2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) | 978-1-7281-9290-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ANTS50601.2020.9342772

Availability Comparison of 5G Network Service

Priyatosh Mandal
Centre for Development of Telematics, New Delhi, India
priyotoshtsp@gmail.com

***Abstract*—5G network service is a set of related and connected virtual network functions. A complete network slice can be virtualized with the use of a network service. Virtual network functions (vnfs) of a network service can be deployed with the use of OpenStack, which is an open source cloud computing platform. In OpenStack, all the vnfs of a network service can be deployed in one compute node or in different compute nodes. In this present work, we derive closed form expression to analyze the availability of network service considering the deployment of vnfs of a network service in one and different compute nodes. In the availability analysis, we consider the failure perspective of compute node(s). Further, we compare the availability of network service considering these two deployment strategies.**

***Index Terms*—OpenStack, vnf, 5G, Availability, network service**

I. Introduction

With the use of cloud computing, it is possible to provide computing resources in an easy, efficient and cost effective way. The OpenStack project is an open source cloud computing platform, which can be used for delivery of virtualized computing resources. OpenStack provides an infrastructure-as-a-service solution through a set of closely related services. Standard interfaces are available to access any specific service. Depending on the capability of cloud computing platform to create, it is possible to install set of necessary services [1]. To create basic cloud computing platform to deploy network service, we need one controller node and one or more compute nodes.

The controller node is the most important node in OpenStack platform. Most of the services run on this node. The compute node is the node where vnfs are created depending on the requirement. There can be many vnfs in a system. These virtual network functions can be deployed as a part of network service or as an individual virtual network function. If vnfs are part of a network service, then it is easy to create and delete a set of vnfs of a network service.

In this present work, we use the term vnf to indicate a virtual machine which executes a specific software. Number of vnfs that can be created are limited by the total capacity in terms of cpu, memory and disk of all the compute nodes in the system. It is possible to create all the vnfs of a network service in single compute node and different compute nodes. All the vnfs of a network service can be deployed in a compute node, provided the resource requirement of all the vnfs are satisfied by the compute node. If the vnfs of a network service are deployed in different compute nodes, then the remaining resources of a compute node can be used for the deployment of vnfs of some other network service. These two approaches of vnf creations can be presented as in Fig. 1, where a network service consists of N vnfs. In first deployment scenario, all the vnfs of network service are deployed in a compute node. In second deployment scenario, all the vnfs are deployed in different compute nodes, i.e., vnf 1 is deployed in one compute node, vnf 2 is deployed in a compute node which is different from the compute node where vnf 1 is deployed and vnf N is deployed in a compute node, which is different from the compute nodes where vnf 1, $\cdots$, vnf $N-1$ are deployed.

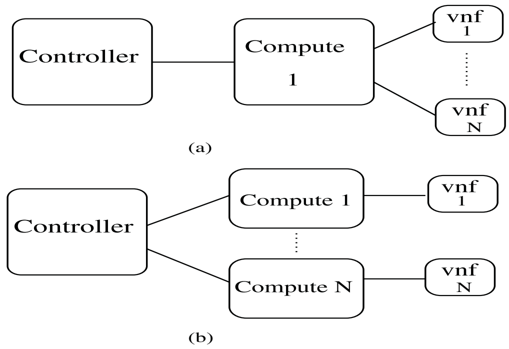

Fig. 1: Conceptual diagram of (a) multiple vnfs in one compute and (b) one vnf in each compute [1].

There are some works available in literature, which are related to this present work. Briefly we describe those here. Complete documents for OpenStack and network service with multiple vnfs creation procedures are available in [1]. In [2], availability definition is formulated considering three state continuous time markov chain (CTMC). The present work concerned about the availability of network service. Here the number of states of CTMC are depended on the number of vnfs of network service.

In [3], delay sensitive vnf scheduling and delay sensitive availability aware vnf scheduling problems are studied. Further, to solve these problems, integer non-linear programming based recursive algorithm is proposed. The procedure to find the traversing delay of

978-1-7281-9290-1/20/$31.00 ©2020 IEEE

a flow in case of totally ordered and partially ordered service function chaining are also described. In [4], a framework for proactive restoration mechanism to ensure service availability is proposed. Mathematical model is proposed to estimate the number of active and idle user equipments due to failure of mobility management entity of 4G-LTE network.

In [5], joint optimization of computing and communication resource for virtual network function placement is studied. Virtual network function placement problem is presented as NP hard. Then to solve this vnf placement problem, a heuristic resource efficient vnf placement algorithm based on hidden markov model is proposed. In [6], resource allocation related problems of network function virtualization are described in detail. It is described that efficient placement of vnfs can increase the overall performance.

In [7], comparison of various orchestration procedures of OpenStack are compared. In [8], performance of Openstack controller is analysed to decrease the delay. In [9], availability of virtual network function is analysed.

During the creation of a network service, it is possible to create all vnfs of a network service in one compute node. Alternatively it is also possible to create all vnfs of a network service in different compute nodes, i.e., no more than one vnf of a network service is part of a compute node. During the creation of vnfs, the compute node can be failed, which will force the controller node to recreate one or more vnfs depending on the number of compute nodes, which are used for the creation of network service.

The failure of compute nodes can be possible due to power failure of compute node or due to any other reason. The failure of compute node causes the failure of all vnfs corresponding to the compute node. So considering the failure of compute node(s) into account, we derive availability expression considering, network service with single compute node and network service with multiple compute nodes. Failure of compute nodes can be known at the controller node with the use of ping command or pacemaker package. As there is no work available, which compare the availability of network service from the perspective of compute node failure, so we analyze and compare the availability of network service considering single compute node with all the vnfs and multiple compute nodes with single vnf in each compute node corresponding to a network service.

A. *Contributions*

Our specific contributions in this paper are as follows:

(a) We derive closed form availability expression of network service considering all the vnfs of a network service in one compute node.

(b) We derive closed form availability expression of network service considering all the vnfs of a network service in different compute nodes.

(c) We analytically compare the availability of two versions of network service deployment.

B. *Paper organization*

In Section II, we describe the procedure to create a network service using OpenStack. In Section III, we analyze the availability of network service considering all vnfs of network service in single compute node and in different compute nodes. In Section IV, we describe analytical results and finally in Section V, we conclude the paper.

Assumptions: We assume one controller node and more than one compute nodes are always available to facilitates the creation of vnfs of a network service. In case a compute node with a vnf fails, then this vnf also fails. So to recreate this failed vnf, other compute nodes are always available.

II. Network service creation procedure using OpenStack

Network service is the set of interconnected virtual network functions. During creation of network service, we need to create network service descriptor. Network service descriptor calls virtual network function descriptors. As a result during the creation of network service using OpenStack, the vnfs which are part of network service can be created. In the virtual network function descriptor, we need to mention the operating system (i.e., image) to be used in the vnf and the capacity in terms of processing units (cpu), memory and disk (i.e., flavor) to create virtual machine. During creation of vnf, the virtual machine is created first. After creation of virtual machine, the necessary software has to be downloaded from a pre-decided software repository. After the availability of the software, the respective softwares has to be installed. After installation of the software, the respective software has to be executed. Then we call the vnf is completely created. In this way all the vnfs have to come up for the network service to be available.

To create virtual network functions and network service using OpenStack, we can use yaml file [1]. Here we briefly describe the content of yaml file to create vnf descriptor and network service. Detail description of network service creation is available in [1]. It specifies virtual deployment unit (VDU), virtual link (VL) and connection point (CP) to create a server. It is possible to create more than one VDUs using single vnf descriptor file. Corresponding to each VDU we need to mention CP and VL. In the VDU section, we specify the parameters to create virtual machine and in the user-data section we download and execute the software for the creation of complete vnf. We specify the content of yaml file as below. The yaml file where we mention the VDU, CP, VL information is called vnf descriptor. The VL is used to provide network connection and connection point is used to combine virtual link and virtual deployment unit.

```
VDUX:
    type: tosca.nodes.nfv.VDU.Tacker
    properties:
        name: name-of-vnf
        image: name-of-image
        flavor: name-of-flavor
        availability_zone: name-of-zone
        user_data_format: RAW
        user_data: —
            #!/bin/sh
            #Download the vnf specific software from a repository
            #Install vnf specific software and execute the vnf specific software
CPX:
    type: tosca.nodes.nfv.CP.Tacker
    properties:
        ip_address: ipaddress-of-vnf
    requirements:
        - virtualLink:
            node: VLX
        - virtualBinding:
            node: VDUX
VLX:
    type: tosca.nodes.nfv.CP.Tacker
    properties:
        network_name: name-of-network
        vendor: Tacker
```

We can create as many vnf descriptors as we want using the above procedures. Once vnf descriptors are created corresponding to each vnf, we call the vnf descriptors from the network service descriptor and finally we create network service with the use of network service descriptor. With the use of availability-zone parameter, which is one of the property of virtual deployment unit, it is possible to deploy vnfs of a network service in one compute node or different compute nodes.

III. Analytical Model

In this section, we analyze the availability of network service considering all vnfs in one compute node and all vnfs of a network service in different compute nodes. In both cases we use CTMC to derive the availability expression of network service. Notations used during the description of CTMC are present in Table I.

TABLE I: Summary of notations

λ	Failure rate of compute node
μ	Rate of creation of virtual network function
N	Number of virtual network functions in a network service

A. Virtual network functions of a network service are in one compute node

In this case, we analyze the availability of network service considering deployment of all vnfs corresponding to a network service in a compute node. We analyze this system using CTMC. Assume that the failure time distribution of compute node is exponential with mean $1/\lambda$, vnf creation time distribution is exponential with mean $1/\mu$ and there are N vnfs in a network service.

We consider the states as the number of vnfs which are created corresponding to a network service. Initially the system is in state 0, i.e., the virtual network function is not created. Once the request of network service creation is generated, all the vnfs are created independently with rate μ. So with rate $N\mu$ the system state changes from state 0 to state 1. After creation of one vnf, with rate $(N-1)\mu$ the system state changes from state 1 to state 2. In this way the change of states continue in the forward direction. With rate μ the system state $(N-1)$ changes into state N. During the creation of vnfs, there can be a possibility of failure of compute node. If the failure happens in a compute node, then all the vnfs has to be recreated in some other compute node. So the transition from any state to state 0 is possible with rate λ. The transition rate diagram for this network service creation is present in Fig. 2.

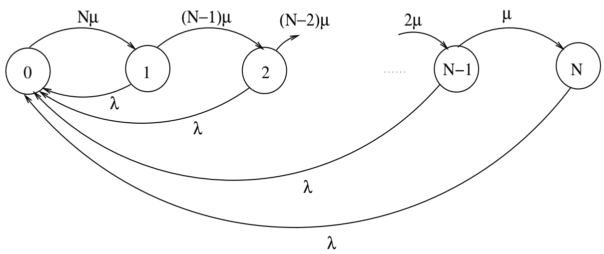

Fig. 2: Transition rate diagram of network service availability considering deployment of all vnfs of a network service in a compute node

We can write the flow balance equations from this transition rate diagram as below.

$$\begin{aligned} N\mu\pi_0 &= \pi_1(\lambda + (N-1)\mu) \\ (N-1)\mu\pi_1 &= \pi_2(\lambda + (N-2)\mu) \\ (N-2)\mu\pi_2 &= \pi_3(\lambda + (N-3)\mu) \\ &\cdots \\ 2\mu\pi_{N-2} &= \pi_{N-1}(\lambda + \mu) \\ \mu\pi_{N-1} &= \pi_N\lambda \end{aligned} \tag{1}$$

From the above flow balance equations we can write, for $n = 1, 2, \cdots, N$

$$\pi_n = \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{\lambda + (N-i)\mu} \right) \pi_0 \tag{2}$$

Since, $\pi_0 + \pi_1 + \cdots + \pi_{N-1} + \pi_N = 1$, we can write after simplification,

$$\pi_0 = \left[1 + \sum_{n=1}^{N} \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{\lambda + (N-i)\mu} \right) \right]^{-1} \tag{3}$$

So the network service availability, A_{v1}^{NS} $(= \pi_N)$ can be given by (4).

$$A_{v1}^{NS} = \left(\prod_{i=1}^{N} \frac{(N-i+1)\mu}{\lambda + (N-i)\mu} \right) \pi_0 \tag{4}$$

So the network service availability, A_{v1}^{NS} can be further simplified as in equation (5).

$$A_{v1}^{NS} = \frac{\prod_{i=1}^{N} \frac{(N-i+1)\mu}{\lambda+(N-i)\mu}}{1 + \sum_{n=1}^{N} \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{\lambda+(N-i)\mu} \right)} \tag{5}$$

B. Virtual network functions of a network service are in different compute nodes

In this case, we analyze the availability of network service considering the deployment of all vnfs of network service in different compute nodes. We analyze this system using CTMC. Here similarly like previous section, we consider that the failure time distribution of compute node is exponential with mean $1/\lambda$, vnf creation time distribution is exponential with mean $1/\mu$ and there are N vnfs in a network service.

We consider the states as the number of vnfs which are created corresponding to a network service. Initially the system is in state 0, i.e., the virtual network function is not created. Once the request of network service creation is generated, all the vnfs are created independently with rate μ. So with rate $N\mu$ the system state changes from state 0 to state 1. After creation of one vnf, with rate $(N-1)\mu$ the system state changes from state 1 to state 2. In this way the system state changes in the forward direction. With rate μ the system state $(N-1)$ changes into state N. During the creation of vnfs, there can be a possibility of failure of compute node. If the failure happens in a compute node, then the vnf corresponding to the compute node goes down. As a result the respective vnf has to be recreated in other compute node, provided no vnf of this network service is already created in this compute node. So the transition from state N to state $(N-1)$ is possible with rate $N\lambda$. Similarly with rate $(N-1)\lambda$ the system state changes from state $(N-1)$ to $(N-2)$. Similarly with rate λ the system state changes from state 1 to state 0. The transition rate diagram for this network service deployment is present in Fig. 3.

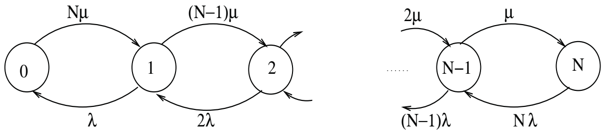

Fig. 3: Transition rate diagram of network service availability, considering deployment of all vnfs of a network service in different compute nodes

We can write the flow balance equations from this transition rate diagram as below.

$$\begin{aligned} N\mu\pi_0 &= \lambda\pi_1 \\ (N-1)\mu\pi_1 &= 2\lambda\pi_2 \\ (N-2)\mu\pi_1 &= 3\lambda\pi_3 \\ &\cdots \\ 2\mu\pi_{N-2} &= (N-1)\lambda\pi_{N-1} \\ \mu\pi_{N-1} &= N\lambda\pi_N \end{aligned} \tag{6}$$

After simplification of previous equations we can write,

$$\pi_n = \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{i\lambda} \right) \pi_0 \tag{7}$$

Since, $\pi_0 + \pi_1 + \cdots + \pi_{N-1} + \pi_N = 1$, we can write after simplification as in equation (8).

$$\pi_0 = \left[1 + \sum_{n=1}^{N} \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{i\lambda} \right) \right]^{-1} \tag{8}$$

So the network service availability, A_{v2}^{NS} $(= \pi_N)$ can be given by (9).

$$A_{v2}^{NS} = \left(\prod_{i=1}^{N} \frac{(N-i+1)\mu}{i\lambda} \right) \pi_0 \tag{9}$$

The network service availability, A_{v2}^{NS} can be further simplified as in equation (10).

$$A_{v2}^{NS} = \frac{\prod_{i=1}^{N} \frac{(N-i+1)\mu}{i\lambda}}{1 + \sum_{n=1}^{N} \left(\prod_{i=1}^{n} \frac{(N-i+1)\mu}{i\lambda} \right)} \tag{10}$$

IV. Results and discussions

In this section, we describe the analytical results which are generated to compare the availability of network service considering the deployment of all the vnfs of a network service in single compute node and in different compute nodes. The default system parameters are specified in Table II.

TABLE II: Default value of system parameters

Failure rate of compute node (λ)	0.00014 (/minute)
Rate of creation of vnf (μ)	1 (/minute)
Number of vnf in a network service (N)	3

A. Availability comparison with respect to failure rate of compute node

In this case, we generate availability of network service corresponding to the failure rate of compute node (λ), considering rate of creation of virtual network function, μ as 1 vnf per minute in Fig. 4. The availability of network service considering deployment in single compute node is higher than the availability of network service considering deployment in multiple compute nodes. Since with the use of multiple compute nodes, the failure rate increases more than the use of single compute node, the availability of network service is more with the use of single compute node than the availability of network service with the use of multiple compute nodes. If we increase the failure rate of compute node, then the system spends more time for creation of virtual network function, so the availability of network service decreases with the increase of failure rate of compute node.

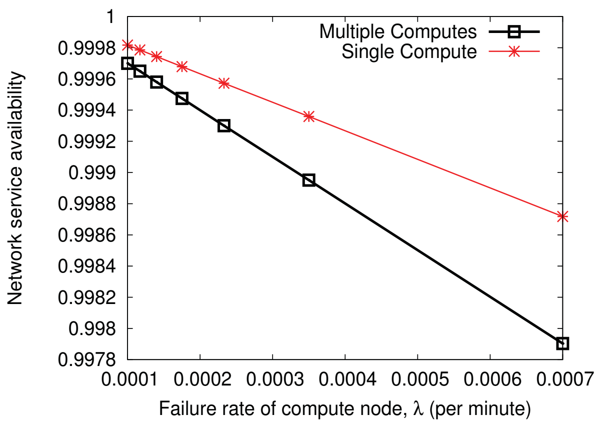

Fig. 4: Availability of network service with respect to failure rate of compute node per minute (λ) where $\mu = 1$ vnf per minute and $N = 3$ vnfs per network service

B. Availability comparison with respect to virtual network function creation rate

We generate availability of network service with respect to virtual function creation rate as in Fig. 5. To generate availability of network service, we consider failure rate of compute node, λ as 0.00014 per minute. Here with the increase of vnf creation rate, the availability increases, since the vnf takes less time to come up with the increase of vnf creation rate. Network service availability considering single compute node is higher than the availability of network service considering multiple compute nodes with single vnf in each compute node corresponding to a network service. With more number of compute nodes, the failure rate of compute node increases, as a result the availability of network service with more number of compute nodes is lower than the availability of network service with single compute node.

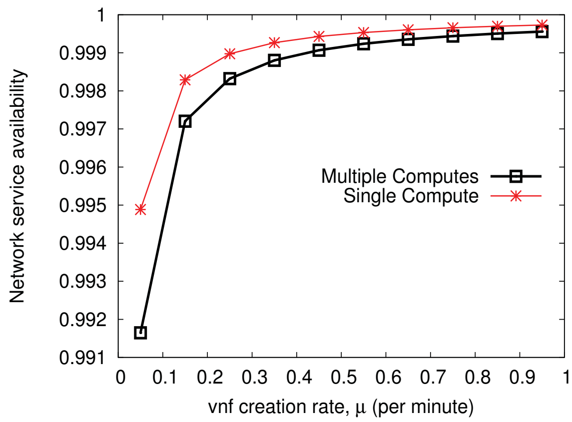

Fig. 5: Availability of network service with respect to vnf creation rate per minute (μ) where $\lambda = 0.00014$ compute node failures per minute and $N = 3$ vnfs per network service

C. Availability comparison with respect to number of vnfs per network service

We generate availability of network service with respect to number of virtual network functions of a network service as in Fig. 6. With the increase of number of vnfs per network service, the availability of network service decreases. In case of multiple compute nodes, with the increase of vnfs, the failure rate increases, so availability decreases. However, in case of single compute node, failure rate of compute node does not increase like multiple compute nodes. So the availability of network service with single compute node is higher than the availability of network service with multiple compute nodes. Since with more number of vnfs, network service takes more time to come up due to failure of vnfs, the availability of network service decreases with the increase of number of vnfs of network service in all cases.

V. Conclusion

We have proposed closed form expressions to find the availability of network service considering failure of compute node(s). We have considered the network service as a set of vnfs. These vnfs can be deployed in single compute node or in different compute nodes. Considering these extreme cases of vnf deployment

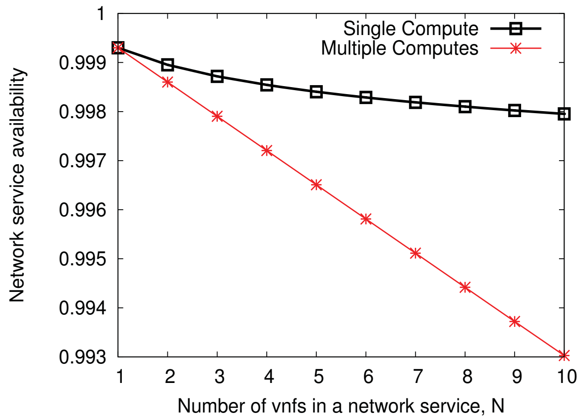

Fig. 6: Availability of network service with respect to number of vnfs per network service and $\lambda = 0.0007$ per minute and $\mu = 1$ per minute

strategies, we have derived closed form expression to find the availability of network service. We have compared the network service availability with respect to failure rate of compute node, creation rate of virtual network function and number of virtual network functions in a network service. In all cases of comparison, we have seen the network service availability considering single compute node based deployment is higher than the network service availability considering deployment based on multiple compute nodes.

References

[1] *OpenStack Documentation*, 2020, https://docs.openstack.org.

[2] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications: Second edition*. John Wiley and Sons, 2006.

[3] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for nfv," *IEEE Trans. on Services Computing*, pp. 1–14, 2019.

[4] T. Taleb, A. Ksentini, and B. Sericola, "On service resilience in cloud-native 5g mobile systems," *IEEE Jnl. on Selected Areas in Comm.*, vol. 34, no. 3, pp. 483–496, 2016.

[5] X. Song, X. Zhang, S. Yu, S. Jiao, and Z. Xu, "Resource-efficient virtual network function placement in operator networks," in *IEEE Global Communications Conf.*, 2017, pp. 1–7.

[6] J. Gil Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Trans. on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[7] P. Mandal and R. Jain, "Comparison of openstack orchestration procedures," in *Intl. Conf. on Smart Applications, Communications and Networking*, 2019, pp. 1–4.

[8] P. Mandal, R. Jain, and V. Surwade, "Performance analysis of openstack controller," in *Intl. Conf. on Smart Applications, Communications and Networking*, 2019, pp. 1–5.

[9] P. Mandal, R. Jain, N. K. Meena, and V. Kumar, "Availability comparison of 5g virtual network function," in *IEEE 3rd 5GWF*, 2020, pp. 31–35.