

$$actSr_j \geq x_{jm} \quad \forall j, \forall m \quad (19)$$

$$util_{ij} \leq a_{ij} \cdot actSr_j \quad \forall j, \forall i \quad (20)$$

$$P_{f,n} = P_{idle,n} \cdot actSw_n + \sum_{e:e.s=n} H_e \cdot e.pw \quad \forall n \quad (21)$$

$$\sum_{e:e.d=n} f_e^{m_1,m_2} - \sum_{e:e.s=n} f_e^{m_1,m_2} = \quad (22)$$
$$d_{m_1,m_2} \cdot \sum_{j}^{J} ((x_{jm_2} \cdot al_{jn}) - (x_{jm_1} \cdot al_{jn})) \quad \forall n, m_1, m_2 \quad (23)$$

$$h_e^{m_1,m_2} \leq (2 - \sum_{j}^{J} (x_{jm_2} \cdot al_{jn}) - \sum_{j}^{J} (x_{jm_2} \cdot al_{jn}) \quad \forall n, e, m_1, m_2 \quad (24)$$

$$h_e^{m_1,m_2} \leq \sum_{j}^{J} (x_{jm_1} \cdot al_{jn}) \quad \forall e, n : e.s = n, m_1, m_2 \quad (25)$$

$$h_e^{m_1,m_2} \leq \sum_{j}^{J} (x_{jm_2} \cdot al_{jn}) \quad \forall e, n : e.d = n, m_1, m_2 \quad (26)$$

$$F_e = \sum_{m_1,m_2} f_e^{m_1,m_2} \quad \forall e \quad (27)$$

$$F_e \leq e.b \cdot H_e \quad \forall e \quad (28)$$

$$h_e^{m_1,m_2} \cdot d_{m_1,m_2} = f_e^{m_1,m_2} \quad \forall e, m_1, m_2 \quad (29)$$

$$actSw_n \leq \sum_{e:e.s=n||e.d=n} H_e \quad \forall n \quad (30)$$

$$actSw_n \geq H_e \quad \forall n, \forall e : e.s = n || e.d = n \quad (31)$$

$$\sum_{m:m=m_1||m_2,e} h_e^{m_1,m_2} \cdot e.lat \leq lat_s \quad \forall s, \forall m : m \in \text{service chain } s \quad (32)$$

$$x_{jm} = \{0,1\}, act_j = \{0,1\}, act_n = \{0,1\}$$
$$h_e^{m_1,m_2} = \{0,1\}, H_e = \{0,1\}$$

Table II: Problem Model

We consider a set of m possible VNF components, each of which can be deployed inside a single VM. Each server and switch has a power consumption model whose description is given in section IV-A. The network topology consists of a set of links connecting the switches, with a given capacity, a maximum latency and an estimated power consumption. All the components belonging to a service chain are exchanging traffic according to a given pattern and have specific resource demands. A specific traffic demand is traversing all the VNFCs and each service chain has a maximum latency bound. To cope with the uncertainty of the resource demands of a given service chain, we assume that the resource demands needed by a VM running a given VNF component are not precisely known. Rather, we assume that we know a mean resource demand and a maximum allowed deviation from the mean, specified as symmetrically distributed upper and lower bounds, as specified for the coefficients $a_{i,j}$ in section III. In more detail, we consider that a VM which is running a specific VNFC m, requires an expected demand $r_{i,m}$ of resource i (e.g. memory or CPU). We model the uncertainty of the demand by introducing a random variable $UncReq_{\Delta r_{i,m}}$, which is symmetrically distributed between $[-\Delta r_{i,m}, +\Delta r_{i,m}] \cdot r_{i,m}$ and with 0 mean.

The objective to minimize the total power consumption due to the active server, switches and links after the placement is computed in (13). Each server shows a resource utilization which is computed by considering all the requests by all VNFCs placed on it and the uncertainty on the demands (14). The constraint (15) assures that the sum of the deviations of the uncertain resource (over all the possible VNFCs) should not exceed the protection level Γ. The server power profile is computed through (16): if the physical machine is not hosting any VNFCs, it can be shut-off and its power consumption is set to 0. Otherwise the power consumption is linearly increasing with the CPU usage (14). The binary decision variable x_{jm} represents the placement of the VNFC m on the physical server j. The model assures that each VNFC m is placed on a single server (17). Constraint (18) guarantees that the server is shut-off, if it is not hosting any VNFCs; on the other side, if at least one VNFC is placed on it, it should be active (19). The constraint (20) avoids that the used resources on a server exceed the maximum available amounts.

The power consumption of the switch is computed as in (21). If the switch is powered on, its power consumption is the static component plus the dynamic one which is dependent on the active outgoing links. If the switch is not used, then the power is set to 0. (22) is the so-called flow conservation constraint: the sum of the flows entering one node should be equal to the sum of the ones exiting from the node (except for the source and the sink). In other words, for each traffic demand and network node, if the flow is generated by the considered node and directed to a different one, then the demand is multiplied by -1 (the traffic source is m_1). Otherwise, when the node is attracting the flow, the demand is multiplied by 1. For intermediate nodes, the difference between the sums should be zero. Constraint (24) assures that, given a demand between m_1 and m_2, if both the source and destination are placed on a server which is connected to the same node, this traffic is not flowing into the network. Considering a link, its source node n and a demand between m_1 and m_2, if m_1 is not allocated to any server connected to n, the flow exiting from the node should be zero (25). The same happens for incoming links (26). The total flow on a link e is the sum of all the demands forwarded through it (27), and it should be not greater than the available capacity

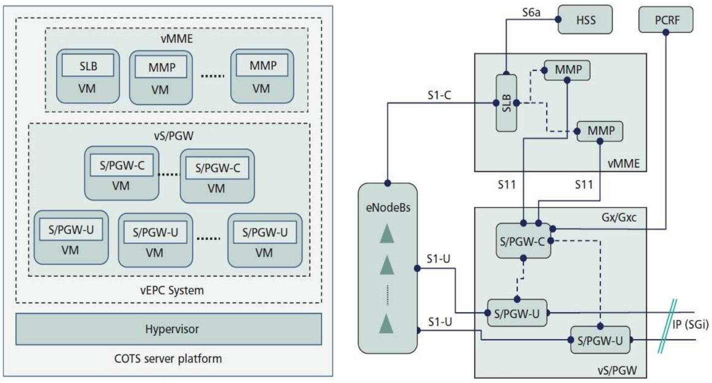


Figure 1: Virtualized Evolved Packet Core (vEPC) system: functional and interface overview [9]

of the active link (28). If a link e is used to carry traffic between m_1 and m_2, then the flow $f_e^{m_1,m_2}$ should be equal to the demand itself; otherwise it must be zero (29). In this model we compute a single path for each traffic demand, or in other words the flow is unsplittable. The constraints (30) and (31) assure that a switch is active if and only if at least one of its connecting links is used to forward traffic; if no link is active, then the switch can be shut-off. The last constraint (32) regards the latency bound for each service chain: the sum of all the path latencies used for the traffic flows of a specific service chain should not exceed the maximum tolerable one.

V. A Numerical Evaluation: the vEPC Use-case

The experimental evaluation of the model was conducted on a particular example of VNF, namely the Evolved Packet Core (EPC). With the advent of cloud and virtualization, the Telecom Operators are moving parts or the next generation networks infrastructure into the cloud, leading to solutions named EPCaaS (EPC as a Service). The EPC is thus composed of different entities, each of which can be considered as a stand-alone VNFC running on a dedicated VM as shown in Figure 1. Among the components there are:

- Evolved NodeB (eNB). The user gets connected to the mobile access network after being associated to an eNB, which represents the base station for the radio technology.
- The Mobility Management Entity (MME) is the component which processes control plane related information (e.g. the signalling traffic for handling the user mobility and security for mobile network access). Another task accomplished by the MME is to keep tracking of User Equipments (UEs) in idle-mode. The MME usually consists of the Signalling Load Balancer (SLB) and the Mobility Management Processor (MMP). Their combination is useful to scale the control traffic originating in eNBs and to balance it among different MMPs. These components execute the real processing tasks of the MME.
- The gateways, Serving GW and Packet Data Network (PDN) GW, process data related to the user plane, as they forward packets from the UE towards external destinations. The S-GW represents the interconnection between the radio access and the core of the EPC and it determines the routing for the packets coming from the UEs. The Serving GW is the interconnection point between the radio-side and the EPC. As its name indicates, this gateway serves the UE by routing the incoming and outgoing IP packets. The PDN GW connects the EPC to the external world and it is responsible for routing the packets to and from the PDNs.
- The Home Subscriber Server (HSS) is the component that handles the database where all the subscribed users related data is residing. Besides the HSS has other functionalities, such as user mobility management, call session establishment and security tasks (e.g. user authentication and access authorization).
- The Policy and Charging Rules Function (PCRF) is in charge of applying policy rules in next-generation networks. It allows the creation of rules and the enforcement of decisions for the subscriber in the network. Among the tasks of the PCRF there are billing, rating, charging, session and call establishment with guaranteed quality of service.

The infrastructure design of the EPC allows to distinguish between the user data (also known as user-plane) and the signalling traffic (also know as the control-plane). This separation clearly simplifies the dimensioning of the Telecom Operator's networks. Hence, two separate VNFs can be considered which basically coincide with the control data-path (e.g. the signalling traffic coming from the eNBs and forwarded towards the S/PGW-C though SLB and MMPs) and the user data-path (user traffic coming from eNBs and destined to the S/PGW-Us). Each one of the VNF can be characterized by a different value of latency, since signalling traffic and user packets may tolerate different values of processing and forwarding latency.

A. Numerical Results

For the experimental evaluation we considered the placement of two VNFs, as discussed in the previous section. The experimental values used for the servers, switches, VNFCs and links are shown in Table III. For the CPU requirements uncertainty we took into account a maximum deviation, $\Delta r_{i,m}$,

Table III: Experimental Values

Physical CPU cores	18, 14, 15, 12, 5, 4, 1, 2, 2, 2, 3, 3
Physical RAM (GB)	9, 5, 7.5, 5, 2.5, 2, 0.5, 1, 1, 1, 1.5, 1.5
Server Idle Power Consumption (%)	20, 30, 20, 30, 40, 40, 30, 30, 30, 20, 30, 30
Server Max Power Consumption (W)	160, 160, 160, 160, 290, 270, 260, 220, 200, 200, 180, 160
VNFC CPU demands	5, 4, 4, 4, 4, 6, 4, 3, 3, 2.5, 1.5, 1, 1.5, 2.5, 2, 3, 1.5, 1.5, 2
VNFC RAM requirements	2, 2, 1.5, 1, 1, 3.5, 1, 1, 1, 1.5, 1.5, 0.5, 1, 1.5, 1, 1.5, 1, 1, 1
Switches Idle Power (W)	100, 70, 140, 120
Links Power (W)	35.4, 7.4, 35.4, 35.4, 35.4, 35.4, 35.4, 7.4, 35.4, 35.4
Links Bandwidth (Mbps)	80, 80, 30, 50, 80, 30, 100, 80, 50, 100

ranging from 10% to 30% of the demand, with step 10. We increase the protection level from 0 (no protection, or in other words deterministic problem) up to the total number of VMs (we assume all VMs demands may deviate to the max). The dimensioning of the VNFs is realized through the parameters found in [9]. By considering the control plane load, measured in terms of events per hour, we compute the number of instances for each type of VNFC required to sustain the estimated hourly traffic bundle. We assume 500.000 signalling events per hour generated by UEs connected to the eNBs. In particular we used 8 eNB, 1 MPP, 1 SLB, 1 S-GWC, 6 S-GWU, 1 HSS and 1 PCRF. Since the deterministic model is already very hard to solve and the additional uncertainty even increases its complexity, we reduced the number of service chains, by considering some random set of VNFCs exchanging traffic out of all the possible combinations. To be more clear, a control plane related service chain can consist of any possible instance of the eNBs sending a certain amount of traffic to the SLB, which in turn forwards packets to the MMP and so on. For our use case we considered these values:

- 12 physical servers and 19 VNFCs;
- 4 network nodes and 10 links;
- 16 service chains and 27 traffic demands.

The model was implemented in Matlab [16] using the Robust Optimization Made Easy (ROME) [17] toolbox. ROME transforms the robust model into its deterministic form, which is then solved using CPLEX [18]. The experimental evaluation was conducted on a cloud based system (Abisko [19]), which is part of the High Performance Computing (HPC) cluster in Uppsala. It is comprised of 328 nodes (a total of 15888 CPU cores), each of which is equipped with 4 AMD Opteron 6238 (Interlagos), 12 cores, 2.6 GHz processors.

We use a very simplified network topology that is shown in Figure 2: for convenience, the bottom layer switches coincide with the top of racks where the physical servers are allocated. Hence, this figure points out the association of each server to a network node: in particular servers 1, 2, 3, 4, 8, 9, 10 are allocated to node 1, while servers 5, 6, 7, 11, 12 are associated with node 2. As we can see from Figure 3, for each demand deviation we present two graphs: the upper one shows the total power consumption of the VNI (in W, left axis) and the constraint violation probability $Pr(\omega, \Gamma)$ for a given Γ (varied on x-axis). The expected power consumption is computed over

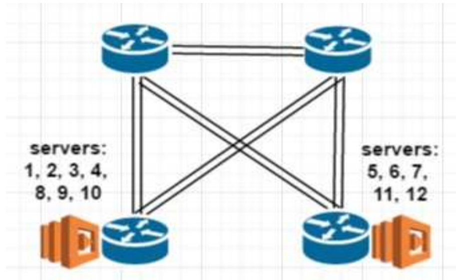
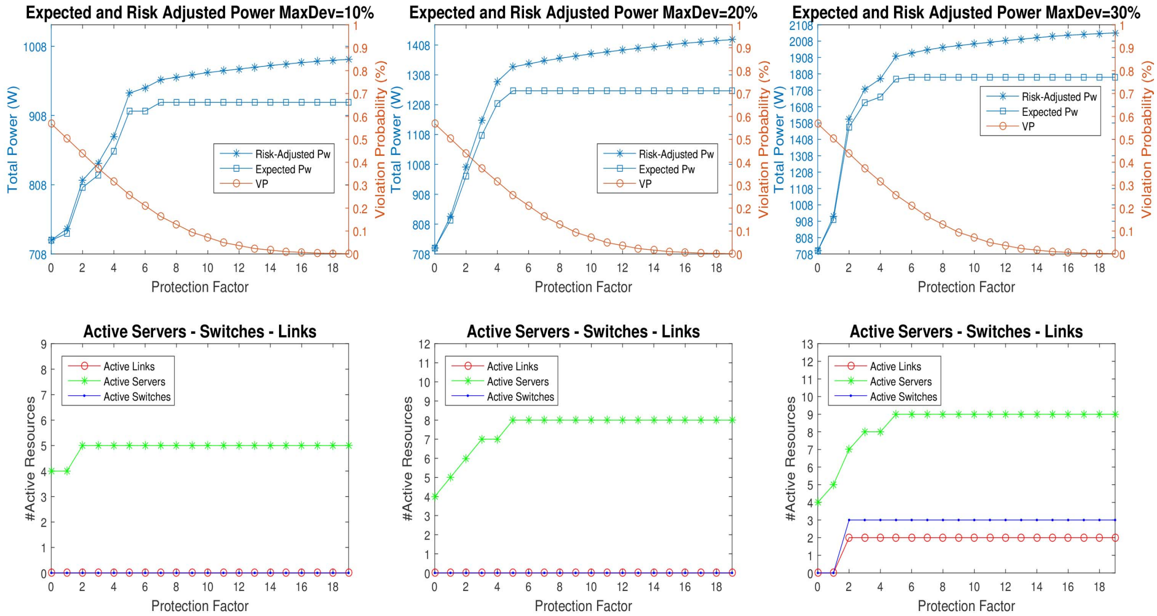


Figure 2: Network Topology

all the active servers, switches and links after the placement, assuming that the CPU demands are at the mean value, while the risk adjusted power assumes that a given number of CPU requests are deviating to the maximum Δ (e.g. $\Gamma = 1$, one CPU requirement deviating at the maximum Δ). On the lower graph we plot the number of active servers, switches and links which are needed to deploy the VNI for different values of the protection level Γ (outcome of the placement).

Figure 3(a) shows the results for maximum CPU demand deviation equal to 10%. When $\Gamma = 0$, the solution of the model is the deterministic one (all input parameters are known precisely). This solution allocates all the VNFCs in the servers 1, 2, 3, 4 that are connected to the node 1 in the network topology and no traffic is going out on any link. The first three servers have a 100% CPU utilization, while server 4 has a 75% CPU usage with a total power consumption of 791 W. Since $\Gamma = 0$ means we do not protect from CPU demand deviations, the expected power consumption coincides with the risk adjusted one. The probability of constraint violation is 56.8%. Consequently, when considering perfect knowledge on all input parameters, there is 56.8% possibility of constraint violation if at least one VNF component deviates from its nominal CPU demand, leading to resource overload and potential SLA violations. When $\Gamma = 1$, the model reallocates the VNFCs in order to block enough CPU resources on each server to accommodate one single VNFC CPU demand variation. This allocation has a total nominal power consumption of 737 W and a higher risk-adjusted power consumption of 742 W (a single CPU demand may be allowed to deviate to the maximum Δ). When the demand of VNFC CPU requirements are varying to the max allowed, four servers are no longer



(a) Deviation 10%

(b) Deviation 20%

(c) Deviation 30%

Figure 3: Results for different deviations and protection levels

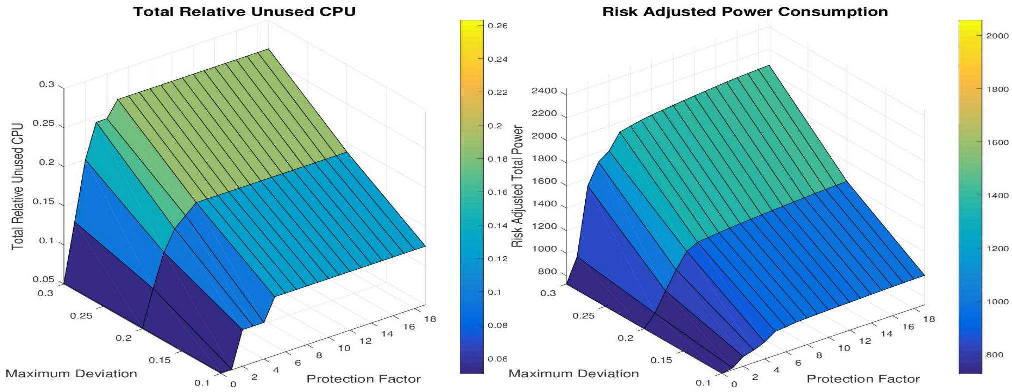
sufficient to protect the solution. Hence, a new server is activated (11) with a CPU utilization equal to 33.33%. For all the Γ values, the VNFCs are allocated so that no traffic flow is being forwarded in the network. Besides when Γ increases, both the expected and the risk adjusted power consumption increase: in particular the expected power is stable after $\Gamma \geq 7$, while the risk adjusted power increases. The expected power is also changing when $1 \leq \Gamma \leq 7$, even if the VNFCs allocation always involves 5 servers. This is due to the fact that the used servers have different levels of energy efficiency.

The same trends can be also observed when we allow a larger deviation percentage. In particular when the maximum variability is 20% (Figure 3(b)), the model protects from deviations by using one more server for each $\Gamma \leq 4$. The maximum number of activated servers is achieved when the protection level is equal to 5, with a total expected power consumption of 1254 W. The number of used servers is stable when considering higher values of Γ. When the allocation is fully protected, the maximum gap between the power values is 12%. In Figure 3(c), the results for a maximum deviation of 30% are shown. The number of used servers is increased of 2 when the protection level goes from 1 to 2, with a total relative unused CPU utilization equal to 20% and an expected power consumption of 1482 W. The highest number of physical machines needed to deploy all the VNFCs and to cope with the CPU variability is reached when $\Gamma = 6$: 9 servers are needed and the total expected power consumption is 1776 W, while the risk adjusted one is 7.2% higher. Interestingly, when the maximum deviation is 30%, the energy consumption is significantly increasing. This is due to the fact that for $\Gamma = 2$, the VNFCs allocation is not suitable to avoid traffic in the network. Instead, the new allocation requires 4 demands to be forwarded in the network, leading to the activation of 3 routers (1, 2, 4) and 2 links (1-4, 4-2), with a total flow equal to 28 on them. For higher protection levels, also more servers are needed leading to further increase in energy consumption. Instead, the same network configuration is operated for higher Γ values, even if with a different total amount of flow according to the allocation scheme.

In Figure 4(a), the total relative unused CPU is plotted: this is the sum of the available CPU of all the active servers minus the total CPU requested by the VNFCs, normalized to the total available CPU. In general, Figure 4(a) shows that when Γ is increasing, the more the allocation sets aside spare CPU to cope with potential deviations. In particular, when the maximum deviation is 10% and $\Gamma = 2$ the relative unused CPU is growing from 5% to 9%, due to the fact that a new server was activated. Then the same servers are utilized until $\Gamma = 5$, when the server 11 is shut-off and server 5 is used instead, with a total relative unused CPU equal to 12.5%. When the allocation scheme is fully protected, the gap between the expected power and the risk-adjusted one is 6.2% (a maximum risk adjusted power of 989 W). Finally Figure 4(b) summarizes the risk adjusted power for different maximum allowed CPU demand deviations for different protection levels.

VI. Conclusions and Future Work

In this paper we modelled the Robust Virtual Network Functions Placement Problem by taking into consideration the physical servers and network resources along with their energy efficiency. The objective of the problem is to find the VNFs placement that jointly minimize the power consumption due to the servers and the switches needed to deploy all the required virtualized functions. The theory of Robust



(a) Risk Adjusted Power Consumption

(b) Total Relative Unused CPU Usage

Figure 4: Results Risk Adjusted Power and Total Relative Unsued CPU

Optimization was applied to the optimization problem to cope with uncertain input parameters, assuming that we have a budget of uncertainty in terms of cardinality constraints, i.e. the number of uncertain parameters that are allowed to deviate from their nominal values. Our numerical evaluation shows how a Telecom Operator can balance the energy consumption and the protection from input parameters that are deviating from their nominal values. By modifying the protection level of the solutions, the operator can calculate more conservative solutions that consume higher total energy or more opportunistic ones at lower energy consumption, having a higher risk of SLA violation. In the future we plan to apply stochastic algorithms to the problem in order to be able to solve also very large instances and develop fast solution heuristic for online optimization, which may be integrated into an ETSI MANO framework for NFV Orchestration such as e.g. Tacker which is based on OpenStack and OpenDaylight.

Acknowledgments

Parts of this work has been funded by the High Quality Networked Services in a Mobile World (HITS) and the READY projects, partly funded by the Knowledge Foundation of Sweden.

References

[1] Chiosi M., et. al., "Network Functions Virtualisation - Introductory white paper," 2015. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[2] Ben-Tal A. and El Ghaoui L. and Nemirovski, A.S., "Robust Optimization. Princeton University Press, Princeton Series in Applied Mathematics," 2009.

[3] Bertsimas D. and Thiele A., "Robust and data-driven optimization-modern decision-making under uncertainty. INFORMS Tutorials in Operations Research: Models, Methods, and Applications for Innovative Decision Making," 2006.

[4] Bertsimas, D. and Sim M., "The Price of Robustness. Operations Research 52," 2004.

[5] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Combined virtual mobile core network function placement and topology optimization with latency bounds," in *Software Defined Networks (EWSDN), 2015 Fourth European Workshop on*, Sept 2015, pp. 97–102.

[6] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions in NFV," *CoRR*, vol. abs/1503.06377, 2015. [Online]. Available: http://arxiv.org/abs/1503.06377

[7] A. P. Bianzino, C. Chaudet, D. Rossi, J. L. Rougier, and S. Moretti, "The green-game: Striking a balance between qos and energy saving," in *Teletraffic Congress (ITC), 2011 23rd International*, Sept 2011, pp. 262–269.

[8] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular '14. New York, NY, USA: ACM, 2014, pp. 33–38. [Online]. Available: http://doi.acm.org/10.1145/2627585.2627592

[9] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 60–66, Dec 2015.

[10] S. Chaisiri, B.-S. Lee, and D. Niyato, "Robust cloud resource provisioning for cloud computing environments," in *Service-Oriented Computing and Applications (SOCA), 2010 IEEE International Conference on*, Dec 2010, pp. 1–8.

[11] E. Zola and A. J. Kassler, "Energy efficient virtual machine consolidation under uncertain input parameters for green data centers," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, Nov 2015, pp. 436–439.

[12] F. D'Andreagiovanni, "Revisiting wireless network jamming by sir-based considerations and multiband robust optimization," *Optimization Letters*, vol. 9, no. 8, pp. 1495–1510, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11590-014-0839-2

[13] S. Coniglio, A. M. C. A. Koster, and M. Tieves, "Virtual network embedding under uncertainty: Exact and heuristic approaches," in *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*, March 2015, pp. 1–8.

[14] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, Aug. 2011. [Online]. Available: http://dx.doi.org/10.1137/080734510

[15] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Computing*, vol. 18, no. 1, pp. 385–402, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10586-014-0404-x

[16] Matlab. [Online]. Available: http://se.mathworks.com/products/matlab

[17] Rome robust optimization made easy user guide. [Online]. Available: http://www.robustopt.com/references/ROME_Guide_1.0.pdf

[18] Ibm cplex. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[19] Abisko hpc cluster. [Online]. Available: https://www.hpc2n.umu.se/resources/abisko