HAL
open science

A network tomography approach for anomaly localization in Service Function Chaining

Mohamed Rahali, Jean-Michel Sanner, Cao-Thanh Phan, Gerardo Rubino

To cite this version:

Mohamed Rahali, Jean-Michel Sanner, Cao-Thanh Phan, Gerardo Rubino. A network tomography approach for anomaly localization in Service Function Chaining. ISNCC 2021 - International Symposium on Networks, Computers and Communications, Oct 2021, Dubai, United Arab Emirates. pp.1-6, 10.1109/ISNCC52172.2021.9615786 . hal-03507232

HAL Id: hal-03507232

https://inria.hal.science/hal-03507232v1

Submitted on 3 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A network tomography approach for anomaly localization in Service Function Chaining

Mohamed Rahali[1], Jean-Michel Sanner[1], Cao-Thanh Phan[1], and Gerardo Rubino[2]

[1]Firstname.Lastname@b-com.com, IRT B<>COM, Rennes, France
[2]Gerardo.Rubino@inria.fr, INRIA Rennes – Bretagne Atlantique, France

***Abstract*—The network slicing concept (probably one of the most important innovation brought by 5G) promises significant flexibility and autonomy for network management. Thanks to its main key features, heavily relying on the NFV and the SDN technologies, new communication services can be designed and deployed much faster than before. However, maintaining the reliability level of conventional networks remains a major open problem. One of its consequences is that the monitoring of the network infrastructure dedicated to this class of services is an essential challenge, which we address in this paper.**

In this paper we describe a new monitoring procedure, customized for NFV-based network infrastructures deployed with the Service Function Chaining (SFC) mechanism, one of the most important key enablers for NFV networks. Our solution allows the deployment of efficient probing schemes that guarantee the localization of multiple simultaneously failed nodes with a minimum cost. This is formulated as a graph matching problem and solved with a max-flow approach. Simulations show that our solution localizes the failed nodes with a small rate of false positives and false negatives.

I. Introduction

The emergence of 5G networks brings the promises of enhanced services with higher throughput, massive connectivity, and lower latency. But the main breakthrough of the technology is network slicing, the capacity of partitioning the network resources totally relying on software, providing specific virtual networks to each client in a very flexible way. This allows to offer customised network infrastructures to satisfy the different requirements of the business offers.

The network virtualization concept aims to offer more flexibility and scalability while decreasing the Capital Expenditures (CAPEX) and Operating Expenses (OPEX) of the network infrastructure. The classic network functions are implemented as a software that can be carried out over generic hardware. Thus, network resources can be dynamically allocated and automatically scaled according to the clients' demands. This flexibility allows managing efficiently network resources by hosting multiple services with varied requirements on the same physical infrastructure. Network virtualization technologies has thus continuously evolved, to reach nowadays the modern concept of slices.

The VNFs composing the slices hosted by the NFV Infrastructure (NFVI) will share the same set of resources and they can be used by multiple services. These VNFs must be executed in a specific order to provide those end-to-end services. The concept of connecting multiple VNFs in an ordered chain to compose an end-to-end service with specific QoS requirements is known as Service Function Chaining (SFC) [1]. The SFC mechanism is complementary to the NFV forwarding graph concept since it enables the interconnection of multiple virtual functions over different domains in a dynamic and flexible way.

In this paper we propose to design a lightweight and scalable probing solution based on end-to-end measurements. Our methodology is particularly useful to detect problems in NFV infrastructures hosting networks service chains. It can be observed that it is always possible, in principle, to infer the state of the intermediate nodes from end-to-end metrics by taking advantage of the correlations created by the shared nodes. However, there are some conditions about the network topology and the collected measurements to be respected in order to ensure accurate estimations.

The process of inferring node or link metrics from end-to-end measurements is called network tomography [2]. Multiple works have been proposed, essentially in the last two decades, to solve this problem following different approaches. Most of these works consider general network topologies and do not consider the specificities of virtual networks infrastructures. In this paper, we propose a tomography monitoring solution that takes into consideration the particularities of the SFC deployment mechanism. We take advantage of the previously proposed works in network tomography for node failure detection and we adapt them for the characteristics of NFV-based infrastructures.

The remainder of the paper is organized as follows. Section II overviews the state-of-the-art on failed node localization. Section III describes the context of our contributions and the problem model formulation. Section IV, presents the main components of our solution, namely the SFC probing designer and the inference algorithm. Section V provides a description of the performance evaluation method and the obtained results. Finally, Section VI concludes the paper and outlines some perspectives for future developments.

II. Background and related work

In this section, we study the existing strategies to identify faulty nodes in a network infrastructure with a special focus on NFV network monitoring.

In [3], the authors propose a Machine Learning approach for anomaly detection in virtual network functions. The proposed

solution includes three main functions: anticipate failures by detecting the first signs of SLA violation, detecting the SLA violation, and identifying the root cause. This tool can help the network administrator to make efficient and fast troubleshooting and take remediation actions like rebooting a Virtual Machine (VM) or scaling the allocated resources. The approach is based on supervised learning models and the training dataset is built by monitoring data coming directly from the VMs. A fault injection process is used to emulate anomalies and provide abnormal states since the collected data obviously represents mostly the normal state. Paper [4] proposes another statistical learning solution for VNF anomaly detection. The idea here is to collect metrics by continuously monitoring the deployed VNFs and then predict the next values using a regression model. Afterward, the predicted values are compared to the reported ones in order to detect any deviation in VNFs behavior.

In [5], the proposed solution tries to find the appropriate positions of monitors and probing paths between them enabling to cover a given network topology and, then, to detect any misbehavior. The problem is formulated as an ILP model, where the goal is to reduce the monitoring cost. The problem is solved using a heuristic approach. An exact solution was proposed before in [6] for the same problem, but the methodology is intractable for large topologies.

Recently, some attention has been accorded to Boolean network tomography. The main idea is to infer the state of single points or links in a network from end-to-end Boolean metrics. In [7] the authors introduce the application of Boolean network tomography for node failure localization in a general network topology. They first define an *"identifiability"* metric to measure the maximum number of simultaneously failed nodes in a given topology. Then, they give necessary and sufficient conditions to localize failure points under different probing schemes. In [8], the authors studied a similar problem. The proposed solution gives some insights for the design of monitoring strategies that maximize the number of identifiable nodes under different probing scenarios. Finally, let us cite [9] that studied the impact of topology properties on the maximal identifiability. They establish a relation between the different typologies classes (directed, undirected, trees, d-dimensional grids, and bounded-degree graphs) and the identifiability of Boolean metrics.

III. General context and problem formulation

A. Service function chaining

The Service Function Chaining (SFC) is a mechanism that enables connecting multiple network functions to create an end-to-end service, taking advantage of the virtualization techniques and the flexibility offered by SDNs. It is a key enabler for NFV networks and provides a fast an efficient tool for fast service deployment. Fig. 1 depicts the main components of the SFC architecture. The first element in the service chain is the Classifier which is responsible for applying the traffic steering policy to match the coming flows with the needed Service Functions (SFs) where the packets can have specific processing. The SFs can be, for example, a firewall or a Deep Packet Inspection (DPI). A Service Function corresponds to a VNF (or to multiple connected VNFs) in the NFV MANO architecture. Finally, the Service Function Forwarder (SFF) is responsible for forwarding the traffic to the SFs or to the next SFFs according to the SFC encapsulation information. The path taken by a packet formed by the SFFs and SFs is called the Service Function Path (SFP).

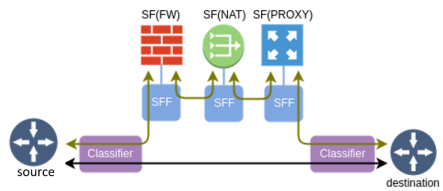


Fig. 1: Network services chains deployed with the SFC mechanism and sharing the same VNFs.

B. Failure detection in NFV networks

Fig. 2 illustrates a simplified example of a virtual network infrastructure deployed with the SFC mechanism, where the deployed services share a set of VNF instances. For each service, a tunnel is created between three VMs, where each VM represents a VNF instance. The mapping of the created services over the NFVI can look like the graph shown in Fig. 2.

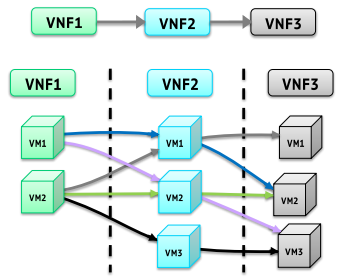


Fig. 2: Virtual Network Function Infrastructure example: The service is composed of three VNFs. Each one is hosted by multiple VMs. For each client, a tunnel is created between three VMs to compose the service.

If there are some failed nodes in the NFVI, efficient troubleshooting to identify the failure root causes can be a complex task. In such cases, the use of end-to-end metrics can be a relevant alternative to enhance the efficiency of the monitoring system. The misbehavior of a VNF can result from memory leaks, disk access problems, network problems, heavy workload, etc [3]. A failed node in the NFVI can lead to the disruption of all the service flows passing through. Therefore, we can take advantage of the correlated end-to-end collected metrics to infer the state of intermediate nodes. However, there are some required conditions to fulfill while designing the monitoring system in order to ensure the efficiency of such methods.

The efficiency of a tomography monitoring system can be evaluated by its capacity to localize accurately the maximum of simultaneously failed nodes. The scope of this paper is to give a methodology to design efficient probing schemes to monitor NFV networks deployed with the SFC mechanism. Thus, the most important step is to determine the required conditions to detect up to k simultaneous failed points in an NFVI hosting multiple network services. The parameter k is an input of the procedure that we will describe. Then, we shall benefit from these conditions to design a monitoring strategy for NFV networks.

C. Network model and notation

TABLE I: **List of main used variables**

variable	description
$\mathcal{N}$	set of *VNFs* composing the service; $\mathcal{N}^i$: the ith VNF
$\mathcal{V}^i$	virtual network function instances representing $\mathcal{N}^i$
$\mathcal{V}^i_j$	instance j of $\mathcal{V}^i$
$\mathcal{L}_{i,j}$	set of links between $\mathcal{V}^i$ and $\mathcal{V}^j$
$\mathcal{V}, \mathcal{L}$	set of VMs and set of links between them
$\mathcal{P}$	path set, P paths
$G = (\mathcal{V}, \mathcal{L})$	network topology
$\mathcal{F}$	set of failed nodes; cardinality of $\mathcal{F} = F$

Tab. I summarizes the notation adopted in this paper. We consider a network service decomposed into a set $\mathcal{N} = \{\mathcal{N}^1, \ldots, \mathcal{N}^N\}$ of N VNFs. To respond to many service requests, the VNFs are instantiated over a set $\mathcal{V}$ of VMs. We denote by $\mathcal{V}^i$ the set of VMs where the VNF $\mathcal{N}^i$ is instantiated. The cardinalities of $\mathcal{V}$ and $\mathcal{V}^i$ are denoted by V and V^i respectively. The whole system represents the NFV Infrastructure (NFVI).

The created service chains make that there is no connection between VMs inside a same $\mathcal{V}^i$ (no "vertical" links in the picture of Fig. 2). Links only exist between VMs in $\mathcal{V}^i$ and $\mathcal{V}^{i+1}$ for $i = 1, 2, \ldots, N-1$. Of course, every VM in $\mathcal{V}^i$ has at least a connection with another VM in $\mathcal{V}^{i+1}$, $1 \leq i \leq N-1$ (if $N \geq 2$). We denote by $\mathcal{L}$ the set of all links (size L), and by $\mathcal{G}$ the graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$.

A node $\mathcal{V}^i_j$ can be either in a failed state (or down) or in a normal state (or up). We denote by X the Boolean vector representing the states of all the nodes, thus a vector with size V. The binary value 1 corresponds to the failed state and 0 to the up state. Let us denote by p a path in the graph corresponding to a chain of VNF instances composing the service. It is represented by a Boolean vector having size V, where component i, $p(i)$, is 1 if node i belongs to p, and to 0 otherwise. The state of a traffic path is up if all the nodes composing it are up. Otherwise, it is a failed path. We denote by Y the Boolean vector representing the paths' states. The goal is to estimate the states of the intermediate nodes from the end-to-end path metrics. Let $\mathcal{P}$ denote the set of such paths, with cardinality P, denoted $p^1, p^2, \ldots, p^P$. Let A denote the Boolean matrix whose rows are the P path vectors. Thus, $A(i, j)$ is equal to 1 if node j belongs to path p^i, and to 0 otherwise.

With this notation, between the nodes' states in X and the path states' in Y, we have the relation

$$A \odot X = Y, \tag{1}$$

where $\odot$ denotes the Boolean matrix product. $Y(r)$, the state on the rth path (row) is 0 if all the nodes composing it are up, i.e., $Y(r) = \vee_{j=1}^{V}(A(r, j) \wedge X(j))$. In next section, we describe how to build a probing schemes that guarantees the localization of a maximum number of faulty nodes.

IV. The SFC probing scheme designer

This section presents the main components of our solution. In the first subsection, we discuss about the tomography problem considered in the paper, that will make the connection with our proposal that starts in Subsections B and C. Subsection D discusses complexity issues.

A. Necessary and sufficient conditions for k-failures detection

The end-to-end paths start from a VM on the left side (in the subset of nodes $\mathcal{V}^1$), and end on some VM on the right side (in $\mathcal{V}^N$). Let us call here *blocks* the N sets of nodes (or VMs) $\mathcal{V}^1, \mathcal{V}^2, \ldots, \mathcal{V}^N$. Once the links built between the VMs instantiating the VNFs that compose the considered network service, we can monitor any path following any sequence of nodes $v_1, v_2, \ldots, v_N$, where v_i belongs to block $\mathcal{V}^i$, provided there is a link (belonging to some tunnel) from v_i to v_{i+1}, $i = 1, 2, \ldots, N-1$, even if $(v_1, v_2, \ldots, v_N)$ is not a tunnel itself (that is, in Fig. 2, even if the links don't have the same color).

In the sequel, we were inspired by the developments made in [7] where the problem of node failures detection is considered in a general setting and for an arbitrary network topology. As in other works related to the failure of network components (for instance, in fault tolerant design, as in [10]), it is often useful to decompose the problem in the cases where the number of failures is bounded by some integer k. The failure of a node or a link is always an event having a small probability, and the simultaneous failures of two or more components is much smaller, even if the considered events are not necessarily independent. So, we will organize the discussion by considering as a global assumption that the number of failures, here of node failures, is $\leq k$ for some fixed integer $k \geq 1$. Since we are in a Boolean domain, observe that when a path is measured as up (value 0), we know that all its nodes are necessarily up. Now, for the design of our approach we focus on the following property: for a selected node v, we consider the fact that for some path p through v, all nodes excepting v are up. In that case, the state of p is the state of v: if v is up, then all nodes in p are up and so is p, and when v fails, then p fails as well. The next propositions provide conditions under which this property holds.

Proposition 1 (Sufficient condition)**.** *Let us assume that the number of failed nodes is $\leq k$, for some integer $k \geq 1$. If for all node $v \in \mathcal{V}^i$ its interior degree (for $i \geq 2$) and its exterior degree (if $i \leq N-1$) are both $> k$, then there exist a path p containing v such that its state is the state of v. The interior*

degree of a node denotes the number of incoming arcs and the exterior degree is the number of outgoing arcs.

Proof. Say $v \in \mathcal{V}^i$. If $i \geq 2$, since v has at least $k+1$ neighbors in $\mathcal{V}^{i-1}$ and since there are at most k nodes down, there is at least one of these neighbors up; let us denote it by n_{i-1}. In the same way, n_{i-1} has at least one neighbor up in $\mathcal{V}^{i-2}$, and so on. The same reasoning is valid when we go to subsets $\mathcal{V}^{i+1}$, $\mathcal{V}^{i+2}$, etc. So, we can build a path $p = (n_1, n_2, \ldots, n_{i-1}, v, n_{i+1}, \ldots, n_N)$, where all nodes except v are up. This means that the state of the path coincides with the state of v: if v is up, p is up, and if the node is down, p is down as well. □

Proposition 2 (Necessary condition)**.** *Let us assume that the number of failed nodes is $\leq k$, for some integer $k \geq 1$. A necessary condition to have the property that for all node v there exist a path p containing v such that p's state is equal to v's state, is that for every node in $\mathcal{V}^i$ its interior degree (for $i \geq 2$) and its exterior degree (for $i \leq N-1$) are both $\geq k$.*

Proof. The proof is by contradiction. We will denote by $deg^-(v)$ the in-degree or interior degree of node v, and by $deg^+(v)$ its out-degree or exterior degree.

Consider first the case of $v \in \mathcal{V}^i$ for some $i \geq 2$, with $deg^-(v) < k$. The existence of path p passing through v with all nodes up except possibly v is in contradiction with previous assumption, because it can happen that all $v' \in \mathcal{V}^{i-1}$ connected to v are down (this is because we can have up to k failed nodes and we assumed that $deg^-(v) < k$). The case of $v \in \mathcal{V}^i$ for some $i \leq N-1$ with $deg^+(v) < k$ is treated exactly in the same way. □

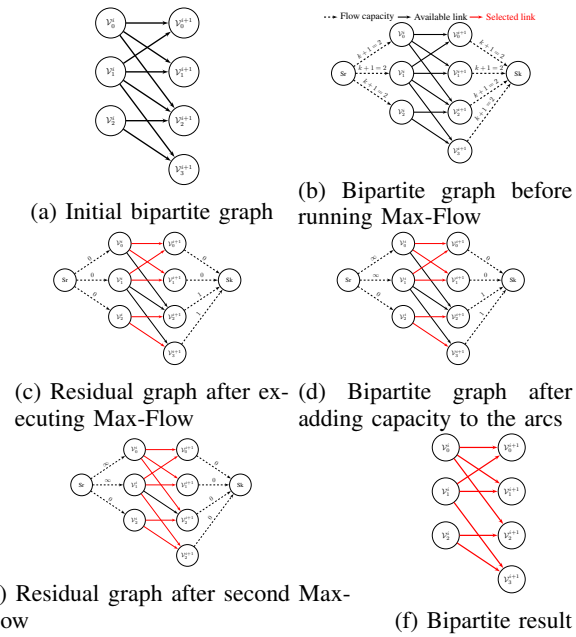
Previous properties are related to more general ones in [7]. In that paper, the discussion covers arbitrary topologies and, as a consequence, the development is more complex, requiring other auxiliary concepts related to the problem. Here, our specific architecture of the NFVI allows us to exhibit very simple properties that lead to the procedures that we describe in next subsection.

Let us denote by d the smallest degree in or out in graph $\mathcal{G}$: $d = \min\{deg^-(v),\ deg^+(v), \text{for all node } v\}$. Observe that the sufficient condition in Proposition 1 means that $k \leq d-1$ and the necessary one in Proposition 2 means that $k \leq d$.

B. Probing scheme design as a matching problem

After defining the conditions on the probing topology that guarantee the localization of the failed nodes, this section proposes an algorithm to build a topology with the minimum number of links that respects these conditions. This simplifies the monitoring operation and reduces the costs since less bandwidth is used.

Let us consider two consecutive blocks $\mathcal{V}^i$ and $\mathcal{V}^{i+1}$ and the links built from the first to the second, which we denote by $\mathcal{L}_{i,i+1}$. Consider the problem of selecting a subset of these links in order to fulfil the sufficient condition given in Proposition 1. In other words, we consider here that we can take any existing link from any node in $\mathcal{V}^i$ to any other node in $\mathcal{V}^{i+1}$ and we want to select a subset of those satisfying the sufficient condition. This means that we have chosen some k satisfying $k \leq d-1$ where d is defined considering all existing links before this selection operation, and that in the given graph, all degrees are $\geq k+1$.



(a) Initial bipartite graph

(b) Bipartite graph before running Max-Flow

(c) Residual graph after executing Max-Flow

(d) Bipartite graph after adding capacity to the arcs

(e) Residual graph after second Max-Flow

(f) Bipartite result

Fig. 3: Matching algorithm illustration. The values on the dotted arcs are capacities, either the initial ones, or the residual ones after a Max-Flow execution. The arcs between blocks have obviously capacity 1.

Previous selection operation between two sets is an instance of a classical matching problem in graph theory, typically solved using a Max-Flow technique. We will solve the problem in this way. The algorithm steps are illustrated in Fig. 3 using the case of $k = 1$.

The **first step** of the process is to add two "fictitious" nodes to the two sets, named Source (Sr) and Sink (Sk), as illustrated in Fig. 3b. The source Sr is connected to all the nodes in $\mathcal{V}^i$ with a flow capacity equal to $k+1$ (= 2 in the example), and similarly for Sk and $\mathcal{V}^{i+1}$. As typically done in matching graph problems, the capacity of the links from $\mathcal{V}^i$ to $\mathcal{V}^{i+1}$ is 1. This limits the number of links going in and out each node to be no more than $k+1$ which in turn enables to limit the number of used links.

In the **second step**, we use a conventional Max-Flow algorithm, like the Ford-Fulkerson or the Edmonds-Karp algorithms [11] to find a maximal flow from Sr to Sk, which we denote by $\mathcal{M}^0_{i,i+1}$. In the obtained flow, if we remove from the graph the arcs where the flow function is 0, the degrees calculated with the remaining arcs do not necessarily satisfy the sufficient condition. In the example given in Fig. 3, where we color in red the selected arcs (see Fig. 3c), the In-degree of $\mathcal{V}^{i+1}_2$ and $\mathcal{V}^{i+1}_3$ is only 1, since the number of links going out of $\mathcal{V}^i_0$ and $\mathcal{V}^i_1$ is limited to 2. We need more links from

and to these nodes.

To avoid such a situation, in the **third step** we add some capacity to specific arcs as illustrated in Fig. 3d. For each node v in $\mathcal{V}^i$ where $deg^+(v) < k+1$, we add capacity to all the arcs from the nodes in $next(v)$ to the sink Sk, where $next(v)$ denotes the nodes in $\mathcal{V}^{i+1}$ that receive arcs from v. Similarly, for each node v in $\mathcal{V}^{i+1}$ for which $deg^-(v) < k+1$, we add capacity to all edges from the source Sr to the nodes in $prev(v)$, where $prev(v)$ denotes the nodes in $\mathcal{V}^i$ that send arcs to v. We actually set the new capacity value of all these arcs to ∞ since there is no risk to select new useless arcs in $\mathcal{L}_{i,i+1}$.

Then, in the **fourth step** illustrated in Fig. 3e, we compute for a second time the MaxFlow between Sr and Sk. It is worth noting that the arcs selected in the first call to the MaxFlow algorithm (in the second step) cannot be reused. The obtained maximum flow is denoted by $\mathcal{M}^1_{i,i+1}$.

This matching process is repeated for each two consecutive blocks. The final obtained graph, denoted by $\mathcal{M}$, comes from composing the optimal matching obtained for each pair of consecutive blocks. We will use this new graph, a partial graph of the initial one (same nodes, part of the links) to localize up to k failures in the considered network.

C. Probing paths selection

After designing the topology that guarantees the location of failed nodes in the network infrastructure, an inference algorithm is required to analyze the end-to-end measurements collected on this topology and then estimate the state of each intermediate node. For this purpose, we use an inference algorithm called ESA [12] that calculates metrics on intermediate nodes from end-to-end measurements. The state of the nodes is given in the form of a probabilistic distribution that indicates the probability of each possible value of the metric.

D. Complexity analysis

The computation complexity of the described process is essentially determined by the Max-Flow algorithm. If we denote by L_{moy} the average number of links between each consecutive pair of sets $\mathcal{V}^i$ and $\mathcal{V}^{i+1}$, and by V_{moy} the average number of vertices in each set $\mathcal{V}^i$, the complexity of the Max-Flow algorithm (Edmonds-Karp for example) can be approximated by $O(V_{moy}L^2_{moy})$. It will be invoked twice. This process is repeated $N-1$ times, corresponding two each pair of VNFs. Indeed, the global complexity of can be approximated by $O(2NV_{moy}L^2_{moy})$. As a result, the computing time increases polynomially with L_{moy} and linearly with the other parameters V_{moy} and N.

V. Evaluation

A. Methodology

Our solution allows to design a probing strategy enabling to localize accurately a maximum number of failed nodes. To evaluate the robustness of the designed slice topologies, we proceed as follows. First, the graph $\mathcal{G}$, which corresponds to the NFVI topology, is randomly generated. Second, the matching algorithm is applied to select a minimum number of links enabling the detection of up to a fixed and given number k of failures upper-bounded by the min degree d. Third, multiple samples of nodes states are simulated, and we compute the end-to-end Boolean path states. The number of simulated failures is varied to check the algorithm's behavior in different situations.

Finally, the inference algorithm estimates the node states from end-to-end simulated measurements, and the estimations are compared to the real states to evaluate the accuracy of the monitoring system.

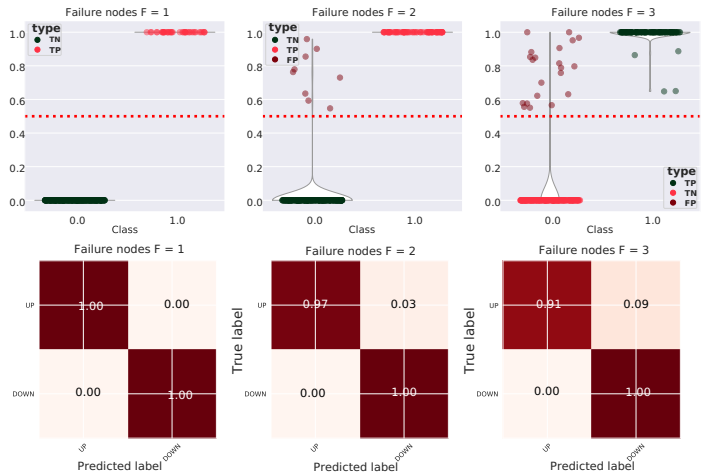


Fig. 4: Failure probability estimations and confusion matrix for 50 tests made with service chain A. The parameter k is fixed to 1, which means that the topology is designed to guarantee only the localization of only one failed node. In the tests, we vary the number of simultaneously failed nodes denoted, from 1 to 3.

B. Results

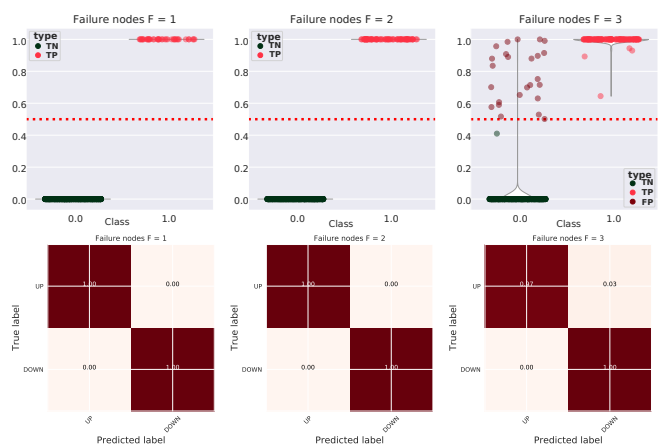


Fig. 5: Failure probability estimations and confusion matrix for 50 tests made with service chain B. The parameter k is equal to 1. We vary the number of failed nodes F from 1 to 3.

Consider first the case of a network service chain A composed of 3 VNFs. Each network function has between 4 and 5 instances. The connections between the two consecutive sets of VNF instances are generated randomly. Each VNF instance has around 3 connections with the previous and the next sets of VNF instances. The matching algorithm selects a minimal number of connections enabling the detection of up to k simultaneous failed VMs, and k is fixed to 1 in these tests. Afterward, we simulate the node metrics as explained before. We vary the number of generated node failures from 1 to 3.

Fig. 4 illustrates the probability dispersion and the confusion matrix for the node states made with 50 tests. With only one failure node, the monitoring system can localize them accurately in all the tests. These results are expected since the tested topology satisfies the sufficient condition for the detection of a single failed node. Therefore the number of false negatives and false positives is zero. However, this condition is not respected when more than one failure are generated as shown in the next tests, Fig. 4.

It is worth noting that if the sufficient condition for the detection of k failures is not satisfied, the monitoring system is still able to detect them if they occur. When the sufficient condition is not fulfilled, what we can say is that one or more situations (a combination of failed nodes) can exist where the monitoring system is unable to identify the failures in a deterministic way. The tests with 2 or 3 simultaneously failed nodes confirm this observation. In fact, with two and three failed nodes, the inference algorithm identifies all of them. However, the number of false positives increases a little bit to reach 3% and 9% respectively.

We make similar tests with a longer network service chain denoted by B, composed of 5 network functions. Fig. 5 illustrates the probability dispersion of the tests made with the second example. With only one failure, the monitoring algorithm identifies them accurately as in the first example. With two failures, the algorithm is still able to localize all malfunctions, which is not the case with the first example. This can be explained by the fact that the second topology is broader than the first one. Therefore, even if there are some cases where the monitoring system is unable to identify the unknown failure points, the probability of selecting these subsets of nodes during the simulations becomes extremely small due to the number of possible combinations that becomes very large. Then, starting from three failures, the numbers of false positives and false negatives start increasing.

VI. Conclusions

In this paper, we study the problem of monitoring in virtual network infrastructures deployed with the SFC mechanism, and we focus on node anomaly localization. We provide a general framework for the solution to this problem and exhibit necessary and sufficient conditions enabling the detection of up to k simultaneous failures in a NFV infrastructure for a given parameter k. These results serve to establish a probing design strategy.

The optimization process is formulated as a matching problem and solved with a classical Max-Flow algorithm. Finally, the end-to-end measurement performed on these topologies is the input for a Boolean metrics inference algorithm in order to estimate the state of each node and to localize the failure points. The simulation results are consistent with the developed model. The main work planned for the continuation of this work is to consider other possible ways of deploying a network of NFVs. Exploring in deep the behaviour of our procedures with much larger examples is another research task to be done in the close future.

Acknowledgements

This work has been partially supported by the European Union's H2020 MonB5G (grant no. 871780) project.

References

[1] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2016.09.001

[2] A. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," *Signal Processing Magazine, IEEE*, vol. 19, pp. 47 – 65, 06 2002.

[3] C. Sauvanaud, K. Lazri, M. Kaâniche, and K. Kanoun, "Anomaly detection and root cause localization in virtual network functions," in *27th IEEE International Symposium on Software Reliability Engineering, ISSRE 2016, Ottawa, ON, Canada, October 23-27, 2016*. IEEE Computer Society, 2016, pp. 196–206.

[4] M. Kourtis, G. Xilouris, G. Gardikis, and I. Koutras, "Statistical-based anomaly detection for NFV services," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, November 7-10, 2016*. IEEE, 2016, pp. 161–166.

[5] E. Salhi, S. Lahoud, and B. Cousin, "Heuristics for joint optimization of monitor location and network anomaly detection," in *Proceedings of IEEE International Conference on Communications, ICC 2011, Kyoto, Japan, 5-9 June, 2011*. IEEE, 2011, pp. 1–5.

[6] ——, "Joint optimization of monitor location and network anomaly detection," in *The 35th Annual IEEE Conference on Local Computer Networks, LCN 2010, 10-14 October 2010, Denver, Colorado, USA, Proceedings*. IEEE Computer Society, 2010, pp. 204–207. [Online]. Available: https://doi.org/10.1109/LCN.2010.5735702

[7] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe, "Node failure localization via network tomography," in *Proceedings of the 2014 Internet Measurement Conference, IMC 2014, Vancouver, BC, Canada, November 5-7, 2014*, C. Williamson, A. Akella, and N. Taft, Eds. ACM, 2014, pp. 195–208.

[8] N. Bartolini, T. He, and H. Khamfroush, "Fundamental limits of failure identifiability by boolean network tomography," in *2017 IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, May 1-4, 2017*. IEEE, 2017, pp. 1–9.

[9] N. Galesi and F. Ranjbar, "Tight bounds for maximal identifiability of failure nodes in boolean network tomography," in *38th IEEE International Conference on Distributed Computing Systems, ICDCS 2018, Vienna, Austria, July 2-6, 2018*. IEEE Computer Society, 2018, pp. 212–222.

[10] N. N. Jara, G. Rubino, and R. Vallejos, "A method for joint routing, wavelength dimensioning and fault tolerance for any set of simultaneous failures on dynamic wdm optical networks," *Optical Fiber Technology*, vol. 38, pp. 30–40, 2017.

[11] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225–231, 1973.

[12] M. Rahali, J. Sanner, and G. Rubino, "Unicast inference of additive metrics in general network topologies," in *27th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS 2019, Rennes, France, October 21-25, 2019*. IEEE Computer Society, 2019, pp. 107–115.