International Journal of Statistics and Systems
ISSN 0973-2675 Volume 21, Number 1 (2026), pp. 13-29

https://www.ripublication.com/ijss.htm

Semi-Markov Performance Analysis of Virtualised Computing System under Network Failures and Software Aging

Rimpaldeep Kaur[*a] **and R.K Bhardwaj**[b]

[a, b]*Department of Statistics,*
Punjabi University, Patiala,
Punjab, India.

Abstract

As virtualization technology becomes increasingly integral to modern computing environments, understanding the challenges it faces such as software aging and network failures, has never been more critical. This paper investigates the phenomenon of software aging in virtualization environments, focusing on a dual-host architecture comprising a main host and a backup host. As virtual machine monitor (VMM) rejuvenation is initiated on the main host, the backup host assumes the critical role of ensuring continuity by facilitating the migration of virtual machines (VMs). However, this process is susceptible to network failures that can significantly impact system performance and reliability. To model the system's behavior under these conditions, we employ the theory of semi-Markov regenerative processes, allowing for a comprehensive analysis of the effects of various types of network failures during VM migration. Our evaluation considers implications on service availability, performance degradation, and overall system robustness.

Keywords: System Performance ;Network failures ; Semi-Markov process; Regenerative-point technique; Cloud Computing

[*]Correspoding author (Email: rimpalgill@icloud.com).

1. INTRODUCTION

As virtualization technology continues to reshape modern computing, enabling efficient resource utilization and scalability, it also brings forth significant challenges, particularly related to network performance and reliability(Aggarwal and Thangaraju, 2020). Virtualized environments typically consist of multiple virtual machines (VMs) running on a single physical host, which rely heavily on the underlying network infrastructure for communication and data transfer(Nejad, 2022). However, as these environments become increasingly complex, they encounter significant challenges, particularly related to software aging and network reliability.

Software aging manifests as a gradual decline in system performance, often leading to failures due to factors such as resource exhaustion and accumulated faults (Cotroneo et al., 2020). To counteract the effects of software aging, rejuvenation techniques have been introduced, where periodic restarts or updates help restore system performance. A hybrid approach integrating preventive failover strategies and real-time rejuvenation scheduling addresses Mandelbugs in cloud systems by using dynamic reliability models and Dynamic Fault Tree analysis(Carberry et al., 2024). However, network failures can disrupt these processes, leading to performance degradation, service interruptions, and potential data loss (Cerveira et al., 2021).

One critical scenario in virtualized environments occurs during the rejuvenation of the virtual machine monitor (VMM) on the main host. To maintain service continuity, VMs are migrated to a backup host. This migration process, while essential for mitigating the effects of software aging, is particularly susceptible to network-related issues. Failures such as packet loss, latency spikes, and connection drops can significantly hinder the migration process, resulting in extended downtime and diminished system reliability.

To achieve this, we employ the theory of semi-Markov regenerative processes, which allows for a nuanced analysis of the system's response to various types of network failure. By investigating service availability, performance degradation, and overall system robustness under these conditions, our objective is to provide insights that will enhance failure management strategies in virtualized environments. Ultimately, this research seeks to contribute to the development of more resilient cloud computing architectures by addressing the critical interplay between network reliability and virtualization technologies.

2. REVIEW OF LITERATURE

The intersection of virtualization technology and network performance has garnered significant attention in recent research, particularly as organizations increasingly rely

on cloud computing architectures. This review of the literature synthesizes key findings related to software aging, network failures, and their implications for virtualized environments.

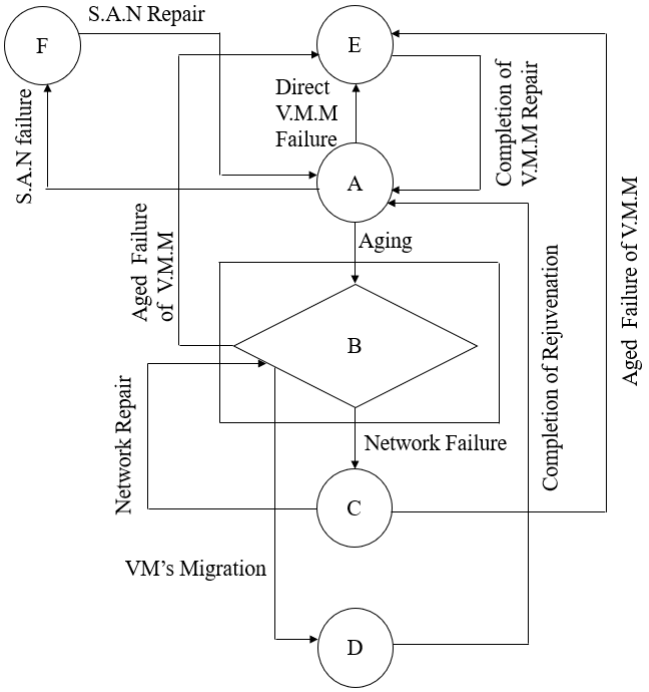
Software aging is a well-documented phenomenon that describes the gradual degradation of system performance due to resource exhaustion, accumulated faults, and increased operational demands. Researchers have explored various rejuvenation techniques to mitigate these effects. Recent studies assess multipurpose VM migration for software rejuvenation and Moving Target Defense, utilizing Stochastic Petri Net models to demonstrate enhancements in both availability and security, despite a trade-off between the two (Torquato et al., 2022). Software aging effects on Docker platforms has been analysed, using a time-series model to predict resource consumption during stress tests on container clusters; findings reveal increasing usage of memory and CPU during scale-up and scale-down operations Oliveira et al. (2021). Several studies have shown that using partial rejuvenation can also improve the system performance and availability compared to systems without rejuvenation or those using full rejuvenation (Rahmani Ghobadi et al., 2022). Furthermore, live VM migration techniques have been adopted to support rejuvenation efforts, optimizing system availability and job completion times (Bai et al., 2020). Iterative methods have also been proposed to enhance real-time task completion, demonstrating the potential for improved system performance through iterative rejuvenation processes (Levitin et al., 2019).

To predict software aging and resource exhaustion, a k-nearest neighbour(K-NN) algorithm is proposed utilizing static and adaptive thresholding methods to enhance service availability through pre-emptive rejuvenation (Parashivamurthy and Cholli, 2023). The challenges of ensuring reliability in public cloud computing by modelling service reliability and implementing AI-based methods are addressed to meet diverse user demands (Meng et al., 2022). A model-driven approach using Stochastic Area Networks is presented to explore the effects of different failure modes and the impact of software rejuvenation on system robusteness and availability (Tola et al., 2021). Multilevel software rejuvenation models utilize distinct policies to regulate system performance. In conclusion, a review of literature highlights the importance of virtualisation and rejuvenation strategies in addressing software aging issues and network failures especially in context of virtual machine(VM) migration to backup hosts.

3. RELATED WORK

Recently, AI-based and conventional live VM migration schemes in cloud computing, focusing on load balancing, energy efficiency, and identifying open research challenges to enhance migration effectiveness in cloud data centers are addressed (Imran et al., 2022) (Haris et al., 2024). Migrating VMs to a backup host mitigates the risk of software aging and potential failures, which has been a focus of many researchers in the past who aim to ensure that VMs can be restored quickly in a controlled environment, thereby minimizing downtime and enhancing overall service availability (Kaur and Anand, 2022). The review of optimization techniques for live virtual machine migration focuses on memory migration but overlooks challenges such as network failures, aging systems, and the economic impacts of migration optimization (Noshy et al., 2018). On the other hand, dynamic resource management in virtualized environments is addressed through a concurrency-aware multiple migration selector, yet it does not take into account rejuvenation strategies, the effects of aging systems on migration efficiency, or their integration with dynamic resource management policies (He et al., 2021). Although there have been significant advancements in system reliability and performance, there are still research gaps in addressing challenges and optimizing strategies, especially concerning rejuvenation techniques during migrations and their integration with network failures. Most existing studies on software aging and rejuvenation in virtualised environments focus only on single-host systems. Although such models are useful, do not account for system resilience in the presence of host failures or the complexity introduced by maintaining service continuity. In contrast, the present work extends beyond single-host models by incorporating dual-host architecture within the virtualised environment. The present work considers a virtualised system with both a main host and a backup host where virtual machines can be migrated during rejuvention. This not only enhances system availability and robustness but provides a more realistic and practical solution for high-reliability environments where uninterrupted service and minimal downtime are critical.

An additional unexplored area is the impact of network failure during virtual machine (VM) migration between two hosts. While the focus has typically been on the migration process itself, the potential for network failure during this operation has not been extensively studied. We investigate the effects of such network failures on the performance of virtualized systems, providing a novel contribution to the literature.



)

Figure 1: Conceptual Model of the system

The system initially operates in state A, which offers three potential transitions. First, the system can move to state F, where a shared memory failure occurs. After the failure is repaired, the system returns to the robust state A. Second, the system may transition from state A to state E, where the failure of the Virtual Machine Monitor (V.M.M) takes place. Once the V.M.M is repaired, the system moves back to state A. The third possibility occurs when the system enters a Failure Probable Zone (state B) due to aging from prolonged software usage. From state B, three sub-paths can follow: The first sub-path involves a V.M.M failure, which leads to a transition to state E for repair, after which the system returns to state A. The second sub-path occurs when aging is detected, prompting the migration of VMs from the active host to a backup host. However, if a network failure occurs between the hosts, the migration process cannot be completed. In this case, the system transitions to state C, where network repair takes place, and once the network is functional, VM migration continues. Following this, the V.M.M rejuvenation process begins on the first host, leading the system to state D. The third sub-path occurs when aging is detected and there is no network failure. The VMs are successfully migrated to the backup host, where the rejuvenation of the V.M.M on the first host begins. Once rejuvenation is completed, data is migrated back to the active host, and the system returns to state A, its robust state

4. NOTATIONS AND ASSUMPTIONS

4.1. Notations and symbols

The notations and symbols used in the model are given in Table 1.

Table 1: Notations and symbols

Notations & Symbols		Descriptions
$g_a(t)/G_a(t)$	:	PDF/CDF of time taken by the system to reach the failure probable zone after aging of V.M.M
$s_c(t)/S_c(t)$	:	PDF/CDF of time taken by the system to complete the rejuvenation process and get back to the robust state
$g_m(t)/G_m(t)$	:	PDF/CDF of time taken by the system to reach the rejuvenation state after migration of VMs from active host to backup host
$m_f(t)/M_f(t)$	:	PDF/CDF of failure time of VMM after aging
$k_f(t)/K_f(t)$	:	PDF/CDF of failure time of VMM
$k_r(t)/K_r(t)$	:	PDF/CDF of repair time of VMM
$v_f(t)/V_f(t)$	:	PDF/CDF of failure time of shared memory
$v_r(t)/V_r(t)$	:	PDF/CDF of repair time of shared memory
$n_f(t)/N_f(t)$	:	PDF/CDF of time taken for a network to enter a failure state during the migration of a Virtual Machine (VM)
$n_r(t)/N_r(t)$	:	PDF/CDF of time taken for a network to reach an aging state after repair
E	:	Set of regenerative states
U	:	Set of up states
γ	:	Failure rate of shared memory
δ	:	Repair rate of shared memory
β	:	Repair rate of V.M.M
b	:	Rate of completion of rejuvenation
λ	:	Failure rate of V.M.M
c	:	Rate at which system reaches failure probable zone
d	:	Rate of rejuvenation after migration
f	:	Failure rate of V.M.M after aging
α	:	Failure rate of network during migration of VM
Ψ	:	Repair rate of network

4.2. Assumptions of Model

- The computing system consists of two hosts, one host is operating i.e., in active state and the other host is backup host which is in passive state i.e., cold standby state.
- There is a single repair-person/ maintenance engineer who is present in system and cannot fail while performing its duty.
- All VMM rejuvenations and repairs are perfect. All the random variables are statistically independent and follow general distributions.
- In Model 1, after completion of repair or rejuvenation of the Virtual Machine Monitor (V.M.M), the software system essentially becomes as robust as new and

the software aging is initiated at the beginning of next highly robust state.

- The failure of Shared Memory is considered only at the initial state of the system.

5. SYSTEM AND STATES DESCRIPTIONS

5.1. State Description

The following are the possible transition states of the system model:

$$S_0 = (H_{1a}, H_{2p}, SAN) \qquad S_1 = (VMM_F, H_{2p}, SAN)$$
$$S_2 = (VMM_{fpz}, H_{2p}, SAN,) \qquad S_3 = (VMM_{rej}, H_{2a}, SAN)$$
$$S_4 = (H_{1a}, H_{2p}, SAN_F) \qquad S_5 = (H_{1a}, H_{2p}, SAN, NF)$$

6. THE MODEL DEVELOPMENT

Let, S_i; i= 0,1,2,3 denote the state of the system. At time t=0, the system starts operation in state S_0. It is observed that all the states are regenerative states. If τ_0,τ_1,τ_2,..... be the time points at which system enters into $S_i \in E$ and X_n be the state visited at instant τ_{n+1} i.e., just after the transition τ_n, then X_n, τ_n, $n \in E$,set of regenerative states represent Markov Renewal stochastic process.

6.1. State Transition Diagrams

The state transition diagram of the Model are represented by fig.2.

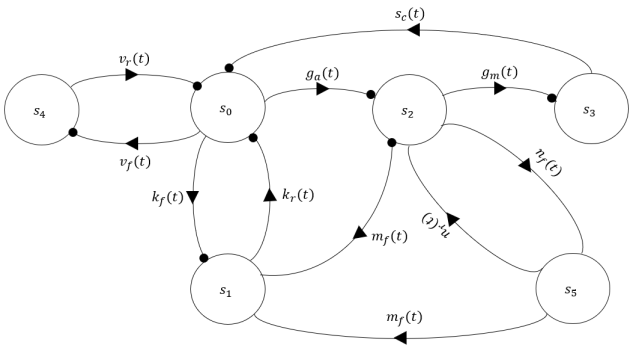


Figure 2: State Transition Diagram

Random Variables utilized in the model development are displayed in Table 2.

Table 2: Random Variables used in the model

Transitions	Descriptions
$S_0 \to S_1$	A random variable with distribution function $K_f(t)$ represents the duration for the VMM of an active host to transition from a running state to a failed state.
$S_0 \to S_2$	A random variable with distribution function $G_a(t)$ represents the duration it takes to reach the failure probable zone after aging.
$S_0 \to S_4$	A random variable with distribution function $V_f(t)$ represents the duration for the shared memory to transition from a running state to a failed state.
$S_1 \to S_0$	A random variable with distribution function $K_r(t)$ denotes the duration for the VMM to transition from a repair state to a robust state.
$S_3 \to S_0$	A random variable with distribution function $S_c(t)$ represents the duration it takes for the system to complete the rejuvenation process and return to a robust state.
$S_4 \to S_0$	A random variable with distribution function $V_r(t)$ denotes the duration for the shared memory to transition from a repair state to a robust state.
$S_2 \to S_1$	A random variable with distribution function $M_f(t)$ represents the duration for the VMM of an active host to reach a failed state after aging.
$S_2 \to S_3$	A random variable with distribution function $G_m(t)$ represents the duration to reach the rejuvenation state after migration of VMs from an active host to a backup host.
$S_2 \to S_5$	A random variable with distribution function $N_f(t)$ represents the duration until the network transitions to a failed state.
$S_5 \to S_2$	A random variable with distribution function $N_r(t)$ represents the duration until the network transitions to an aging state after repair.
$S_5 \to S_1$	A random variable with distribution function $M_f(t)$ represents the duration for the VMM of an active host to reach a failed state after aging and while the network undergoes repair.

6.2. The Transition Probabilities and Mean Sojourn Times

Simple Probabilistic consideration yields the following expression for non-zero elements:

$$p_{i,j} = Q_{i,j}(\infty) = \int_0^\infty q_{i,j}(t)\, dt = \tilde{Q}_{i,j}(0) \tag{1}$$

we get:

$$p_{0,1} = \int_0^\infty k_f(t)\bar{V}f(t)\bar{G}_a(t), dt \qquad p_{0,2} = \int_0^\infty g_a(t)\bar{K}f(t)\bar{V}_f(t), dt$$

$$p_{0,4} = \int_0^\infty v_f(t)\bar{K}f(t)\bar{G}_a(t), dt \qquad p_{1,0} = \int_0^\infty k_r(t), dt$$

$$p_{3,0} = \int_0^\infty s_c(t), dt \qquad p_{4,0} = \int_0^\infty v_r(t), dt$$

$$p_{2,3} = \int_0^\infty g_m(t)\bar{M}_f(t)\bar{N}_f(t), dt \qquad p_{2,1} = \int_0^\infty m_f(t)\bar{G}_m(t)\bar{N}_f(t), dt$$

$$p_{2,5} = \int_0^\infty n_f(t)\bar{M}_f(t)\bar{G}_m(t), dt \qquad p_{5,1} = \int_0^\infty m_f(t)\bar{N}r(t), dt$$

$$p_{5,2} = \int_0^\infty n_r(t)\bar{M}f(t), dt$$

The mean sojourn time in state S_i is given by

$$\mu_i = \int_0^\infty P(T_i > t)dt \tag{2}$$

The mean sojourn times of this model are as follows:

$$\mu_0 = \int_0^\infty \bar{K}_f(t)\bar{V}_f(t)\bar{G}_a(t)dt \qquad \mu_1 = \int_0^\infty \bar{K}_r(t)dt$$

$$\mu_2 = \int_0^\infty \bar{G}_m(t)\bar{M}_f(t)\bar{N}_f(t)dt \qquad \mu_3 = \int_0^\infty \bar{S}_c(t)dt$$

$$\mu_4 = \int_0^\infty \bar{V}_r(t)dt \qquad \mu_5 = \int_0^\infty \bar{M}_f(t)\bar{N}_r(t)dt$$

7. SYSTEM PERFORMANCE

7.1. Reliability and Mean Time to System Failure (MTSF)

Let $\phi_i(t)$ be the c.d.f of the passage time from regenerative state $S_i \varepsilon E$ up to a failed state. Regarding failed state as absorbing state, we have the following recursive relations for $\phi_i(t)$:

$$\begin{aligned}\phi_i(t) = &\sum_{\substack{i,j\in(E\cap U)\\ k,l,m...\in\bar{U}}} \big[Q_{i,j}(t) + Q_{i,j.k}(t) + Q_{i,j.kl} + \cdots + Q_{i,j.klm...}(t)\big][s]\phi_j(t) \\ &+ \sum_{\substack{i\in(E\cap U)\\ f\in\bar{U}}} Q_{i,f}(t); i = 0,2,3,5\end{aligned} \tag{3}$$

Taking Laplace Stieltjes Transformation of eqn. (3) then MTSF is given by

$$MTSF(T_1) = \lim_{s\to 0}\left(\frac{1-\tilde{\phi}_0(s)}{s}\right) = \frac{N_{31}}{D_{31}} \tag{4}$$

where,

$$N_{31} = \mu_0(1 - p_{2,5}p_{5,2}) + \mu_2 p_{0,2} + \mu_3 p_{0,2}p_{2,3} + \mu_5 p_{0,2}p_{2,5}$$
$$D_{31} = 1 - p_{2,5}p_{5,2} - p_{0,2}p_{2,3}p_{3,0}$$

7.2. Steady state availability

Let $M_i(t)$ be the probability that the system is up initially in state $S_i \in E$ is up at time t without visiting any other regenerative state. Further, let $A_i(t)$ = P[$S_i \in U$, at time t without $S_i \in E$ at $t = 0$], then,

$$\delta_{i,j.k,l,...} = \begin{cases} 1 & \text{if there is a transition from } S_i \text{ to } S_j \text{ via } S_k, S_l, \ldots \\ 0 & \text{otherwise} \end{cases}$$

The set of recursive relations for $A_i(t)$ are as follows:

$$A_i(t) = M_{i\in(E\cap U)}(t) + \sum_{\substack{i,j\in(E\cap U)\\ k,l,m...\in\bar{U}}} [q_{i,j,k}(t) + \delta_{i,j,k,l,...}\{q_{i,j,k}(t) + q_{i,j,k,l}(t) + ...\}][c]A_j(t) \quad ; i = 0,1,2,3,4,5 \tag{5}$$

where,

$$M_0(t) = \int_0^\infty \bar{K}_f(t)\bar{V}_f(t)\bar{G}_a(t)dt \qquad M_2(t) = \int_0^\infty \bar{G}_m(t)\bar{M}_f dt$$
$$M_3(t) = \int_0^\infty \bar{S}_c(t)dt \qquad M_5(t) = \int_0^\infty \bar{N}_r(t)\bar{M}_f(t)dt$$

Solving the above recurrence equations using Laplace Transforms, the steady state availability is given by:

$$A_0(\infty) = \lim_{s\to 0} sA_0^*(s) = \frac{N_{32}}{D_{32}} \tag{6}$$

where,

$$N_{32} = (1 - p_{2,5}p_{5,2})\mu_0 + p_{0,2}\mu_2 + p_{0,2}p_{2,5}\mu_5 + p_{0,2}p_{2,3}\mu_3$$
$$D_{32} = (1 - p_{2,5}p_{5,2})(\mu_0 + p_{0,1}\mu_1 + p_{0,4}\mu_4) + p_{0,2}\mu_2 + p_{0,2}p_{2,3}\mu_3 + p_{0,2}p_{2,5}\mu_5 + (p_{0,2}p_{2,5}p_{5,1} + p_{0,2}p_{2,1})\mu_1$$

Suppose that the different random variables included in the model follows Weibull distribution with different parameters. Let the probability density function:

$$g_a(t) = c\eta t^{\eta-1}\exp(-ct^\eta) \quad s_c(t) = b\eta t^{\eta-1}\exp(-bt^\eta) \quad v_f(t) = \gamma\eta t^{\eta-1}\exp(-\gamma t^\eta)$$
$$v_r(t) = \delta\eta t^{\eta-1}\exp(-\delta t^\eta) \quad k_f(t) = \lambda\eta t^{\eta-1}\exp(-\lambda t^\eta) \quad k_r(t) = \beta\eta t^{\eta-1}\exp(-\beta t^\eta)$$
$$g_m(t) = d\eta t^{\eta-1}\exp(-dt^\eta) \quad m_f(t) = f\eta t^{\eta-1}\exp(-ft^\eta) \quad n_f(t) = \alpha\eta t^{\eta-1}\exp(-\alpha t^\eta)$$
$$n_r(t) = \psi\eta t^{\eta-1}\exp(-\psi t^\eta)$$

We obtain the following result:

$$N_{31} = \Gamma\left(\frac{\eta+1}{\eta}\right)\left[\left(\frac{(\lambda+\gamma+c)}{(\lambda+\gamma+c)^{\frac{\eta+1}{\eta}}}\right)\left(1-\frac{\alpha\psi}{(d+f+\alpha)(\psi+f)}\right)\right.$$
$$+\left(\frac{c}{c+\lambda+\gamma}\right)\left[\left(\frac{(d+f+\alpha)}{(d+f+\alpha)^{\frac{\eta+1}{\eta}}}\right)\right.$$
$$\left.\left.+\left(\frac{cdb}{b^{\frac{\eta+1}{\eta}}(d+f+\alpha)}\right)+\left(\frac{\alpha(\psi+f)}{(\psi+f)^{\frac{\eta+1}{\eta}}(d+f+\alpha)}\right)\right]\right]$$
$$D_{31} = 1-\left(\frac{c}{c+\lambda+\gamma}\right)\left(\frac{d}{d+f+\alpha}\right)+\left(\frac{\alpha}{d+f+\alpha}\right)\left(\frac{\psi}{\psi+f}\right)$$
$$N_{32} = N_{31}$$
$$D_{32} = \Gamma\left(\frac{\eta+1}{\eta}\right)\left[\left(1-\frac{\alpha\psi}{(d+f+\alpha)(\psi+f)}\right)\left(\frac{\lambda+\gamma+c}{(\lambda+\gamma+c)^{\frac{\eta+1}{\eta}}}\right)\right.$$
$$+\left(\frac{\lambda}{c+\lambda+\gamma}\right)\left(\frac{\beta}{\beta^{\frac{\eta+1}{\eta}}}\right)+\left(\frac{\gamma}{c+\lambda+\gamma}\right)\left(\frac{\delta}{\delta^{\frac{\eta+1}{\eta}}}\right)$$
$$+\left(\frac{c}{(c+\lambda+\gamma)(d+f+\alpha)}\right)\left[\left(\frac{\alpha(\psi+f)}{(\psi+f)^{\frac{\eta+1}{\eta}}}\right)+\left(\frac{bd}{b^{\frac{\eta+1}{\eta}}}\right)+\left(\frac{f\alpha\beta}{(\psi+f)\beta^{\frac{\eta+1}{\eta}}}\right)\right]\Bigg]$$
$$+\left(\frac{c(d+f+\alpha)}{(\lambda+\gamma+c)(d+f+\alpha)^{\frac{\eta+1}{\eta}}}\right)$$

8. RESULTS AND DISCUSSIONS

A set of hypothetical numerical values are assigned to different parameters as outlined in Table 3.

Table 3: Data Set

Parameters	Values
Rejuvenation rate(d)	0.1 to 0.5
Degradation rate(c)	0.03 and 0.05
Failure rate of shared memory(γ)	0.0002 and 0.0007
Repair rate of shared memory(δ)	0.295 and 0.695
Repair rate of V.M.M(β)	0.695 and 0.95
Rate of completion of rejuvenation(b)	0.0495
Direct Failure rate of V.M.M(λ)	0.002 and 0.005
Aged failure rate of V.M.M(f)	0.3 and 0.5
Rate of Network failure(α)	0.0003 and 0.0007
Rate of Network repair(ψ)	0.175 and 0.795

Table 4 showcases the behavior of the Mean time to system failure (MTSF) as the rejuvenation rate (d) varies, while other parameters remain constant. These parameters include the VMM direct failure rate (λ), VMM aged failure rate (f), VMM degradation rate (c), shared memory failure rate (γ), shared memory repair rate (δ), network failure rate (α), network repair rate (ψ), and completion rate (b). The table clearly indicates that higher values of the rejuvenation rate (d) lead to significant boost in MTSF and overall reliability of the system. Conversely, it has been observed that increase in the direct failure rate (λ) from 0.002 to 0.005, the VMM aged failure rate (f) from 0.5 to 0.7, the shared memory failure rate (γ) from 0.0002 to 0.0007, the network failure rate (α) from 0.0003 to 0.0007 and the degradation rate (c) from 0.03 to 0.05 decreases MTSF, which can lead to operational inefficiencies, higher costs, and increased risks, significantly impacting the system's performance and reliability. Furthermore, it has been observed that higher values of the shape parameter (η) shown in the table significantly reduce MTSF because:

- $(\eta < 1)$: A decreasing failure rate over time leads to a higher MTSF because the surviving items become increasingly robust.
- $(\eta = 1)$: The failure rate remains constant. As a result, the MTSF is determined by the constant failure rate, with no variation in reliability over time.
- $(\eta > 1)$: An increasing failure rate over time leads to a lower MTSF because the likelihood of failure grows with time, reducing the average lifespan.

Therefore, the shape parameter (η) fundamentally alters the failure dynamics, significantly influencing the mean time to system failure.

Similarly, Table 5 present key findings on system availability as the rejuvenation rate (d) varies, while other parameters remain fixed. These parameters include the VMM direct failure rate (λ), VMM aged failure rate (f), VMM repair rate (β), VMM degradation rate (c), shared memory failure rate (γ), shared memory repair rate (δ), network failure rate, network repair rate and completion rate (b). The data clearly show that higher rejuvenation rates (d), along with increased repair rates for shared memory (δ), network (ψ) and VMM (β), significantly enhances system availability. Conversely, increases in the direct failure rate of VMM (λ) from 0.002 to 0.005, the aged failure rate of VMM (f) from 0.5 to 0.7, the degradation rate (c) from 0.03 to 0.05, network failure rate (ψ) from 0.0003 to 0.0007 and the shared memory failure rate (γ) from 0.0002 to 0.0007 negatively impact system availability. As the shape parameter (η) of the Weibull distribution increases, the system experiences a higher failure rate over time. This increased failure rate results in more frequent maintenance, longer downtimes, higher

maintenance costs, operational disruptions, and potentially lower customer satisfaction. All these factors contribute to decreased system availability.

9. CONCLUSION

This study integrates the impact of network failure into the rejuvenation and migration processes of Virtual Machine Monitors (VMMs), providing a comprehensive analysis of how network reliability affects system performance. Using a semi-Markov regenerative process model, we have explored how network reliability influences the effectiveness of rejuvenation strategies and overall system performance.

Our findings illustrate that while rejuvenation through VMM migration to a backup host can significantly enhance mean time to system failure (M.T.S.F), system availability, and network failures introduce additional complexities. Effective rejuvenation not only prolongs system longevity but also mitigates the adverse effects of aging. However, network reliability plays a crucial role in determining the success of migration processes, with network failures potentially impacting the efficacy of rejuvenation efforts and leading to increased operational challenges.

The study demonstrates that strategic management of network reliability, alongside optimized rejuvenation and repair rates, is essential for maximizing system performance and minimizing downtime. By addressing network failure risks and improving migration reliability, organizations can better manage system aging, reduce maintenance costs, and enhance overall profitability.

Ultimately, our research underscores the importance of a holistic approach to system management that encompasses both rejuvenation strategies and network reliability considerations. Future work should further investigate adaptive rejuvenation policies and advanced network fault-tolerance techniques to refine system performance and ensure sustained operational efficiency.

REFERENCES

Aggarwal, V. and Thangaraju, B. (2020). Performance analysis of virtualisation technologies in nfv and edge deployments. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–5. IEEE. DOI: 10.1109/CONECCT50063.2020.9198367.

Bai, J., Chang, X., Machida, F., Trivedi, K. S., and Han, Z. (2020). Analyzing software rejuvenation techniques in a virtualized system: Service provider and user views. *IEEE Access*, 8:6448–6459. DOI: 10.1109/ACCESS.2019.2963397.

Carberry, J. R., Rahme, J., and Xu, H. (2024). Real-time rejuvenation scheduling for cloud systems with virtualized software spares. *Journal of Systems and Software*, 217:112168. DOI: 10.1016/j.jss.2024.112168.

Cerveira, F., Barbosa, R., and Madeira, H. (2021). Mitigating virtualization failures through migration to a co-located hypervisor. *IEEE Access*, 9:105255–105269. DOI: 10.1109/ACCESS.2021.3098644.

Cotroneo, D., Iannillo, A. K., Natella, R., and Pietrantuono, R. (2020). A comprehensive study on software aging across android versions and vendors. *Empirical Software Engineering*, 25:3357–3395. DOI: 10.1504/IJSETA.2024.141315.

Haris, R. M., Barhamgi, M., Nhlabatsi, A., and Khan, K. M. (2024). Optimizing pre-copy live virtual machine migration in cloud computing using machine learning-based prediction model. *Computing*, 106(9):3031–3062. DOI: 10.1007/s00607-024-01318-6.

He, T., Toosi, A. N., and Buyya, R. (2021). Camig: Concurrency-aware live migration management of multiple virtual machines in sdn-enabled clouds. *IEEE Transactions on Parallel and Distributed Systems*, 33(10):2318–2331. DOI: 10.1109/TPDS.2021.3139014.

Imran, M., Ibrahim, M., Din, M. S. U., Rehman, M. A. U., and Kim, B. S. (2022). Live virtual machine migration: A survey, research challenges, and future directions. *Computers and Electrical Engineering*, 103:108297. DOI: 10.1016/j.compeleceng.2022.108297.

Kaur, H. and Anand, A. (2022). Review and analysis of secure energy efficient resource optimization approaches for virtual machine migration in cloud computing. *Measurement: Sensors*, 24:100504. DOI: 10.1016/j.measen.2022.100504.

Levitin, G., Xing, L., and Huang, H.-Z. (2019). Optimization of partial software rejuvenation policy. *Reliability Engineering & System Safety*, 188:289–296. DOI: 10.1016/j.ress.2019.03.011.

Meng, S., Luo, L., Qiu, X., and Dai, Y. (2022). Service-oriented reliability modeling and autonomous optimization of reliability for public cloud computing systems. *IEEE Transactions on Reliability*, 71(2):527–538. DOI: 10.1109/TR.2022.3154651.

Nejad, B. (2022). Virtualisation. In *Introduction to Satellite Ground Segment Systems Engineering: Principles and Operational Aspects*, pages 199–209. Springer.

Noshy, M., Ibrahim, A., and Ali, H. A. (2018). Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications*, 110:1–10. DOI: 10.1016/j.jnca.2018.03.002.

Oliveira, F., Araujo, J., Matos, R., and Maciel, P. (2021). Software aging in container-based virtualization: an experimental analysis on docker platform. In *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–7. IEEE. DOI: 10.23919/CISTI52073.2021.9476625.

Parashivamurthy, S. and Cholli, N. G. (2023). Software aging prediction-a new approach. *International Journal of Electrical & Computer Engineering (2088-8708)*, 13(2). DOI: 10.11591/ijece.v13i2.pp1773-1781.

Rahmani Ghobadi, Z., Rashidi, H., and Alizadeh, S. (2022). On multiple objective of software rejuvenation models with several policies. *Journal of Electrical and Computer Engineering Innovations (JECEI)*, 10(1):25–36. DOI: 10.22061/jecei.2021.7905.448.

Tola, B., Jiang, Y., and Helvik, B. E. (2021). Model-driven availability assessment of the nfv-mano with software rejuvenation. *IEEE Transactions on Network and Service Management*, 18(3):2460–2477. DOI: 10.1109/TNSM.2021.3090208.

Torquato, M., Maciel, P., and Vieira, M. (2022). Software rejuvenation meets moving target defense: Modeling of time-based virtual machine migration approach. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*, pages 205–216. IEEE. DOI: 10.1109/ISSRE55969.2022.00029.

Table 4: Effect of various parameters on MTSF w.r.t Rejuvenation rate (d)

Rejuvenation rate (d)	$T_1(\lambda = 0.002, \gamma = 0.0002, c = 0.03, f = 0.5, \alpha = 0.0003, \psi = 0.175)$	$T_1(\lambda = 0.005)$	$T_1(\gamma = 0.0007)$	$T_1(c = 0.05)$	$T_1(f = 0.7)$	$T_1(\alpha = 0.0007)$	$T_1(\psi = 0.795)$
for $\eta < 1$ ($\eta = 0.7$)							
0.1	240.20157	209.49747	234.55486	140.83604	220.50034	240.13219	240.23795
0.2	372.05831	322.38457	362.84414	262.19954	315.65063	371.88747	372.16439
0.3	567.51889	487.81170	552.59909	451.57223	457.44117	567.20037	567.72625
0.4	824.10309	701.60814	800.95577	707.28162	644.55511	823.59439	824.44081
0.5	1139.46076	960.03516	1105.22798	1027.71270	875.72546	1138.72277	1139.95559
for $\eta = 1$							
0.1	46.09814	41.51919	45.26611	32.44299	41.84352	46.08448	46.10586
0.2	73.72405	65.41923	72.19652	58.99454	61.77132	73.68908	73.74637
0.3	115.22820	100.79684	112.54269	100.35307	91.87246	115.16184	115.27203
0.4	170.07142	146.74263	165.68148	156.15653	131.86035	169.96437	170.14313
0.5	237.74213	202.44083	231.02776	226.05460	181.45905	237.58584	237.84756
for $\eta > 1$ ($\eta = 1.7$)							
0.1	11.24009	10.39180	11.08703	9.40889	10.08508	11.23710	11.24232
0.2	18.23107	16.48154	17.91052	16.38291	15.11970	18.22265	18.23700
0.3	28.81112	25.54093	28.20397	27.18020	22.78624	28.79467	28.82253
0.4	42.84210	37.33579	41.80749	41.70865	33.01107	42.81521	42.86064
0.5	60.19259	51.65576	58.57056	59.87876	45.72317	60.15305	60.21977

Table 5: Effect of various parameters on System Availability w.r.t Rejuvenation rate (d)

Rejuvenation rate (d)	$A_1(\lambda = 0.002, \gamma = 0.0002, c = 0.03, f = 0.5, \delta = 0.295, \beta = 0.695, b = 0.0495, \alpha = 0.0003, \psi = 0.175)$	$A_1(\lambda = 0.005)$	$A_1(\gamma = 0.0007)$	$A_1(c = 0.05)$	$A_1(f = 0.7)$	$A_1(\delta = 0.695)$	$A_1(\beta = 0.95)$	$A_1(\alpha = 0.0007)$	$A_1(\psi = 0.795)$
for $\eta < 1$ ($\eta = 0.7$)									
0.1	0.990372	0.988967	0.989725	0.982788	0.989702	0.990537	0.993733	0.990371	0.990372
0.2	0.992015	0.990733	0.991409	0.986373	0.991120	0.992173	0.994795	0.992014	0.992016
0.3	0.993149	0.991948	0.992569	0.988685	0.992174	0.993302	0.995527	0.993148	0.993150
0.4	0.993978	0.992835	0.993418	0.990300	0.992988	0.994127	0.996063	0.993977	0.993979
0.5	0.994611	0.993511	0.994065	0.991492	0.993635	0.994757	0.996471	0.994609	0.994612
for $\eta = 1$									
0.1	0.966796	0.963303	0.965422	0.951707	0.964205	0.967113	0.975340	0.966792	0.966797
0.2	0.972494	0.969149	0.971178	0.961163	0.969188	0.972798	0.979584	0.972489	0.972496
0.3	0.976423	0.973180	0.975147	0.967425	0.972884	0.976717	0.982503	0.976417	0.976425
0.4	0.979295	0.976127	0.978049	0.971877	0.975734	0.979582	0.984633	0.979289	0.979297
0.5	0.981487	0.978377	0.980263	0.975205	0.977998	0.981769	0.986256	0.981481	0.981489
for $\eta > 1$ ($\eta = 1.7$)									
0.1	0.902338	0.895476	0.900121	0.883378	0.894185	0.902761	0.917199	0.902330	0.902342
0.2	0.918161	0.910996	0.915912	0.903614	0.908054	0.918576	0.930776	0.918149	0.918167
0.3	0.929269	0.921911	0.927001	0.917612	0.918501	0.929678	0.940263	0.929254	0.929276
0.4	0.937499	0.930009	0.935219	0.927875	0.926654	0.937904	0.947269	0.937484	0.937506
0.5	0.943844	0.936258	0.941555	0.935725	0.933196	0.944245	0.952657	0.943828	0.943850