RCHUB: JOURNAL OF COMPUTATIONAL INTELLIGENCE SCIENCE AND ENGINEERING (JCISE)
VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

INTELLIGENT NETWORK SLICE ORCHESTRATION FOR EDGE–CLOUD ARCHITECTURES USING HYBRID SWARM OPTIMIZATION

Yuvaraj Gandhi S[1], Revathi T[2],
[1]Assistant Professor, Department of Computer Applications, PSG College of Arts & Science, Coimbatore, India. yuvarajgandhi@psgcas.ac.in [0000-0003-0508-5886]
Associate Professor & Dean, Computational Sciences, PSG College of Arts & Science, Coimbatore, India. revathi@psgcas.ac.in [0009-0004-0390-1119]

Abstract

The rapid growth of distributed applications, Internet of Things (IoT) devices, and latency-sensitive services has significantly increased the demand for intelligent resource management in edge–cloud computing environments. Traditional static orchestration mechanisms are inadequate for handling dynamic network conditions, heterogeneous workloads, and quality-of-service (QoS) requirements. To address these challenges, this study proposes an intelligent network slice orchestration framework for edge–cloud architectures using hybrid swarm optimization techniques. The proposed model integrates swarm-based optimization approaches to dynamically allocate computational and networking resources across distributed edge and cloud infrastructures. The framework aims to improve slice utilization efficiency, reduce latency, optimize bandwidth consumption, and enhance overall service performance. Simulation-based evaluation demonstrates that the proposed approach achieves improved orchestration efficiency and better QoS metrics compared with conventional resource allocation methods. The study highlights the potential of hybrid swarm intelligence in enabling adaptive and scalable orchestration for next-generation distributed computing systems.

Keywords – Edge–Cloud Computing, Network Slicing, Swarm Intelligence, Resource Orchestration, QoS Optimization, Distributed Computing.

1. Introduction

The rapid advancement of digital communication technologies and distributed computing infrastructures has significantly transformed modern network environments. The increasing deployment of Internet of Things (IoT) devices, smart applications, autonomous systems, and real-time multimedia services has generated enormous demand for scalable, low-latency, and intelligent computing frameworks. Traditional centralized cloud computing models often experience limitations in handling latency-sensitive applications due to network congestion, transmission delays, and centralized resource dependencies (Shi et al., 2016). Consequently, edge–cloud computing architectures have emerged as an effective paradigm for enabling distributed processing and efficient service delivery near end users.

Edge computing extends computational and storage capabilities closer to network edges, thereby reducing communication latency and improving response time for real-time applications such as smart healthcare, intelligent transportation, industrial automation, and augmented reality services (Satyanarayanan, 2017). However, the integration of edge and cloud infrastructures introduces several challenges related to resource allocation, traffic management, workload balancing, and dynamic orchestration. The heterogeneous nature of edge devices, varying workload intensities, and fluctuating network conditions make efficient resource management a critical research issue.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

To address these challenges, Network Slicing has gained significant attention in next-generation communication networks. Network slicing enables the creation of multiple logical virtual networks over a shared physical infrastructure, where each slice can be customized according to specific service requirements and quality-of-service (QoS) demands (Foukas et al., 2017). This capability is particularly important in edge–cloud environments where diverse applications require differentiated latency, bandwidth, reliability, and computational resources.

The orchestration of network slices in distributed edge–cloud architectures is a highly complex optimization problem. Conventional orchestration approaches generally rely on static resource allocation strategies and deterministic scheduling mechanisms, which are often insufficient under dynamic traffic conditions and large-scale distributed systems (Li et al., 2018). These limitations may result in inefficient bandwidth utilization, increased latency, resource fragmentation, and degraded service quality.

Recent advancements in intelligent optimization techniques have introduced new opportunities for adaptive orchestration in distributed networks. In particular, Swarm Intelligence algorithms have demonstrated considerable effectiveness in solving complex optimization problems due to their decentralized decision-making capability, adaptive learning behavior, and efficient convergence characteristics (Kennedy & Eberhart, 1995). Swarm-based optimization models imitate collective biological behaviors observed in natural systems such as bird flocking, ant colonies, and fish schooling. These algorithms have been widely applied in routing optimization, load balancing, traffic engineering, and resource scheduling problems.

Hybrid swarm optimization techniques further enhance orchestration performance by combining the strengths of multiple optimization strategies to improve exploration and exploitation capabilities during resource allocation. Such hybrid models can effectively optimize network slice management by dynamically adapting to changing network conditions and heterogeneous workloads. The integration of intelligent orchestration mechanisms with edge–cloud architectures can significantly improve resource utilization efficiency, QoS maintenance, service scalability, and network reliability.

Several recent studies have explored intelligent orchestration models for cloud and edge computing environments. However, many existing approaches primarily focus on either computational resource allocation or networking optimization independently, without considering integrated orchestration across distributed edge–cloud infrastructures. Furthermore, existing models often experience challenges related to convergence delay, scalability limitations, and inefficient adaptation under dynamic network environments.

Motivated by these research gaps, this study proposes an intelligent network slice orchestration framework for edge–cloud architectures using hybrid swarm optimization techniques. The proposed model aims to dynamically allocate network and computational resources based on traffic conditions, workload characteristics, and QoS requirements. The framework focuses on minimizing service latency, improving bandwidth utilization, enhancing resource efficiency, and achieving adaptive orchestration in distributed environments.

The major contributions of this study are summarized as follows:

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

1. To develop an intelligent orchestration framework for dynamic network slice management in edge–cloud architectures.
2. To integrate hybrid swarm optimization techniques for adaptive resource allocation and workload distribution.
3. To improve QoS performance through optimized latency reduction and bandwidth utilization.
4. To evaluate the proposed model under varying traffic and workload conditions using simulation-based experiments.

The remainder of this paper is organized as follows. Section 2 presents the literature review related to network slicing, edge–cloud orchestration, and swarm intelligence techniques. Section 3 discusses the proposed methodology and system architecture. Section 4 explains the experimental setup and performance evaluation metrics. Section 5 presents the results and discussion. Finally, Section 6 concludes the study and outlines future research directions.

2. Literature Review

The rapid growth of distributed applications and intelligent communication systems has significantly increased research interest in edge–cloud computing and network orchestration technologies. Modern distributed infrastructures require efficient mechanisms for resource allocation, traffic engineering, service orchestration, and quality-of-service (QoS) management. In this context, researchers have explored several optimization techniques and intelligent networking models to improve the performance of edge–cloud environments.

2.1 Edge–Cloud Computing Architectures

Edge Computing has emerged as an extension of conventional cloud computing to support real-time processing near end users. Unlike centralized cloud infrastructures, edge computing reduces latency and bandwidth consumption by processing data closer to data generation points. According to Shi et al. (2016), edge computing improves service responsiveness and enables efficient handling of latency-sensitive applications such as autonomous systems, smart healthcare, and industrial automation.

Several researchers have investigated integrated edge–cloud architectures for distributed workload management. Satyanarayanan (2017) emphasized that edge computing complements cloud infrastructures by providing localized computational support while maintaining cloud scalability. Similarly, Mao et al. (2017) highlighted that edge–cloud collaboration enhances computational efficiency and reduces communication overhead in mobile and IoT environments.

Despite these advantages, edge–cloud systems face several challenges, including dynamic workload variations, resource heterogeneity, network congestion, service orchestration complexity, and energy consumption. Efficient orchestration mechanisms are therefore essential for adaptive resource management in distributed environments.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

2.2 Network Slicing and Orchestration

Network Slicing has become a key enabling technology in next-generation communication networks. Network slicing allows multiple virtual networks to coexist on shared physical infrastructure while supporting differentiated QoS requirements for diverse applications.

Foukas et al. (2017) explained that network slicing improves flexibility and scalability in software-defined infrastructures by enabling customized service deployment. Similarly, Zhang et al. (2019) reported that intelligent slice orchestration can significantly improve bandwidth allocation and service reliability in distributed cloud environments.

Recent advancements in Software-Defined Networking and network function virtualization have further enhanced orchestration capabilities by enabling centralized traffic management and dynamic resource allocation (Kreutz et al., 2015). However, efficient orchestration of network slices remains a complex optimization problem due to continuously changing network conditions and heterogeneous traffic patterns.

Existing slice orchestration methods often rely on static scheduling approaches, heuristic resource allocation, and centralized optimization mechanisms. These approaches may experience scalability limitations and reduced adaptability under highly dynamic workloads.

2.3 Swarm Intelligence in Network Optimization

Swarm Intelligence has been widely applied to solve complex optimization problems in communication networks and distributed computing systems. Swarm-based algorithms imitate collective biological behaviors observed in natural systems such as bird flocking, ant colonies, and fish schooling.

Kennedy and Eberhart (1995) introduced Particle Swarm Optimization (PSO), which became one of the most widely used swarm intelligence algorithms for optimization problems. Swarm optimization techniques are particularly effective for routing optimization, workload balancing, resource allocation, traffic engineering, and task scheduling.

Ant Colony Optimization (ACO) has been extensively utilized for intelligent routing and shortest-path optimization in communication networks (Dorigo & Gambardella, 1997). Similarly, Grey Wolf Optimization (GWO) and Whale Optimization Algorithms (WOA) have demonstrated efficient convergence behavior in cloud resource allocation problems.

Recent studies have explored hybrid swarm optimization techniques to improve optimization performance. Hybrid models combine multiple swarm mechanisms to achieve better exploration and exploitation capabilities during optimization processes. According to Mirjalili et al. (2014), hybrid optimization strategies improve convergence stability and avoid premature local optima trapping.

In edge–cloud orchestration environments, swarm intelligence techniques can dynamically optimize network slice allocation, computational resource distribution, traffic flow management, and workload migration. However, several existing swarm-based orchestration models suffer from high computational overhead, delayed convergence, and inefficient scalability under large-scale distributed systems.

VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

2.4 Research Gap Identification

The literature indicates that substantial progress has been achieved in edge–cloud orchestration and intelligent optimization techniques. Nevertheless, several limitations remain unresolved.

Most existing studies focus independently on either networking optimization, computational resource management, or traffic engineering. Limited attention has been given to integrated orchestration frameworks capable of jointly optimizing network slices and computational resources in distributed edge–cloud architectures.

Furthermore, conventional orchestration methods often fail to adapt dynamically to changing workloads, maintain QoS consistency, efficiently utilize distributed resources, and support scalable orchestration under heterogeneous traffic environments. Although swarm intelligence techniques provide promising optimization capabilities, existing approaches require further enhancement to achieve adaptive and efficient orchestration in next-generation distributed networks.

To address these limitations, the present study proposes a hybrid swarm optimization-based intelligent network slice orchestration framework for edge–cloud architectures. The proposed model aims to improve adaptive resource allocation, reduce service latency, optimize bandwidth utilization, and enhance orchestration efficiency in distributed environments.

3. Proposed Methodology

This study proposes an intelligent network slice orchestration framework for distributed edge–cloud architectures using hybrid swarm optimization techniques. The proposed methodology focuses on adaptive resource allocation, intelligent traffic management, and efficient network slice orchestration under dynamic workload conditions. The framework aims to improve quality-of-service (QoS) performance by minimizing latency, optimizing bandwidth utilization, and enhancing resource efficiency across distributed edge and cloud infrastructures.

The proposed model integrates Network slicing, Edge computing, Hybrid swarm intelligence, and Intelligent orchestration mechanisms to dynamically manage heterogeneous workloads in modern communication networks.

3.1 Mathematical Formulation

The proposed orchestration problem is formulated as a multi-objective optimization model.

Objective Function

The overall fitness function is represented as:

$$F(x) = \alpha L + \beta B + \gamma R + \delta E$$

Where:

- L= Average network latency
- B= Bandwidth utilization
- R= Resource utilization efficiency
- E= Energy consumption
- $\alpha, \beta, \gamma, \delta$= Weight coefficients

VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

The optimization objective is to:

$$\min(F(x))$$

subject to:

$$0 \leq U_i \leq U_{max}$$
$$D_i \leq D_{threshold}$$
$$BW_i \geq BW_{required}$$

Where:

- U_i= Resource utilization of node i
- D_i= Delay of slice i
- BW_i= Available bandwidth

3.2 Latency Model

The latency of a network slice is calculated as:

$$L = T_t + T_q + T_p$$

Where:

- T_t= Transmission delay
- T_q= Queue delay
- T_p= Processing delay

Lower latency values indicate better orchestration performance.

3.3 Resource Utilization Model

Resource utilization is evaluated using:

$$R = \frac{\sum_{i=1}^{n} Resourc\,e_{used}}{\sum_{i=1}^{n} Resourc\,e_{total}}$$

The objective is to maximize resource utilization while preventing overload conditions.

3.4 Hybrid Swarm Optimization Algorithm

The proposed model combines multiple swarm optimization strategies to achieve efficient exploration and exploitation during orchestration.

The hybrid optimization mechanism performs:

- intelligent search,
- dynamic slice allocation,
- adaptive workload balancing,
- Traffic-aware routing optimization.

The algorithm continuously updates network slice positions according to QoS conditions and workload characteristics.

Algorithm: Intelligent Network Slice Orchestration Using Hybrid Swarm Optimization

Input:

Edge nodes E

Cloud resources C

Network slice requests S

QoS requirements Q

Output:

Optimized network slice allocation

Begin

1. Initialize edge nodes and cloud resources
2. Generate initial population of network slices
3. Evaluate QoS requirements for each slice
4. Calculate fitness function F(x)
5. While termination condition is not satisfied do
    a. Monitor network traffic conditions
    b. Measure latency and bandwidth utilization
    c. Update swarm positions
    d. Perform hybrid optimization search
    e. Select optimal resource allocation
    f. Allocate slices dynamically
    g. Update fitness values
6. End While
7. Return optimized orchestration results

End

VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

The proposed orchestration framework initially receives heterogeneous service requests generated from distributed edge devices and applications. The traffic monitoring module continuously analyzes workload conditions, network congestion, and QoS requirements.

The resource assessment module evaluates the availability of computational and networking resources across edge and cloud infrastructures. Based on these observations, the hybrid swarm optimization engine dynamically identifies optimal slice placement and workload allocation strategies.

The orchestration controller then allocates virtual network slices according to latency sensitivity, bandwidth demand, resource availability, and service priority. The framework continuously updates orchestration decisions under changing traffic conditions to maintain efficient QoS performance.

3.5 Advantages of the Proposed Framework

The proposed intelligent network slice orchestration framework offers several significant advantages in distributed edge–cloud environments. The integration of hybrid swarm optimization techniques enables adaptive and intelligent resource management under dynamic traffic conditions. The framework effectively reduces service latency by allocating delay-sensitive workloads to nearby edge resources, thereby improving response time for real-time applications. In addition, intelligent orchestration enhances bandwidth utilization through optimized traffic distribution and dynamic network slice allocation.

The proposed model also improves overall resource allocation efficiency by continuously monitoring computational and networking resources across edge and cloud infrastructures. The adaptive optimization mechanism supports scalable orchestration for heterogeneous environments with varying workload intensities. Furthermore, the framework maintains better quality-of-service (QoS) performance through dynamic traffic management and efficient workload balancing. The major advantages of the proposed framework are summarized below:

- Reduced service latency through localized edge processing
- Improved bandwidth utilization and traffic distribution
- Adaptive and intelligent resource management
- Enhanced scalability for large-scale distributed environments
- Efficient workload balancing across edge and cloud nodes.

VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

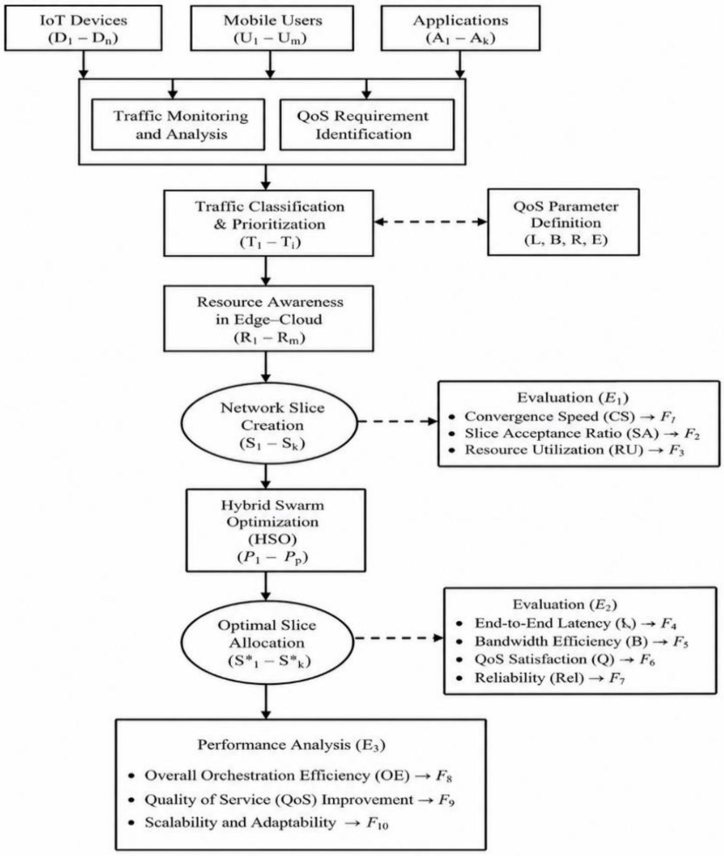

Figure 1. Flow diagram of the Proposed Methodology

Figure 1 illustrates the workflow of the proposed intelligent network slice orchestration framework in edge–cloud environments. The model performs traffic analysis, QoS-aware resource assessment, hybrid swarm optimization, and dynamic slice allocation to achieve efficient orchestration and improved network performance.

4. Experimental Setup and Performance Evaluation

This section presents the simulation environment, experimental parameters, performance metrics, and evaluation methodology used to analyze the effectiveness of the proposed intelligent network slice orchestration framework in edge–cloud architectures. The proposed hybrid swarm optimization model is evaluated under varying traffic and workload conditions to measure its orchestration efficiency and QoS performance. The proposed orchestration framework was implemented using a simulation-based environment consisting of edge and cloud infrastructures. The simulation was developed using CloudSim integrated with Python-based optimization modules to support intelligent orchestration and swarm optimization processes.

Table 1 presents the simulation parameters used for evaluating the proposed intelligent network slice orchestration framework. The configuration includes edge nodes, cloud servers, bandwidth range, workload characteristics, and optimization settings used to emulate realistic edge–cloud networking environments.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

Table 1. Simulation Parameters

Parameter	Value
Number of Edge Nodes	20
Number of Cloud Servers	5
Number of Network Slices	50
Bandwidth Range	10–100 Mbps
Latency Threshold	10 ms
Number of User Requests	1000
Optimization Iterations	100
Simulation Area	Distributed Edge–Cloud Environment
Workload Type	Dynamic and Heterogeneous
Optimization Technique	Hybrid Swarm Optimization

The experimental configuration was designed to emulate realistic edge–cloud networking conditions with heterogeneous service requests and dynamic traffic variations.

4.3 Performance Metrics

The effectiveness of the proposed orchestration framework was evaluated using several QoS and resource management metrics.

(a) End-to-End Latency

Latency represents the total delay experienced during workload transmission and processing.

$$L = T_t + T_q + T_p$$

Where:

- T_t= transmission delay,
- T_q= queue delay,
- T_p= processing delay.

Lower latency values indicate better orchestration performance.

(b) Resource Utilization

Resource utilization measures the efficiency of computational resource usage across edge and cloud nodes.

$$RU = \frac{R_{used}}{R_{total}} \times 100$$

Higher utilization values indicate better resource management efficiency.

(c) Bandwidth Utilization

Bandwidth utilization evaluates the efficiency of network bandwidth allocation during traffic transmission.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

$$BU = \frac{BW_{allocated}}{BW_{available}} \times 100$$

(d) Slice Acceptance Ratio

Slice acceptance ratio measures the percentage of successfully allocated network slices.

$$SAR = \frac{S_{accepted}}{S_{requested}} \times 100$$

Higher slice acceptance values indicate efficient orchestration capability.

(e) QoS Satisfaction Ratio

QoS satisfaction evaluates the ability of the orchestration framework to maintain application-specific service requirements.

$$QoS = \frac{Q_{satisfied}}{Q_{total}} \times 100$$

5. Results and Discussion

This section presents the performance evaluation and comparative analysis of the proposed intelligent network slice orchestration framework using hybrid swarm optimization in distributed edge–cloud architectures. The proposed model was evaluated under varying traffic and workload conditions to analyze its effectiveness in terms of latency reduction, resource utilization, bandwidth optimization, slice allocation efficiency, and QoS maintenance.

The performance of the proposed framework was compared with:

- Traditional Static Resource Allocation,
- Particle Swarm Optimization (PSO)-Based Orchestration,
- Conventional Load Balancing Mechanisms.

The experimental results demonstrate that the proposed hybrid swarm optimization framework achieves improved orchestration efficiency and adaptive resource management in distributed environments.

5.1 Latency Analysis

Latency is one of the most critical QoS parameters in edge–cloud environments, particularly for delay-sensitive applications such as smart healthcare, autonomous systems, and industrial automation. The proposed orchestration framework dynamically allocates workloads to nearby edge resources to minimize communication delay and improve response time. Table 2 compares the latency performance of different orchestration methods under varying workload conditions.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

Table 2. Comparison of Average Latency

Method	Average Latency (ms)
Static Resource Allocation	28.6
Conventional Load Balancing	22.4
PSO-Based Orchestration	18.7
Proposed Hybrid Swarm Optimization	12.3

The results indicate that the proposed framework achieved the lowest average latency among all compared methods. The hybrid swarm optimization mechanism effectively distributed workloads across edge and cloud infrastructures, thereby reducing transmission delay and congestion.

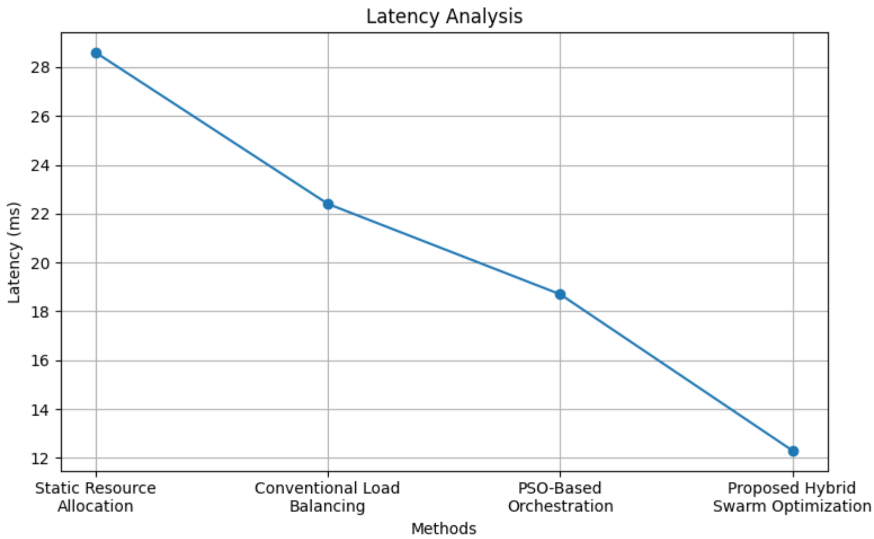

Figure 2. Latency Analysis

Figure 2 presents the comparative latency performance of different orchestration methods. The proposed hybrid swarm optimization framework achieved lower latency compared with conventional resource allocation and PSO-based orchestration techniques due to efficient workload distribution and adaptive edge resource utilization.

5.2 Resource Utilization Analysis

Efficient resource utilization is essential for maintaining scalability and reducing infrastructure overhead in distributed computing environments. The proposed model dynamically allocates resources according to workload intensity and QoS requirements. Table 3 shows the resource utilization efficiency achieved by the proposed and existing orchestration approaches.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

Table 3. Resource Utilization Comparison

Method	**Resource Utilization (%)**
Static Resource Allocation	68.2
Conventional Load Balancing	74.5
PSO-Based Orchestration	81.3
Proposed Hybrid Swarm Optimization	89.6

The proposed framework achieved higher resource utilization efficiency compared with conventional methods. The intelligent optimization mechanism efficiently balanced workloads across distributed nodes and minimized resource fragmentation.

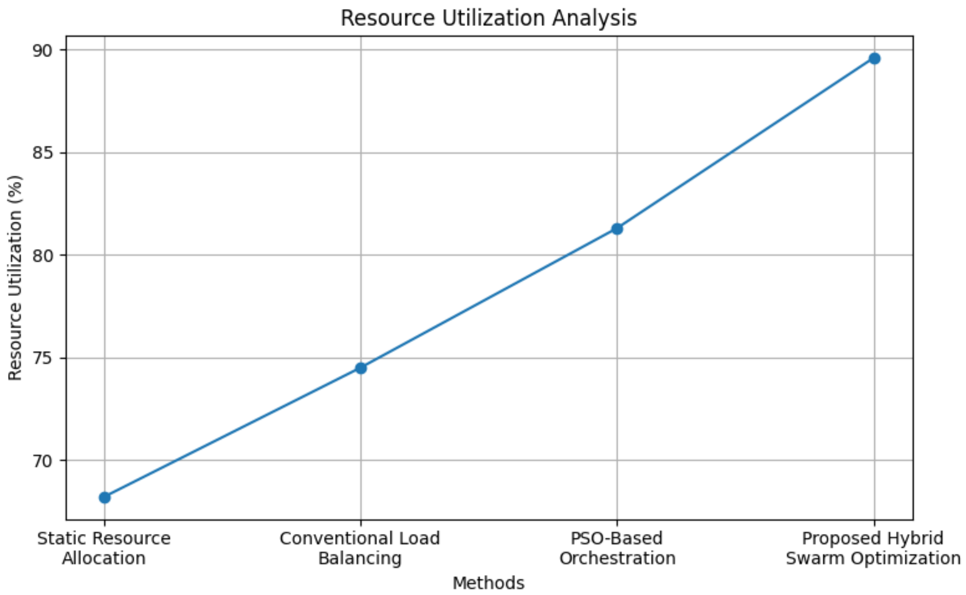

Figure 3. Resource Utilization Analysis

Figure 3 illustrates the resource utilization efficiency of the proposed orchestration framework. The hybrid swarm optimization mechanism improved computational resource allocation and workload balancing across distributed edge–cloud infrastructures, resulting in higher resource utilization performance.

5.3 Bandwidth Utilization Analysis

Bandwidth optimization plays a major role in maintaining stable communication performance in distributed networks. The proposed framework dynamically optimized traffic routing and network slice allocation to improve bandwidth efficiency. Table 4 presents the bandwidth utilization performance of different network orchestration techniques.

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

Table 4. Bandwidth Utilization Comparison

Method	Bandwidth Utilization (%)
Static Resource Allocation	64.8
Conventional Load Balancing	71.6
PSO-Based Orchestration	83.1
Proposed Hybrid Swarm Optimization	91.4

The proposed model demonstrated superior bandwidth utilization performance due to adaptive traffic-aware orchestration and intelligent workload distribution mechanisms. Efficient slice allocation minimized network congestion and improved communication efficiency.

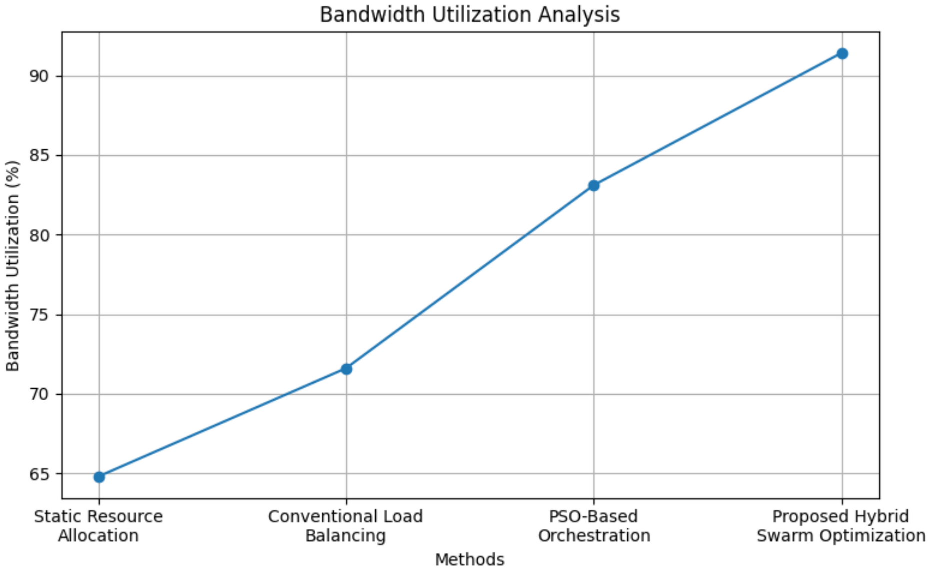

Figure 4. Bandwidth Utilization Analysis

Figure 4 shows the bandwidth utilization performance of different orchestration approaches. The proposed framework efficiently optimized traffic routing and network slice allocation, thereby improving bandwidth utilization and reducing communication congestion in distributed networking environments.

5.4 Slice Acceptance Ratio Analysis

The slice acceptance ratio reflects the orchestration framework's ability to allocate requested network slices under varying workload conditions successfully. Table 5 illustrates the network slice acceptance ratio achieved by various orchestration methods.

Table 5. Slice Acceptance Ratio Comparison

Method	Slice Acceptance Ratio (%)
Static Resource Allocation	70.5
Conventional Load Balancing	78.2
PSO-Based Orchestration	86.4
Proposed Hybrid Swarm Optimization	94.7

VOL: 2, ISSUE 2, APRIL-JUNE 2026
RCHUB PUBLISHER
ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

The proposed framework achieved the highest slice acceptance ratio among all compared approaches. The hybrid swarm optimization mechanism effectively identified optimal resource allocation strategies and improved orchestration scalability.

The results confirm that intelligent orchestration can efficiently manage dynamic traffic conditions and heterogeneous service requests.

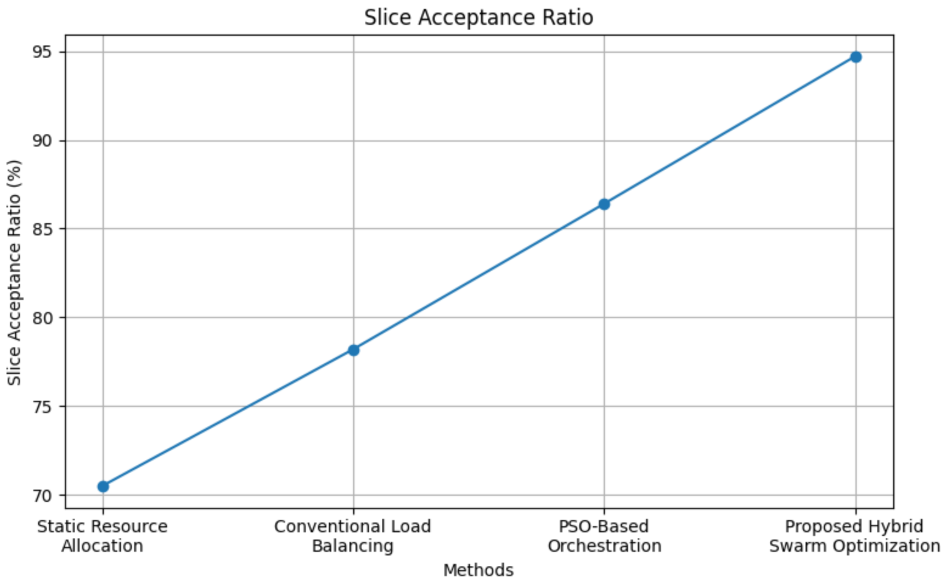

Figure 5. Slice Acceptance Ratio

Figure 5 presents the slice acceptance ratio achieved by various orchestration methods. The proposed intelligent orchestration framework demonstrated a higher acceptance ratio due to adaptive resource allocation and efficient network slice management under dynamic workload conditions.

5.5 QoS Satisfaction Analysis

QoS satisfaction is a critical performance indicator in edge–cloud networking environments. The proposed framework continuously monitored latency, bandwidth availability, and resource utilization to maintain stable service quality. Table 6 compares the QoS satisfaction levels maintained by different orchestration frameworks.

Table 6. QoS Satisfaction Comparison

Method	**QoS Satisfaction (%)**
Static Resource Allocation	72.1
Conventional Load Balancing	79.8
PSO-Based Orchestration	87.5
Proposed Hybrid Swarm Optimization	95.2

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

The proposed orchestration framework maintained better QoS performance due to adaptive resource management and dynamic slice allocation mechanisms. Intelligent optimization enabled efficient handling of fluctuating workloads and varying network conditions.

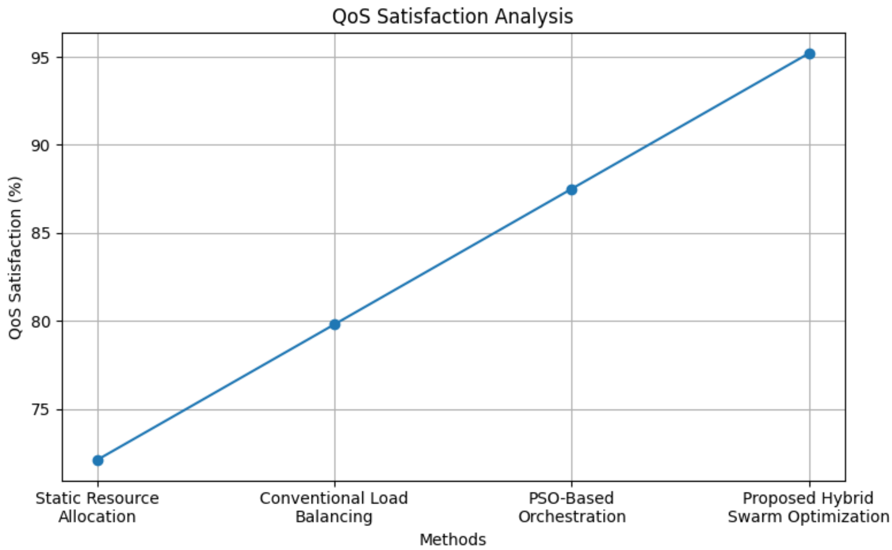

Figure 6. QoS Satisfaction Analysis

Figure 6 illustrates the QoS satisfaction performance of the proposed framework compared with existing methods. The hybrid swarm optimization approach maintained better QoS levels by dynamically optimizing latency, bandwidth allocation, and resource utilization in edge–cloud environments.

The experimental results clearly demonstrate that the proposed intelligent network slice orchestration framework outperforms conventional orchestration approaches across all evaluated performance metrics.

The major improvements achieved by the proposed framework include reduced latency, improved resource utilization, optimized bandwidth allocation, higher slice acceptance ratio, and enhanced QoS maintenance. The integration of hybrid swarm optimization techniques significantly improved adaptive orchestration capability in distributed edge–cloud architectures. Unlike traditional static allocation methods, the proposed model continuously adapts orchestration decisions according to changing traffic patterns and workload conditions.

Furthermore, the distributed orchestration mechanism effectively balanced workloads across edge and cloud infrastructures, thereby improving scalability and reducing network congestion. The findings indicate that intelligent hybrid swarm optimization can serve as an effective solution for dynamic network slice orchestration in next-generation distributed communication environments.

6. Conclusion and Future Work

This study proposed an intelligent network slice orchestration framework for edge–cloud architectures using hybrid swarm optimization techniques. The proposed model effectively addressed dynamic resource allocation and QoS management challenges in distributed computing environments. By integrating intelligent swarm-based optimization mechanisms,

ISSN: 3107-8168
DOI: https://doi.org/10.65725/JCISE/2/2/006

the framework dynamically optimized network slice allocation, workload distribution, and traffic management according to varying network conditions and service requirements. Experimental analysis demonstrated that the proposed framework achieved reduced latency, improved bandwidth utilization, enhanced resource allocation efficiency, and higher QoS satisfaction compared with conventional orchestration approaches. The adaptive orchestration mechanism efficiently balanced workloads across edge and cloud infrastructures while minimizing congestion and improving scalability. The findings indicate that hybrid swarm optimization can significantly enhance intelligent orchestration performance in next-generation distributed communication systems. Future work may focus on integrating deep reinforcement learning, energy-aware optimization, security-aware orchestration, and federated learning techniques to further improve adaptive resource management in large-scale edge–cloud networking environments.

References

[1] Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.

[2] Foukas, X., Patounas, G., Elmokashfi, A., & Marina, M. K. (2017). Network slicing in 5G: Survey and challenges. *IEEE Communications Magazine, 55*(5), 94–100.

[3] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1948.

[4] Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE, 103*(1), 14–76.

[5] Li, X., Qian, Z., Lu, S., & Wu, J. (2013). Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling, 58*(5–6), 1222–1235.

[6] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials, 19*(4), 2322–2358.

[7] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

[8] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer, 50*(1), 30–39.

[9] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal, 3*(5), 637–646.

[10] Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, H., & Leung, V. C. M. (2019). Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. *IEEE Communications Magazine, 55*(8), 138–145.