



Availability and Reliability of Service Function Chain: A Quantitative Evaluation View

Yaru Li¹ · Lin Li¹ · Jing Bai¹ · Xiaolin Chang¹ · Yingying Yao¹ · Peide Liu^{2,3}

Received: 23 November 2022 / Accepted: 6 March 2023
© The Author(s) 2023

Abstract

Service function chain (SFC) based on network function virtualization (NFV) technology can handle network traffic flexibly and efficiently. The virtual network function (VNF), as the core function unit of SFC, can experience software aging, which reduces the availability and reliability of SFC and even leads to service interruption, after it runs continuously for a long time. Software rejuvenation technique can effectively combat software aging. However, its effectiveness in improving the availability and reliability of SFC needs to be evaluated. Compared with existing models, this paper proposes a semi-Markov model to capture the behaviors of each VNF in a SFC from the occurrence of software aging to recovery by software rejuvenation technique under the condition that the failure times and recovery times follow general distribution, while considering trigger intervals of software rejuvenation technique. We then derive the calculation formulas of the steady-state availability, transient availability, and reliability, which are applied to evaluate the effectiveness of software rejuvenation technique. Finally, we conduct sensitivity analysis and numerical experiments to analyze the effects of system parameters, the number of VNFs and trigger interval of software rejuvenation technique on availability and reliability of SFC, and the effects of time-varying parameters on transient availability of SFC.

Keywords Availability · Reliability · Service function chain · Software aging · Software rejuvenation

Abbreviations

| | |
|-------|-------------------------------------|
| EDTMC | Embedded discrete time Markov chain |
| MTTF | Mean time to failure |
| NFV | Network function virtualization |
| SFC | Service function chain |
| SRN | Stochastic reward nets |
| VNF | Virtual network function |

✉ Lin Li
lilin@bjtu.edu.cn

Yaru Li
20120475@bjtu.edu.cn

Jing Bai
19112047@bjtu.edu.cn

Xiaolin Chang
xlchang@bjtu.edu.cn

Yingying Yao
yyyao@bjtu.edu.cn

Peide Liu
liupd@sdufe.edu.cn

1 Introduction

With the continuous emergence of use cases, such as augmented reality (AR), virtual reality (VR), and autonomous driving, network traffic is growing explosively. According to GlobeNewswire's prediction, the global mobile data traffic will reach 211EB per month by 2026 [1]. However, in traditional networks, network functions are implemented on dedicated hardware devices, resulting in a series of problems, such as high cost and poor scalability. Network function virtualization (NFV) technology is an effective way to solve the above problems. It decouples network functions from hardware devices, and network traffic reaches users according to user needs through service function chain (SFC) composed

¹ Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China

² School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, Shandong, China

³ Shandong Key Laboratory of Blockchain Finance, Shandong University of Finance and Economics, Jinan, China

of multiple virtual network functions (VNFs) in a specific order. For example, the 5G network is cut into a number of virtual end-to-end network slices to support new use cases and diversified services required by multidimensional performance [2]. Each slice provides users with specific services through SFC [3], as shown in Fig. 1. At present, Huawei [4], Cisco [5], and other equipment manufacturers also support SFC technology.

VNF is the core function unit of SFC [6]. However, after a long period of continuous operation of VNF, phenomenon of software aging will occur, leading to the destruction of the availability and reliability of SFC and even causing service interruption [7]. In the past 3 decades, outage events caused by software aging have occurred frequently. For example, the Patriot missile defense system failed to intercept enemy missiles due to software aging caused by the continuous accumulation of rounding errors, resulting in significant losses [8].

Software rejuvenation technique can effectively combat software aging before service interruption, thus achieving the goal of high availability and high reliability of SFC [9]. However, the effectiveness of software rejuvenation technique in improving the availability and reliability of SFC needs to be evaluated. The quantitative analysis method based on analytical model is an effective method to evaluate the service availability and reliability in virtualization systems. In recent years, research teams have developed various models. Some studies assumed that the occurrence times of all events followed exponential distribution, and some studies ignored the trigger intervals of software rejuvenation technique. In addition, few studies comprehensively

evaluated the effectiveness of software rejuvenation technique from three aspects: steady-state availability, transient availability, and reliability. Therefore, when applying analytical model-based methods to evaluate the availability and reliability of SFC, there are still many problems to be solved, as follows:

- In a SFC, each VNF has different resource requirements, resulting in different occurrence times of their abnormal events and recovery events. Therefore, how to capture the differences of VNF behaviors is a problem to be solved.
- The occurrence time of event does not necessarily follow the exponential distribution in the actual system. For example, since the failure rate of components increases with time in practice, the occurrence times of failure events should follow the distribution function with increasing failure rate, such as hypoexponential distribution. Therefore, how to accurately capture the occurrence time of each event in the actual system is a problem to be solved.
- The trigger intervals of software rejuvenation technique can affect its effectiveness. In addition, triggering software rejuvenation technique at intervals corresponding to the optimal availability of SFC does not necessarily achieve the optimal reliability. Therefore, how to analyze the impact of trigger intervals of software rejuvenation technique on each metric is a problem to be solved.
- In many key use cases, such as autonomous driving, transient analysis is more important than steady-state analysis. Therefore, how to evaluate the effectiveness of software rejuvenation technique from both transient and steady-state aspects is a problem to be solved.

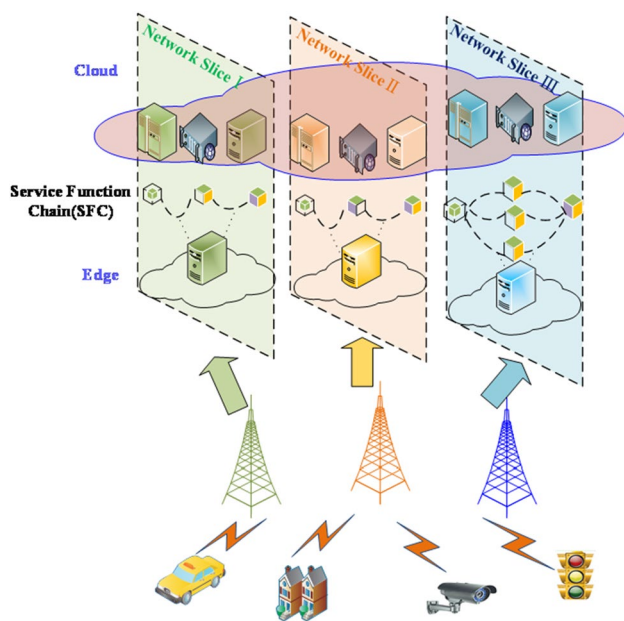


Fig. 1 Network slicing

To overcome the limitations of the above model, we propose a semi-Markov model, which describes the behaviors of each VNF in a SFC from suffering from software aging to recovery by software rejuvenation technique. As far as we know, it is the first time to apply a semi-Markov model to quantitatively evaluate the effectiveness of software rejuvenation technique from steady-state availability, transient availability, and reliability of SFC, taking into account trigger intervals of software rejuvenation technique. The SMP model can avoid the loss of accuracy inherent caused by the inability of non-state-space models to capture the time dependencies between abnormal and recovery behaviors. In addition, compared with the continuous-time Markov chain, it can relax the assumption that the occurrence times of all events follow exponential distribution, to capture SFC system behaviors more accurately. The main contributions of this paper are summarized as follows:

- We propose an effective semi-Markov model to quantitatively evaluate the effectiveness of software rejuvena-

tion technique, which describes the behaviors of the SFC system deploying software rejuvenation technique from suffering from software aging to recovery. In addition, our model can capture the time dependence between various behaviors.

- We derive the calculation formulas of steady-state availability, transient availability and reliability of the SFC composed of any number of VNFs. These formulas can prevent the problem of state-space explosion caused by the increase of the number of VNFs in a SFC.
- We carry out simulation experiments to verify the correctness of the method proposed in this paper. Sensitivity analysis and numerical analysis experiments are also carried out to quantitatively evaluate the impact of system parameters on various metrics. The numerical results provide a theoretical basis for operators to design and deploy SFC elastically.

The rest of this paper is arranged as follows. Section 2 introduces related work. Section 3 introduces the system description, the semi-Markov model proposed in this paper, and the process of calculating availability and reliability. Section 4 introduces the results of numerical experiments. Section 5 presents the conclusions and future work.

2 Related Work

Analytical model-based methods and simulation are two types of model-based quantitative evaluation methods, which can be cross-verified to make the evaluation results more accurate [20]. The goal of this paper is to evaluate the availability and reliability of SFC using the analytical model-based method. Therefore, this section focuses on the studies which applied the analytical model-based method to analyze the availability and reliability of service in the virtualization system.

Analytical models for evaluating availability and reliability can be divided into the following three categories: non-state-space models, such as reliability block diagrams (RBD); state-space models, such as Markov model and semi-Markov model; multi-level models, namely the combination of non-state-space model and state-space model [20].

Fan et al. [10] estimated the SFC availability utilizing a RBD model, in which primary VNFs in the SFC were protected by backup VNFs. Wang et al. [11] applied a RBD model to analyze the availability of SFC executed in parallel, in which the working SFCs used multiple backup SFCs for protection. These non-state-space models did not allow to depict the time dependence of abnormal behaviors (namely software aging and failure behaviors) and recovery behaviors.

Nguyen et al. [12] studied the availability of virtualized server system deploying virtual machine real-time migration technology and failover technology by constructing stochastic reward nets (SRNs) model. Zhu et al. [13] considered a virtualization system where both virtual machines and virtual machine monitors can fail, and analyzed the availability and reliability of the system. Machida et al. [14] proposed a semi-Markov model to study the effectiveness of different software recovery strategies in improving availability. The authors in [15–17] evaluated the availability and reliability of the vehicle platooning service, the SFC consisting of serial VNFs and the SFC consisting of serial and parallel VNFs based on semi-Markov models, respectively.

Mauro et al. [18] described the dependence between VNFs in a SFC by constructing a RBD model, and captured the behaviors of a single VNF by constructing SRN models. In addition, the authors in [19] used a RBD model and SRN model to evaluate the availability of IP multimedia subsystem (IMS).

The differences between our work and the existing studies are as follows:

- The studies [10] and [11] failed to capture the interaction between abnormal behaviors and recovery behaviors of components. The studies [12] and [13] assumed that the occurrence times of all events followed exponential distribution. The studies [10–19] ignored the impact of trigger intervals of software rejuvenation technique on service reliability and availability. The models developed in studies [12–14] were not applicable to evaluating the availability and reliability of services composed of multiple sub-services. Different from these models, our model allows to analyze the time dependence between abnormal behaviors and recovery behaviors of each VNF, as well as the time dependence between various behaviors of VNFs in a SFC, when the occurrence times of failure events and recovery events follow any type of distribution (namely, general distribution). In addition, it can also capture the behaviors of system, considering trigger intervals of software rejuvenation technique.
- The studies [10–14, 17, 18] only analyzed one or two of the three metrics of steady-state availability, transient availability, and reliability of service. Different from these studies, this paper derives formulas for calculating steady-state availability, transient availability and reliability to analyze the effectiveness of software rejuvenation technique in multiple dimensions.
- The studies [11–14] did not carry out simulation experiments. The studies [10, 11, 13–15] and [17] did not carry out sensitivity analysis experiments. Different from these studies, this paper verifies the correctness of the model and formulas proposed in this paper by performing simulation experiments. In addition, by conducting sensitivity

analysis experiments, we identify bottlenecks that restrict the improvement of the effectiveness of software rejuvenation technique, laying the foundation for optimizing the availability and reliability of SFC.

Table 1 shows the comparison between our work and the aforementioned studies.

3 System Description and Model

This section first introduces the system illustrated in this paper. Then, the semi-Markov model constructed in this paper is introduced. Finally, the calculation formulas of steady-state availability, transient availability, and reliability metrics are given.

3.1 System Description

Figure 2 shows an example of the SFC system architecture studied in this paper. There are multiple hosts in the system, and each host operating system runs multiple containers. The active containers execute active VNFs in a SFC and backup containers execute backup VNFs which are used to support failover technology. We assume that the backup resources running on each host are sufficient, so there is an available backup VNF at any time. User requests are processed sequentially by VNFs in a SFC. After running for a period of time, these VNFs can suffer from software aging and failure caused by software aging. The details are as follows: if an active VNF is detected to suffer from software aging, a healthy backup VNF will be selected to support failover technology. After a certain interval, the failover technology is triggered, that is, the backup container takes over the request being processed. During failover or the

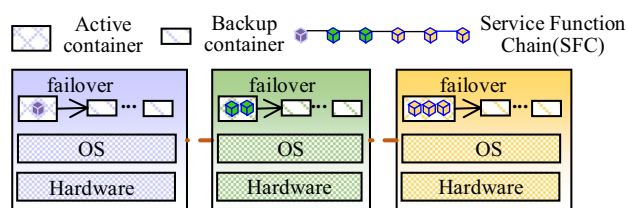


Fig. 2 System architecture

trigger interval of failover technology, if other VNFs are detected to suffer from software aging, all VNFs in the system will be restarted. If a VNF is detected to fail, this component will be repaired. After it is repaired, all hosts in the system will be rebooted.

In addition, we assume that the occurrence times of failure events and recovery events in the system follow general distribution, and the occurrence times of software aging events follow exponential distribution.

3.2 Semi-Markov Model

Define an n -tuple index $(i_{S1}, i_{S2}, i_{S3}, \dots, i_{Sn})$ to represent the system state, where i_{Sn} represents the state of the n th VNF. Each VNF may have five states: healthy (H), software aging (A), failure (F), failover (L), and restart (R). Each state is defined as follows:

- Healthy (H): in this state, all VNFs can run efficiently.
- Software aging (A): in this state, software aging occurs, and the rate of executing requests slows down.
- Failure: in this state, VNF fails due to software aging.
- Failover: in this state, failover technology will be triggered.
- Restart: In this state, VNF will be restarted.

Table 1 Comparison of existing models discussed in Sect. 2

| No | Trigger interval of software rejuvenation technique | Model | Distribution | | Metric | | | Sensitivity analysis | Simulation |
|-----------|---|-----------------|----------------------|--------------------------|------------------------|---------------------------|-------------|----------------------|------------|
| | | | General distribution | Exponential distribution | Transient availability | Steady-state availability | Reliability | | |
| [10] 2017 | × | Non-state-space | × | × | × | √ | × | × | √ |
| [11] 2021 | × | Non-state-space | × | × | × | √ | × | × | × |
| [12] 2018 | × | State-space | × | √ | × | √ | × | √ | × |
| [13] 2020 | × | State-space | × | √ | √ | √ | √ | × | × |
| [14] 2017 | × | State-space | √ | √ | × | √ | × | × | × |
| [15] 2021 | × | State-space | √ | √ | √ | √ | √ | × | √ |
| [16] 2022 | × | State-space | √ | √ | √ | √ | √ | √ | √ |
| [17] 2022 | √ | State-space | √ | √ | × | √ | √ | × | √ |
| [18] 2017 | × | Multi-level | × | √ | × | √ | × | √ | √ |
| [19] 2018 | √ | Multi-level | √ | √ | √ | √ | √ | √ | √ |

There are 5^n system states, among which the number of meaningless states is $5^n - 2n - 3$. For example, because VNFs in a SFC are connected together in a sequential order, the request processing stops when a VNF fails. Therefore, the state $(F_{S1}, H_{S2}, H_{S3}, \dots, H_{Sn})$ is meaningless.

Based on the aforementioned analysis, the semi-Markov model can be used to capture the behaviors of each VNF in a SFC from the occurrence of software aging to the recovery using software rejuvenation technique. The state sequence of this random process at transition occurrence time points forms an embedded discrete time Markov chain (EDTMC). The occurrence times of failure and recovery events follow general distribution. Figure 3 shows an example of the semi-Markov model, which is used to describe the behaviors of a SFC consisting of 6 VNFs. The definition of variables used in the figure is shown in Table 2. In this model, the SFC system starts with state (H_{S1}, \dots, H_{S6}) . After a period of operation, VNFs in the system can suffer from software aging. If the 1st VNF suffers from software aging, the system will enter state (D_{S1}, \dots, H_{S6}) . When the system stays at state (D_{S1}, \dots, H_{S6}) , if the 1st VNF fails, the system will enter state (F_{S1}, \dots, F_{S6}) , if one of other VNFs suffers from software aging, the system will enter state (R_{S1}, \dots, R_{S6}) , and if failover technology is triggered after a certain interval, the system will enter state (L_{S1}, \dots, H_{S6}) . When

the system stays at state (L_{S1}, \dots, H_{S6}) , if the 1st VNF fails, the system will enter state (F_{S1}, \dots, F_{S6}) , if one of the other VNFs suffers from software aging, the system will enter state (R_{S1}, \dots, R_{S6}) , and if the backup container takes over requests, the system will enter state (H_{S1}, \dots, H_{S6}) . When the system stays at state (R_{S1}, \dots, R_{S6}) , the system enters state (H_{S1}, \dots, H_{S6}) after restarting all VNFs. From state (F_{S1}, \dots, F_{S6}) , the system returns back to state (H_{S1}, \dots, H_{S6}) after repairing the failed VNF and rebooting all VNFs. The subsequent state transitions after other VNFs suffer from software aging are similar to that after the 1st VNF suffers from software aging.

3.3 Transient Availability Analysis

The transient availability $\pi_{\text{availability}}(t)$ of the SFC composed of n VNFs can be calculated by solving the probability that the system is unavailable at time t , which is shown in Eq. (1):

$$\begin{aligned} \pi_{\text{availability}}(t) &= 1 - \pi_{S_{(2n+1)}}(t) - \pi_{S_{(2n+2)}}(t) \\ &= 1 - \sum_{i=0}^{2n+2} \pi_{S_i}(0)(V_{S_i S_{(2n+1)}}(t) + V_{S_i S_{(2n+2)}}(t)), \end{aligned} \tag{1}$$

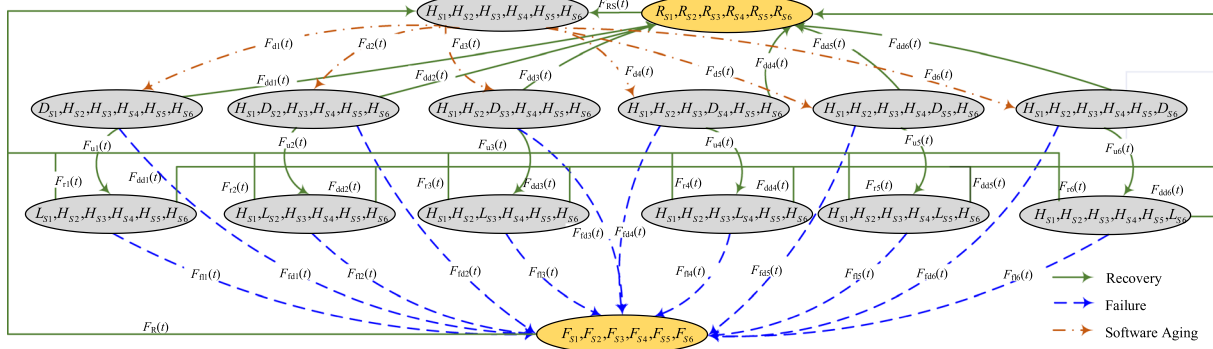


Fig. 3 Semi-Markov model

Table 2 Variable definition

| Variable | Definition | Distribution | Default |
|------------------|--|--------------------------|-----------|
| T_{di} | The cumulative distribution function $F_{di}(t)$ of the time it takes for the i th VNF from state H to D | Exponential distribution | 10–3600 h |
| T_{fdi}/T_{fi} | The cumulative distribution function $F_{fdi}(t)/F_{fi}(t)$ of the time it takes for the i th VNF from state D/L to F | General distribution | 5–1200 h |
| T_{ri} | The cumulative distribution function $F_{ri}(t)$ of the time it takes for the i th VNF from state L to H | General distribution | 5–12 s |
| T_{ui} | The cumulative distribution function $F_{ui}(t) = u(t - a_i)$ of the time it takes for the i th VNF from state D to L | Unit step function | 0–1200 h |
| T_R | The cumulative distribution function $F_R(t)$ of the time of repairing the SFC system | General distribution | 0.8–1.2 h |
| T_{RS} | The cumulative distribution function $F_{RS}(t)$ of the time of restarting all VNFs | General distribution | 10–20 s |
| T_{ddi} | The cumulative distribution function $F_{ddi}(t)$ of the minimum time it takes for other VNFs from state H to D after the i th VNF suffers from software aging | Exponential distribution | – |

where $\pi_{S_i}(0)$ ($0 \leq i \leq 2n+2$) is the initial state probability and $V_{S_i S_j}(t)$ ($0 \leq i, j \leq 2n+2$) is the non-zero element in the transient solution matrix of conditional transition probability $\mathbf{V}_S(t)$.

The calculation process of $\mathbf{V}_S(t)$ is shown in Eq. (2):

$$\mathbf{V}_S^{\sim}(s) = \mathbf{E}_S^{\sim}(s) + \mathbf{K}_S^{\sim}(s)\mathbf{V}_S^{\sim}(s), \quad (2)$$

where $\mathbf{V}_S^{\sim}(s)$, $\mathbf{K}_S^{\sim}(s)$ and $\mathbf{E}_S^{\sim}(s)$ are Laplace–Stieltjes transform of $\mathbf{V}_S(t)$, kernel matrix $\mathbf{K}_S(t)$ and diagonal matrix $\mathbf{E}_S(t)$, respectively [20]. The non-zero element $k_{S_i S_j}(t)$ ($0 \leq i, j \leq 2n+2$) in the kernel matrix can be solved by Eqs. (3)–(11), and the non-zero element $E_{S_i S_i}(t)$ ($0 \leq i \leq 2n+2$) in the diagonal matrix can be solved by Eq. (12) [20]:

$$k_{S_0 S_i}(t) = \int_0^t \prod_{r \in B_i} (1 - F_{dr}(t)) dF_{di}(t), \quad (3)$$

$$k_{S_i S_{(n+i)}}(t) = F_{ui}(t) \left(1 - \int_0^{a_i} (1 - F_{ddi}(t)) dF_{fdi}(t) - \int_0^{a_i} (1 - F_{fdi}(t)) dF_{ddi}(t) \right), \quad (4)$$

$$k_{S_i S_{(2n+1)}}(t) = \int_0^t (1 - F_{ui}(t)) (1 - F_{ddi}(t)) dF_{fdi}(t), \quad (5)$$

$$k_{S_i S_{(2n+2)}}(t) = \int_0^t (1 - F_{ui}(t)) (1 - F_{fdi}(t)) dF_{ddi}(t), \quad (6)$$

$$k_{S_{(i+n)} S_0}(t) = \int_0^t (1 - F_{fi}(t)) (1 - F_{ddi}(t)) dF_{ri}(t), \quad (7)$$

$$k_{S_{(i+n)} S_{(2n+1)}}(t) = \int_0^t (1 - F_{ri}(t)) (1 - F_{ddi}(t)) dF_{fi}(t), \quad (8)$$

$$k_{S_{(i+n)} S_{(2n+2)}}(t) = \int_0^t (1 - F_{ri}(t)) (1 - F_{fi}(t)) dF_{ddi}(t), \quad (9)$$

$$k_{S_{(2n+1)} S_0}(t) = F_R(t), \quad (10)$$

$$k_{S_{(2n+2)} S_0}(t) = F_{RS}(t), \quad (11)$$

$$E_{S_i S_i}(t) = 1 - \sum_{j=0}^{2n+2} k_{S_i S_j}(t). \quad (12)$$

Therefore, at time t , the probabilities of the system in the unavailable states can be solved by Eqs. (13)–(15):

$$\pi_{S_{(2n+1)}}(t) = L^{-1} \left(-E_{S_{(2n+1)} S_{(2n+1)}}^{\sim}(t) \left(\sum_{i=1}^n (k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(i+n)}}^{\sim}(t) k_{S_{(i+n)} S_{(2n+1)}}^{\sim}(t) + k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(2n+1)}}^{\sim}(t)) \right) \right) / A, \quad (13)$$

$$\pi_{S_{(2n+2)}}(t) = L^{-1} \left(-E_{S_{(2n+2)} S_{(2n+2)}}^{\sim}(t) \left(\sum_{i=1}^n (k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(i+n)}}^{\sim}(t) k_{S_{(i+n)} S_{(2n+2)}}^{\sim}(t) + k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(2n+2)}}^{\sim}(t)) \right) \right) / A, \quad (14)$$

$$A = \sum_{i=1}^n (k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(i+n)}}^{\sim}(t) k_{S_{(i+n)} S_0}^{\sim}(t) + k_{S_0 S_i}^{\sim}(t) k_{S_i S_{(i+n)}}^{\sim}(t) k_{S_{(i+n)} S_{(2n+2)}}^{\sim}(t) + k_{S_{(i+n)} S_{(2n+1)}}^{\sim}(t) k_{S_{(2n+1)} S_0}^{\sim}(t) + k_{S_{(i+n)} S_{(2n+1)}}^{\sim}(t) k_{S_{(i+n)} S_{(2n+2)}}^{\sim}(t) + k_{S_{(2n+2)} S_0}^{\sim}(t) + k_{S_{(2n+2)} S_0}^{\sim}(t) k_{S_i S_{(2n+1)}}^{\sim}(t) k_{S_{(2n+1)} S_0}^{\sim}(t) + k_{S_{(2n+2)} S_0}^{\sim}(t) + k_{S_{(2n+2)} S_0}^{\sim}(t) k_{S_{(2n+2)} S_0}^{\sim}(t)). \quad (15)$$

3.4 Steady-State Availability Analysis

The steady-state availability $\pi_{\text{availability}}$ of the SFC composed of n VNFs can be calculated by solving the steady-state probabilities of the system in the unavailable states, which is shown in Eq. (16):

$$\pi_{\text{availability}} = 1 - \pi_{S_{(2n+1)}} - \pi_{S_{(2n+2)}} = 1 - (V_{S_{(2n+1)}} h_{S_{(2n+1)}} + V_{S_{(2n+2)}} h_{S_{(2n+2)}}) / \left(\sum_{i=0}^{2n+2} V_{S_i} h_{S_i} \right), \quad (16)$$

where h_{S_i} is the sojourn time of the system in state S_i ($0 \leq i \leq 2n+2$), which can be solved by Eqs. (17)–(21) [20].

$$h_{S_0} = \int_0^{\infty} \prod_{r \in B} (1 - F_{dr}(t)) dt, \quad (17)$$

$$h_{S_i} = \int_0^{a_i} (1 - F_{fdi}(t)) (1 - F_{ddi}(t)) dt, \quad (18)$$

$$h_{S_{(i+n)}} = \int_0^{\infty} (1 - F_{fi}(t)) (1 - F_{ddi}(t)) (1 - F_{ri}(t)) dt, \quad (19)$$

$$h_{S_{(2n+1)}} = \int_0^{\infty} (1 - F_R(t)) dt, \quad (20)$$

$$h_{S_{(2n+2)}} = \int_0^{\infty} (1 - F_{RS}(t)) dt. \quad (21)$$

V_{S_i} is the steady-state probability of the EDTMC for system state S_i ($0 \leq i \leq 2n+2$). The calculation process of V_{S_i} is shown in Eqs. (22)–(27):

$$V_{S_0} = 1/M, \quad (22)$$

$$V_{S_i} = p_{S_0 S_i} / M, \tag{23}$$

$$V_{S_{(i+n)}} = p_{S_0 S_i} p_{S_i S_{(i+n)}} / M, \tag{24}$$

$$V_{S_{(2n+1)}} = \sum_{i=1}^n (p_{S_0 S_i} p_{S_i S_{(i+n)}} p_{S_{(i+n)} S_{(2n+1)}} + p_{S_0 S_i} p_{S_i S_{(2n+1)}}) / M, \tag{25}$$

$$V_{S_{(2n+2)}} = \sum_{i=1}^n (p_{S_0 S_i} p_{S_i S_{(i+n)}} p_{S_{(i+n)} S_{(2n+2)}} + p_{S_0 S_i} p_{S_i S_{(2n+2)}}) / M, \tag{26}$$

$$M = 1 + \sum_{i=1}^n (p_{S_0 S_i} p_{S_i S_{(i+n)}} p_{S_{(i+n)} S_{(2n+1)}} + p_{S_0 S_i} p_{S_i S_{(2n+1)}} + p_{S_0 S_i} p_{S_i S_{(i+n)}} p_{S_{(i+n)} S_{(2n+2)}} + p_{S_0 S_i} p_{S_i S_{(2n+2)}} + p_{S_0 S_i} + p_{S_0 S_i} p_{S_i S_{(i+n)}}), \tag{27}$$

where $p_{S_i S_j}$ ($0 \leq i, j \leq 2n+2$) is the non-zero element in the one-step transition probability matrix \mathbf{P}_S , which can be obtained by solving $\mathbf{P}_S = \lim_{t \rightarrow \infty} \mathbf{K}_S(t)$ [20].

3.5 Reliability Analysis

The mean time to failure (MTTF) is one of the metrics widely used in evaluating reliability. The MTTF of the SFC composed of n VNFs can be calculated by solving the mean time of the system from the beginning to the failure [20], which is shown in Eq. (28):

$$\text{MTTF} = \sum_{i=0}^{2n} V_{S_{i^*}}^* h_{S_{i^*}}^*, \tag{28}$$

where $V_{S_{i^*}}^*$ is the expected number of accesses to state S_{i^*} ($0 \leq i^* \leq 2n$) before reaching the absorption state represented by the yellow state in Fig. 3 and $h_{S_{i^*}}^*$ is the sojourn time of the system in state S_{i^*} ($0 \leq i^* \leq 2n$). $V_{S_{i^*}}^*$ can be obtained by solving Eqs. (29)–(32), where $p_{S_{i^*} S_{j^*}}$ ($0 \leq i^*, j^* \leq 2n$) can be obtained by solving

$\mathbf{P}_S^* = \lim_{t \rightarrow \infty} \mathbf{K}_S^*(t)$ [20]. $h_{S_{i^*}}^*$ can be solved by Eqs. (29)–(32):

$$V_{S_0}^* = -1/W, \tag{29}$$

$$V_{S_{i^*}}^* = -p_{S_0 S_{i^*}} / W, \tag{30}$$

$$V_{S_{(i+n)}}^* = -p_{S_0 S_{i^*}} p_{S_{i^*} S_{(i+n)}} / W, \tag{31}$$

$$W = \sum_{i^*=1}^n (p_{S_0 S_{i^*}} p_{S_{i^*} S_{(i+n)}} p_{S_{(i+n)} S_0}) - 1. \tag{32}$$

4 Experimental Result

In this section, we first perform simulation to prove the approximate accuracy of our proposed model and derived formulas. Then we conduct sensitivity analysis experiments and numerical experiments to analyze the effects of system parameters, the number of VNFs and trigger interval of software rejuvenation technique on availability and reliability of SFC, and the effects of time-varying parameters on transient availability on SFC.

4.1 Experimental Configuration

Tables 2 and 3 show the default values of variables and the types of cumulative distribution functions that were used in the experiment, respectively. Note that some default values are set according to literature [21–23], and the use of other default values and cumulative distribution function types is only an example to prove the effectiveness of the model proposed in this paper. In this section, we use Maple software [23] to perform simulation experiments and numerical analysis experiments. Note that simulation and numerical analysis software can be implemented in any programming language.

Table 3 Type of cumulative distribution function and default value used in the experiment

| Cumulative distribution function | Distribution | Parameter | Default value |
|----------------------------------|--------------------|---|------------------------|
| $F_{di}(t)$ | Exponential | μ_i | 2.78e-4 ~ 0.1 |
| $F_{idi}(t)/F_{ni}(t)$ | Hypoexponential | $\theta_{di1}, \theta_{di1}/\theta_{i1}, \theta_{i1}$ | 1.2e-2, 8.96e-4 ~ 6e-2 |
| $F_{ri}(t)$ | Exponential | γ_{oi} | 300 ~ 720 |
| $F_{ui}(t)$ | Unit step function | a_i | 0 ~ 1200 |
| $F_R(t)$ | Exponential | β | 0.83 ~ 1.25 |
| $F_{RS}(t)$ | Exponential | σ | 180 ~ 360 |
| $F_{ddi}(t)$ | Exponential | $\sum_{j=1}^n \mu_j - \mu_i$ | / |

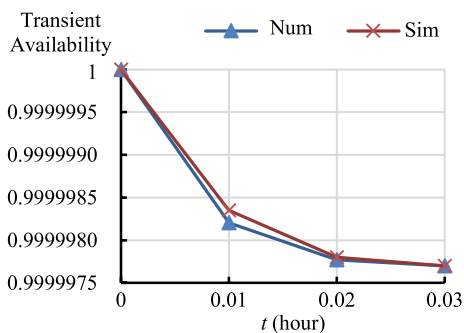


Fig. 4 Comparison of numerical and simulation results for transient availability

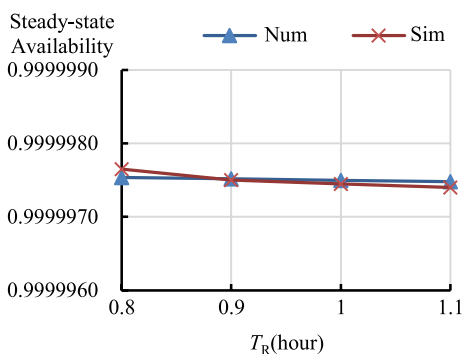


Fig. 5 Comparison of numerical and simulation results for steady-state availability

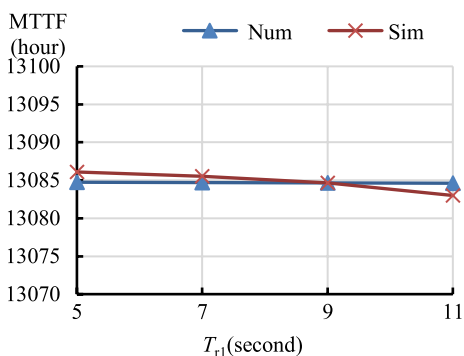


Fig. 6 Comparison of numerical results and simulation results for MTTF

4.2 Verification of Model and Formulas

Figures 4, 5, and 6 show the comparison of numerical results and simulation results of transient availability, steady-state availability and MTTF of SFC, respectively. ‘Num’ and ‘Sim’ in these figures represent numerical results and simulation results, respectively. From these figures, it can be observed that the difference between the numerical results and the related simulation results is very small, which

Table 4 Sensitivity of availability and MTTF

| Parameter | Sensitivity of steady-state availability | Sensitivity of MTTF |
|---------------|--|---------------------|
| μ_1 | $-7.67e-8$ | $-2.23e-1$ |
| θ_{11} | $-3.35e-8$ | $-5.46e-4$ |
| θ_{12} | $-3.34e-8$ | $-5.43e-4$ |
| γ_{o1} | $1.05e-12$ | $6.13e-6$ |
| β | $2.54e-6$ | 0 |
| σ | $3.05e-7$ | 0 |

proves the approximate accuracy of our proposed model and derived formulas.

4.3 Sensitivity Analysis

Table 4 shows the results of sensitivity analysis of steady-state availability and MTTF of SFC, with respect to system parameters. We observe:

- The steady-state availability and MTTF of SFC increase with the increase of the VNF aging time and failure time, and decrease with the increase of the failover time. The steady-state availability of SFC decreases with the increase of the system repair time and the time of restarting all VNFs. The MTTF of SFC is independent of the system repair time and the time of restarting all VNFs.
- Compared with other parameters, system repair time and the time of restarting all VNFs have the greatest impact on steady-state availability. The VNF aging time and failure time have the greatest impact on MTTF.

These experimental results can help service providers identify bottlenecks that affect the improvement of the availability and reliability of SFC.

4.4 Effect of Trigger Interval of Software Rejuvenation Technique on Steady-State Availability and MTTF

Figure 7 shows the numerical results of the steady-state availability of SFC under different trigger intervals of software rejuvenation technique (a_1) and system repair times (T_R). Figure 8 shows the numerical results of MTTF of SFC under different trigger intervals of software rejuvenation technique (a_1) and VNF failure times (T_{fd1}). It can be observed from Figs. 7 and 8 that with the increase of the trigger interval of software rejuvenation technique, the steady-state availability of SFC first increases and then decreases, and the MTTF of SFC decreases. This is because when the trigger interval of software rejuvenation

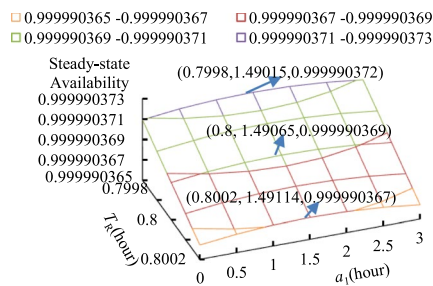


Fig. 7 Steady-state availability under different trigger intervals of software rejuvenation technique and system repair time

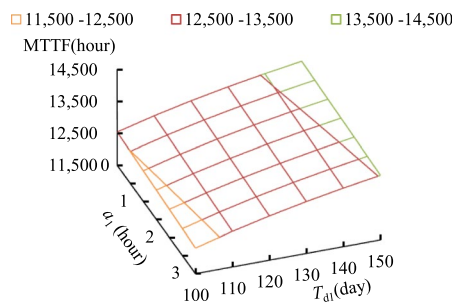


Fig. 8 MTTF under different trigger intervals of software rejuvenation technique and VNF failure time

Table 5 Steady-state availability, transient availability and MTTF under different number of VNFs

| Number of VNFs | Steady-state availability | Transient availability ($t=0.01$) | MTTF (h) |
|----------------|---------------------------|-------------------------------------|-----------|
| $n=4$ | 0.9999989729 | 0.99999999995280 | 43,083.26 |
| $n=5$ | 0.9999981959 | 0.99999999992086 | 22,187.87 |
| $n=6$ | 0.9999971571 | 0.99999999988058 | 13,084.72 |
| $n=7$ | 0.9999958547 | 0.999999934763408 | 8461.62 |

technique is less than the optimal value, the sojourn time of SFC in available states increases with the increase of trigger intervals of software rejuvenation technique. When the trigger interval of software rejuvenation technique is greater than the optimal value, the failure probability of VNF before triggering software rejuvenation technique increases with the increase of trigger interval. We can also observe the optimal trigger interval of software rejuvenation technique and the corresponding maximum steady-state availability and MTTF of SFC. For example, when T_R is 0.8 h, the maximum steady-state availability is 0.999990369, that is, the downtime allowed is about 5 min and 1.4 s per year, which is achieved at $a_1 = 1.49065$ h. In addition, as the system repair time increases, the optimal trigger interval of software rejuvenation technique corresponding to the maximum steady-state availability of SFC increases, and the corresponding maximum steady-state availability of SFC increases.

4.5 Effect of the Number of VNFs on Steady-State Availability, Transient Availability, and MTTF

Table 5 shows the numerical results of the steady-state availability, transient availability, and MTTF of SFC under different numbers of VNFs (n). It can be observed from Table 5 that as the number of VNFs increases, the steady-state availability, transient availability, and MTTF of SFC decrease. This is because the increase in the number of VNFs in a SFC leads to an increase in the number of components that may fail, thus increasing the time the system stays in the unavailable states.

4.6 Effect of Time-Varying Parameters on Transient Availability

Figure 9 shows the impact of time-varying parameters on transient availability of SFC. The gray line indicates that as the number of VNFs (n) increases from 6 to 7 in the 2nd hour, the transient availability decreases and then becomes stable. The yellow line indicates that when the VNF aging time (T_{d1}) increases from 10 to 100 h in the 2nd hour, the transient availability also increases. When the VNF aging time decreases to 50 h in the 4th hour, the transient availability decreases and then becomes stable. The blue

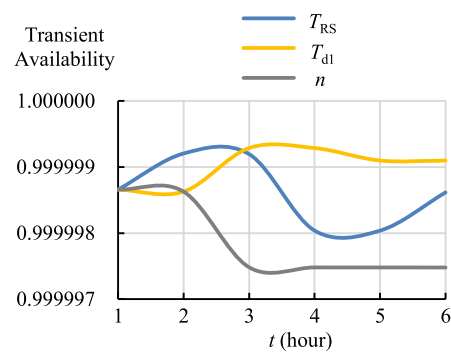


Fig. 9 Effect of time-varying parameters on transient availability

line indicates that when the time of restarting all VNFs (T_{RS}) decreases from 15 to 10 s in the 1st hour, the transient availability increases. When the time of restarting all

VNFs increases to 20 s in the 3rd hour, the transient availability decreases. When the time of restarting all VNFs decreases to 15 s in the 5th hour, the transient availability increases but decreases compared with the transient availability in the 1st hour.

5 Conclusions and Future Work

This paper proposes a semi-Markov model to quantitatively analyze the effectiveness of software rejuvenation technique on the steady-state availability, transient availability, and reliability of SFC. The sensitivity analysis results reveal that the system repair time and the time of restarting all VNFs have the greatest impact on the availability of SFC. The VNF aging time and failure time have the greatest impact on the reliability of SFC. The numerical experiment reveals that with the increase of the trigger interval of software rejuvenation technique, the availability of SFC first increases and then decreases, and the reliability of SFC decreases. As the number of VNFs increases, the availability and reliability of SFC decreases.

This paper assumes that the backup VNFs are sufficient. However, in practice, due to resource and cost constraints, the number of backup VNFs is limited, resulting in no backup VNFs available when triggering failover technology. Therefore, in the future, we will study the effect of the number of backup VNFs and their abnormal behaviors on the effectiveness of software rejuvenation technique.

Acknowledgements The authors are grateful to the editor and the anonymous reviewers for their constructive comments.

Author Contributions All the authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by the first three authors. The first draft of the manuscript was written by YL and all the authors commented on previous versions of the manuscript. All the authors read and approved the final manuscript.

Funding The work was supported in part by Beijing Municipal Natural Science Foundation under Grant no. M22037, China.

Availability of Data and Materials Data and material will be made available on reasonable request.

Declarations

Conflict of Interest The authors declare that they have no conflicts of interest.

Ethics Approval and Consent to Participate The authors confirm that this manuscript has not been submitted to any other journal for simultaneous consideration. This is an original work that has not been published elsewhere in any form or language.

Consent for Publication All the authors confirm that they all agreed with the content and gave explicit consent to submit the manuscript for consideration.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Qu, K., Zhuang, W., Ye, Q., Xuemin Shen, Xu., Li, J.R.: Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks. *IEEE Trans. Commun.* **68**(4), 2394–2408 (2020)
2. Mordor intelligence, network slicing market growth, trends, COVID-19 impact, and forecasts (2022–2027), [Online]. <https://www.mordorintelligence.com/industry-reports/network-slicing-market>
3. Mosahebfard, M., Vardakas, J.S., Verikoukis, C.V.: Modelling the admission ratio in NFV-based converged optical-wireless 5G networks. *IEEE Trans. Veh. Technol.* **70**(11), 12024–12038 (2021)
4. HUAWEI, Configuring a service chain, [Online]. <https://support.huawei.com/enterprise/en/doc/EDOC1100116573/cce80eb/configuring-a-service-chain>
5. Cisco, Implementing NSH Based Service Chaining, [Online]. https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-5/ip-addresses/configuration/guide/b-ip-addresses-cg-asr9000-65x/b-ip-addresses-cg-asr9000-65x_chapter_01111.html
6. Mohamad, A., Hassanein, H.S.: On demonstrating the gain of SFC placement with VNF sharing at the edge. *GLOBECOM* 1–6 (2019)
7. Matias, R., Andrzejak, A., Machida, F., Elias, D., Trivedi, K.S.: A systematic differential analysis for fast and robust detection of software aging. *SRDS* 311–320 (2014)
8. Grottko, M., Trivedi, K.S.: Aging, fast and slow. *Computer* **55**(5), 73–75 (2022)
9. Grottko, M., Li, L., Vaidyanathan, K., Trivedi, K.S.: Analysis of software aging in a web server. *IEEE Trans. Reliab.* **55**(3), 411–420 (2006)
10. Fan, J., Guan, C., Zhao, Y., Qiao, C.: Availability-aware mapping of service function chains. *INFOCOM* 1–9 (2017)
11. Wang, M., Cheng, Bo., Wang, S., Chen, J.: Availability- and Traffic-aware placement of parallelized SFC in data center networks. *IEEE Trans. Netw. Serv. Manag.* **18**(1), 182–194 (2021)
12. Nguyen, T.A., Min, D., Choi, E.: A comprehensive evaluation of availability and operational cost for a virtualized server system using stochastic reward nets. *J. Supercomput.* **74**(1), 222–276 (2018)
13. Zhu, H., Bai, J., Chang, X., Mistic, J.V., Mistic, V.B., Yang, Y.: Stochastic model-based quantitative analysis of edge UPF service dependability. *ICA3PP* **2**, 619–632 (2020)

14. Machida, F., Xiang, J., Tadano, K., Maeno, Y.: Lifetime extension of software execution subject to aging. *IEEE Trans. Reliab.* **66**(1), 123–134 (2017)
15. Bai, J., Chang, X., Trivedi, K.S., Han, Z.: Resilience-driven quantitative analysis of vehicle platooning service. *IEEE Trans. Veh. Technol.* **70**(6), 5378–5389 (2021)
16. Bai, J., Chang, X., Machida, F., Jiang, L., Han, Z., Trivedi, K.: Impact of service function aging on the dependability for MEC service function chain. *IEEE Trans. Depend. Secure Comput.* (Early Access) (2022). <https://doi.org/10.1109/TDSC.2022.3150782>
17. Bai, J., Chang, X., Machida, F., Han, Z., Yang, Xu., Trivedi, K.S.: Quantitative understanding serial-parallel hybrid SFC services: a dependability perspective. *Peer-to-Peer Netw. Appl.* **15**(4), 1923–1938 (2022)
18. Di Mauro, M., Longo, M., Postiglione, F., Carullo, G., Tambasco, M.: Service function chaining deployed in an NFV environment: an availability modeling. *CSCN* 42–47 (2017)
19. Di Mauro, M., Galatro, G., Longo, M., Postiglione, F., Tambasco, M.: IP multimedia subsystem in an NFV environment: availability evaluation and sensitivity analysis. *NFV-SDN* 1–6 (2018)
20. Trivedi, K.S., Bobbio, A.: *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge University Press, Cambridge (2017)
21. Tola, B., Nencioni, G., Helvik, B.E., Jiang, Y.: Modeling and evaluating NFV-enabled network services under different availability modes. *DRCN* 1–5 (2019)
22. Machida, F., Kim, D.S., Trivedi, K.S.: Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. *Perform. Eval.* **70**(3), 212–230 (2013)
23. Tola, B., Nencioni, G., Helvik, B.E.: Network-aware availability modeling of an end-to-end NFV-enabled service. *IEEE Trans. Netw. Serv. Manag.* **16**(4), 1389–1403 (2019)
24. Maplesoft, “Maple,” [Online]. <http://www.maplesoft.com/products/maple>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.