Reinforcement Learning-enabled Auctions for Self-Healing in Service Function Chaining

Marios Avgeris
Dept. of Systems and Computer Eng.
Carleton University
Ottawa, Canada
MariosAvgeris@cunet.carleton.ca

Aris Leivadeas
Dept. of Software Engineering and IT
École de Technologie Supérieure (ÉTS)
Montréal, Canada
aris.leivadeas@etsmtl.ca

Ioannis Lambadaris
Dept. of Systems and Computer Eng.
Carleton University
Ottawa, Canada
ioannis@sce.carleton.ca

2023 IEEE International Conference on Communications Workshops (ICC Workshops) | 979-8-3503-3307-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCWORKSHOPS57953.2023.10283498

***Abstract*—Service Function Chaining (SFC), defines the capability of interconnecting a number of *ordered* Service Functions (SFs) to create composite network services. A critical issue in SFC is the autonomic fault recovery, i.e., bringing the system back to its normal operation after a hardware or software failure. To address this challenge, in this paper, we propose a novel distributed methodology that treats the SFC Self-Healing problem in an Edge-Cloud infrastructure, while accounting for the various stakeholders. In particular, the individual SFC healing decisions are iteratively optimized and determined, while a Reinforcement Learning (RL)-based SFC-to-datacenter association procedure is realized. This process is complemented by a combinatorial auction-based resource allocation mechanism that resolves the potential SFC collocations at the end of each iteration. The proper operation, effectiveness and efficiency of our proposed healing mechanism is assessed under various evaluation scenarios.**

***Index Terms*—Service Functions, Self-Healing, Edge/Cloud Computing, Reinforcement Learning, Combinatorial Auction.**

I. Introduction

With the emergence of 5G networks, the requirements for high Quality of Service (QoS) and ubiquitous service availability have become stricter. Together with the everlasting growth in the number of interconnected applications and devices, this has caused a paradigm shift in the network service delivery to the end users; from network topologies being statically deployed and tailored to specific services, to dynamically composing network functions in network services. This flexible and dynamic deployment model called Service Function Chaining (SFC), essentially hands over the traffic of a service to a predefined ordered list of chained Service Functions (SFs) or Virtual Network Functions (VNFs) (e.g., load balancers and firewalls) [1]. In this context, Software Defined Networking (SDN) and Network Function Virtualization (NFV) are the main technologies enabling the SFC deployment, configuration and lifecycle management in a timely and cost efficient way.

However, network outages can still disrupt user experience. According to [2], node failure probability could range between 60-99.8% with the increase in the network density and complexity. That is why the design and realization of highly robust and resilient systems in the 5G era calls for enhanced physical and virtual infrastructure management. Specifically, in the context of critical applications and everyday operations that 5G promises to support, rapid network failure detection and mitigation are of paramount importance, as most of the potential failures cannot be predicted or identified in advance to allow proactive corrections. Therefore, Self-Organizing and Self-Healing mechanisms are developed to meet such requirements, including intelligent monitoring, detection, diagnosis and failure compensation capabilities [3].

A. Related Work & Motivation

Recently, several researchers have studied the problem of SFC/NFV Self-Healing. In [3], the authors introduce SELFNET, a framework that enables autonomic network management functionalities and they focus on the Self-Healing case of reactively or preventively dealing with the detected or predicted network failures. A Self-Healing framework for SFCs in 5G networks is discussed in [4] that ensures service availability, by dynamically redeploying failed VNF instances.

The aforementioned works either provide architecture descriptions or present practical solutions to the specific problem of SFC/NFV Self-Healing. However, when it comes to mathematically formulating the problems of VNF deployment and network Self-Healing individually, significant attention has been drawn to utilizing Reinforcement Learning (RL)-enabled techniques. Regarding the former problem, the authors in [5] propose a Deep Reinforcement Learning (DRL)-based solution to tackle the VNF placement problem, considering the dynamic changes in the network load of a SDN/NFV-enabled infrastructure. Specifically, they formulate the VNF placement problem as a Binary Integer Programming model, aiming to minimize a weighted cost consisting of the placement, rejection and running sub-costs. On the other hand, in [6], end-to-end service-level performance predictions are taken into account to perform autonomous VNF placement. This makes the proposed RL-based framework, more resilient to dynamic network conditions and hardware heterogeneity.

Regarding network Self-Healing, the work in [7] proposes a mechanism to deal with cell outages in 5G ultra dense networks, while maximizing throughput and guaranteeing the QoS demands for each user. The authors examine a DRL solution to this NP-hard problem that utilizes K-means clustering and deep neural networks. Almost the exact same DRL concept is also applied in [8], though this time the focus is put on cell outage compensation in massive IoT environments.

B. Contribution & Paper Organization

Despite the efforts made in the previous works, the issues of accounting for various stakeholders in an Edge-Cloud

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.

infrastructure (i.e., infrastructure providers and SFC vendors), examining the dynamics among them and treating the healing process in a decentralised way, still remain notably unexplored. Especially in the context of 5G where services are expected to respond in milliseconds, the rapid restoration of a system to normal operation becomes a challenge. In this work, our goal is to exactly deal with these issues, by proposing a distributed Self-Healing mechanism where the various SFC vendors act as independent agents formulating their healing strategy, while the infrastructure provider aims at maximizing its monetary gains. The key contribution of this paper is threefold:

- An Edge-Cloud infrastructure is considered, consisting of interconnected datacenters and hosting multiple SFCs. To enable the distributed and autonomous SFC Self-Healing as a reactive response to a node outage, an RL-based mechanism is introduced. During its online phase, each SFC independently selects a node to heal to, aiming at optimizing its benefit, while accounting for the service's computational, networking and QoS requirements.
- Datacenter resources are naturally limited. Thus, the infrastructure provider opts for maximizing its revenue by optimizing the allocated resources to the competing SFCs in each node. For this, a monotone and truthful combinatorial auction-based mechanism is integrated, where the price paid by each SFC is calculated using a generalized Vickrey-Clarke-Groves (VCG) scheme [9], [10].
- We propose a framework that combines the two aforementioned mechanisms, where the SFC-to-node healing associations are iteratively optimized. Detailed numerical results, obtained via simulation, evaluate and demonstrate the effectiveness and efficiency of the proposed work.

The rest of this paper is organized as follows. Section II presents the system model and the concepts of RL in the Self-Healing problem. In Section III the revenue driven resource allocation problem for each node is formulated and solved. Section IV presents the performance evaluation of our proposed framework and Section V concludes the paper.

II. System Model

We consider an Edge-Cloud infrastructure comprising of $|N|$ interconnected nodes, $N = \{1, ..., |N|\}$ being the respective set, that form a graph (Figs. 1, 2). In this infrastructure, SFCs from various vendors consist of interconnected sequences of VNFs. The VNFs are deployed as Virtual Machines (VMs) and the SFCs may span across different nodes. The performance of an SFC depends on the individual performances of the VNF instances in this chain. For example, if a single VNF instance is faulty or overloaded, the whole chain is affected and consequently all traffic is dropped. This work examines the case where a single outage is experienced in one of the infrastructure's nodes and this affects $|S|$ SFCs; this may result to $|S|$ VNFs needing to be relocated to functional nodes, with $S = \{1, ..., |S|\}$ being the respective set.

A. Resource Allocation Auction Model

To consider a healing as successful, the SFC's resource and QoS requirements have to be met. To this end, each node

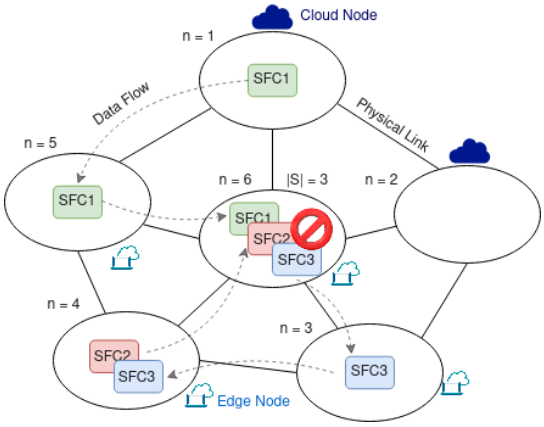


Fig. 1: Initial state: VNFs of SFCs 1, 2 and 3, currently placed in Edge node 6, need healing as the node is out of order.

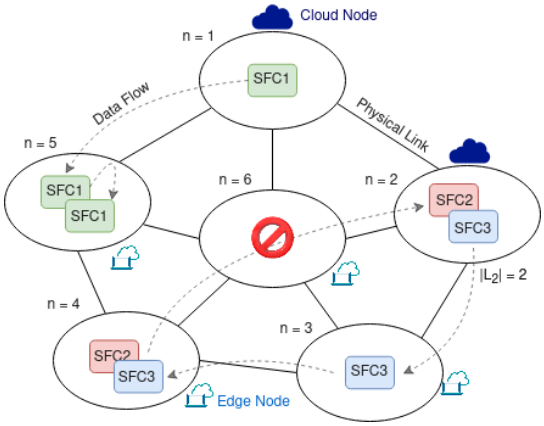


Fig. 2: Final state: VNF of SFC 1 has been redeployed to Edge node 5 and the VNFs of SFC 2 and 3 to Cloud node 2.

$n \in N$ is characterized by a vector $\boldsymbol{c_n} = [c_n^1, ..., c_n^M] \in \mathbb{R}^{1\times M}$, denoting its available resources. The dimensionality M represents the number of different kinds of resources considered for the infrastructure nodes, e.g., if CPU, $memory$ and $bandwidth$ are considered, then $M = 3$ and c_n^1, c_n^2, c_n^3 stand for node's n available CPU, memory and bandwidth resources respectively. Following this notation, each VNF $s \in S$ is characterized by a vector $\boldsymbol{w_s} = [w_s^1, ..., w_s^M] \in \mathbb{R}^{1\times M}$, denoting its required resources. Also, $d_{s,n} \in \mathbb{R}^+$ is the QoS metric representing the additional propagation delay induced by relocating VNF s to node n. This metric is proportional to the additional hops that are added to the SFC's path. It can be calculated by employing Dijkstra's algorithm to find the shortest paths which connect n to the nodes in which the preceding and succeeding VNFs in the SFC sequence are deployed (e.g., $d_{2,2} = 1$ in Fig. 2). The SFC's maximum tolerable additional propagation delay is defined as q_s. Other relocation-related delays (e.g., SFC initial configuration) are considered similar between the nodes so we neglect them from the decision making. A relocation of VNF s to node n is described by $x_{s,n} \in \{0, 1\}$ and is considered successful when the resource requirements of VNF s are fulfilled, i.e., $x_{s,n} == 1$. Multiple SFCs may try to heal to a node, so

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.

we formulate the problem of determining the "winners", i.e., the SFCs that successfully heal there, as a combinatorial auction. This kind of modeling is suitable for an Edge-Cloud environment, as SFCs' demand is a bundle of computing and networking resources and not individual items.

Let $L_n = \{l_1^n, ..., l_{|L_n|}^n\} \subseteq S$ be the discrete subset of SFCs competing for node's n finite resources, $\forall n \in N$. As an example, in Fig. 2, for $n = 2$, we have $L_2 = \{2, 3\}$, as SFCs 2 and 3 heal there. Subsequently, each node gets to decide which of these competing SFCs will be accepted. Let $j \in L_n$ be the index for the locally competing SFCs and r_j be the bid value that SFC j is willing to pay for the requested resources if it is the winner, i.e. its budget. The objectives of our auction-based problem are: (i) determine the set of winning SFCs and (ii) calculate the payment $p_{j,n}$ to be paid by each winning SFC, such that: $\sum_{j \in L_n} w_j^m x_{j,n} \leq c_n^m, \forall m = 1, ..., M$ and $0 \leq p_{j,n} \leq r_j$, when $x_{j,n} == 1$, alternatively, $p_{j,n} = 0$, when $x_{j,n} == 0$. In general, an auction tries to maximize the sum of the bid values, since maximizing the total bid values generates a high revenue P for the infrastructure, given that the payment computation is truthful, i.e., the winners pay at most their bid value and the losers pay nothing. Summarizing, the winners determination problem can be modelled as a Multi-dimensional 0-1 Knapsack problem (MKP) [11]:

$$\max_{x_{j,n}} \quad P = \sum_{j \in L_n} r_j x_{j,n} \tag{1a}$$

$$\text{s.t.} \quad \sum_{j \in L_n} w_j^m x_{j,n} \leq c_n^m, \quad \forall m = 1, ..., M, \tag{1b}$$

$$x_{j,n} \in \{0, 1\}, \quad \forall j \in L_n, \tag{1c}$$

where $x_{j,n}$ denotes whether the SFC j has been healed or not to the examined node n, w_j^m are the SFC's requirements and c_n^m the node's available resources. Note that MKP is a classic combinatorial problem, therefore, the complexity of its brute force solution is $O(|L_n| \cdot 2^{|L_n|})$, $|L_n|$ being the number of SFCs competing for the same resources. Since the complexity of this algorithm grows exponentially, it can only be used for a very small number of competing SFCs. The procedure described in Section III-A is followed to derive a tractable solution which alleviates this restriction.

B. Stochastic Learning Automata (SLA)-based Node Selection

Towards enabling the distributed and autonomous SFC healing, we consider that the SFC vendors act as Stochastic Learning Automata (SLA) [12], by making autonomous decisions regarding the node that they will be healed to, optimizing their individual benefit [13]. Their respective decisions are led by two factors: (i) the resource requirements of their respective VNFs have to be met by the new node and (ii) they try to minimize the additional induced delay. This is an iterative process that takes turns with the auctioning procedure and eventually converges to a favorable healing for each SFC. More details on this follow in Section III. Consequently, each SFC's s benefit/reward from healing their VNF to a node n at iteration k of the SLA healing algorithm is formulated as:

$$\mathscr{R}_{s,n}^{(k)} = A \cdot (1 - \frac{1}{1 + e^{-Z_1(d_{s,n} - q_s)}}) + B \cdot x_{s,n} + C \cdot (1 - \frac{1}{1 + e^{-Z_2(p_{s,n} - r_s)}}), \in [0, 1], \tag{2}$$

where $A, B, C \in [0, 1]$ are empirically selected coefficients that satisfy the condition $0 \leq A + B + C \leq 1$ and assist in fine-tuning and driving the solution towards a desirable state. $Z_1, Z_2 \in \mathbb{R}^+$ are gain coefficients which help bring the respective terms of Eq. (2) close to 1 when $d_{s,n} \leq q_s$ and $p_{s,n} \leq r_s$ and close to 0 otherwise. Based on the SLA theory, the SFC's probabilities of selecting the same or a different node in the next iteration are updated as:

$$\mathscr{P}_{s,n}^{(k+1)} = \mathscr{P}_{s,n}^{(k)} + b \cdot \mathscr{R}_{s,n}^{(k)} \cdot (1 - \mathscr{P}_{s,n}^{(k)}), n^{(k+1)} = n^{(k)}, \tag{3}$$

$$\mathscr{P}_{s,n}^{(k+1)} = \mathscr{P}_{s,n}^{(k)} - b \cdot \mathscr{R}_{s,n}^{(k)} \cdot \mathscr{P}_{s,n}^{(k)}, n^{(k+1)} \neq n^{(k)}. \tag{4}$$

Parameter $0 < b < 1$ is the learning rate of the SLA algorithm that controls the exploration of different healing alternatives; the presence of b prevents inertia phenomena as well, where an SFC would oscillate between decisions. For large values of b, the SLA algorithm converges fast to a stable healing with low accuracy in terms of optimality and vice-versa, accounting for increased solution exploration versus exploitation. The system converges to a stable solution when at least one state probability is close to 1, for each s.

III. Self-Healing: Winners Determination and Payment Computation

A. Reduction of the Winners Determination Problem

A specific case of MKP is the classical Single-dimensional 0-1 Knapsack problem (SKP), with $m = 1$, which is substantially less computationally intensive. Through the use of Dynamic Programming, it can be solved in a pseudo-polynomial time [14]. Therefore, in this section, we introduce a procedure to reduce the dimensions of the MKP to one and thus transform the original problem into an SKP (*Algorithm 1*). The basic idea behind the proposed procedure is to match the requirements of the competing SFCs to predefined VM instance sizes, or "flavors", that are offered in each node. Let each node advertise its F available flavors, $F = \{1, ..., |F|\}$ being the respective set, as vectors $\boldsymbol{u_f} \in \mathbb{R}^{1 \times M}, f \in F$, which denote the flavor's offered resources in each dimension (i.e., CPU, memory and bandwidth). Each of these flavors has a single-dimension weight, a scalar v_f, which is defined by the infrastructure provider. Then, the available capacity of each node n and the reduced, single-dimensioned weights for each SFC j, are also defined as scalars c'_n and w'_j respectively.

Note that two symbols are defined for operations between vectors: *i)* $\boldsymbol{a} \oslash \boldsymbol{b}$ returns the smallest element of the Hadamard (or element-wise) integer division between the two vectors and *ii)* $\boldsymbol{a} \leq \boldsymbol{b}$ is a logical operator which is equal to 1 when all the elements of $\boldsymbol{a}$ are less than or equal to the corresponding elements of $\boldsymbol{b}$. Finally, $\lceil a \rceil$ is the ceiling operator for a scalar. This procedure allows for formulating the SKP reduced winner

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.

determination problem as follows:

$$\max_{x_{j,n}} \quad P = \sum_{j \in L_n} r_j x_{j,n} \tag{5a}$$

$$\text{s.t.} \quad \sum_{j \in L_n} w'_j x_{j,n} \leq c'_n, \tag{5b}$$

$$x_{j,n} \in \{0,1\}, \quad \forall j \in L_n. \tag{5c}$$

This reduced SKP can be solved optimally in pseudo-polynomial time with a computational complexity equal to $O(|L_n| \cdot c'_n)$, by using the Bellman recursion [14]. Additionally, the complexity of the algorithm transforming the MKP to an SKP problem is equal to $O(|L_n| \cdot F)$, thus the overall pseudo-polynomial complexity of the resource allocation problem becomes equal to $O(|L_n| \cdot (F + c'_n))$.

Algorithm 1 Reducing MKP to SKP

Input: $\boldsymbol{w_j}, \boldsymbol{c_n}, \boldsymbol{u_f}, \forall j \in L_n, \forall f \in F$
Output: $w'_j, c'_n, \forall j \in L_n$

$c'_n \leftarrow \boldsymbol{c_n} \oslash \boldsymbol{u_1}$ **/* $\boldsymbol{u_1}$ → "smallest" VM flavor */**
for $f \in F$ **do**
  if ($\boldsymbol{c_n} \oslash \boldsymbol{u_f}$ is 0) **then** $v_f \gg c'_n$ **else** $v_f \leftarrow \lceil c'_n / (\boldsymbol{c_n} \oslash \boldsymbol{u_f}) \rceil$
  for $j \in L_n$ **do**
    if ((w'_j is $unassigned$) and ($\boldsymbol{w_j} \leq \boldsymbol{u_f}$)) **then**
      $w'_j \leftarrow v_f$
  end for
  if (w'_j is $unassigned$) **then** $w'_j \gg c'_n$
end for

B. Combinatorial Auction for Resource Allocation (CA-RA)

To complete the combinatorial auction-based mechanism, we wrap the winner determination problem with a payment scheme (*Algorithm 2*); for that we use the generalized VCG scheme. There, it is proven that if the winner determination problem is solved optimally, then for a bidder, the payment should be the sum of the declared bid values of other bidders minus the sum of such values that would have been obtained if the bidder had not participated in the auction. Therefore, the payment p_j of winning SFC j is defined as $p_j \leftarrow P_{-j} + r_j - P$, where P_{-j} is the optimal sum of bid values obtained from Eq. (5), had SFC j not participated in the auction. A desired property of a combinatorial auction mechanism is truthfulness i.e., the bidders benefit the most when they reveal their true valuations to the mechanism. Additionally, the winner determination algorithm needs to be monotone, i.e., a bidder only increases its chance of getting its requested bundle by bidding higher. A monotone allocation algorithm allows finding the *critical value* of a winning bidder, which is the minimum they need to bid to get their requested bundle.

Proof. To prove that CA-RA is monotone, we prove that for any SFC j, if r_j is a winning bid value, then every higher bid value $r'_j > r_j$ also wins, considering the other bids unchanged. Indeed, if we assume by contradiction that r'_j is not a winning bid, and let CA-RA produce the winners sets with optimal sum of bid values P and P', respectively, then we must have the case that $P' \geq P - r_j + r'_j > P$. This means that CA-RA should produce the winners set that correspond to P', which is a contradiction with the hypothesis. Thus, CA-RA is a monotone algorithm. The proof of the truthfulness property of the VCG payment scheme is detailed clearly in [10]. ■

Algorithm 2 CA-RA Mechanism

Input: $\boldsymbol{c_n}, \boldsymbol{u_f}, \forall f \in F$
Output: $x_j, p_j \forall j \in L_n$

for $j \in L_n$ **do** **/* Bids Collection */**
  Collect bids $(w_j^1, ..., w_j^M, r_j)$
end for
/* Winners Determination */
Reduce the MKP to SKP using *Algorithm 1*.
Solve the reduced resource allocation problem, (5).
for $j \in L_n$ **do** **/* Payments Computation */**
  if $x_{j,n} == 1$ **then** **/*** for every "winner" */
    $P_{-j} \leftarrow$ optimal sum of bids if $L_n \setminus \{j\}$ in (5).
    $p_j \leftarrow P_{-j} + r_j - P$
  end if
end for

C. Reinforcement Learning-based Self-Healing (RL-SH)

Summarizing, the proposed Self-Healing mechanism operates in two steps per iteration, as described in *Algorithm 3*: (i) each SFC autonomously selects a node to heal to, based on the probabilities of the SLA algorithm, Eq. (3), (4). (ii) A decision is made on which of these relocations can be successfully performed, by reducing the MKP to SKP and solving the combinatorial auction-based resource allocation problem (*Algorithm 2*). The outcomes of this decision are fed back to the SLA algorithm to update the rewards, Eq. (2), and probabilities, Eq. (3), (4), for the next iteration.

Algorithm 3 RL-SH Mechanism

Input: $\boldsymbol{w_s}, q_s, \boldsymbol{c_n}, \boldsymbol{u_f}, \forall s \in S, \forall n \in N, \forall f \in F$
Output: $x_{s,n}, p_{s,n}, d_{s,n} \forall s \in S, \forall n \in N$

$k \leftarrow 0$
while !$converged$ **do**
  for $s \in S$ **do** **/* SFC-to-Node Association (L_n) */**
    Select a Node n to heal to, based on the
    probabilities $\mathscr{P}_{s,n}^{(k+1)}$ defined in Eq. (3), (4).
  end for
  /* Reward Calculation */
  for $n \in N$ **do**
    Determine the successful SFC relocations $x_{s,n}$
    and respective payments $p_{s,n}$ using *Algorithm 2*.
  end for
  for $s \in S$ **do**
    Calculate reward $\mathscr{R}_{s,n}^{(k)}$ using Eq. (2).
    /* Probabilities Calculation */
    Update probabilities $\mathscr{P}_{s,n}^{(k+1)}$ using Eq. (3), (4).
  end for
  $k \leftarrow k + 1$
end while

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.

IV. Numerical Results

For the evaluation we consider a hybrid infrastructure, consisting of N interconnected Edge and Cloud nodes, where the Cloud nodes have significantly more available resource capacity, but are placed in greater distance than the Edge ones. Regarding the resources, a 3-dimensional capacity is assumed ($M = 3$), which corresponds to CPU, memory and bandwidth, in $[cores, GB, Gbps]$ respectively. For the Edge nodes, the available capacity $\boldsymbol{c_n}$ ranges from $[2, 4, 1]$ to $[24, 48, 10]$, while for the Cloud nodes from $[24, 48, 10]$ to $[96, 192, 40]$. Three VM flavors ($F = 3$) are offered in each node, $\boldsymbol{u_1} = [2, 4, 0.5]$, $\boldsymbol{u_2} = [4, 8, 1]$ and $\boldsymbol{u_3} = [8, 16, 2]$. For the experiments that follow, a single node is randomly placed out of order. The SFCs have different lengths, QoS and resource demands, while they are considered already deployed in the infrastructure prior to the outage. The resource requirements $\boldsymbol{w_s}$ of each VNF range between $[1, 2, 0.5]$ and $[8, 16, 2]$. Note that the number of the deployed SFCs and the functional nodes remain stationary throughout the simulation duration. The learning rate is $b = 0.6$, unless otherwise explicitly stated.

First, we investigate the performance of the CA-RA mechanism. To this end, we assume that each SFC has an available budget r_s that scales with the SFC's requirements and defines its bids; thus, it is randomly selected in the range $[0.10 \cdot w'_s, 0.30 \cdot w'_s]$, w'_s being the scalar SFC weight defined in Section III-A. We then implement a simple Fixed-Price payment algorithm, for which the payments computation part of *Algorithm 2* is replaced with the following: $p_j \leftarrow G(w'_j)$ if $r_j \geq G(w'_j)$ and $p_j \leftarrow 0$ otherwise, $G(w'_j)$ being the fixed price that the node has defined for the VM flavor that matches the resource requirements of SFC j (*Algorithm 1*, line 6). We consider and compare CA-RA with three different fixed-price schemes: a sublinear one, where the fixed VM price grows sublinearly with the VM flavor size, a linear one where the growth is linear and a superlinear one with an exponential price growth. For this comparison, 100 repetitions of the experiment were executed with $|N| = 35$ nodes and $|S| = 20$ SFCs, with fluctuating budgets in the given range and the results where averaged. In Fig. 3, we see that the proposed mechanism outperforms the others both in average successful healing ratio and revenue collected by the nodes of the infrastructure, while maintaining the average cost (price paid by each SFC) lower than the rest. In detail, if we consider a linear fixed-price as the baseline, changing to a sublinear pricing results to a higher healing ratio but decreases the generated revenue, as SFCs pay less than what they pay in the linear case. Naturally, the opposite trend is observed for the superlinear pricing scheme. All in all, by altering the fixed pricing scheme we can only improve one healing aspect while sacrificing another. This happens because the fixed-price schemes consider those SFCs who bid at least the fixed value, while CA-RA determines prices based on the market demand and supply.

Next, we evaluate the proper functioning of the RL-SH iterative optimization process in whole. Here, a Monte Carlo simulation over various infrastructure and SFC configurations is performed. This simulation includes 100 repetitions for each scenario and the results are averaged and indicatively presented in Fig. 4. In particular, in Fig. 4a, the convergence behavior of the SLA algorithm is showcased, in an infrastructure consisting of $|N| = 15$ nodes. For the first iteration, the selection of the healing node for each SFC is performed randomly, as their probabilities $\mathscr{P}^1_{s,n}$ are equal, $\forall s \in S, \forall n \in N$. We observe that the algorithm quickly learns which placement maximizes the reward function for each SFC, in approximately 50 SLA iterations. However, the bigger the number $|S|$ of SFCs that needs healing, the less the mean reward achieved, as finding a solution that minimizes both the cost and the additional delay while satisfying the resource requirements for each SFC becomes more challenging.

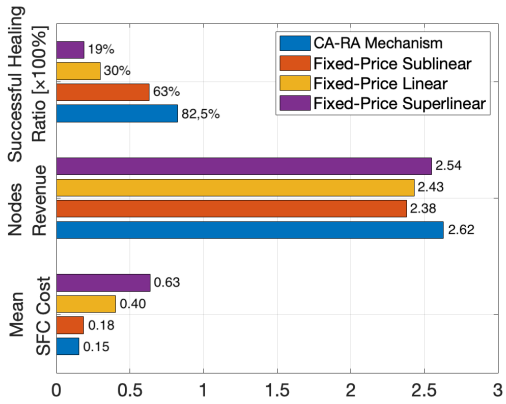
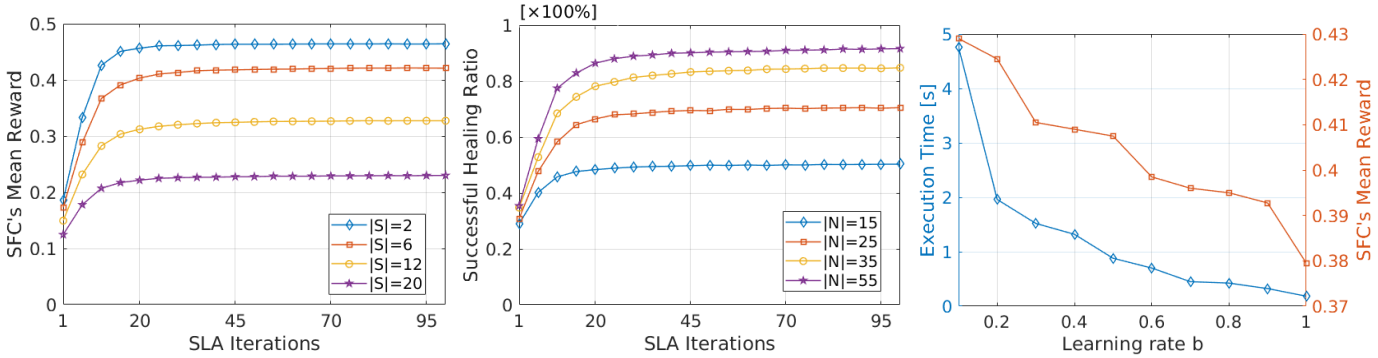


Fig. 3: CA-RA compared to different Fixed-Price schemes.

Fig. 4b presents the successful healing ratio for $|S| = 20$ SFCs, per SLA iteration, for various sizes of the infrastructure. Once again, the SLA algorithm quickly learns the most fitting solution for each SFC, as from an initial 30-35% mean success ratio we move to a 82-90% one, when the search space is adequate for this setting ($|N| \geq 35$). Regarding the performance and speed of convergence of the SLA-based Self-Healing algorithm, the results of a brief analysis are presented in Fig. 4c ($|S| = 20$, $|N| = 35$, fixed). The execution time is calculated from the moment the node outage is detected until a satisfying healing solution is found for every SFC. Apparently, as the value of the learning rate b decreases, the exploration of the possible SFC-to-node healing alternatives becomes exhaustive. This results in increasing real execution times, yet produces slightly higher mean rewards for the SFCs.

In the last part of the evaluation, we perform a comparison of the SLA-based Self-Healing algorithm in whole, with three baseline solutions: a randomly selected healing ("Random"), a healing solution among the neighboring nodes of the out-of-order node ("Distance") and a healing solution among the nodes with the most available resources ("Capacity"). Once again, the results are averaged over 100 experiment repetitions and presented in Fig. 5 ($|S| = 20$, $|N| = 35$, fixed). Specifically, in Fig. 5a the dominance of the proposed solution is demonstrated; the SLA-based algorithm scores twice the healing success ratio of the compared solutions, which mostly reside to healings that either satisfy the SFCs' resource or QoS requirements. This ability of better satisfying

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.



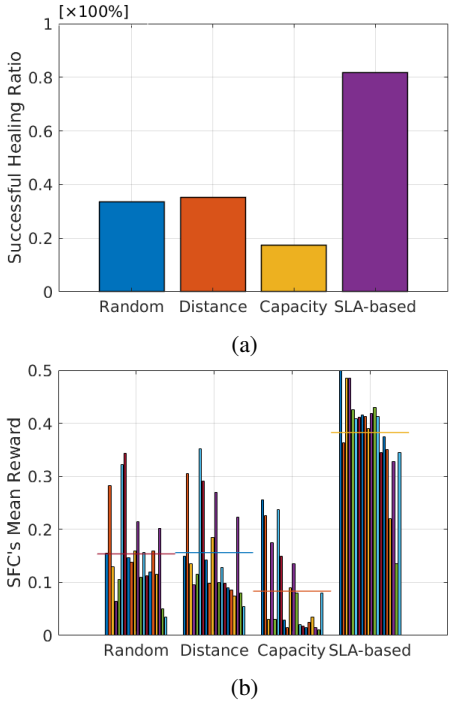
(a) (b) (c)

Fig. 4: Overall performance of the RL-SH mechanism.

the budget, resource and QoS requirements is also reflected in Fig. 5b, where the individual mean rewards gained per SFC are illustrated, together with their average values.

V. Conclusion

In order to address the resilience concern of an Edge-Cloud infrastructure, we proposed a RL-enabled, Self-Healing mechanism, that recovers the SFC connectivity, in a seamless manner in real time. The main idea, was to enable the automatic recovery of SFC-based services in the face of failures, while respecting their budget, resource and QoS requirements. Thus, a distributed solution was proposed, where the service vendors act autonomously and the infrastructure providers sensibly allocate their available resources to maximize their revenue, by offering them through combinatorial auctions. The effectiveness of the proposed mechanism was evaluated under various configurations. Future work will focus on investigating and incorporating failure detection and diagnosis mechanisms that would complement the proposed solution into a holistic network resiliency framework. Additionally, we will look into ways of implementing the auctioning mechanism in a realistic way for an Edge-Cloud, SFC-enabled infrastructure.



(a)

(b)

Fig. 5: RL-SH compared to three baseline algorithms.

References

[1] A. Leivadeas, M. Falkner, I. Lambadaris, and G. Kesidis, "Optimal virtualized network function allocation for an SDN enabled cloud," *Computer Standards & Interfaces*, vol. 54, pp. 266–278, 2017, sI: Standardization SDN&NFV.

[2] A. Asghar, H. Farooq, and A. Imran, "Self-healing in emerging cellular networks: Review, challenges, and research directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1682–1709, 2018.

[3] J. P. Santos, et al., "SELFNET Framework self-healing capabilities for 5G mobile networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1225–1232, 2016.

[4] S.-Y. Song and F. J. Lin, "Dynamic Fault Management in Service Function Chaining," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020.

[5] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, 2019.

[6] M. Bunyakitanon, X. Vasilakos, R. Nejabati, and D. Simeonidou, "End-to-end performance-based autonomous VNF placement with adopted reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 534–547, 2020.

[7] X. Yang, P. Yu, L. Feng, F. Zhou, W. Li, and X. Qiu, "A deep reinforcement learning based mechanism for cell outage compensation in 5G UDN," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 476–481.

[8] J. Guo, Z. Wang, X. Shi, X. Yang, P. Yu, L. Feng, and W. Li, "A deep reinforcement learning based mechanism for cell outage compensation in massive IoT environments," in *2019 15th International Wireless Communications & Mobile Computing Conf. (IWCMC)*. IEEE, 2019.

[9] E. H. Clarke, "Multipart pricing of public goods," *Public choice*, pp. 17–33, 1971.

[10] T. Groves, "Incentives in teams," *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

[11] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1–21, 2004.

[12] C. Unsal, "Intelligent navigation of autonomous vehicles in an automated highway system: Learning methods and interacting vehicles approach," Ph.D. dissertation, Virginia Tech, 1997.

[13] M. Diamanti, E. E. Tsiropoulou, and S. Papavassiliou, "Resource Orchestration in UAV-assisted NOMA Wireless Networks: A Labor Economics Perspective," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[14] D. Pisinger, "Where are the hard knapsack problems?" *Computers & Operations Research*, vol. 32, no. 9, pp. 2271–2284, 2005.

Authorized licensed use limited to: Carleton University. Downloaded on December 28,2023 at 17:24:38 UTC from IEEE Xplore. Restrictions apply.