A Fault-Tolerant Queuing Model for Delay-Sensitive Applications in Fog Computing

Hibat Eallah Mohtadi, Mohamed Hanini, Amine Benmakhlouf, Abdelkrim Haqiq

***Abstract*—In fog computing systems, fog nodes frequently experience state changes that impact their reliability. Maintaining reliability and low latency in face of dynamic workloads and node failures is essential for high-quality service delivery. This paper introduces a fault-tolerant queuing model for fog computing environments, designed to optimize task offloading by applying queuing theory to analyze key performance metrics. Our model evaluates system reliability by examining fog nodes in active and standby states, accounting for parameters such as task arrival rates, standby duration rates, and service times. By calculating steady-state probabilities and other metrics, we assess mean delay, failure probability, and overall system resilience. Numerical results demonstrate that our approach could be used to reduces delays and improves fault tolerance, even under high node failure rates. The proposed model enhances adaptability and reliability, making it well-suited for delay-sensitive applications within fog networks. This research suggests that tuning standby durations can optimize task handling, enhancing fault tolerance and Quality of Service (QoS) in dynamic fog computing environments.**

***Index Terms*—Fog computing, Queuing theory, Failure, Fault tolerance, Reliability.**

I. Introduction

Fog Computing (FC) brings together near-end user edge devices, storage, communication, and computing resources for deployment in processing, supervision, configuration, measurement, and management tasks. From a certain viewpoint, FC can be seen as an extension of both cloud and edge computing [1]. It offers unique characteristics such as low latency and location awareness through wide geographical distribution, enhancing QoS by enabling swift, responsive interactions and customized services based on user preferences. Additionally, FC ensures mobility support for IoT devices, enabling immediate responses, which is crucial for applications like smart cities that require real-time feedback [2]. [3] proposes an edge-enabled virtual honeypot-based intrusion detection system for securing Vehicle-to-Everything (V2X) networks. By utilizing machine learning techniques, their approach enhances V2X security by detecting and mitigating potential threats in real time, ensuring reliable communication in dynamic vehicular environments. By facilitating data processing at the network edge, FC effectively addresses bandwidth limitations and network congestion, offering improved scalability, security, and energy efficiency. This makes FC valuable for managing the large volumes of data generated by IoT devices, contributing to more efficient and reliable urban infrastructures. Similarly, healthcare benefits from local data analysis, reducing delays and energy consumption, while autonomous vehicles leverage Vehicle-to-Vehicle (V2V) communication for minimal latency and high performance [4]. To support these applications, FC is designed with a multi-layered architecture that includes the cloud layer, the fog layer, and the device layer. The device layer comprises IoT devices that generate data, while the fog layer consists of fog nodes responsible for processing and storing this data. The cloud layer offers supplementary computing resources and storage as needed [5]. Task offloading in FC refers to transferring computational tasks from end devices to fog nodes or the cloud, leveraging their processing capabilities and optimizing resource utilization [6]. While this process and given the benefits of fog nodes in delivering fast computing, storage, and networking services, they remain vulnerable to various failures that can impact embedded fog nodes and disrupt service continuity. Downtime caused by these failures can lead to significant loss and damage service providers reputations. Failures can stem from hardware issues like environmental disruptions or internal malfunctions, software errors such as corrupted data or configuration problems, or connectivity issues from link failures [7]. FC faces several significant challenges that can affect service availability. Key issues include network failure due to the distributed nature of fog nodes, high latency caused by poor resource management. [8] proposes a hybrid genetic algorithm to optimize task offloading in edge-cloud environments, focusing on reducing latency and energy consumption while balancing computational loads. [9] presents a queuing theory approach to task scheduling in cloud computing, using a generalized processor-sharing queue model under heavy traffic approximation. Their work highlights how analytical models can effectively optimize resource allocation and minimize delays, providing valuable insights for task scheduling in distributed systems like fog computing. Additionally, heterogeneity in devices and limited computational resources in fog nodes create difficulties in balancing task loads efficiently. Ensuring reliable data processing while minimizing delays remains a central challenge [10].

Most current studies in fog computing focus heavily on resource management and service provisioning, assuming continuous availability of the physical infrastructure. However, this assumption is unrealistic due to the frequent occurrence of network failures and disruptions in infrastructure. Acknowledging these potential failures is crucial for developing robust, fault-tolerant systems that can maintain service quality even when parts of the network are inaccessible.

Manuscript received November 19, 2024; revised February 7, 2025.

Hibat Eallah Mohtadi is a PhD student of Hassan First University of Settat, Faculty of Sciences and Techniques, B.P.577, Settat 26000, Morocco. (e-mail: h.mohtadi@uhp.ac.ma).

Mohamed Hanini is a professor of Hassan First University of Settat, Faculty of Sciences and Techniques, B.P.577, Settat 26000, Morocco. (e-mail: mohamed.hanini@uhp.ac.ma).

Amine Benmakhlouf is a professor of Hassan First University of Settat, Faculty of Sciences and Techniques, B.P.577, Settat 26000, Morocco. (e-mail: Amine.benmakhlouf@uhp.ac.ma).

Abdelkrim Haqiq is a professor of Hassan First University of Settat, Faculty of Sciences and Techniques, B.P.577, Settat 26000, Morocco. (e-mail: abdelkrim.haqiq@uhp.ac.ma).

Effective strategies must account for infrastructure instability to ensure reliable service delivery in real-world environments [11]. The main contributions of this paper are summarized as follows:

- **Fault-Tolerant Queuing Model for Fog Computing**: A queuing-theoretic model is developed to analyze the fault tolerance of a multi-node FC system. Each fog node is modeled as an M/M/1 queue with active and standby states, enabling an evaluation of system resilience under different configurations and failure scenarios.
- **Optimization of Fog Node Configuration for Fault Tolerance**: We investigate how the number of fog nodes impacts total delay and the probability of nodes being in standby and in active states and the mean number of tasks providing insights into selecting a node configuration that balances performance with system reliability in fault-prone environments.
- **Analytical Evaluation of Key Fault Tolerance Metrics**: Key performance metrics, including total system delay and standby probability, are derived analytically to assess the trade-offs between delay, fault tolerance, and system availability in fog computing networks.
- **Numerical Simulations to Validate Fault Tolerance**: Numerical simulations are conducted to validate the theoretical model, showing how the proposed approach enhances fault tolerance by maintaining service continuity and reducing delay under varying node configurations.

The remainder of this paper is structured as follows: Section II reviews related work, providing context for fault tolerance in fog computing. In Section III, we describe the system model and problem formulation in detail. Section IV presents the resolution of the proposed queuing model, evaluating system reliability and performance. Section V provides numerical results that validate our model, and finally, Section VI concludes the paper with a summary of findings and suggestions for future research on enhancing fault tolerance in FC systems.

II. Related Work

A. Failures and service degradation in fog computing networks

Failures and the degradation of service rates are significant concerns in FC networks, where the distributed and resource-constrained nature of fog nodes exposes them to frequent operational challenges. Studies have shown that these networks face various types of failures, such as hardware malfunctions, connectivity disruptions, and computational overloads, which degrade service quality and user experience [12] [13]. Issues like limited bandwidth, intermittent connectivity, and resource exhaustion further increase the likelihood of service degradation, particularly in time-sensitive applications such as real-time data processing [14]. Network topology and node placement directly impact system resilience, highlighting how strategic configurations can mitigate some failure risks.

The heterogeneity and geographical dispersion of fog nodes also contribute to frequent service interruptions. As [15] notes, even minor delays in one segment of the network can propagate, leading to cascading failures that severely impact overall performance. [16] emphasizes the critical issue of failure risks in heterogeneous fog environments, underlining how these vulnerabilities can significantly degrade service quality and reliability. This study draws attention to the inherent challenges in maintaining continuous, high-quality service within FC networks due to potential failures across diverse computational resources.

In environments like cluster-based wireless sensor networks (WSNs) that employ Free-Space Optical (FSO) technology, external factors further complicate reliability. [17] discusses the challenges of maintaining consistent performance in WSNs, emphasizing that environmental factors such as fog and rain can severely degrade communication reliability, making robust cluster head localization critical to network stability.

Moreover, [18] underscores the challenges of sustaining service continuity in fog networks following infrastructure faults. The authors highlight the risks that faults pose to service function chains, stressing the need for resilience to maintain uninterrupted functionality and mitigate degradation in service quality.

Typically, failures in fog computing environments can occur for variety of reasons:

Hardware degradation: Partial hardware failures, such as malfunctioning processor cores or degraded storage performance, can lead to reduced processing capacity. In such cases, fog nodes may continue to function but with diminished capabilities. This scenario is discussed in the context of fault tolerance in FC, where systems are designed to handle hardware failures gracefully[19].

Network connectivity issues: Limited or unstable network connectivity can result in a lower effective service rate for fog nodes. Despite connectivity challenges, fog nodes may still process tasks locally, albeit with reduced efficiency. This situation is addressed in studies focusing on service placement and resource management in FC environments [20].

Battery-related failures: In battery-powered fog nodes, a drop in battery performance can cause the node to reduce computational power to conserve energy while still handling tasks at a reduced rate. This scenario is examined in studies on energy efficiency and resource management in FC[21].

These findings collectively underscore the importance of addressing not only the root causes of failure but also the system's ability to manage degradation effectively, ensuring consistent service quality in FC networks.

B. Fault tolerance strategies in fog computing networks

Fault tolerance is a critical mechanism in fog and cloud computing, enabling systems to continue processing in the event of hardware or software failures. Due to frequent node failures caused by hardware malfunctions, network disruptions, and unpredictable node availability, fault tolerance is essential to maintain system reliability under faulty conditions. Strategies such as redundancy, checkpointing, load balancing, and failure prediction have been proposed to address these challenges, enhancing the system's ability to sustain resource availability despite disruptions.

In FC, various studies underscore the significance of fault tolerance. [22] proposes a fault-tolerant framework for Social Internet of Things (SIoT) systems, integrating dynamic task offloading, redundancy, and replication to ensure continuity during fog node disruptions. Their approach predicts failures and reroutes tasks proactively, minimizing service interruptions. Similarly, [23] presents a decentralized task allocation method for IoT environments, where tasks are redundantly distributed across nodes to ensure robustness, especially in high-failure, resource-constrained conditions. [24] addresses fault tolerance in edge computing with a selective aggregation method that reroutes tasks among edge nodes to maintain service continuity during failures while preserving data privacy and reducing latency. For stateful applications, [25] examines persistent storage solutions that allow fog applications to recover from failures by retaining critical data, balancing trade-offs in performance, storage, and recovery time.

In cloud environments, fault tolerance is also enhanced through advanced scheduling and redundancy techniques. For example, [26] propose a scheduling method using multi-level queues and LSTM-based workload prediction to dynamically adjust resources, preventing overload and ensuring recovery from node failures. [27] suggests a task-duplication strategy for geo-distributed clouds, replicating tasks across locations to mitigate delays and complete tasks despite network or node failures. In workflows prone to failures, [28] combines replication heuristics with checkpointing, enabling tasks to resume from saved states and minimizing downtime. In IoT-enabled wireless sensor networks (WSNs), [29] proposes an intelligent routing algorithm that dynamically selects optimal routes, rerouting data to avoid malfunctioning nodes, which enhances both reliability and energy efficiency.

Addressing the unique fault tolerance challenges in FC and SIoT environments, [30] introduces an automata-based dynamic scheduling approach tailored for distributed, often unreliable fog resources. This model adapts to task failures in real-time by leveraging automata theory, enabling task adjustments based on current resource availability.

Overall, these studies highlight diverse approaches to achieving fault tolerance across fog, cloud, and IoT networks, all aimed at enhancing service reliability, minimizing downtime, and ensuring data integrity in failure-prone conditions.

Research gap: The gap in existing work on fault tolerance in fog and cloud computing lies in several key areas. Firstly, there is limited focus on dynamic, independent state management of fog nodes, particularly with active-standby transitions that do not rely on shared spares, a feature necessary for geographically dispersed and resource-constrained fog environments. While transient queue analysis and computational techniques are explored, there is a lack of analytical modeling for both transient and steady-state fault tolerance metrics tailored specifically to fog networks. Additionally, studies rarely address how the configuration and number of fog nodes impact critical performance metrics like standby probabilities and overall delay.

The proposed model in this work addresses the identified research gaps by introducing an analytical framework that captures the dynamic behavior of fog nodes transitioning between Active and Standby states without relying on shared spare resources. This independence aligns with the constraints of geographically dispersed and resource-limited fog environments. By leveraging a Quasi-Birth–Death (QBD) process, the model provides a unified analysis of both transient and steady-state fault tolerance metrics, enabling a comprehensive evaluation of the system's short- and long-term behavior. Furthermore, the model integrates critical performance metrics, such as standby and active state probabilities, mean delay, and task accumulation, while explicitly considering the configuration and number of fog nodes. This approach not only enhances the understanding of fault tolerance in fog networks but also guides the design and optimization of these systems to balance resource utilization, delay, and reliability effectively.

III. Model description

In this section, the studied system architecture is first described, including the configuration of fog nodes. Following this, the mathematical model of the system is presented, along with a transient probability analysis to examine state transitions, offering insights into system performance and reliability.

A. System architecture

The architecture of fog computing is structured into three layers 1: the end devices layer, the fog layer, and the cloud layer [1].

The end devices layer at the lowest level, consists of various connected devices such as smartphones, laptops, vehicles, drones, and IoT devices (e.g., wearables and sensors) that generate data requiring processing and potentially real-time responses. These end devices communicate with nearby fog nodes to offload data processing, which reduces latency and bandwidth usage by avoiding direct cloud communication.

The fog layer which acts as an intermediary between the end devices and the cloud. This layer is composed of base stations, local servers, and gateway devices positioned closer to the edge of the network, such as servers deployed in smart cities or cellular towers. These fog nodes perform local data processing, analysis, and storage to provide faster, localized services, reducing the need for cloud-based processing and enabling low-latency responses.

The cloud layer provides centralized storage, extensive analytics, and long-term data processing capabilities. While fog nodes handle immediate and localized data processing, the cloud layer performs complex computations, stores large datasets, and integrates data into broader analytics.

Together, these layers create a hierarchical architecture where data flows from end devices to fog nodes for initial processing, minimizing data transmission to the cloud. This structure enhances efficiency, scalability, and fault tolerance, making FC ideal for IoT and edge applications that require prompt, reliable responses[4].

B. Analytical model for the fog computing nodes

In this model, the effect of fault tolerance in a FC environment is analyzed. The system consists of a dispatcher and N fog nodes operating in parallel. To account for the unavailability of fog nodes, each queue is modeled with

Fig. 1. Studied fog Computing architecture

two states: **standby state** and **active state**. The dispatcher dynamically assigns tasks to each fog node, which are represented as individual $M/M/1$ queues. Each fog node operates in one of two states: an **active state**, where tasks are processed at a standard service rate, and a **standby state**, where tasks are processed at a reduced service rate instead of suspending service entirely. This two-state setup, with transitions between active and standby states based on queue conditions, enhances fault tolerance by ensuring continued task processing even when nodes temporarily enter lower-capacity modes. Task arrivals are governed by a Poisson process, with rates that vary depending on the fog node's state, and the standby durations follow an exponential distribution. This structure provides a resilient model for handling dynamic workload distributions while maintaining continuity and minimizing downtime across the fog network.

The following assumptions and notations are defined for modeling purposes.

- It is assumed that tasks arrive at each fog node according to a Poisson process with a rate that depends on the server's state. The arrival rate is denoted by λ_S when the fog node is in the standby state and by λ_B when it is in the active state.
- The fog node operates at varying service rates. Instead of suspending service during the standby state, it continues to process tasks at a reduced service rate, μ_S, while it operates at the standard service rate, μ_B, in the active state.
- The duration of the standby period follows an exponential distribution with parameter β,

To enhance comprehension, this paper includes a Table I contains descriptions of the symbols used.
Let $N(t)$ defines the number of tasks in the queue at time t, $S(t)$ the state of the fog node at time t.
where

$$S(t) \equiv \begin{cases} 1; & \text{the fog node is on standby state} \\ 2; & \text{the fog node is on active state} \end{cases}$$

Then $\{S(t), N(t); t \geq 0\}$ is a Markov process with the state space

$$\Omega = \{(i, n) \mid i = 1, 2; n = 0, 1, 2, 3, \ldots\}$$

TABLE I
NOTATIONS OF KEY PARAMETERS.

Notations	Definitions
λ_S	The incoming task arrival rate when the server is in the standby state.
λ_B	The incoming task arrival rate when the server is in the active state.
μ_S	The mean service rate of the fog node when the server is in the standby state.
μ_B	The mean service rate of the mobile fog node when the server is in the active state.
β	The standby rate.
$P_1(n)$	The joint steady-state probability that n tasks are in the system during the standby state.
$P_2(n)$	The joint steady-state probability that n tasks are in the system during the active state.
N	The number of fog nodes in the system.
P_{fail}	Failure probability.
$D_{\max}$	The delay threshold.
N_F	The number of failed nodes.

C. Transient probabilities analysis

In this subsection, we analyze the transient probabilities of the FC system, focusing on how the system's state evolves over time under varying operational conditions. Transient probability analysis is essential for understanding the short-term behavior of the system, particularly during transitions between different states, such as from the standby state to the active state. By modeling these probabilities, we can quantify the likelihood of the system being in a specific state at any given time, which helps assess performance

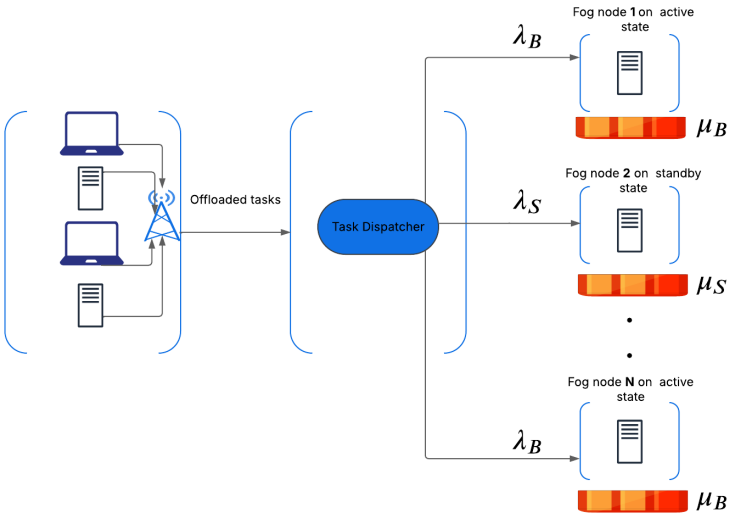

Fig. 2. Studied fog computing queue model

metrics. This analysis provides insights into how efficiently the fog nodes handle dynamic workloads and adapt to fluctuations in task arrivals.

The steady-state probabilities of the system are giving:

When the fog node is in the standby state

$$\lambda_S P_1(0) = \mu_S P_1(1) + \mu_B P_2(1), \quad n = 0 \tag{1}$$

$$(\beta + \lambda_S + \mu_S) P_1(n) = \mu_S P_1(n+1) + \lambda_S P_1(n-1), \quad n \geq 1 \tag{2}$$

When the fog node is in the active state

$$(\lambda_B + \mu_B) P_2(1) = \beta P_1(1) + \mu_B P_2(2) \tag{3}$$

$$(\lambda_B + \mu_B) P_2(n) = \mu_B P_2(n+1) + \lambda_B P_2(n-1) + \beta P_1(n), \quad n \geq 1 \tag{4}$$

The equations (1)–(4) can be written in the following transition rate matrix Q in a block-tridiagonal form:

$$Q = \begin{bmatrix} F_{00} & F_{01} & & & & \\ F_{10} & F & G & & & \\ & E & F & G & & \\ & & E & F & G & \\ & & & \vdots & \vdots & \vdots \end{bmatrix}$$

where $F_{00} = -\lambda_S, F_{01} = (\lambda_S, 0), F_{10} = (\mu_S, \mu_B)^{\mathrm{T}}$ and

$$E = \begin{bmatrix} \mu_S & 0 \\ 0 & \mu_B \end{bmatrix}, \quad G = \begin{bmatrix} \lambda_S & 0 \\ 0 & \lambda_B \end{bmatrix},$$
$$F = \begin{bmatrix} -(\lambda_S + \beta + \mu_S) & \beta \\ 0 & -(\lambda_B + \mu_B) \end{bmatrix}.$$

IV. The Resolution of the Proposed Queuing Model

A. Model Resolution

To analyze the QBD process, it is necessary to solve for the minimal non-negative solution of the matrix quadratic equation

$$R^2 E + RF + G = 0 \tag{5}$$

where R is called rate matrix, the unique non-negative solution with spectral radius less than one of the quadratic equation (5).

Theorem 1. *If $\rho_B = \frac{\lambda_B}{\mu_B} < 1$ the matrix equation (5) has the minimal non-negative solution*

$$R = \begin{bmatrix} r_S & \frac{\beta r_S}{\mu_B(1-r_S)} \\ 0 & \rho_B \end{bmatrix}$$

Where

$$r_S = \frac{1}{2\mu_S}\left(\lambda_S + \beta + \mu_S - \sqrt{(\lambda_S + \beta + \mu_S)^2 - 4\lambda_S\mu_S}\right).$$

With $0 < r_S < 1$

Proof

Since the matrices F, E, G of equation (5) are all upper triangular, so let

$$R = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix}$$

substituting R^2 and R into (5), we get the following set of equations.

$$\mu_S r_{11}^2 - (\lambda_S + \mu_S + \beta) r_{11} + \lambda_S = 0 \tag{6}$$

$$\mu_B (r_{11} r_{12} + r_{12} r_{22}) + \beta r_{11} - (\lambda_B + \mu_B) r_{12} = 0 \tag{7}$$

$$\mu_B r_{22}^2 - (\lambda_B + \mu_B) r_{22} + \lambda_B = 0 \tag{8}$$

Lemma 1. *r_S satisfies the following relationship:*

$$\frac{\lambda_S}{r_S} = \lambda_S + \beta + \mu_S(1 - r_S) \tag{9}$$

equivalently, we have

$$\frac{\lambda_S}{r_S} = \frac{\beta}{1 - r_S} + \mu_S$$

Proof
We divide the equation (6) by r_S, we obtain the equation (9).

Let (S, N) be the stationary limit of the QBD process $\{(S(t), N(t)); t\}$

$$\begin{aligned} p_0 &= p_1(0), \quad p_j = (p_1(j), p_2(j)), \quad j \geqslant 1 \\ p_i(j) &= P\{S = i, N = j\} \\ &= \lim_{t\to\infty} P\{S(t) = i, N(t) = j\} \quad (i,j) \in \Omega. \end{aligned}$$

Theorem 2. *If $\rho_B < 1$, the stationary probability distribution of (N, S) is given by:*

$$p(j) = \begin{cases} Br_S^j, & j \geqslant 0, \\ B\frac{\beta r_S}{\mu_b(1-r_S)} \sum_{i=0}^{j-1} r_S^i \rho_b^{j-1-i}, & j \geqslant 1. \end{cases} \quad (10)$$

where

$$B = (1 - r_S)(1 - \rho_B) \left[1 - \rho_B + \frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)}\right]^{-1}$$

Proof
With the matrix-geometric solution method [31], we have

$$p_j = (p_1(j), p_2(j)) = (p_1(1), p_2(1))\, R^{j-1}, \quad j \geqslant 1 \quad (11)$$

and $(p_1(0), p_1(1), p_2(1))$ satisfies $(p_1(0), p_1(1), p_2(1))\, B[R] = 0$,

$$\begin{aligned} B[R] &= \begin{bmatrix} -\lambda_S & \lambda_S & 0 \\ \mu_{\mathrm{S}} & RE + F & \\ \mu_{\mathrm{B}} & & \end{bmatrix} \\ &= \begin{bmatrix} -\lambda_S & \lambda_S & 0 \\ \mu_{\mathrm{S}} & -\lambda_S - \beta - \mu_{\mathrm{S}}(1 - r_S) & \frac{\beta}{1-r_S} \\ \mu_{\mathrm{B}} & 0 & -\mu_{\mathrm{B}} \end{bmatrix}. \end{aligned}$$

Substituting $B[R]$ into the above equation, we obtain the set of equations

$$\begin{cases} -\lambda_S p_1(0) + \mu_{\mathrm{S}} p_1(1) + \mu_{\mathrm{B}} p_2(1) = 0, \\ \lambda_S p_1(0) - (\lambda_S + \beta + \mu_{\mathrm{S}}(1 - r_S))\, p_1(1) = 0, \\ \dfrac{\beta}{1 - r_S} p_1(1) - \mu_{\mathrm{B}} p_2(1) = 0. \end{cases} \quad (12)$$

Taking $p_1(0) = B$, we get:

$$(p_1(0), p_1(1), p_2(1)) = B\left(1, r_S, \frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)}\right)$$

Substituting $(p_1(1), p_2(1))$ and R^{j-1} into (11), we obtain (10).
We determine the constant factor B by considering the normalization condition. Using equation (10) we obtain the probability that the fog node is in the standby state:

$$P\{i = 1\} = \sum_{j=0}^{\infty} p_1(j) = B_0(1 - \rho_B) \quad (13)$$

The probability that the fog node is on active state:

$$P\{i = 2\} = \sum_{j=1}^{\infty} p_2(j) = B_0 \left[\frac{\beta r_s}{\mu_{\mathrm{B}}(1 - r_s)}\right] \quad (14)$$

where

$$B_0 = \left[1 - \rho_B + \frac{\beta r_s}{\mu_{\mathrm{B}}(1 - r_S)}\right]^{-1}$$

B. Performance parameters

We define several QoS parameters to evaluate the performance of the proposed FC system using the analytical model presented in this study.

1) The mean number of tasks distribution at k^{th} fog node:

Theorem 3. *Let L_{MM1} be the stationary mean number of tasks in a $M/M/1$ queue, where the queue follows a geometric distribution with parameter $1 - \rho_B$ and L_{SV} be the additional mean number of tasks with the standby. If $\rho_B < 1$ and $\mu_{\mathrm{B}} > \mu_{\mathrm{S}}$, Then the queue length L can be decomposed into the sum of two independent random variables: $L = L_{MM1} + L_{SV}$*

$$P\{L_{\mathrm{SV}} = j\} = B^* \left(1 - \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}\right)(1 - r_S) r_S^j, \quad j \geqslant 1 \quad (15)$$

$$P\{L_{\mathrm{SV}} = 0\} = B^*(1 - r_S) \quad (16)$$

where

$$B^* = \left[1 - r_S + r_S\left(1 - \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}\right)\right]^{-1}.$$

Proof
The probability generating function of L the stationary queue length according to the Theorem 2 is giving by :

$$\begin{aligned} L(z) =& \sum_{j=0}^{\infty} z^j p_1(j) + \sum_{j=1}^{\infty} z^j p_2(j) \\ =& K\left[\frac{1}{1 - r_S z} + \frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)} \frac{z}{1 - r_S z} \frac{1}{1 - \rho_B z}\right] \\ =& \frac{1 - \rho_B}{1 - \rho_B z} B^* \left[\frac{1 - r_S}{1 - r_S z}(1 - \rho_B z)\right. \\ &\left. + \frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)} \frac{z(1 - r_S)}{1 - r_S z}\right] \end{aligned}$$

where

$$B^* = \left[1 - \rho_B + \frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)}\right]^{-1}. \quad (17)$$

using equation (9), we get

$$\frac{\beta r_S}{\mu_{\mathrm{B}}(1 - r_S)} = \rho_B - r_S \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}$$

Substituting the above relation into the expression of equation (17), we have

$$B^* = \left[1 - r_S + r_S\left(1 - \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}\right)\right]^{-1}.$$

Then, we obtain $L_{\mathrm{SV}}(z)$

$$L_{\mathrm{SV}}(z) = B^* \left[1 - r_S + r_S \left(1 - \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}\right) \frac{z(1-r_S)}{1 - r_S z}\right] \quad (18)$$

We determine (15) by expanding (18) in power series of z,
With the stochastic decomposition structures in Theorem 3, we can easily get means

$$E\left(L_{\mathrm{SV}}\right) = \frac{(1 - \mu_{\mathrm{S}}/\mu_{\mathrm{B}})}{(1 - r_S \mu_{\mathrm{S}}/\mu_{\mathrm{B}})} \frac{r_S}{1 - r_S} \quad (19)$$

$$E\left(L\right) = \frac{\rho_B}{1 - \rho_B} + E\left(L_{\mathrm{SV}}\right). \quad (20)$$

2) The mean delay distribution at k^{th} fog node:

Theorem 4. *Let W_{MM1} be the mean delay of tasks in a $M/M/1$ queue, where the queue is exponentially distributed with parameter $\mu_B(1-\rho_B)$ and W_{SV} be the additional delay due to the standby. If $\rho_B < 1$ and $\mu_{\mathrm{B}} > \mu_{\mathrm{S}}$, Then the stationary delay W can be giving by the sum of two independent random variables as follow: $W = W_{MM1} + W_{\mathbf{SV}}$*

Where the Laplace-Stieltjes Transform (LST) of W_{SV} is

$$\begin{aligned} W^*_{\mathrm{SV}}(s) =& B^* \left[\frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}(1 - r_S)\right. \\ &\left. + \left(1 - \frac{\mu_{\mathrm{S}}}{\mu_{\mathrm{B}}}\right) \frac{(\lambda_S/r_S)(1-r_S)}{(\lambda_S/r_S)(1-r_S) + s}\right]. \end{aligned} \quad (21)$$

With the help of Theorem 4, which provides a stochastic decomposition, we can derive the mean delay as shown below.

$$\begin{aligned} E\left(W_{\mathrm{SV}}\right) &= \frac{1 - \mu_{\mathrm{S}}/\mu_{\mathrm{B}}}{1 - r_S \mu_{\mathrm{S}} \mu_{\mathrm{B}}} \cdot \frac{r_S}{(1 - r_S)} \left(\frac{1}{\lambda_B} + \frac{1}{\lambda_S}\right) \\ &= \left(\frac{1}{\lambda_B} + \frac{1}{\lambda_S}\right) E\left(L_{\mathrm{SV}}\right) \end{aligned} \quad (22)$$

$$E(W) = \frac{1}{\mu_{\mathrm{B}}(1 - \rho_B)} + E\left(W_{\mathrm{SV}}\right). \quad (23)$$

In our analytical model, we extend the evaluation of QoS parameters to the entire FC network, consisting of N fog nodes. Specifically, the total mean number of tasks in the system, $E(L_{\text{total}})$, assuming identical performance characteristics for all nodes.

$$E(L_{\text{total}}) = \sum_{k=1}^{N} E(L) \quad (24)$$

3) Failure probability: In the proposed $M/M/1$ queuing model, a request is considered failed if its delay exceeds a predefined threshold, D_{max}. Unlike systems with finite buffers, the $M/M/1$ model does not drop tasks due to overflow. Instead, task failure is primarily determined by the delay experienced by each task. The delay in an $M/M/1$ queue follows an exponential distribution. The probability of a task's delay exceeding D_{max} is given by:

$$P(W > D_{\text{max}}) = e^{-\mu(1-\rho)D_{\text{max}}} \quad (25)$$

The overall probability of failure is then obtained as a weighted sum of the state-specific failure probabilities, using the steady-state probabilities $P\{i = 1\}$ and $P\{i = 2\}$ to account for the proportion of time the fog node spends in each state. This approach provides a comprehensive evaluation of task failure in the FC system, considering both operational states. The overall probability of failure, P_{fail}; is given by:

$$\begin{aligned} P_{\text{fail}} = & P\{i = 1\} \times e^{-\mu_S(1-r_S)D_{\text{max}}} \\ & + P\{i = 2\} \times e^{-\mu_B(1-\rho_B)D_{\text{max}}} \end{aligned} \quad (26)$$

4) Reliability analysis of the fog computing system: The reliability R is calculated as the complement of the failure probability P_{fail}, which represents the probability that a task's delay exceeds D_{max}. This failure probability takes into account the likelihood of the fog node being in each state and the corresponding delay distributions. Then, the reliability R is given by:

$$R = 1 - P_{\text{fail}} \quad (27)$$

5) The number of failed nodes: To estimate the reliability of the FC network, we calculate the expected number of failed nodes. Given that each fog node has a probability of failure P_{fail}, the expected number of failed nodes out of N total fog nodes is given by:

$$N_F = N \times P_{\text{fail}} \quad (28)$$

V. Numerical results

In this section, we present numerical insights into the impact of key parameters on the QoS and reliability metrics in FC networks under different configurations. Our analysis examines how the variability of factors such as the number of faulty nodes, standby duration rates, arrival rates, and the number of fog nodes influences performance measures like steady-state probabilities, mean delay, task accumulation, reliability, and total delay. For this purpose, The values for the parameters used are cited in Table II, Table III, Table IV, Table V, Table VI :

TABLE II
Parameters and values used in Fig.3

Parameter	Value
μ_B	1.0 tasks/sec
μ_S	0.5 tasks/sec
β	0.5
λ_S	1 tasks/sec
λ_B	0.5 tasks/sec
D_{max}	2.0 seconds

The plot in Fig.3 illustrates the steady-state probability of a FC system as a function of the number of faulty nodes (k) for various system sizes ($N = 4, 5, 6, 10$) under steady-state conditions. The results show that as the number of faulty nodes increases, the probability initially rises, reaching a peak, and subsequently decreases. The peak represents the most likely configuration where the system achieves a balance between active and faulty nodes. For smaller systems ($N = 4, 5$), the peak occurs at lower values of k, reflecting

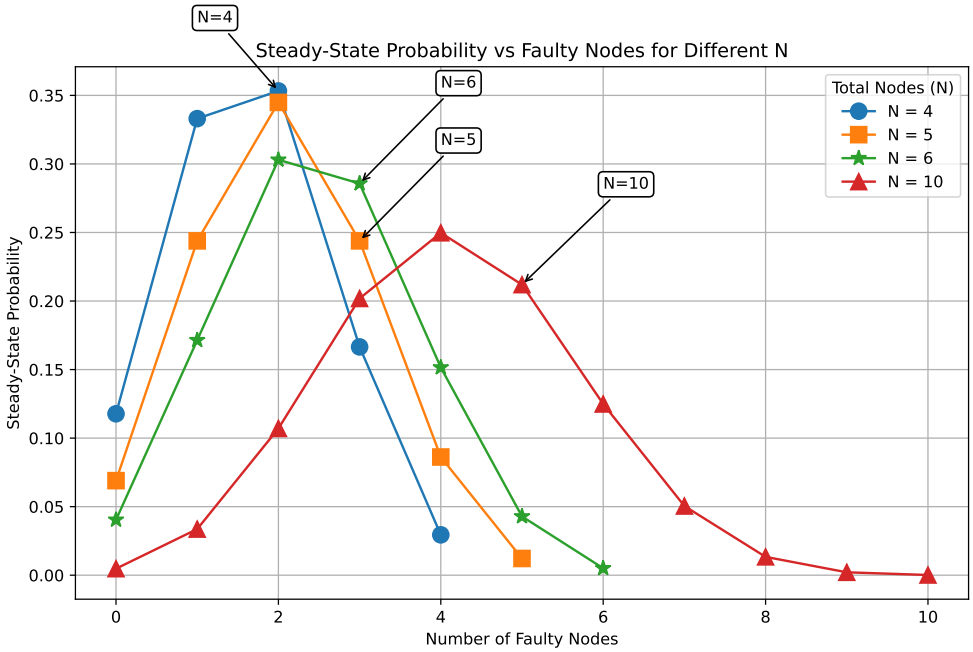

Fig. 3. Impact of faulty nodes on steady-state probability of the system for varying sizes.

limited fault tolerance and narrow variability in operational configurations. Conversely, larger systems ($N = 6, 10$) demonstrate a broader probability distribution with peaks at higher faulty nodes, indicating greater resilience to faults. For instance, the peak at $k = 5$ for $N = 10$ suggests the system is most stable when half of the nodes are faulty. These findings highlight the increased fault tolerance and scalability of larger systems, where dynamic workloads and resource allocation are better managed. However, smaller systems are more sensitive to node failures, requiring robust fault-tolerance mechanisms to maintain performance. This analysis underscores the importance of considering system size when designing FC architectures, as larger systems provide enhanced reliability at the cost of higher resource and energy requirements.

The work in [32] illustrate that larger fog networks show enhanced fault tolerance, maintaining system activity even with increasing faults, and that there is a threshold where additional faults impact the system less significantly. This supports the idea that larger network sizes are optimal for enhancing reliability in FC environments.

TABLE III
PARAMETERS AND VALUES USED IN FIG.4

Parameter	Value
μ_B	1.0 tasks/sec
μ_S	0.5 tasks/sec
λ_B	0.7 tasks/sec
λ_S	0.4 tasks/sec

The plot in Fig.4 demonstrates that the mean delay in fog networks increases with the number of faulty nodes, with the effect varying based on network size N. Smaller networks (e.g., $N = 4$) experience a sharper increase in delay as faults accumulate, indicating higher sensitivity to node failures and limited fault tolerance. In contrast, larger networks (e.g., $N = 10$) exhibit a more gradual delay increase, suggesting that they are better equipped to absorb faults without severely impacting task delay. This resilience in larger networks indicates that adding nodes enhances fault tolerance, allowing tasks to be processed with minimal delay even as some nodes fail. These findings highlight the importance of network size in maintaining performance and reliability, particularly in FC environments where delay sensitivity is critical.

Leveraging the relationship between delay and fault conditions revealed by the model, a task offloading algorithm can be developed to dynamically reroute tasks based on the current health of nodes and prevailing fault conditions. The model's insights into how delays escalate with increased faults can be integrated into the algorithm, enabling it to prioritize nodes with lower expected delays for task assignments.

TABLE IV
PARAMETERS AND VALUES USED IN FIG.5

Parameter	Value
μ_S	5 tasks/sec
μ_B	10 tasks/sec
λ_S	8 tasks/sec

The Fig.5 shows the relationship between the mean number of tasks and the arrival rate (λ_B) for different values of β. At lower values of β (e.g., $\beta = 0.1$), the system is slower to transition to the active state, resulting in a higher mean number of tasks as the arrival rate increases. As β increases (e.g., $\beta = 1$), the system becomes more responsive, allowing it to adapt more effectively to higher arrival rates, thereby reducing the mean number of tasks in the queue. At high values of β (e.g., $\beta = 10$), the system reaches near-optimal responsiveness, with the mean number

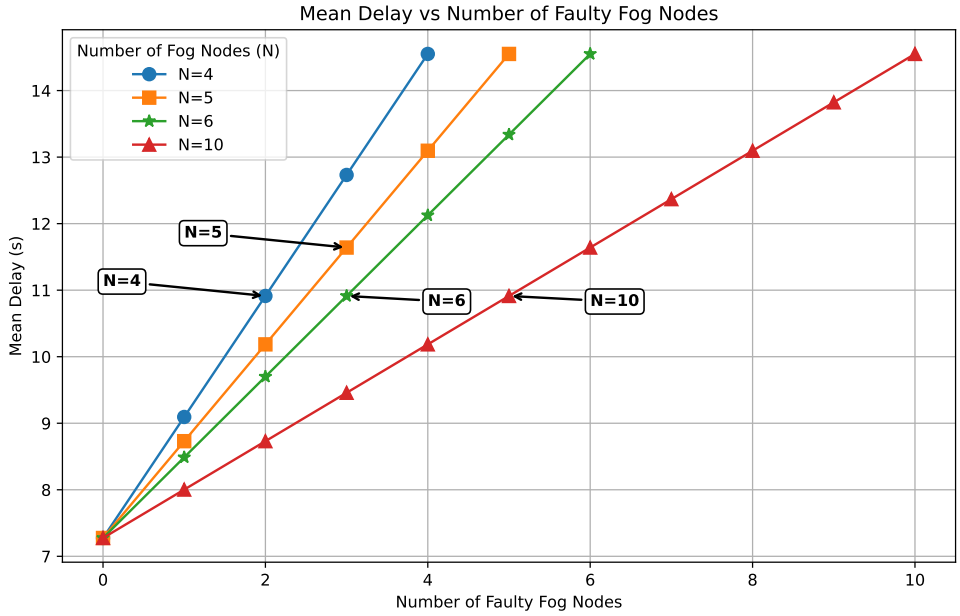

Fig. 4. Mean Delay vs. Number of faulty fog nodes for different network sizes N

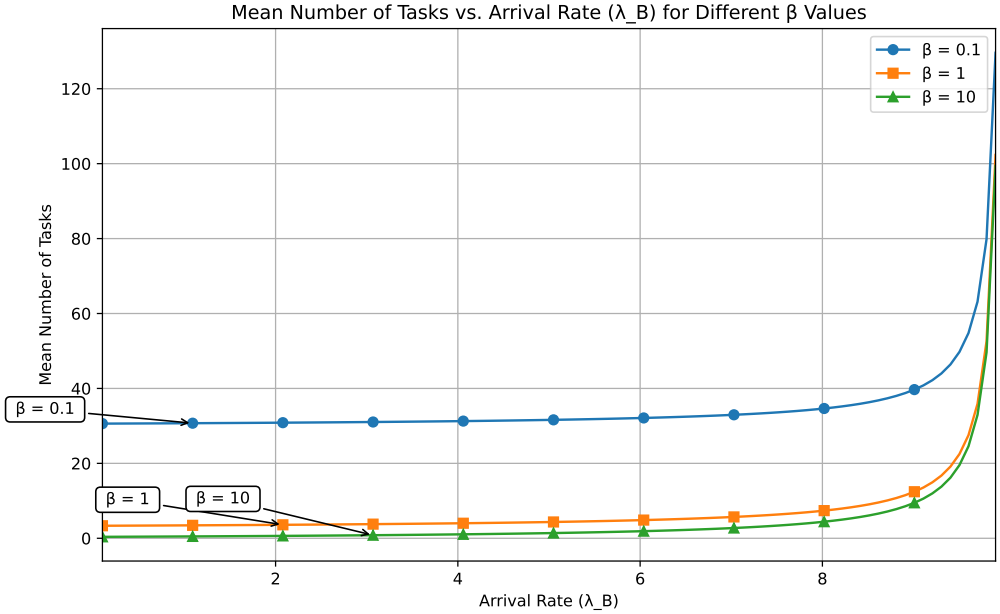

Fig. 5. Mean number of tasks sensitivity to arrival rate λ_B across different values of β

of tasks stabilizing even as λ_B increases. This suggests an optimal range for β that maximizes responsiveness without further gains in reducing task accumulation, highlighting the importance of tuning β to balance system efficiency and resource utilization in FC networks.

Whit higher values of β (closer to 10), the curves for the mean number of tasks become very close, indicating a diminishing effect of β on system responsiveness. This occurs because, beyond a moderate value of β, the system transitions from standby to active states frequently and quickly enough to handle incoming tasks efficiently. At higher values, the probability of the system being in standby becomes very low, leading the system to spend most of its time in the active state, where further increases in β have minimal impact on the queue. This saturation effect is typical in queuing systems, where certain parameters lose influence past a threshold, making additional increases in β ineffective for improving performance.

As the arrival rate λ_B nears the service rate $\mu_B = 10$(higher values), the mean number of tasks in the queue increases sharply for all β values. This occurs because, when λ_B

approaches μ_B, the system utilization ρ_B approaches 1, meaning the system is nearly saturated, with tasks arriving almost as quickly as they are processed. At high utilization, even minor fluctuations in arrivals or processing times cause significant backlogs, resulting in a steep rise in the mean number of tasks. At this point, the impact of β further diminishes, as the system remains mostly in the active state, where additional increases in β have minimal effect on queue length. This behavior reflects typical queuing instability as the system approaches its capacity limits.

TABLE V
PARAMETERS AND VALUES USED IN FIG.6

Parameter	Value
μ_B	1.0 tasks/sec
μ_S	0.5 tasks/sec
λ_S	0.4 tasks/sec

The Fig.6 shows that **higher transition rates** β, which correspond to shorter average standby durations, lead to increased reliability R as the system can more rapidly transition back to the active state to handle incoming tasks. Specifically, as β increases, reliability improves across all delay thresholds D_{max}, indicating that the system's responsiveness is enhanced by more frequent returns to the active state. Conversely, lower β values result in extended standby periods, leading to lower reliability, especially at smaller D_{max}, as tasks may be delayed waiting for the system to exit standby. This behavior highlights a trade-off: higher β values support greater responsiveness and reliability, beneficial in high-demand settings, whereas lower β values may save energy but risk increased delays, making them more suitable for scenarios prioritizing energy efficiency over immediate responsiveness.

TABLE VI
PARAMETERS AND VALUES USED IN FIG.8

Parameter	Value
μ_B	1.0 tasks/sec
μ_S	0.5 tasks/sec
λ_B	0.7 tasks/sec
λ_S	0.4 tasks/sec
β	0.1

The plot in Fig.8 illustrates the relationship between the **number of fog nodes** N, the **total delay** $E(W_{\text{total}})$, and the and **system probability**. As N increases, the total delay decreases significantly due to the distribution of tasks across a larger number of nodes, reducing the workload on each node. Simultaneously, the system-wide probability stabilizes, indicating improved reliability and fault tolerance as the system becomes less sensitive to individual node states. Higher delays are observed with fewer nodes, reflecting the strain on limited resources, whereas lower delays with more nodes demonstrate the benefits of parallelization. The results highlight the trade-offs between delay and reliability, emphasizing the importance of selecting an optimal number of fog nodes to balance performance and fault tolerance. These insights provide a foundation for resource planning and system optimization in FC networks.

Using this plot, we can determine an optimal number of fog nodes based on your system's delay tolerance. For instance, if there is a maximum delay threshold that must not be exceeded, we can find values of N where the total delay stays below this threshold. Within this range, selecting an N that maximizes the standby probability will ensure that the system operates efficiently while respecting the delay constraint. This approach enables a balanced decision on fog node deployment, prioritizing either energy savings (higher standby probability) or performance (lower delay) depending on the operational needs of the FC network.

Our findings on **mean delay and task queue length** align with [33] where its observed that higher responsiveness parameters (β) in fog systems efficiently reduce delays and manage task queues by enabling faster transitions to active states, particularly as arrival rates increase . Similarly, **reliability and failure probability** trends in our study correspond to observations by [34], demonstrating that increased β values enhance system reliability by reducing the likelihood of queue saturation and minimizing task delays under high utilization scenarios. Lastly, [35] confirm our results on **system size and fault tolerance**, highlighting that larger fog networks effectively distribute tasks and absorb node failures, maintaining lower delays and higher reliability across varying fault conditions. The findings of [36] align with the proposed work, as both demonstrate that higher transition rates between standby and active states significantly enhance reliability by reducing delays and ensuring timely task processing. Additionally, both studies highlight the trade-off between energy efficiency and responsiveness, confirming that systems with faster recovery times achieve better reliability under delay-sensitive conditions.

The plot in Fig.7 illustrates the relationship between the mean delay (W) and the standby duration rate (β) for different active service rates (μ_B). As β increases, the mean delay initially decreases significantly due to faster transitions from standby to active states, then stabilizes at higher β values. Systems with higher μ_B consistently achieve lower delays across all β, highlighting the importance of active service rate in reducing task processing time. For low μ_B systems, increasing β is critical to mitigate delays, while for high μ_B systems, the benefits of optimizing β diminish, with β values between 4 and 6 offering a balance. This analysis shows that optimizing β is essential for resource-constrained systems, but enhancing μ_B yields better results for high-capacity systems.

The Fig.9 presents a comparative analysis of the **mean delay** as a function of **failure percentage** between the proposed model and the Fault-Tolerant System Model (FSTM) in [37], that employs a hybrid fault-tolerance mechanism that integrates replication, checkpointing, and resubmission to handle failures in fog-cloud environments. It is observed that the proposed model consistently achieves significantly lower delay values compared to FSTM. For instance, at 5% failure, the proposed model maintains a delay of **7.2 seconds**, whereas FSTM exhibits a much higher delay of **2000 seconds**. As the failure percentage increases to 25%, the proposed model experiences a gradual rise in delay to

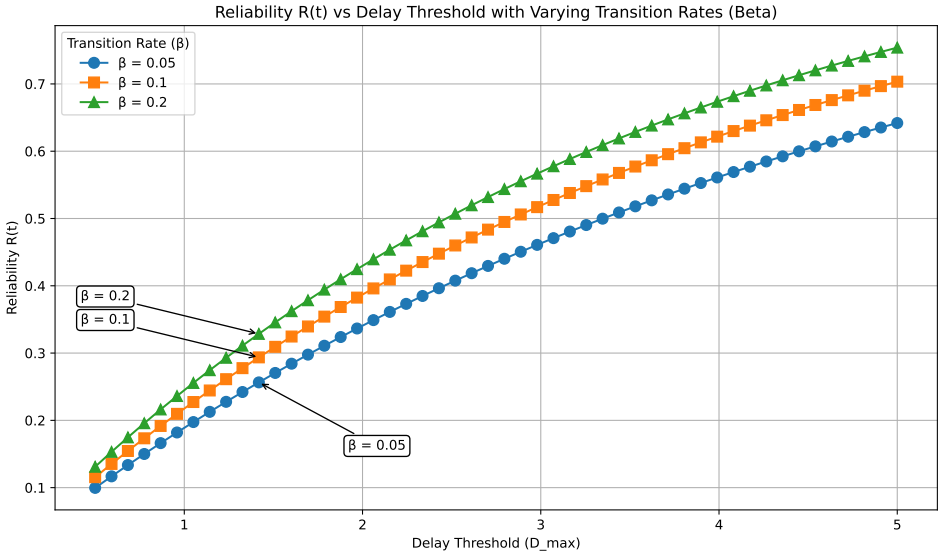

Fig. 6. Effect of standby duration rate β on reliability R across delay thresholds D_{max}

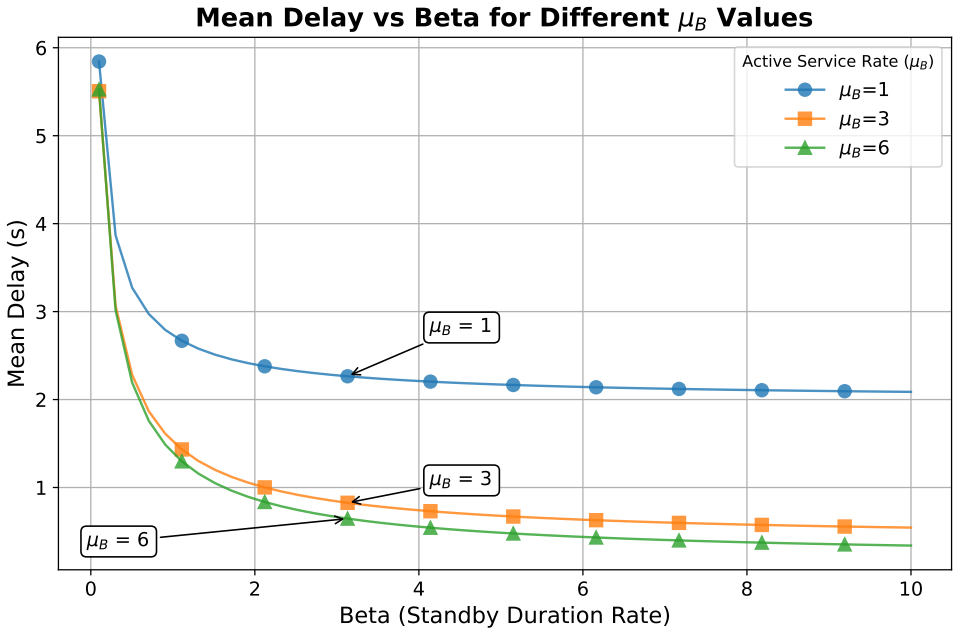

Fig. 7. Impact of standby duration rate (β) on mean delay for different active service rates (μ_B)

8.0 seconds, while FSTM's delay escalates sharply to **4000 seconds**.

From this, it is identified that the proposed model is not only more efficient but also more resilient to increasing fault rates, ensuring minimal delay even under adverse conditions. The logarithmic scale further highlights the vast disparity between the two methods, underscoring the superior fault-tolerant mechanism of the proposed model.

VI. Conclusion

In this study, Fog Computing network with dynamic task offloading and fault-tolerance mechanisms was investigated, addressing critical challenges such as system reliability under node failures and the effects of varying standby durations. The model was designed to incorporate both active and standby states for fog nodes, with performance metrics evaluated based on task arrival rates, service rates, and standby duration rates. This approach was developed to reflect real-world scenarios, where transitions between states are utilized to optimize resource efficiency and ensure system robustness.

For future research, it is suggested that adaptive standby and transition rate mechanisms be explored to adapt to real-time traffic conditions. Additionally, energy-efficient strategies for high-reliability fog networks could be developed, and machine learning techniques could be integrated to dynamically optimize task offloading in response to changing network conditions.

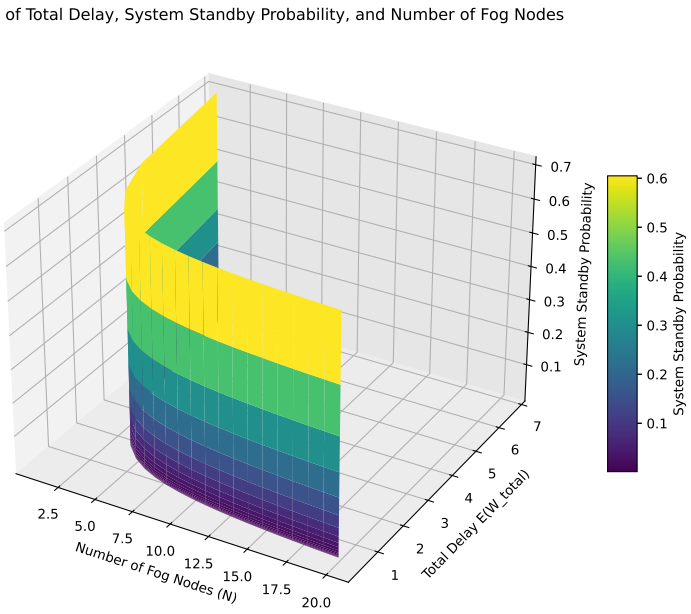

Fig. 8. Analysis of total delay, standby probability, and fog node number in a FC system

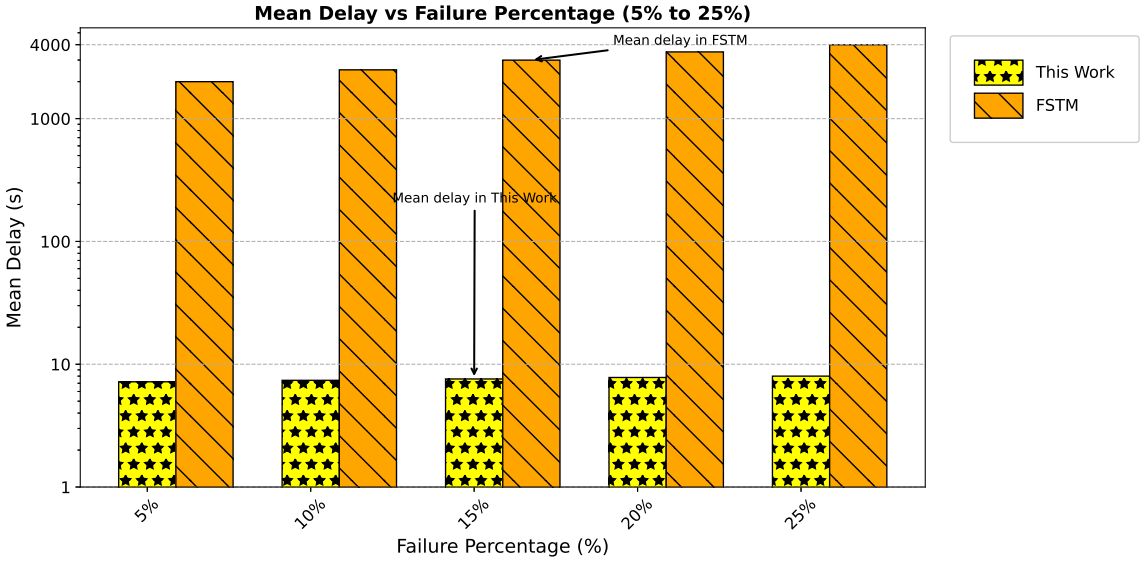

Fig. 9. Comparative analysis of mean delay vs failure percentage for proposed model and FSTM in [37]

REFERENCES

[1] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog computing: a comprehensive architectural survey," *IEEE access*, vol. 8, pp. 69 105–69 133, 2020.

[2] M. Songhorabadi, M. Rahimi, A. MoghadamFarid, and M. H. Kashani, "Fog computing approaches in iot-enabled smart cities," *Journal of Network and Computer Applications*, vol. 211, p. 103557, 2023.

[3] S. Thangam and S. S. Chakkaravarthy, "An edge-enabled virtual honeypot based intrusion detection system for vehicle-to-everything (v2x) security using machine learning," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp. 1374–1384, 2024.

[4] R. Das and M. M. Inuwa, "A review on fog computing: issues, characteristics, challenges, and potential applications," *Telematics and Informatics Reports*, p. 100049, 2023.

[5] K. Behravan, N. Farzaneh, M. Jahanshahi, and S. A. H. Seno, "A comprehensive survey on using fog computing in vehicular networks," *Vehicular Communications*, p. 100604, 2023.

[6] F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, and S. Papavassiliou, "Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions," *Computer Networks*, vol. 195, p. 108177, 2021.

[7] G. Cohen, "Downtime, outages and failures-understanding their true costs," 2022.

[8] B. Wang, B. Lv, and Y. Song, "A hybrid genetic algorithm with integer coding for task offloading in edge-cloud cooperative computing," *IAENG International Journal of Computer Science*, vol. 49, no. 2, pp. 503–510, 2022.

[9] M. Ghazali and A. B. Tahar, "A queuing theory approach to task scheduling in cloud computing with generalized processor sharing queue model and heavy traffic approximation," *IAENG International*

Journal of Computer Science, vol. 51, no. 10, pp. 1604–1611, 2024.

[10] J. Lee, H. Ko, D. Suh, S. Jang, and S. Pack, "Overload and failure management in service function chaining," in *2017 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2017, pp. 1–5.

[11] E. Shiriaev, T. Ermakova, E. Bezuglova, M. A. Lapina, and M. Babenko, "Reliablity and security for fog computing systems," *Information*, vol. 15, no. 6, p. 317, 2024.

[12] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 581–594, 2018.

[13] J. Lim, "Versatile cloud resource scheduling based on artificial intelligence in cloud-enabled fog computing environments," *Hum.-Centric Comput. Inf. Sci*, vol. 13, p. 54, 2023.

[14] Z. Chang, L. Liu, X. Guo, and Q. Sheng, "Dynamic resource allocation and computation offloading for iot fog computing system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3348–3357, 2020.

[15] S. A. Ansar, J. K. Samriya, M. Kumar, S. S. Gill, and R. A. Khan, "Intelligent fog-iot networks with 6g endorsement: Foundations, applications, trends and challenges," *6G Enabled Fog Computing in IoT: Applications and Opportunities*, pp. 287–307, 2023.

[16] S. Seifhosseini, M. H. Shirvani, and Y. Ramzanpoor, "Multi-objective cost-aware bag-of-tasks scheduling optimization model for iot applications running on heterogeneous fog environment," *Computer Networks*, vol. 240, p. 110161, 2024.

[17] Y. E. Hamouda, "Optimal cluster head localization for cluster-based wireless sensor network using free-space optical technology and genetic algorithm optimization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 10, pp. 3693–3713, 2024.

[18] N. Siasi, M. Jasim, and N. Ghani, "Post-fault restoration of service function chains in fog networks," *Computer Networks*, p. 110580, 2024.

[19] A. Reyana, S. Kautish, K. A. Alnowibet, H. M. Zawbaa, and A. Wagdy Mohamed, "Opportunities of iot in fog computing for high fault tolerance and sustainable energy optimization," *Sustainability*, vol. 15, no. 11, p. 8702, 2023.

[20] F. Sarkohaki and M. Sharifi, "Service placement in fog–cloud computing environments: a comprehensive literature review," *The Journal of Supercomputing*, pp. 1–33, 2024.

[21] R. A. Alsemmeari, M. Y. Dahab, B. Alturki, A. A. Alsulami, and R. Alsini, "Towards an effective service allocation in fog computing," *Sensors*, vol. 23, no. 17, p. 7327, 2023.

[22] V. Mohammadi, A. M. Rahmani, A. Darwesh, and A. Sahafi, "Fault tolerance in fog-based social internet of things," *Knowledge-Based Systems*, vol. 265, p. 110376, 2023.

[23] M. Mudassar, Y. Zhai, L. Liao, and J. Shen, "A decentralized latency-aware task allocation and group formation approach with fault tolerance for iot applications," *IEEE Access*, vol. 8, pp. 49 212–49 223, 2020.

[24] Q. Wang and H. Mu, "Privacy-preserving and lightweight selective aggregation with fault-tolerance for edge computing-enhanced iot," *Sensors*, vol. 21, no. 16, p. 5369, 2021.

[25] Z. Bakhshi, G. Rodriguez-Navas, and H. Hansson, "Analyzing the performance of persistent storage for fault-tolerant stateful fog applications," *Journal of systems architecture*, vol. 144, p. 103004, 2023.

[26] F. B. Abbasi, A. Rezaee, S. Adabi, and A. Movaghar, "Fault-tolerant scheduling of graph-based loads on fog/cloud environments with multi-level queues and lstm-based workload prediction," *Computer Networks*, vol. 235, p. 109964, 2023.

[27] H. Chen, J. Wen, W. Pedrycz, and G. Wu, "Big data processing workflows oriented real-time scheduling algorithm using task-duplication in geo-distributed clouds," *IEEE Transactions on Big Data*, vol. 6, no. 1, pp. 131–144, 2018.

[28] A. R. Setlur, S. J. Nirmala, H. S. Singh, and S. Khoriya, "An efficient fault tolerant workflow scheduling approach using replication heuristics and checkpointing in the cloud," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 14–28, 2020.

[29] P. Chanak, I. Banerjee, and S. Bose, "An intelligent fault-tolerant routing scheme for internet of things-enabled wireless sensor networks," *International Journal of Communication Systems*, vol. 34, no. 17, p. e4970, 2021.

[30] S. Ghanavati, J. Abawajy, and D. Izadi, "Automata-based dynamic fault tolerant task scheduling approach in fog computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 488–499, 2020.

[31] M. F. Neuts, "Matrix-geometric solutions in stochastic models, volume 2 of johns hopkins series in the mathematical sciences," 1981.

[32] P. Zhang, Y. Chen, M. Zhou, G. Xu, W. Huang, Y. Al-Turki, and A. Abusorrah, "A fault-tolerant model for performance optimization of a fog computing system," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1725–1736, 2021.

[33] R. Beraldi, C. Canali, R. Lancellotti, and G. P. Mattia, "Distributed load balancing for heterogeneous fog computing infrastructures in smart cities," *Pervasive and Mobile Computing*, vol. 67, p. 101221, 2020.

[34] X. Qin, Y. Li, X. Song, N. Ma, C. Huang, and P. Zhang, "Timeliness of information for computation-intensive status updates in task-oriented communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 623–638, 2022.

[35] G. Proietti Mattia, M. Magnani, and R. Beraldi, "A latency-levelling load balancing algorithm for fog and edge computing," in *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2022, pp. 5–14.

[36] C. Shekhar, N. Kumar, A. Gupta, A. Kumar, and S. Varshney, "Warm-spare provisioning computing network with switching failure, common cause failure, vacation interruption, and synchronized reneging," *Reliability Engineering & System Safety*, vol. 199, p. 106910, 2020.

[37] A. Alarifi, F. Abdelsamie, and M. Amoon, "A fault-tolerant aware scheduling method for fog-cloud environments," *PloS one*, vol. 14, no. 10, p. e0223902, 2019.