The Scientific Temper (2024) Vol. 15 (4): 3225-3233
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.32
https://scientifictemper.com/

RESEARCH ARTICLE

Optimizing IoT application deployment with fog - cloud paradigm: A resource-aware approach

K. Mohamed Arif Khan*, A.R. Mohamed Shanavas

Abstract
Fog computing is the architecture that most researchers use to build latency-sensitive Internet of Things (IoT) applications. By placing resource-constrained fog devices near the network's edge, fog computing design delivers less delay than the cloud computing paradigm. Fog nodes use the available resources to process the incoming data, which lowers the data amount that needs to be transferred to the server of the cloud. A system contains fog devices with various levels of computing power. The best system performance is only possible when the appropriate sensor nodes are connected to the parent fog node. In this study, we introduce a cluster head selection algorithm for effective network resource utilization through application deployment in a fog-cloud environment for internet of things-based applications. With the introduction of fog computing, the processing is animatedly dispersed through the cloud layers and fog, enabling the deployment of an application's modules closer to the foundation of fog-layer devices. The method is general and may be used with various network topologies and a broad range of standardized IoT applications, regardless of load.

Keywords: Internet of Things, Cloud computing, Fog computing, Fog-cloud paradigm, Cluster head selection algorithm, Network utilization, Energy consumption.

Introduction

IoT is a concept that uses wireless sensor networks (WSN) and the internet as its main technologies to create a virtual network that communicates with the physical world. IoT is also known as a worldwide network of «things» that have sensors, electronics, and software built into them. It connects the gadgets whose IP addresses can be used to uniquely identify them. IoT makes it possible for these online, linked gadgets to sense, gather, and interact with one another to enhance the quality of life. In order to create a pervasive environment and a ubiquitous experience, intelligent sensors and actuators will be deployed. IoT is expanding quickly to offer a new level of services that boost societal and economic development [Petrenko et al., 2018].

Thanks to recent advancements in applications that operate progressively, like brilliant urban communities, savvy clinical consideration, clever transportation, the shrewd lattice, and so forth, the internet of things (IoT), installed sensors, and shrewd contraptions associated with the web are showing expected development. There will be over 30.5 billion sensor-empowered objects on the web by 2022, as per Global Information Enterprise (IDC). As indicated by gauges, the worldwide IoT market will reach $2 trillion by 2022 and include 53 billion connected wearable gadgets. It is difficult for service providers to handle and process the data created by the increasing number of internet-connected devices. A mature technology, cloud computing offers long-term storage, high data dispensation, and data analytics powered by artificial intelligence (AI), which supports a variety of IoT applications [Petrenko, A. S., Petrenko, S. A., Makoveichuk, K. A., and Chetyrbok, P. V., 2018]. However, because of long transmission postponements and organization blockage from detecting gadgets to the cloud server farms, customary cloud innovation's handling power is becoming inertness inclined and less appropriate for time-basic cutting edge IoT applications like intelligent transportation system/frameworks (ITS), brilliant urban communities, imaginative medical care, savvy agribusiness,

PG & Research Department of Computer Science, Jamal Mohamed College (Autonomous), (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India.

***Corresponding Author:** K. Mohamed Arif Khan, PG & Research Department of Computer Science, Jamal Mohamed College (Autonomous), (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India, E-Mail: kmarif.mca@gmail.com

How to cite this article: Khan, K.M.A., Shanavas, A.R.M. (2024). Optimizing IoT application deployment with fog - cloud paradigm: A resource-aware approach. The Scientific Temper, **15**(4):3225-3233.
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.32

Source of support: Nil

Conflict of interest: None.

Received: 24/10/2024 **Accepted:** 20/11/2024 **Published** : 20/12/2024

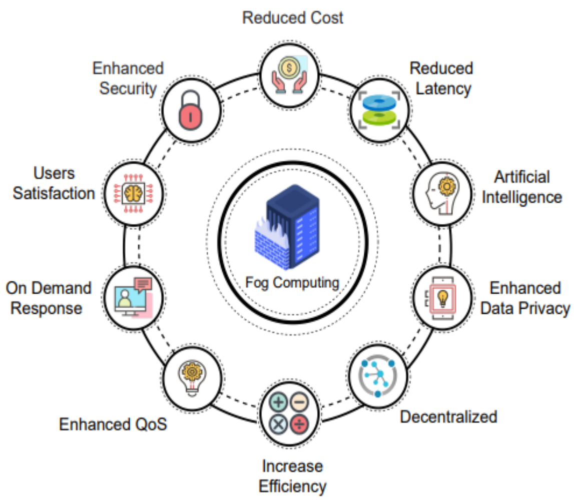

Figure 1: Advantages of fog computing

assembling, development, and mining [Shahid, H., Shah, M. A., Almogren, A., Khattak, H. A., Din, I. U., Kumar, N., and Maple, C., 2021]. Fog computing in a dispersed environment is a novel computing paradigm that Cisco has developed to address these problems. For additional processing and analytics, a number of diverse fog machines can link and send their computer and storage resources to nearby units [Awan, K. A., Din, I. U., Almogren, A., Khattak, H. A., and Rodrigues, J. J., 2021]. According to Figure 1, the key goals of implementing the Low energy and latency are goals of the fog computing paradigm, consumption, lower costs, and improved quality of services (QoS) for service workers and quality of experience (QoE) for end operators.

Incorporation of IoT and Fog computing

A distributed networking architecture is frequently required for the integration of IoT and fog computing in order to gather data from geographically dispersed sources like sensors and data centers [Moysiadis, V., Sarigiannidis, P., and Moscholios, I., 2018]. The cloud is not a workable solution to meet the need for distributed applications because of the high latency rate. IoT devices are reliant on centralized data centers in a planned information system [Awan, K. A., Din, I. U., Almogren, A., Khattak, H. A., and Rodrigues, J. J., 2021]. This system was developed for IoT applications that were integrated. A distributed networking system is frequently used by the IoT and fog computing combo to gather data from geographically dispersed sources like sensors and data centers. It provides IoT services that are more effective and affordable. IoT devices can handle data processing duties to make life easier by lowering the installation and integration costs for sophisticated data processing.

Large-scale sensor networks can be implemented using fog computing, which solves an issue with many IoT devices. This is one of the technology's key advantages. IoT sensors and gadgets were developed by numerous vendors, making it challenging to choose the best parts. However, the setups and needs vary for each IoT application. The performance of IoT devices under different workflow compositions is a notable characteristic.

The newest smart-generation technologies have an impact on the entire business environment. The Internet of Things (IoT) is made up of uniquely recognized smart objects and devices [Moysiadis, V., Sarigiannidis, P., and Moscholios, I., 2018]. Numerous applications, such as waste management systems, intelligent traffic light systems, logistic control systems, emergency services, and industrial control, can benefit from the IoT's relevant solutions [Firouzi, F., Farahani, B., & Marinšek, A., 2022]. The two most appealing applications for IoT are smart healthcare equipment and wearable sensors. Fog computing, as described in Table 1, can tackle a variety of problems [Shahid, H., Shah, M. A., Almogren, A., Khattak, H. A., Din, I. U., Kumar, N., and Maple, C., 2021]. Fog computing increases the performance of the cloud and adds greater flexibility to the level of the end devices on the main network. In order to establish a workable, expandable solution, it also shared processing capabilities.

Research Objectives

- To propose an algorithm of fog computing paradigm.
- To create a model for evaluation of the proposed cloud paradigm.

Review of Literature

CISCO created the fog computing paradigm, which moves data and services from the cloud to the network's edge. Based on distributed computing, it manages data processing, storage, and services at the network's edge devices [Ammad et al., 2020]. Millions of connected devices in the burgeoning IoT generate a lot of data, which needs to be analyzed quickly. Fog computing satisfies these criteria. Fog nodes feature an abstraction layer that conceals device heterogeneity and offers a consistent, programmable interface through virtualization. To coordinate the services and resources among the fog nodes, orchestration is necessary. Fog computing offers IoT apps extra assistance in addition to reducing latency and bandwidth usage. For IoT end devices, fog nodes can be tracked to offer location awareness. It can enable high availability and scalability for extensive IoT applications by being regionally distributed. IoT device mobility is supported via fog computing protocols. Promoting interoperability and flexibility in IoT applications, it also addresses the heterogeneity challenges of IoT [Bose, et al., 2019].

Figure 2 illustrates the end device, fog, and cloud tiers of the three-tiered fog computing-based IoT application architecture. Simple sensors to various types of devices that can be connected to the internet are all included in the end device tier's IoT sensing devices. This tier's primary

Table 1: IoT problems and their resolution through fog computing

Challenges of IoT	*What problems can fog solve*
IoT security challenges	1) A fog system may monitor the security state of adjacent devices and carry out malware-scanning operations. 2) Act as a proxy for updating software credentials and quickly identify threats.
Latency restrictions	Different computation jobs are carried out by the fog, making it the best solution for handling time-sensitive data.
Network capacity limitations	Fog computing can provide hierarchical data processing, enabling the flow of data from the cloud to IoT devices. If applications, networks, and computer resources are readily available when needed, data processing has taken place.
Continuity of services	Even if there is a network connection issue, fog computing can operate autonomously to ensure unbroken services.
Devices with limited resources	When certain tasks cannot be moved to the cloud or used with fog computing can minimise the complexity, cost of ownership, and power consumption of the device.

goal is to gather data about its surroundings and transmit it to the fog tier. Based on distributed computing, the fog tier manages data processing, storage, and services at the network's edge devices, such as access points, gateways, and routers. The cloud tier gets the data from the fog nodes and manages it globally [Santos et al., 2019]. Additionally, it offers data presentation in its ultimate form according to IoT application specifications. Fog computing closes the gap between end devices and the cloud by bringing networking, communication, and distributed computing capabilities closer to the end devices in an IoT environment. Fog computing helps IoT devices stay secure online in addition to the benefits of lower bandwidth usage and latency. Fog nodes give further security to the internet of devices since they operate in the continuum from clouds to devices [Xavier et al., 2020].

Fog computing (FC) is an emerging technology that improves existing distributed computing offices to the organization endpoints to convey diminished inactivity by means of spatial appropriation [Naha et al., 2020]. A message-passing interface is used by the devices participating in distributed computing to facilitate communication and support decentralised models of systems where multiple network devices carry out all computational operations. In distributed computing, a number of novel computation paradigms have evolved.

Mainframe computing, which makes use of batch processing, is the first stage. For the examination of the influence of technology integration capacity, the mainframe environment was suitable [Wei et al., 2021]. Cluster computing was conceived in the early 1960s. Virtualization as a notion dates back to the late 1960s. In the 1990s, a computing paradigm known as "grid computing" and "utility computing"[Haseeb et al., 2021] evolved in which a grid of interconnected computers makes computational decisions collectively. The cloud computing concept is preceded by utility computing. Beginning in the early 2000s, cloud computing [Wei et al., 2021] has gained popularity.

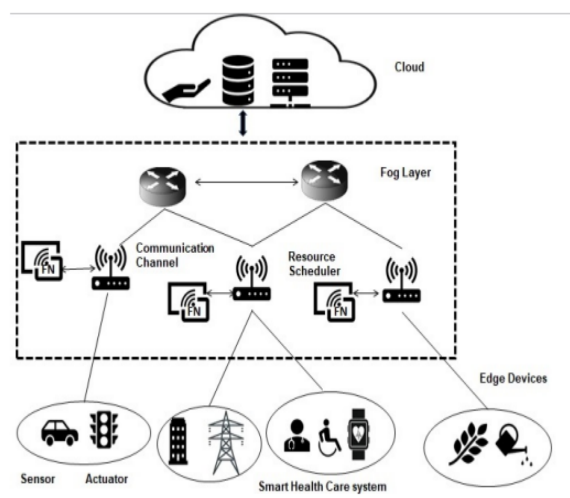

Figure 2: Layered architecture of fog computing

The usage of fog computing could give users faster access. In order to service a wide range of applications, the edge limit of an application upheld the figuring limit of cloudlets [Din et al., 2021]. Cloudlets are tiny computer nodes near the users' base stations that collaborate with the cloud and fog to support a wide range of applications. All of the applications for fog computing are evolving in a way that allows for high-performance computing (HPC) in networked systems [Wang et al., 2019].

Every one of the information and handling connected with every client's PC regularly moves in these organized frameworks when gadgets and clients move starting with one mark of access and then onto the next [Tejaswini et al., 2018]. Users may find it easier to retrieve their data in urgent situations with the help of data migration. There are several sensitive situations where delays might result in hazardous conditions, such as in healthcare and transportation systems [Haseeb et al., 2021]. The fog computing paradigm offers quick access to resources for all time-centric applications and the management of resources to improve utilization in order to get the best performance for the least amount of money. Utilizing resources effectively is important for many reasons, including resource management, cost, and response time. However, putting fog computing into practice in a real-time situation is exceedingly difficult. The processing of resources is complicated by the large volume, data velocity, and variety, which might have an impact on resource utilization [Muheidat et al., 2018].

The Architecture of Fog Computing

An effective computing model called fog computing uses distributed storage, computing, and networking services to link IoT gadgets with cloud servers. These services are partially located at the network edge, though. A decentralized method called fog computing primarily supports applications close to edge devices. Numerous fog computing constructions have been documented in the works over time. However, we concentrate on the three-layer typical fog computing architecture in our survey. The OpenFog group has identified the 3-tier fog computing planning as the most significant, trustworthy, and practical processing platform and analyzing IoT applications [Apat et al., 2020]. The following discussion focuses on the three layers of the fog computing building, which are depicted in Figure 2.

- • Tier 1. IoT Hardware/Software: This tire is made up of different IoT device types, including sensors (like smoke detectors, temperature sensors, and humidity sensors), as well as different smart device types, such as smartphones, self-driving cars, smart home appliances, smart healthcare equipment, etc. Terminal nodes are a common name for these sophisticated equipment and sensors. The basic function of IoT devices is to produce real-time IoT software and information is collected from the environment and sent to computer processors (such as cloud data centers or fog devices) for storage and analysis. The IoT gadgets are presumably dispersed over the world and outfitted with GPS.
- • Tier 2. Fog devices: Included in Tier 2's fog devices are mobile phones, tablets, laptops, desktop computers, notepads, and different-edge equipment (such as routers, modems, switches, and gateways). The limited processing and storage capabilities of the heterogeneous fog devices are dispersed throughout the network. The majority of applications that require both real-time and delay should opt to dump their compute data in fog campaigns due to the distributed nature. Each fog node, however, is only able to process and disseminate the tiny event-based and delay-intensive apps locally to other fog devices that are within its series. Applications that depend on resources and processing ought to be moved to the cloud data center. The first goal of fog devices is to carry out user requests. However, fog devices occasionally also distribute whole or partitioned data among the other fog devices in the area. However, by raising the computational burden on other fog devices, this tactic reduces latency overall.
- Tier 3. Cloud data centre: Tier cloud data centers with numerous The third layer of the architecture consists of a number of diverse cloud servers for processing information and a set of cloud-based warehouses for long-term storage of data. Due to the cloud data center's massive processing and storage capability, a great deal of dependent on resources next-generation IoT programs that depend on the analytical findings of the vast past information set should be offloaded there. The critical, processed data is also discharged from the fog and IoT devices to the cloud warehouse for long-term archival [GJ, B. K., 2018].

Methodology

An emerging idea called fog-cloud connects resource-constrained fog devices to cloud servers with abundant resources to perform IoT applications. The fog-cloud system gives constrained processing power over fog devices placed close to the nodes of the sensor while centralising resourceful cloud servers. The fog-cloud architecture is useful for implementing applications on a large scale since it distributes resources in a decentralized manner. This paradigm allows mobility, reduced network load, and minimal latency for the development of IoT applications. Figure 2 depicts the paradigm of fog-cloud computing in general, with resource-constrained fog nodes supplying resources near the edge devices. The cloud server functions as a consolidated unit for the collecting of all the data coming from the edge nodes after going through initial processing by fog nodes. The proposed algorithm is executed in the iFogSim-Eclipse tool.

Proposed Algorithms

We provide two integrated strategies to enable resource-aware deployment of application modules using the IoT fog-cloud paradigm.

Algorithm 1: The module mapping algorithm allows for the placement of fog clouds. It provides an effective network infrastructure map of an application's modules [Shukla et al., 2019]. The organization hubs and application modules are arranged in climbing request as per their ability and prerequisite, separately, in the wake of being given the arrangements of organization hubs N and application modules V. The subsequent stage is to construct a key-esteem pair with Organization Hub as the key and Application Module as the worth.

Algorithm 2: The cluster head selection algorithm executes all the application modules that must be installed, initializing the network and sorting energy and distance in each iteration, with high energy chosen as the cluster head and less distance. When the cluster or sensor ID is less than 10, just data is transferred.

Clustering creates a hierarchy of clusters, or collections of sensing nodes, that gather and send data to their cluster heads (CH). The base station (BS), which serves as the intermediary between the end user and the network, receives the data after it has been fused and grouped by the CH. For the energy-constrained network, the clustering technique is crucial for power conservation. By selecting a cluster head, the network's load can be balanced effectively, lowering energy use and extending cluster lifetime [Awaisi et al., 2019].

Proposed Paradigm

This study proposes a resource-aware Fog computing paradigm for IoT applications that efficiently controls the link between parental controls and edge nodes by taking into consideration the volume of sensed data at end devices and the processing abilities at the fog layer that are available [GJ, B. K., 2018]. The suggested technique looks over the entire edge layer before allocating suitable each fog device already present in the network with edge devices. In order to equalise the handling load on mist hubs in like manner to their processing limit, the strategy puts edge hubs underneath haze hubs because of the distinguishing pace of sensors situated at the edge hubs. In the suggested method, edge devices are registered and divided into edge nodes with both low and high sensing rates. Then, based on the resources available at the fog nodes, a combination of edge nodes from the categorized edge nodes is assigned to the fog devices [Hassan et al., 2022].

The proposed approach receives input from fog nodes and edge devices. The program first divides the edge devices into categories based on the sensors' rate of detection that are connected to these nodes of edge. The edge gadget is put in the set KL if its sensing rate is lower than the predetermined rate; otherwise, it is positioned in the set KH [Apat et al., 2020]. The program then looks for the best edge devices by examining the sets KH and KL as a whole. For optimal performance, the algorithm distributes appropriate edge expedients to fog nodes. An edge device is designated to a fog device by way of child nodes if the resources needed for processing the sensed capacity by the edge ploy are fewer than those at the fog device [Perala, S. S. N., Galanis, I., and Anagnostopoulos, I., 2018].

According to the amount of data felt by the child devices, the suggested method assigns them to the parent fog nodes. Considering that the volume of the deliberate burden is associated with the identifying pace of the detecting gadget, edge gadgets with higher detecting rates are dispensed to haze hubs with better information handling limits to diminish the tension in the organization. As a consequence, the load is evenly distributed with the available network possessions, reducing the system's total stress. Contrarily, the cloud architecture directly transfers all perceived load for processing to a cloud server, which causes heavy network traffic utilization. The standard fog paradigm reduces network consumption relative to the cloud model but is fewer networks effective than the suggested model because fog resources cannot be provided according to sensed load [Azizi et al., 2019].

Algorithm 1: Proposed module mapping

```
Function Module Map
Generate placement requests
    List<Placement Request> placement Requests = new Array
List<> ();
      for (Sensor s: sensors)
         if (s.get Id () > 10) then
         Map<String, Integer> placed Microservices Map = new
HashMap<> ();
         Placed Microservices Map. put («sensor Module», s.get
Gateway Device Id());
         Placement Request p = new Placement Request (s.get
App Id(), s.get Gateway
            Device Id (), s.get Gateway Device Id(), placed
Microservices Map);
         Placement Requests. add(p);
         end if
      end for
Return (module map)
End Function
```

Results and Discussion

A sophisticated surveillance program is put into place on various scales to verify the efficacy of the suggested plan. The no. of cameras watching the region under investigation is increased in each testing scenario. Throughout all of the simulations, a total of 7 regions are beneath surveillance. The cameras are linked to fog nodes connected to the cloud server in each of the simulated scenarios. The network allocates one fog device per area under surveillance, providing resources near the network's edge to monitor and detect activity there. Each physical topology generated by the simulation has a different no., of cameras per monitored area. Each fog node has a starting connection of two cameras, which is augmented with each subsequent topology. The program was utilized in network topologies that we provided as a JavaScript Object Notation (JSON) file to evaluate the suggested algorithm. Three distinct network topologies and various workloads were used to vary the scenario; the graphical representation of one of these topologies, as generated by iFogSim, is shown in Fig. 3. The experiment iterations on configurations with 2, 4, and 6 Fog channels used two devices each Fog gateway. Tables 2, 3, and 4 contain the experimental network configurations. Figure 3 Iterative deployment is one of the network

Algorithm 2: Cluster head selection

Start
Initialization network
(Position, Energy)
Collecting the data of each sensor
While the energy and the sensor == 0
Clustering based on the energy and distance
Cluster head chosen
Transfer the data to the cluster head
The cluster head transfers the data to the base station
Energy Reduce while transferring the data
End

topologies employed. These topologies have been used in the simulation, each with different workloads but essentially the same standardised network structure.

Energy consumption Comparison

The application response time (delay), network utilization, and power usage of the modeled applications using both placement approaches were compared between the projected Fog-Cloud placement method (Cluster head selection Algorithm) and the conventional Cloud-based placement methodology. The simulation's results (Figure 4) show that the proposed placement technique has a hugely positive influence on network utilization, response time for applications, and energy use across all the above network topologies used.

Effective module mapping had a significant impact on end-to-end latency as well, with highly favorable outcomes for the placement of fog clouds using the chosen strategy, as shown in Figure 4. It illustrates the variance in energy consumption between the two placement strategies. Using the suggested strategy, we aim to balance the energy consumption at high-cost (Cloud data centers) and low-cost (Fog layer) sites. The comparison of energy consumed based on some components, was done between the proposed algorithm of the Fog computing paradigm and the traditional fog computing paradigm. The graph was

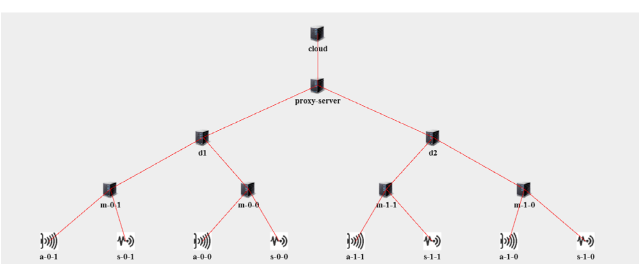

Figure 3: Model created for the assessment of the proposed cloud paradigm

Table 2: Proposed network configuration – Latency

Component between		*Latency (ms)*
Cloud	Proxy Server	50
Proxy server	Fog Device Gateway	10
Fog device gateway	Device (0,1)	10
Device (0,1)	Sensor (0,1)	2
Device (0,1)	Actuator (0,1)	3

plotted between energy consumed and device, sensor, and actuators. The proposed algorithm is showing the significant results of simulations.

The canny observation application is developed on both the novel load-aware utilisation of resources fog-cloud paradigm and the conventional cloud and cloud computing fog paradigms for assessment. The cameras used in the simulations have information detection frequencies that range from 5 to 20 ms. The proposed method is compared to conventional cloud and fog paradigms by developing a number of simulation situations at various scales. The variables being monitored in all of these tests are the end-to-end delay, network consumption, and cloud processing cost. A comparison of the network usage when the application is implemented using various paradigms is shown in Figure 4. Comparing the suggested algorithm to fog and cloud-based implementations, the network burden is successfully reduced.

Network Consumption Comparison

In a cloud-based implementation, the cloud server must process all of the sensed records from the system, increasing latency in direct proportion to the quantity of cloud-connected sensors. However, in a fog-based computing paradigm, the fog nodes also perform data processing at the intermediate level, bringing down the volume of information that should be dealt with by the cloud server and advancing full circle times. The laid out procedure's primary goal is to reduce dormancy and stress on the organization by giving reasonable mist resources for the edge hubs that compare with the rate that the data seen by the edge hubs to the

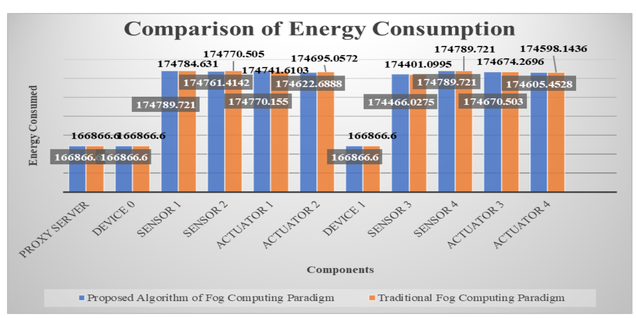

Figure 4: Comparison of energy consumption

Table 3: Proposed network configuration – Values of device

Components	*Upstream (Mbps)*	*Downstream (Mbps)*	*RAM (MB)*	*MIPS*
Cloud	1000	10000	6000	20000
Proxy server	10000	10000	4000	8000
Fog device gateway	10000	10000	4000	6000
Device (0,1)	100	10000	2048	2000
Rate per execution	0.01	0.0	0.0	0.0

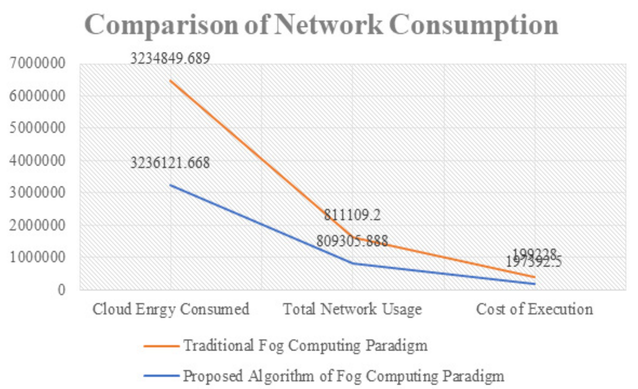

Figure 5: Network usage, cost consumption, overall energy consumption of proposed and traditional paradigm

Table 4: Energy consumed

Proxy server	*166866.6*	*166866.6*
Device 0	166866.6	166866.6
Sensor 1	174789.721	174784.631
Sensor 2	174761.4142	174770.505
Actuator 1	174770.155	174741.6103
Actuator 2	174622.6888	174695.0572
Device 1	166866.6	166866.6
Sensor 3	174466.0275	174401.0995
Sensor 4	174789.721	174789.721
Actuator 3	174670.503	174674.2696
Actuator 4	174605.4528	174598.1436

Table 5: Network usage, cost consumption, overall energy consumption of proposed and traditional paradigm

Cloud energy consumed	3236121.668	3234849.689
Total network usage	809305.888	811109.2
Cost of execution	197392.5	199228

amount of cloud-associated sensors. In any case, in a haze-based figuring worldview, the haze hubs likewise perform information handling at the middle-of-the-road level, bringing down the volume of information that should be taken care of by the cloud server and advancing quickly full circle times. The laid-out system's fundamental goal is to decrease inactivity and stress on the organization by giving reasonable mist resources for the edge hubs that compare with the rate that the data seen by the edge hubs. By inspecting the detecting recurrence of the sensors situated at the edge gadgets, the suggested model calculates the amount of information entering from the edge devices. The program then connects the appropriate gadgets to add fog to the node sensors based on the fog devices' resources. There is a consumption of network comparison based on cost consumption and the energy consumption is given in table 5. As shown in Figure 5, the suggested policy lowers the quantity of data that processing must be placed on the cloud server by allocating appropriate fog resources in response to edge device demand. This lowers the cost of processing in the cloud. The comparative graph based on network usage, cost consumption, and overall energy consumption between the proposed and traditional algorithm has been illustrated in Figure 5.

Conclusion

Effective Deployment of IoT Modules

We demonstrated the successful deployment of application modules for IoT-based apps on fog cloud infrastructure, showcasing efficient utilization of network infrastructure resources.

Addressing Latency Issues

Fog computing as an emerging paradigm, effectively addresses latency in time-sensitive IoT applications while managing the strain on network resources due to the massive increase in IoT usage across various industries.

Categorizing Static Elements

We increased network efficiency and broadened the scope of these applications by categorizing and considering static elements, outlining the essential features that impact the performance of IoT applications.

Cluster Head Selection Algorithm

The proposed Cluster Head Selection Algorithm outperforms the typical Brute Force approach, which is often NP-hard, by leveraging its logarithmic complexity to manage the connection between edge devices and fog nodes.

Reducing Latency and Network Consumption

The proposed algorithm effectively manages the network's processing resources and detected load, significantly reducing system latency and network consumption.

Simulation and Comparison

Using iFogSim toolbox, we constructed simulations of a distributed camera network application for intelligent surveillance at various scales, demonstrating that the proposed algorithm significantly lowers processing costs, delay and network consumption compared to conventional fog architectures and cloud.

Future Work and Enhancements

Future work will involve implementing more applications of the proposed design, modifying the method to study the various parameters, addressing system node failures and integrating dynamic Directed Acyclic Graph (DAG) characteristics. Additionally, we plan to explore scheduling practices for resources on Fog Devices post-deployment.

Acknowledgments

I am deeply grateful to all those who contributed to the successful completion of this research paper titled «OPTIMIZING IOT APPLICATION DEPLOYMENT WITH FOG - CLOUD PARADIGM: A RESOURCE AWARE APPROACH.»

I would like to extend my sincere gratitude to my supervisor, Dr. A.R. Mohamed Shanavas, whose guidance, support and encouragement have been invaluable throughout this research. His expertise and insights have been fundamental to shaping this work.

Conflict of Interest

The authors of this research paper titled «optimizing IoT application deployment with fog - cloud paradigm: a resource-aware approach» declare that there is no conflict of interest regarding the publication of this paper.

References

Ammad, M., Shah, M. A., Islam, S. U., Maple, C., Alaulamie, A. A., Rodrigues, J. J., & Tariq, U. (2020). A novel fog-based multi-level energy-efficient framework for IoT-enabled smart environments. *IEEE Access*, *8*, 150010-150026.

Apat, H. K., Bhaisare, K., Sahoo, B., & Maiti, P. (2020, March). Energy-efficient resource management in fog computing supported medical cyber-physical systems. In *2020 International Conference on computer science, Engineering and Applications (ICCSEA)* (pp. 1-6). IEEE.

Awaisi, K. S., Abbas, A., Zareei, M., Khattak, H. A., Khan, M. U. S., Ali, M., ... & Shah, S. (2019). Towards a fog-enabled efficient car parking architecture. *IEEE Access*, *7*, 159100-159111.

Awan, K. A., Din, I. U., Almogren, A., Khattak, H.A., & Rodrigues, J. J. (2021). EdgeTrust-a lightweight data - centric trust management approach for the green internet of edge things. PREPRINT available at Research Square.

Azizi, S., Khosroabadi, F., & Shojafar, M. (2019). A priority-based service placement policy for fog-cloud computing systems. *Computational Methods for Differential Equations*, *7*(4 (Special Issue)), 521-534.

Bose, S., Sarkar, D., & Mukherjee, N. (2019). A framework for heterogeneous resource allocation in the sensor-cloud environment. *Wireless Personal Communications*, *108*, 19-36.

Din, I. U., Bano, A., Awan, K. A., Almogren, A., Altameem, A., & Guizani, M. (2021). LightTrust: lightweight trust management for edge devices in the industrial Internet of Things. *IEEE Internet of Things Journal*.

Firouzi, F., Farahani, B., & Marinšek, A. (2022). The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT). *Information Systems*, *107*, 101840.

GJ, B. K. (2018, December). Internet of Things (IoT) and cloud computing based persistent vegetative state patient monitoring system: a remote assessment and management. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (pp. 301-305). IEEE.

Haseeb, K., Din, I. U., Almogren, A., Ahmed, I., & Guizani, M. (2021). Intelligent and secure edge-enabled computing model for sustainable cities using green internet of things. *Sustainable Cities and Society*, *68*, 102779.

Hassan, S. R., Ahmad, I., Nebhen, J., Rehman, A. U., Shafiq, M., & Choi, J. G. (2022). Design of latency-aware IoT modules in heterogeneous fog-cloud computing networks. *Cmc-Computers Materials & Continua*, *70*(3), 6057-6072.

Moysiadis, V., Sarigiannidis, P., & Moscholios, I. (2018). Towards distributed data management in fog computing. *Wireless Communications and Mobile Computing*, *2018*. Moysiadis, V., Sarigiannidis, P., & Moscholios, I. 2018(1).

Muheidat, F., Tawalbeh, L. A., & Tyrer, H. (2018, January). Context-aware, accurate, and real-time fall detection system for elderly people. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)* (pp. 329-333). IEEE.

Naha, R. K., Garg, S., Chan, A., & Battula, S. K. (2020). Deadline-based dynamic resource allocation and provisioning algorithms in the fog-cloud environment. *Future Generation Computer Systems*, *104*, 131-141.

Perala, S. S. N., Galanis, I., & Anagnostopoulos, I. (2018, May). Fog computing and efficient resource management in the era of Internet-of-Video Things (IoVT). In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE.

Petrenko, A. S., Petrenko, S. A., Makoveichuk, K. A., & Chetyrbok, P. V. (2018, January). The IIoT/IoT device control model is based on narrow-band IoT (NB-IoT). In *2018 IEEE Conference of Russian Young Researchers in Electrical and electronic engineering (EIConRus)* (pp. 950-953). IEEE.

Santos, I. L., Pirmez, L., Delicato, F. C., Oliveira, G. M., Farias, C. M., Khan, S. U., & Zomaya, A. Y. (2019). Zeus: A resource allocation algorithm for the cloud of sensors. *Future Generation Computer Systems*, *92*, 564-581.

Shahid, H., Shah, M. A., Almogren, A., Khattak, H. A., Din, I. U., Kumar, N., & Maple, C. (2021). Machine learning-based mist computing enabled the internet of battlefield things. *ACM Transactions on Internet Technology (TOIT)*, *21*(4),1-26.

Shukla, S., Hassan, M. F., Khan, M. K., Jung, L. T., & Awang, A. (2019). An analytical model to minimize the latency in healthcare internet-of-things in the fog computing environment. *PloS one*, *14*(11), e0224934.

Tejaswini, S., Sriraam, N., & Pradeep, G. C. M. (2018, October).

Cloud-based framework for pain scale assessment in NICU-a primitive study with infant cries. In *2018 3rd International Conference on Circuits, Control, Communication, and Computing (I4C)* (pp. 1-4). IEEE.

Wang, T., Liang, Y., Jia, W., Arif, M., Liu, A., & Xie, M. (2019). Coupling resource management based on fog computing in smart city systems. *Journal of Network and Computer Applications*, *135*, 11-19.

Wei, W., Yang, R., Gu, H., Zhao, W., Chen, C., & Wan, S. (2021). Multi-objective optimization for resource allocation in vehicular cloud computing networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 25536-25545.

Xavier, T. C., Santos, I. L., Delicato, F. C., Pires, P. F., Alves, M. P., Calmon, T. S., ... & Amorim, C. L. (2020). Collaborative resource allocation for Cloud of Things systems. *Journal of Network and Computer Applications*, *159*, 102592.