Comparative Analysis of the Uplink Bandwidth Allocation in Wireless Sensor Networks: DRPA and FL-WFQ Case Study

Sravya Pallantla, and D. Haritha

***Abstract*—An intelligent utilization of resources in wireless sensor networks (WSNs) significantly impacts overall energy efficiency and communication performance in constrained sensor nodes. Weighted Fair Queuing (WFQ) is commonly used to allocate uplink bandwidth according to queue weights. Recently, Federated Learning (FL) has been combined with WFQ to dynamically predict optimal queue weights under varying traffic and mobility conditions. This paper presents a comparative analysis of two allocation approaches: (i) Data-Rate-Proportional Allocation (DRPA) and (ii) Federated Learning-enabled WFQ (FL-WFQ). Simulation results show that FL-WFQ improves efficiency by up to 119.22% (average 49.38%) and fairness by up to 54.79% compared to DRPA under dynamic mobility scenarios.**

***Index Terms*—Wireless sensor networks, edge computing, federated learning, weighted fair queuing.**

I. Introduction

The exponential expansion of Internet of Things (IoT) technology has brought forth a massive increase in data flow from interlinked devices such as sensors, actuators, and smart appliances. WSNs are the communication backbone of such IoT infrastructures and are typically composed of nodes with information sensing, processing, and transmission capabilities and limited power and resources [1]. With limited computational capability, battery power capacity, and buffer sizes, these nodes rely on resource sharing among edge nodes and end devices as a performance boost [2]. In modern network developments, *Edge Computing* has emerged as a promising approach to address these challenges, distributing computation directly to the data-handling location. Edge nodes process data locally and aggregate information rather than passing it to remote cloud servers [3], thus decreasing latency and bandwidth consumption leading to enhanced scalability. Despite these advantages, the efficient resource utilization of edge resources in the form of bandwidth, memory, and processing power is challenging to achieve due to time-varying network topologies, heterogeneous traffic loads, and unpredictable end device mobility [4], [5]. Learning-assisted scheduling and queue management techniques have also been explored in recent literature to dynamically adapt bandwidth allocation in edge-assisted IoT and sensor networks [6], [7].

Manuscript received March 31, 2026; revised April 28, 2026. Date of publication August 27, 2026. Date of current version August 27, 2026. The associate editor prof. Hideya So has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Jawaharlal Nehru Technological University Kakinada, India (e-mails: sravyapallantla@gmail.com, harithadasari9@yahoo.com).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0297

A. Resource Allocation Challenges in WSNs

In WSNs, where multiple devices share the uplink bandwidth to the edge server, a fair and efficient scheduling of resources is necessary. Queuing-based scheduling algorithms such as WFQ, Weighted Round Robin (WRR) and First-In-First-Out (FIFO) have been extensively studied for bandwidth allocation [8]. Of these, WFQ offers a robust mechanism by assigning weighted bandwidth shares to multiple flows, ensuring fairness and prioritization according to traffic characteristics. Traditional WFQ implementations are relatively static in approach; weights are preassigned and do not adapt to dynamic changes in data rates, link conditions, and node mobility [9]. As a result, the efficiency and fairness of these networks degrade significantly under non-stationary conditions.

To overcome this inadequacy, it has been reported in recent works that there are adaptive WFQ mechanisms using heuristic or optimization-based methods. For example, metaheuristic algorithms such as Ant Colony Optimization and Whale Optimization have been used for adaptive scheduling in wireless networks [4]. Although these algorithms increase throughput, they require centralized computation and frequent parameter updates, leading to high communication overhead, which is unfavorable for energy-limited WSNs. Therefore, decentralized and data-driven strategies have attracted increasing interest for dynamic edge resource allocation.

B. Federated Learning for Edge Resource Optimization

FL is a distributed architecture for machine learning that allows multiple edge nodes to train a model in collaboration without sharing raw data across a network [10]. Each edge node trains a local model using its locally generated data and shares only the model updates with an FL server for aggregation. This method protects data privacy, minimizes communication overhead, and enables learning with heterogeneous and non-independent data distributions [11].

The FL and WFQ techniques together, or FL-WFQ, enable predictive and adaptive allocation of resources in edge networks. Instead of employing static weights, the global FL model calculates optimal queue weights from node data rates, queue lengths, and channel conditions in the local network.

In previous work [12], FL-WFQ was shown to achieve high throughput, fairness, and energy efficiency for both static and mobility-aware WSN scenarios. However, although FL-WFQ provides strong adaptability, it also incurs additional computational cost and training overhead; hence, it may be worthwhile to compare FL-WFQ to simpler allocation methods. This work builds upon our previous study [12], where a FL-based resource allocation model was developed. In this paper, we extend it by performing a comparative analysis with DRPA.

C. Data-Rate-Proportional Allocation

More computationally straightforward is the *DRPA* approach, where bandwidth is assigned explicitly to devices based on their average data rates. This model supposes that devices that drive higher quantities of data traffic should be given larger shares of the bandwidth. Despite its simplicity and low cost to develop, DRPA does not necessarily possess the intelligence to adapt to dynamic conditions such as temporary congestion, packet loss, or mobility-induced link degradation [13]. Hence, its effectiveness is limited in ensuring fairness and network-wide efficiency in actual dynamic edge networks.

DRPA and FL-WFQ both attempt to control bandwidth between heterogeneous sensor nodes, though DRPA has a deterministic and static design while FL-WFQ uses adaptive and data-driven approaches. DRPA and FL-WFQ are selected due to their contrasting characteristics: DRPA provides a simple baseline, while FL-WFQ introduces adaptive intelligence through learning-based optimization. This paper focuses on comparing the performances of both these paradigms in various mobility, traffic, and channel conditions in a single simulation environment. We evaluate under which scenarios FL-WFQ tends to outperform DRPA and when the additional learning overhead of FL is worth the performance gains based on such factors as throughput efficiency, fairness, or packet delivery ratio. This work builds upon our previous study, extending it by performing a comparative analysis between DRPA and FL-WFQ under mobility-aware conditions.

The main contributions of this work include:

- Comparative analysis of DRPA and FL-WFQ in WSNs
- Evaluation under heterogeneous traffic and mobility scenarios
- Quantitative assessment of efficiency and fairness improvements

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system model and allocation schemes. Section IV discusses the simulation setup and results. Section V concludes the paper.

II. Related Work

A. Mobility Models in WSNs

The mobility of WSNs is of crucial importance for interlink stability, throughput and fair network bandwidth allocation [14], [15]. Conventional, static allocation models fail when nodes have uncertain movement patterns, generating link failures and variable channel conditions. In the literature, diverse mobility models have been reported to simulate real-world motions in simulation settings [14], [15]. We employ the *Random Walk 2D Model* so that every node moves in a randomly selected direction always at the same speed for a specified amount of time and then selects another approach. The *Random Direction Model*, on the other hand, guarantees a better spatial distribution by stipulating that nodes must move toward a specified line before entering the next direction. According to the *Random Waypoint Model* of MANET research, every node can rest randomly and then move to a new spot at a random speed [16]. The *Gauss–Markov Model* correlates the evolution of the velocity and direction updates and its fidelity in representing the dynamics of mobile or vehicular trajectories [17]. In the context of edge-assisted WSNs, these mobility patterns determine the packet arrival rate and queue dynamics at the edge node. Hence, mobility-aware dynamic resource allocation algorithms like FL-enabled WFQ are essential for ensuring fairness and efficiency [12].

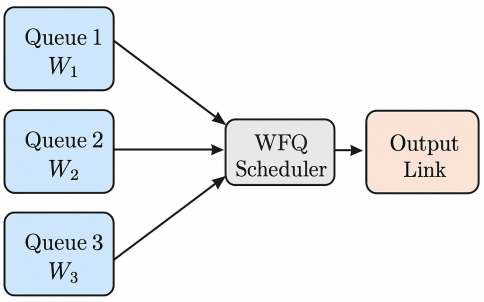

Fig. 1. WFQ mechanism ensures bandwidth distribution proportional to queue weights.

B. Weighted Fair Queuing

WFQ is a packet scheduling algorithm that ensures that a fair sharing of bandwidth is maintained among multiple traffic flows with different service priorities. In general, it uses an approximation of the ideal Generalized Processor Sharing (GPS) model, where the associated flow i is given a weight W_i equal to the share of total link capacity it obtains [18]. The WFQ scheduler operates a separate queue for every flow and sends packets in a weighted round-robin fashion, to make sure that each flow occupies a fair share of the bandwidth with regard to its weight, as shown in Fig. 1.

In flow i, let L_i^k denote the k-th packet length, and $V(t)$ represent the system's virtual time. The start and finish times of a packet are defined as:

$$S_i^k = \max(F_i^{k-1}, V(a_i^k)), \quad F_i^k = S_i^k + \frac{L_i^k}{W_i} \tag{1}$$

where a_i^k is the packet arrival time. The packet with the smallest finish time F_i^k is transmitted first. WFQ thus provides proportional service fairness while minimizing queuing delays [8].

Having these advantages, classical WFQ implementations are static; the weights are assigned as is, and do not adjust to dynamic channel variations or traffic heterogeneities; therefore, they do not account for performance changes at different

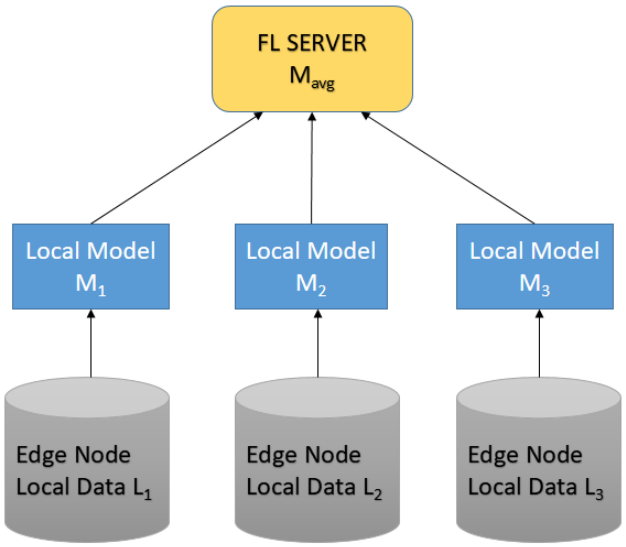

Fig. 2. FL-WFQ model for adaptive resource allocation. Edge nodes collaboratively train local models and share updates with a central FL server that aggregates them into a global model.

time points. This limitation prompts the application of machine learning and federated intelligence for responsive scheduling.

C. Federated Learning-based WFQ

FL has recently gained significant attention for distributed resource optimization in WSNs. Lightweight FL frameworks have been proposed to reduce communication overhead and preserve data privacy in resource-constrained IoT systems [19]. Adaptive FL approaches further enhance scalability and robustness in heterogeneous environments [20]. FL improves the WFQ in that edge nodes learn on their own distributed data without centralizing the information to give them optimal weights. At the edge nodes, local traffic data is captured and trained to train a model that produces queue weights (in the present instance W_i), computed from node characteristics (data rate (S_i), queue occupancy (Q_i), and historical throughput (here, defined as T_i)) [10], [11]. The global FL model is calculated at the central server with the Federated Averaging (FedAvg) model as follows:

$$M_{\text{global}} = \sum_{i=1}^{N} \frac{|D_i|}{\sum_{j=1}^{N} |D_j|} M_i \tag{2}$$

The resulting aggregated model is then redistributed per edge node to execute an update in its local WFQ scheduling parameters. FL-WFQ is iteratively implemented across communication rounds, adapting to traffic variations and mobility. The overall FL-WFQ architecture, showing distributed edge nodes and a central FL server exchanging model updates, is illustrated in Fig. 2.

In contrast with centralized learning, FL-WFQ minimizes communication overhead by only transmitting model parameters rather than raw sensor data. Moreover, it guarantees data privacy and scalability for vast, distributed WSN setups [5]. In practice, FL-WFQ achieves better fairness and throughput than both traditional WFQ and WRR in mobile edge environments.

D. Performance Metrics

The measures to evaluate resource allocation mechanisms in WSNs are:

Fairness represents the degree to which resource distribution is even between a number of users, tasks, or nodes of the system. At the edge, especially in FL scenarios, being fair stops any one node or user from overpowering available resources and this could result in service degradation for others or starvation of resources for others. If we consider a system with M flows, then $X = (x_1, x_2, \ldots, x_M)$ is the allocation vector and x_i is the fraction of resources allotted to the i^{th} flow in this system. The fairness of this allocation, defined as $F(X)$, can be calculated as follows [21]:

$$F(X) = 1 - \frac{2\sigma_x}{H - L} \tag{3}$$

Here H and L denote the maximum and minimum throughput values observed among all flows, while σ_x is the standard deviation of throughput values resulting from the allocation X. This metric provides a normalized measure of fairness, where higher values indicate more equitable distribution of resources.

Efficiency, in contrast, evaluates the degree to which allocated resources are effectively utilized. For each flow i, let I_i be the input data rate and T_i be the achieved throughput. The overall efficiency of the allocation X is given by:

$$E(X) = \frac{1}{M} \sum_{i=1}^{M} \frac{T_i}{I_i} \times 100 \tag{4}$$

These metrics together offer a comprehensive evaluation framework for comparing the effectiveness of DRPA and FL-WFQ under diverse network scenarios.

E. Traditional Scheduling Strategies

Prioritized scheduling and fixed priority algorithms such as FIFO and WRR were introduced for the early WSN scheduling on time and location data types. FIFO is straightforward but not justifiable in case of heterogeneous data rates [22]. WRR was designed to extend this by giving fixed priority values in addition to providing variable service among the sensor devices [23]. But static weighting makes inefficient use of bandwidth when node traffic fluctuates on the fly.

WFQ, which approximates the true Generalized Processor Sharing model [18], was introduced to provide a more accurate method. WFQ ensures a bounded delay and bandwidth distribution where each node is assigned a corresponding portion of the bandwidth. While it is fairly designed, WFQ assumes preconfigured weights, which cannot be applied to adaptive or mobile environments. Several works that extend a model of WFQ with dynamic weighting mechanisms however tend either to rely on centralized data or to be prohibitively expensive to compute [8].

F. Data-Rate-Based and Adaptive Allocation Schemes

To simplify the scheduling of low-power WSNs, proportional allocation methods that use node charge data rates can be applied. In such methods bandwidth or timeslots are

shared with a proportion to the nodes' own average data rate to guarantee throughput proportionality [24]. Despite performance on static networks, such methods are sensitive to traffic heterogeneity and unfair under congestion.

Approaches for adaptive usage of resource allocation brought queue-length and delay-based weighting considerations [25], which made responsive response to the traffic situations in part the result of partial load balancing, using these models. However, such algorithms rely on large centralized monitoring and are unable to scale up as the number of nodes increases.

G. Machine Learning and Federated Learning of WSNs

Machine learning has been widely used to model traffic patterns and predict optimal resource allocation decisions in WSNs and IoT networks [26]. Recent advances in deep learning and federated learning have further improved bandwidth allocation, scalability, and adaptive scheduling in dynamic wireless environments [7], [19], [20]. Other methods, including regression, reinforcement learning, and neural networks have been developed to obtain dynamic results for the bandwidth or energy distribution variables. However, traditional ML algorithms need to be trained in one place, making it impractical in privacy-sensitive or data-limited edge environments. Recent studies highlight the growing importance of FL for resource allocation in WSNs, enabling privacy-preserving distributed optimization and improved scalability under heterogeneous network conditions [7], [27].

FL proposed by McMahan *et al.* [10], overcomes this limitation by enabling shared model training without sharing raw data. FL is a class of ML that aggregates local model updates from edge nodes to obtain a global model with data privacy and scaling. Recent works had used FL to optimize wireless resource allocation, which included spectrum sharing [28] and power control [11]. FL has been proposed for adaptation of bandwidth and optimization of energy in WSN contexts [5]. Recent studies have further demonstrated that federated edge intelligence enables scalable and privacy-preserving resource management while adapting to heterogeneous traffic and mobility conditions [7], [29]. Recent research extends FL-based optimization toward mobility-aware and real-time scheduling scenarios, addressing the limitations of static datasets and centralized learning architectures in dynamic WSN environments [29], [30].

H. Mobility-Aware Resource Allocation

Additionally, we have mobility issues in WSN scheduling since node connectivity and link quality fluctuate over time. Proposed to mitigate packet loss and optimize the fairness of the WFQ according to node mobility, mobility-aware WFQ variants and dynamic queue adjustment mechanisms have been introduced [4]. Through continuous updating of the global model in the light of real-time mobility observations at the edge nodes, FL improves the adaptability to the situation [12]. Recent mobility-aware learning models further confirm that incorporating node movement characteristics into learning-based schedulers improves fairness and stability in highly dynamic WSNs [30]. Such achievements lead to distributed intelligence at network edge, supporting smart city and vehicular IoT use cases.

I. Research Gap

Literature review shows that:

1) Classical WFQ does achieve fairness but is less adaptive to dynamic requirements.
2) DRPA provides simplicity yet doesn't meet traffic conditions.
3) FL-based scheduling adds flexibility and privacy but needs further validating against simpler proportional scheduling methods.

Therefore, this work aims to take up the gap and compare these *DRPA* techniques to *FL-enabled WFQ* for exploring trade-offs between computational efficiency, fairness, and adaptability for dynamic edge-assisted WSNs. While prior work focused on FL-based optimization, a direct comparison between DRPA and FL-WFQ under mobility-aware scenarios remains limited, which is addressed in this study.

III. Methodology

A. System Model

The system model is an example of N wireless sensor nodes $n_1, n_2, \ldots, n_N$ communicating with a single edge node through IEEE 802.11-based wireless links. Each node periodically sends packets, each of which has:

- Average data rate: S_i, in bits per second.
- Packet length: L_i, in bits.
- Sleep interval: $T_{sleep,i}$ (in seconds).

The overall simulation duration is T_{sim}, where each node sends several packets:

$$N_{pkts,i} = \frac{T_{sim}}{T_{sleep,i}} \tag{5}$$

and, consequently, the sum of the data sent by node i is given by:

$$D_i = N_{pkts,i} \times L_i \tag{6}$$

Hence the average data rate of every node is:

$$S_i = \frac{D_i}{T_{sim}} \tag{7}$$

The total available uplink bandwidth at the edge node is denoted by B_{UL}, and is shared among all sensor nodes as:

$$\begin{gathered} \mathbf{A} = [A_1, A_2, \ldots, A_N], \\ \text{such that } \sum_{i=1}^{N} A_i \leq B_{UL} \end{gathered} \tag{8}$$

The aim of both DRPA and FL-enabled WFQ to maximize overall network performance and fairness is to determine the proper allocation of A_i values.

Algorithm 1 Data-Rate-Proportional Allocation (DRPA)

Input: average data rates S_i for nodes $i = 1, 2, \ldots, N$, total uplink bandwidth B_{UL}
Output: bandwidth allocation A_i for each node
Compute cumulative data rate:
$S_{\text{total}} \leftarrow \sum_{j=1}^{N} S_j$
for $i = 1$ to N **do**
    $W_i \leftarrow \frac{S_i}{S_{\text{total}}}$
    $A_i \leftarrow W_i \times B_{UL}$
end for
Return allocation vector $A = [A_1, A_2, \ldots, A_N]$

B. Data-Rate-Proportional Allocation

Within the DRPA scheme a node's available bandwidth is directly proportional to the average data generation rate S_i. Underlying the philosophy it is assumed that nodes that generate data traffic are given corresponding bandwidth if they send in higher data. An allocation weight W_i and an assigned bandwidth A_i are given by:

$$W_i = \frac{S_i}{\sum_{j=1}^{N} S_j}, \quad A_i = W_i \times B_{UL} \tag{9}$$

In this paper, W_i is the normalized value for the contribution of each node to the overall network data rate. Its simplicity allows low cost computation, easy implementation and adaptation to low-power WSNs in static traffic, as well as reduced number of nodes needed. But with dynamic or mobile deployments, the deterministic nature of DRPA becomes a constraint, causing channel conditions and node load to fluctuate very rapidly. The model doesn't account for queue build-ups or congestion at the edge node, leading to bandwidth underutilization or inequitable distribution among nodes. The pseudocode for DRPA is presented in Algorithm 1.

The computational complexity of DRPA is O(N), as it requires a single pass over all nodes for proportional allocation.

C. Federated Learning-Enabled WFQ

FL-WFQ is a variant of the current state-of-the-art model. It extends classical WFQ by dynamically predicting queue weights (W_i), using a FL framework. Each edge node n_i maintains a local dataset:

$$D_i = (S_i, Q_i, T_i, A_i) \tag{10}$$

where Q_i is queue length, and T_i is observed throughput. Each node trains a local model M_i to predict a queue weight based on these features. Instead of making raw data centralized, nodes would share only their model parameters with a central FL server. The FL server combines the original local models to fit a global model M_{avg} using the Federated Averaging (FedAvg) algorithm:

$$M_{avg} = \sum_{i=1}^{N} \frac{|D_i|}{\sum_{j=1}^{N} |D_j|} M_i \tag{11}$$

We then redistribute the global model; all edge nodes update their respective local WFQ scheduling weights:

$$W_i^{(t+1)} = f_{M_{avg}}(S_i, Q_i, T_i) \tag{12}$$

Algorithm 2 Federated Learning-Based Weighted Fair Queuing (FL-WFQ)

Input: Local datasets D_i, initial global model M_{avg}^0
Output: Adaptive queue weights W_i
for each communication round $t = 1, 2, \ldots, T$ **do**
    for all nodes n_i in parallel **do**
        Train local model M_i^t using dataset D_i
        Send model updates ΔM_i^t to the FL server
    end for
    Aggregate local updates at FL server:
    $M_{\text{avg}}^{t+1} \leftarrow \sum_{i=1}^{N} \frac{|D_i|}{\sum_{j=1}^{N} |D_j|} M_i^t$
    Broadcast updated global model M_{avg}^{t+1} to all nodes
    Update local queue weights:
    $W_i^{t+1} \leftarrow f_{M_{\text{avg}}^{t+1}}(S_i, Q_i, T_i)$
end for
Return: Adaptive queue weights W_i

where $f_{M_{avg}}$ is the learned function for weight prediction at round $t + 1$. In combination with the previous mechanism of algorithm, the iterative training, aggregation, and post-progression process guarantees the adaptation of the algorithm to mobility and traffic changes, leading to a balance between fairness and efficiency. We summarize the workflow in Algorithm 2. The FL-WFQ approach has higher computational complexity due to iterative training and aggregation, approximately $O(N \times R \times E)$, where R is the number of FL rounds and E is the number of training epochs.

As illustrated in Fig.3, this workflow comparison emphasizes their systemic and operational distinctions between the DRPA and FL-enabled WFQ algorithms. 3 guarantees the computational lightweightness of DRPA with less flexibility for network dynamics such as queue build-up and node mobility. In contrast, the FL-WFQ workflow (right side of Fig. 3) is largely based on distributed intelligence through FL. An edge node in this approach retrieves parameters such as data rate (S_i), queue size (Q_i), and throughput (T_i) in order to train a local model that can realistically predict a queue's weights in real-time. These local models are periodically aggregated by a centralized FL server to form the global model that produces dynamic weight predictions. This feedback loop of repeated learning and iteration can dynamically adjust bandwidth allocation for different mobility and traffic conditions—making FL-WFQ a fairer solution that achieves higher throughput than DRPA.

Overall, Fig. 3 highlights that DRPA is based on a fixed rule-based path, whereas FL-WFQ shows a closed-loop adaptive system capable of intelligent decision-making through federated model updates.

D. Comparative Analysis

The two allocation schemes can be contrasted as follows:

- DRPA: Lightweight, static, ideal for low-mobility and low-traffic variance environments.
- FL-WFQ: Adaptive, learning-based, applicable to heterogeneous and mobile WSNs with varying loads.

While DRPA relies solely on proportional data rate distribution, FL-WFQ utilizes a data-driven learning approach to balance fairness, efficiency, and adaptability simultaneously,

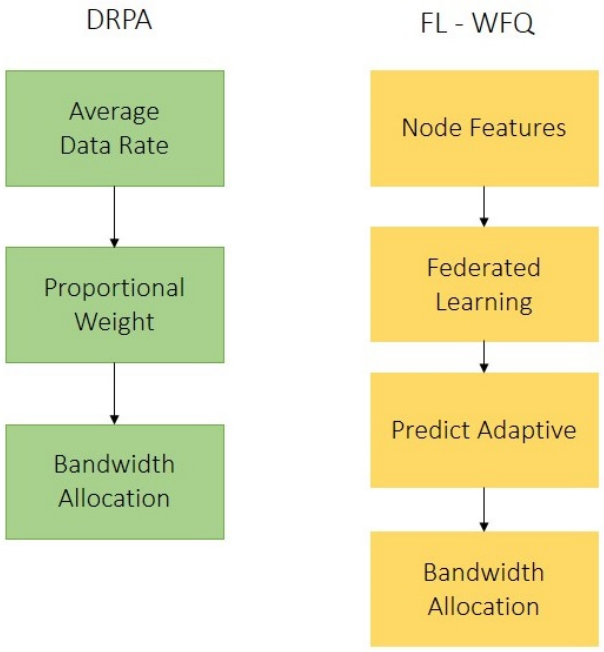

Fig. 3. Workflow comparison between DRPA and FL-WFQ. DRPA uses static proportional weights, whereas FL-WFQ employs federated learning to predict adaptive queue weights under dynamic network conditions.

which is consistent with recent studies on federated edge intelligence and adaptive resource management in wireless IoT networks [29]. The improved performance of FL-WFQ is attributed to its adaptive learning capability, which dynamically adjusts queue weights based on real-time traffic and mobility conditions. The improved performance of FL-WFQ is mainly due to its ability to learn from changing network conditions and adjust queue weights dynamically. In contrast, DRPA follows a fixed proportional allocation strategy that cannot respond to sudden changes in traffic or node mobility. While FL-WFQ introduces additional computational and communication overhead because of local model training and federated aggregation, it provides better adaptability in dynamic environments. Therefore, DRPA is more suitable for lightweight and stable network deployments, whereas FL-WFQ is a better choice for mobility-aware wireless sensor networks where maintaining fairness and efficient bandwidth utilization is essential.

IV. Implementation and Results

A. Simulation Setup

Simulations were conducted using the NS-3 network simulator [31]. The testbed consisted of eight wireless sensor nodes and one edge node configured with IEEE 802.11 (5 GHz) links as shown in Fig.4. Table I summarizes the simulation parameters used. Each end device generates data traffic as shown in Table II. The NS-3 simulator is used to model the wireless sensor network environment, including node deployment, mobility, traffic generation, and communication behavior. The simulation incorporates modules for wireless communication, mobility modeling, and packet transmission to emulate realistic network conditions. Input parameters such as node density, mobility speed, traffic rates, and link capacity are configured to generate diverse scenarios. The simulator outputs performance metrics including throughput, delay, efficiency, and fairness, which are used to evaluate the DRPA and FL-WFQ approaches. NS-3 modules for wireless communication, mobility modeling, and traffic generation were used to simulate realistic WSN scenarios.

TABLE I
Simulation Parameters

Parameter	Value
Simulation Time	1000 s
Wireless Standard	IEEE 802.11 (5 GHz)
Modulation	16-QAM
Transmit Power	16 dBm
Propagation Model	Log-Distance
Antenna Gain	2 dBi
Mobility Models	Gauss–Markov, Random Direction 2D, Random Walk 2D, Random Waypoint

Fig. 5 illustrates the workflow of the NS-3 simulation framework, where input parameters are used to generate network scenarios, followed by data collection, model training, and performance evaluation.

The simulation setup is consistent with our prior work [12], where parameters such as grid size (150–300 m), mobility speed (2–5 m/s), and link capacity (1–5 kbps) are varied to generate diverse network scenarios.

The FL implementation was developed using NS-3.14, Python 3.10, TensorFlow 2.15, Keras 2.15, Flower 1.8, Scikit-learn 1.5, NumPy 1.26, and Pandas 2.2. The NS-3 simulator was used for network scenario generation, while Python-based machine learning frameworks were employed for model training and evaluation.

B. Dataset and Training

For each node, the average data rate S_i was computed as:

$$S_i = \frac{N_{pkts} \times L_i}{T_{sim}} \tag{13}$$

where N_{pkts} is the number of packets sent during the simulation, L_i is the packet length (in bits), and T_{sim} is the total simulation duration.

The dataset generation process follows our prior work [12], where ns-3 simulations are used to generate input-output tuples based on parameters such as queue sizes, weights, grid size, mobility speed, and link capacity. Each network scenario produces tuples of the form (Wi, Qi, G, M, L), where the corresponding efficiency or fairness is recorded as output. These samples are collected periodically during simulation and used for training the FL model.

The FL framework was implemented using the Flower (FLWR) library with three participating clients. The dataset contained 1424 samples, each represented by 19 input features and one target output. The data were partitioned among the clients for local training. Before training, selected features were normalized and subsequently standardized using the

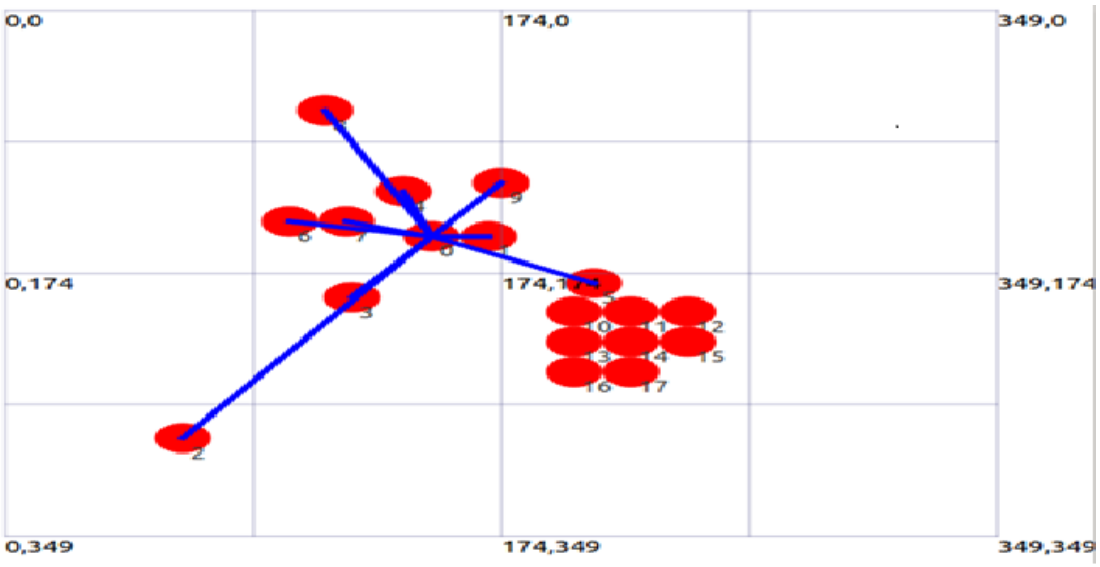

Fig. 4. System model showing sensor nodes communicating with an edge node using IEEE 802.11 links, where uplink bandwidth B_{UL} is distributed among N devices.

TABLE II
END DEVICE TRAFFIC PARAMETERS

Type of Device	Average Sleep Time (s)	Mean Data Rate (bps)	Packet Length (bits)	Mobility
Smoke Sensor	4	462	234	Yes
Motion Sensor	4	11388	234	Yes
Amazon Echo Hub	4	462	144	No
Smart Phone	4	2461	327	Yes
Smart Plug	241	462	144	No
Smart Bulb	4	462	94	No
Smart Camera	4	2461	234	Yes
Wireless Speaker	4	462	144	Yes

StandardScaler technique to ensure stable convergence. The neural network was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 30.

Several neural network configurations were evaluated during model development. The search space included architectures with 2–6 hidden layers and neuron sizes ranging from 32 to 256. The final architecture (64–128–256–128–64) was selected empirically because it provided the best trade-off between prediction accuracy, convergence stability, and computational cost. Simpler architectures resulted in underfitting, while deeper architectures showed negligible performance gains and increased training overhead.

To reduce overfitting, Batch Normalization and Dropout layers (dropout rate = 0.4) were incorporated into the network. In addition, Early Stopping was employed with a patience value of 25 epochs, restoring the best model weights when no further validation improvement was observed. A custom stopping criterion was also used to terminate training when the validation loss dropped below 0.01.

The training hyperparameters were selected through empirical experimentation. Multiple combinations of learning rates, batch sizes, and network configurations were evaluated, and the final settings were chosen based on validation performance and convergence behavior. No automated hyperparameter optimization technique was applied in the current study.

The global model was computed using the Federated Averaging (FedAvg) algorithm, which aggregated weighted updates from all participating nodes. The aggregated model was then distributed back to the edge nodes to predict dynamic WFQ weights for real-time bandwidth scheduling and resource allocation. The FL model is trained iteratively across multiple communication rounds, where edge nodes train local models and share updates with the central server for aggregation using the FedAvg algorithm. The FL process was executed for 15 communication rounds across three participating clients.

C. Results and Discussion

Fig. 6 demonstrates that FL-WFQ shows better efficiency than DRPA among all mobility models, up to 15% higher. This improvement is due to the adaptive weight prediction capability of FL-WFQ, which dynamically adjusts bandwidth allocation based on real-time queue conditions and traffic variations, unlike the static DRPA approach. Thanks to its adaptive weight prediction approach that adjusts bandwidth according to its demand (traffic and queue) during a given operation cycle. Similarly, Fig. 7 presents a fairness comparison for the 3 schemes. Although FIFO offers limited fairness and DRPA takes a hit at high-mobility situations, FL-WFQ allows for almost the same allocation of resources among nodes in that Jain's index is close to unity. The FL integration allows to

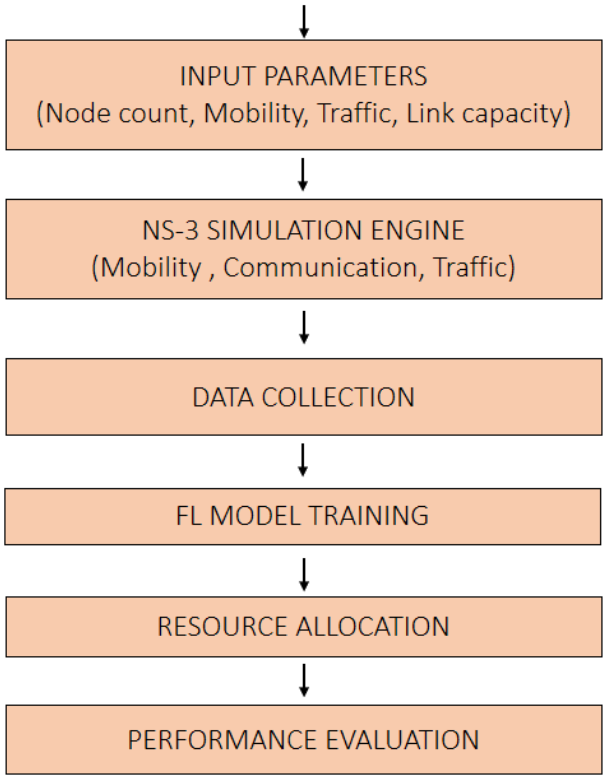

Fig. 5. Block diagram of NS-3 simulation framework used for resource allocation analysis

update the model in real time based on different data rates and queue structures for equitable scheduling.

Table III provides an overview of the network efficiency and fairness performance of FL-WFQ and DRPA in different mobility models. Note that FL-WFQ surpasses DRPA in all mobility scenarios, which is attributed to its adaptive weight prediction and decentralized instruction design, in that it employs a decentralized learning mechanism. The results clearly indicate that FL-WFQ maintains higher fairness and efficiency across all mobility models due to its learning-based adaptive mechanism.

This is similar to the efficiency improvements, which can range from 7.31% under the Random Walk 2D model to a maximum of 119.22% under the Random Direction 2D model, having a mean gain of 49.38%. This remarkable improvement illustrates the flexibility of FL-WFQ in dynamically distributing bandwidth according to the shifting rates of data and link environments, as well as the static proportional allocation used by DRPA. The fairness index improves by around 38.62%; the highest improvement (54.79%) is recorded when the model moves in Random Direction 2D. This shows that FL-WFQ maintains a balanced bandwidth distribution in the sensor nodes, even in the case of changing mobilities and different traffic patterns. The outcomes clearly show that merging FL with the WFQ architecture supports more effective utilization of throughput and fairness. The flexibility of FL-WFQ is advantageous in mobile WSNs under dynamic network topology and data traffic. Overall, FL-WFQ shows strong performance improvement in throughput capacity and fairness, confirming the use of FL in WFQ for mobility-aware edge networks.

FL-WFQ consistently performs better across all mobility models because it can adapt to changing network conditions during operation. As sensor nodes move, factors such as link quality, traffic load, and queue occupancy change continuously. Unlike DRPA, which allocates bandwidth based only on the average data rate, FL-WFQ learns from these changing conditions through the federated learning process and updates the queue weights accordingly. This allows the scheduler to allocate bandwidth more effectively, resulting in better resource utilization and a fairer distribution among sensor nodes. The benefits become more evident in highly dynamic mobility models, where frequent changes in network topology make static allocation methods less effective.

Although FL-WFQ achieves better efficiency and fairness, DRPA still has practical advantages in certain scenarios. The DRPA algorithm is simple to implement and requires very little computation because bandwidth allocation is based only on the data generation rate of each node. It does not require model training, parameter updates, or communication between participating nodes, making it suitable for small-scale or relatively static wireless sensor networks where traffic conditions remain stable. In such deployments, the lower computational and communication overhead of DRPA may be more beneficial than the additional performance improvements offered by FL-WFQ. However, as network mobility and traffic variability increase, the adaptive learning capability of FL-WFQ becomes increasingly valuable.

D. Reproducibility

All simulation parameters, mobility models, and training configurations are clearly defined to ensure reproducibility of the results.

E. Simulation Workflow

The workflow includes scenario generation, dataset collection, federated model training, aggregation, and performance evaluation under different mobility conditions. For example, a typical simulation begins with initializing sensor nodes and mobility models, followed by traffic generation based on device characteristics. Local models are then trained at edge nodes and aggregated at the central server. Finally, the updated model predicts queue weights, which are used for bandwidth allocation and performance evaluation.

V. Conclusion

The results demonstrate that FL-WFQ significantly outperforms DRPA, achieving up to 119.22% improvement in efficiency and 54.79% improvement in fairness. However, the approach introduces additional computational overhead, which will be addressed in future work through lightweight and energy-aware optimization techniques. Despite its performance advantages, the proposed FL-WFQ framework introduces additional computational and communication overhead due to local model training and federated aggregation. These requirements may affect deployment in highly resource-constrained

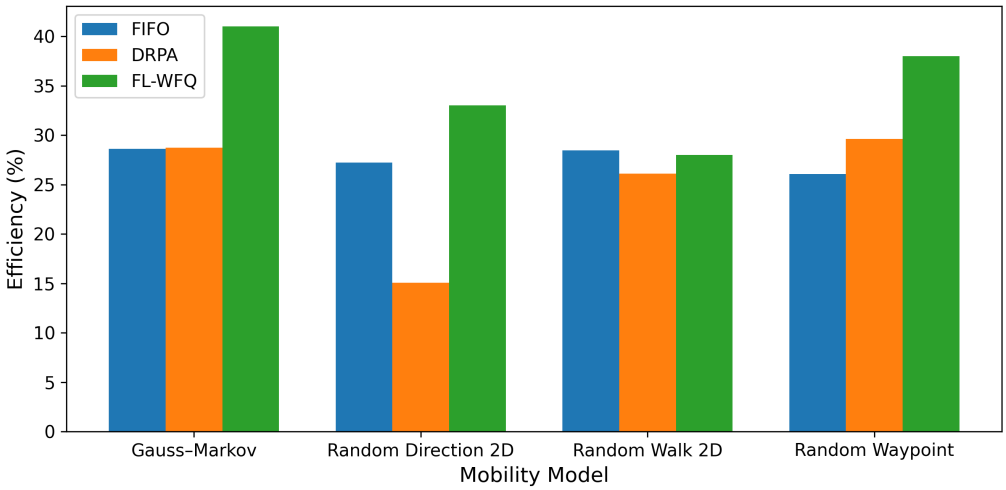

Fig. 6. Comparison of Efficiency between FIFO, DRPA, and FL-WFQ with different mobility models. FL-WFQ has the highest overall utilization, particularly on Gauss–Markov and Random Waypoint models.

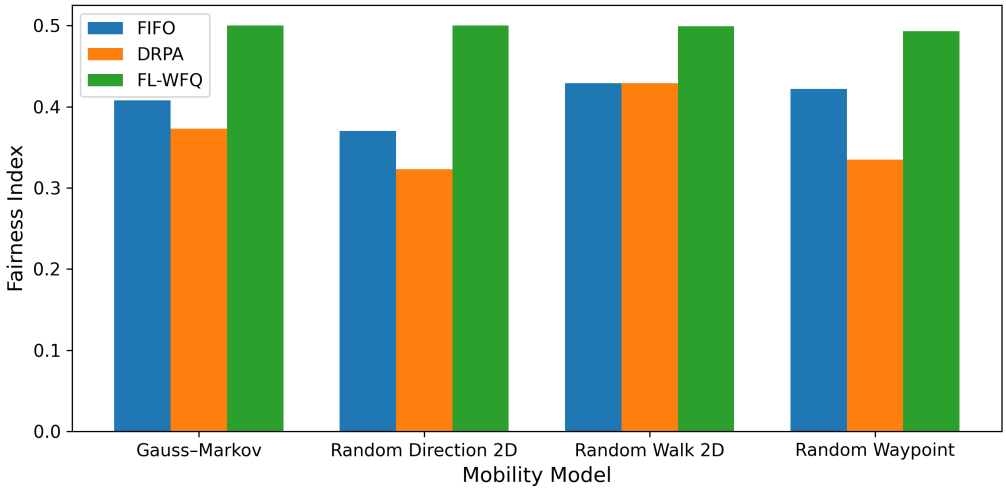

Fig. 7. Fairness comparison between mobility models. All the time, FL-WFQ presents a higher fairness index than FIFO and DRPA, evidence for adaptive balancing of queues.

sensor nodes. Lightweight neural architectures, model pruning, and edge-assisted training mechanisms may help reduce these costs in future implementations. Although FL-WFQ requires additional computation for model training and aggregation, the performance gains in terms of efficiency and fairness justify this overhead in dynamic wireless sensor networks. On the other hand, DRPA remains a suitable alternative for applications with limited computational resources or relatively stable traffic conditions, where a simple and lightweight allocation strategy is sufficient. Future work will explore energy-efficient FL models and scalability improvements for large-scale WSN deployments.

REFERENCES

[1] S. Chilukuri and D. Pesch. Achieving optimal cache utility in constrained wireless networks through federated learning. In *Proceedings of the IEEE 21st International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 254–263. IEEE, 2020.

[2] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian. Resource management approaches in fog computing: A comprehensive review. *Journal of Grid Computing*, 18(1):1–42, 2020.

[3] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Transactions on Wireless Communications*, 16(8):4924–4938, 2017.

[4] Q.V. Pham, T.H. Nguyen, and W.J. Hwang. Woa-based dynamic scheduling for mobile wireless sensor networks. *Sensors*, 20(19):5486, 2020.

[5] Hao Yu, Sen Yang, and Xin Jin. Resource management in federated learning: A survey. *IEEE Internet of Things Journal*, 8(5):3516–3531, 2021.

[6] M. Alam and R. Hussain. Adaptive scheduling in edge-assisted wireless sensor networks. *IEEE Systems Journal*, 19(1):233–244, 2025.

[7] X. Li and M. Chen. Edge intelligence with federated learning for iot systems. *Scientific Reports*, 2024.

[8] Gang Chang, Adam Varga, and Sándor Molnár. Design and implementation of queue management and scheduling in ns-3. In *Proceedings of the 2015 Workshop on ns-3*, pages 67–74, 2015.

[9] D. Militani, S. Vieira, E. V. ao, K. Neles, R. Rosa, and D. Z. Rodríguez. A machine learning model for resource allocation service for access point on wireless network. In *Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks*

TABLE III
COMPARISON OF FL-WFQ AND DRPA ACROSS DIFFERENT MOBILITY MODELS

Mobility Model	DRPA Eff.	FL-WFQ Eff.	Efficiency Gain (%)	DRPA Fair.	FL-WFQ Fair.	Fairness Gain (%)
Gauss–Markov	28.74	41.00	42.65	0.373	0.500	34.00
Random Direction 2D	15.05	33.00	**119.22**	0.323	0.500	**54.79**
Random Walk 2D	26.10	28.00	7.31	0.429	0.499	16.37
Random Waypoint	29.61	38.00	28.33	0.335	0.493	47.32

(SoftCOM), pages 1–6, 2019.

[10] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[11] Q. Li, Z. Wen, and B. He. Federated learning-based resource allocation for wireless edge computing networks. *IEEE Internet of Things Journal*, 7(12):11887–11897, 2020.

[12] Sravya Pallantla, D. Haritha, and Shanti Chilukuri. Mobility-aware resource allocation in edge networks for smart neighborhoods. *SSRG International Journal of Electronics and Communication Engineering*, 12(8):188–198, 2025.

[13] M. Eisen, S. Shukla, D. Cavalcanti, and A. S. Baxi. Communication-control co-design in wireless edge industrial systems. In *Proceedings of the IEEE 18th International Conference on Factory Communication Systems (WFCS)*, pages 1–8. IEEE, 2022.

[14] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[15] S. Temene, K. Djouani, A. Kurien, and A. Hamam. Mobility-aware resource allocation in mobile wireless sensor networks: A survey. *Sensors*, 19(13):1–29, 2019.

[16] C. Bettstetter. Smooth is better than sharp: A random mobility model for simulation of wireless networks. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 19–27. ACM, 2001.

[17] B. Liang and Z. J. Haas. Predictive distance-based mobility management for pcs networks. In *Proceedings of IEEE INFOCOM*, pages 1377–1384. IEEE, 1999.

[18] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, 1993.

[19] R. Singh and S. Patel. Lightweight federated learning for resource-constrained iot networks. *Sensors*, 2025.

[20] H. Wang and L. Zhao. Adaptive federated learning for heterogeneous iot environments. *IEEE Access*, 2024.

[21] Tobias Hoßfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martin Varela. Definition of qoe fairness in shared systems. *IEEE Communications Letters*, 21(1):184–187, 2016.

[22] A.S. Tanenbaum and D.J. Wetherall. *Computer Networks*. Prentice Hall, 2011.

[23] D. Stiliadis and A. Varma. Latency-rate servers: A general model for analysis of flow control and scheduling algorithms. *IEEE/ACM Transactions on Networking*, 6(5):611–624, 1998.

[24] Y. Hu and Z. Wang. Proportional fair scheduling in wireless sensor networks. *Computer Communications*, 31(14):3420–3429, 2008.

[25] J. Chen and S. Liu. Adaptive bandwidth allocation based on queue-aware scheduling in wireless networks. *IEEE Access*, 7:16549–16558, 2019.

[26] Z. Zhou, X. Xu, and J. Li. Machine learning for wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 22(3):1682–1709, 2020.

[27] Y. Zhang and A. Kumar. Machine learning-based resource management in wireless sensor networks: A survey. *Computer Networks*, 2025.

[28] S. Niknam, H.S. Dhillon, and J.H. Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6):46–51, 2020.

[29] R. Patel and K. Mehta. Dynamic resource allocation in mobile iot networks using learning-based models. *IEEE Internet of Things Journal*, 12(3):1987–1999, 2025.

[30] L. Chen and H. Zhang. Mobility-aware task scheduling in edge computing systems. *IEEE Transactions on Vehicular Technology*, 73(5):6890–6902, 2024.

[31] The ns-3 network simulator. Available at http://www.nsnam.org, 2015.

Sravya Pallantla is currently pursuing the Ph.D. degree at Jawaharlal Nehru Technological University, Kakinada, India. Her research interests include computer networks and machine learning, with a focus on resource allocation in wireless networks.

D Haritha received the Ph.D. degree from the Jawaharlal Nehru Technological University, Kakinada, India. She has nearly 20 years of experience in teaching undergraduate and postgraduate students of computer science and engineering. Her research interests include machine learning and image processing.