



Stochastic Modeling of Energy Consumption in Tethered High Altitude Platforms with Dynamic Sleep Control

Harini R¹ · Dharmaraja Selvamuthu¹ 

Received: 29 September 2025 / Revised: 17 July 2026 / Accepted: 7 September 2026

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2026

Abstract

High-Altitude Platforms (HAPs) are emerging as promising aerial infrastructures for extending wireless coverage, improving resilience, and enabling rapid deployment. Among them, tethered HAPs (tHAPs) are particularly attractive due to their stable positioning and continuous power supply; however, their persistent energy demand remains a major concern. Communication payloads, signal processing hardware, and backhaul links operate continuously and consume significant power, increasing operational costs and raising sustainability challenges. To address this, we develop a stochastic queueing model that embeds dynamic sleep modes under an adaptive threshold policy to improve energy efficiency. The model also incorporates predictive maintenance and a standby server mechanism to ensure seamless service continuity and maintain quality of service (QoS) during failures and repair. The model is governed by a bi-variate Markov process and analyzed using the supplementary variable technique, through which the stability conditions and steady-state performance measures are derived. In addition, cost optimization is carried out using three metaheuristic optimization techniques, demonstrating that energy efficiency and operational reliability can be jointly achieved. Numerical evaluations show that the proposed framework substantially reduces power consumption and operational costs while maintaining robust performance, providing an effective strategy for the sustainable deployment of tHAP-assisted networks.

Keywords Energy consumption · tHAP · Dynamic sleep · Quorum queues · Cost optimization

Mathematics Subject Classification (2010) 60J28 · 68M20 · 90B22 · 60K15 · 60K25

1 Introduction

As 6G visions take shape, non-terrestrial networks (NTNs) are increasingly recognized for their role in extending coverage and enhancing resilience beyond terrestrial limits. NTNs

Harini R and Dharmaraja Selvamuthu contributed equally to this work.

✉ Dharmaraja Selvamuthu
dharma@maths.iitd.ac.in

Harini R
harini220696@gmail.com

¹ Department of Mathematics, Indian Institute of Technology Delhi, New Delhi 110016, India

extend terrestrial coverage using aerial and spaceborne platforms, enabling connectivity in remote, underserved, and emergency scenarios. Among them, high-altitude platforms (HAPs) have emerged as cost-effective, rapidly deployable solutions that provide wide-area coverage and act as super macro base stations (SMBSSs) that complement terrestrial networks in 6G. Operating typically in the stratosphere at altitudes of 17–25 km, HAPs benefit from near-stationary positioning above weather systems and air traffic. This enables connectivity over large footprints with lower propagation delay than satellites. Within this family, tethered HAPs (tHAPs) are gaining attention due to their stable positioning, continuous ground-based power supply, and secure backhaul, enabling virtually unlimited flight endurance. This unique vantage point enables them to deliver large coverage footprints with a reduced propagation delay compared to satellites.

Unlike free-flying HAPs, tHAPs are physically connected to a ground station via a tether that carries both power and data links, ensuring sustained operation without dependence on onboard fuel or solar limitations. This makes them highly suitable for dense urban Internet of Things (IoT) deployments, smart city applications, disaster recovery, and emergency communication restoration. Furthermore, tHAPs can act as localized hubs for 6G edge services, seamlessly integrating with terrestrial base stations while providing flexible and resilient NTN connectivity at a fraction of satellite deployment costs. An architecture diagram of tHAP is shown in Fig. 1.

In the broader NTN context, space air ground integrated networks (SAGINs) are emerging as a key 6G architecture where satellites, aerial platforms, and terrestrial systems cooperate to provide seamless global coverage. HAPs serve as the air segment, bridging terrestrial base stations and low-Earth orbit (LEO) satellites. While LEO constellations offer global reach, they face intermittent connectivity, which can be complemented by HAPs through

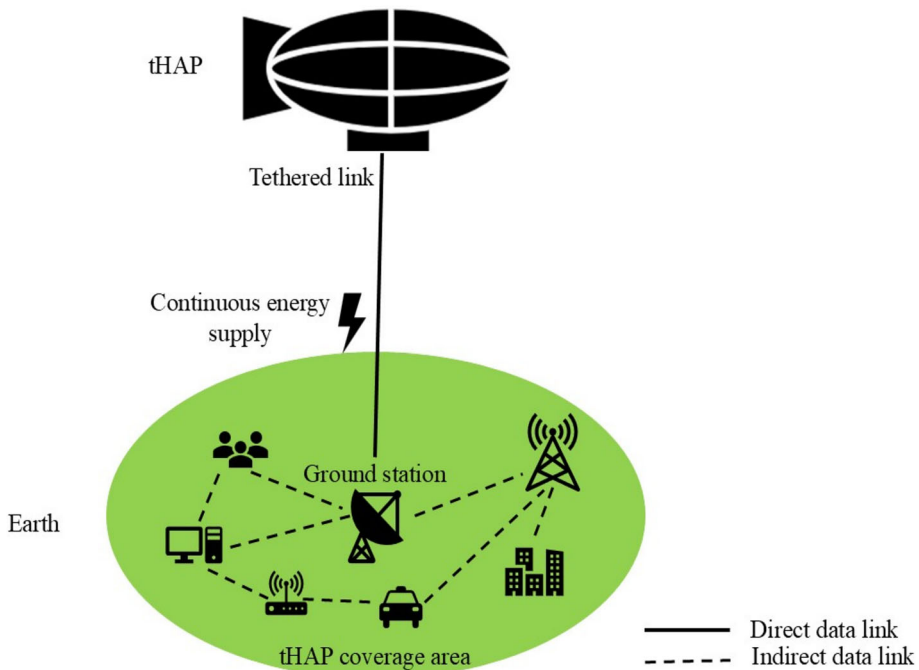


Fig. 1 Architecture diagram of tHAP

quasi-stationary, lower-latency service. When deployed as SMBSSs, HAPs can also aggregate IoT traffic, improve uplink reliability, and enhance resilience in high-demand or emergency scenarios.

Despite these advantages, the energy demand of tHAPs remains a critical concern. Communication payloads, high-frequency radios, and backhaul systems consume substantial power, driving up operational costs and raising sustainability challenges. This creates a strong need for stochastic models that jointly address energy efficiency, service reliability, and cost optimization. Motivated by these challenges, we develop a stochastic queueing framework for tHAPs that integrates dynamic sleep strategies, predictive maintenance, standby mechanisms, stability analysis, and cost optimization, offering a comprehensive foundation for sustainable and resilient NTN deployment.

Early energy-saving approaches for tHAPs often relied on static operational states or binary ON/OFF schemes, which are insufficient for handling the highly dynamic and bursty nature of traffic in non-terrestrial networks. To overcome these limitations, dynamic sleep mechanisms have emerged as a practical solution for reducing the energy consumption of tHAPs. Unlike rigid strategies, dynamic sleep policies allow the communication payload of a tHAP to transition gradually into light, moderate, or deep sleep modes, each offering different trade-offs between energy savings and reactivation delay.

These tiered sleep modes enable fine-grained energy management for tHAPs:

- Light sleep powers down high-consumption transceiver modules, enabling rapid wake-up within milliseconds, making it suitable for short idle intervals or temporary dips in IoT traffic.
- Moderate sleep further deactivates digital signal processing and control units, achieving deeper energy savings with slightly longer reactivation times, ideal for extended low-load periods.
- Deep sleep retains only minimal control logic while shutting down most subsystems, reducing energy use significantly. Although wake-up may take seconds, it is effective for prolonged quiet intervals, such as overnight in rural deployments.

Transitions between these states are governed by adaptive threshold policies. If demand falls below predefined thresholds for a given duration, the system progresses into deeper sleep states. This prevents frequent oscillations between active and idle states, thus ensuring both energy efficiency and operational stability in long-term deployments.

Beyond the server-side power management, the access behaviour of user equipment must also be considered. A key component of this behaviour is the retrial mechanism triggered when immediate service is unavailable. This behaviour naturally arises in tHAP-assisted wireless communication systems due to the way users handles unsuccessful access attempts. When an uplink attempt cannot be served—because the tHAP is in a sleep state, the random-access channel is congested, or the scheduler is fully occupied, then the user does not drop the request. Instead, the user performs a backoff procedure and reattempts access, similar to the contention-based procedures used in cellular networks. This behaviour is frequently observed in real deployments involving large numbers of IoT and massive Machine-Type Communications (mMTC) devices. For instance, devices such as smart meters, agricultural sensors, UAVs, or remote monitoring units periodically send short packets; if the tHAP is momentarily unavailable or operating in a reduced-power mode, these devices simply retry after a short waiting time. Such retransmission-based reattempt behaviour is well documented in grant-free massive random access systems for IoT communications (Bian et al. 2023). Therefore, incorporating a retrial mechanism accurately reflects the practical operation of

massive IoT systems supported by tHAPs, where access delays, sleep mode transitions, and temporary congestion naturally trigger reattempts.

Further, connection requests are not always processed individually in tHAP-assisted wireless systems. Modern communication payloads operate using time–frequency resource blocks or subframes, which are the smallest schedulable units at the physical layer. Before initiating a transmission, the scheduler aggregates a group of pending requests to efficiently populate a resource block, thereby reducing signaling overhead, minimizing repeated wake-up activations, and improving spectral utilization. For tHAPs, activating high-power transceivers and baseband chains for very small workloads is energy-inefficient, especially under the multi-level sleep architecture shown in Fig. 2. This behaviour naturally corresponds to a batch-service mechanism, where service begins only after a minimum number of requests are available. Such quorum-based service is particularly relevant in practical IoT deployments, where thousands of devices—such as environmental sensors, utility meters, drones, and surveillance systems—transmit short packets in periodic or synchronized bursts. For example, smart-city air-quality sensors upload readings every few minutes, while utility meters often generate simultaneous bursts during billing intervals. In these scenarios, requests tend to accumulate in groups, making batch scheduling both energy-efficient and operationally realistic. Therefore, employing a fixed batch size r in our model captures

- the physical-layer grouping of requests into resource blocks,
- the energy cost associated with activating high-power transceivers for very small loads, and
- the naturally bursty and correlated behaviour of IoT traffic supported by tHAPs.

Such scheduler-driven aggregation and clustered access behavior in IoT and mMTC systems have been widely reported in recent studies on 5G/6G resource allocation and random access mechanisms (Firouzi and Rahmani 2024; Daraseliya et al. 2025).

Additionally, ensuring long-term reliability is equally important for tHAP operations. Predictive maintenance (PM) can be employed to proactively detect and mitigate potential faults in the payload or tethered systems before they escalate into service outages. This is particularly important in real tHAP deployments, where the communication payload, power converters, cooling units, stabilization mechanisms, and the tether itself operate continuously and are subject to thermal stress, mechanical load, and environmental variations. Regular monitoring and proactive servicing help identify early signs of deterioration and prevent unexpected failures during long-duration missions. Despite its practical importance, PM has not been explicitly modeled in existing analytical studies of HAP or tHAP systems, making its integration into our framework a notable contribution.

A standby server mechanism ensures continuity by activating only during failures, minimizing downtime without significant energy overhead. By integrating these features, tHAPs can achieve a more balanced trade-off between energy efficiency, service reliability, and scalability, making them a practical and sustainable component of future SAGINs.

1.1 Motivation

Energy efficiency in wireless networks has been extensively studied for terrestrial base stations. Earlier works (Aggarwal et al. 2025; Deepa and Haridass 2023; Deepa et al. 2023; Kalita and Selvamuthu 2023) have applied stochastic and queueing models with different traffic assumptions and buffer constraints. However, these studies confirmed the potential of energy-saving strategies, they remained focused on terrestrial infrastructures.

More recently, tHAPs have attracted attention as super macro base stations with continuous ground-based power. A few studies (Jain et al. 2022; Priya et al. 2025) (preprint) have investigated the energy consumption of tHAPs using queueing models and stochastic analysis, highlighting their potential role as energy-efficient enablers in future non-terrestrial networks.

These studies provide valuable foundations; however, most of them remain narrow in scope, focusing primarily on evaluating energy usage under specific conditions or assumptions. They typically emphasize performance measures without addressing broader operational challenges such as system reliability, adaptability to varying traffic dynamics, or integration of optimization objectives.

Our work goes beyond these limitations by developing a stochastic queueing framework for tHAPs that captures traffic fluctuations through retrials and batch arrivals. The model also incorporates reliability aspects such as predictive maintenance and standby mechanisms to ensure QoS. We embed dynamic multi-level sleep modes under an adaptive threshold policy and derive analytical stability conditions. Furthermore, cost optimization is performed using three distinct techniques, providing a comprehensive foundation for the design of sustainable and resilient tHAP-assisted networks.

1.2 Novelty

Unlike existing tHAP studies that mainly focus on average energy consumption, our work develops a unified stochastic queueing framework. The unique novelty of this study lies in embedding a three-level adaptive sleep strategy within a queueing model that incorporates reliability mechanisms, cost optimization, retrials, and batch traffic. In addition, the incorporation of PM as an internal reliability mechanism further distinguishes our model, enabling proactive fault handling rarely addressed in prior tHAP literature. This combination yields both analytical stability conditions and practical guidelines for designing sustainable and resilient tHAP networks.

1.3 Research Contribution

The major contributions of this work are summarized below:

- **Energy-Efficient tHAP Framework:** We develop a stochastic modeling framework for tHAPs that explicitly incorporates energy consumption with multi-level dynamic sleep control. This approach allows the communication payload to operate adaptively under varying traffic intensities, thereby improving sustainability.
- **Traffic Dynamics with Retrials and Batch Arrivals:** The model captures realistic traffic patterns by considering batch arrivals of requests and retrial phenomena when immediate service is unavailable, ensuring a closer alignment with practical network behavior.
- **Reliability Aware Mechanisms:** We integrate predictive maintenance and a standby server strategy into the tHAP system, ensuring service continuity during failures and significantly improving service availability.
- **Analytical Performance Evaluation:** The stability and steady-state probability vectors are derived using supplementary variable technique, enabling the computation of key performance measures of the proposed system.
- **Cost Optimization:** A cost model is formulated, and optimization is carried out using three metaheuristic algorithms. The analysis demonstrates that the proposed tHAP

framework achieves substantial reductions in operational costs compared to existing approaches.

- **Practical Relevance:** The proposed model is highly applicable to real-world scenarios such as emergency communication, rural connectivity, and dense IoT deployments, where energy-efficient yet reliable tethered platform operation is critical.

1.4 Literature Survey

Energy efficient operation of HAPs has gained significant attention in recent years. For instance, Abbasi et al. (2024) provided a comprehensive discussion on the importance of HAPS in future 6G architectures, underscoring key challenges and motivating further research on their efficient operation, while Xu et al. (2023) highlighted the role of NTN and integrated space air ground sea systems in enabling energy-aware connectivity. Farooqi et al. (2025) evaluated HAPs as a sustainable, solar-powered alternative to terrestrial 5G base stations. They showed that one HAP can replace about 25 ground stations, offering major energy savings and improved environmental sustainability. Recently, Amponis et al. (2025) proposed a Composite Cost Metric (CCM) that enhances energy-efficient routing in integrated UAV-LEO satellite networks for 6G NTNs.

Liu et al. (2025) introduced a Lyapunov-DRL based strategy that significantly improves energy-efficient computation offloading and resource allocation in AAV/HAP-assisted AIoT systems while maintaining queue stability. More recently, Jain and Gautam (2025) analyzed energy management reliability in solar-powered HAPs using a three-level continuous-time Markov chains (CTMCs) and reliability block diagrams (RBD) framework, offering an accurate evaluation of the battery and solar-cell subsystems through k-out-of-n models and machine-learning-based prediction. For tethered HAPs specifically, Jain et al. (2022) developed a retrial queueing framework ($MAP/PH^{[3]}/1$) to study power management, and Vishnevsky et al. (2020) analyzed the reliability of tethered telecommunication modules. Reliability and availability aspects were also examined using semi-Markov modeling by Mittal et al. (2024).

In parallel, Raj and Dharmaraja (2023) studied multi-layer HAP-LEO systems for energy savings, and Raj (2024) extended this with Markov regenerative processes to enhance efficiency. Priya et al. (2025)(preprint) explored stochastic modeling of tethered HAP-based smart macro base stations, emphasizing energy efficiency in dense deployments. More recently, Song et al. (2024) positioned HAPs as new energy efficiency enablers in the 6G era.

The literature on queueing theory has evolved over decades, with retrial queues gaining notable attention for their relevance in modeling systems with repeated customer attempts. Furthermore, extensive research on retrial queues under varied system dynamics has been carried out. In particular, batch arrival retrial queues have gained popularity, with recent studies by Samanta and Rashmi (2025) and Ayyappan and Nithya (2025), who introduced constant retrial policies and non-Markovian features.

The foundational text by Chaudhry and Templeton (1983) introduced the basics of bulk queues, setting the stage for quorum-based models, where service occurs in batches of fixed size $r \leq 1$. Tadj et al. (2006, 2012) have contributed significantly to this area, analyzing quorum queues under various realistic conditions. More recently, Kashifalghitaa and Al-Obaidi (2023) investigated different arrival scenarios in quorum queues under N-policy, highlighting the continued interest and development in this specialized queueing framework.

Recent studies on retrial queueing systems have explored the use of standby servers and PM to improve reliability and efficiency. Liu et al. (2024) and Toth et al. (2023) examined models

with backup or parallel servers, addressing failures and customer impatience. Meanwhile, Rathinasamy et al. (2023) and Sapna et al. (2024) focused on PM in failure-prone systems, highlighting their impact on system performance, availability, and optimal control strategies.

Table 1 provides a comprehensive comparison of existing studies on energy efficiency in base stations (BS) and HAP, emphasizing the superior performance and novel contributions of the proposed work.

The structure of the paper unfolds as follows: Section 2 outlines the proposed tHAP model in detail. Section 3 presents the system's stability conditions, steady-state behavior, and the governing equations. Key performance metrics are analyzed in Section 4. Section 5 provides statistical insights into the interaction between various system components. Section 6 evaluates the model's energy efficiency. Section 7 focuses on cost optimization using three distinct metaheuristic techniques. Finally, Section 8 summarizes the conclusions and offers directions for future research.

2 Model Formulation

This section proposes a stochastic framework that describes the dynamic behaviour of a tHAP operating under adaptive energy control mechanisms. This framework facilitates an in-depth analysis of system performance and energy efficiency by highlighting the interaction between adaptive power-saving responses and fluctuations in user demand. Figure 2 illustrates the proposed model.

Arrival Process With an arrival rate of κ , requests stream in batches following a Poisson process. Let $\mathcal{A}_k$ represent the number of requests associated with the k^{th} batch arrival, with the assumption that $\mathcal{A}_k$, ($k = 1, 2, 3, \dots$) follows a standard distribution. The PGF of $\mathcal{A}$ is denoted by $\mathcal{A}(z)$ where $Pr[\mathcal{A}_k = n] = \mathcal{X}_n$, $n = 1, 2, 3, \dots$

Retrial Process Incoming requests encountering a busy, sleeping, failed, or maintenance-state tHAP are blocked and placed in an orbit, a group of pending requests. However, when a retrial request finds the server available, service begins immediately. The orbit's requests act independently of one another and are tenacious as they keep trying again until they receive service. Further, we assume that the retry times of the requests are randomly distributed and equally divided throughout the orbit with a random distribution $\mathcal{S}(x)$ with a corresponding Laplace Stieltjes Transform (LST) $\mathcal{S}^*(s)$.

Service Process The service mechanism operates in fixed-size batches, where the server processes exactly R connection requests simultaneously during each service cycle. If fewer than R requests are available, the server remains idle until the batch size requirement is met. This setup reflects scenarios where group processing improves throughput and resource utilization. First Come, First Served (FCFS) processing prevails for each incoming request. The service time Q is assumed to have a general distribution function $Q(x)$ and a corresponding LST $Q^*(s)$.

Vacation Process Following the completion of a service, if there is no immediate request to serve or the system shifts to a low-power mode, the tHAP enters a light sleep state (SM1) with probability ϖ_1 . Upon completion of SM1, the tHAP checks the number of pending requests; if at least N are present, the tHAP resumes service with probability $(1 - \varpi_2)$, otherwise it enters a moderate sleep state (SM2) with probability ϖ_2 . After SM2, the request

Table 1 Comparative summary of existing models for energy efficiency in tHAP systems

Author(s)	Queueing Model	Sleep Strategy	Batch Arrival; Service; Retrial	Reliability	Optimization
Aggarwal et al. (2025)	$M/G/1$	4 static sleep states	✓	✓	✓
Aggarwal and Selvamuthu (2024)	Semi-Markov	Markovian sleep model	✓	✓	✓
Harini and Indhira (2024)	$M^{[X]}/G/1$	Two sleep modes	✓	✓	✓ Cost optimization
Kalita and Selvamuthu (2023)	$M/G/1$	Two static sleep levels	✓	✓	✓
Deepa et al. (2023)	$M/G/1$	Two sleep modes + N policy	✓	✓	✓
Deepa and Haridass (2023)	$M/G/1$ (finite buffer)	Variant sleep states	✓	✓	✓
Jain et al. (2022)	$MAP/PH^{[3]}/1$	Power-aware ON/OFF	✓	✓	✓
Priya et al. (2025) (preprint)	Stochastic tHAP model	Multi-level sleep (light/deep)	✓	✓	✓
Song et al. (2024)	Binary optimization	Multi-stage sleep for HAPS	✓	✓	Policy-based energy optimization
Raj and Dharmaraja (2023)	HAP-LEO multi-layer Markov model	Dynamic sleep modes	✓	✓	✓
Raj (2024)	Markov regenerative process (HAP-LEO)	Multi-level sleep	✓	✓	✓
Proposed Model	$M^{[X]}/G^{[R]}/1$	Dynamic 3- level sleep (SM1–SM3)	✓	Maintenance + Standby	WOA, ABC, GWO

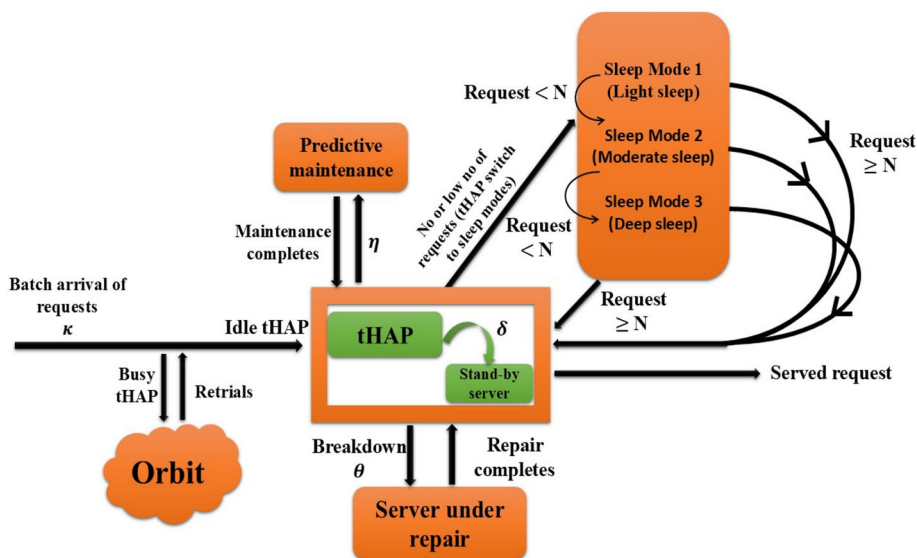


Fig. 2 A visual representation of the proposed model

count is checked again. If still less than N , the tHAP transitions into deep sleep (SM3) with probability ϖ_3 ; otherwise, it returns to active mode with probability $1 - \varpi_3$. This multi-level sleep mechanism enables adaptive energy saving based on traffic conditions while maintaining service responsiveness. Further, SM1, SM2, and SM3 are characterized by increasing sleep durations S_1 , S_2 and S_3 respectively. The sleep durations S_i ($i=1,2,3$) for each mode SM_i are modelled as random variables with cumulative distribution functions $\mathcal{L}_i(x)$ and a corresponding LST $\mathcal{L}_i^*(s)$.

Maintenance Process The server transitions into an idle state to perform PM at a rate η , even when n customers are present in the orbit. The maintenance duration follows a distribution $\mathcal{M}(x)$ and a corresponding LST $\mathcal{M}^*(s)$. This rule reflects proactive servicing based on operational time rather than system emptiness, aiming to reduce unexpected failures and improve system reliability.

Breakdown Process Despite operating normally, the tHAP remains susceptible to failures, resulting in short-term outages. These breakdowns, regarded as catastrophic events, are triggered by exogenous Poisson processes with rate θ .

Repair Process When a tHAP fails, it is immediately dispatched for repair. The repair time is modeled as a general distribution $\mathcal{R}(y)$ with LST $\mathcal{R}^*(s)$. During repair, the tHAP is unavailable for processing new connection requests. Interrupted requests resume their remaining service once the tHAP and associated service channel are operational again.

Stand-by server To maintain service continuity, a standby server is activated whenever the primary tHAP is under repair. The standby server immediately takes over pending requests, providing uninterrupted processing. The service times of the standby server are assumed to follow an exponential distribution with rate $\delta > 0$.

Every stochastic process in the system is assumed to be independent of the others, which is a standard approach used to maintain analytical tractability. Although tHAP operations may exhibit correlations among sleep transitions, retrials, repairs, and maintenance events, capturing such interactions would require a far more involved joint-process formulation. Such independence assumptions are standard in queueing and reliability models, and despite minor correlations that may occur in practice, they offer a reasonable and widely accepted approximation of overall system behaviour.

3 Analysis of Steady-State Probabilities

The analytical framework in this section is based on classical stochastic process and queueing theory, where system evolution is modeled using continuous-time Markov processes and steady-state conditions are derived using standard analytical techniques. Such approaches are well established in the literature and have been discussed extensively in recent works by Selvamuthu (2025), which provide a rigorous theoretical foundation for the analysis of retrial and batch-service queueing systems.

An $M^{[X]}/G^{[R]}/1$ retrial queue with dynamic sleep modes is used to model the client request retrial process in the proposed tHAP system. Here, $\Psi(t)$ denotes the tHAP's state, which corresponds to $(0, 1, 2, 3, 4, 5, 6)$, such as idle, busy, in one of the sleep modes, undergoing maintenance, or undergoing repair. Thus, we have

$$\Psi(t) = \begin{cases} 0, & \text{when the tHAP is momentarily inactive at } t, \\ 1, & \text{when the tHAP is operational at } t, \\ 2, & \text{when the tHAP resides in its light sleep state (SM1) at time } t, \\ 3, & \text{when the tHAP resides in its moderate sleep state (SM2) at } t, \\ 4, & \text{when the tHAP resides in its deep sleep state (SM3) at } t, \\ 5, & \text{when the tHAP resides in its maintenance state at time } t, \\ 6, & \text{when the tHAP resides under repair at } t. \end{cases}$$

The state of the system at given time t is further shown to be defined by the bi-variate Markov process $\{(\Psi(t), \Xi(t)); t \geq 0\}$. $\Xi(t)$ tracks the number of requests in the orbit at that moment. To retain the Markov property under generally distributed retrial, service, sleep, maintenance, and repair times, supplementary variables are introduced. The variables

Table 2 Notations used for sojourn time distribution in the model

Description	Retrial	Service	Sleep (i=1,2,3)	Maintenance	Repair
Cumulative distribution function	$\mathcal{S}(x)$	$\mathcal{Q}(x)$	$\mathcal{L}_i(x)$	$\mathcal{M}(x)$	$\mathcal{R}(y)$
Probability density function	$s(x)$	$q(x)$	$l_i(x)$	$m(x)$	$r(y)$
Laplace–Stieltjes transform	$\mathcal{S}^*(x)$	$\mathcal{Q}^*(x)$	$\mathcal{V}_i^*(x)$	$\mathcal{M}^*(x)$	$\mathcal{R}^*(y)$
Elapsed time distribution	$\mathcal{S}^0(t)$	$\mathcal{Q}^0(t)$	$\mathcal{L}_i^0(t)$	$\mathcal{M}^0(t)$	$\mathcal{R}^0(t)$

$S^0(t)$, $Q^0(t)$, $L_i^0(t)$ ($i = 1, 2, 3$), $M^0(t)$, and $R^0(t)$ represent the corresponding elapsed (age) times. Further, the distributional functions, their corresponding Laplace-Stieltjes transforms, elapsed-time distributions, and model parameters used throughout the analysis are summarized in Tables 2 and 3 respectively.

Theorem 1 *Ergodicity of the embedded Markov chain $\{W_n; n \in N\}$ holds true when $\rho < R$, where $\rho = S^*(\kappa + \eta) + \kappa E(X)\sigma + [S^*(\kappa + \eta) + \kappa\sigma]\aleph - \eta\sigma[\kappa E(X)E(P) - R]$ and $\aleph = -\kappa E(X)\varpi_1[E(L_1) + \varpi_2[E(L_2) + \varpi_3 E(L_3)]]Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)$.*

Proof The proof strategy mirrors that of Theorem 1 as provided by Kumar and Madheswari (2003). $\square$

Assume that $\{t_n; n = 1, 2, \dots\}$ is the collection of epochs at which an active, sleep, maintenance or repair period ends. Thus, an embedded Markov chain is formulated for the proposed system using a sequence of random vectors $W_n = \{(\Psi(t_n+), \Xi(t_n+))\}$. Hence, the stability of the proposed system follows whenever $\rho < R$.

In the steady state, it is assumed that $S(0) = 0$, $S(\infty) = 1$, $Q(0) = 0$, $Q(\infty) = 1$, $\mathcal{L}_i(0) = 0$, $\mathcal{L}_i(\infty) = 1$, ($i = 1, 2, 3$), $\mathcal{M}(0) = 0$, $\mathcal{M}(\infty) = 1$ are continuous at $x = 0$ and $\mathcal{R}(0) = 0$ and $\mathcal{R}(\infty) = 1$ are continuous at $y = 0$. Accordingly, the functions $\omega(x)$, $\beta(x)$, $v_i(x)$, ($i = 1, 2, 3$), $\xi(x)$ and $\mu(y)$ are used to define the conditional completion rates (hazard rates), for the retrial, active sleep modes, maintenance, and repair operations respectively.

$$\omega(x)dx = \frac{dS(x)}{1 - S(x)}; \quad \beta(x)dx = \frac{dQ(x)}{1 - Q(x)};$$

Table 3 Model parameters and their descriptions

Symbol	Description
κ	Arrival rate of requests
R	Required batch size to initiate service
ϖ_1	Probability of entering SM1 (light sleep) after service completion
ϖ_2	Probability of entering SM2 if requests $< N$ after SM1
ϖ_3	Probability of entering SM3 if requests $< N$ after SM2
N	Wake-up threshold
η	Predictive maintenance rate
θ	Breakdown rate of the server
δ	Standby server rate
$P_I(t)$	Probability that the tHAP remains idle at time t
$P_A(t)$	Probability that the tHAP is active at time t
$P_{S,1}(t)$	Probability that the tHAP is under light sleep at time t
$P_{S,2}(t)$	Probability that the tHAP is under moderate sleep at time t
$P_{S,3}(t)$	Probability that the tHAP is under deep sleep at time t
$P_M(t)$	Probability that the tHAP remains under maintenance at time t
$P_B(t)$	Probability that the tHAP is under repair at time t

$$v_1(x)dx = \frac{d\mathcal{L}_1(x)}{1 - \mathcal{L}_1(x)}; \quad v_2(x)dx = \frac{d\mathcal{L}_2(x)}{1 - \mathcal{L}_2(x)}; \quad v_3(x)dx = \frac{d\mathcal{L}_3(x)}{1 - \mathcal{L}_3(x)};$$

$$\xi(x)dx = \frac{d\mathcal{M}(x)}{1 - \mathcal{M}(x)}; \quad \mu(y)dy = \frac{d\mathcal{R}(y)}{1 - \mathcal{R}(y)}.$$

Accordingly, the probabilities $P_I(t) = P\{\Psi(t) = 1, \Xi(t) = 0\}$; and the corresponding probability densities are defined as follows,

$$P_{R,n}(x, t)dx = P\{\Xi(t) = 0, \Psi(t) = n, x \leq \mathcal{S}^0(t) < x + dx\}, \text{ for } t, x \geq 0, n \geq 1.$$

$$P_{A,n}(x, t)dx = P\{\Xi(t) = 1, \Psi(t) = n, x \leq \mathcal{Q}^0(t) < x + dx\}, \text{ for } t, x \geq 0, n \geq 0.$$

$$P_{S,i,n}(x, t)dx = P\{\Xi(t) = 2(3, 4), \Psi(t) = n, x \leq \mathcal{L}_i^0(t) < x + dx\}, \text{ for } t, x \geq 0,$$

$$(i = 1, 2, 3), n \geq 0.$$

$$P_{M,n}(x, t)dx = P\{\xi(t) = 3, \Psi(t) = n, x \leq \mathcal{M}^0(t) < x + dx\}, \text{ for } t, x \geq 0, n \geq 0.$$

$$P_{B,n}(x, y, t)dx = P\{\Theta(t) = 4, \Xi(t) = n, y \leq \mathcal{R}^0(t) < y + dy/\mathcal{Q}^0(t) = x\},$$

$$\text{for } t \geq 0, (x, y) \geq 0, n \geq 0.$$

Assuming that the system satisfies the stability condition, the limiting probabilities for $x > 0, n \geq 0$, and $(i = 1, 2, 3)$ are given by,

$$P_I = \lim_{t \rightarrow \infty} P_I(t); \quad P_{R,n}(x) = \lim_{t \rightarrow \infty} P_{R,n}(x, t);$$

$$P_{A,n}(x) = \lim_{t \rightarrow \infty} P_{A,n}(x, t); \quad P_{S,i}(x) = \lim_{t \rightarrow \infty} \Gamma_i(x, t);$$

$$P_{M,n}(x) = \lim_{t \rightarrow \infty} P_{M,n}(x, t); \quad \text{and } P_{B,n}(x, y) = \lim_{t \rightarrow \infty} P_{B,n}(x, y, t).$$

The following equations, which describe the system's dynamic behavior, are derived using the supplementary variable method.

$$\kappa P_I = \int_0^\infty P_{S,1,0}(x)v_{1,0}(x)dx + \int_0^\infty P_{S,2,0}(x)v_{2,0}(x)dx + \int_0^\infty P_{S,3,0}(x)v_{3,0}(x)dx. \quad (1)$$

$$\frac{dP_{R,n}(x)}{dx} + (\kappa + \eta + \omega(x))P_{R,n}(x) = 0; \quad n \geq 1. \quad (2)$$

$$\frac{dP_{A,n}(x)}{dx} + (\kappa + \theta + \beta(x))P_{A,n}(x) = \kappa \sum_{k=1}^n x_k P_{A,n-k}(x); \quad n \geq N. \quad (3)$$

$$\frac{dP_{S,1,n}(x)}{dx} + (\kappa + v_1(x))P_{S,1,n}(x) = \kappa \sum_{k=1}^n x_k P_{S,1,n-k}(x); \quad n \leq N-1. \quad (4)$$

$$\frac{dP_{S,2,n}(x)}{dx} + (\kappa + v_2(x))P_{S,2,n}(x) = \kappa \sum_{k=1}^n x_k P_{S,2,n-k}(x); \quad n \leq N-1. \quad (5)$$

$$\frac{dP_{S,3,n}(x)}{dx} + (\kappa + v_3(x))P_{S,3,n}(x) = \kappa \sum_{k=1}^n x_k P_{S,3,n-k}(x); \quad n \leq N-1. \quad (6)$$

$$\frac{dP_{M,n}(x)}{dx} + (\kappa + \xi(x))P_{M,n}(x) = \kappa \sum_{k=1}^n a_k P_{M,n-k}(x); \quad n \geq 0. \quad (7)$$

$$\frac{dP_{B,n}(x, y)}{dy} + (\kappa + \delta + \mu(y))P_{B,n}(x, y) = \kappa \sum_{k=1}^n a_k P_{B,n-k}(x, y); \quad n \geq 0. \quad (8)$$

The steady state boundary conditions at $x = 0$, $y = 0$ are as follows:

$$P_{R,n}(0) = \varpi_1 \int_0^\infty P_{A,n} \beta(x) dx + \varpi_2 \int_0^\infty P_{S,1,n}(x) v_1(x) dx + \varpi_3 \int_0^\infty P_{S,2,n}(x) v_2(x) dx \\ + \int_0^\infty P_{S,3,n}(x) v_3(x) dx + \int_0^\infty P_{M,n}(x) \xi(x) dx + \int_0^\infty P_{B,n}(x, y) \mu(y) dy; \quad n \geq 1. \quad (9)$$

$$P_{A,1}(0) = \int_0^\infty P_{R,1}(x) \omega(x) dx + \kappa x_1 P_I; \quad n = 0. \quad (10)$$

$$P_{A,n}(0) = \int_0^\infty P_{R,n+R}(x) \vartheta(x) dx + \kappa x_{n+R} P_I + \kappa \sum_{k=1}^n x_k \int_0^\infty P_{R,n-k+R}(x) dx; \quad n \geq N.$$

$$P_{S,1,n}(0) = \begin{cases} \varpi_1 \int_0^\infty P_{A,n} \beta(x) dx; & n \leq N-1, \\ 0; & n \geq N. \end{cases} \quad (11)$$

$$P_{S,2,n}(0) = \varpi_2 \int_0^\infty P_{S,1,n}(x) v_1(x) dx; \quad n \leq N-1. \quad (12)$$

$$P_{S,3,n}(0) = \varpi_3 \int_0^\infty P_{S,2,n}(x) v_2(x) dx; \quad n \leq N-1. \quad (13)$$

$$P_{M,0}(0) = \eta P_{R,0}(x); \quad n = 0. \quad (14)$$

$$P_{M,n}(0) = \eta \int_0^\infty P_{R,n}(x) dx; \quad n \geq 1. \quad (15)$$

$$P_{B,n}(x, 0) = \theta \int_0^\infty P_{A,n}(x); \quad n \geq 0. \quad (16)$$

The normalizing condition is as follows,

$$P_I + \sum_{n=1}^{\infty} \int_0^\infty P_{R,n}(x) dx + \sum_{n=0}^{\infty} \left[\int_0^\infty P_{A,n}(x) dx + \int_0^\infty P_{S,1,n}(x) dx + \int_0^\infty P_{S,2,n}(x) dx \right. \\ \left. + \int_0^\infty P_{S,3,n}(x) dx + \int_0^\infty P_{M,n}(x) dx + \int_0^\infty \int_0^\infty P_{B,n}(x, y) dy dx \right] = 1.$$

3.1 The Steady State Solution

The generating function technique is employed to obtain the steady state solution of the proposed model. The generating functions for $|z| < 1$ are defined as follows to solve the above equations.

$$u(z) = \sum_{n=1}^{\infty} \mathcal{X}_n z^n; \quad P_R(x, z) = \sum_{n=1}^{\infty} P_{R,n}(x) z^n; \quad P_R(0, z) = \sum_{n=1}^{\infty} P_{R,n}(0) z^n; \\ P_A(x, z) = \sum_{n=0}^{\infty} P_{A,n}(x) z^n; \quad P_A(0, z) = \sum_{n=0}^{\infty} P_{A,n}(0) z^n; \\ P_{S,i}(x, z) = \sum_{n=0}^{\infty} P_{S,i,n}(x) z^n; \quad P_{S,i}(0, z) = \sum_{n=0}^{\infty} P_{S,i,n}(0) z^n, \quad (i = 1, 2, 3); \\ P_M(x, z) = \sum_{n=0}^{\infty} P_{M,n}(x) z^n; \quad P_M(0, z) = \sum_{n=0}^{\infty} P_{M,n}(0) z^n;$$

$$P_B(x, y, z) = \sum_{n=0}^{\infty} P_{B,n}(x, y)z^n; \quad P_B(x, 0, z) = \sum_{n=0}^{\infty} P_{B,n}(x, 0)z^n.$$

Multiplying the boundary conditions and the steady-state Eqs. 2 to 16 by z^n and summing over n where $(n = 0, 1, 2, \dots)$, yields,

$$\frac{\partial}{\partial x} P_R(x, z) + (\kappa(1 - u(z)) + \eta + \omega(x))P_R(x, z) = 0. \quad (17)$$

$$\frac{\partial}{\partial x} P_A(x, z) + [\kappa(1 - u(z)) + \theta + \beta(x)]P_A(x, z) = 0. \quad (18)$$

$$\frac{\partial}{\partial x} P_{S,i}(x, z) + [\kappa(1 - u(z)) + v_i(x)]P_{S,i}(x, z) = 0; \quad (i = 1, 2, 3). \quad (19)$$

$$\frac{\partial}{\partial x} P_M(x, z) + [\kappa(1 - u(z)) + \xi(x)]P_M(x, z) = 0. \quad (20)$$

$$\frac{\partial}{\partial y} P_B(x, y, z) + [\kappa(1 - u(z)) + \delta(1 - \frac{1}{z}) + \mu(x)]P_B(x, z) = 0. \quad (21)$$

$$\begin{aligned} P_R(0, z) = & \varpi_1 \int_0^{\infty} P_A(x)\beta(x)dx + \varpi_2 \int_0^{\infty} P_{S,1}(x)v_1(x)dx + \varpi_3 \int_0^{\infty} P_{S,2}(x)v_2(x)dx \\ & + \int_0^{\infty} P_{S,3}(x)v_3(x)dx + \int_0^{\infty} P_M(x)\xi(x)dx + \int_0^{\infty} P_B(x, y)\mu(y)dy \\ & - (\kappa + \eta)P_I. \end{aligned} \quad (22)$$

$$P_A(0, z) = \frac{1}{zR} \int_0^{\infty} P_R(x, z)\omega(x)dx + \frac{\kappa}{zR} \mathcal{X}(z) \int_0^{\infty} P_R(x, z)dx + \mathcal{X}(z) \frac{\kappa P_I}{zR}.$$

$$P_{S,1}(0, z) = \varpi_1 \int_0^{\infty} P_A(x, z)\beta(x)dx. \quad (23)$$

$$P_{S,2} = \varpi_2 \int_0^{\infty} P_{S,1}(x, z)v_1(x)dx. \quad (24)$$

$$P_{S,3}(0, z) = \varpi_3 \int_0^{\infty} P_{S,2}(x, z)v_2(x)dx. \quad (25)$$

$$P_M(0, z) = \eta \left[\int_0^{\infty} P_R(x, z)dx + P_I \right]. \quad (26)$$

$$P_B(x, 0, z) = \theta \int_0^{\infty} P_A(x, z)dx. \quad (27)$$

Solving the partial differential Eqs. 17 to 21, we get,

$$P_R(x, z) = P_R(0, z)[1 - \mathcal{S}(x)]e^{-(\kappa+\eta)x}. \quad (28)$$

$$P_A(x, z) = P_A(0, z)[1 - \mathcal{Q}(x)]e^{-(\mathcal{A}(z)x)}. \quad (29)$$

$$P_{S,i}(x, z) = P_{S,i}(0, z)[1 - \mathcal{L}_i(x)]e^{-(\mathcal{C}(z)x)}; \quad (i = 1, 2, 3). \quad (30)$$

$$P_M(x, z) = P_M(0, z)[1 - \mathcal{M}(x)]e^{-(\mathcal{C}(z)x)}. \quad (31)$$

$$P_B(x, y, z) = P_B(x, 0, z)[1 - \mathcal{R}(z)]e^{-(\mathcal{B}(z)x)}. \quad (32)$$

where, $\mathcal{C}(z) = \kappa(1 - u(z))$; $\mathcal{A}(z) = \mathcal{C}(z) + \theta$; $\mathcal{B}(z) = \mathcal{C}(z) + \delta(1 - \frac{1}{z})$.

Theorem 2 Under the stability condition $\rho < R$, the probability generating functions (PGFs) of the request-count distributions corresponding to the idle, active, sleep, maintenance, and repair states of the tHAP are given by

$$P_R(z) = \frac{P_I \sigma \left\{ \kappa \mathcal{X}(z) \left[\mathcal{Q}^*(\mathcal{A}(z)) \varsigma + \frac{\theta(1-\mathcal{Q}^*(\mathcal{A}(z)))}{\mathcal{A}(z)} \right] - \eta(1 - \mathcal{M}^*(\mathcal{C}(z)))z^R - \kappa z^R \right\}}{Dr(z)}. \quad (33)$$

$$P_A(z) = \frac{P_I \left\{ \mathcal{S}^*(\kappa + \eta) \kappa [\mathcal{X}(z) - 1] - \mathcal{S}^*(\kappa + \eta) \eta (1 - \mathcal{M}^*(\mathcal{C}(z))) \right\} (1 - \mathcal{Q}^*(\mathcal{A}(z)))}{Dr(z) \mathcal{A}(z)}. \quad (34)$$

$$P_{S,1}(z) = \frac{P_I \varpi_1 \mathcal{Q}^*(\mathcal{A}(z)) (1 - \mathcal{L}_1^*(\mathcal{C}(z))) \left\{ \mathcal{S}^*(\kappa + \eta) \kappa [\mathcal{X}(z) - 1] - \mathcal{S}^*(\kappa + \eta) \eta (1 - \mathcal{M}^*(\mathcal{C}(z))) \right\}}{Dr(z) \mathcal{C}(z)}. \quad (35)$$

$$P_{S,2}(z) = \frac{P_I \varpi_1 \varpi_2 \mathcal{Q}^*(\mathcal{A}(z)) \mathcal{L}_1^*(\mathcal{C}(z)) (1 - \mathcal{L}_2^*(\mathcal{C}(z))) \left\{ \mathcal{S}^*(\kappa + \eta) \kappa [\mathcal{X}(z) - 1] - \mathcal{S}^*(\kappa + \eta) \eta (1 - \mathcal{M}^*(\mathcal{C}(z))) \right\}}{Dr(z) \mathcal{C}(z)}. \quad (36)$$

$$P_{S,3}(z) = \frac{P_I \varpi_1 \varpi_2 \varpi_3 \mathcal{Q}^*(\mathcal{A}(z)) \mathcal{L}_1^*(\mathcal{C}(z)) \mathcal{L}_2^*(\mathcal{C}(z)) (1 - \mathcal{L}_3^*(\mathcal{C}(z))) \left\{ \mathcal{S}^*(\kappa + \eta) \kappa [\mathcal{X}(z) - 1] - \mathcal{S}^*(\kappa + \eta) \eta (1 - \mathcal{M}^*(\mathcal{C}(z))) \right\}}{Dr(z) \mathcal{C}(z)}. \quad (37)$$

$$P_M(z) = \frac{P_I \eta \mathcal{S}^*(\kappa + \eta) \left\{ z^R - \left[\mathcal{Q}^*(\mathcal{A}(z)) \varsigma + \frac{\theta(1-\mathcal{Q}^*(\mathcal{A}(z)))}{\mathcal{A}(z)} \right] \right\} (1 - \mathcal{M}^*(\mathcal{C}(z)))}{Dr(z) \mathcal{C}(z)}. \quad (38)$$

$$P_B(z) = \frac{\theta P_I (1 - \mathcal{R}^*(\mathcal{B}(z))) (1 - \mathcal{Q}^*(\mathcal{A}(z))) \left\{ \mathcal{S}^*(\kappa + \eta) \kappa [\mathcal{X}(z) - 1] - \mathcal{S}^*(\kappa + \eta) \eta (1 - \mathcal{M}^*(\mathcal{C}(z))) \right\}}{Dr(z) \mathcal{B}(z) \mathcal{A}(z)}. \quad (39)$$

where,

$$P_I = \frac{R - \left\{ \mathcal{S}^*(\kappa + \eta) + \kappa E(X) \sigma + [\mathcal{S}^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X) E(P) - R] \right\}}{[\mathcal{S}^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]] [E(Q) (1 - \theta E(R)) + \varpi_1 \mathcal{Q}^*(\theta) [E(L_1) + \varpi_2 [E(L_2) + \varpi_3 E(L_3)]]] + \mathcal{S}^*(\kappa + \eta) [R - 1 + \eta E(P) R + \aleph [1 - \eta E(P)]]}. \quad (40)$$

$$Dr(z) = z^R - [\mathcal{S}^*(\kappa + \eta) + \kappa \mathcal{A}(z) \sigma] \left[\mathcal{Q}^*(\mathcal{A}(z)) \varsigma + \frac{\theta(1 - \mathcal{Q}^*(\mathcal{A}(z)))}{\mathcal{A}(z)} \right] - \eta \sigma z^R \mathcal{M}^*(\mathcal{C}(z)).$$

$$\sigma = \frac{\mathcal{S}^*(\kappa + \eta)}{(\kappa + \eta)}.$$

$$\varsigma = 1 - \varpi_1 (1 - \mathcal{L}_1^*(\mathcal{C}(z))) - \varpi_1 \varpi_2 \left[(1 - \mathcal{L}_2^*(\mathcal{C}(z))) + \varpi_3 \mathcal{L}_2^*(\mathcal{C}(z)) (1 - \mathcal{L}_3^*(\mathcal{C}(z))) \right].$$

$$\varsigma' = -\kappa E(X) \varpi_1 [E(L_1) + \varpi_2 [E(L_2) + \varpi_3 E(L_3)]].$$

$$\aleph = \varsigma' \mathcal{Q}^*(\theta) - \kappa E(X) E(Q) - \theta E(Q).$$

Proof Using Eqs. 22 to 26 together with Eqs. 28 to 31, and integrating with respect to x , the probability generating functions are obtained as

$$P_R(z) = \int_0^\infty P_R(x, z)dx; P_A(z) = \int_0^\infty P_A(x, z)dx; P_{S,i}(z) = \int_0^\infty P_{S,i}(x, z)dx;$$

$$(i = 1, 2, 3); P_M(z) = \int_0^\infty P_M(x, z)dx;$$

Similarly, using Eq. 27 together with Eq. 32, the corresponding probability generating functions are defined as

$$P_B(x, z) = \int_0^\infty P_B(x, y, z)dy, P_B(z) = \int_0^\infty P_B(x, z)dx.$$

Consequently, Eqs. 33–39 are obtained.

The normalizing condition is used to determine the unknown probability P_I , which denotes the probability that the server remains inactive. Specifically, P_I is obtained by setting $z = 1$ in Eqs. 33 to 39 and applying L'Hôpital's rule whenever required.

$$P_I + P_R(1) + P_A(1) + P_{S,1}(1) + P_{S,2}(1) + P_{S,3}(1) + P_M(1) + P_B(1) = 1.$$

□

Theorem 3 Under the stability constraint $\rho < R$, the PGF of the number of requests in the system and the orbit are given by:

$$K(z) = P_I + P_R(z) + P_{S,1}(z) + P_{S,2}(z) + P_{S,3}(z) + z[P_A(z) + P_M(z) + P_B(z)] \quad (41)$$

$$H(z) = P_I + P_R(z) + P_{S,1}(z) + P_{S,2}(z) + P_{S,3}(z) + P_A(z) + P_M(z) + P_B(z) \quad (42)$$

where P_I is represented by Eq. 40.

Proof The proof follows directly by substituting Eqs. 33 to 39 into Eqs. 41 and 42. □

4 System Performance Measures

When the system satisfies the stability condition $\rho < R$, as described below, the steady state probabilities of various system states are obtained from Eqs. 33 to 39.

1. Let P_R signify the stationary probability that the tHAP is not active during the retrial interval.

$$P_R = \frac{P_I \sigma \left[\kappa E(X) + \kappa [\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] - \eta \kappa E(X)E(P) - \kappa R \right]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X)\sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X)E(P) - R] \right\}}. \quad (43)$$

2. Let P_A signify the stationary probability that the tHAP is active

$$P_A = \frac{P_I [S^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]] E(Q)}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X)\sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X)E(P) - R] \right\}}. \quad (44)$$

3. $P_{S,1}$ signify the steady state probability associated with the tHAP remaining in SM1.

$$P_{S,1} = \frac{P_I \varpi_1 Q^*(\theta) E(L_1) [S^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X)\sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X)E(P) - R] \right\}}. \quad (45)$$

4. $P_{S,2}$ signify the steady state probability associated with the tHAP remaining in SM2.

$$P_{S,2} = \frac{P_I \varpi_1 \varpi_2 Q^*(\theta) E(L_2) [S^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X) \sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X) E(P) - R] \right\}}. \quad (46)$$

5. $P_{S,3}$ signify the steady state probability associated with the tHAP remaining in SM3.

$$P_{S,3} = \frac{P_I \varpi_1 \varpi_2 \varpi_3 Q^*(\theta) E(L_3) [S^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X) \sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X) E(P) - R] \right\}}. \quad (47)$$

6. P_M signify the steady state probability associated with the tHAP under maintenance.

$$P_M = \frac{P_I \eta S^*(\kappa + \eta) E(P) [R - [\zeta' Q^*(\theta) - \kappa E(X) E(Q) - \theta E(Q)]]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X) \sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X) E(P) - R] \right\}}.$$

7. P_B signify the steady state probability associated with the tHAP under repair.

$$P_B = \frac{P_I E(Q) E(R) [S^*(\kappa + \eta) \kappa E(X) [1 - \eta E(P)]]}{R - \left\{ S^*(\kappa + \eta) + \kappa E(X) \sigma + [S^*(\kappa + \eta) + \kappa \sigma] \aleph - \eta \sigma [\kappa E(X) E(P) - R] \right\}}.$$

4.1 Mean Size of a System and its Orbit

When the system is in steady state,

- (i) Under the stability condition, the mean orbit size L_q is computed by taking the derivative of Eq. 42 with respect to z and evaluating it at $z = 1$.

$$L_q = H'(1) = \lim_{z \rightarrow 1} \frac{d}{dz} H(z) = P_I \left[\frac{Nr_2'''(1) Dr_1''(1) - Dr_1'''(1) Nr_2''(1)}{3(Dr_1''(1))^2} \right].$$

- (ii) Under the stability condition, the mean system size L_s is computed by taking the derivative of Eq. 41 with respect to z and evaluating it at $z = 1$.

$$L_s = K'(1) = \lim_{z \rightarrow 1} \frac{d}{dz} K(z) = P_I \left[\frac{Nr_1'''(1) Dr_1''(1) - Dr_1'''(1) Nr_1''(1)}{3(Dr_1''(1))^2} \right].$$

The parameter values are given in Appendix A.

- (iii) Little's law facilitates the computation of the expected system time

$$W_s = \frac{L_s}{\kappa E(X)}$$

and queueing time

$$W_q = \frac{L_q}{\kappa E(X)},$$

both of which are measured in units of time since they represent the average delay experienced by a request in the system and in the queue, respectively.

4.2 Special Cases

- No batch arrival, single exponential service, no retrial, no repair, no maintenance, no standby server, no SM2 and no SM3 with $N=1$. $Pr[\mathcal{A}_k = n] = 1$, $R = 1$, $S^*(\kappa + \eta) \rightarrow 1$, $\theta = 0$, $\eta = 0$, $\delta = 0$, $\varphi_2 = 0$, $\varphi_3 = 0$. Accordingly, the proposed model reduces to a $M/M/1$ queue with single vacation. In this case, $K(z)$ can be simplified to the following expression which agrees with the results established in recent works by Gross et al. (2019).

$$K(z) = \frac{P_I \kappa (1-z) [z - 1 - \mathcal{Q}^*(\kappa(1-z)) [1 - \varpi_1 (1 - \mathcal{L}_1^*(\kappa(1-z)))] [1 + \kappa^2 (1-z) + \kappa z]]}{\kappa (1-z) [z - [1 + \kappa^2 (1-z)] [\mathcal{Q}^*(\kappa(1-z)) [1 - \varpi_1 (1 - \mathcal{L}_1^*(\kappa(1-z)))]]]}$$

$$K(z) = \frac{P_I \kappa (1-z) [z - 1 + \kappa \mathcal{Q}^*(\kappa(1-z)) (z - \kappa(1-z))]}{\kappa (1-z) [z - \mathcal{Q}^*(\kappa(1-z)) \kappa^2 (1-z)]}$$

- No batch arrival, No batch service, no repair, no maintenance, no standby server, no sleep modes with $N=1$. $Pr[\mathcal{A}_k = n] = 1$, $R = 1$, $\theta = 0$, $\eta = 0$, $\delta = 0$, $\varphi_1 = 0$, $\varphi_2 = 0$, $\varphi_3 = 0$. Then, the proposed model reduces to an $M/G/1$ retrial queue the resulting expression for $K(z)$ is consistent with the results reported in the recent monograph on retrial queues by Templeton and Falin (2023).

$$K(z) = \frac{P_I \kappa (1-z) [z + S^*(\kappa + \eta) [1 - \mathcal{Q}^*(\kappa(1-z))] + \sigma \mathcal{Q}^*(\kappa(1-z)) \kappa [z - \kappa(1-z)]]}{\kappa (1-z) [z - \mathcal{Q}^*(\kappa(1-z)) [S^*(\kappa + \eta) + \kappa^2 (1-z) \sigma]]}$$

- No batch arrival, single exponential service, no retrial, no repair, no maintenance, no standby server, no sleep modes with $N=1$. $Pr[\mathcal{A}_k = n] = 1$, $R = 1$, $\theta = 0$, $\eta = 0$, $\delta = 0$, $\varphi_1 = 0$, $\varphi_2 = 0$, $\varphi_3 = 0$. Then, the proposed model becomes a simple $M/M/1$ queue whose $K(z)$ (PGF) agrees with the outcome given recently by Selvamuthu (2025) and the classical formulation given by Gross et al. (2019).

5 Numerical Illustrations

To validate the analytical findings and demonstrate the impact of various system parameters on performance, a series of numerical experiments were conducted. The results are presented in Tables 4, 5, and 6, which examine the influence of distinct system parameters on key performance indicators such as the average number of customers, average waiting time of customers and steady-state probabilities associated with various server modes. These tabulated values clearly reveal the sensitivity of the dynamic behavior of the proposed thAP system under different configurations.

Table 4 The influence of maintenance rate (η) on W_s , W_q , $P_{S,1}$, P_A

Maintenance Rate	W_s	W_q	$P_{S,1}$	P_A
1.6	0.0237	0.0319	0.0479	0.0137
1.7	0.0241	0.0324	0.0454	0.0130
1.8	0.0246	0.0329	0.0434	0.0124
1.9	0.0250	0.0335	0.0418	0.0119
2.0	0.0256	0.0342	0.0404	0.0115

Table 5 The influence of service rate (β) on L_s , L_q , P_R , P_I

Service Rate	L_s	L_q	P_R	P_I
0.1	0.9579	0.8823	0.0044	0.0069
0.2	0.7168	0.6476	0.0046	0.0083
0.3	0.5351	0.4756	0.0048	0.0097
0.4	0.4180	0.3706	0.0048	0.0111
0.5	0.3624	0.3286	0.0049	0.0124

As the maintenance rate (η) increases from 1.6 to 2.0, the system and queue waiting times (W_s , W_q) increase due to more frequent interruptions which is illustrated in Table 4. The steady state probability of being in light sleep ($P_{S,1}$) and active mode (P_A) both decrease, as the tHAP spends more time under maintenance. The system experiences more frequent pauses in service, which reduces the opportunity for completing batches and initiating sleep transitions. Because requests continue to arrive even during maintenance, more users remain in the system for longer periods, leading to a gradual rise in the waiting times W_s and W_q . At the same time, ($P_{S,1}$) and (P_A) both decrease, since maintenance occupies a larger portion of the operational cycle and leaves fewer intervals where service or sleep-mode switching can occur. These observations are obtained for the parameter settings $\kappa = 5$, $\delta = 7$, $\theta = 6$, $\varphi_1 = 0.9$, $\mu = 6.5$, $\beta = 4$, while varying only the maintenance rate (η).

In Table 5, with a higher service rate (β) from 0.1 to 0.5, the system and queue sizes (L_s , L_q) decrease due to faster processing. However, the steady state probability of retrial (P_R) and the chance of no users (P_I) increase, since requests are cleared quicker. A higher (β) allows each batch to be completed more quickly, so fewer customers accumulate in the system, resulting in reduced L_s and L_q . Because service finishes sooner, the server reaches periods with no pending requests more often, which leads to a gradual rise in the idle probability (P_I). The slight increase in (P_R) reflects the fact that orbiting users find the server free more frequently and attempt service sooner. Further, these results are generated under the parameter values $\kappa = 5$, $\delta = 7$, $\theta = 6$, $\eta = 0.3$, $\varphi_2 = 0.7$, $\varphi_3 = 0.8$, with variation applied solely to the service rate (β).

In Table 6, as breakdown rate (θ) increases, system and queue sizes (L_s , L_q) and waiting time (W_s , W_q) all increase, because frequent failures lead to more waiting and congestion. Here, the system experiences more frequent interruptions in service, which prevents batches from completing smoothly. As a result, customers remain in the system and orbit for longer durations, leading to larger values of L_s and L_q . Because each breakdown forces the server into a non-operational state while arrivals and retrials continue, the overall waiting times W_s and W_q also grow substantially. The outcomes are presented for the parameter configuration $\kappa = 5$, $\delta = 7$, $\mu = 6.5$, $\beta = 4$, $\varphi_1 = 0.9$, $\varphi_3 = 0.8$, where only the breakdown rate (θ) is varied.

Table 6 The influence of breakdown rate (θ) on L_s , L_q , W_s , W_q

Breakdown Rate	L_s	L_q	W_s	W_q
6	0.1938	0.1887	0.8560	0.8247
7	0.4659	0.4770	1.0260	0.9888
8	0.7570	0.7718	1.3666	1.3210
9	1.0113	0.9747	2.1466	2.0966
10	1.2126	1.1683	3.1609	3.2985

Figure 3 includes 2D plots that visualize the relationship between individual system parameters and performance indicators. These plots help demonstrate trends and monotonic behavior as key inputs vary.

Figure 3(a) depicts that as the maintenance rate (η) increases, both waiting times (W_s) and (W_q) show a steady upward trend. At the same time, the tHAP becomes less active (P_A) and spends less time in light sleep ($P_{S,1}$) mode. This pattern follows naturally, (i.e.,) as the maintenance rate (η) increases, the tHAP enters the maintenance state more frequently, reducing the proportion of time available for active service and sleep transitions. Since no service is provided during maintenance, arriving and retrial requests continue to accumulate in the orbit, which results in higher W_s and W_q). The decrease in P_A occurs because frequent maintenance interrupts the service process and limits the opportunities for initiating new batches. Likewise, $P_{S,1}$ declines, as sleep transitions can occur only after a completed service, and a higher maintenance rate reduces the number of such opportunities.

Figure 3(b) depicts that as the service rate (β) increases, the plots for system size (L_s) and queue length (L_q) slope downward. This reflects smoother, faster processing reducing congestion. Interestingly, since the system becomes more efficient, P_R and P_I slightly increase suggesting more frequent periods where the server is waiting for work. This behaviour is consistent with the model structure, as the service rate (β) increases, the tHAP clears requests

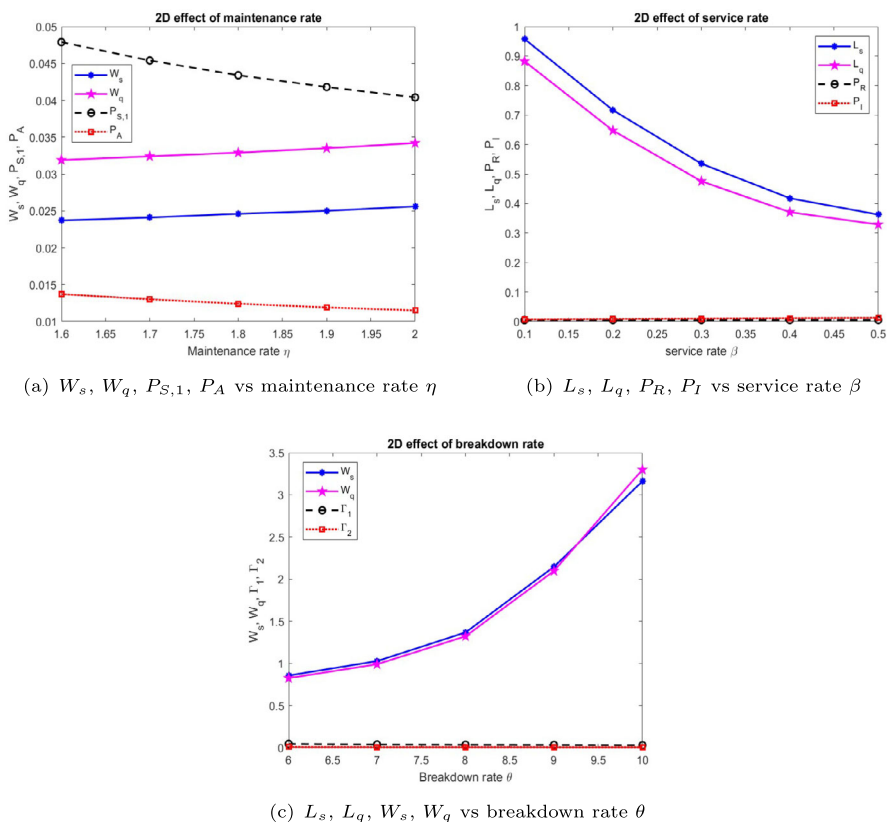


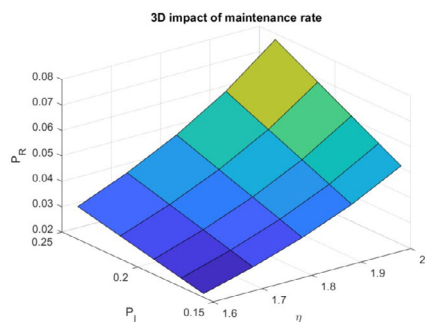
Fig. 3 Visualization of graph data in two dimensions

more quickly, which reduces both L_s and L_q . Faster service lowers congestion and shortens customer waiting times. At the same time, once batches are completed more rapidly, the server experiences more intervals where it has no pending work, which explains the slight increase in P_I and P_R . Consequently, a higher service rate reduces the system load and increases the likelihood that arriving or retrial customers encounter an idle server.

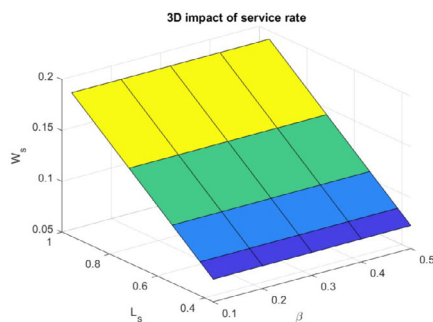
Figure 3(c) depicts that as the breakdown rate (θ) increases, there is a sharp increase in system size (L_s) and queue length (L_q) along with waiting time of the system (W_s) and queue (W_q). This highlights how system failures severely impact performance more breakdowns mean more users in queue and longer delays. This outcome is aligned with the model dynamics, as θ increases, the tHAP moves into the repair state more frequently, during which no service can be offered. However, arrivals and retrials continue, causing requests to accumulate in the orbit. This leads to larger values of L_s and L_q , along with noticeably longer waiting times W_s and W_q . The growth in congestion is expected, as higher breakdown frequency reduces the overall effective service availability and increases the periods during which the system is unable to process requests.

Figure 4 shows 3D surface plots illustrating the joint effect of two parameters simultaneously. These graphs provide a more comprehensive view of system sensitivity and interaction effects among parameters.

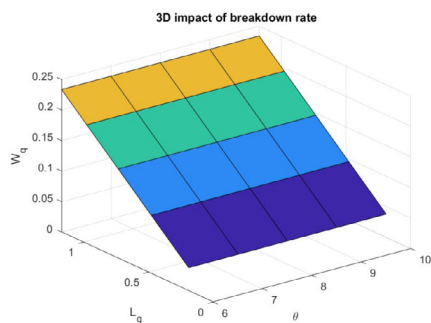
Figure 4(a) shows how both P_I and P_R rise as maintenance rate increases. The plot forms a rising surface, visually emphasizing how higher maintenance slows the system down, reducing efficiency. The rising surface shows that when maintenance becomes more frequent,



(a) P_I, P_R vs maintenance rate η



(b) L_s, W_s vs service rate β



(c) L_q, W_q vs breakdown rate θ

Fig. 4 Graphical depiction in 3D space

the system exhibits a greater tendency to move in and out of non-service states, creating more opportunities for idle periods to appear once maintenance concludes. At the same time, incoming requests are more likely to reattempt access because the server's availability becomes less predictable. The 3D trend therefore reflects how increased maintenance activity disrupts the natural flow of service and produces more fluctuations between idle and retrial events.

Figure 4(b) displays a descending surface where L_s and W_s both decrease with increased service rate. It shows the combined benefit of faster processing on queue length and delay. Note that, the downward sloping surface indicates that higher processing capability compresses the overall workload more effectively, keeping both the queue content and the time spent in the system at consistently lower levels. Rather than capturing only the direct effect of β , the surface demonstrates how reductions in workload and delay occur simultaneously across different operating conditions. This joint behaviour emphasizes the stabilizing influence of faster service in scenarios with fluctuating traffic intensity.

Figure 4(c) reveals that as breakdown rate increases, L_q and W_q rise sharply together. The surface curves upward, illustrating the heavy impact of breakdowns on user experience. Further, this upward surface demonstrates that even moderate increases in θ amplify the pressure on the system, as interruptions create pockets of unserved work that build up rapidly. The surface representation highlights that the adverse effects are not merely linear or additive; longer queues and extended waiting times tend to reinforce each other as breakdowns become more frequent. This combined escalation is evident in the steepening of the surface for larger values of θ .

6 Evaluation of Energy Conservation

6.1 Power Savings Factor

Power Saving Factor (PSF) is defined as the percentage of total operational time during which the tHAP resides in energy-saving sleep modes. By using the Eqs. 45 - 47, we obtain the following,

$$PSF = (P_{S,1} + P_{S,2} + P_{S,3}) \times 100\%$$

6.2 Expected Energy Consumption in the Proposed Model

$E_{proposed}$ refers to the total average energy consumed per unit time by a tHAP under an energy-aware architecture that includes active operation, idle periods, sleep states, mainte-

Table 7 Assumed power consumption across tHAP states

Server states	Values J/s
E_{Active}	234.2
E_{Idle}	38.2
E_{SM1}	25.5
E_{SM2}	2.9
E_{SM3}	2
E_{Main}	30
E_{Repair}	120

nance, and repair modes. It quantifies the long-term power usage considering both traffic dynamics and energy-saving mechanisms.

Consequently, the average energy consumption of the tHAP over a given time interval for the proposed model can be approximated using the following formulation. Furthermore, Table 7 presents the assumed energy consumption values across various tHAP operational states, based on the findings in Liu et al. (2015); Niu et al. (2015); Lahdekorpi et al. (2017). Also, all energy terms are measured in J/s, representing average power consumption.

$$E_{proposed} = E_{Active}P_A + E_{Idle}P_I + E_{SM1}P_{S,1} + E_{SM2}P_{S,2} + E_{SM3}P_{S,3} + E_{Main}P_M + E_{Repair}P_B, \quad (48)$$

where,

- E_{Active} = The energy consumed when the tHAP is fully operational and serving the requests, involving all major subsystems like transceivers, baseband units, and cooling systems.
- E_{Idle} = The energy drawn when the tHAP is powered on but not currently serving any requests a moderate consumption state with readiness to transition.
- $E_{SM1}, E_{SM2}, E_{SM3}$ = The energy drawn during light, moderate, and deep sleep modes respectively, where increasing sleep depth progressively shuts down non-essential components, yielding significant energy savings.
- E_{Main} = The energy consumed during PM, where the tHAP remains partially active for diagnostic checks and software-level servicing. Since only essential control and monitoring units are operational, the power draw is modest. A value that is slightly below idle mode is assumed, reflecting minimal hardware usage.
- E_{Repair} = The energy consumed when the tHAP undergoes hardware fault recovery. Though user traffic is suspended, internal testing modules, control logic, and possibly the standby server are active, resulting in moderate power usage. A value which is substantially lower than active mode but higher than idle is assumed, accounting for partial system engagement during repair.

6.3 Expected Energy Consumption Percentage

Expected Energy Conservation Percentage ($E_{percent}$) measures the proportion of energy saved by the system compared to an always-on baseline scenario. It represents the overall efficiency gain achieved through power-saving mechanisms in percentage terms. Thus, from Eq. 48 we obtain,

$$E_{percent} = \left(1 - \frac{E_{Proposed}}{E_{Baseline}}\right) \times 100$$

where, $E_{Baseline}$ = represents the expected energy consumption of the tHAP without any sleep mechanisms, i.e., in a traditional always-on setting.

In this baseline model:

The tHAP never enters sleep modes,

All idle or low-traffic periods are treated as idle time (consuming idle-state power),

Sleep mode probabilities are assumed to be zero,

The tHAP cycles only between active, idle, maintenance, and repair states.

6.4 Expected Delay

In queueing theory, expected delay (also known as sojourn time) refers to the average time a customer or request spends in the system from the moment it arrives until its service is completed. This includes all time spent waiting in the queue, or activation delays, service time, and additional delays due to system behaviors such as sleep transitions or maintenance periods.

In the context of the proposed 5G tHAP with dynamic sleep control, the expected delay becomes a composite measure of multiple components as mentioned below,

$$E_{delay} = E_{Qdelay} + E_{Sdelay} + E_{SMdelay} + E_{Mdelay}$$

where,

- E_{Qdelay} = Average waiting time in the queue before service begins = $\frac{\kappa E(X_2)E(Q^2)}{2(1-\kappa E(X)E(Q^1))}$.
- E_{Sdelay} = Delay due to batch service, as service begins only when a fixed-size batch is formed
 $= \sum_{r=1}^R P(R=r)E(Q_1)$.
- $E_{SMdelay}$ = Delay incurred due to the tHAP being in a sleep state (light, moderate, or deep)
 $= \frac{\varpi_1 Q^*(\theta) [S^*(\kappa+\eta)\kappa E(X)[1-\eta E(P)]] [E(L_1)+\varpi_2(E(L_2))+\varpi_3(E(L_3))]}{R-[S^*(\kappa+\eta)+\kappa\sigma][\zeta' Q^*(\theta)-\kappa E(X)E(Q)-\theta E(Q)]-\eta\sigma[\kappa E(X)E(P)+R]}$.
- E_{Mdelay} = Downtime delay due to PM events
 $= \frac{\eta S^*(\kappa+\eta)E(P)[R-\zeta' Q^*(\theta)-\kappa E(X)E(Q)-\theta E(Q)]}{R-[S^*(\kappa+\eta)+\kappa\sigma][\zeta' Q^*(\theta)-\kappa E(X)E(Q)-\theta E(Q)]-\eta\sigma[\kappa E(X)E(P)+R]}$.

6.5 Proposed System Throughput

In a batch service system, throughput is defined as the average number of requests served per unit time, considering that services are completed in batches rather than individually.

$$Throughput = P_A \times \frac{r}{E(Q_1)}.$$

6.6 Numerical Observation

To assess the efficiency of the proposed tHAP model, a numerical analysis is performed focusing on key performance indicators such as energy consumption and expected delay. The analysis aims to capture the dynamic behavior of the system under realistic operating conditions and quantify the potential benefits of energy-saving mechanisms.

The data in Table 8 illustrates the impact of increasing sleep transition probabilities $(\varpi_1, \varpi_2, \varpi_3)$, corresponding respectively to light, moderate, and deep sleep states, on key

Table 8 Effect of sleep transition probabilities $(\varpi_1, \varpi_2, \varpi_3)$

Sleep Mode	L_s	L_q	W_s	W_q
(0.2,0.3,0.4)	2.9496	1.7920	0.2705	0.2500
(0.4,0.5,0.6)	3.6963	2.9632	0.2857	0.3375
(0.6,0.7,0.8)	3.5003	3.0301	0.3390	0.4580

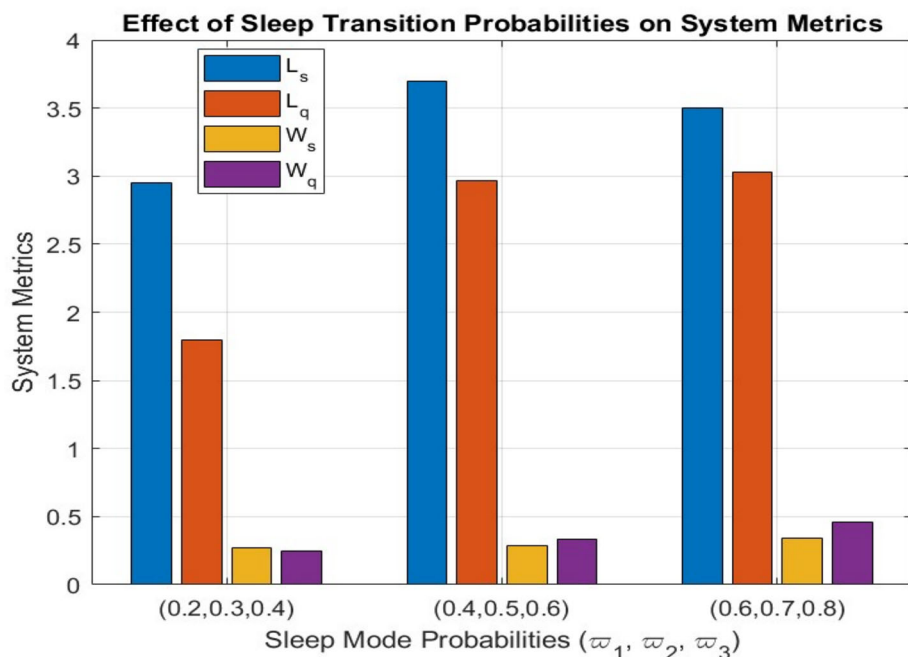


Fig. 5 Effect of sleep transition probabilities

system performance measures such as L_s , L_q , W_s , and W_q . When the sleep-transition probabilities increase, the system becomes more likely to enter the deeper sleep states, particularly the SM2 to SM3 pathway. Since these deeper sleep modes have longer mean sojourn times and larger wake-up delays, the server spends a greater proportion of time unavailable for processing requests. This reduces the effective service capacity and increases the number of arrivals and retries that accumulate while the server is inactive. Consequently, the queue and orbit grow, leading to larger values of W_q and W_s . In addition, deeper sleep modes introduce longer contiguous periods of unavailability, which amplifies delay through increased variability in service opportunities. Overall, the rise in waiting time is primarily driven by the higher probability of entering SM3, which imposes longer wake-up delays and lowers short-term service availability.

Figure 5 illustrates the response of queue length and delay metrics to increasing sleep-transition probabilities. As the heights of the bars increase, it becomes evident that higher transition probabilities drive the system more frequently into deeper low-power states, highlighting the effectiveness of the sleep controller in achieving energy savings. At the same time, the visual trend shows that prolonged residence in deeper sleep modes leads to longer wake-up delays and a reduction in immediate service availability. Consequently, requests accumulate in the system, resulting in increased queue lengths (L_s and L_q) and waiting times (W_s and W_q). This graphical representation clearly emphasizes the inherent trade-off between improved energy efficiency and degraded delay performance, indicating that sleep-transition probabilities must be carefully tuned to maintain an acceptable quality of service.

Table 9 examines how system performance changes as the depth of sleep modes increases—from only SM1, to SM1+SM2, and finally SM1+SM2+SM3. As deeper modes are added, the

Table 9 Impact of sleep mode depth on performance metrics

Sleep Mode	L_s	W_s	PSF%
SM1	3.9264	0.2398	28.66
SM1+SM2	4.1704	0.3547	34.74
SM1+SM2+SM3	5.9749	0.4256	52.13

tHAP spends longer cumulative time in low-power states, since transitions into SM2 and SM3 involve progressively longer sleep durations. This reduces immediate service availability. Consequently, the system size L_s increases because requests accumulate while the server remains in deeper sleep states, and the system time W_s also increases due to longer wake-up delays and additional sleep-to-active transitions before service can resume.

At the same time, the Power Saving Factor (PSF%) improves substantially—rising from 28.66% under SM1 to 52.13% when all three sleep modes are enabled—because deeper sleep modes turn off more subsystems for longer intervals. Thus, the observed increase in L_s and W_s is driven not by traffic intensity, but by extended suspension of service caused by deeper sleep stages, particularly SM3, which has the slowest reactivation. The corresponding gain in PSF highlights the inherent energy–delay trade-off of multi-level sleep strategies.

Figure 6 presents a bar-graph comparison of system size (L_s), delay (W_s), and (PSF%) across increasing sleep modes. The bars clearly illustrate that as additional sleep modes are introduced, the bars for (L_s) and (W_s) rise progressively, reflecting the longer wake-up paths and reduced immediate service availability associated with deeper sleep states. In contrast, the PSF% bar increases sharply with each added sleep layer, visually emphasizing the substantial

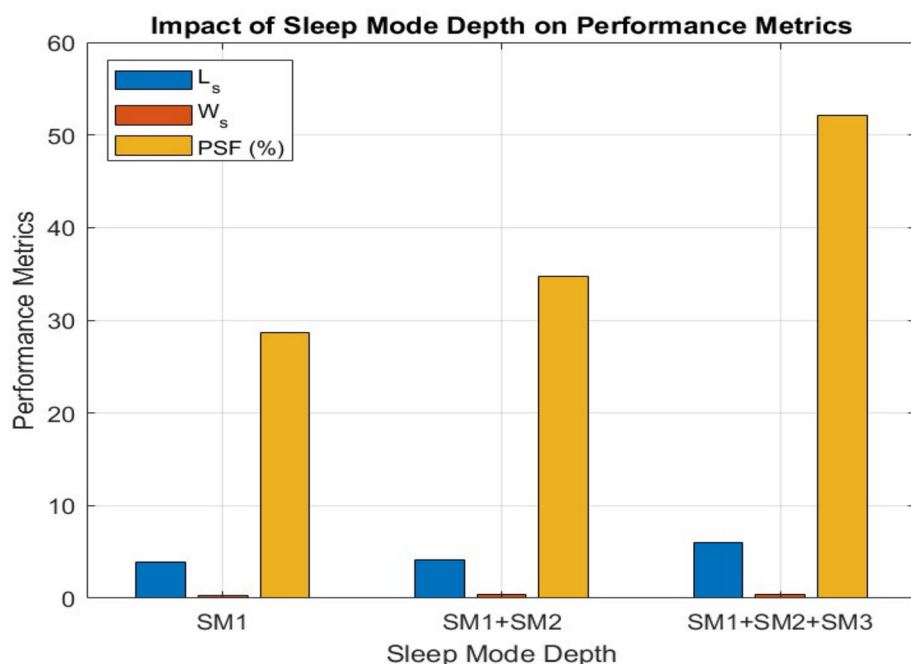
**Fig. 6** Effect of sleep mode depth on performance metrics

Table 10 Energy consumption comparison for proposed and baseline models

Sleep Mode	$E_{proposed}$ (with sleep)	$E_{baseline}$ (without sleep)	$E_{percent}$
SM1	5.1466	5.9650	16.53
SM1+SM2	3.6275	4.7597	23.78
SM1+SM2+SM3	2.8100	3.9276	28.46

energy benefits gained when the tHAP spends more cumulative time in low-power modes. The bar pattern highlights the central trade-off: deeper sleep modes lead to markedly higher energy efficiency but also increased delay and temporary build-up of queued work. Thus, the bar graph reinforces the need for an efficient balance between energy savings and QoS performance when selecting the appropriate sleep-mode depth.

Table 10 presents a comparative evaluation of energy consumption between the proposed sleep-enabled tHAP model ($E_{proposed}$) and a baseline architecture operating without any sleep mechanisms ($E_{baseline}$). Across all three configurations - SM1, SM1+SM2, and SM1+SM2+SM3- the proposed model consistently exhibits lower energy usage. This reduction arises because the tHAP in the proposed framework actively transitions into light, moderate, and deep sleep modes during low-traffic periods, allowing several power-hungry subsystems (transceiver chain, cooling units, baseband processors) to shut down or operate at minimal levels. In contrast, the baseline model is forced to remain either active or idle at all times, resulting in sustained energy draw even when no requests are present.

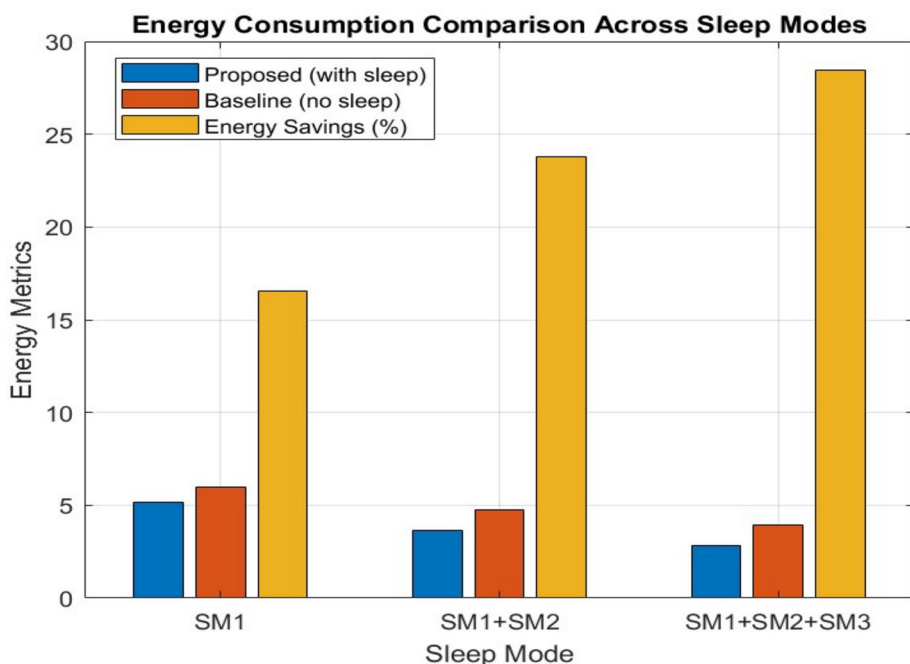
**Fig. 7** Energy consumption comparison

Table 11 Impact of sleep mode depth on expected delay

Sleep Mode	E_{delay}	Throughput
SM1	0.4559	0.2894
SM1+SM2	2.2506	0.1016
SM1+SM2+SM3	3.7676	0.0636

As deeper sleep modes are incorporated, the energy percent of the proposed model becomes more pronounced. When only SM1 is enabled, the proposed system already achieves a reduction of 16.53% in energy consumption compared to the no-sleep baseline. Activating SM2 and SM3 further strengthens this benefit, raising the savings to 28.46%. This widening gap arises because the deeper modes deactivate more subsystems and keep the tHAP in low-power operation for longer durations, thereby reducing the average power drawn by the system. Consequently, the proposed architecture consistently delivers greater energy efficiency, with larger improvements as additional sleep levels are utilised.

Thus, the proposed model achieves meaningful energy conservation because it dynamically adapts its operational state based on demand, whereas the baseline consumes near-constant power regardless of traffic fluctuations.

Figure 7 compares the total energy consumption of the baseline (always-on) model with the proposed multi-sleep tHAP architecture. The bar graph shows a consistent reduction in energy usage under the proposed model, and this gap widens as deeper sleep modes are enabled. This downward shift occurs because multi-level sleep progressively disables more hardware modules, resulting in significant operational power savings. The accompanying energy-saving percentage bar further emphasizes that as the system moves from shallow to

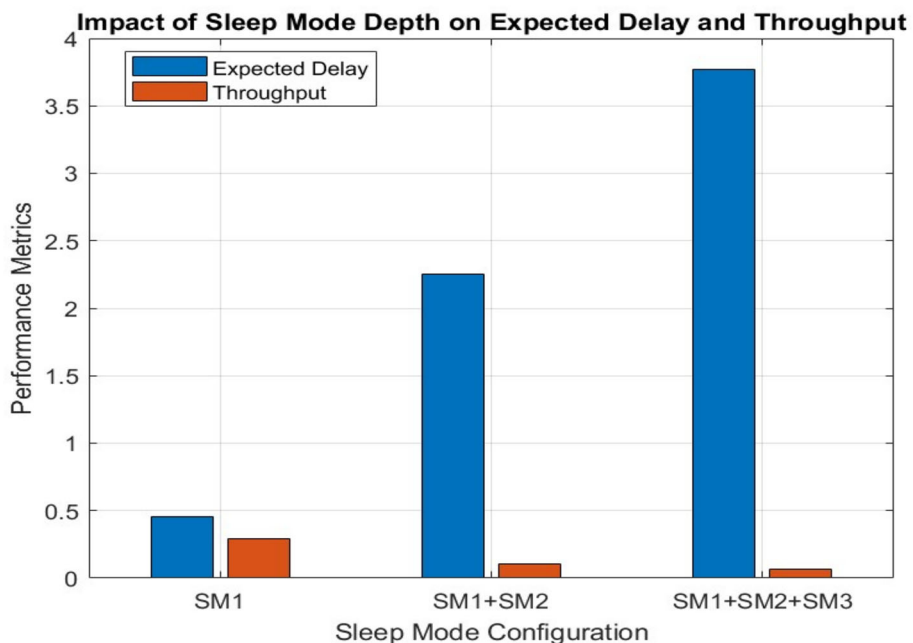
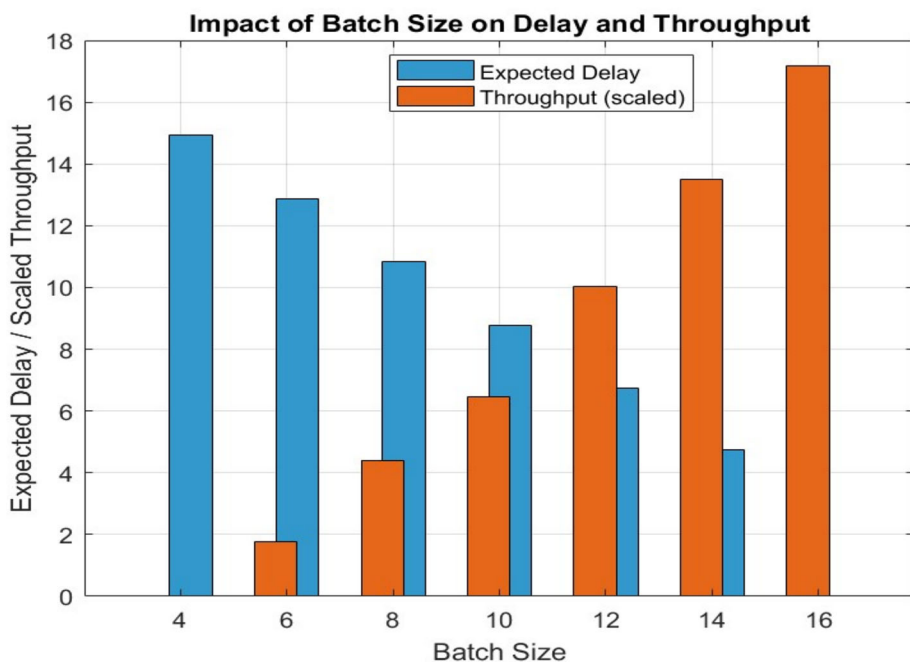
**Fig. 8** Impact of sleep mode depth on expected delay

Table 12 Impact of batch size on delay and throughput

Batch Size	E_{delay}	Throughput
5	14.9457	0.0059
7	12.8809	0.0147
9	10.8249	0.0215
11	8.7886	0.0335
13	6.7566	0.0450
15	4.7316	0.0572

deeper sleep configurations, the percentage of conserved energy increases steadily, demonstrating the strong efficiency gains achieved by dynamic sleep control.

Table 11 evaluates how the expected delay and system throughput change when the tHAP is allowed to use deeper sleep modes. As additional sleep levels are enabled, the server experiences longer inactive intervals before returning to service. These extended off-periods reduce the frequency with which customers can be served, so each request remains in the system for a longer duration, resulting in an increase in E_{delay} . At the same time, deeper sleep modes reduce the number of service completions possible within a given time window, which lowers the effective throughput. This is not due to increased traffic or congestion, but directly caused by the reduction in active-service availability: as the server spends more time in SM2 and especially SM3, fewer service opportunities occur per unit time.

**Fig. 9** Impact of batch size

Thus, Table 11 highlights the classical delay–throughput trade-off: enabling deeper sleep conserves more energy but reduces the short-term service rate, which in turn raises delay and lowers the observed throughput.

Figure 8 illustrates how the expected delay grows as the sleep mode depth increases. The bar graph shows a clear upward progression in delay when moving from shallow to deeper sleep configurations. This is visually intuitive because deeper sleep stages require longer reactivation times, and the bar spacing highlights how these wake-up penalties accumulate. Unlike the table, which reports numerical changes, the bar graph emphasizes the magnitude of delay growth and makes the compounding effect of sleep depth immediately evident. Though deeper modes significantly cut energy consumption, the bars highlight the corresponding rise in delay, reinforcing the key system trade-off between efficiency and responsiveness.

Table 12 demonstrates the impact of varying batch sizes on system performance in terms of expected delay and throughput. As shown in Table 12, increasing the batch size leads to a monotonic decrease in expected delay and a significant increase in throughput. This behavior stems directly from the quorum-based service mechanism: a larger batch allows the tHAP to serve more accumulated requests within a single service cycle, thereby distributing the service time across multiple users. As batch size grows, the server clears orbiting customers more rapidly, reducing repeated retrials and lowering congestion. At the same time, throughput improves because each service completion now contributes a higher number of served requests, even though the service time per batch remains unchanged. Thus, the improvement arises not from faster service, but from more efficient utilization of each service opportunity, particularly under moderate-to-heavy load where batches are readily formed.

Figure 9 illustrates how changes in batch size influence the system's expected delay and throughput. The bar plot shows a clear opposing trend: as the batch size increases, the expected delay steadily decreases, while the throughput rises correspondingly. This visual pattern reflects the operational advantage of processing multiple arrivals together which clear more work per activation, reducing the average waiting time experienced by each individual request. At the same time, servicing more requests per cycle boosts the effective throughput, which is why the orange bars grow consistently with batch size. Overall, the figure highlights that enlarging the batch size enhances system productivity while simultaneously lowering delay, demonstrating a favourable performance gain without additional complexity.

Table 13 compares the system with and without PM. The introduction of PM produces a noticeable improvement in throughput and a slight reduction in expected delay. This stems from the fact that PM proactively prevents catastrophic breakdowns, thereby reducing the time the tHAP spends in the repair state—a mode that typically incurs long downtime and interrupts ongoing service. With fewer breakdown events, the system remains available for active service more frequently, which directly enhances throughput. Additionally, although PM introduces short maintenance intervals, these are substantially less disruptive than unplanned failures; hence, the overall delay decreases marginally. Energy consumption also drops under PM because longer repair periods (which consume moderate power but yield no service) are

Table 13 Impact of predictive maintenance

Predictive Maintenance	<i>Throughput</i>	<i>E_{delay}</i>	<i>E_{proposed}</i>	<i>E_{percent}</i>
Without PM	0.0224	9.5906	3.8792	28.46
With PM	0.0375	9.5834	2.7255	22.95

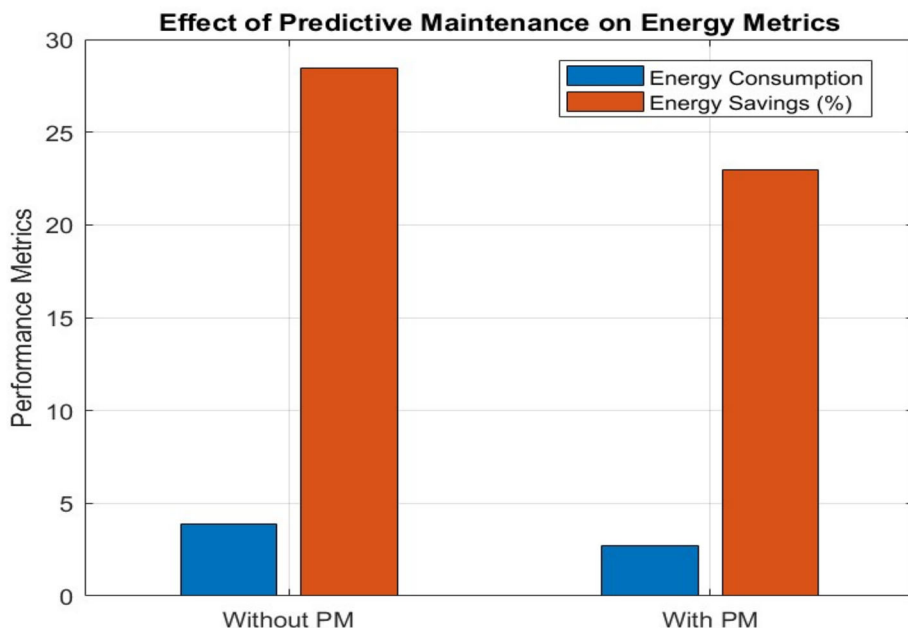


Fig. 10 Impact of predictive maintenance

replaced by shorter, controlled maintenance intervals. As a result, PM reduces wasted operational energy and improves the overall energy-saving percentage of the system.

Figure 10 visually contrasts the system performance with and without PM. The bar heights make it clear that incorporating PM simultaneously lowers the overall energy consumption and moderates the energy-saving percentage. The drop in the blue bar shows that the system expends less energy per unit time when PM is enabled, while the orange bar indicates that the proportion of energy saved remains substantial even though it decreases slightly. This graphical pattern highlights that PM shifts the system toward a more stable and energy-efficient operating regime, with fewer abrupt repair intervals and more controlled operation. Overall, the figure reinforces that PM offers a practical balance, reduced energy usage with consistent savings, thus making the system more reliable and economical.

6.7 Comparative Analysis

To demonstrate the performance improvement offered by the proposed model, a quantitative comparison with the existing model (Priya et al. 2025) is carried out using throughput, average energy consumption, and energy-saving percent metrics under identical system parameters. Figure 11(a)–(c) summarize the comparative results.

The throughput analysis depicted in Fig. 11(a) is performed over the SM3 range 5–9 to emphasize the service-intensive regime. In this region, the proposed model consistently achieves higher throughput than the existing model; for example, at SM3 = 5, the proposed model attains a throughput of 0.095 compared with 0.068 for the existing model, and this trend persists across all SM3 values. This improvement is primarily attributed to the integrated predictive maintenance module, standby server support, batch-service discipline, and retrial-control mechanisms that collectively ensure more efficient utilization of active ser-

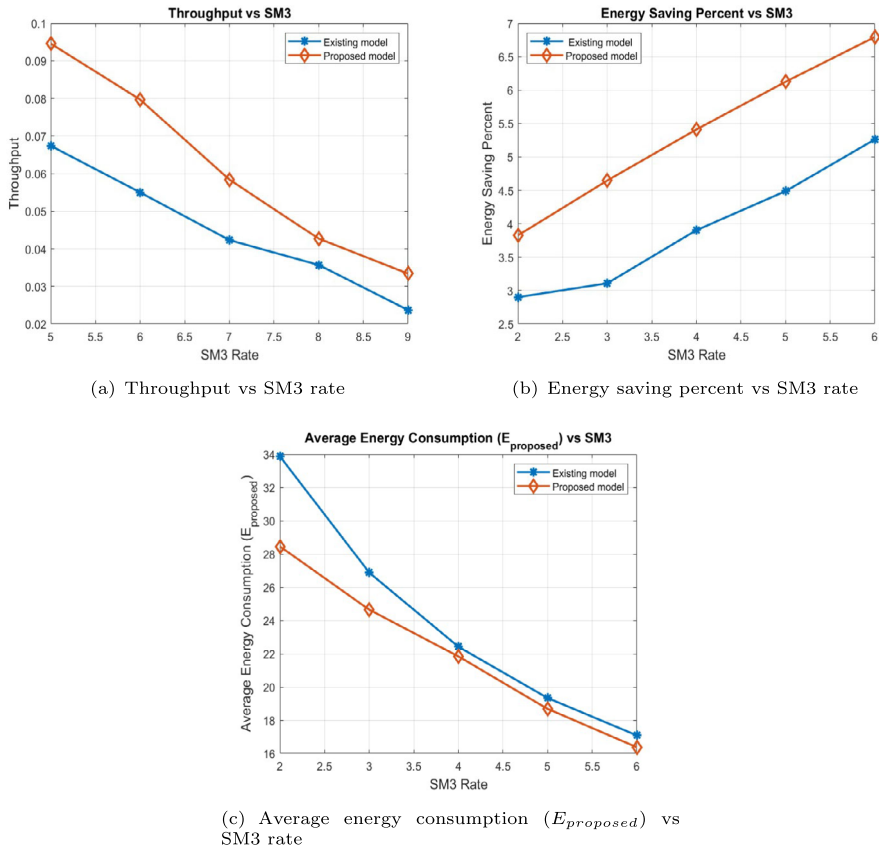


Fig. 11 Comparison between existing model and proposed model

vice periods. These system features reduce service interruptions and maintain smoother traffic flow, enabling the proposed model to outperform the existing model in throughput.

Energy-oriented metrics are evaluated over the SM3 range 2–6, where low-power operation is more critical. The impact of energy-saving percent ($E_{percent}$) presented in Fig. 11(b) shows that the proposed model provides substantially higher energy-efficiency than the existing one. For instance, the savings increase from 3.8% to 6.8% for proposed model as SM3 increases from 2 to 6, whereas the existing model improves only from 2.9% to 5.3% in the same range. Similarly, average energy consumption ($E_{proposed}$) remains lower for the proposed model across all values of SM3 which is depicted in Fig. 11(c). It is clearly noticeable that when $SM3 = 2$, proposed model consumes 28.4 J compared with 33.8 J in the existing model, and at $SM3 = 6$ the consumption reduces to 16.4 J for the proposed model versus 17.1 J for existing model.

Thus, the superior energy-efficiency performance of the proposed model arises from its deeper integration of adaptive sleep strategies and reliability-aware operational controls. The proposed model employs optimized transitions into low-power sleep modes, which substantially reduce the duration of high-energy active states. As a result, even when the system experiences similar traffic conditions as the existing model, the proposed mode consumes less total energy over time. Furthermore, the standby server and predictive maintenance compo-

nents minimize unexpected breakdowns and eliminate unnecessary wake–sleep oscillations, both of which are major contributors to energy wastage in airborne communication platforms. By stabilizing the service process and preventing spurious activations, these mechanisms allow the platform to harvest greater energy savings while simultaneously reducing average energy consumption. Collectively, these design enhancements explain why the proposed model consistently demonstrates higher energy-saving percent and lower average energy consumption than the existing model (Priya et al. 2025).

7 Cost Minimization in Energy-Aware tHAP

Cost optimization is a vital component in the design and management of modern energy-efficient wireless systems, where balancing operational performance with resource consumption is key. In the proposed tHAP model, various system behaviors including retrial traffic, multi-level sleep modes, PM, and repair dynamics introduce complexities that impact overall performance and operating cost. To address this, we formulate a comprehensive system-level cost function that captures the influence of these parameters under steady-state conditions, without combining separate objectives such as energy and delay explicitly.

To efficiently search for optimal configurations that minimize this cost, we employ three powerful nature-inspired metaheuristic techniques: the Whale Optimization Algorithm (WOA), the Artificial Bee Colony (ABC) algorithm, and the Grey Wolf Optimizer (GWO). These algorithms are particularly well-suited for solving complex, non-convex optimization problems due to their global exploration capability, convergence reliability, and minimal dependence on problem-specific gradient information. Through comparative evaluation, the proposed optimization framework demonstrates its effectiveness in fine-tuning key system parameters, leading to enhanced energy efficiency and reduced operational cost.

7.1 Cost Analysis

To evaluate the economic performance of the proposed system, a cost analysis framework is developed to determine the most efficient values for key service parameters, namely γ_b and γ_v . The total cost is modeled as a function comprising multiple operational components, each reflecting the cost incurred by the tHAP's different activities over time. These components such as service operation, sleep states, maintenance, and repair are considered on a per-unit-time basis. The cost structure assumes linearity, which facilitates straightforward optimization and provides insight into how parameter tuning impacts overall system expenditure.

$\mathcal{G}_h$ - The cost incurred per unit time for holding each request within the tHAP.

$\mathcal{G}_b$ - Per-unit-time expense when the tHAP operates under standard active conditions.

$\mathcal{G}_v$ - Per-unit-time expense when the tHAP is under sleep modes.

$\mathcal{G}_m$ - Per-unit-time expense when the tHAP is under maintenance

$\mathcal{G}_f$ - Per-unit-time expense when the tHAP is under maintenance.

$\mathcal{G}_1$ - Cost associated with serving a request during the tHAP's busy periods.

$\mathcal{G}_2$ - Cost associated with retaining a request during the tHAP's sleep modes.

Consequently, the expected cost function takes the form:

$$TC = \mathcal{G}_h L_q + \mathcal{G}_b P_A + \mathcal{G}_v (P_{S,1} + P_{S,2} + P_{S,3}) + \mathcal{G}_m P_M + \mathcal{G}_f P_B + \mathcal{G}_1 \gamma_b + \mathcal{D}_2 \gamma_v. \quad (49)$$

The substantial non-linearity of the cost function in Eq. 49 makes it analytically intractable. Therefore, recognizing that the total cost depends on the service rates (γ_b, γ_v) , we resort to a meta-heuristic approach for optimization.

7.2 Performance-Centric Cost Tuning using WOA, ABC, and GWO

In the context of energy-efficient tHAP operations, cost optimization plays a pivotal role in balancing service performance with resource utilization. Rather than optimizing for individual performance metrics, this approach seeks to determine the best service rate parameters specifically γ_b^* for the active mode and γ_v^* for the low-power (sleep) mode that minimize the total operational cost of the system. By formulating the problem as a constrained minimization task, the model captures how parameter tuning influences the economic and energy efficiency of the tHAP under varying traffic and system dynamics.

To solve this nonlinear and analytically intractable optimization problem, three nature-inspired metaheuristic algorithms as mentioned above are employed. Each algorithm explores the solution space to find the optimal pair (γ_b^*, γ_v^*) that yields the minimum total cost.

Mathematically, the cost-minimization problem takes the form:

$$TC(\gamma_b^*, \gamma_v^*) = \underset{\gamma_b^*, \gamma_v^*}{\text{Minimize}} TC(\gamma_b, \gamma_v).$$

To analyze how the cost function responds to varying parameters, the relevant cost values used are summarized in Table 14.

7.3 Formulation of the Optimization Problem

In the optimization study, Eq. 49 is used as the objective function, where the decision variables are γ_b and γ_v . These variables directly influence the cost components related to delay and energy consumption. The search space for both γ_b and γ_v is restricted to the interval [2, 8] to ensure that the optimization is carried out within a feasible and operationally meaningful range. Values smaller than 2 correspond to unrealistically short sleep or vacation periods that provide negligible energy benefits, while values larger than 8 lead to excessively long inactive durations that significantly increase delays. Furthermore, choosing this interval guarantees that the stability condition $\rho < R$ is satisfied for all feasible parameter values, ensuring that the system load remains strictly below the effective service capacity during the optimization. Thus, the interval [2, 8] provides a balanced and safe region in which Eq. 49 is meaningfully minimized while preserving system stability. Consequently, the metaheuristic search determines the optimal service-rate parameters γ_b and γ_v , which minimize the total operational cost while satisfying all the feasibility and stability constraints.

Table 14 Distinct cost profiles considered for analytical comparison

Cost sets	$\mathcal{G}_h$	$\mathcal{G}_b$	$\mathcal{G}_v$	$\mathcal{G}_m$	$\mathcal{G}_f$	$\mathcal{G}_1$	$\mathcal{G}_2$
Set 1	95	25	15	55	20	15	12
Set 2	75	45	25	25	10	25	15
Set 3	115	15	20	30	5	30	20

7.4 Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA) is a bio-inspired metaheuristic proposed by Mirjalili and Lewis (2016) that mimics the bubble-net hunting behavior of humpback whales. It balances exploration and exploitation through encircling, spiral updating, and search for prey, making it effective for solving nonlinear and multidimensional optimization problems. Nadimi-Shahraki et al. (2023) gave a systematic approach on WOA.

In the case of WOA, system parameters such as $\omega = 2.5$, $\delta = 4$, $v_3 = 9$, $\varpi_1 = 0.25$, $\varpi_2 = 0.5$, $\varpi_3 = 0.25$, $\xi = 3.3$, and $\mu = 13$ are fixed while optimizing the overall cost. The decision variables are restricted to the interval (Aggarwal et al. 2025; Daraseliya et al. 2025), and the optimization is executed for 150 iterations with a population size of 75, resulting in a total of 11,250 function evaluations. The search is performed in a two-dimensional space corresponding to (γ_b, γ_v) , with initial candidate solutions sampled uniformly within the feasible range. The main WOA control parameter decreases linearly from 2 to 0 during the search to gradually shift the algorithm from exploration to exploitation. At each iteration, candidate solutions are updated using the standard encircling and bubble-net mechanisms, and the fitness of each whale is evaluated using the objective function in Eq. 49. Any candidate solution that moves outside the allowable range is automatically adjusted to the nearest valid boundary. A convergence curve is recorded by storing the best cost value at every iteration. The algorithm terminates when the maximum iteration count is reached or when no noticeable improvement in the objective value is observed, and the final outputs are the optimal values (γ_b^*, γ_v^*) and the minimum cost.

Algorithm 1 outlines the procedural steps of the WOA algorithm in pseudocode form, while Table 15 summarizes the impact of parameters (κ, η, θ) on the minimum total cost and optimal service rate (γ_b^*, γ_v^*) employing WOA.

Algorithm 1 Pseudo Code for WOA Algorithm.

```

Initialize whale population randomly within the search space.
Evaluate the fitness of each whale.
Identify the best solution (leader).
for every iteration do
    Update c, C, D, m and q
    for each whale do
        if q < 0.5 then
            if C < 1 then
                Update position using encircling strategy.
            else if C ≥ 1 then
                Move randomly toward another whale (exploration).
            end if
        else if q ≥ 0.5 then
            Update position using spiral movement (exploitation).
        end if
    end for
    Evaluate new positions and update the best solution if improved.
end for
Return the best solution found.

```

Table 15 Impact of κ , η , θ on $(TC, \gamma_b^*, \gamma_v^*)$ using WOA

Parameters	$(TC, \gamma_b^*, \gamma_v^*)$		
	Cost set 1	Cost set 2	Cost set 3
κ	0.5 (35.4404,2.1770,2.9923)	(69.1882,1.7523,2.8664)	(83.7735,1.9102,2.7403)
	0.7 (38.4624,2.2572,2.9192)	(74.7859,1.8323,2.7983)	(93.8934,2.0315,2.6080)
	0.9 (40.2894,2.3839,2.7850)	(81.0015,1.9042,2.7162)	(104.6975,2.1455,2.4505)
η	11 (47.2833,2.3630,2.8032)	(69.1225,1.7547,2.8741)	(86.7321,1.9378,2.7090)
	13 (47.1808,2.3799,2.7942)	(69.0378,1.7660,2.8754)	(86.6553,1.9383,2.7342)
	15 (47.1071,2.3771,2.8013)	(68.9489,1.7761,2.8868)	(86.6292,1.9482,2.7386)
θ	5 (47.3524,2.3620,2.8004)	(71.8884,1.7935,2.8347)	(86.0791,1.9105,2.6704)
	6 (45.4617,2.3346,2.8146)	(71.6885,1.7793,2.8129)	(85.7830,1.9027,2.6352)
	7 (44.2915,2.3184,2.8244)	(71.6201,1.7695,2.7958)	(85.6795,1.9032,2.5979)

7.5 Artificial Bee Colony Optimization (ABC)

The Artificial Bee Colony (ABC) algorithm is a swarm intelligence-based optimization technique developed by Karaboga and Basturk (2007). It simulates the foraging behavior of honey bees in search of high-quality food sources. The bee colony is typically divided into three types: employed bees, onlooker bees, and scout bees, each contributing differently to exploration and exploitation. Employed bees exploit known food sources, onlooker bees select food sources based on shared information, and scout bees explore new potential regions randomly. ABC is particularly known for its strong exploration capabilities and robustness in handling complex, multi-modal optimization problems. Recently, Mathavavisakan and Indhira (2025), combined ABC and neuro-fuzzy systems to minimize cost in a batch arrival retrial queue with working vacations.

In the case of ABC, the number of employed bees and onlooker bees is set to 37 each (half of the total population size of 75), and the number of scout bees is determined adaptively based on the abandonment criteria. System parameters such as $\delta = 4$, $\omega = 2.5$, $v_3 = 9$, $\varpi_1 = 0.25$, $\varpi_2 = 0.5$, $\varpi_3 = 0.25$, $\xi = 3.3$, and $\mu = 13$ are fixed while optimizing the overall cost. The abandonment limit is fixed at 100, and the acceleration coefficient is set to 2, which controls the neighbourhood search around each food source. The optimisation is carried out over 100 iterations, with the decision variables restricted to the interval (Aggarwal et al. 2025; Daraseliya et al. 2025). The search is performed in a two-dimensional space corresponding to the decision vector (γ_b, γ_v) , and all initial candidate solutions are generated uniformly within this feasible domain. Across the 100 iterations and 75 bees, the algorithm performs a total of $75 \times 100 = 7,500$ objective-function evaluations. At each iteration, the fitness associated with each candidate is calculated from the objective value in Eq. 49 using an exponential scaling, and onlooker bees select food sources via a roulette-wheel selection mechanism based on these fitness values. Any solution that fails to improve for 100 consecutive trials is abandoned and replaced by a scout bee through random reinitialisation within the same bounds. A convergence curve is recorded by storing the best cost value at every iteration. Upon termination, the algorithm returns the minimum cost obtained along with the corresponding optimal parameters (γ_b^*, γ_v^*) .

Algorithm 2 outlines the procedural steps of the ABC algorithm in pseudocode form, while Table 16 summarizes the impact of parameters (κ, η, θ) on the minimum total cost and optimal service rate (γ_b^*, γ_v^*) employing ABC.

Table 16 Impact of κ , η , θ on $(TC, \gamma_b^*, \gamma_v^*)$ using ABC

Parameters	$(TC, \gamma_b^*, \gamma_v^*)$		
	Cost set 1	Cost set 2	Cost set 3
0.5	(34.5370,2.2023,2.9539)	(69.1821,1.7515,2.8696)	(83.0791,1.9088,2.6715)
0.7	(39.6393,2.2596,2.9016)	(74.7860,1.8316,2.8021)	(90.6220,1.9915,2.5730)
0.9	(44.2916,2.3118,2.8267)	(81.0004,1.9101,2.7056)	(98.8850,2.0724,2.4579)
11	(35.4811,2.2053,2.9559)	(69.1223,1.7581,2.8735)	(83.6642,1.9608,2.6325)
13	(35.2332,2.2102,2.9547)	(69.0378,1.7707,2.8731)	(83.5694,1.9684,2.6315)
15	(35.0451,2.2100,2.9557)	(68.9829,1.7766,2.8765)	(83.5100,1.9811,2.6322)
5	(34.6070,2.1783,2.9901)	(69.1822,1.7526,2.8718)	(83.7319,1.9520,2.6283)
6	(35.1703,2.1943,2.9642)	(69.5452,1.7451,2.8426)	(82.6787,1.9382,2.6055)
7	(35.6372,2.2085,2.9497)	(69.8410,1.7391,2.8208)	(82.0822,1.9252,2.5762)

7.6 Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer models the leadership hierarchy and hunting strategy of grey wolves in nature introduced by Mirjalili et al. (2014). By simulating the roles of alpha, beta, delta, and omega wolves, the algorithm effectively navigates between global exploration and local exploitation during optimization. A Markovian retrieval queue with feedback and failures, optimized via GWO was studied by Dhibar and Jain (2025).

In the case of GWO, the control parameter decreases linearly from 2 to 0 over the course of 150 iterations to balance exploration and exploitation. The population size is set to 75, and the decision variables are restricted to the interval (Aggarwal et al. 2025; Daraseliya et al. 2025). System parameters such as $\omega = 2.5$, $\delta = 4$, $v_3 = 9$, $\varpi_1 = 0.25$, $\varpi_2 = 0.5$, $\varpi_3 = 0.25$, $\xi = 3.3$, and $\mu = 13$ are fixed while optimizing the overall cost. The optimization is performed in a two-dimensional search space corresponding to (γ_b, γ_v) , with all wolves initialised uniformly within this domain. At each iteration, the fitness of every wolf

Algorithm 2 Pseudo Code for ABC Algorithm.

```

Initialize food sources (solutions) randomly.
Evaluate the fitness of all food sources.
while stopping criterion is not met do
  for each employed bee do
    Generate a neighbor solution.
    Apply greedy selection.
  end for
  Calculate selection probabilities.
  for each onlooker bee do
    Choose a food source based on probability.
    Generate a neighbor solution.
    Apply greedy selection.
  end for
  if any food source exceeds the abandonment limit then
    Replace with a new randomly generated solution.
  end if
  Memorize the best solution found so far.
end while
Return the best food source.

```

Table 17 Impact of κ , η , θ on $(TC, \gamma_b^*, \gamma_v^*)$ using GWO

Parameters	$(TC, \gamma_b^*, \gamma_v^*)$		
	Cost set 1	Cost set 2	Cost set 3
0.5	(35.6930,2.2777,2.9535)	(69.1356,1.8326,2.8807)	(82.9769,1.9120,2.6866)
0.7	(39.6395,2.2541,2.9007)	(74.9031,1.9059,2.9415)	(91.5815,2.0016,2.5770)
0.9	(43.9960,1.6231,1.4774)	(78.0182,1.9160,2.9061)	(55.7902,2.2717,2.3350)
11	(37.4061,2.2340,2.9242)	(72.4468,2.3886,2.8481)	(85.6639,1.9676,2.6011)
13	(37.1923,2.2350,2.9319)	(71.8163,1.8150,2.8388)	(85.5828,1.9753,2.6631)
15	(37.0313,2.2413,2.9288)	(71.8002,1.8208,2.8632)	(85.4316,1.9813,2.7427)
5	(37.2602,2.7724,3.3868)	(71.9313,1.8321,2.8071)	(87.0401,1.9991,2.7354)
6	(37.4491,2.2223,2.9480)	(73.7959,1.5083,2.8024)	(86.8946,1.9947,2.6006)
7	(37.5528,2.2601,2.9425)	(74.2202,1.5850,2.4067)	(85.2134,1.9195,2.4732)

is evaluated using the objective function in Eq. 49, and any wolf that steps outside the feasible range is returned to the nearest valid boundary. The hierarchy of the pack (alpha, beta, and delta wolves) is updated based on these fitness values, and the remaining wolves update their positions using the standard leader-guided mechanism. A convergence curve is generated by recording the best cost value at each iteration. With 75 wolves and 150 iterations, the search involves a total of $75 \times 150 = 11,250$ objective-function evaluations. The algorithm terminates after the maximum iteration count is reached and returns the minimum cost along with the optimal parameters (γ_b^*, γ_v^*) .

Algorithm 3 outlines the procedural steps of the GWO algorithm in pseudocode form, while Table 17 summarizes the impact of parameters (κ, η, θ) on the minimum total cost and optimal service rate (γ_b^*, γ_v^*) employing GWO.

Algorithm 3 Pseudo Code for GWO Algorithm.

```

Initialize grey wolf population within bounds.
Evaluate fitness of each wolf.
Identify alpha (best), beta (second), and delta (third) wolves.
for t = 1 to MaxIterations do
    Update control parameter.
    for each wolf Xi do
        for each leader (alpha, beta, delta) do
            Compute vectors A and C.
            Compute distance and candidate position.
        end for
        Update Xi as the average of all three positions.
    end for
    Evaluate fitness of updated wolves.
    Update alpha, beta, and delta accordingly.
end for
Return the alpha wolf as the optimal solution.

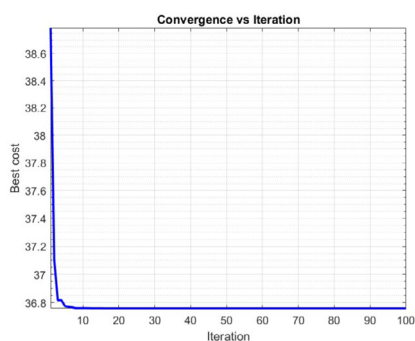
```

7.7 Comparative Analysis within WOA, ABC and GWO

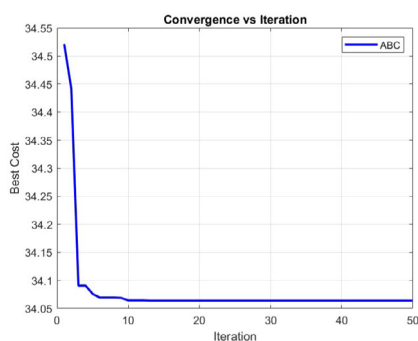
To evaluate the effectiveness of the optimization methods employed, a comparative analysis was conducted among the WOA, ABC and GWO. All algorithmic simulations were executed in MATLAB, which was used to generate the corresponding cost values and comparative results presented from Tables 15 to 17. The performance of each algorithm was assessed based on the final minimized cost value and consistency across multiple simulation runs. Results indicated that ABC slightly outperformed WOA and GWO in terms of achieving a lower total system cost, suggesting its stronger exploitation capabilities in fine-tuning service rate parameters. Meanwhile, WOA demonstrated competitive performance with fewer fluctuations, and GWO showed reasonable results though with slightly higher variation. Overall, all three algorithms effectively minimized the objective function, but ABC emerged as the most efficient for the given problem setting.

7.8 Convergence Behavior of WOA, ABC, and GWO

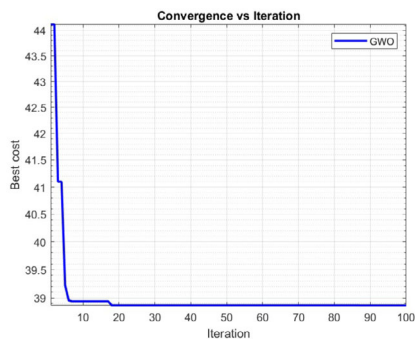
The convergence patterns of the three metaheuristic algorithms were also examined to understand their search dynamics. ABC displayed a steady and rapid convergence toward the



(a) Convergence in WOA



(b) Convergence in ABC



(c) Convergence in GWO

Fig. 12 Convergence of the cost function

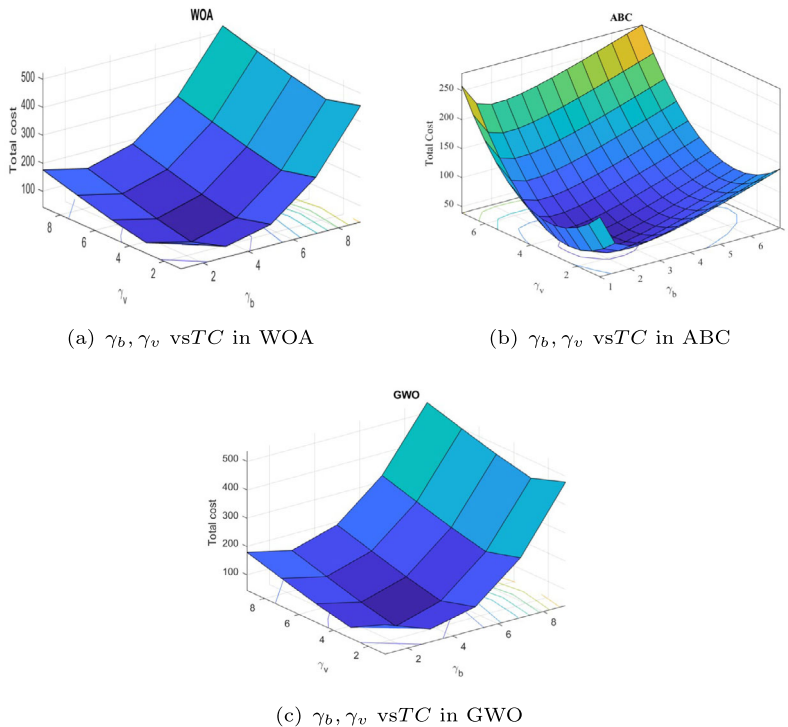


Fig. 13 Optimality of the cost function

optimal cost value, indicating its efficiency in balancing exploration and exploitation. WOA converged slightly more slowly but consistently, maintaining stability without premature convergence. In contrast, GWO showed faster initial progress but exhibited some stagnation in later iterations, hinting at limited exploitation in fine-tuning the solution. These observations affirm that while all algorithms are capable of navigating the cost landscape effectively, ABC provides the most robust convergence behavior for the formulated optimization problem. The convergence behavior of the three algorithms is illustrated in Fig. 12, while the optimality and variation of the total cost function across optimization runs are depicted in Fig. 13.

8 Conclusion and Future Work

In this study, a novel energy-efficient modeling framework for a tHAP incorporating a retrieval queueing system enhanced with multi-level sleep modes and dynamic system behaviors was presented. By integrating three distinct sleep states SM1 (light), SM2 (moderate), and SM3 (deep) the model captures a traffic-aware sleep strategy that intelligently balances energy consumption with service responsiveness. The stochastic analysis, supported by steady-state probability solutions and performance measures, confirms the effectiveness of deeper sleep modes in significantly reducing power usage, especially during periods of low traffic.

Furthermore, a cost optimization framework was formulated and addressed using three distinct metaheuristic algorithms, effectively identifying near-optimal parameter. Comparative results validate the robustness and performance of the proposed solution under realistic

tHAP operating conditions. Overall, this study offers a comprehensive and scalable solution for next-generation tHAP design, promoting intelligent energy management without compromising user experience. The integration of queuing dynamics, predictive control, and optimization provides a strong foundation for future green 5G and beyond network architectures.

Future work may focus on integrating more advanced queueing scenarios, such as priority-based or heterogeneous service classes, to better reflect diverse traffic patterns in 5G networks. The model can also be extended to include multi-server systems or interconnected tHAPs, capturing the effects of load balancing and handovers. Incorporating adaptive sleep mode transitions driven by real-time traffic prediction using machine learning can further optimize energy efficiency. Finally, leveraging hybrid metaheuristics or reinforcement learning could enable smarter, real-time cost and energy management in dynamic network environments.

Appendix A

$$\begin{aligned}
 Nr_2''(1) = & 2 \left\{ -\kappa E(X) [R(R-1) - S^*(\kappa + \eta) [\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] - \eta \right. \\
 & E(X)E(P)\kappa + \sigma [\kappa E(X)[1 - \kappa E(Q) + \eta E(P)] + \kappa [\zeta' Q^*(\theta) - \theta E(Q) - R]] \\
 & + [\kappa E(X)S^*(\kappa + \eta)[1 - \eta E(P)]] [\varpi_1 \kappa E(X)Q^*(\theta)[E(L_1) + \varpi_2[E(L_2) + \varpi_3 E(L_3)]] \\
 & \left. + 1 - Q^*(\theta)] - S^*(\kappa + \eta) \eta [\kappa E(X)E(P)[R - [\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)]] \right\}. \\
 Nr_2'''(1) = & 3 \left\{ -\kappa E(X_2) \left[R(1 - \kappa) - S^*(\kappa + \eta) [1 + \zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] \right. \right. \\
 & \left. - \eta E(X)E(P) + \sigma [\kappa E(X)[1 - \kappa E(Q) + \eta E(P)] + \kappa [\zeta' Q^*(\theta) - \theta E(Q) - R]] \right] \\
 & - \kappa E(X) \left[R(R-1)[1 - \kappa - \eta\sigma] - S^*(\kappa + \eta) [1 - 2[\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta \right. \\
 & E(Q)] - [\zeta'' Q^*(\theta) - 2\kappa E(X)E(Q)\zeta' - \kappa [E(X_2)E(Q) + E(X)E(Q_2)] - \theta E(Q_2)] \\
 & - \kappa E(X_2)\sigma + 3\eta\sigma [2\kappa E(X)E(P)R + \kappa [E(X_2)E(P) + E(X)E(P_2)]] + \sigma [\kappa E(X_2) \\
 & + 2\kappa E(X)[\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] + \kappa [\zeta'' Q^*(\theta) - 2\kappa E(X)E(Q)\zeta' - \kappa \\
 & [E(X_2)E(Q) + E(X)E(Q_2)] - \theta E(Q_2)] + \eta \kappa [E(X_2)E(P) - E(P_2)E(X)] - \kappa \\
 & \left. R(R-1) \right] + [\kappa S^*(\kappa + \eta) [E(X_2)[1 - \eta E(P)] - \eta E(X)E(P_2)] [\varpi_1 Q^*(\theta) \kappa E(X) \\
 & [E(L_1) + \varpi_2[E(L_2) + \varpi_3 E(L_3)]] + 1 - Q^*(\theta)] + [\kappa E(X)S^*(\kappa + \eta)(1 - \eta E(P))] \\
 & \left[\varpi_1 Q^*(\theta) \kappa [E(X)[E(L_{12}) + \varpi_2[E(L_{22}) + \varpi_3 E(L_{32})]] E(X_2)E(L_1) + E(X_2)[E(L_1) \right. \\
 & + \varpi_2[E(L_2) + \varpi_3 E(L_3)]] - 2\varpi_1(\kappa E(X))^2 [E(Q)[E(L_1) + \varpi_2[E(L_2) + \varpi_3 E(L_3)]] \\
 & + \varpi_2 Q^*(\theta) [E(L_2)E(L_1) + \varpi_3 [E(L_1)E(L_3) + E(L_2)E(L_3)]] + 2(1 - Q^*(\theta)) \\
 & \left. \theta E(R) \right] - S^*(\kappa + \eta) \eta [\kappa E(X_2)E(P) + E(X)E(P_2))(R - [\zeta' Q^*(\theta) - \kappa E(X)E(Q) \\
 & - \theta E(Q)] + \kappa E(X)E(P)(R(R-1) - [\zeta'' Q^*(\theta) - 2\kappa E(X)E(Q)\zeta' - \kappa [E(X_2)E(Q) \\
 & + E(X)E(Q_2)] - \theta E(Q_2)]) \left. \right\}.
 \end{aligned}$$

$$\begin{aligned}
Dr_1''(1) &= -2\kappa E(X) \left[R - [S^*(\kappa + \eta) + \kappa E(X)\sigma] - [S^*(\kappa + \eta) + \kappa\sigma][\zeta' Q^*(\theta) - \kappa E(X) \right. \\
&\quad \left. E(Q) - \theta E(Q)] - \eta\sigma[\kappa E(X)E(P) + R] \right]. \\
Dr_1'''(1) &= -3\kappa E(X_2) \left[R - [S^*(\kappa + \eta) + \kappa E(X)\sigma] - [S^*(\kappa + \eta) + \kappa\sigma][\zeta' Q^*(\theta) - \kappa E(X) \right. \\
&\quad \left. E(Q) - \theta E(Q)] - \eta\sigma[R - \kappa E(X)E(P)] \right] - 3\kappa E(X) \left[R(R-1) - [S^*(\kappa + \eta) + \kappa \right. \\
&\quad \left. E(X_2)\sigma] - 2[S^*(\kappa + \eta) + \kappa E(X)\sigma][\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] - [S^*(\kappa + \eta) \right. \\
&\quad \left. + \kappa\sigma][\zeta'' Q^*(\theta) - 2\kappa E(X)E(Q)\zeta' - \kappa[E(X_2)E(Q) + E(X)E(Q_2)] - \theta E(Q_2)] - \eta\sigma \right. \\
&\quad \left. [R(R-1) - \kappa[E(X_2)E(P) + E(X)E(P_2) + 2E(X)E(P)R]] \right]. \\
Nr_1'''(1) &= Nr_2'''(1) - 6 \left[S^*(\kappa + \eta)\eta\kappa E(X)E(P)[\zeta' Q^*(\theta) - \kappa E(X)E(Q) - \theta E(Q)] - R \right. \\
&\quad \left. + 2(1 - Q^*(\theta))[1 - \theta E(R)][\kappa E(X)S^*(\kappa + \eta)[1 - \eta E(P)]] \right]. \\
\zeta'' &= -\varpi_1\kappa \left[E(X)[E(L_{12}) + \varpi_2[E(L_{22}) + \varpi_3 E(L_{32})]] + E(X_2)[E(L_1) + \varpi_2[E(L_2) \right. \\
&\quad \left. + \varpi_3 E(L_3)]] \right] + 2\varpi_1\varpi_2(\kappa E(X))^2[E(L_1)E(L_2) + \varpi_3[E(L_2)E(L_3)]].
\end{aligned}$$

Acknowledgements The authors acknowledge all the competing authorities in preparing the manuscript.

Funding The report study has been prepared with the support of DST-RSF research project, with scientific project No. 64800 (DST)

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate Not applicable.

Conflicts of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Abbasi O, Yadav A, Yanikomeroglu H, Dao ND, Senarath G, Zhu P (2024) HAPS for 6G networks: potential use cases, open challenges, and possible solutions. *IEEE Wirel Commun* 31(3):324–331. <https://doi.org/10.1109/MWC.012.2200365>
- Aggarwal A, Selvamuthu D (2024) Energy analysis using semi-Markov modeling for the base station in 5G networks. *Int J Commun Syst* 37(4):e5678. <https://doi.org/10.1002/dac.5678>
- Aggarwal A, Kalita P, Selvamuthu D (2025) Stochastic modeling of a base station in 5G wireless networks for energy aspects using advanced sleep mechanism. *Methodol Comput Appl Probab* 27(3):58. <https://doi.org/10.1007/s11009-025-10187-1>
- Amponis G, Lagkas T, Bouzinis P, Radoglou-Grammatikis P, Sarigiannidis A, Sarigiannidis P, Argyriou V (2025) Introducing Energy Efficient Routing in UAV-Satellite NTN for Dynamic 6G Interconnectivity. *IEEE Transactions on Communications*. <https://doi.org/10.1109/TCOMM.2025.3634253>

- Ayyappan G, Nithya S (2025) Analysis of non-Markovian batch arrival retrial queue with priority services, vacation and vacation interruption under random environment, negative arrival, immediate feedback, differentiate breakdown, and repair. *Int J Math Oper Res* 31(2):143–163. <https://doi.org/10.1504/IJMOR.2025.147024>
- Bian X, Mao Y, Zhang J (2023) Grant-free massive random access with retransmission: Receiver optimization and performance analysis. *IEEE Trans Commun* 72(2):786–800. <https://doi.org/10.1109/TCOMM.2023.3328553>
- Chaudhry ML, Templeton JG (1983) A first course in bulk queues. Wiley, New York
- Daraseliya A, Sopin E, Begishev V, Koucheryavy Y, Samouylov K (2025) Resource allocation in 5G cellular IoT systems with early transmissions at the random access phase. *Sensors* 25(7):2264. <https://doi.org/10.3390/s25072264>
- Deepa V, Haridass M (2023) Simulation analysis of a base station using finite buffer M/G/1 queueing system with variant sleeps. *Methodol Comput Appl Probab* 25(4):80. <https://doi.org/10.1007/s11009-023-10052-z>
- Deepa V, Haridass M, Selvamuthu D, Kalita P (2023) Analysis of energy efficiency of small cell base station in 4G/5G networks. *Telecommun Syst* 82(3):381–401. <https://doi.org/10.1007/s11235-022-00987-y>
- Dhibar S, Jain M (2025) Metaheuristics and strategic behavior of Markovian retrial queue under breakdown, vacation and Bernoulli feedback. *Appl Intell* 55(4):273. <https://doi.org/10.1007/s10489-024-05978-x>
- Farooqi MZ, Laimio J, Manner J (2025) 5G in the air: energy efficient future 5G high altitude platform stations. In: 2025 IEEE conference on technologies for sustainability. IEEE, pp 1–8. <https://doi.org/10.1109/SusTech63138.2025.11025605>
- Firouzi R, Rahmani R (2024) Delay-sensitive resource allocation for IoT systems in 5G O-RAN networks. *Internet of Things* 26:101131. <https://doi.org/10.1016/j.iot.2024.101131>
- Gross D, Shortle JF, Thompson, J.M., & Harris C.M. (2019) Fundamentals of Queueing Theory, 4th edn. Wiley Series in Probability and Statistics. Wiley, India
- Harini R, Indhira K (2024) Dynamical modelling and cost optimization of a 5G base station for energy conservation using feedback retrial queue with sleeping strategy. *Telecommun Syst* 86(4):661–690. <https://doi.org/10.1007/s11235-024-01155-0>
- Jain I, Gautam A (2025) Reliability analysis of energy management subsystem in a high altitude platform station using hierarchical stochastic models and machine learning. *J Ind Manag Optim* 21(2):1269–1299. <https://doi.org/10.3934/jimo.2024124>
- Jain V, Vishnevsky VM, Selvamuthu D, Raj R (2022) Analysis of power management in a tethered high altitude platform using $MAP/PH^{[3]}/1$ retrial queueing model. In: International conference on distributed computer and communication networks. Cham: Springer Nature Switzerland, pp 218–230. https://doi.org/10.1007/978-3-031-23207-7_17
- Kalita P, Selvamuthu D (2023) Stochastic modelling of sleeping strategy in 5G base station for energy efficiency. *Telecommun Syst* 83(2):115–133. <https://doi.org/10.1007/s11235-023-01001-9>
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Optim* 39(3):459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Kashifalghitaa HHMH, Al-Obaidi AHM (2023) Some arrival cases in quorum queues under N-policy. *Utilitas Mathematica* 120:437–454
- Kumar BK, Madheswari SP (2003) $M^{[X]}/G/1$ retrial queue with multiple vacations and starting failures. *Opsearch* 40(2):115–137. <https://doi.org/10.1007/BF03398688>
- Lahdekorpi P, Hronec M, Jolma P, Moilanen J (2017) Energy efficiency of 5G mobile networks with base station sleep modes. In: 2017 IEEE conference on standards for communications and networking (CSCN). IEEE, pp 163–168. <https://doi.org/10.1109/CSCN.2017.8088616>
- Liu C, Natarajan B, Xia H (2015) Small cell base station sleep strategies for energy efficiency. *IEEE Trans Veh Technol* 65(3):1652–1661. <https://doi.org/10.1109/TVT.2015.2413382>
- Liu X, Xu X, Liu M (2024) Strategy analysis of retrial queue with parallel customer and standby server. *Methodol Comput Appl Probab* 26(3):27. <https://doi.org/10.1007/s11009-024-10098-7>
- Liu Q, Wang S, Qi Z, Si Z, Liu Q (2025) Energy-efficient joint computation offloading and resource allocation optimization in UAV/HAP-assisted AIoT networks. *IEEE Trans Green Commun Network* 9(4):1936–1950. <https://doi.org/10.1109/TGCN.2025.3554040>
- Mathavavisakan NM, Indhira K (2025) Neuro fuzzy system and metaheuristic cost optimization of a batch arrival retrial queue with variable server model under working vacation. *Palestine J Math* 14(1):606–627
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>

- Mittal N, Ivanova N, Jain V, Vishnevsky V (2024) Reliability and availability analysis of high-altitude platform stations through semi-Markov modeling. *Reliab Eng Syst Safety* 252:110419. <https://doi.org/10.1016/j.res.2024.110419>
- Nadimi-Shahraki MH, Zamani H, Asghari Varzaneh Z, Mirjalili S (2023) A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. *Arch Comput Methods Eng* 30(7):4113–4159. <https://doi.org/10.1007/s11831-023-09928-7>
- Niu Z, Guo X, Zhou S, Kumar PR (2015) Characterizing energy-delay tradeoff in hyper-cellular networks with base station sleeping control. *IEEE J Sel Areas Commun* 33(4):641–650. <https://doi.org/10.1109/JSAC.2015.2393494>
- Priya S, Sudhesh R, Kalita P, Dharmaraja S (2025) Stochastic modeling and analysis of energy efficiency in tethered high altitude platforms based smart macro base station. <https://doi.org/10.21203/rs.3.rs-5598511/v1>
- Raj R (2024) Stochastic modeling of enhancing energy efficiency for HAP in HAP-LEO systems using Markov regenerative process. *Comput Netw* 242:110253. <https://doi.org/10.1016/j.comnet.2024.110253>
- Raj R, Dharmaraja S (2023) Stochastic modelling of multi-layer HAP-LEO systems in 6G for energy saving: an analytical approach. *Comput Commun* 210:22–34. <https://doi.org/10.1016/j.comcom.2023.07.037>
- Rathinasamy P, Krishnasamy K, Karnakumar S, Ponnusamy B (2023) Navigating customer demand in a failure prone retrial queueing system with predictive maintenance. *Int J Oper Res* 20:12–24. https://doi.org/10.6886/IJOR.202308_20.0002
- Samanta SK, Rashmi KM (2025) Analytical and computational aspects of a batch arrival retrial queue with a constant retrial policy. *Quality Technology & Quantitative Management* 1-28. <https://doi.org/10.1080/16843703.2025.2455894>
- Sapna, Jain A, Sharma R (2024) Preventive maintenance and availability exploratory in a retrial queueing system with working vacation using ANFIS. In: International conference on mathematical modeling, computational intelligence techniques and renewable energy. Singapore: Springer Nature Singapore, pp. 629–643. https://doi.org/10.1007/978-981-96-1449-3_49
- Selvamuthu D (2025) Introduction to stochastic processes: queues, finance, and credit risk. Springer Nature. <https://doi.org/10.1007/978-981-97-6152-4>
- Song T, Lopez D, Meo M, Piovesan N, Renga D (2024) High altitude platform stations: the new network energy efficiency enabler in the 6G era. In: 2024 IEEE wireless communications and networking conference (WCNC) (pp. 1-6). IEEE. <https://doi.org/10.1109/WCNC57260.2024.10571153>
- Tadj L, Choudhury G, Tadj C (2006) A bulk quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule. *Stochastics Int J Prob Stoch Process* 78(1):1–11. <https://doi.org/10.1080/17442500500397574>
- Tadj L, Choudhury G, Reka K (2012) A two-phase quorum queueing system with Bernoulli vacation schedule, setup, and N-policy for an unreliable server with delaying repair. *Int J Serv Oper Manag* 12(2):139–164. <https://doi.org/10.1504/IJSOM.2012.047103>
- Templeton JGC, Falin GI (2023) Retrial queues. CRC Press
- Toth A, Sztrik J, Bardavelidze A (2023) Simulating retrial queues with finite source, two-way communication to the orbit, backup server, and impatient customers. In: International conference on information technologies and mathematical modelling. Cham: Springer Nature Switzerland, pp. 316–328. https://doi.org/10.1007/978-3-031-65385-8_23
- Vishnevsky VM, Kozyrev DV, Rykov VVE, Nguyen ZF (2020) Reliability modeling of an unmanned high-altitude module of a tethered telecommunication platform. *Intell Syst Technol* 4:26–38. <https://doi.org/10.14357/20718632200403>
- Xu J, Kishk MA, Alouini MS (2023) Space-air-ground-sea integrated networks: modeling and coverage analysis. *IEEE Trans Wireless Commun* 22(9):6298–6313. <https://doi.org/10.1109/TWC.2023.3241341>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.