# Optimizing Bandwidth Cost, Energy Consumption, Delay, and Load in a Cooperative Fog Environment

Maisha Manarat[1], Mirza Mohd Shahriar Maswood[1]
[1]Dept of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh
Email: maishamanarat@gmail.com, shahriar@ece.kuet.ac.bd

***Abstract*—Fog computing is an emerging architecture that provides storage, processing power, and networking capabilities for Internet-of-Things (IoT) devices. Devices using fog computing can assign computationally demanding tasks and data to adjacent fog nodes. This approach can reduce the bandwidth cost, energy consumption, and delay. However, to reduce one cost, another cost might increase, such as reducing energy consumption may increase the delay. Therefore, balancing these costs efficiently is crucial in fog computing. In this paper, we present a joint optimization strategy for bandwidth cost, energy consumption, delay minimization, and load balancing in a cooperative fog environment. Our approach involves using a Mixed-Integer Linear Programming (MILP) model to optimize the total cost and distribute the workload effectively among the fog nodes. We present a comprehensive evaluation of our approach, using simulations to demonstrate its effectiveness in reducing energy consumption, delay, and bandwidth cost. Our results demonstrate that our method can notably enhance the performance of fog computing systems, making them more efficient and cost-effective for Internet of Things applications.**

***Index Terms*—Edge Computing, Fog Computing, IoT, Delay minimization, Load Balancing, Energy efficiency, Cloud Computing, Optimizing Cost**

## I. Introduction

It is anticipated that the Internet of Things (IoT) will emerge as the Internet of the future given its recent explosive growth. With billions of IoT devices set to be installed by 2025 [1], these interconnected devices produce massive volumes of data that require instantaneous processing and analysis. However, handling all this data in the cloud may lead to network-wide traffic and issues due to a lack of storage resources. Additionally, smart devices have limitations in processing capacity, battery life, storage, and bandwidth, making cloud computing an attractive solution for providing low-cost, scalable resources to applications. Nevertheless, cloud computing has its own limitations and cannot solve all problems. For instance, real-time applications such as streaming, gaming, and augmented reality are too latency-sensitive for cloud deployment. Furthermore, IoT applications often require mobility, location awareness, and geo-distribution which pose challenges to cloud deployment.

Fog computing, defined by Cisco as the expansion of cloud computing to the edges of the network, enables real-time applications, frequent services, and mobile big data analysis. Fog is essentially a more localized version of the cloud, situated nearer to the network edge. As one of the paradigms of Mobile Edge Computing (MEC), fog computing lowers transmission latency by deploying computers near the network edge. However, efficiently coordinating the cooperation between fog nodes remains a challenging task. To address the challenges posed by cloud deployment, fog computing has emerged as a promising solution.By extending cloud computing from the network's core to its periphery, fog computing becomes closer to the location of IoT devices. This approach minimizes latency, critical for real-time applications, frequent services, and mobile big data analysis. Fog computing deploys servers at the edge of networks to reduce transmission and propagation latency. However, the efficient cooperation of fog nodes remains a challenging problem.

To tackle this issue, in this paper, we introduce a novel Mixed-Integer Linear Programming (MILP) based optimization model to minimize a composite objective function. Our goal is to reduce the cost of bandwidth, power consumption, delay, and load balancing by utilizing two different kinds of resources: server resources (CPUs' processing capacity) and network resources (bandwidth). We use several weight factors connected to each goal in the composite objective function to assess performance.

The remnant of the paper is structured as follows: Section II discusses related works, Section III explains the problem formulation, Section IV delineates the simulation study setup and result analysis, and Section V concludes the research and outlines future directions for this study.

## II. Related Works

Since its introduction in 2012, fog computing, a comparatively new academic area, has garnered a lot of attention. Numerous studies have been conducted on optimizing resource allocation and management in fog computing networks.

Several studies have been conducted to optimize the deployment of Internet of Things (IoT) applications on fog infrastructure. In order to maximize cost efficiency, a resource allocation strategy was developed in [2], which concentrated on resource allocation in fog computing networks. The researchers proposed an autonomic methodology for efficient IoT service placement [3]. Utilizing the advantages of fog computing for highly computational activities on mobile robotic sensors/actuators was the topic of research [4]. The computational load among various Edge/Fog nodes is divided to reduce battery usage and computational latency. The computational data is optimally distributed across peer robots by a fog node (server node) taking into account runtime restrictions including mobility, position, battery capacity, and available processing power. In [5], a joint optimization algorithm for the access system was proposed, which focused on reducing transmission delay and processing delay between User Equipment (UEs), fog nodes, and fog servers. Similar to this, [6] took into account how to allocate radio and computational resources to meet as many user requests as feasible while minimizing process complexity and power consumption for Small Cell Clouds (SCCs). There were two phases in the resource allocation procedure. Initially, a metric-based scheduling rule was used to determine how resources should be distributed to support SCs. Using a scheduling strategy and an optimization target, compute clusters for unfulfilled requests were constructed optimally in the second stage. A MILP formulation was created in [7] to distribute resources to clients as efficiently as possible while reducing location-dependent costs like resource provision and carbon emission costs. The study specifically took into account the servers' idle power consumption, which reflected the variability of their power consumption.

In previous studies, different optimization algorithms have been proposed to optimize resource allocation in cloud and fog computing environments. For instance, [8] proposed a MILP formulation to optimize resources between the network and data center. Multiple algorithms were used to solve the optimization problem. In [9], the Alternating Direction Method of Multipliers (ADMM) algorithm was used to reduce the energy cost, response time, and rental cost of cloud resources. A different study, [10], used the Marine Predator's Algorithm (MPA), a disruption operator, and the Chimp Optimization Algorithm (ChOA) to find the best way to schedule IoT application-related tasks in a fog environment.

The following works are related to the optimization of fog computing resources for various objectives. The Multi-Objectives Grey Wolf Optimizer (MGWO) algorithm was applied in [11] to optimize the work distribution of the fog broker in order to minimize energy consumption and latency. [12] addressed the issue of bandwidth wastage in the distributed deployment of application components by considering their dependencies. In order to balance the burden of Internet of Things jobs over fog nodes while taking response time and communication cost into account, [13] suggested two nature-inspired meta-heuristic scheduling algorithms (Ant Colony Optimization and Particle Swarm Optimization). [14] proposed a non-linear model to allocate cloud-to-fog resources based on factors like execution efficiency, service response rate, and reboot rate. [15] focused on mixed-criticality applications with safety criticality and time-dependent performance requirements. Finally, [16] optimized bandwidth cost and load balancing in fog computing resources using MILP technique.

Although these studies considered several issues linked to cloud and fog computing environments, the optimization of bandwidth cost, power consumption, delay, and load balancing in a cooperative fog environment has not been jointly considered. This study aims to address this gap by proposing a joint optimization framework for minimizing these factors in a cooperative fog environment.

## III. PROBLEM FORMULATION

In our proposed system, the fog servers act as intermediate computational nodes, providing resources to handle various situations such as emergency scenarios like natural disasters or

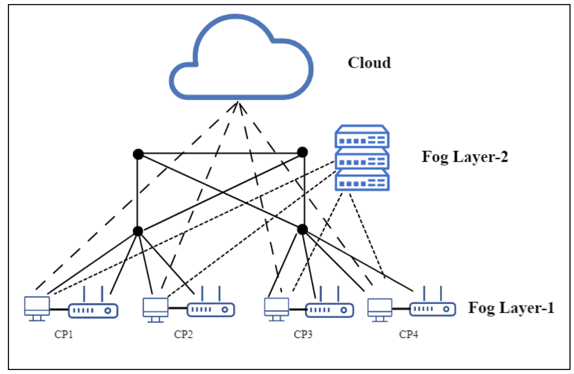


Fig. 1. Three-layer fog-cloud topology

fire breakouts, which require additional compute and network resources with a focus on speed. Conversely, for scenarios such as processing information from industrial equipment or monitoring traffic rule-breaking, bandwidth becomes the primary requirement.

Therefore, we propose a cooperative fog computing architecture that allows for borrowing additional resources from neighbouring regions of the same fog layer, specifically layer-1 as shown in Fig. 1, provided that resources are available and idle. Moreover, services can be accessed from a higher fog layer (layer-2) perhaps even the cloud layer in cases where resources in the same fog layer are insufficient.

### A. Constraints

In our model, we consider that each request consists of 2-tuple <h, r>. Here $h$ represents the bandwidth demand of the request, and $r$ represents the amount of processing resources required from a serving host. $h_i$ represents the entire bandwidth demand coming from cluster point $i$. Thus $h_i$ can be expressed mathematically as:

$$\sum_{f \in F} y_i^f = h_i; \quad i \in I \tag{1}$$

$x_{ip}^f$ is a path flow variable that represents the bandwidth assigned to a certain path $p$ from cluster point $i$ to fog server $f$:

$$\sum_{p \in P} x_{ip}^f = y_i^f \qquad ; i \in I, f \in F \tag{2}$$

When any bandwidth is allocated from any cluster point $i$ on a particular path $p$ to satisfy a portion of the bandwidth requirement $h_i$, then all links connected with that path have to bear that portion of the demand $h_i$. Thus the flow on every $i \in I$ connection is:

$$\sum_{p \in P} \delta_{ipl}^f x_{ip}^f = z_{il}^f \qquad ; i \in I, f \in F, l \in L \tag{3}$$

The total bandwidth requirement from a link $l$ must not cross the capacity of that link times the maximum link utilization. To provide link-level load balancing, this constraint is necessary.

$$\sum_{i \in I} \sum_{f \in F} z_{il}^f \leq c_l u \qquad l \in L \tag{4}$$

TABLE I
NOTATIONS USED IN FORMULATION

| **Constants:** | |
|---|---|
| **Symbol** | **Definition** |
| F | Set of fog servers |
| $I_f$ | Set of cluster points |
| Q | Set of frequencies in which a particular server can run |
| $L_f$ | Set of links |
| $h_i$ | Total Bandwidth demand for $i$ |
| $c_l$ | Available capacity on link $l$ |
| $X_l$ | Maximum queuing delay of link $l$ |
| $D_i$ | Maximum allowable delay |
| K | Set of line segments of convex delay curve |
| $\rho_l$ | Entire delay of the link $l$ |
| D | Entire allowable delay |
| $s_l$ | Utilization of link |
| $d_l$ | Queuing delay of the link due to buffering |
| $X_l$ | Maximum queuing delay of link $l$ |
| $\hat{X}_l$ | A fixed constant static delay of link $l$ |
| M | A large positive number |
| $\varepsilon$ | A very small positive number |
| $\delta^f_{ipl}$ | Link path indicator |
| $b^f_q$ | Power consumption at frequency $q$ |
| $a^f_q$ | Capacity of fog center f at frequency $q$ |
| $\theta^1_k$ | Coefficient of the linear function that approximates the convex delay curve for the kth line segment |
| $\theta^2_k$ | Coefficient of the linear function that approximates the convex delay curve for the kth line segment |
| $\alpha$ | Weighting factor |
| $\mu$ | Weighting factor |
| $\Omega$ | Weighting factor |
| $\gamma$ | Weighting factor |
| **Variables:** | |
| $y^f_i$ | Bandwidth allocation from $i$ to $f$ |
| $x^f_{ip}$ | Bandwidth allocation in path p from $i$ to $f$ |
| $z^f_{il}$ | Bandwidth requirement on link $l$ from $i$ to $f$ |
| $e^f$ | CPU processing requirement |
| $g_f$ | Server resource allowance for a fog server |
| $\tilde{z}^f_{il}$ | Binary decision variable to indicate link used $f$ from $i$ to f |
| $w^f_{iq}$ | Binary decision variable for choosing the optimum frequency $q$ from the range of available frequencies of $f$ servers |
| $\tilde{y}^f_l$ | Binary decision variable to indicate that traffic originated from cluster point, $i$ is served by fog server $f$ |
| $\phi^f_{il}$ | Total delay of link $l$ from $i$ to f |
| $\tilde{v}_l$ | Binary decision variable to indicate if link $l$ is used to satisfy any request |
| u | Maximum utilization of any link |

For any link, the maximum link utilization cannot be more than 1 at any point.

$$u \leq 1 \tag{5}$$

Constraints 6 and 7 are used for the identification of the links that are used to establish the paths from cluster point $i$ to fog server $f$.

$$z^f_{il} \leq M\tilde{z}^f_{il} \qquad ;i \in I, f \in F, l \in L \tag{6}$$

$$z^f_{il} \geq \varepsilon\tilde{z}^f_{il} \qquad ;i \in I, f \in F, l \in L \tag{7}$$

The overall quantity of the resource demand generated by the requests from cluster point $i$ is the summation of the resources allocated to all the fog servers f selected to satisfy the requests:

$$\sum_{f \in F} g^f_i = r_i \qquad ,i \in I \tag{8}$$

$e^f$ is the CPU processing capacity required from a fog server $f$ to satisfy the request coming from a cluster point:

$$\sum_{i \in I} g^f_i = e^f \qquad ,f \in F \tag{9}$$

Now we are adding a binary shadow variable $\tilde{y}^f_i$ corresponding to $y^f_i$ to track one-to-one mapping from cluster point $i$ to fog server f using a small positive number $\varepsilon$ and a large positive number M.

$$y^f_i \leq M\tilde{y}^f_i \qquad ;i \in I, f \in F \tag{10}$$

$$y^f_i \geq \varepsilon\tilde{y}^f_i \qquad ;i \in I, f \in F \tag{11}$$

We then discuss the resource allocation of $g$ to the correct tuple $<$ i, $f$ $>$, ensuring that this is in line with the shadow $\tilde{y}^f_i$ variable

$$g^f_i \leq M\tilde{y}^f_i \qquad ;i \in I, f \in F \tag{12}$$

$$g^f_i \geq \in \tilde{y}^f_i \qquad ;i \in I, f \in F \tag{13}$$

The total required resource must be less or equal to the available resource:

$$e^f \leq \sum_{q \in Q} a^f_q w^f_q \qquad ,f \in F \tag{14}$$

Lastly, a specific fog server $f$ running at a particular frequency will produce a specific capacity $a^f_q$. However, a fog server can not run multiple frequencies at a time:

$$\sum_{q \in Q} w^f_q \leq 1 \qquad ,f \in F \tag{15}$$

The utilization of link $l$ can be written mathematically as:

$$s_l = \frac{\sum_{i \in I}\sum_{f \in F} z^f_{il}}{c_l} \tag{16}$$

Constraints (17),(18), and (19) are used to classify all available fog server links which are used to fulfill the requests.

$$\tilde{v}_l \geq \tilde{z}^f_{il} \qquad l \in L, i \in I, f \in F \tag{17}$$

$$\tilde{v}_l \leq \sum_{i \in I}\sum_{f \in F} \tilde{z}^f_{il} \qquad l \in L \tag{18}$$

$$\tilde{v}_l \leq 1 \qquad l \in L, f \in F \tag{19}$$

Constraints (20) and (21) are used to calculate the piece-wise linear queuing delay on the links with the use of the coefficient $\theta_k^1$ and $\theta_k^1$.

$$\theta_k^1 + \theta_k^2 s_l \leq d_l + (1 - \tilde{v}_l) X_l \qquad k \in K, l \in L \tag{20}$$

$$d_l \leq X_l \tilde{u}_l \tag{21}$$

The total delay of link the l is the summation of the queuing delay $d_l$ and static latency $\hat{X}_l$

$$\rho_l = d_l + \hat{X}_l \qquad l \in L \tag{22}$$

Constraints (23) to (26) are used to make sure that only the delay of the links that are used to satisfy the request is considered.

$$\phi_{il}^f \leq M \tilde{z_{il}^f} \qquad i \in I, f \in F, l \in L \tag{23}$$

$$\phi_{il}^f \leq \rho_l \qquad i \in I, f \in F, l \in L \tag{24}$$

$$\phi_{il}^f \geq \rho_l - (1 - \tilde{z_{il}^f}) M \qquad ; i \in I, f \in F, l \in L \tag{25}$$

$$\phi_{il}^f \geq 0 \qquad ; i \in I, f \in F, l \in L \tag{26}$$

The constraint (27) is used to satisfy that the total delay must not exceed the maximum allowable delay.

$$\sum_{l \in L} \sum_{f \in F} \phi_{il}^f \leq D_i \qquad i \in I \tag{27}$$

### B. Objective Function

This work aims to achieve the following four objectives: (i) to minimize bandwidth cost of routing; (ii) to minimize the maximum server resource utilization, (iii) to minimize maximum link utilization of network links, and (iv) to minimize delay. These goals are presented with the composite objective function:

$$\begin{aligned} \min \alpha \sum_{f \in F} \sum_{l \in L} \sum_{i \in I} z_{il}^f + \mu \sum_{f \in F} \sum_{q \in Q} b_q^f w_q^f + \Omega u \\ + \gamma \sum_{f \in F} \sum_{l \in L} \sum_{i \in I} \phi_{il}^f \end{aligned} \tag{28}$$

In conclusion, the aim of the optimization problem is to minimize (28) subject to the constraints (1) to (27).

## IV. Simulation Study Setup and Result Analysis

To analyze the optimization problem, we employed a fog computing architecture consisting of layer-1 fog, layer-2 fog, and cloud nodes, as depicted in Fig. 1. Our cooperative fog computing network utilized a portion of the cloud's computational resources, with the maximum number of servers to be used determined by the service provider. Since cloud computing can be accessed on demand, the amount of servers used varied depending on cost reduction criteria, but always remained below the maximum.

We assigned distinct values of CPU frequencies to the fog servers, varying the available computing capacity of each layer, based on the assumption that all fog servers in each layer had the same capacity. With the lowest capacity found in an area of layer-1 fog, the greater capacity in layer-2 fog, and the largest capacity in the cloud layer, the cost to reach a higher layer's

TABLE II
Requests Parameters for Homogeneous Demand Sets

| | | | | |
|---|---|---|---|---|
| **h** | 8 | 8 | 8 | 8 |
| **r** | 8 | 8 | 8 | 8 |

TABLE III
Requests Parameters for Heterogeneous Demand Sets

| | | | | |
|---|---|---|---|---|
| **h** | 8 | 5 | 4 | 2 |
| **r** | 8 | 5 | 4 | 2 |

TABLE IV
Topology Related Parameters

| | |
|---|---|
| **Number of Cluster Points** | 4 |
| **Number of Links** | 297 |
| **Number of Nodes** | 285 |
| **Number of Frequencies of Servers** | 8 |

TABLE V
Frequency of the Servers in Different Layers

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Layer-1** | 0.1 | 1.5 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 | 0.45 |
| **Layer-2** | 0.5 | 7.5 | 1.0 | 1.25 | 1.5 | 1.75 | 2.0 | 2.25 |
| **Cloud Layer** | 2.5 | 3.75 | 5.0 | 6.25 | 7.5 | 8.75 | 10.0 | 11.25 |

server increased with the layer. To solve the objective function, To solve the MILP program, we employed AMPL/CPLEX as our tool.

### A. Changes of Bandwidth Cost with $\alpha$:

The results of our analysis are presented in Fig. 2 and Fig. 3, which show the changes between bandwidth cost and $\alpha$ for homogeneous and heterogeneous demand sets, respectively. We found that when minimizing bandwidth cost is given lower priority (smaller value of $\alpha$), the bandwidth demand is spread across all potential destinations, i.e., layer-1 fog, layer-2 fog, and cloud, resulting in the highest cost for bandwidth. As $\alpha$ increases, a series of improvements are observed. Initially, the cloud layer is excluded, which is considered the costliest due to its distance from the cluster points. After a certain increase in $\alpha$, fog nodes on layer-2 are also excluded. Finally, demand is

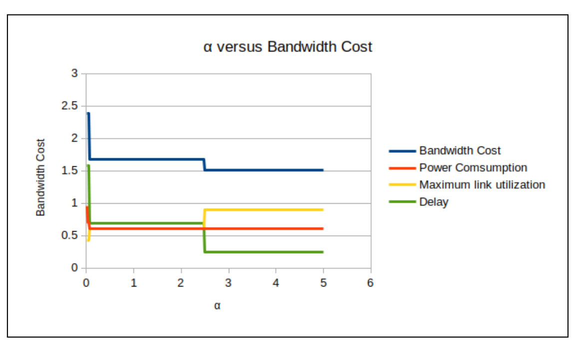


Fig. 2. Varying $\alpha$ for homogeneous demand set

met only through layer-1 fog, resulting in the lowest bandwidth cost.

As shown in Fig. 3, with smaller values of $\alpha$, the delay is high, but it decreases as $\alpha$ increases. This is because, with a low value of $\alpha$, the bandwidth demand is split to layer-1, layer-2, and cloud, causing data to transmit longer distances, resulting in high delay. The higher the value of $\alpha$, the less distance to cross and the lower the delay. We also observed that with lower values of $\alpha$, the power consumption is high, but it decreases as $\alpha$ increases. This is because with a low value of $\alpha$, the bandwidth demand is split to layer-1, layer-2, and cloud. As shown in Table V, the frequencies of the servers in the cloud are the highest, while the frequencies of layer-2 fog servers are lower, and layer-1 fog servers have the lowest frequencies. Therefore, with the increase of $\alpha$, power consumption is reduced. Furthermore, we found that with lower values of $\alpha$, the maximum link utilization is low, but it increases as $\alpha$ increases.

### *B. Changes of Power Consumption with $\mu$:*

The changes in power consumption with $\mu$ for the homogeneous demand set and heterogeneous demand set are illustrated in Fig. 4 and Fig. 5, respectively.

It was discovered that when power consumption is given lower priority, i.e., smaller value of $\mu$, resource demand is spread across all possible destinations, including layer-1 fog, layer-2 fog, and cloud. This results in the highest cost for power consumption. However, as $\mu$ increases, a series of improvements are observed. First, the cloud layer, which has the highest frequency according to Table V, is excluded. After a certain increase in $\mu$, fog nodes on layer-2 are also excluded. Finally, demand is met only through layer-1 fog. Meanwhile,

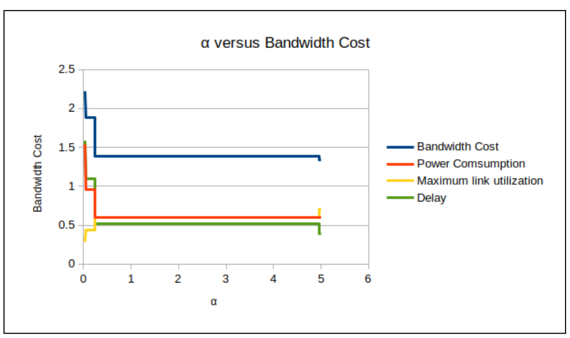


Fig. 3. Varying $\alpha$ for heterogeneous demand set

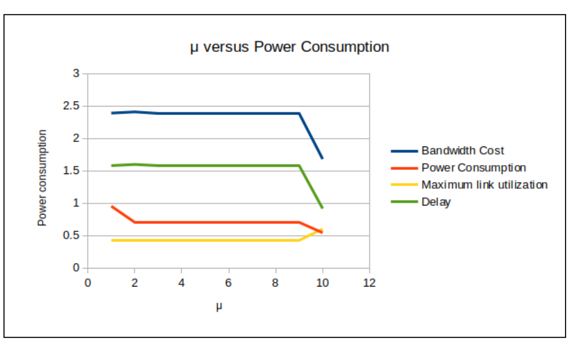


Fig. 4. Varying $\mu$ for homogeneous demand set

it was observed that with lower value of $\mu$, delay is high. However, with the increase of $\mu$, delay reduces. This is because, with a low value of $\mu$, resource demands are distributed to layer-1, layer-2, and cloud, causing data to transmit longer distances and resulting in higher delays. As $\mu$ increases, the data only travels through layer-1 fog, reducing the distance and therefore decreasing the delay.

Furthermore, it was observed that with lower value of $\mu$, bandwidth cost is high. However, with the increase of $\mu$, bandwidth cost reduces. This is because with a low value of $\mu$, resources are distributed to layer-1, layer-2, and cloud, resulting in a higher cost for bandwidth. As $\mu$ increases, resources are only used in layer-1 fog, reducing the cost. Finally, it was found that with lower value of $\mu$, maximum link utilization is low. However, with the increase of $\mu$, maximum link utilization increases.

### *C. Changes of Maximum Link Utilization with $\Omega$:*

The study analyzed maximum link utilization and bandwidth cost for homogeneous and heterogeneous demand sets (Fig. 6

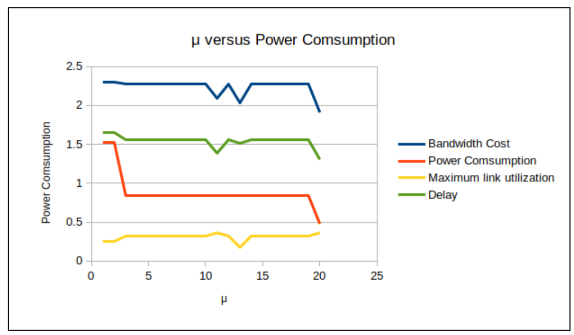


Fig. 5. Varying $\mu$ for heterogeneous demand set

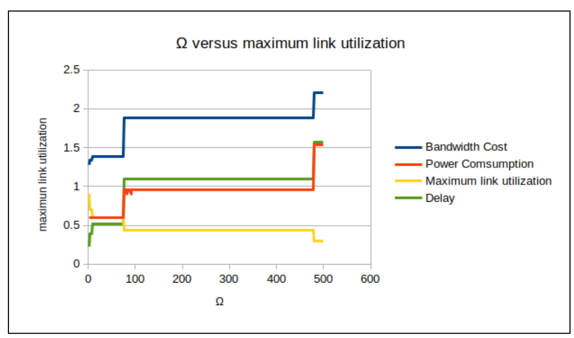


Fig. 6. Varying $\Omega$: for homogeneous demand set

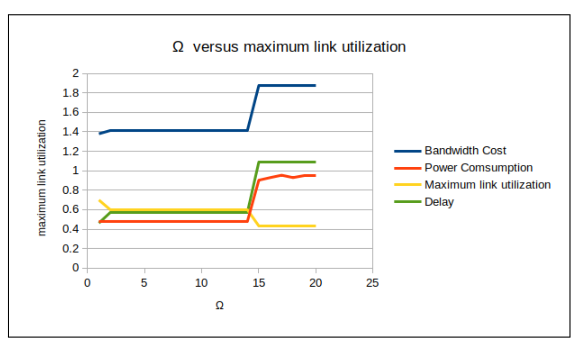


Fig. 7. Varying $\Omega$: for heterogeneous demand set

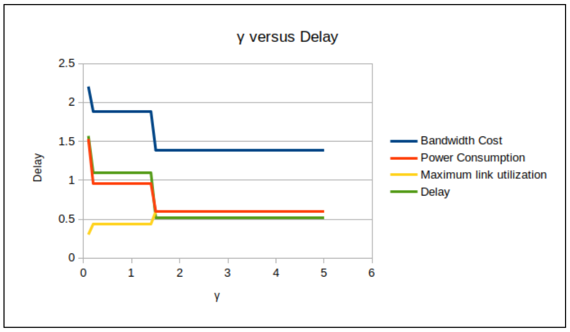


Fig. 8. Varying $\gamma$ for homogeneous demand set

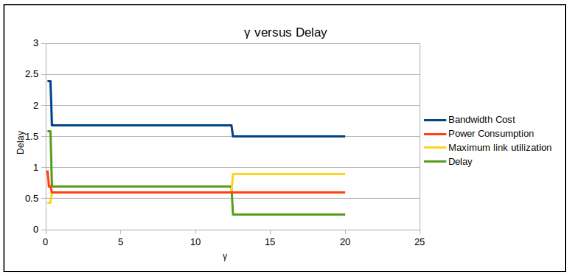


Fig. 9. Varying $\gamma$ for heterogeneous demand set

and Fig. 7). Results showed that lower values of $\Omega$ resulted in higher maximum link utilization, while higher values excluded cloud and layer-2 fog nodes. Lower $\Omega$ values also resulted in a lower delay, bandwidth cost, and power consumption. However, increasing $\Omega$ resulted in a higher delay, bandwidth cost, and maximum link utilization. Prioritizing maximum link utilization is important when optimizing fog computing resources. The findings can be used to design efficient resource allocation schemes for fog computing systems.

### D. Changes of Delay with $\gamma$:

The study conducted experiments on homogeneous and heterogeneous demand sets to observe the changes in delay, bandwidth cost, power consumption, and maximum link utilization with respect to the value of $\gamma$ (Fig. 8 and Fig. 9). The results showed that when delay minimization is given a lower priority (lower $\gamma$ value), requests are distributed among all possible destinations resulting in high delay and bandwidth costs. However, as $\gamma$ increases, requests are served only through layer-1 fog, reducing delay and bandwidth costs. Additionally, power consumption decreases as $\gamma$ increases due to fewer requests being directed to higher frequency cloud and layer-2 fog servers. Finally, maximum link utilization increases with an increase in $\gamma$.

## V. Conclusion and Future Work

The primary objective of this study was to minimize bandwidth cost, power consumption, and delay and achieve load balancing in a three-layer fog-cloud computing environment. To accomplish this goal, we proposed a novel MILP optimization formulation that takes into account all these factors. We evaluated the system's efficiency for both homogeneous and heterogeneous requests from cluster points in a network and analyzed the performance variation in terms of bandwidth cost, power consumption, delay, and maximum link utilization. The level of priority can be controlled by varying the weighting factors connected to the composite objective function. This approach provides service providers with the flexibility to prioritize different objective feature variables based on the network's needs.

In the future, we aim to investigate the effect of other parameters, such as processing time and storage capacity, and explore new techniques to further optimize the system's efficiency. Additionally, we want to provide a heuristic to evaluate our method in a dynamic traffic engineering setting.

## References


[1] Department, S. Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025(in billions). , https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] Jia, B., Hu, H., Zeng, Y., Xu, T. & Yang, Y. Double-matching resource allocation strategy in fog computing networks based on cost efficiency. *Journal Of Communications And Networks*. **20**, 237-246 (2018)

[3] Ghobaei-Arani, M. & Shahidinejad, A. A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. *Expert Systems With Applications*. **200** pp. 117012 (2022)

[4] Mushunuri, V., Kattepur, A., Rath, H. & Simha, A. Resource optimization in fog enabled IoT deployments. *2017 Second International Conference On Fog And Mobile Edge Computing (FMEC)*. pp. 6-13 (2017)

[5] Chen, Y., Sun, E. & Zhang, Y. Joint optimization of transmission and processing delay in fog computing access networks. *2017 9th International Conference On Advanced Infocomm Technology (ICAIT)*. pp. 155-158 (2017)

[6] Oueis, J., Strinati, E. & Barbarossa, S. The fog balancing: Load distribution for small cell cloud computing. *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. pp. 1-6 (2015)

[7] Maswood, M., Nasim, R., Kassler, A. & Medhi, D. Cost-efficient resource scheduling under qos constraints for geo-distributed data centers. *NOMS 2018-2018 IEEE/IFIP Network Operations And Management Symposium*. pp. 1-9 (2018)

[8] Maswood, M. & Medhi, D. Optimal connectivity to cloud data centers. *2017 IEEE 6th International Conference On Cloud Networking (CloudNet)*. pp. 1-6 (2017)

[9] Vakilian, S., Fanian, A., Falsafain, H. & Gulliver, T. Node cooperation for workload offloading in a fog computing network via multi-objective optimization. *Journal Of Network And Computer Applications*. **205** pp. 103428 (2022)

[10] Attiya, I., Abualigah, L., Elsadek, D., Chelloug, S. & Abd Elaziz, M. An intelligent chimp optimizer for scheduling of IoT application tasks in fog computing. *Mathematics*. **10**, 1100 (2022)

[11] Saif, F., Latip, R., Hanapi, Z. & Shafinah, K. Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing. *IEEE Access*. **11** pp. 20635-20646 (2023)

[12] Ramzanpoor, Y., Hosseini Shirvani, M. & Golsorkhtabaramiri, M. Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. *Complex & Intelligent Systems*. **8**, 361-392 (2022)

[13] Hussein, M. & Mousa, M. Efficient task offloading for IoT-based applications in fog computing using ant colony optimization. *IEEE Access*. **8** pp. 37191-37201 (2020)

[14] Sharmila, P. & Brahmananda, S. Levy Flight Adopted Particle Swarm Optimization-based Resource Allocation Strategy in Fog Computing. *Chinese Journal Of Electronics*. **33** pp. 1-11 (2023)

[15] Barzegaran, M., Cervin, A. & Pop, P. Performance optimization of control applications on fog computing platforms using scheduling and isolation. *IEEE Access*. **8** pp. 104085-104098 (2020)

[16] Maswood, M., Rahman, M., Alharbi, A. & Medhi, D. A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment. *IEEE Access*. **8** pp. 113737-113750 (2020)