An Approach for Service Function Chain Routing and Virtual Function Network Instance Migration in Network Function Virtualization Architectures

Vincenzo Eramo, Emanuele Miucci, Mostafa Ammar, *Fellow, IEEE*, *ACM*, and Francesco Giacinto Lavacca

***Abstract*— Network function virtualization foresees the virtualization of service functions and their execution on virtual machines. Any service is represented by a service function chain (SFC) that is a set of VNFs to be executed according to a given order. The running of VNFs needs the instantiation of VNF Instances (VNFIs) that in general are software modules executed on virtual machines. The virtualization challenges include: 1) where to instantiate VNFIs; ii) how many resources to allocate to each VNFI; iii) how to route SFC requests to the appropriate VNFIs in the right sequence; and iv) when and how to migrate VNFIs in response to changes to SFC request intensity and location. We develop an approach that uses three algorithms that are used back-to-back resulting in VNFI placement, SFC routing, and VNFI migration in response to changing workload. The objective is to first minimize the rejection of SFC bandwidth and second to consolidate VNFIs in as few servers as possible so as to reduce the energy consumed. The proposed consolidation algorithm is based on a migration policy of VNFIs that considers the revenue loss due to QoS degradation that a user suffers due to information loss occurring during the migrations. The objective is to minimize the total cost given by the energy consumption and the revenue loss due to QoS degradation. We evaluate our suite of algorithms on a test network and show performance gains that can be achieved over using other alternative naive algorithms.**

***Index Terms*— Network function virtualization, migration policy, power consumption, Markov decision process.**

I. Introduction

The concept of NFV originated from service providers looking to increase the agility and flexibility of deploying new network services to support growing customer demands [1]. NFV offers new ways to design, orchestrate, and manage network services. NFV decouples network functions from underlying hardware so these functions can run as software images on commodity hardware as well as custom-built hardware. It is a framework that provides virtualization of network services such as routing, load balancing, firewall services, intrusion detection and prevention, and network address translation into building blocks. These services can be chained together to create network service chains tailored for different use cases [2], [3]. Because the infrastructure is simplified and streamlined, new and expanded services can be created quickly and with less expense. Implementations of the paradigm have also been proposed [4] and the performance has been investigated [5]. To support the NFV technology both ETSI [6], [7] and IETF [8] have been defining novel network architectures able to allocate resource for Virtualized Network Function (VNF) as well as to manage and orchestrate NFV to support services. In particular the service is represented by a Service Function Chain (SFC) [8] that is a set of VNFs that have to be executed according to a given order. Any VNF is run on a VNF Instance (VNFI) implemented with one Virtual Machine (VM) to which resources (cores, RAM memory,....) are allocated to execute a VNF of a given type (e.g., a virtual firewall, or a load balancer) [9].

The VNF consolidation, routing and placement problems in an NFV environment are similar to the ones involved in cloud computing environments. The main difference is that a Service Function Chain (SFC) has a sequence requirement [10] which means the traffic must be steered to traverse through predefined ordered network functions. This important difference makes the algorithms implemented in cloud computing environment not applicable in NFV architectures.

VNFI consolidation allows for energy consumption savings. Unfortunately it involves reconfiguration costs characterized, for instance, by the revenue loss due to the QoS degradation experienced when servers are shut down during migration. For this reason migration is not recommended in some cases while, in other cases, only limited function migration can be justified. While server consolidation in cloud computing environments has been studied extensively,there is only limited research that proposes and investigates consolidation algorithms for NFV architectures. The network function consolidation problem is investigated by Wen *et al.* [11]. They propose a greedy heuristic allowing for the SFC reconfiguration and the instantiation/removal of VNFIs but reconfiguration costs are not considered. Ghaznavi *et al.* [12] propose a consolidation algorithm called Simple Lazy Facility Location (SLFL) that optimizes the placement of the VNF Instances in response to on-demand workload. SLFL chooses the VNFIs to be migrated on the basis of the reconfiguration costs that the migration

Manuscript received March 26, 2016; revised November 11, 2016; accepted February 6, 2017; approved by IEEE/ACM Transactions on Networking Editor K. Argyraki. Date of publication March 2, 2017; date of current version August 16, 2017. This work was supported by TELECOM ITALIA within the research agreement Definition and Evaluation of Protocols and Architectures of a Network Automation Platform for TI-NFV Infrastructure.

V. Eramo, E. Miucci, and F.G. Lavacca are with the DIET, Sapienza University of Rome, 00184 Rome, Italy (e-mail: vincenzo.eramo@uniroma1.it; emanuele.miucci@uniroma1.it; francescogiacinto.lavacca@uniroma1.it).

M. Ammar is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280 USA (e-mail: ammar@cc.gatech.edu).

Digital Object Identifier 10.1109/TNET.2017.2668470

involves in the current instant and without taking into account benefits and penalties that these migrations lead in successive instants.

Two main contributions are provided in the paper. The first one is to propose an algorithm for the SFC routing and VNFI placement in which the dimensioning of the VNFI in terms of number of cores is not a-priori established as assumed in the most contributions proposed in literature, [11] but it is an output of the problem; the proposed algorithm has the objective to perform SFC routing and VNF placement during the Peak Hour Interval so as to minimize the SFC bandwidth rejected. The second contribution is to propose a migration policy of the VNFIs allocated by the SFC routing and VNF placement in the periods in which traffic changes occurs. The objective of the migration policy is to minimize a total cost on the overall temporal horizon, taking into account both the energy consumption and the reconfiguration costs. The policy provides an appropriate trade-off between the energy saving advantages due to the consolidation and the disadvantages of the revenue loss of an operator due to the QoS degradation occurring during the VNFI migration. We give an Integer Linear Programming (ILP) formulation of the optimal problem and, due to its complexity, we propose a solution in cycle-stationary traffic scenarios and when the possible mappings of VNFIs to servers are evaluated a-priori according to a given optimality criteria. We evaluate the effectiveness of the proposed solution heuristic by comparing its results to that of exact solutions of the ILP formulation for the case of small network. Finally, we compare performance of the proposed migration policy to a simple local policy in which the migration decisions consider only instantaneous reconfiguration costs with no longer term consideration.

The rest of this paper is organized as follows. The main related works are discussed in Section II. Section III provides a summary of our approach. The network and traffic model are introduced in Section IV. Section V is devoted to illustrate the heuristic for the resource dimensioning and the SFC routing in the Peak Hour Interval (PHI). The proposed migration policy, proposed for the total cost minimization in VNFI migration scenarios, is introduced in Section VI. Some numerical results are shown in Section VII to prove the effectiveness of the proposed migration policy and to compare it to the policy based on current reconfiguration costs only. Finally the main conclusions and future research items are mentioned in Section VIII.

II. Related Work

The SFC routing and VNFI placement is a problem widely studied in literature [10], [13]–[15]. In these works the formulation of the problem is given and its objective can be the minimization of the active servers, the minimization of the latency, the minimization of the number of SFCs rejected. In all of the these cases the authors highlight how the problem is NP-hard and for this reason heuristics are proposed.

All of the proposed routing and placement algorithms [10], [16], [17] assume the availability of VNFIs needing a given number of cores and providing a given processing capacity [18], [19]. In other words, the number of cores allocated is an input to the problem and not a variable to be optimized.

Most of the papers on the NFV topic are focused on the study of routing and placement algorithms. Few studies propose and investigate resource consolidation techniques to be applied, for instance, to save power consumption in low traffic periods. Wen *et al.* [11] propose a heuristic for SFC reconfiguration and compare its results to the the optimal solution implemented in CPLEX. They do not consider reconfiguration costs characterizing the revenue loss of an operator due to the QoS degradation occurring when SFCs are reconfigured. Ghaznani *et al.* [12] propose consolidation algorithms based on horizontal scaling techniques in which the processing capacity dedicated to a VNF is increased/decreased by instantiating/removing VNFIs and without changing the processing capacity allocated to the VNFIs. The study's objective is to find a solution minimizing the network operational costs (bandwidth, server processing capacity) by taking into account the reconfiguration costs characterized by a penalty due to the migration of VNFIs that are consolidated/de-consolidated in servers in the periods of low/high traffic. The authors observe how the general problem is very complex and only the penalty due to the VNFI migrations at the present instant are considered and not the future ones.

In this paper, due to the advantages to avoid complex NFV state management issue, we investigate solutions based on vertical scaling techniques [12], [20]–[22] in which the VNFIs are dimensioned to achieve the processing capacity required by the traffic. When the traffic increases/decreases, rather than add/remove VNFIs, we consider increasing/decreasing their processing capacity. The main contribution is the study of consolidation, routing and placement problems in NFV architectures when vertical scaling techniques are adopted.

We have already investigated the SFC routing and placement problems in [28] but in this paper we extend the Maximizing the Accepted SFC Requests Number (MASRN) algorithm proposed in order to improve the blocking performance in the scenario in which bandwidth rather than processing capacity is the constrained resource.

We also outline that the innovative contribution of this paper is the proposal and the investigation of a new VNFI consolidation technique taking into account the reconfiguration costs as in [12] but with important difference of choosing the VNFIs to be migrated according to not only the actual penalty but also the future ones that the migrations may lead to. We formulate the optimization problem in the case in which the operational and reconfiguration costs are characterized by the energy consumption and the information loss due to VNFI migrations respectively. The proposed solution is based on the application of the Markov Decision Process theory, already applied by ourselves to solve other problems (overlay network [24], virtual network embedding [23], [25], [26]).

III. Overview of Approach

This paper proposes and investigates solutions for the deployment of NFVs with the objective of minimizing the operator cost. To achieve this we consider two main problems as illustrated in Fig. 1.a.

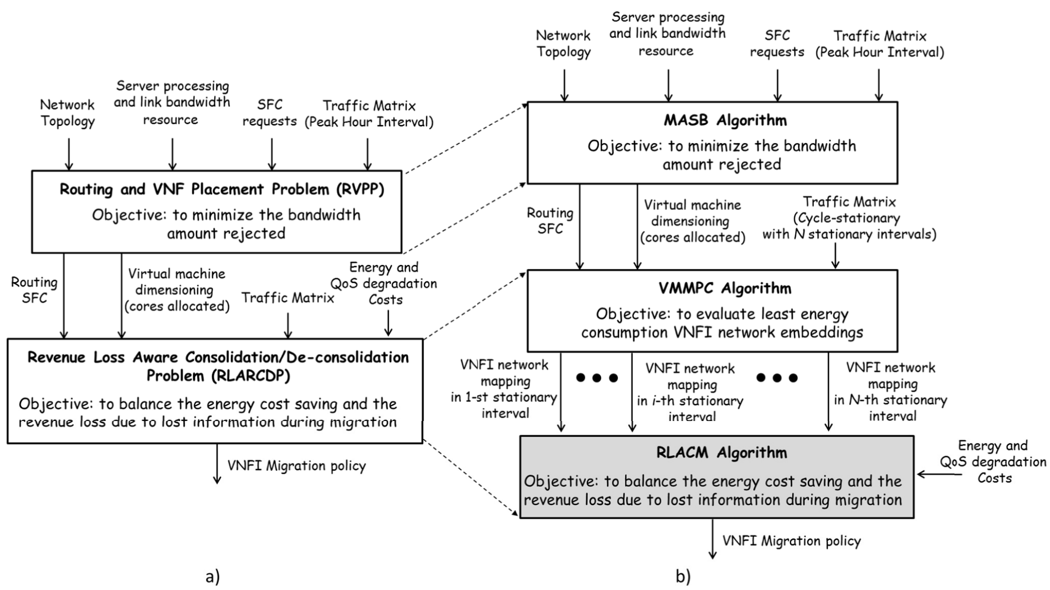


Fig. 1. Inputs, outputs and objectives of RVPP and RLARCDP (a); MASB, VMMPC and RLACM algorithms developed to solve RVPP and RLARCDP (b).

- Routing and VNF Placement Problem (RVPP). The objective is to maximize the amount of data that can be processed within the network at peak traffic. We assume we are given: network topology, server processing and link bandwidth resources, SFC requests, and the traffic matrix at the peak hour. Our optimization here yields as output, VNFI placement, processing resource (number of cores) allocation to VNFIs and the SFC routing.
- Revenue Loss Aware Resource Consolidation/ De-consolidation Problem (RLARCDP). The objective is to minimize the network operation cost which is the sum of the energy consumption costs and the revenue loss due to network disruption caused by VNFI migration [27] as functions are consolidated or de-consolidated. To achieve this objective the number of cores allocated to each VNFI is decreased in low traffic conditions, applying vertical scaling, and the VNFIs are migrated so as to reduce the number of servers turned on and consequently the energy consumption; the number of migrations and the VNFIs to migrate are chosen by a policy whose objective is to balance the energy saving and revenue loss due to the disruption caused by the migration. The inputs to RLARCDP are the outputs of RVPP, the traffic matrix progression over time and the costs; the output is a migration policy.

In this paper we propose and investigate some algorithms to solve the RVPP and RLARCDP as illustrated in Fig. 1.b. We have studied in [28] the optimal RVPP and a heuristic, referred to as Maximizing the Accepted SFC Requests Number (MASRN), has been proposed. A new version of the algorithm called Maximum Accepted SFC Bandwidth (MASB) is introduced in Section V. While MASRN has the objective of minimizing the number of SFCs rejected, MASB is defined so as to minimize the bandwidth rejected. Furthermore MASB provides more efficient bandwidth utilization by allowing the mapping of service functions of a same SFC on the same server.

The optimal RLARCDP is studied in Subsection VI-A where an Integer Linear Programming (ILP) formulation is given in cycle-stationary traffic scenario with N stationary intervals ($N = 24$ is the typical value for daily traffic). Because the problem is NP-hard, we propose a solution when a number N of VNFI mappings, each one characterized by a particular allocation of the VNFIs to the servers, is chosen; each one of the N mappings is determined so as to minimize the energy consumption in the corresponding stationary interval. The proposed solution is based on the definition of two algorithms:

- VNFI Mapping Minimizing the Power Consumption (VMMPC) algorithm: its objective is to evaluate the VNFI mapping minimizing the energy consumption in each stationary interval; the operation mode of the VMMPC is described in Subsection VI-C
- Revenue Loss Aware Choosing Mapping (RLACM) algorithm: its objective is to choose the mapping to be applied in each stationary interval; it is based on Markov Decision Process theory and it chooses the mapping,evaluated in the overall time horizon, so as to minimize the cost of the energy consumption and the revenue loss due to the QoS degradation occurring when migrations are performed to change the mapping from a stationary interval to another one. RLACM is described in Subsection VI-B.

IV. Network and Traffic Model

Next we introduce the main terminology used to represent the physical network, VNF and the SFC traffic request [41].

A. Physical Network

We represent the physical network as a directed graph $G^{PN} = (V^{PN}, E^{PN})$ where V^{PN} and E^{PN} are the sets of physical nodes and links respectively. The set V^{PN} of nodes is given by the union of the three node sets V_A^{PN}, V_R^{PN} and V_S^{PN} that are referred to access, switching and server nodes respectively. The server nodes and links are characterized by:

- $N_{core}^{PN}(w)$: processing capacity of the server node $w \in V_S^{PN}$ in terms of number of cores available;
- $C^{PN}(d)$: bandwidth of the physical link $d \in E^{PN}$;

B. VNF and SFC Traffic Requests

We assume that F types of VNFs can be provisioned as firewall, IDS, proxy, load balancers, $\cdots$. We denote with $F^{VNF} = \{f_1, f_2, \cdots, f_F\}$ the set of VNF types, f_i being the $i-th$ VNF type. The packet processing time of the VNF f_i is denoted with t_i^{proc} $(i = 1, \cdots, F)$.

We also assume that the network operator is receiving T Service Function Chain (SFC) requests known in advance. The $i-th$ SFC request $(i = 1, \cdots, T)$ is characterized by the graph $G_i^{SFC} = (V_i^{SFC}, E_i^{SFC})$ where V_i^{SFC} represents the set of access and VNF nodes and E_i^{SFC} $(i = 1, \cdots, T)$ denotes the links between them. In particular the set V_i^{SFC} is given by the union of $V_{i,A}^{SFC}$ and $V_{i,F}^{SFC}$ denoting the set of access and VNFs nodes respectively.

We assume cycle-stationary traffic conditions because it is well known that data-center traffic matrices exhibit strong diurnal patterns and are typically cycle-stationary [40], [42], [43]. We denote with N the number of stationary intervals after which the same traffic characteristic occurs again. We denote the duration of a stationary interval with Δt and we assume that in the $h-th$ $(h = 0, 1, \cdots, N-1)$ stationary interval the traffic state S_h occurs and characterized by the values of bandwidth $BW_{h,i}$ $(i = 1, \cdots, T)$ of the T offered SFCs.

Though the proposed algorithms in the next sections could be defined in the general case, we start from the assumptions adopted in the main papers proposed in literature [10] according to which the VNFs request only processing resources, while memory and disk resources are not considered; the assumption will allow us to investigate the case studies in which CPU-intensive appliances are used while the investigation for the case of appliances that need allocation of bandwidth and disk are left as research future item.

Next we introduce the following notations:

- α_{vw}, it assumes the value 1 if the SFC access node $v \in \cup_{i=1}^{T} V_{i,A}^{SFC}$ characterizes an SFC request starting/terminating from/to the physical access nodes $w \in V_A^{PN}$; otherwise its value is 0;
- β_{vk}, it assumes the value 1 if the VNF node $v \in \cup_{i=1}^{T} V_{i,F}^{SFC}$ needs the application of the VNF f_k $(k \in [1..F])$; otherwise its values is 0;
- $B_h^{SFC}(v)$, the processing capacity requested by the VNF node $v \in \cup_{i=1}^{T} V_{i,F}^{SFC}$ in the traffic state S_h $(h = 0, 1, \cdots, N-1)$; the parameter value depends on both the packet length L_p and the bandwidth of the packet flow incoming to the VNF node; its values equals $\sum_{k=1}^{F} \beta_{vk} BW_{h,i} t_k^{proc} / L_p$ if $v \in V_i^{SFC}$ that is if the node v belongs to the graph $G_i^{SFC} = (V_i^{SFC}, E_i^{SFC})$ of the $i-th$ SFC offered;
- $C_h^{SFC}(e)$: bandwidth requested by the link $e \in \cup_{i=1}^{T} E_i^{SFC}$ in the traffic state S_h $(h = 0, 1, \cdots, N-1)$; its value equals $BW_{h,i}$ if $e \in E_i^{SFC}$ that is if the edge e belongs to the graph $G_i^{SFC} = (V_i^{SFC}, E_i^{SFC})$ of the $i-th$ SFC offered.

Next we will assume that $h = 0$ denotes the index of the traffic state (S_0) related to the Peak Hour Interval (PHI) in which the offered traffic is maximum.

V. SFC Routing and VNF Instance Dimensioning Problem

The Routing and VNF Placement Problem (RVPP) illustrated in Fig. 1.a consists in determining the servers in which the VNF nodes are executed and the network paths in which the virtual links of the SFC are routed. The arrangement is performed by taking into account the Peak Hour Interval (PHI) traffic demand and without violating both the server processing and physical link capacities. The objective of RVPP is to minimize the fraction of total bandwidth rejected. We assume that all of the servers create a VNF instance for each type of VNF that will be shared by the SFCs using that server and requesting that type of VNF. For this reason each server will activate F VNF instances, one for each type of VNF. The solution of RVPP also leads to evaluate the number of cores to be assigned to each VNF instance.

We have reported in [28] the optimal formulation of the problem in the case in which the objective is to minimize the number of SFC requests rejected. The introduced optimization problem is intractable because it requires solving a NP-hard bin packing problem [29]. Due to its high complexity, it is not possible to solve the problem directly in a timely manner given the large number of servers and network nodes. For this reason we have proposed an efficient heuristic referred to as the Maximizing the Accepted SFC Requests Number (MASRN) heuristic. In this paper we propose a modified version of MASRN, referred to as Maximizing the Accepted SFC Bandwidth (MASB). The main differences between MASRN and MASB are the following: i) MASB has the objective to minimize, instead of the number of SFC requests rejected, the fraction of total SFC bandwidth rejected; ii) MASB overcomes the main limit of MASRN that maps the VNFs of a same SFC in different servers; conversely MASB allows for the assignment of the processing resources of a same server to more than one VNF of an SFC when it is needed; this greater flexibility of MASB leads to lower blocking in the scenario with limited link bandwidth and high server processing capacity.

Next we briefly report the main steps of the MASB algorithm that is applied before the phase in which the resources are consolidated and a migration policy is applied. Next we describe the inputs, the outputs, the variables and the operation mode of the MASB algorithm.

The inputs of the MASB algorithm are:

- G^{PN}, the graph representing the physical nodes and links;
- P, set of all the paths in G^{PN};
- δ_{ep}, the binary function assuming value 1 or 0 if the network link e belongs or not to the path $p \in P$ respectively;
- G_i^{SFC}, $(i = 1, \cdots, T)$, the graphs representing the SFCs that will be handled in decreasing order of offered bandwidth;

The outputs of the MASB algorithm are:

- x_i, binary variable assuming the value 1 if the $i-th$ SFC request is accepted; otherwise its value is zero;
- y_{wk}, integer variable characterizing the number of cores allocated to the VNF instance of type k in the server $w \in V_S^{PN}$;
- z_{vw}^k, binary variable assuming the value 1 if the VNF node $v \in \cup_{i=1}^T V_{i,F}^{SFC}$ is served by the VNF instance of type k in the server $w \in V_S^{PN}$;
- u_{dp}, binary variable assuming the value 1 if the virtual link $d \in \cup_{i=1}^T E_i^{SFC}$ is embedded in the physical network path $p \in P$; otherwise its value is zero;

The main variables of the MASB algorithm are:

- $A^k(w)$, the load of the cores allocated for the VNF instance of type $k \in [1..F]$ in the server $w \in V_S^{PN}$;
- $S_N(w)$, defined as the stress of the node $w \in V_S^{PN}$ [30] and characterizing the server load; its value is tied to the variables $A^k(w)$ according to the following expression:

$$S_N(w) = \sum_{k=1}^{F} A^k(w) \tag{1}$$

- $S_L(e)$, defined as the stress of the physical network link $e \in E^{PN}$ [30] and characterizing the link load.
- χ^{server}: average server utilization efficiency given by the average ratio of the server stress to the number of cores allocated to each server;
- χ^{link}: average link utilization efficiency given by the average ratio of the link stress to the link capacity;

The main steps of the MASB algorithm are reported in the flow chart of Fig. 2. In the first phase the variables $A^k(w)$, $S_N(w)$ and $S_L(e)$ are initialized. Next the SFCs are sorted in bandwidth decreasing order. For the $i-th$ SFC offered, one of the following two mapping procedures of the graph G_i^{SFC} is performed. If χ^{server} is lower than χ^{link}, more processing resources are available and the MASB algorithm chooses the least loaded server (lowest stress) as candidate in which to map the entire SFC. Conversely if more bandwidth resources than the processing ones ($\chi^{server} \geq \chi^{link}$) are available, the candidate mapping, that is a set of servers in which to host the service functions of the SFC, is chosen according to the MASRN algorithm. The determination of the servers is based on the potential approach proposed in [30]. In such a way the mapping is chosen so as to balance the use of the server and link resources.

Once evaluated the mapping, the algorithm checks if resources are actually available. If they are not, the SFC request is rejected otherwise it is accepted and the outputs z_{vw}^k and u_{dp} are updated. In particular notice that if it has been decided to map the entire SFC in one server only and processing resources are not available, the SFC request is rejected. In the updating phase the node and link stresses $S_N(w)$ and $S_L(e)$ respectively are updated as well as the core loads $A^k(w)$ allocated to the VNFs, the number y_{wk} of cores used, the average server utilization χ^{server} and the average link utilization χ^{link}. When all of the SFCs requests have been considered the algorithm ends by reporting its outputs.

The computational complexity of MASB is the same of MASRN [28] and it is given by $O(F(F+N_s)N_l log(N_s+N_n))$ where F is the maximum number of VNFs in an SFC and N_s, N_n, N_l are the number of servers, switches and links of the network respectively.

VI. Energy and Reconfiguration Cost Aware VNF Instance Allocation Problem

When traffic reduction occurs, the VNF Instance can be dimensioned with a number of cores lower than the one evaluated in Section V during the Peak Hour Interval and consolidation techniques can be applied by migrating VNFIs so as to occupy fewer servers and to save power consumption. The migration can be performed when the VNFIs are supported by Virtual Machines (VM) but at the price of a information loss when the VMs are moved. Because this loss may impact the Quality of Service offered by a network operator, VNFI migration policies have to be defined that allow for a right compromise between power consumption saving and QoS degradation due to migration. The Integer Linear Programming (ILP) formulation of the optimal Revenue Loss Aware Resource Consolidation/De-consolidation Problem (RLARCDP) mentioned in Section III is illustrated in Subsection VI-A. Due to the NP-hard complexity, we solve it when a-priori mappings are selected. For this reason we propose the Revenue Loss Aware Choosing Mapping (RLACM) algorithm, illustrated in Subsection VI-B, based on the Markov Decision Process theory and whose objective is to minimize in cycle-stationary time interval the total cost given by the energy cost and the revenue loss due to the bits lost during the migration. The RLACM algorithm is based on the choice of N mappings, one for each stationary interval. The mappings are determined according to the VNFI Mapping Minimizing the Power Consumption (VMMPC) algorithm reported in Subsection VI-C, that minimizes the energy consumption in each time interval.

A. The ILP Formulation of the Optimal Revenue Loss Aware Resource Consolidation/De-Consolidation Problem (RLARCDP)

Though the optimal problem can be defined in general traffic scenarios, we focus its definition in cycle-stationary traffic conditions. The procedure illustrated in Section V identifies the VNFIs and the interconnection among them and the access nodes. For this reason we can introduce the new graph $G^{VNFI} = (V^{VNFI}, E^{VNFI})$ where V^{VNFI} represents the set of traffic nodes (access and VNFI nodes) and E^{VNFI} denotes the links between them. In particular the set V^{VNFI} is given by the union of V_A^{VNFI} and V_F^{VNFI} denoting the set of access and VNFIs nodes respectively. The introduced graph will be characterized by parameters whose values are dependent on the traffic state S_h $(h = 0, 1, \cdots, N-1)$. The main parameters are:

- $B_h^{VNFI}(v)$: number of cores required by the VNFI node $v \in V_F^{VNFI}$ in the traffic state S_h $(h = 0, 1, \cdots, N-1)$;
- $C_h^{VNFI}(d)$: bandwidth requested by the logical link $d \in E^{VNFI}$ in the traffic state S_h $(h = 0, 1, \cdots, N-1)$;

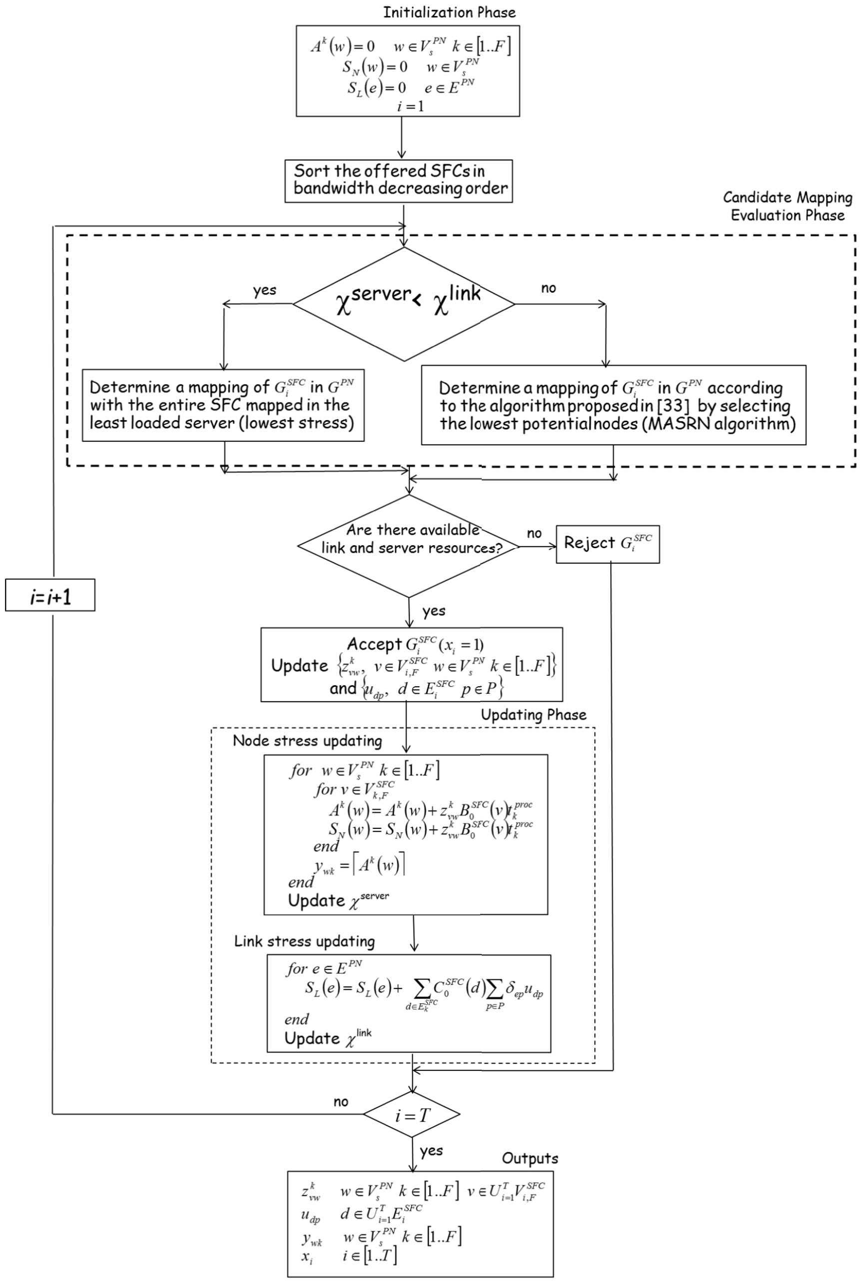


Fig. 2. The main steps of the MASB algorithm.

- $T_h^{VNFI}(v) = \sum_{d \in I(v)} C_h^{VNFI}(d)$: sum of the incoming link bandwidths in the VNFI node $v \in V_F^{VNFI}$ in the traffic state S_h $(h = 0, 1, \cdots, N-1)$; $I(v)$ denotes the set of incoming links to the node $v \in V_F^{VNFI}$;
- $a^{VNFI}(d)$ and $b^{VNFI}(d)$: origin and destination nodes of the link $d \in E^{VNFI}$;
- ϵ_{vw}, it assumes the value 1 if the access node $v \in V_A^{VNFI}$ has to be mapped in the physical access node $w \in V_A^{PN}$; otherwise its value is 0.
- δ_{ep}, the binary function assuming value 1 or 0 if the network link e belongs or not to the path $p \in P$ respectively;

After introducing the main notations, we formulate the Optimal RLARCDP when the objective function to be minimized is composed by two components: i) the static one represented by the energy consumption in the states S_h $(h = 0, 1, \cdots, N-1)$; ii) the reconfiguration one occurring during the state transitions and due to the revenue loss for the information loss when VNF Instance are migrated.

The solution to the optimization problem is characterized by the following variables:

- σ_{vw}^h, binary variable assuming the value 1 if the VNFI node v is embedded in the server node w in the state S_h $(h = 0, 1, \cdots, N-1)$; otherwise its value is zero;
- η_{dp}^h, binary variable assuming the value 1 if the virtual link d is routed on the physical network path p in the state S_h $(h = 0, 1, \cdots, N-1)$; otherwise its value is zero.

Two other variables are introduced to give a linear formulation of the problem:

- λ_w^h, binary variable assuming the value 1 if the server node w is switched on in the state S_h $(h = 0, 1, \cdots, N-1)$; otherwise its value is zero;
- ξ_v^h, binary variable assuming the value 1 if the node v is migrating when a state change from S_h to S_{h+1} occurs; otherwise its value is zero;

The introduced variables have to satisfy the following constraints of the optimization problem:

$$\sum_{w \in V_S^{PN}} \sigma_{vw}^h = 1 \quad h \in [0..N-1] \; v \in V_F^{VNFI} \tag{2}$$

$$\sum_{p \in P} \eta_{dp}^h = 1 \quad h \in [0..N-1] \; d \in E^{VNFI} \tag{3}$$

$$\eta_{dp}^h \le \sigma_{a^{VNFI}(d)a^{PN}(p)}^h \quad a^{VNFI}(d) \in V_F^{VNFI} \quad h \in [0..N-1] \quad p \in P \tag{4}$$

$$\eta_{dp}^h \le \sigma_{b^{VNFI}(d)b^{PN}(p)}^h \quad b^{VNFI}(d) \in V_F^{VNFI} \quad h \in [0..N-1] \quad p \in P \tag{5}$$

$$\eta_{dp}^h \le \epsilon_{a^{VNFI}(d)a^{PN}(p)}^h \quad a^{VNFI}(d) \in V_A^{VNFI} \quad h \in [0..N-1] \quad p \in P \tag{6}$$

$$\eta_{dp}^h \le \epsilon_{b^{VNFI}(d)b^{PN}(p)}^h \quad b^{VNFI}(d) \in V_A^{VNFI} \quad h \in [0..N-1] \quad p \in P \tag{7}$$

$$\sum_{v \in V_S^{VNFI}} \sigma_{vw}^h B_h^{VNFI}(v) \le N_{core}^{PN}(w) \quad h \in [0..N-1] \quad w \in V_S^{PN} \tag{8}$$

$$\sum_{d \in E^{VNFI}} C_h^{VNFI}(d) \sum_{p \in P} \delta_{ep} \eta_{dp}^h \le C^{PN}(e) \quad h \in [0..N-1] \quad e \in E^{PN} \tag{9}$$

$$\lambda_w^h \le \sum_{v \in V_S^{VNFI}} \sigma_{vw}^h \quad h \in [0..N-1] \; w \in V_S^{PN} \tag{10}$$

$$\lambda_w^h \ge \frac{1}{| V_S^{VNFI} |} \sum_{v \in V_S^{VNFI}} \sigma_{vw}^h \quad h \in [0..N-1] \quad w \in V_S^{PN} \tag{11}$$

$$\xi_v^h \ge \sigma_{vw}^h - \sigma_{vw}^{(h+1) \bmod N} \quad h \in [0..N-1] \quad w \in V_S^{PN} \tag{12}$$

In particular constraint (2) guarantees that each VNFI is assigned to exactly one server node. The routing of one virtual link of the graph $G^{VNFI} = (V^{VNFI}, E^{VNFI})$ in only one network path is expressed by (3). Constraints (4)-(7) establish that when the virtual link d is routed on the physical network path p then $a^{PN}(p)$ and $b^{PN}(p)$ must be the physical network nodes that the virtual nodes $a^{VNFI}(d)$ and $b^{VNFI}(d)$ are assigned to. Finally constraint (8) guarantees that the total number of the cores occupied by the VNFIs assigned to the server node $w \in V_S^{PN}$ is smaller than or equal to the available number $N_{core}^{PN}(w)$ of cores in that server. Constraint (9) guarantees that the total bandwidth carried by the physical network link $e \in E^{PN}$ is lower than its capacity $C^{PN}(e)$. Constraints (10-11) establish that any server is switched on when it hosts at least one VNFI. Finally constraint (12) establishes that a VNFI is migrating when it is not hosted by the same server in the traffic states S_h and $S_{(h+1) \bmod N}$.

The objective function C^{tot} to be minimized depends on two components: the first one expresses the energy consumption costs in the state S_h $(h = 0, 1, \cdots, N-1)$ and the second one expresses the reconfiguration costs involved when VNF instances are migrating during the traffic state transitions. Hence we can report the following expression for C^{tot}:

$$C^{tot} = \sum_{h=0}^{N-1} (C_{EC}^h + C_R^{h,(h+1) \bmod N}) \tag{13}$$

where C_{EC}^h is the energy consumption cost in the state S_h and $C_R^{h,(h+1) \bmod N}$ is the reconfiguration cost when the transition from the state S_h to the state $S_{(h+1) \bmod N}$ occurs. The reconfiguration cost $C_R^{h,(h+1) \bmod N}$ is characterized by the revenue loss of a network operator for the information loss due to the VNF Instance migration from the state S_h to the state $S_{(h+1) \bmod N}$. Indeed when any migration happens, the Virtual Machine supporting the VNF instance is not able to carry on its function in a critical period T_{down} referred to in the literature as downtime of the Virtual Machine [31].

For the evaluation of C_{EC}^h we assume a linear model [32] of the server power consumption versus the traffic handled by the server. A fixed power contribution is also considered when traffic is absent. Hence we assume the following expression of the power consumption P_S^w of the server w:

$$P_S^w(\varsigma_c) = P_{idle} + (P_{max} - P_{idle})\varsigma_c \tag{14}$$

wherein P_{max} and P_{idle} are the server power consumption in maximum load condition and when traffic is absent respectively; ς_c is the core load normalized to the number of cores.

As reported in [32], the server power consumption should be modeled with a function more complex than the linear one and it should be approximated with a polynomial function at least of seven degree; the assumption of a linear model is needed in order to give an ILP formulation of the optimal RLARCDP and compare the results to the ones of the heuristic proposed in Subsection VI-B. The RLACM/VMMPC heuristic introduced in Subsections VI-B and VI-C respectively, is flexible and it would allow us to include more complex power consumption models.

The assumption of linear server power consumption model allows us to write the following expression for C_{EC}^h:

$$C_{EC}^h = \beta_e \Delta t (P_{idle} \sum_{w \in V_S^{PN}} \lambda_w^h + \sum_{w \in V_S^{PN}} \frac{\sum_{v \in V_S^{VNFI}} \sigma_{vw}^h T_h^{VNFI}(v) t_v^{proc} / L_{max}}{N_{core}^{PN}(w)}) \tag{15}$$

where β_e denotes the cost per consumed power watt, L_p is the packet length and t_v^{proc} is the processing time of the packets handled by the VNFI $v \in V_S^{VNFI}$. The expression (15) adds the power contribution of the switched on server nodes.

The reconfiguration cost $C_R^{h,(h+1) \bmod N}$ takes into account the cost due to the amount of lost Gbit during the downtime

T_{down} for the migrant VNF Instances when a reconfiguration occurs during the transition from the state S_h to the state $S_{(h+1) \bmod N}$. If the parameter β_d characterizes the revenue loss of a network operator per one Gbit of lost traffic, we can write:

$$C_R^{h,(h+1) \bmod N} = \beta_d T_{down} \sum_{v \in V_S^{VNFI}} T_{(h+1) \bmod N}^{VNFI}(v) \xi_v^h \tag{16}$$

Summation in (16) adds the total bandwidth handled by the migrant VNF instances ($\xi_v^h \neq 0$).

To prove that RLARCDP is strongly NP-hard it is sufficient to consider the case of one stationary interval (N=1) and infinite link bandwidth. In such a case the RLARCDP reduces to the Multi-dimensional Bin Packing Problem that [33], Garey and Graham [34] have shown to be strongly NP-hard. Due to the high complexity of the RLARCDP, we propose the Revenue Loss Aware Choosing Mapping (RLACM) algorithm described in Subsection VI-B in which the problem is solved when a-priori mappings are chosen. These mappings are evaluated with the application of the VMMPC algorithm described in Subsection VI-C.

B. Revenue Loss Aware Choosing Mapping (RLACM) Algorithm

The objective of the Revenue Loss Aware Choosing Mapping (RLACM) algorithm is to solve the RLARCDP in the condition in which a-priori mappings are chosen, being a mapping the assignment of the VNF nodes and links of the graph $G^{VNFI} = (V^{VNFI}, E^{VNFI})$ to server nodes and physical network paths respectively.

The performance of the RLACM algorithm will be evaluated in the case in which the N mappings belonging to the set $\Theta = \{\theta_0, \cdots, \theta_h, \cdots, \theta_{N-1}\}$ are chosen, being θ_h the mapping minimizing the energy consumption in the traffic state S_h. The determination of these mappings will be evaluated according to the VNFI Mapping Minimizing the Power Consumption (VMMPC) algorithm of Subsection VI-C. The output of the RLACM algorithm is a migration policy characterized by the choice of the mapping $M_h \in \Theta$ to be applied in the state S_h ($h = 0, 1, \cdots, N-1$) so as to minimize the objective function expressed by (13). Not all of the mappings in Θ can be applied in any state S_h but only the admissible ones that satisfy the conditions expressed by Eqs (8) and (9). We denote with $\theta_{h,l}$ ($l = 1, \cdots, n_h$) the n_h mappings belonging to Θ and admissible for the traffic condition S_h ($h = 0, \cdots, N-1$).

We report a possible representation of the admissible states in Fig. 3 in the case of cycle-stationary traffic with $N = 4$. The states are organized in N levels and the $h-th$ level ($h = 0, 1, 2, 3$) report the mappings admissible for the state S_h. In this case we have $n_0 = 1$, $n_1 = 2$, $n_2 = 3$ and $n_3 = 2$ admissible states for the traffic states S_0, S_1, S_2 and S_3 respectively. The objective is to determine a policy that establishes which mapping to apply when traffic changes happens. Formally a policy is characterized by the set of integer values $\mathcal{D} = \{d_{h,l} h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h\}$ where

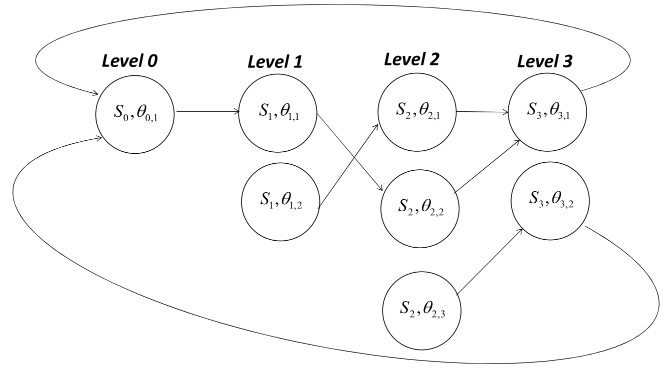


Fig. 3. An example of mapping policy characterized by $\mathcal{D} = \{d_{0,1} = 1, d_{1,1} = 2, d_{1,2} = 1, d_{2,1} = 1, d_{2,2} = 1, d_{2,3} = 2, d_{3,1} = 1, d_{3,2} = 1\}$.

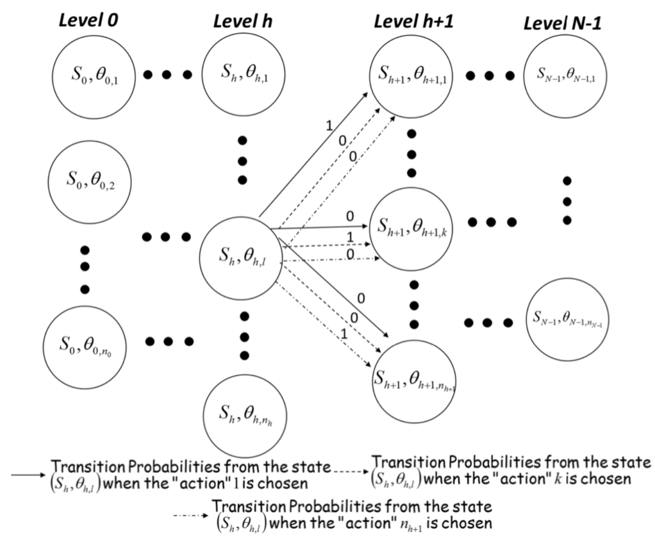


Fig. 4. Bi-dimensional Discrete Time Markov Chain in which each state is characterized by some possible *actions*; the choice of an *action* in each state determines a mapping policy.

$d_{h,l}$ establishes that, from the state $(S_h, \theta_{h,l})$, the mapping $\theta_{(h+1) \bmod N, d_{h,l}}$ has to be applied when the new traffic condition $S_{(h+1) \bmod N}$ occurs. We can represent the policy $\mathcal{D}$ in the case of Fig. 3 with arrows between states. In this case the policy is expressed by the set of integer values $\mathcal{D} = \{d_{0,1} = 1, d_{1,1} = 2, d_{1,2} = 1, d_{2,1} = 1, d_{2,2} = 1, d_{2,3} = 2, d_{3,1} = 1, d_{3,2} = 1\}$.

Our objective is to determine the policy $\mathcal{D}^G$ that minimizes the total cost in a cycle-stationarity period. We can determine the policy $\mathcal{D}^G$ by finding the optimal policy in a Discrete Time Markov Decision Process (DTMDP) [24], [35].

An MDP is characterized by a four-tuple (Ω_S,Ω_A,Ω_P,Ω_C) [35] where Ω_S is the state set, Ω_A is the action set, Ω_P is the probability set and Ω_C is the cost set.

The state set Ω_S is characterized by the bi-dimensional states $(S_h, \theta_{h,l})$ ($h = 0, \cdots, N-1; l = 1, \cdots, n_h$) and represented in Fig. 4 for the general case.

The action set Ω_A contains all of the possible actions; in particular the actions for the state $(S_h, \theta_{h,l})$ are characterized

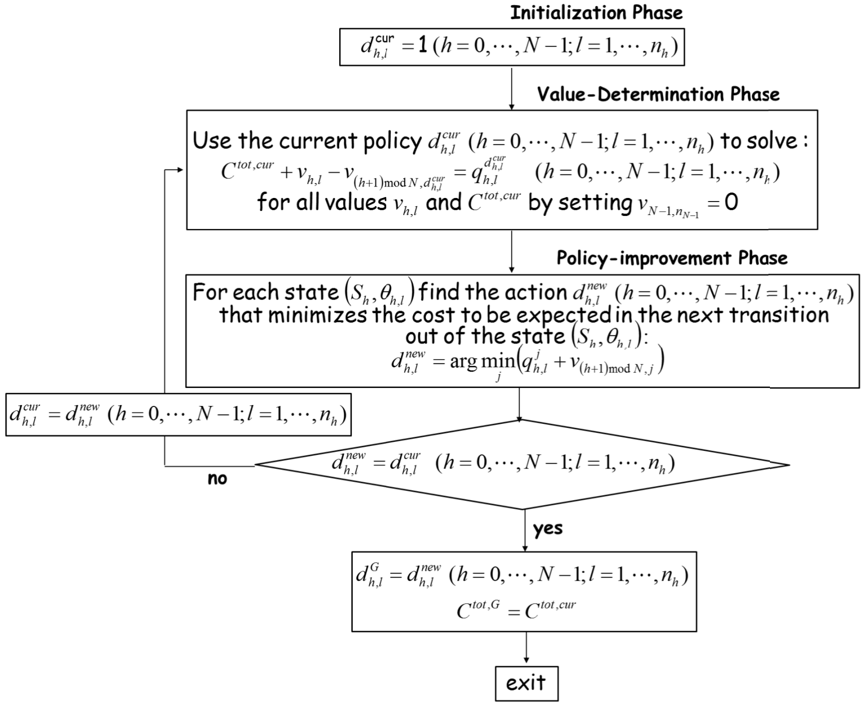


Fig. 5. Flow diagram of the policy-iteration method [24], [35] for the evaluation of the policy based on the cost minimization in a cycle-stationary interval.

by the integer variables $d_{h,l} \in [1..n_{(h+1) \bmod N}]$, being $n_{(h+1) \bmod N}$ the number of actions for that state.

The probability set Ω_P is characterized by the transition probabilities $p_{h,l}^{j,k,d_{h,l}}$ from the state $(S_h, \theta_{h,l})$ to the state $(S_j, \theta_{j,k})$ when the action $d_{h,l}$ is applied. The transition probabilities $p_{h,l}^{j,k,d_{h,l}}$ are reported in Fig. 4 for the *actions* $d_{h,l}$=1, $d_{h,l}$=k and $d_{h,l}$=$n_{(h+1) \bmod N}$. We can notice how the choice of the *action* $d_{h,l}$ involves only a transition with probability 1 from the state $(S_h, \theta_{h,l})$ to the state $(S_{(h+1) \bmod N}, \theta_{(h+1) \bmod N, d_{h,l}})$. Thus we can simply write:

$$p_{h,l}^{j,k,d_{h,l}} = \begin{cases} 1 & if\ j = (h+1) \bmod N,\ k = d_{h,l} \\ 0 & otherwise \end{cases} \tag{17}$$

The cost set Ω_C is characterized by the costs $q_{h,l}^{d_{h,l}}$ to be expected versus $d_{h,l}$ in the transition out of the state $(S_h, \theta_{h,l})$. For the evaluation of $q_{h,l}^{d_{h,l}}$ we observe that if the choice of the *action* $d_{h,l}$ in the state $(S_h, \theta_{h,l})$ leads not to change the applied mapping, that is $\theta_{(h+1) \bmod N, d_{h,l}} \equiv \theta_{h,l}$, the reconfiguration costs are not involved and only the energy consumption cost $C_{EC}(\theta_{h,l})$ of applying the mapping $\theta_{h,l}$ has to be considered. Otherwise when a mapping change occurs, also the reconfiguration cost $C_R(\theta_{h,l}, \theta_{(h+1) \bmod N, d_{h,l}})$ from the mapping $\theta_{h,l}$ to the mapping $\theta_{(h+1) \bmod N, d_{h,l}}$ has to be added. Hence we can write:

$$q_{h,l}^{d_{h,l}} = \begin{cases} C_{EC}(\theta_{h,l}) \\ \qquad\qquad if\ \theta_{(h+1) \bmod N, d_{h,l}} \equiv \theta_{h,l} \\ C_{EC}(\theta_{h,l}) + C_R(\theta_{h,l}, \theta_{(h+1) \bmod N, d_{h,l}}) \\ \qquad\qquad otherwise \end{cases} \tag{18}$$

Once established the transition probabilities $p_{h,l}^{j,k,d_{h,l}}$ and the cost $q_{h,l}^{d_{h,l}}$ as a function of the *action* $d_{h,l}$ we can find the policy $\mathcal{D}^G = \{d_{h,l}^G, h = 0, 1 \cdots, , N-1; l = 1, 2, \cdots, n_h\}$ by applying the policy-iteration method [24], [35] whose key steps are reported in the flow diagram of Fig. 5. The procedure begins with the initialization phase in which the current policy $\mathcal{D}^{cur} = \{d_{h,l}^{cur} = 1\ h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h\}$ is chosen. The determination of the relative values $v_{h,l}$ $(h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h)$ [24] occurs in the Value-Determination phase in which a linear system is solved allowing the evaluation of the relative values as well as the cost $C^{tot,cur}$ of the policy $\mathcal{D}^{cur}$; it is proved that the system has ∞^1 solution and for this reason one of the relative values, in our case $v_{N-1,n_{N-1}}$, has to be set to zero. In the policy-improvement phase, the new policy $\mathcal{D}^{new} = \{d_{h,l}^{new}, h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h\}$ is evaluated as improvement of the policy $\mathcal{D}^{cur}$ by using the relative values evaluated in the previous step. It is proved that the cost of $\mathcal{D}^{new}$ is smaller than or equal the one of $\mathcal{D}^{cur}$ [24], [35]. Finally in the end of this phase if the policies $\mathcal{D}^{new}$ and $\mathcal{D}^{cur}$ are different, a new iteration cycle is accomplished with $\mathcal{D}^{cur} = \{d_{h,l}^{new}, h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h\}$. Otherwise the procedure ends with the global policy given by $\mathcal{D}^G = \{d_{h,l}^{cur}, h = 0, 1, \cdots, N-1; l = 1, 2, \cdots, n_h\}$ and the total cost $C^{tot,glo}$ of the policy $\mathcal{D}^G$ is equal to $C^{tot,cur}$.

We conclude discussing the computational complexity of the RLACM algorithm. [36], Papadimitriou and Tsitsiklis [37] analyzed the computational complexity of MDPs and they showed that the problem is P-complete. Further they showed that for MDP with deterministic transition probabilities (all 0's and all 1's) as ours, the optimal cost can be found efficiently by implementing procedures executed in parallel; the algorithm they give is strongly polynomial.

C. VNFI Mapping Minimizing the Power Consumption (VMMPC) Algorithm

The VMMPC algorithm is applied to consolidate the server resource. It was inherited from the mapping algorithm proposed in [38] for the consolidation of virtual routers in physical nodes. The application of the VMMPC algorithm to the VNF context has the objective to consolidate server resources as much as possible in order to minimize the power consumption. It is introduced to evaluate the set $\Theta = \{\theta_0, \cdots, \theta_h, \cdots, \theta_{N-1}\}$ of the mappings chosen a-priori and used by the RLACM algorithm. Starting from the mapping Γ evaluated by the MASB algorithm in Section V, the VMMPC algorithm employs the VNFI instance migration to consolidate the server resources. The algorithm 1 reports the key steps of the VMMPC algorithm. The inputs of the algorithm are the physical network graph $G^{PN} = (V^{PN}, E^{PN})$, the VNFI graph $G^{VNFI} = (V^{VNFI}, E^{VNFI})$, the mapping Γ and the residual server and links resources still available after the execution of the MASB algorithm. The VMMPC algorithm is characterized by the following inputs:

- σ_{vw}^{peak}, binary variable assuming the value 1 if the VNFI node v is mapped in the server node w in the mapping Γ; otherwise its value is zero;
- η_{dp}^{peak}, binary variable assuming the value 1 if the virtual link d is routed on the physical network path p in the mapping Γ; otherwise its value is zero.
- $C_{r,peak}^{PN}(e)$, residual capacity of the physical network link $e \in E^{PN}$ after the mapping Γ has been evaluated;
- $N_{r,peak}^{PN,core}(w)$, residual cores number of the server node $w \in V_S^{PN}$ after the mapping Γ has been evaluated.

The output of the VMMPC algorithm is the mapping θ_h $(h = 0, \cdots, N-1)$ in the traffic state S_h $(h = 0, \cdots, N-1)$. In particular, we also apply the consolidation algorithm in the PHI (state S_0) in order to save energy consumption. This mapping is characterized by the following parameters:

- φ_{vw}^{h}, binary variable assuming the value 1 if the VNFI node v is mapped in the server node w for the mapping θ_h in the traffic state S_h; otherwise its value is zero;
- τ_{dp}^{h}, binary variable assuming the value 1 if the virtual link d is routed on the physical network path p for the mapping θ_h in the traffic state S_h; otherwise its value is zero.

The following variables are introduced in the algorithm 1:

- $C_r^{PN}(e)$, residual capacity of the physical network link $e \in E^{PN}$;
- $N_r^{PN,core}(w)$, residual cores number of the server node $w \in V_S^{PN}$.

Other variables with the subscript *aux* appearing in algorithm 1, are auxiliary variables only and are used when the change of a variable is not definitive yet.

The energy efficiency parameter $\eta(w)$ of the server node w is defined as the ratio of the server power consumption to the amount of bandwidth handled by VNFIs hosted by the

Algorithm 1 VMMPC Algorithm

1: **Input:** Physical Network Graph $G^{PN} = (V^{PN}, E^{PN})$; VNF Instance Graph $G^{VNFI} = (V^{VNFI}, E^{VNFI})$; $\{\sigma_{vw}^{peak}, v \in V_F^{VNFI}\ w \in V_S^{PN}\}$; $\{u_{dp}^{peak}, d \in E^{VNFI}\ p \in P\}$; $C_{r,peak}^{PN}(e), e \in E^{PN}$; $N_{r,peak}^{PN,core}(w), w \in V_S^{PN}$;
2: ***/*Initialization Phase*/***
3: $C_r^{PN}(e)$=$C_{r,peak}^{PN}(e), e \in E^{PN}$;
$N_r^{PN,core}(w)$=$N_{r,peak}^{PN,core}(w), w \in V_S^{PN}$;
$V_{aux,S}^{PN}$=V_S^{PN};
$\varphi_{vw}^{h} = \sigma_{vw}^{peak}, v \in V_F^{VNFI}\ w \in V_S^{PN}$;
$\tau_{dp}^{h} = u_{dp}^{peak}, d \in E^{VNFI}\ p \in P$
4: **while** $V_{aux}^{SN} \neq \emptyset$ **do**
5: **select** the server node $w \mid w = arg(\min_{s \in V_{aux,S}^{PN}} \eta(s))$
6: Ω=$V_{aux,S}^{SN} - \{w\}$
7: **while** $\Omega \neq \emptyset$ **do**
8: $C_{r,aux}^{PN}(e)$=$C_r^{PN}(e), e \in E^{PN}$;
$N_{r,aux}^{PN,core}(q)$=$N_r^{PN,core}(q), q \in V_S^{PN}$;
φ_{vq}^{aux}=$\varphi_{vq}^{h}, v \in V_F^{VNFI}\ q \in V_S^{PN}$;
τ_{dp}^{aux}=$\tau_{dp}^{h}, d \in E^{VNFI}\ p \in P$;
9: **select** the server node $z \mid z = arg(\max_{s \in V_{aux,S}^{PN}} \eta(s))$
10: ***/* Server Resource Availability Check Phase*/***
11: **if** $\sum_{v \in V_F^{VNFI}} \varphi_{vw}^{h} B_h^{VNFI}(v) \leq N_r^{PN,core}(z)$ **then**
12: $N_{r,aux}^{PN,core}(w) = N_{r,aux}^{PN,core}(w) - \sum_{v V_F^{VNFI}} \varphi_{vw}^{h} B_h^{VNFI}(v)$
13: **update** $\varphi_{vw}^{aux}, v \in V_F^{VNFI}\ w \in V_S^{PN}$;
14: **else**
15: **Go to line** 35
16: **end if**
17: ***/*Physical Network Paths Resource Deallocation Phase*/***
18: **determine** the set E_{aux}^{VNFI} of edges in E^{VNFI} involved in the migration
19: **for** $d \in E_{aux}^{VNFI}, p \in P, e \in E^{PN}$ **do**
20: $C_{r,aux}^{PN}(d)$=$C_{r,aux}^{PN}(d)$-$\tau_{dp}^{aux} \delta_{ep} C_h^{VNFI}(d)$
21: **end for**
22: ***/* Link Resource Availability Check Phase*/***
23: **for** $d \in E_{aux}^{VNFI}$ **do**
24: **determine** the shortest path p with end physical network node z in which to route the virtual links d
25: u_{dp}^{aux}=1
26: **end for**
27: **for** $d \in E_{aux}^{VNFI}, p \in P, e \in E^{PN}$ **do**
28: **if** $\tau_{dp}^{aux} \delta_{ep} C_h^{VNFI}(d) \leq C_{r,aux}^{PN}(e)$ **then**
29: $C_{r,aux}^{PN}(e)$=$C_{r,aux}^{PN}(e)$-$\tau_{dp}^{aux} \delta_{ep} C_h^{VNFI}(d)$
30: **else**
31: **Go to line** 35
32: **end if**
33: **end for**
34: **Go to line** 37
35: Ω=$\Omega - \{z\}$
36: **end while**
37: $N_r^{PN,core}(w)$=$N_{r,aux}^{PN,core}(w), w \in V_S^{PN}$;
$C_r^{PN}(e)$=$C_{r,aux}^{PN}(e), e \in E^{PN}$;
φ_{vw}^{h}=$\varphi_{vw}^{aux}, v \in V_F^{VNFI}\ w \in V_S^{PN}$;
τ_{dp}^{h}=$\tau_{dp}^{aux}, d \in E^{VNFI}\ p \in P$;
38: $V_{aux,S}^{PN}$=$V_{aux,S}^{PN} - \{w\}$
39: **end while**
40: **Output:** $\{\varphi_{vw}^{h}, v \in V_F^{VNFI}\ w \in V_S^{PN}\}$; $\{\tau_{dp}^{h}, d \in E^{VNFI}\ p \in P\}$

server w. The algorithm considers the server nodes one by one (line 4) in order increasing of $\eta(w)$ (line 5) and tries turning them off by migrating the VNF Instances in other server nodes chosen in decreasing order of $\eta(w)$ (line 9). Obviously the turning off is possible only if both the server node hosting the VNF instances has sufficient cores and the virtual links involved in the migration can be re-routed in new physical network paths with sufficient bandwidth. The two conditions are verified in the node (lines 11-16) and link (lines 23-33) availability check phases respectively. In the old shortest paths the resources are deallocated in the physical network paths resource deallocation phase (lines 18-21). Finally the outputs (line 40) of the VMMPC algorithm are the selected values for the parameters φ^h_{vw}, $\{v \in V_F^{VNFI}\ w \in V_S^{PN}\}$, τ^h_{dp}, $\{d \in E^{VNFI}\ p \in P\}$ characterizing the embedding θ_h.

Next we report a complexity analysis of the proposed algorithm. The complexity can be evaluated according to the following remarks: i) the VMMPC algorithm performs N_s steps (line 4) where at each step the least energy efficient server, among the switched on ones, is selected (line 5); the complexity of this operation is N_s^2; ii) the servers in which to try migrating a VNFI are selected in decreasing order of energy efficiency (line 9); the complexity of this operations is N_s; the rerouting of paths between VNFI is accomplished by using the Dijkstra whose complexity, when heap binary data structure is used, is $O(N_l log(N_s + N_n))$. According to these remarks the complexity of the VMMPC algorithm is $O(N_s^3 N_l log(N_s + N_n))$.

VII. Numerical Results

We will provide some results on the SFC blocking performance evaluated in the Peak Hour Interval and when the MASB algorithm introduced in Section V is applied. We will also verify the effectiveness of the VNFI migration policies introduced in Subsection VI-B by comparing the results of the RLACM/VMMPC algorithms to that achieved by solving the optimal RLARCDP.

A given number T of SFC requests are generated, each one randomly selected from the ones illustrated in Fig. 6: all of SFCs are composed by two access nodes and in addition we can have one Firewall (FW) only (Fig. 6.a), one FW and one Intrusion Detection System (IDS) (Fig. 6.b) and one FW, one IDS and one Encryption VPN (EV) (Fig. 6.c). In the considered case study we also assume that the SFC handles traffic flows of packet length equal to 1500 bytes. The evaluation is carried out according to the data reported in [12] where commercial appliances are used with the values of 120 μs, 160 μs and 82.76 μs for the packet processing times of the FW, IDS, EV VNFs respectively. The bandwidth of each SFC is selected among the values of the set [100Mbps, 150Mbps, 200Mbps, 250Mbps, 300Mbps] according to a Zipf distribution [39].

We consider the network reported in Fig. 7. It is composed by five core nodes, five edge nodes and six access nodes in which the SFC requests are randomly generated and terminated. Four Network Function Virtualization (NFV) sites are connected to the network. Each of them is composed by two

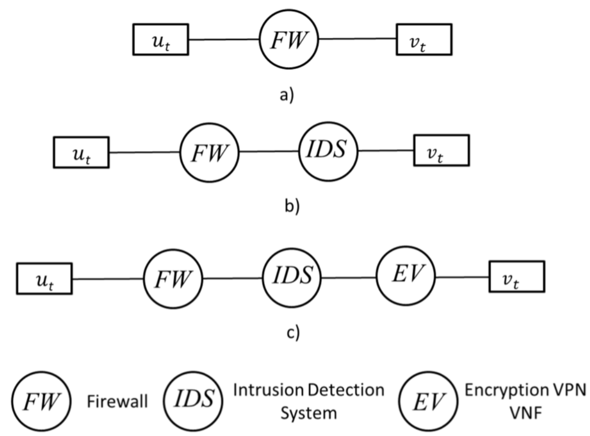


Fig. 6. Three possible alternatives of SFCs are generated. All of the three SFCs are composed by two access nodes u_t v_t; the first one is composed by one FW only (a), the second one is composed by one FW and one IDS (b) and the third one is composed by one FW, one IDS and one EV (c).

switches which eight servers are connected to. In the basic configuration, 40 Gbps links are considered except the links connecting the server to the switches in the NFV sites whose the bit rate is equal to 10 Gbps. The 64 servers are equipped with 48 cores each one.

Next we provide the results on the SFC blocking performance in the PHI and the cost evaluation of the migration policies in Section VII-A and VII-B respectively.

A. SFC Blocking Performance in the Peak Hour Interval

The objective of the study carried out in this section is twofold: i) to evaluate how much the MASB algorithm outperforms the old version of the algorithm (MASRN) [28] due to the possibility to embed more than one VNF of an SFC in the same server with the consequent advantage to improve the performance when the blocking is due to shortage of link bandwidth; ii) to compare the performance of MASRN and MASB to that one of the SFC routing and VNF placement algorithm proposed by Bari *et al.* [18] and based on both a-priori dimensioning of the VNFI in terms of cores and horizontal scaling techniques. The Bari's algorithm assumes that VNFIs of fixed processing capacity and consuming a given number of cores are a-priori established. The SFC routing and the VNF placement is not based on uniform resource occupancy and tries to instantiate the minimum number of VNFI by using as much as possible the ones already instantiated. Further it saves bandwidth by embedding as much as possible the VNFs of an SFC in the same server. The servers and the network paths in which to embed the SFC are chosen according to the following procedure: i) a multi-graph is built with a number of levels equal to the number of VNFs of the SFC; ii) each level is composed by nodes representing the servers with processing capacity available to host the corresponding VNF; each node of the graph is labeled with a cost given by the available processing capacity of the VNFI (already or to be instantiated) in the corresponding server; iii) an edge between two nodes of the graph is labeled with a cost given by the sum of the bandwidths to be allocated on the

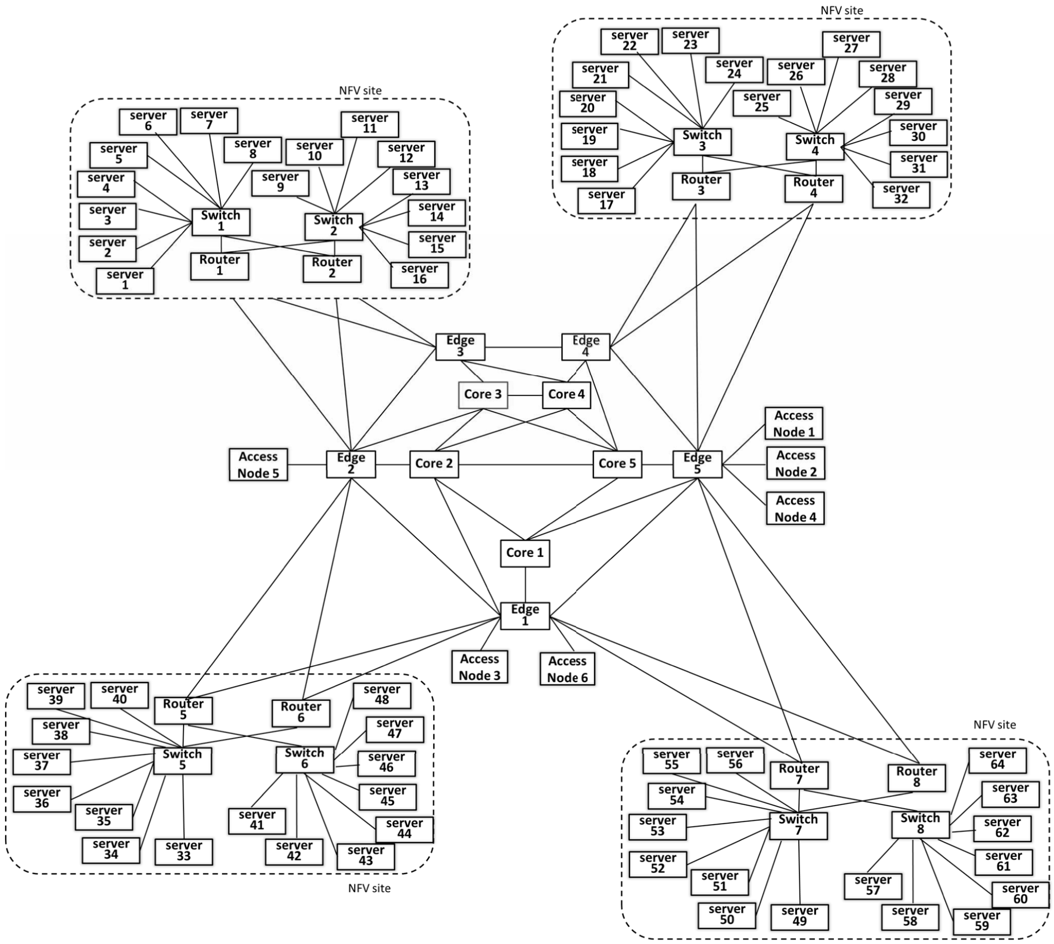


Fig. 7. The network topology is composed by six access nodes, five edge nodes, five core nodes and four NFV sites. Each NFV site is composed by two routers, two switches and sixteen servers. Eight servers are connected to each switch.

links of the shortest path between the corresponding servers; iv) the servers are chosen by applying the Viterbi algorithm in the multi-graph while the network paths are chosen to be the shortest paths connecting the servers. After the choice of the servers and network paths, the Bari's algorithm checks if processing and bandwidth resources are available.

We report in Fig. 8 the fraction of dropped bandwidth of the SFC offered as a function of the offered number of SFCs for the MASRN, MASB and Bari's algorithms. We show in Fig. 8 two sets of curves: the first one obtained in the basic scenario and the second one when the link bandwidth is scaled by a factor γ equal to 0.1. This second case study is introduced to investigate the performance of the algorithms in the case in which the SFC blocking is due to shortage of link bandwidth instead of node processing one. The average number of cores allocated in the 48 servers is illustrated in Fig. 9.a and Fig. 9.b in the basic and scaled scenario respectively. In the case of the Bari's algorithm the processing capacities assigned to the VNFI supporting FW, IDS and EV VNFs have been chosen equal to 400Mbps, 600Mbps and 580Mbps with the use of 4, 8 and 4 cores respectively [19]. From Fig. 8 we can observe how MASRN and MASB algorithms have similar performance and they outperform significantly the Bari's algorithm in the

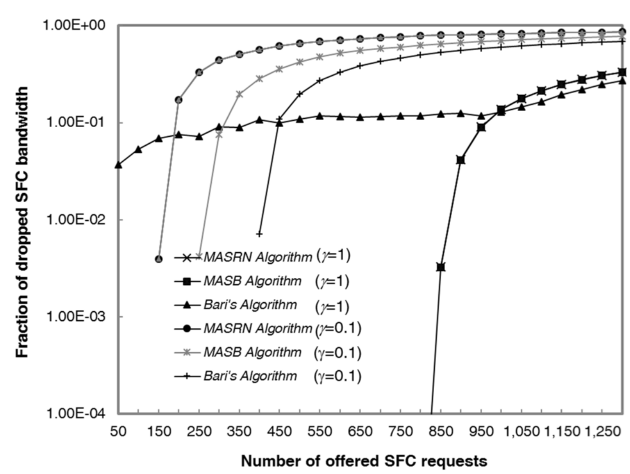


Fig. 8. Fraction of rejected bandwidth as a function of the number of SFC offered. The performance of the MASRN, MASB and Bari's algorithms are compared in the basic scenario ($\gamma = 1$) and in the case in which the link bandwidth is reduced to the 10% of the basic scenario ($\gamma = 0.1$).

basic scenario ($\gamma = 1$) when fewer than 1000 SFCs are offered; after this value the algorithms has performance comparable; as a matter of example, when 100 SFCs are offered, we achieve

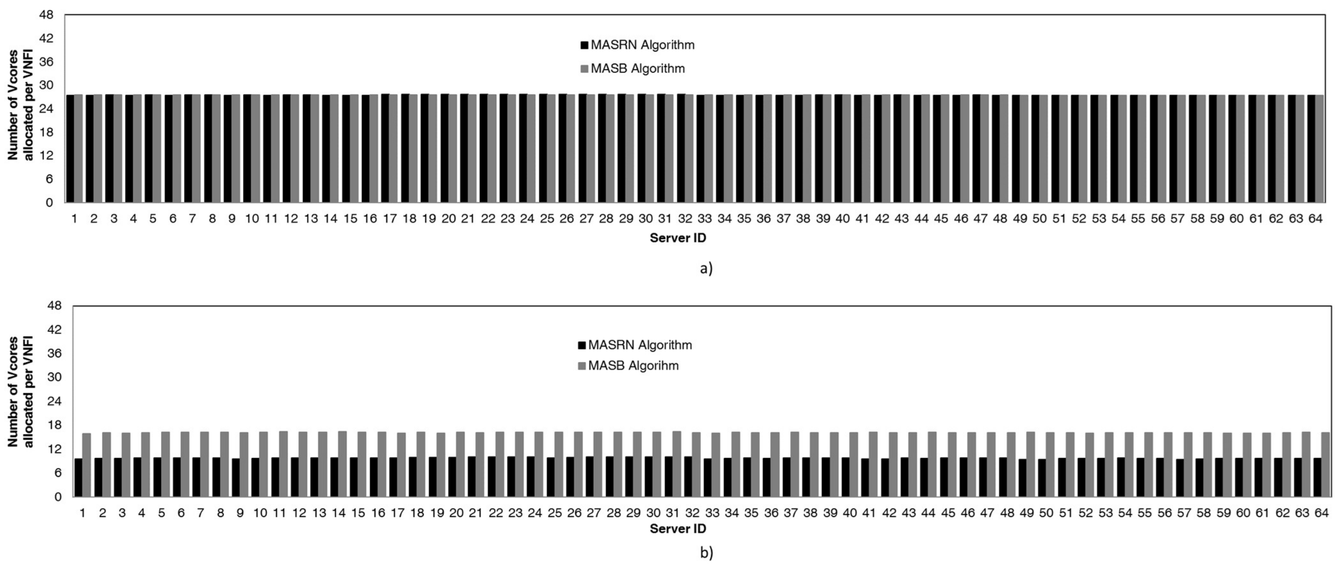


Fig. 9. Number of allocated cores in each server for the MASRN and MASB algorithms when $T = 450$. The cases of basic scenario ($\gamma = 1$) (a) and that in which the link bandwidth is reduced to 10% of the basic scenario ($\gamma = 0.1$) (b) are reported.

values of $3.28\cdot10^{-3}$, $3.23\cdot10^{-3}$ and $1.23\cdot10^{-1}$ for the rejected bandwidth fraction of the MASRN, MASB and Bari's algorithms respectively. The better performance is due to two main reasons: i) the lower fragmentation of processing resources due to the application of a vertical scaling technique in MASRN and MASB algorithms that handle traffic increase by allocating more cores to the VNFI instead of increasing the number of VNFI instantiated; ii) the choice of the Bari's algorithm of selecting the same server for the VNFs of an SFC, even when the server is highly utilized that leads to unavailability of cores of new VNFI for more than one VNF in the phase of resource availability check; iii) the uniform utilization of the processing and bandwidth resources of MASRN and MASB that allows for higher probability in accommodating the SFCs while the re-use of VNFI instantiated in Bari's algorithm leads to choosing the most loaded servers with a consequent higher blocking probability. The uniform occupancy of resources in MASRN and MASB algorithms is shown in Fig. 9.a where we report in the case of $T = 450$, the number of cores occupied in the servers that are represented in the x-axis with the identification (ID) of Fig. 7.

We also notice from Fig. 8 how MASB outperforms MASRN in the scenario in which the link bandwidth is scaled ($\gamma = 0.1$). In the case of 200 SFCs offered, the possibility in MASB of embedding more than one VNF of an SFC in the same server allows for an improvement of two orders of magnitude with respect to MASRN. The rejected bandwidth fraction is equal to $4.19 \cdot 10^{-3}$ and $3.29 \cdot 10^{-1}$ for MASB and MASRN algorithms respectively. The improvement is confirmed in Fig. 9.b where we observe how MASB better employs the processing resources increasing the number of cores allocated. Finally we can notice from Fig. 8 how in the case γ=0.1 the Bari's algorithm allows for performance remarkably better than MASRN and MASN and it is able to overcome its performance obtained in the basic scenario in which the link capacities are higher. The better performance and the strange behavior of the Bari's algorithm are due to the following reasons: i) its bandwidth saving thanks to its capacity of embedding entire SFC in a same server; ii) its characteristic of saturation of the link bandwidth resources before the server processing ones that leads to overcome the problem of choosing a same server for more than one VNF when processing capacity is not available for all.

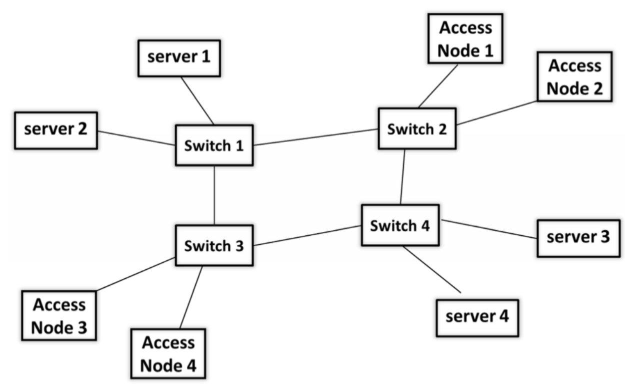


Fig. 10. Network composed by four servers, four switches and four access nodes.

TABLE I
EXECUTION TIME OF THE ILP AND RLACM/VMMPC ALGORITHMS

		ILP	RLACM/VMMPC Heuristic
N	2	22 sec.	1 sec.
	4	162 sec.	3 sec.
	6	2048 sec.	4 sec.
	8	5457 sec.	5 sec.

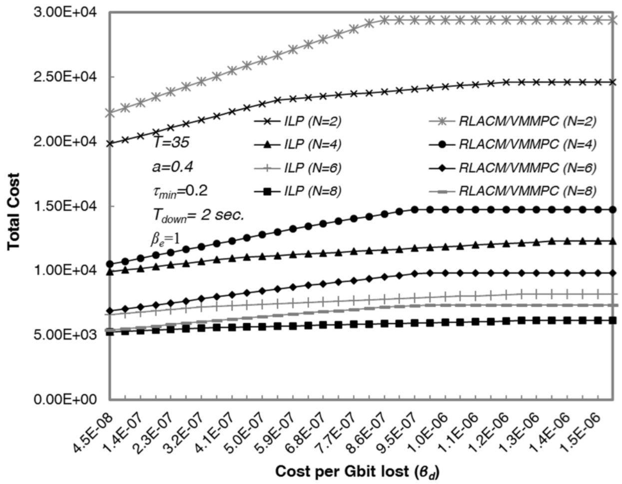


Fig. 11. Comparison between ILP and RLACM/VMMPC algorithms for the network of Fig. 10. The total cost is reported as a function of the cost β_d per Gbit lost for $T = 35$ SFCs and N equal to 2,4,6,8.

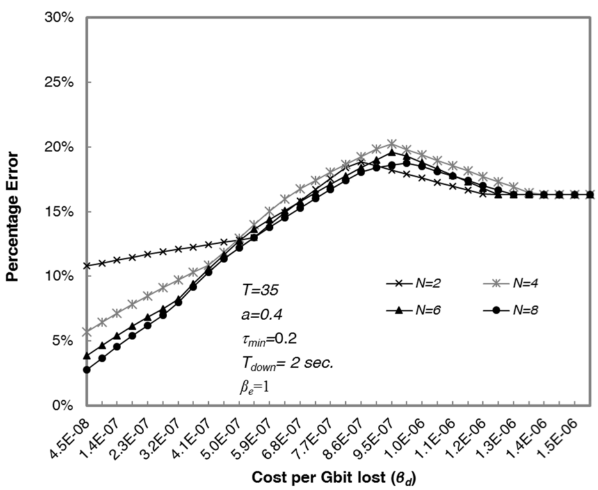


Fig. 12. Percentage error between ILP and RLACM/VMMPC algorithms as a function of the cost β_d per Gbit lost for $T = 35$ SFCs and N equal to $2, 4, 6, 8$.

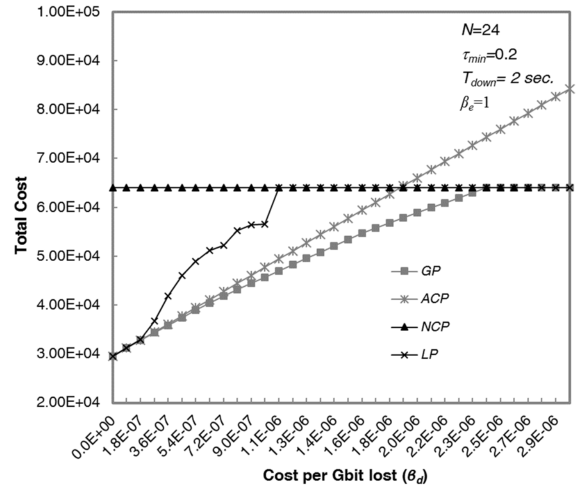


Fig. 13. The total cost of the policies $\mathcal{D}^{ac}$, $\mathcal{D}^{nc}$ $\mathcal{D}^{\mathcal{L}}$ and $\mathcal{D}^{\mathcal{G}}$ as a function of the cost β_d per Gbit lost when $T = 500$, $N = 24$, $\tau_{min} = 0.2$, $a = 1$, $T_{down} = 2\ sec$ and $\beta_e = 1$.

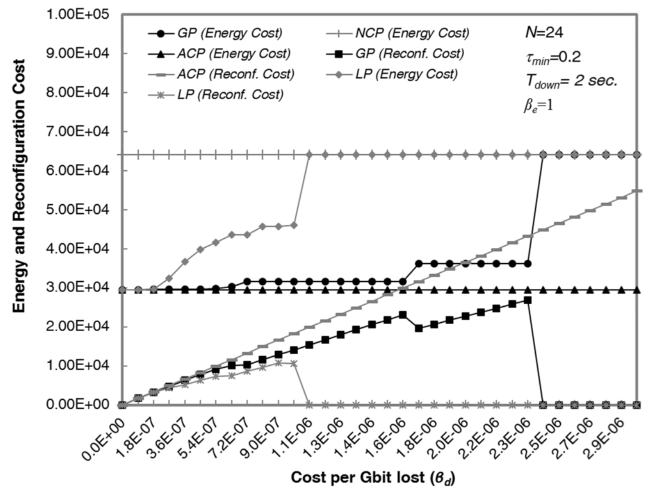


Fig. 14. The energy and reconfiguration costs of the policies $\mathcal{D}^{ac}$, $\mathcal{D}^{nc}$ $\mathcal{D}^{\mathcal{L}}$ and $\mathcal{D}^{\mathcal{G}}$ as a function of the cost β_d per Gbit lost when $T = 500$, $N = 24$, $\tau_{min} = 0.2$, $a = 1$, $T_{down} = 2\ sec$ and $\beta_e = 1$.

B. Cost Evaluation of the VNFI Migration Policies

We evaluate the effectiveness of the introduced VNF migration policies when T SFCs are generated where the bandwidth offered by each of them in the PHI assumes values in the set [100Mbps, 150Mbps, 200Mbps, 250Mbps, 300Mbps] according to a Zipf distribution. The SFCs are routed in the PHI according to the heuristic proposed in Section V. We assume a cycle-stationary traffic scenario with N intervals and where the bandwidth of the SFC is modulated by the scale factor τ_h in the $h-th$ interval $(h = 0, \cdots, N-1)$ chosen according to the classical sinusoidal trend and given by the following expression:

$$\tau_h = \begin{cases} 1 & if\ h = 0 \\ 1 - 2\dfrac{h}{N}(1-\tau_{min}) & h = 1, \cdots, \dfrac{N}{2} \\ 1 - 2\dfrac{N-h}{N}(1-\tau_{min}) & h = \dfrac{N}{2}+1, \cdots, N-1 \end{cases} \tag{19}$$

where $\tau_0 = 1$ and $\tau_{\frac{N}{2}} = \tau_{min}$ denote the scale factors in the peak and least traffic conditions respectively.

We assume that the servers are characterized by a maximum power P_{max} equal to 1000 W and the idle power P_{idle} equal to a times the maximum power P_{max} where the parameter a characterizes how much the server power is dependent on the handled traffic; its value can vary from 0 to 1 where $a = 1$ corresponds to the case of server with no rate adaptive power consumption while $a = 0$ corresponds to that of servers in which the idle power consumption is zero and consequently dependent on the handled traffic only.

In this section we first compare the results of the RLACM/VMMPC algorithms to the ones achieved by solving the Integer Linear Programming (ILP) formulation of the optimal Revenue Loss Aware Resource Consolidation/De-consolidation Problem (RLARCDP) mentioned in Subsection VI-A. The results of the ILP formulation are achieved by using the CPLEX solver. The comparison

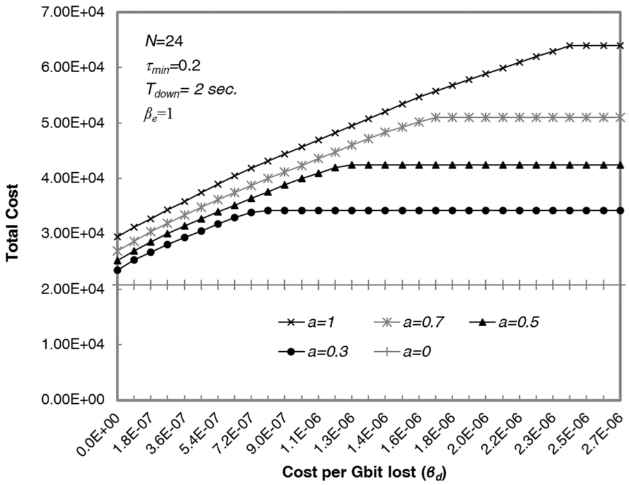


Fig. 15. The total cost of the policy $\mathcal{D}^{\mathcal{G}}$ as a function of the cost β_d per Gbit lost when $T = 500$, $N = 24$, $\tau_{min} = 0.2$, $T_{down} = 2\ sec$, $\beta_e = 1$ and a equal to 0, 0.3, 0.5, 0.7 and 1.

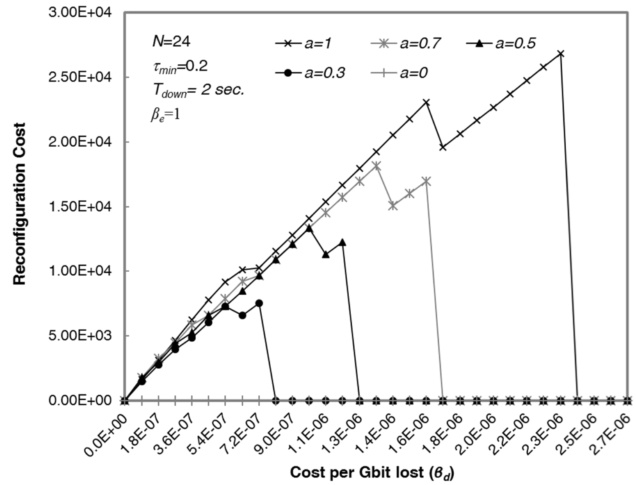


Fig. 17. The reconfiguration cost of the policy $\mathcal{D}^{\mathcal{G}}$ as a function of the cost β_d per Gbit lost when $T = 500$, $N = 24$, $\tau_{min} = 0.2$, $T_{down} = 2\ sec$, $\beta_e = 1$ and a equal to 0, 0.3, 0.5, 0.7 and 1.

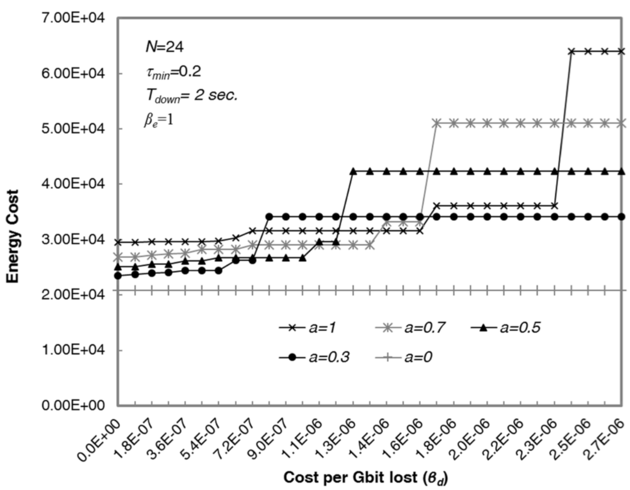


Fig. 16. The energy cost of the policy $\mathcal{D}^{\mathcal{G}}$ as a function of the cost β_d per Gbit lost when $T = 500$, $N = 24$, $\tau_{min} = 0.2$, $T_{down} = 2\ sec$, $\beta_e = 1$ and a equal to 0, 0.3, 0.5, 0.7 and 1.

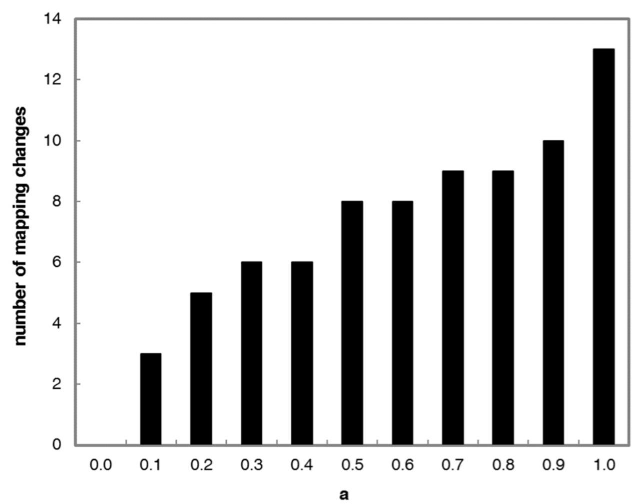


Fig. 18. The number of mapping changes involved by the policy $\mathcal{D}^{\mathcal{G}}$ in the case of $\beta_d = 4.5 \cdot 10^{-7}$ and as a function of the parameter a.

performance has been carried out on a machine characterized by 3.40 GHz Intel i7-3770 processor and by an 8 GB memory. After the better performance of RLACM/VMMPC algorithms with respect to traditional algorithms will be discussed.

Due to the high computation times of the ILP solution, we carry out the comparison between ILP and RLACM/VMMPC algorithms in the case of the small network of Fig. 10 composed by four servers, four switches and four access nodes from which the SFCs are originated and terminated. 40 Gbps links are considered except the links connecting the server to the switches whose the bit rate is equal to 10 Gbps. The servers are equipped with 48 cores each one. They are characterized by a value equal to 0.4 of the parameter a. The downtime T_{down} is set equal to 2 sec. We consider $T = 35$ SFCs each one characterized by the graph reported in Fig. 6.b and composed by two access node, one FW and one IDS. We assume cycle-stationary traffic of parameters $\tau_{min} = 0.2$ and N equal to $2, 4, 6, 8$. We report in Fig. 11 the total cost as a function of the cost β_d per Gbit of lost traffic. We assume the cost β_e per consumed power watt equal to 1. The percentage error of the results provided by the RLACM/VMMPC algorithms with respect to the ILP solution is given in Fig. 12. We can see that the RLACM/VMMPC algorithms finds solutions with a percentage error in order of 20% in the worst case and as small as 7% in the case of low reconfiguration costs ($\beta_d \leq 2.37 \cdot 10^{-7}$) when many migrations are involved. An in-depth investigation allows us to affirm that the critical issue of the proposed solution is the VMMPC algorithm with its constraint of mapping all of the VNFIs of a same server on any other one and without the possibility to distribute these VNFIs in more than one server. Finally we show in Tab. I the execution time of the ILP and RLACM/VMMPC algorithms for N equal to 2, 4, 6, 8. We can notice how the RLACM/VMMPC heuristic, due to its polynomial complexity, allows for computation times remarkably lower than the ILP solution.

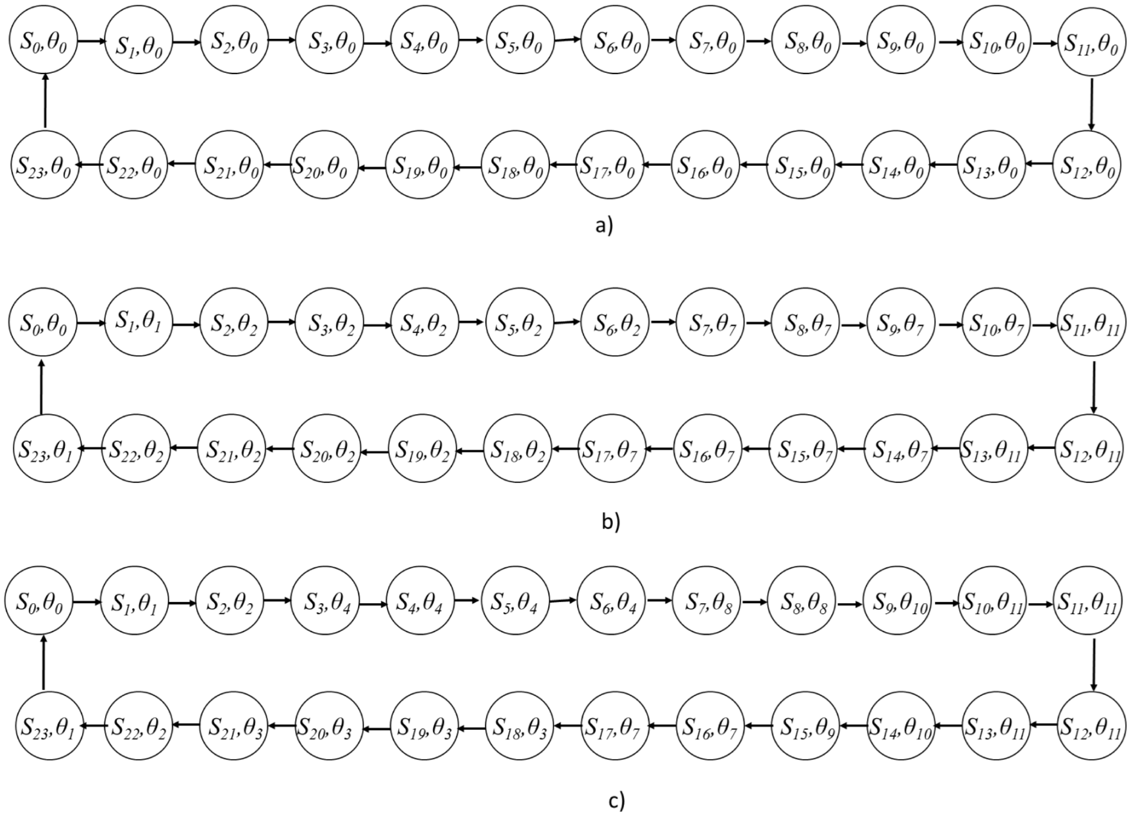


Fig. 19. The mapping changes involved by the policy $\mathcal{D}^{\mathcal{G}}$ in the case of $\beta_d = 4.5 \cdot 10^{-7}$. The cases $a = 0$ (a), $a = 0.5$ (b) and $a = 1$ (c) are represented.

Next we evaluate the effectiveness of the policy $\mathcal{D}^{\mathcal{G}}$ introduced in Subsection VI-B for the bigger network case of Fig. 7 and minimizing the total cost in a cycle-stationarity period. The policy $\mathcal{D}^{\mathcal{G}}$ aims at minimizing the sum of the energy and reconfiguration costs during all of the cycle-stationary period. We compare the policy $\mathcal{D}^{\mathcal{G}}$ to the three classical Never Change, Always Change and Local policies $\mathcal{D}^{nc}$, $\mathcal{D}^{ac}$ and $\mathcal{D}^{\mathcal{L}}$ respectively. When the $\mathcal{D}^{nc}$ policy is selected, the only mapping θ_0 is applied and the reconfiguration cost is absent. The mapping θ_0 is the one obtained by the application of the VMMPC algorithm during the PHI. It is admissible for all of the traffic conditions and for this reason it is chosen as mapping of the policy $\mathcal{D}^{nc}$. Conversely when the policy $\mathcal{D}^{ac}$ is selected the least power consumption admissible mapping is applied during the $i - th$ interval of the cycle-stationarity period. Notice how the policy $\mathcal{D}^{ac}$ allows for the minimization of the energy consumption at the expense of high reconfiguration costs. Finally $\mathcal{D}^{\mathcal{L}}$ is a simple policy based on a cost minimization in each stationary interval and decides the mapping to be applied without taking into account future reconfiguration costs; its operation mode is similar to the one proposed in [12]. When traffic state changes from S_i to S_{i+1} the algorithm chooses as mapping to be applied in the state S_{i+1} the admissible one minimizing the difference between the reconfiguration cost and the energy saving involved in a transition from the mapping applied actually in the state S_i.

We report in Fig. 13 the total cost and in Fig. 14 the energy and reconfiguration costs of the policies $\mathcal{D}^{\mathcal{G}}$, $\mathcal{D}^{\mathcal{L}}$, $\mathcal{D}^{nc}$ and $\mathcal{D}^{ac}$ as a function of the cost β_d per Gbit of lost traffic, when $T = 500$ SFCs are considered. We assume the cost β_e per consumed power watt equal to 1 while servers with power consumption independent of the handled traffic ($a = 1$) are taken into account. The traffic profile is characterized by the scale factors expressed by Eq. (19) in which N and τ_{min} are chosen equal to 24 and 0.2 respectively. Finally the downtime T_{down} is set equal to 2 sec. From Fig. 13 we can observe how the proposed policy $\mathcal{D}^{\mathcal{G}}$ performs better than the policies $\mathcal{D}^{\mathcal{L}}$, $\mathcal{D}^{nc}$ and $\mathcal{D}^{ac}$ for all of the values of β_d and allows for a minimization of the total cost. For instance for $\beta_d = 9.90 \cdot 10^{-7}$ the cost values 45691, 47763, 64000 and 47763 are obtained for the policies $\mathcal{D}^{\mathcal{G}}$, $\mathcal{D}^{\mathcal{L}}$, $\mathcal{D}^{nc}$ and $\mathcal{D}^{ac}$ respectively. As expected the policies $\mathcal{D}^{ac}$ and $\mathcal{D}^{nc}$ reach the cost value of $\mathcal{D}^{\mathcal{G}}$ only for low and high values of β_d respectively. In fact in the former case reconfiguration costs are negligible and the optimal policy is the one minimizing the power consumption that is $\mathcal{D}^{ac}$. Conversely when the reconfiguration cost are high, the optimal policy is the one in which VNFI migrations are not performed at all and it is convenient to apply the same mapping solution in all of the traffic intervals as the policy $\mathcal{D}^{nc}$ makes. From Fig. 13 we can notice how, due to the high reconfiguration costs, the policy $\mathcal{D}^{ac}$ has cost values increasing as β_d increases and for β_d higher than or equal to $1.9 \cdot 10^{-6}$ the policy $\mathcal{D}^{ac}$ has performance worse than the policies $\mathcal{D}^{nc}$, $\mathcal{D}^{\mathcal{L}}$ and $\mathcal{D}^{\mathcal{G}}$. Finally we can notice how the policy $\mathcal{D}^{\mathcal{L}}$, based on a simple benefit/cost comparison in a stationary interval performs worse than the policy $\mathcal{D}^{\mathcal{G}}$ based on the application of Markov Decision Process theory. The performance improvement of $\mathcal{D}^{\mathcal{G}}$ can reach in some cases values of 27%.

Next we evaluate the behavior of the policy $\mathcal{D}^{\mathcal{G}}$ when we vary the parameter a characterizing how much the server power consumption is dependent on the handled traffic. In Fig. 15-17 we report the total, energy and reconfiguration costs of the policy $\mathcal{D}^{\mathcal{G}}$ respectively as a function of β_d. The values a equal to 0, 0.3, 0.5, 0.7 and 1 are considered. First of all, from Fig. 15 we can notice how the total cost is as much lower as a is lower. When the reconfiguration costs are negligible ($\beta_d \leq 7.2 \cdot 10^{-7}$) this is due to the use of more power effective servers the lead to a decrease in energy cost when the parameter a decrease as shown in Fig. 16. Conversely when β_d increases ($\beta_d > 7.2 \cdot 10^{-7}$) the policy $\mathcal{D}^{\mathcal{G}}$ allows for a reduction of the reconfiguration costs when a decreases as shown in Fig. 17; the policy has a right behavior by reducing the number of migrations and consequently the reconfigurations costs when a decreases and the switching off of a server is less advantageous due to the lower fix power consumption cost. That is confirmed in Fig. 18 where we show a bar graph reporting the number of mapping changes as a function of the parameter a. The mappings applied are reported in Figs 19.a, Fig. 19.b and Fig. 19.c for the cases $a = 0$, $a = 0.5$ and $a = 1$ respectively. Here we show in the case of $\beta_d = 4.5 \cdot 10^{-7}$, the mappings $\theta \in \Theta$ applied in each traffic state S_i ($i = 0, \cdots, N-1$). The traffic state S_i represents the traffic condition characterized by the scale factor τ_i. The symmetry of the scale factor τ_i ($\tau_i = \tau_{N-i}$, $i = 1, \cdots, 11$) leads to the coincidence of some mappings of the set Θ and in particular we have $\theta_i = \theta_{N-i}$, ($i = 1, \cdots, 11$). We can notice from Fig. 19 that the number of mapping changes decreases when the parameter a decreases. As represented in Fig. 19.a, when $a = 0$ only the mapping θ_0 is applied in all of the traffic states and no mapping changes occur. In the case $a = 0.5$ of Fig. 19.b eight mapping changes occur that is from (S_0, θ_0) to (S_1, θ_1), from (S_1, θ_1) to (S_2, θ_2), from (S_6, θ_2) to (S_7, θ_7), from (S_{10}, θ_7) to (S_{11}, θ_{11}), from (S_{13}, θ_{11}) to (S_{14}, θ_7), from (S_{17}, θ_7) to (S_{19}, θ_2), from (S_{22}, θ_2) to (S_{23}, θ_1), from (S_{23}, θ_1) to (S_0, θ_0). Conversely in the case $a = 1$, 13 mappings changes occur as represented in Fig. 19.c.

VIII. CONCLUSIONS

The aim of this paper is to propose a migration policy that establishes when and where migrations of VNFI have to be accomplished so as to minimize a total cost characterized by the sum of the energy cost and the reconfiguration cost occurring when the VNFIs are moved from the initial location. We have formulated the optimization problem and because of its complexity we have proposed a heuristic based on the Markov Decision Process theory and applicable in cycle-stationary traffic scenarios and when the possible mappings, characterizing where the VNFIs are instantiated, are a-priori determined. We have compared the results of the heuristic to the ones of the ILP formulation of the optimal problem in the case of a simple network with four servers. The percentage error achieved was approximately 20% in the worst case. Finally the heuristic has been compared to a simple policy in which the migrations are chosen according to a benefit/cost evaluation in a stationary interval without taking into account future reconfiguration costs. In this case we have shown how the proposed heuristic performs better and allows for an improvement on the order of 27% with respect to the simple policy.

REFERENCES

[1] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[2] P. Veitch, M. J. McGrath, and V. Bayon, "An instrumentation and analytics framework for optimal and robust NFV deployment," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 126–133, Feb. 2015.

[3] R. Yu, G. Xue, V. T. Kilari, and X. Zhang, "Network function virtualization in the multi-tenant cloud," *IEEE Netw.*, vol. 29, no. 3, pp. 42–47, May/Jun. 2015.

[4] Z. Bronstein, E. Roch, J. Xia, and A. Molkho, "Uniform handling and abstraction of NFV hardware accelerators," *IEEE Netw.*, vol. 29, no. 3, pp. 22–29, May/Jun. 2015.

[5] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[6] ETSI Industry Specification Group (ISG) NFV. (Jan. 2015). *ETSI Group Specifications on Network Function Virtualization*. [Online]. Available: http://docbox.etsi.org/ISG/NFV/Open/Published/

[7] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov./Dec. 2014.

[8] J. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, document RFC 7665 of The Internet Engineering Task Force (IETF) Service Function Chaining (SFC) Working Group (WG), 2015. [Online]. Available: https://datatracker.ietf.org/doc/rfc7665/

[9] M. Yoshida, W. Shen, T. Kawabata, K. Minato, and W. Imajuku, "MORSA: A multi-objective resource scheduling algorithm for NFV infrastructure," in *Proc. 16th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2014, pp. 1–6.

[10] Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," Cornell Univ. Library, Ithaca, NY, USA, Tech. Rep. arXiv: 1608.00095. [Online]. Available: https://arxiv.org/abs/1608.00095

[11] T. Wen, H. Yu, G. Sun, and L. Liu, "Network function consolidation in service function chaining orchestration," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jul. 2016, pp. 1–6.

[12] M. Ghaznavi *et al.*, "Elastic virtual network function placement," in *Proc. 4th IEEE Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 1–6.

[13] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *Proc. 10th Int. Conf. Netw. Service Manage. (CNSM)*, Rio de Janeiro, Brazil, Nov. 2014, pp. 418–423.

[14] R. Mijumbi *et al.*, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. IEEE Conf. Netw. Softwarization*, Apr. 2015, pp. 1–9.

[15] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Istanbul, Turkey, Apr. 2014, pp. 2402–2407.

[16] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[17] X. Li and C. Qian, "A survey of network function placement," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2016, pp. 1–6.

[18] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[19] M. Ghaznavi, N. Shahriar, R. Ahmed, and R. Boutaba, "Service function chaining simplified," Cornell Univ. Library, Ithaca, NY, USA, Tech. Rep. arXiv: 1601.00751. [Online]. Available: https://arxiv.org/abs/1601.00751

[20] S. Rajagopalan, D. Williams, H. Jamjoom, and A. Warfield, "Split/merge: System support for elastic execution in virtual middleboxes," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement.*, Lombard, IL, USA, Apr. 2013, pp. 227–240.

[21] A. Gember-Jacobson *et al.*, "OpenNF: Enabling innovation in network function control," in *Proc. ACM SIGCOMM*, Chicago, IL, USA, Aug. 2014, pp. 163–174.

[22] J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, "BlindBox: Deep packet inspection over encrypted traffic," in *Proc. ACM SIGCOMM*, London, U.K., 2015, pp. 213–226.

[23] V. Eramo, E. Miucci, and M. Ammar, "Study of reconfiguration cost and energy aware VNE policies in cycle-stationary traffic scenarios," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1281–1297, Apr. 2016.

[24] J. Fan and M. H. Ammar, "Dynamic topology configuration in service overlay networks: A study of reconfiguration policies," in *Proc. 25th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[25] V. Eramo, E. Miucci, and M. Ammar, "Study of migration policies in energy-aware virtual router networks," *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1919–1922, Nov. 2014.

[26] V. Eramo, E. Miucci, L. Compagnone, and M. Ammar, "Evaluation of reconfiguration cost and energy aware virtual network embedding policies," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1–5.

[27] W. Cerroni and F. Callegati, "Live migration of virtual network functions in cloud-based edge networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2963–2968.

[28] V. Eramo, A. Tosti, and E. Miucci, "Server resource dimensioning and routing of service function chain in NFV network architectures," *J. Elect. Comput. Eng.*, vol. 2016, Feb. 2016, Art. no. 7139852.

[29] M. F. Zhani, Q. Zhang, G. Simona, and S. Boutaba, "VDC planner: Dynamic migration-aware virtual data center embedding for clouds," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ghent, Belgium, May 2013, pp. 18–25.

[30] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. 25th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[31] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Comput.*, vol. 16, no. 2, pp. 249–264, Jun. 2013.

[32] J. A. Aroca, A. Chatzipapas, A. F. Anta, and V. Mancuso, "A measurement-based characterization of the energy consumption in data center servers," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2863–2877, Dec. 2015.

[33] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Trans. Services Comput.*, vol. 3, no. 4, pp. 266–278, Oct./Nov. 2016.

[34] M. Garey, R. Graham, and D S Johnson, "Resource constrained scheduling as generalized bin packing," *J. Combinat. Theory Ser. A*, vol. 21, no. 3, pp. 257–298, Nov. 1976.

[35] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA, USA: MIT Press, 1960.

[36] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving Markov decision problems," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1995, pp. 394–402.

[37] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, Aug. 1987.

[38] V. Eramo, S. Testa, and E. Miucci, "Evaluation of power saving and feasibility study of migrations solutions in a virtual router network," *J. Elect. Comput. Eng.*, vol. 2014, Mar. 2014, Art. no. 910658.

[39] H. Kikuchi and K. Takahashi, "Zipf distribution model for quantifying risk of re-identification from trajectory data," in *Proc. 13th Annu. Conf. Privacy, Secur. Trust (PST)*, Jul. 2015, pp. 14–21.

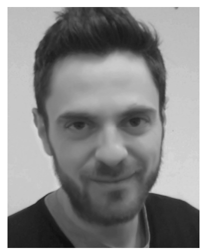
[40] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet access traffic measurement and analysis," in *Proc. 4th Int. Conf. Traffic Monitor. Anal.*, Aug. 2012, pp. 29–42.

[41] P. Quinn and J. Guichard, "Service function chaining: Creating a service plane via network service headers," *Computer*, vol. 47, no. 11, pp. 38–44, Nov. 2014.

[42] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband Internet traffic," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf.*, Chicago, IL, USA, Oct. 2009, pp. 90–102.

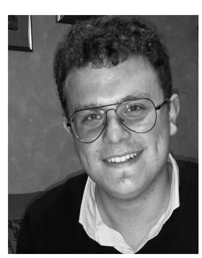
[43] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling Internet traffic dynamics of cellular devices," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Modeling Comput. Syst.*, San Jose, CA, USA, Jun. 2011, pp. 305–316.

Vincenzo Eramo received the Laurea degree in electronics engineering and the Dottorato di Ricerca (Ph.D.) degree in information and communications engineering from the University of Roma La Sapienza in 1995 and 2001, respectively. In 1996, he was a Researcher with the Scuola Superiore Reiss Romoli. In 1997, he joined the Fondazione Ugo Bordoni, as a Researcher with the Telecommunication Network Planning Group. From 2002 to 2005, he was an Assistant Professor, and from 2006 to 2010, an Aggregate Professor with the INFOCOM Department, University of Rome La Sapienza. He is currently an Associate Professor with the Department of Engineering of Information, Electronics and Telecommunications. His research activities have been carried out in the framework of national and international projects. In particular, he was a Scientific Coordinator of the University of Roma La Sapienza of E-PhotoONe+ and BONE, two networks of excellence focusing on the study of optical networks and financed by European Commissions (FP6 and FP7) from 2006 to 2007 and 2008 to 2011, respectively. He received the Exemplary Editor Award 2016 of the IEEE COMMUNICATIONS LETTERS. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS from 2011 to 2015. He has been an Associate Editor of the IEEE COMMUNICATION LETTERS since 2014.



Emanuele Miucci received the Laurea (M.Sc.) (*cum laude*) degree in telecommunications engineering and the Ph.D. degree from the University of Rome La Sapienza, Italy, in 2010 and 2014, respectively. He has been a Post-Doctoral Student with the Department of Engineering of Information, Electronics and Telecommunications, University of Rome La Sapienza, since 2014. His current research interests are in the fields of virtualization, power saving techniques, all-optical networks, and switching architectures.

Mostafa Ammar (F'02) received the S.B. and S.M. degrees from the Massachusetts Institute of Technology in 1978 and 1980, respectively, and the Ph.D. degree from the University of Waterloo, ON, Canada, in 1985. He has been with Georgia Tech since 1985. He is currently a Regents' Professor with the School of Computer Science, Georgia Institute of Technology. His research interests are in network architectures, and protocols and services. He has contributions in the areas of multicast communication and services, multimedia streaming, content distribution networks, network simulation, disruption-tolerant networks, virtual network design, and most recently in mobile cloud computing. He has authored extensively in these areas. To date, 33 Ph.D. students have completed their degrees under his supervision, and many have gone on to distinguished careers in academia and industry. He was an elected Fellow of the IEEE in 2002 and a Fellow of the ACM in 2003. His awards include the IBM Faculty Partnership Award in 1996, the Best Paper Award at the 7th WWW conference in 1998, the Georgia Tech Outstanding Doctoral Thesis Advisor Award in 2006, the Outstanding Service Award from the IEEE Technical Committee on Computer Communications in 2010, and the ACM Mobihoc Best Paper Award in 2012. He has served the research community in multiple roles. Most notably, he served as the Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING from 1999 to 2003, and he was the Co-TPC Chair of the IEEE ICNP 1997, the ACM CoNEXT 2006, and the ACM SIGMETRICS 2007 conferences.



Francesco Giacinto Lavacca received the Laurea (M.Sc.) (*cum laude*) degree in electronic engineering from the University of Rome La Sapienza, Italy, in 2013, where he is currently pursuing the Ph.D. degree in information technology engineering with DIET. He has been a Visiting Researcher with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. His current research interests are in the fields of all-optical networks and switching architectures, 5G networks, and network function virtualization.