Risk Assessment and Management Decisions

www.ramd.reapress.com

Risk Assess. Manage. Decis. Vol. 1, No. 1 (2024) 102–118.

Paper Type: Original Article

Optimizing System Availability in Client-Server Network through Fog Computing: A Stochastic Model with Foggy Markovian Paths

Ibrahim Yusuf[1,4],*, Khadija Salihu Auta[2], Muhammad Kabeer[3]

[1] Department of Mathematical Sciences, Bayero University, Kano, Nigeria; iyusuf.mth@buk.edu.ng.
[2] Department of Computer Science, Bayero University Kano, Nigeria; autakhadijat@gmail.com.
[3] Department of Computer Science, Federal University Dutsinma, Katsina, Nigeria; mkabeer@fudutsinma.edu.ng.
[4] Operation Research Group, Bayero University, Kano, Nigeria; iyusuf.mth@buk.edu.ng.

Citation:

Received: ----- *Revised:* ----- *Accepted:*---	Yusuf, I., Salihu Auta, Kh., & Kabeer, M. (2024). Optimizing system availability in client-server network through fog computing: a stochastic model with foggy markovian paths. *Risk assessment and management decisions, 1* (1), 102-118.

Abstract

The main goal of this paper's extensive analysis of a client-server fog computing network is to increase system availability. Fog computing, which extends cloud computing to the network's edge, necessitates a robust and reliable architecture to handle distributed computational tasks effectively. To achieve this, the paper introduces a sophisticated architecture comprising five distinct subsystems: A, B, C, D, and E. Each subsystem plays a critical role in ensuring the seamless operation of the network. Subsystem A represents the clients, the devices or applications that generate computational tasks. Subsystem B consists of fog nodes strategically placed closer to the clients to process data with minimal latency. Subsystem C includes the servers, which provide more substantial computational power and storage capacity. Subsystems D and E function as first- and second-level load balancers to distribute the workload efficiently across the network. The arrangement of these subsystems is meticulously designed to enhance the overall performance and availability of the network. The system can distribute and manage computational tasks more effectively by organizing clients, fog nodes, servers, and load balancers in a series-parallel configuration. This setup allows optimal resource utilization and ensures the network can handle varying loads without compromising availability. To model the availability dynamics of the network, the study employs differential-difference equations and a transition diagram. These mathematical tools help understand the system's long-run availability under different conditions. The analysis involves conducting numerical experiments thoroughly documented using tables and graphs. These visual aids effectively illustrate how various network parameters influence the optimization of system availability. The findings from these experiments underscore the vital role of load balancers and fog nodes configured in a series-parallel arrangement. This configuration not only facilitates optimal task distribution but also significantly boosts the overall availability of the system. The study concludes by emphasizing the effectiveness of this approach, highlighting it as a strategic method to enhance system availability in client-server fog computing networks. The results of this study provide valuable insights for researchers, system administrators, and network architects. By demonstrating the benefits of a series-parallel configuration of fog nodes and load balancers, the paper offers practical guidance for improving the performance and reliability of fog computing environments. These findings can help stakeholders design more resilient and efficient networks, ultimately advancing the field of fog computing.

Keywords: Network, Availability, Computing, Reliability.

1 | Introduction

The landscape of research in information technology has witnessed an abundance of studies focusing on the performance analysis of client-server systems. These investigations have significantly contributed to

Corresponding Author: iyusuf.mth@buk.edu.ng

understanding efficiency, response times, and resource utilization within traditional client-server architectures. However, despite the wealth of knowledge in performance analysis, there is a conspicuous gap in our understanding of availability modelling and assessment tailored to the novel paradigm of client-server networks augmented by fog computing. The conventional research emphasis on performance analysis often centres around optimizing throughput, minimizing latency, and maximizing resource utilization. While these aspects are undeniably pivotal for gauging the efficiency of client-server systems, ensuring the continuous and reliable availability of services is equally critical, especially in dynamic and distributed environments. As an innovative extension of cloud computing, fog computing introduces a decentralized approach to processing data by distributing computational resources closer to the network's edge. This paradigm shift, however, brings forth a set of unique challenges and opportunities, particularly in the realm of availability. Unlike traditional client-server setups, fog computing incorporates edge nodes that act as intermediaries, and the implications of these nodes on system availability remain a comparatively unexplored domain in the current body of research.

The essence of fog computing lies in its role as a transformative enabler. It provides a vital framework for applications and businesses striving to flourish in environments characterized by dynamism and exacting demands. It acts as a linchpin for sustained success in the face of evolving technological landscapes and increasing user expectations. By strategically bringing computational processes closer to the network's edge, fog computing acts as a catalyst, elevating client-server systems' overall performance and fortitude.

Literature on fog computing is abundant and continuously expanding, reflecting the growing interest and research efforts in this field. To cite a few prominent examples, Mahmud et al. [1] explored the potential of fog computing in healthcare applications, examining its role in enabling real-time monitoring, analysis, and decision-making in medical environments. Mouradian et al. [2] conducted a comprehensive survey of fog computing architectures, technologies, and applications, providing a detailed overview of the state-of-the-art developments in the field. Yi et al. [3] proposed a fog computing-based framework for smart cities, focusing on efficiently managing urban infrastructure, services, and resources through distributed computing at the network edge. Zhang et al. [3] delve into an in-depth analysis and evaluation of the frameworks and configurations prevalent in fog computing. Within their examination, they shed light on the pertinent security and trust challenges that accompany these architectures. Yousefpour et al. [4] extensively explore the landscape of computing paradigms and their interconnectedness. They delve into a survey that delineates the nuances of various computing paradigms and highlights their shared characteristics and distinctions. Moreover, the researchers meticulously construct a taxonomy specifically tailored to fog computing, shedding light on its multifaceted nature and classifying its diverse endeavours alongside related computing paradigms. Through their meticulous categorization efforts, Yousefpour et al. [4] provide valuable insights into the evolving field of fog computing and its intricate relationship with other computing paradigms.

Yi et al. [3] undertake an extensive survey aimed at elucidating key concepts within the realm of fog computing, alongside exploring its potential application scenarios. This comprehensive investigation identifies many issues that designers and implementers may encounter when tasked with the development and deployment of fog computing systems. Through their meticulous examination, Yi et al. [3] illuminate various facets of these challenges, ranging from technical intricacies to practical considerations, thus providing valuable insights for researchers, engineers, and practitioners alike in the burgeoning field of fog computing. Stojmenovic et al. [5] delve deeply into the multifaceted advantages of fog computing, a burgeoning paradigm in distributed computing. The study elucidates the diverse applications of fog computing across a spectrum of real-world scenarios through meticulous analysis.

Alli and Alam [6] conducted an in-depth investigation into fog edge computing through an extensive survey. Their primary aim was to establish a comprehensive understanding as a foundational framework for proposed solutions within studies involving IoT-Fog-Cloud ecosystems. This research delves into the intricacies of fog edge computing. This paradigm bridges the gap between traditional cloud computing and IoT devices by bringing computational capabilities closer to the data source. By analyzing various aspects such as architecture, deployment models, communication protocols, and application domains, the study aims to provide insights into the evolving landscape of fog edge computing. Their study contributes to developing efficient and scalable solutions tailored to address the challenges and requirements of IoT environments, where real-time processing, low latency, and bandwidth optimization are critical factors.

Bonomi et al. [7] assert that fog computing serves as an optimal platform for critical IoT services and applications such as connected vehicles, Smart grids, smart cities, and Wireless Sensors and Actuators Networks (WSANs). Its ability to address the specific demands of these domains, including low latency, scalability, and real-time processing, positions fog computing as a fundamental enabler of IoT-driven innovations in various sectors. Caiza et al. [8] provide valuable insights into the current state of fog computing technology in industrial applications. By addressing key considerations such as architecture, security, latency, and energy consumption, the study contributes to a comprehensive understanding of the opportunities and challenges associated with adopting fog computing in industrial settings. Hu et al. [9] conducted a comprehensive overview and summary of fog computing, covering various aspects, including model architecture, key technologies, applications, challenges, and open issues. The study aims to provide a holistic understanding of fog computing, addressing its conceptual framework, technological components, practical applications, and the obstacles and areas for further exploration within the field.

Yusuf and Auta [10] thoroughly analyzed the availability aspect of a distributed system comprising two hosts acting as clients linked to two servers through a load balancer. The study investigated the system's performance under four distinct maintenance scenarios. These maintenance options included perfect repair and replacement upon total failure, minimal repair upon failure occurrence, total replacement upon complete failure, and online Preventive Maintenance (PM) upon partial failure. Additionally, the study explored the integration of a fault-tolerant factor alongside total replacement upon complete failure.

Availability modelling in the context of client-server networks through fog computing necessitates a nuanced examination of factors such as fault tolerance, redundancy, and the intricate interplay between edge computing and the central server. It requires researchers to delve into uncharted territories, developing models that encapsulate the dynamic nature of fog-enabled architectures and their impact on the reliability of services. The focus on performance analysis of client-server systems has been prominent, but availability modelling and assessment in the context of client-server networks through fog computing represent a niche area of research. However, the importance of system availability in ensuring uninterrupted services makes this a crucial field of study. By delving into these areas, researchers can contribute to the growing body of knowledge on the availability modelling and assessment of client-server networks through fog computing, addressing a crucial aspect of ensuring reliable and uninterrupted services in modern distributed computing environments.

Optimizing system availability in a client-server network through fog computing involves implementing a sophisticated and resilient architecture that leverages the power of edge computing to enhance the overall performance and reliability of the system. Fog computing extends the capabilities of cloud computing by bringing computation, storage, and networking resources closer to the network edge, thereby reducing latency and improving responsiveness. This approach is particularly beneficial in client-server architectures where timely and efficient communication between clients and servers is crucial.

The existing body of literature about enhancing performance analysis in client-server networks is extensive and has garnered significant attention. However, a noticeable gap persists in the realm of evaluating and optimizing critical availability and performance metrics within this domain. Specifically, there has been limited exploration of the intricate interplay between availability and performance metrics, such as dependability, Mean Time to Failure (MTTF), availability, Mean Time to Repair (MTTR), and Mean Time between Failure (MTBF), within the context of client-server networks when leveraging the paradigm of fog computing.

The current research landscape has predominantly centred on performance optimization, throughput enhancement, and latency reduction in client-server networks. While these aspects are undeniably pivotal for ensuring efficient data transfer and user experiences, the broader spectrum of reliability metrics has not been comprehensively addressed. Reliability, a cornerstone in network engineering, encompasses the system's ability to consistently perform its intended functions over time, thereby highlighting its dependability and resilience. Within this context, there is a notable gap in the literature concerning applying fog computing to enhance availability metrics in client-server networks. Metrics such as MTTF, which signifies the expected time until a system failure, and MTTR, indicating the average time required to restore the system after a failure, are crucial benchmarks for evaluating system robustness and recovery capabilities. Availability, representing the proportion of time a system is operational, is equally vital and plays a pivotal role in determining the overall user experience and business continuity. As the landscape of network architectures evolves with the integration of fog computing, it becomes imperative to bridge the existing gap in the literature by delving into the systematic evaluation and optimization of reliability and performance metrics. This endeavour not only contributes to the theoretical understanding of network reliability but also holds practical implications for industries and enterprises relying on robust and dependable client-server infrastructures. Addressing this research gap is essential for unlocking the full potential of fog computing in fortifying the reliability and overall performance of client-server networks in the face of dynamic and demanding environments.

In response to the identified research gap, this paper embarks on a comprehensive exploration focused on modelling and optimizing availability within a client-server system leveraging fog computing. The primary objective is to augment system availability, a critical aspect often overlooked in the existing literature. The proposed framework delves into a sophisticated network architecture encompassing five intricately arranged subsystems: clients, fog nodes, servers, first-level load balancers, and second-level load balancers. This sequential arrangement is meticulously designed to facilitate the efficient distribution and management of computational tasks throughout the network. In essence, this paper endeavours to fill the existing void in the literature by providing a detailed exploration of availability modelling and optimization within a client-server system facilitated by fog computing. The proposed network architecture and subsystem arrangement offer a promising avenue for enhancing system reliability and performance in dynamic and demanding environments, laying the groundwork for advancements in the field of network engineering and distributed computing.

Notations

CL: client.

FN: fog node.

LB: load balancer.

SV: server.

v_1 / m_1: rate of failure/repair of client.

v_2 / m_2: rate of failure/repair of first-level load balancer.

v_3 / m_3: rate of failure/repair of fog nodes.

v_4 / m_4: rate of failure/repair of server.

v_5 / m_5: rate of failure/repair of second-level load balancer.

c: coverage factor.

2 | Description and States of the Proposed Network/System

This architecture in *Fig. 1* below is designed to distribute and manage computational tasks efficiently between the clients and servers, leveraging fog computing for improved latency and responsiveness. Here's an overview.

Subsystem A (clients): this subsystem consists of two active parallel clients. The clients serve as the end-user devices or applications that initiate requests to the network. Clients send tasks or data to the network for processing. Communication ceased whenever both clients were down.

Subsystem B (first level load balancer): the first load balancer is responsible for distributing incoming client requests to the fog nodes. It ensures the even distribution of tasks among the available fog nodes, optimizing resource usage.

Subsystem C (fog nodes): these are distributed computing nodes situated at the network's edge, closer to the clients. Fog nodes are equipped to process tasks locally, reducing latency and improving response times for certain applications. Each fog node is active and capable of handling computational tasks.

Subsystem D (second level load balancer): the second load balancer operates at a higher level, distributing tasks from the fog nodes to the servers. It ensures that the computational load is evenly distributed among the available servers.

Subsystem E (servers): these are the backend servers responsible for executing more resource-intensive tasks that may not be suitable for processing at the fog nodes or clients. Servers receive tasks from the fog nodes and return the results to them for further distribution to the clients.

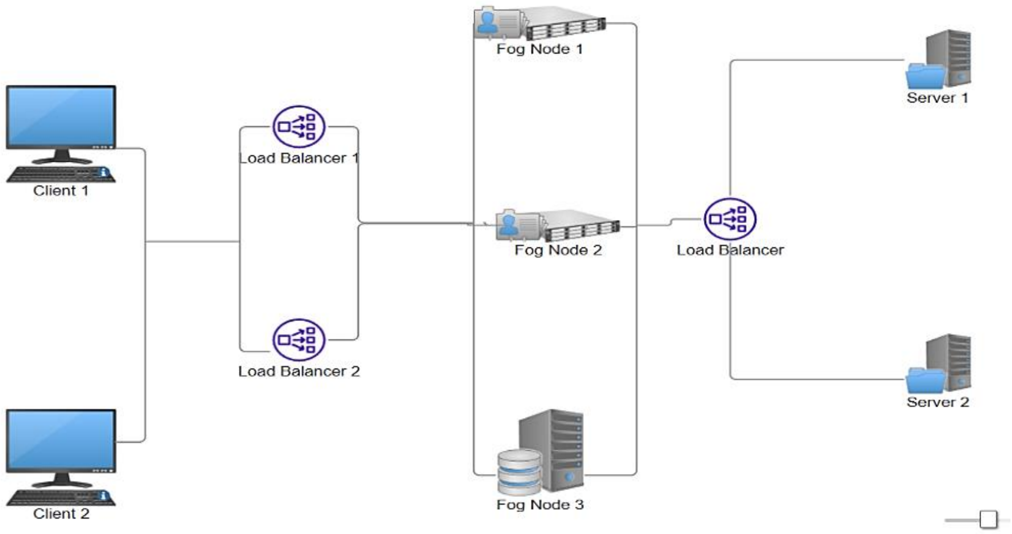

Fig. 1. Reliability block diagram of the network.

Table 1. State of the network.

State	CL 1	CL 2	1st Level LB 1	LB 2	FN 1	FN 2	FN 3	2nd Level LB	SV 1	SV 2	Network Status
0	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good
1	Good	Good	Failed	Good	Good	Good	Good	Good	Good	Good	Good
2	Failed	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good
3	Good	Good	Good	Good	Failed	Good	Good	Good	Good	Good	Good
4	Good	Good	Good	Good	Failed	Failed	Good	Good	Good	Good	Good
5	Good	Good	Good	Good	Good	Good	Good	Good	Failed	Good	Good
6	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Failed	Idle	Idle	Down
7	Idle	Idle	Failed	Failed	Idle	Idle	Idle	Idle	Idle	Idle	Down
8	Failed	Failed	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Down
9	Idle	Idle	Idle	Idle	Failed	Failed	Failed	Idle	Idle	Idle	Down
10	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Failed	Failed	Down

Table 2. Related studies.

Article	Problem	Solution	Weakness
Yi et al. [3]	Inherent limitations and challenges of cloud computing include unreliable latency, lack of mobility support, and location awareness.	The paper discusses computation offloading to overcome resource constraints on mobile devices, enabling performance improvements, storage savings, and extended battery life.	The paper discusses the challenges related to the reliability requirement of fog computing, particularly in highly dynamic environments. Traditional approaches such as periodical check-pointing, rescheduling of failed tasks, and replication may not suit the highly dynamic nature of fog computing, leading to potential reliability issues.
Melnik et al. [11]	Reliability issue of Information Control Systems (ICS)	The authors considered simplified ICS structures to synthesize models of element workload and their reliability functions.	The model can be tested using better simulation environments.
Pereira et al. [12]	Reliable smart mobility applications in vehicular environments, which require low-latency and short-distance local connections.	The authors proposed a hybrid VANET/fog computing architecture that enables the development of reliable smart mobility applications.	It is important to note that while the study evaluates the performance of the architecture using real vehicular mobility traces.
Al-khafajiy M. et al. [13]		They propose a novel Optimal Fog Algorithm (OFA).	The model does not consider the actual networking environment and hence lacks knowledge of network topology.
Chen et al. [15]	The paper aims to address the challenge of assessing the reliability of Multi-state Cloud Computer Networks (MCCNs), which are complex networks comprising IoT devices, fog servers, and cloud servers for data transmission.	The paper constructs a mathematical model to elucidate the flow relationship among IoT devices, edge servers, and cloud servers in modern cloud computer networks.	The paper may rely on certain assumptions about the network or data transmission, and these assumptions could impact the generalizability of the findings. The practical implementation and scalability of the proposed algorithm in real cloud and fog computing networks may not be fully addressed.
Montoya-Munoz et al. [15]	The paper addresses the problem of ensuring reliable data collection in Smart Farming supported by the Internet of Things (IoT).	The paper proposes an approach based on fog computing to provide reliability in IoT data collection in smart farming. The Failure Detection Mechanism (FDM) finds outliers in datasets by Machine Learning (ML) algorithms, and the Failure Recovery Mechanism (FRM) replaces the identified outliers with data inferred by using interpolation techniques.	The study only evaluates the proposed approach in a single case study involving a Colombian coffee smart farm, which may limit the generalizability of the results to other types of smart farming systems.
Kabeer et al. [16]	Software defined network single point of failure and network bottlenecks.	Introduced a novel hierarchical Distributed Software Defined Network (DSDN) to address network bottlenecks.	Tradeoffs regarding cost were made, making the approach only viable in cases of high importance.

Table 2. Continued.

Article	Problem	Solution	Weakness
Das and Inuwa [17]	Specifically, the paper discusses the advantages of fog computing and its relationship to cloud and edge computing and presents a taxonomy based on contemporary research.	A fog computing Taxonomy is presented based on contemporary fog computing research about security challenges, services issues, operational issues, and data management.	More papers could have been reviewed.
Proposed study	Investigates the impact of configuring fog nodes and load balancers in a series-parallel arrangement on system availability. Determining the most effective configuration for these subsystems to enhance availability and how to improve the performance and dependability of fog computing environments.	The study aims to provide a comprehensive approach to optimizing system availability and performance in client-server fog computing networks.	The study appears to focus primarily on availability. Other crucial aspects such as security, latency, and throughput are not mentioned, which are also important for fog computing networks' overall performance and reliability.

3 | Availability Model Formulation and Presentation

The system is modelled based on the transition diagram of the proposed system in Section 3. Kolmogorov's differential-difference equations are derived for the calculation of the state probabilities. The system availability can then be obtained from the solutions of the state probabilities of the only working states of the system. In order to analyze the availability of the system, let's define $T_i(q)$ to be the probability that the system is in state i at time q and $T(q)=[T_1(q),T_2(q),\ldots,T_{10}(q)]$ be the probability row vector with initial conditions.

$$T_k(0)=\begin{cases}1, & k=0,\\ 0, & k=1,2,3,\ldots,10\end{cases}. \tag{1}$$

The steady-state probability of systems availability can be obtained from the solutions $T_i(q),\ i=0,1,2,3,\ldots,10$. State 0,1,2,3,4, and 5 are the only working states of the system in *Table 1*.

The following set of differential equations were derived from the transition diagram in *Table 1*.

$$\begin{pmatrix} T_0'(q)\\ T_1'(q)\\ T_2'(q)\\ T_3'(q)\\ T_4'(q)\\ T_5'(q)\\ T_6'(q)\\ T_7'(q)\\ T_8'(q)\\ T_9'(q)\\ T_{10}'(q) \end{pmatrix} = \begin{pmatrix} -y_0 & 0 & 0 & 0 & 0 & 0 & m_5 & m_2 & m_1 & m_3 & m_4\\ 2\upsilon_2 c & -\upsilon_2 c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 2\upsilon_1 c & 0 & -2\upsilon_1 c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 3\upsilon_3 c & 0 & 0 & -2\upsilon_3 c & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 2\upsilon_3 c & -\upsilon_3 c & 0 & 0 & 0 & 0 & 0 & 0\\ 2\upsilon_4 c & 0 & 0 & 0 & 0 & -\upsilon_4 c & 0 & 0 & 0 & 0 & 0\\ \upsilon_5 c & 0 & 0 & 0 & 0 & 0 & -m_5 & 0 & 0 & 0 & 0\\ 0 & \upsilon_2 c & 0 & 0 & 0 & 0 & 0 & -m_2 & 0 & 0 & 0\\ 0 & 0 & \upsilon_1 c & 0 & 0 & 0 & 0 & 0 & -m_1 & 0 & 0\\ 0 & 0 & 0 & 0 & \upsilon_3 c & 0 & 0 & 0 & 0 & -m_3 & 0\\ 0 & 0 & 0 & 0 & 0 & \upsilon_4 c & 0 & 0 & 0 & 0 & -m_4 \end{pmatrix} \begin{pmatrix} T_0(q)\\ T_1(q)\\ T_2(q)\\ T_3(q)\\ T_4(q)\\ T_5(q)\\ T_6(q)\\ T_7(q)\\ T_8(q)\\ T_9(q)\\ T_{10}(q) \end{pmatrix}, \tag{2}$$

where $y_0=(2\upsilon_1+2\upsilon_2+3\upsilon_3+2\upsilon_4+\upsilon_5)c$.

In steady state, the derivatives of *Eq. (2)* are set to zero and so that *Eq. (2)* is now

$$\begin{pmatrix} -y_0 & 0 & 0 & 0 & 0 & 0 & m_5 & m_2 & m_1 & m_3 & m_4 \\ 2\upsilon_2 c & -\upsilon_2 c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\upsilon_1 c & 0 & -2\upsilon_1 c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3\upsilon_3 c & 0 & 0 & -2\upsilon_3 c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\upsilon_3 c & -\upsilon_3 c & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\upsilon_4 c & 0 & 0 & 0 & 0 & -\upsilon_4 c & 0 & 0 & 0 & 0 & 0 \\ \upsilon_5 c & 0 & 0 & 0 & 0 & 0 & -m_5 & 0 & 0 & 0 & 0 \\ 0 & \upsilon_2 c & 0 & 0 & 0 & 0 & 0 & -m_2 & 0 & 0 & 0 \\ 0 & 0 & \upsilon_1 c & 0 & 0 & 0 & 0 & 0 & -m_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \upsilon_3 c & 0 & 0 & 0 & 0 & -m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \upsilon_4 c & 0 & 0 & 0 & 0 & -m_4 \end{pmatrix} \begin{pmatrix} T_0(q) \\ T_1(q) \\ T_2(q) \\ T_3(q) \\ T_4(q) \\ T_5(q) \\ T_6(q) \\ T_7(q) \\ T_8(q) \\ T_9(q) \\ T_{10}(q) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad \textbf{(3)}$$

The normalizing condition for this problem is

$$T_0(\infty) + T_1(\infty) + T_2(\infty) + T_3(\infty) + \ldots + T_{10}(\infty) = 1. \quad \textbf{(4)}$$

The steady-state availability of the system, which is the sum of probabilities of operation states, is given by

$$A_q(\infty) = T_0(\infty) + T_1(\infty) + T_2(\infty) + T_3(\infty) + T_4(\infty) + T_5(\infty). \quad \textbf{(5)}$$

The system of linear equations in *Eq. (3)* with the help of *Eq. (4)* is solved using the MATLAB package to obtain the state probabilities below

$$T_0(\infty) = \frac{2m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_1(\infty) = \frac{4m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_2(\infty) = \frac{4m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_3(\infty) = \frac{3m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_4(\infty) = \frac{6m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_5(\infty) = \frac{4m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_6(\infty) = \frac{2m_1m_2m_3m_4\upsilon_5}{m_1m_2m_3m_4(23m_5 + v_5c) + 2cm_1m_2m_5(2m_3v_4 + 3m_4v_3) + 4cm_3m_4m_5(m_1v_2 + m_2v_1)}.$$

$$T_7(\infty)=\frac{4m_1m_3m_4m_5\upsilon_2}{m_1m_2m_3m_4(23m_5+v_5c)+2cm_1m_2m_5(2m_3v_4+3m_4v_3)+4cm_3m_4m_5(m_1v_2+m_2v_1)}.$$

$$T_8(\infty)=\frac{4m_2m_3m_4m_5\upsilon_1}{m_1m_2m_3m_4(23m_5+v_5c)+2cm_1m_2m_5(2m_3v_4+3m_4v_3)+4cm_3m_4m_5(m_1v_2+m_2v_1)}.$$

$$T_9(\infty)=\frac{6m_1m_2m_4m_5\upsilon_3}{m_1m_2m_3m_4(23m_5+v_5c)+2cm_1m_2m_5(2m_3v_4+3m_4v_3)+4cm_3m_4m_5(m_1v_2+m_2v_1)}.$$

$$T_{10}(\infty)=\frac{4m_1m_2m_3m_5\upsilon_4}{m_1m_2m_3m_4(23m_5+v_5c)+2cm_1m_2m_5(2m_3v_4+3m_4v_3)+4cm_3m_4m_5(m_1v_2+m_2v_1)}.$$

The explicit availability expressions in *Eq. (5)* are now

$$A_q=\frac{23m_1m_2m_3m_4m_5}{m_1m_2m_3m_4(23m_5+v_5c)+2cm_1m_2m_5(2m_3v_4+3m_4v_3)+4cm_3m_4m_5(m_1v_2+m_2v_1)}.$$

4 | Results and Discussion

This section aims to conduct numerical experiments to observe the impact of system parameter variations on availability using the MATLAB package. The results are presented in tables and figures. Consistent parameter values are employed throughout the experiments: v1=0.2, v2=0.2, v3=0.1, v4=0.2, v5=0.3, m1=0.6, m2=0.7, m3=0.8, m4=0.8, m5=0.5, and c=0.1.

Table 2. Availability of fog model with respect to v_1 for different values of m_1.

	Availability				
	m_1				
v_1	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
0	0.9055	0.9055	0.9055	0.9055	0.9055
0.2	0.8915	0.8984	0.9008	0.9020	0.9027
0.4	0.8779	0.8915	0.8961	0.8984	0.8998
0.6	0.8647	0.8846	0.8915	0.8949	0.8970
0.8	0.8519	0.8779	0.8869	0.8915	0.8942

Table 3. Availability of fog model with respect to v_2 for different values of m_2.

	Availability				
	m_2				
v_2	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
0	0.9176	0.9176	0.9176	0.9176	0.9176
0.2	0.9031	0.9103	0.9127	0.9139	0.9146
0.4	0.8892	0.9031	0.9079	0.9103	0.9117
0.6	0.8756	0.8961	0.9031	0.9067	0.9089
0.8	0.8625	0.8892	0.8984	0.9031	0.9060

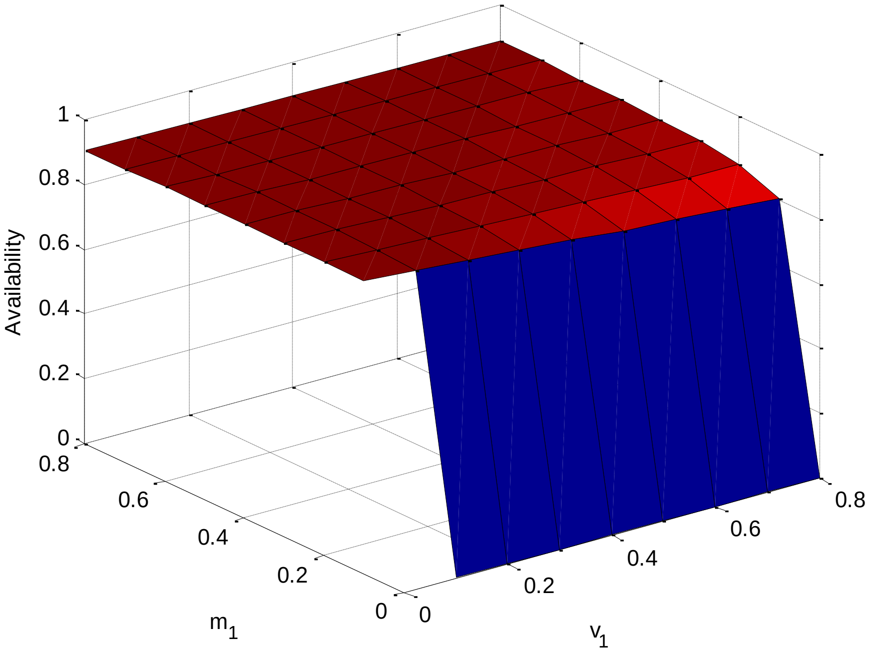

Fig. 2. Availability of fog model with respect to v_1 and m_1.

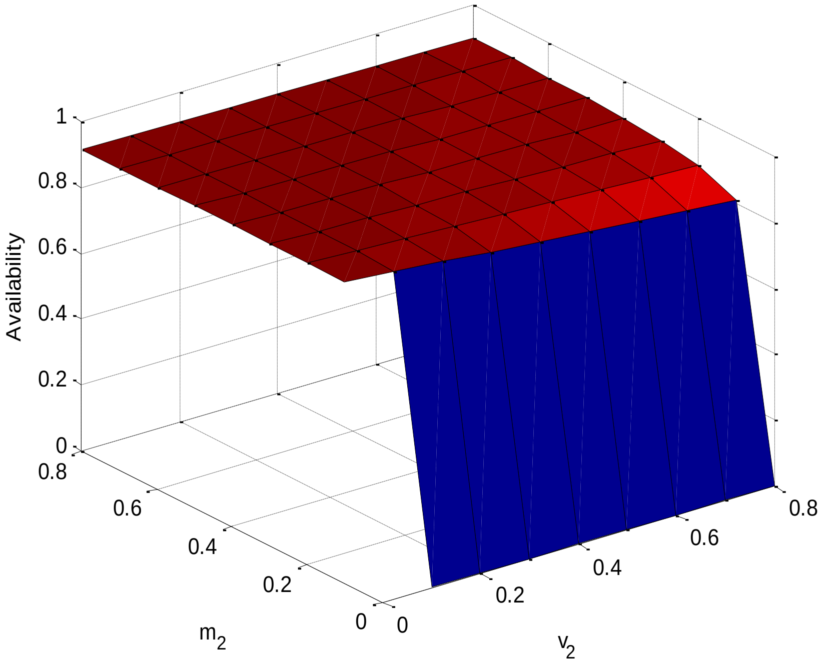

Fig. 3. Availability of fog model with respect to v_2 and m_2.

Table 4. Availability of fog model with respect to v_3 for different values of m_3.

	Availability				
	m_3				
v_3	0.2	0.4	0.6	0.8	1.0
0	0.9357	0.9357	0.9357	0.9357	0.9357
0.2	0.9134	0.9244	0.9281	0.9300	0.9311
0.4	0.8921	0.9134	0.9207	0.9244	0.9266
0.6	0.8718	0.9026	0.9134	0.9189	0.9222
0.8	0.8525	0.8921	0.9062	0.9134	0.9178

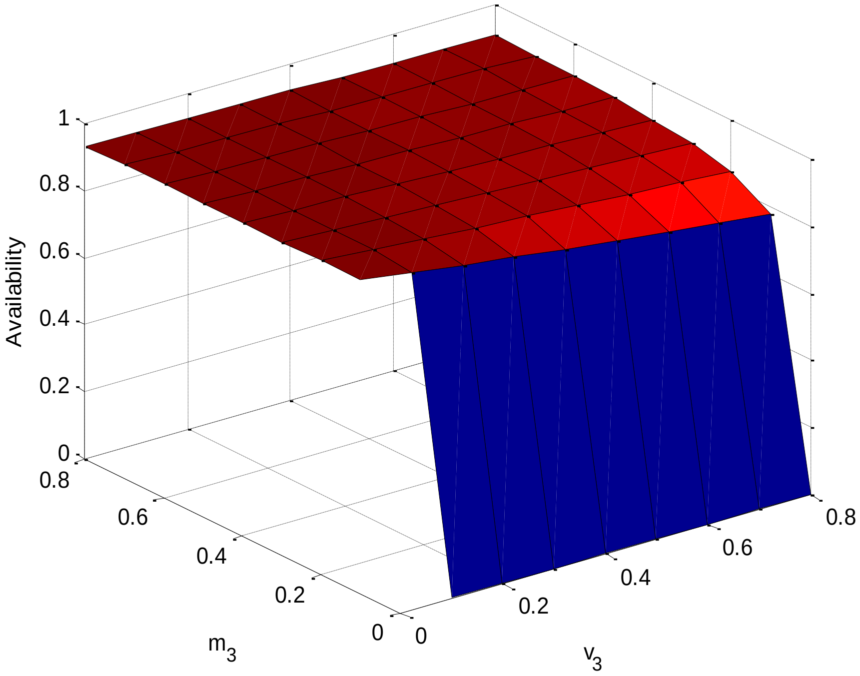

Fig. 4. Availability of fog model with respect to v_3 and m_3.

Table 5. Availability of fog model with respect to v_4 for different values of m_4.

	Availability				
	m_4				
v_4	0.2	0.4	0.6	0.8	1.0
0	0.9641	0.9641	0.9641	0.9641	0.9641
0.2	0.9482	0.9561	0.9588	0.9601	0.9609
0.4	0.9328	0.9482	0.9535	0.9561	0.9577
0.6	0.9179	0.9405	0.9482	0.9521	0.9545
0.8	0.9035	0.9328	0.9430	0.9482	0.9514

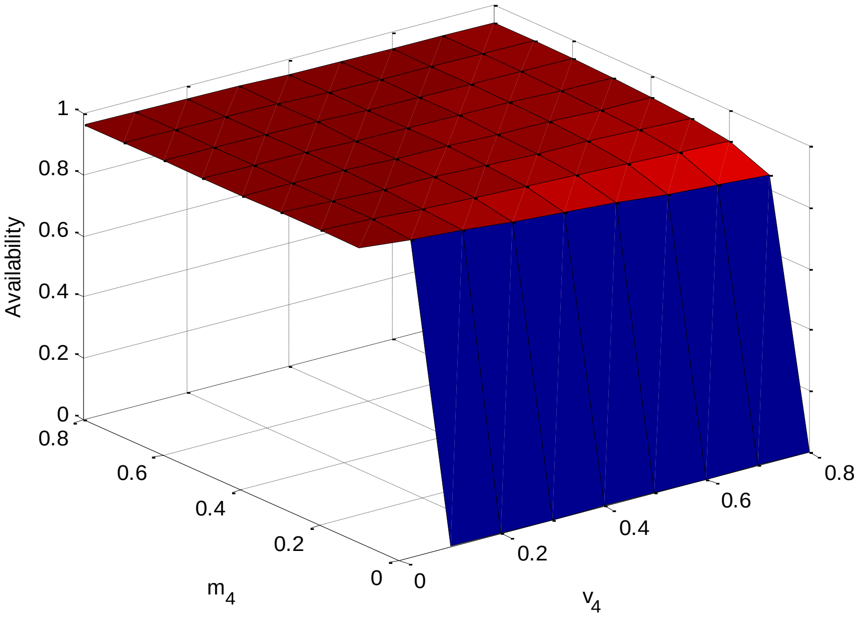

Fig. 5. Availability of fog model with respect to v_4 and m_4.

Table 6. Availability of fog model with respect to v_5 for different values of m_5.

v_5	Availability m_5 0.2	0.4	0.6	0.8	1.0
0	0.9820	0.9820	0.9820	0.9820	0.9820
0.2	0.9736	0.9778	0.9792	0.9799	0.9803
0.4	0.9655	0.9736	0.9764	0.9778	0.9786
0.6	0.9574	0.9695	0.9736	0.9757	0.9770
0.8	0.9455	0.9655	0.9709	0.9736	0.9753

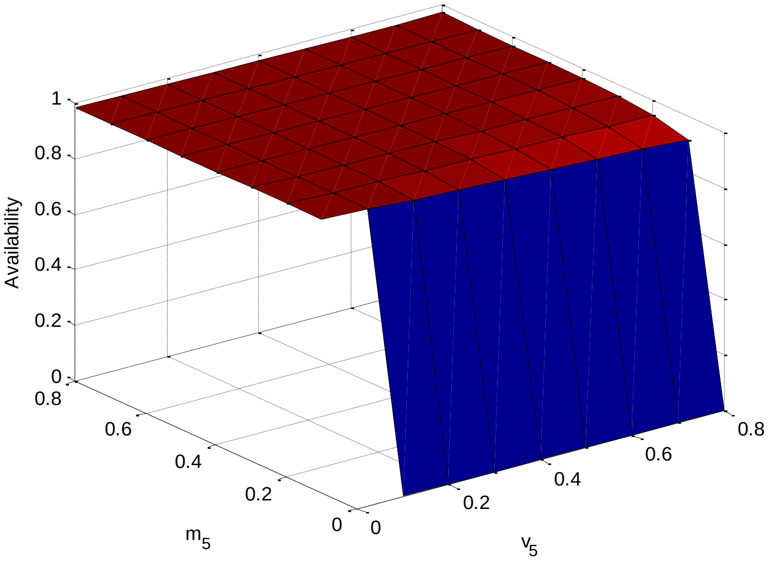

Fig. 6. Availability of fog model with respect to v_5 and m_5.

Table 2 and *Fig. 2* provide a detailed breakdown of how variations in client failure rates and repair rates influence the overall availability of the system in a client-server fog computing network. This information is valuable for understanding the dynamics of system reliability and making informed decisions to enhance network performance. The table and figure comprehensively represent the intricate relationship between client failure, repair rates, and their collective influence on the system availability within a client-server fog computing network. The table and figure are pivotal in unravelling the nuanced dynamics of system reliability under varying conditions, shedding light on how specific parameters contribute to the overall availability of the network. The table and figure also illuminate a crucial aspect of the study, as it meticulously explores the ramifications of increasing client failure rates on the system's availability. Spanning a spectrum from 0 to 0.8, the client failure rates are systematically examined, providing a detailed insight into the system's behaviour under different stress levels. Notably, as the client failure rate escalates across this range, the system's availability exhibits a discernible and consistent decline.

This decline is observed across a diverse set of scenarios, each characterized by varying values of the client's repair rate, adjusted between 0.2 and 1. The findings on the left side of the table underscore a negative correlation between client failure rates and system availability, signifying the system's vulnerability to disruptions caused by increased client failures. The table and figure encapsulate a wealth of information on how variations in client failure rates and repair rates intricately shape the landscape of system availability within the context of a client-server fog computing network. The nuanced insights gleaned from this table contribute to the academic understanding of network dynamics and hold practical implications for system administrators and network architects seeking to optimize the performance and dependability of fog computing environments.

Table 3 and *Fig. 3* present a comprehensive analysis of the impact of v2, denoting the failure rate of the initial level load balancer, on the overall availability of the system across a spectrum of m2 values, representing the repair rate of the same load balancer. The findings unmistakably demonstrate a consistent trend: as the failure rate escalates from 0 to 0.8, irrespective of the repair rate variation, there is a noticeable decline in the system's availability. This insight underscores the critical relationship between failure rates and system reliability, providing valuable insights for system optimization and maintenance strategies. The surface plot depicted in *Fig. 3* vividly illustrates the intricate interplay between the system's availability and both the failure and repair rates of the first-level load balancer. As evidenced by the figure, there exists a clear correlation. As the repair rates escalate, there is a corresponding uptick in the system's availability, whereas an increase in the failure rate correlates with a decline in system availability. This graphical representation effectively captures the dynamic relationship between these variables, providing valuable insights into the factors influencing system reliability and highlighting the critical importance of maintaining optimal failure and repair rates for sustaining high availability levels.

Table 4 and *Fig. 4* display the effect of the failure rate of fog nodes on system availability for different repair rates (m3) of fog nodes. The data presented in the table unequivocally indicates a trend wherein the system's availability experiences a decline corresponding to an increase in the failure rate of fog nodes. Notably, upon analyzing the varying values of fog nodes' repair rate (m3), a conspicuous pattern emerges: system availability peaks when the repair rate attains the value of 0.9357. This observation underscores the critical role of maintenance and repair operations in sustaining system availability, with optimal repair rates evidently contributing to enhanced operational reliability and performance. The surface plot depicted in *Fig. 4* provides a comprehensive visualization of the interplay between two critical parameters: the failure rate v3 and the repair rate m3 of fog nodes and their collective impact on system availability within the fog computing environment. Discerning trends emerge through a meticulous examination of this plot, elucidating the intricate relationship between these variables and system availability.

One salient observation gleaned from *Fig. 4* is the discernible correlation between the repair rate m3 of fog nodes and system availability. As the repair rate escalates, there is a conspicuous increase in system availability. This positive correlation underscores the crucial role of efficient repair and maintenance operations in enhancing the fog computing system's overall reliability and operational uptime. Essentially, higher repair rates are associated with greater resilience against failures, thereby bolstering system availability. The surface plot also reveals a contrasting trend concerning the failure rate v3 of fog nodes. Notably, as the failure rate of fog nodes escalates, there is a discernible decrease in system availability. This negative correlation underscores the detrimental impact of increased failure rates on the operational reliability and availability of the fog computing system. Such observations underscore the importance of mitigating failure risks through proactive measures, including robust fault-tolerance mechanisms and efficient maintenance strategies.

Table 5 and *Fig. 5* meticulously illustrate the influence of the failure rate of the second-level load balancer on system availability across various repair rates of the same load balancer. A clear trend emerges through a comprehensive examination of the data presented in the table, indicating a decline in system availability as the failure rate of the load balancer increases. This trend underscores the system's vulnerability to load balancer failures and emphasizes the criticality of maintaining optimal performance in these components. Significantly, upon scrutinizing the diverse values of the load balancer repair rate (m3), a conspicuous pattern surfaces: system availability peaks when the repair rate attains the value of 0.9641. This observation underscores the pivotal role of maintenance and repair operations in preserving system availability. It accentuates the importance of strategically optimizing repair rates to ensure enhanced operational reliability and performance within the system. These contrasting observations underscore the intricate dynamics within the system architecture, where the effectiveness of maintenance and repair operations significantly influences system availability.

By strategically optimizing repair rates and implementing robust maintenance protocols, organizations can mitigate the impact of failures and ensure consistent operational reliability, thereby facilitating seamless service delivery and enhancing overall system performance. One notable observation from *Fig. 5* highlights the correlation between the repair rate m4 of the second-level load balancer and system availability. As the repair rate increases, a conspicuous uptick in system availability becomes apparent. This positive correlation underscores the pivotal role of efficient repair and maintenance operations in bolstering the fog computing system's overall reliability and operational continuity. Elevated repair rates indicate heightened resilience against failures, thereby fortifying system availability and minimizing potential downtimes.

Conversely, the surface plot also unveils a contrasting trend concerning the failure rate v4 of the second-level load balancer. Notably, with the load balancer's failure rate escalating, there is a discernible decline in system availability. This negative correlation underscores the adverse impact of heightened failure rates on the operational dependability and availability of the fog computing system. Such observations underscore the imperative of proactively mitigating failure risks by implementing robust fault-tolerance mechanisms and adopting efficient maintenance strategies. The insights from *Fig. 5* emphasize the criticality of striking a delicate balance between repair and failure rates within the fog computing ecosystem. Organizations can cultivate a resilient infrastructure capable of sustaining optimal operational performance and ensuring seamless service delivery in dynamic and demanding computing environments by prioritizing effective maintenance practices and diligently addressing potential failure points.

Table 6 and *Fig. 6* meticulously illustrate the influence of the failure rate of the server on system availability across various repair rates of the same server. A clear trend emerges through a comprehensive examination of the data presented in the table, indicating a decline in system availability as the failure rate of the server increases. This trend underscores the system's vulnerability to server failures and emphasizes maintaining optimal performance in these components. Upon scrutinizing the diverse values of the server repair rate (m5), the discernment of a conspicuous pattern is of particular significance: system availability peaks when the repair rate attains the value of 0.9820. This observation underscores the pivotal role of maintenance and repair operations in preserving system availability. It accentuates the importance of strategically optimizing repair rates to ensure enhanced operational reliability and performance within the system.

In contrast, *Table 5* offers a different perspective. It delineates the impact of the repair rate of the server on system availability. Contrary to the diminishing availability associated with escalating failure rates, this table reveals a distinct pattern: an augmentation in system availability corresponding to an increase in the repair rate of the server. Essentially, higher repair rates are observed to bolster system reliability and availability positively, thereby offsetting the adverse effects of server failures.

These insights underscore the intricate dynamics within the system architecture, wherein effective maintenance and repair operations significantly influence system availability. By strategically optimizing repair rates and implementing robust maintenance protocols, organizations can mitigate the impact of failures and ensure consistent operational reliability, thereby facilitating seamless service delivery and enhancing overall system performance. One noteworthy observation from *Fig. 6* underscores the evident correlation between the server's repair rate (m5) and system availability. As the repair rate increases, a discernible enhancement in system availability becomes pronounced. This positive correlation accentuates the pivotal role of efficient repair and maintenance operations in fortifying the fog computing system's overall reliability and operational continuity. Elevated repair rates indicate heightened resilience against failures, bolstering system availability and mitigating potential downtimes.

Conversely, the surface plot also unveils a contrasting trend pertaining to the server's failure rate (v5). Notably, as the server's failure rate escalates, system availability has a discernible downturn. This negative correlation underscores the adverse ramifications of elevated failure rates on the operational dependability and availability of the fog computing system. Such observations highlight the imperative of proactively addressing failure risks by implementing robust fault-tolerance mechanisms and adopting efficient maintenance strategies.

5 | Conclusion

The present study constitutes a thorough exploration of distributed system availability within the realm of fog computing, with a primary emphasis on the influence of fault tolerance factors. Within the architectural framework, we introduce five distinct subsystems denoted as A, B, C, D, and E, each endowed with specific functions crucial for ensuring the seamless operation of the network. These subsystems, spanning clients, fog nodes, servers, first-level load balancers, and second-level load balancers, synergistically contribute to the efficient distribution and management of computational tasks. Central to our investigation is the modelling of the network's availability dynamics and the computation of its long-term availability. Through rigorous numerical experiments meticulously documented via tables and graphs, our study unveils the intricate interplay between various network parameters and the optimization of system availability. The key findings underscore the pivotal roles played by load balancers and fog nodes, particularly when arranged in a series-parallel configuration are of particular significance. This specific setup not only facilitates optimal task distribution but also markedly enhances the overall system availability. Indeed, the strategic integration of fog nodes and load balancers in a series-parallel configuration emerges as a cornerstone approach in fortifying system availability within client-server fog computing networks. The insights gleaned from our study hold substantial value for diverse stakeholders, including researchers, system administrators, and network architects, all seeking to elevate the performance and reliability of fog computing environments. By leveraging the findings presented herein, stakeholders are empowered to devise informed strategies aimed at optimizing system availability and reinforcing the resilience of fog computing infrastructures to address the ever-evolving computational demands of modern applications effectively.

Author Contributions

Ibrahim Yusuf, Khadija Salihu Auta, and Muhammad Kabeer jointly contributed to the research design, model development, and manuscript writing. Ibrahim Yusuf led the conceptualization and modeling of the stochastic system and its foggy Markovian paths, while Khadija Salihu Auta and Muhammad Kabeer contributed to the computational experiments and analysis.

Funding

No specific funding was received for this study from any funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: a taxonomy, survey and future directions. In *Internet of everything: algorithms, methodologies, technologies and perspectives* (pp. 103–130). Springer. https://doi.org/10.1007/978-981-10-5861-5_5

[2] Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2017). A comprehensive survey on fog computing: state-of-the-art and research challenges. *IEEE communications surveys & tutorials*, *20*(1), 416–464. DOI:10.1109/COMST.2017.2771153

[3] Yi, S., Li, C., & Li, Q. (2015). *A survey of fog computing: concepts, applications and issues* [presentation]. Proceedings of the 2015 workshop on mobile big data (pp. 37–42).

[4] Zhang, P. Y., Zhou, M. C., & Fortino, G. (2018). Security and trust issues in Fog computing: a survey. *Future generation computer systems*, *88*, 16–27. DOI:10.1016/j.future.2018.05.008

[5] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., …& Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: a complete survey. *Journal of systems architecture*, *98*, 289–330. DOI:10.1016/j.sysarc.2019.02.009

[6] Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2016). An overview of Fog computing and its security issues. *Concurrency and computation: practice and experience*, *28*(10), 2991–3005. DOI:10.1002/cpe.3485

[7] Alli, A. A., & Alam, M. M. (2020). The fog cloud of things: a survey on concepts, architecture, standards, tools, and applications. *Internet of things (Netherlands)*, *9*, 100177. DOI:10.1016/j.iot.2020.100177

[8] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). *Fog computing and its role in the internet of things* [presentation]. Proceedings of the first edition of the mcc workshop on mobile cloud computing (pp. 13–16). https://doi.org/10.1145/2342509.234251

[9] Caiza, G., Saeteros, M., Oñate, W., & Garcia, M. V. (2020). Fog computing at industrial level, architecture, latency, energy, and security: a review. *Heliyon*, *6*(4). DOI:10.1016/j.heliyon.2020.e03706

[10] Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of network and computer applications*, *98*, 27–42. DOI:10.1016/j.jnca.2017.09.002

[11] Yusuf, I., & Auta, A. A. (2021). Availability analysis of a distributed system with homogeneity in client and server under four different maintenance options. *Life cycle reliability and safety engineering*, *10*(4), 355–371. DOI:10.1007/s41872-021-00177-w

[12] Melnik, E. V., Klimenko, A. B., & Ivanov, D. Y. (2018). Fog-computing concept usage as means to enhance information and control system reliability. *Journal of physics: conference series* (Vol. 1015, p. 32175). IOP Publishing. DOI: 10.1088/1742-6596/1015/3/032175

[13] Pereira, J., Ricardo, L., Luís, M., Senna, C., & Sargento, S. (2019). Assessing the reliability of fog computing for smart mobility applications in VANETs. *Future generation computer systems*, *94*, 317–332. DOI:10.1016/j.future.2018.11.043

[14] Al-khafajiy, M., Baker, T., Al-Libawy, H., Maamar, Z., Aloqaily, M., & Jararweh, Y. (2019). Improving fog computing performance via Fog-2-Fog collaboration. *Future generation computer systems*, *100*, 266–280. DOI:10.1016/j.future.2019.05.015

[15] Chen, Y. F., Huang, D. H., Huang, C. F., & Lin, Y. K. (2020). *Reliability evaluation for a cloud computer network with fog computing* [presentation]. 2020 IEEE 20th international conference on software quality, reliability and security companion (QRS-C) (pp. 682–683). 10.1109/QRS-C51114.2020.00119

[16] Montoya-Munoz, A. I., & Rendon, O. M. C. (2020). An approach based on fog computing for providing reliability in iot data collection: a case study in a colombian coffee smart farm. *Applied sciences (Switzerland)*, *10*(24), 1–16. DOI:10.3390/app10248904

[17] Kabeer, M., Yusuf, I., & Sufi, N. A. (2023). Distributed software defined network-based fog to fog collaboration scheme. *Parallel computing*, *117*, 103040. DOI:10.1016/j.parco.2023.103040

[18] Das, R., & Inuwa, M. M. (2023). A review on fog computing: Issues, characteristics, challenges, and potential applications. *Telematics and informatics reports*, *10*, 100049. DOI:10.1016/j.teler.2023.100049