IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 September 2022, accepted 16 October 2022, date of publication 19 October 2022, date of current version 27 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3215758

RESEARCH ARTICLE

Toward Efficient Hierarchical Federated Learning Design Over Multi-Hop Wireless Communications Networks

TU VIET NGUYEN[1], (Member, IEEE), NHAN DUC HO[2], HIEU THIEN HOANG[2], CUONG DANH DO[2,3], (Member, IEEE), AND KOK-SENG WONG[2], (Member, IEEE)

[1]Wireless Communications and Connectivity Division, Broadcom Ltd., San Diego, CA 92127, USA
[2]College of Engineering and Computer Science, VinUniversity, Hanoi 10000, Vietnam
[3]Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14850, USA

Corresponding author: Kok-Seng Wong (wong.ks@vinuni.edu.vn)

This work was supported in part by the VinUniversity Seed Grant Program.

ABSTRACT Federated learning (FL) has recently received considerable attention and is becoming a popular machine learning (ML) framework that allows clients to train machine learning models in a decentralized fashion without sharing any private dataset. In the FL framework, data for learning tasks are acquired and processed locally at the edge node, and only the updated ML parameters are transmitted to the central server for aggregation. However, because local FL parameters and the global FL model are transmitted over wireless links, wireless network performance will affect FL training performance. In particular, the number of resource blocks is limited; thus, the number of devices participating in FL is limited. Furthermore, edge nodes often have substantial constraints on their resources, such as memory, computation power, communication, and energy, severely limiting their capability to train large models locally. This paper proposes a two-hop communication protocol with a dynamic resource allocation strategy to investigate the possibility of bandwidth allocation from a limited network resource to the maximum number of clients participating in FL. In particular, we utilize an ordinary hierarchical FL with an adaptive grouping mechanism to select participating clients and elect a leader for each group based on its capability to upload the aggregated parameters to the central server. Our experimental results demonstrate that the proposed solution outperforms the baseline algorithm in terms of communication cost and model accuracy.

INDEX TERMS Federated learning, distributed machine learning, multi-hop wireless networks, communication-efficiency, bandwidth optimization.

I. INTRODUCTION

Machine learning (ML) has emerged as one of the evolving technologies of the modern-day. The success of ML systems depends on the availability of high-quality data collected from various sources such as sensors and internet-of-things (IoT) devices. However, a single entity might not own all the data it needs to train the ML model it wants; instead, valuable data examples or features might be scattered in different organizations or entities. For example, autonomous vehicle sensing data sit in data silos, and privacy concerns limit sharing such data for ML tasks. Consequently, large amounts and diverse data from different vehicles are not fully exploited by ML.

The concept of federated learning (FL) was first introduced by McMahan et al. [1]. The main idea is to train the ML models in a decentralized fashion where no private dataset is sent to a central repository [2]. Specifically, the data for the learning tasks are acquired and processed locally at the edge node, and only the updated ML parameters are transmitted to the server for aggregation purposes [3], [4]. The goal of FL is to train a single ML model using all the data available in a cooperative way without moving the training data across the organizational or personal boundaries [5]. FL has been

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

successfully deployed in many applications in many industries, including healthcare, telecommunications, IoT, manufacturing, and surveillance system. For instance, in a smart transportation system, a traffic management agency wants to improve traffic congestion and traffic signal control by training a high-quality FL model based on the local updates collected from the vehicles in a target region. Given the wide applications of FL, guaranteeing that such a cooperative learning process is reliable is becoming essential research topic.

Despite significant recent milestones in FL, there are several fundamental challenges that yet need to be addressed in order to enable its promise [6]. For instance, edge nodes have often substantial constraints on their resources such as memory, computation power, communication, and energy, which severely limits their capability of training large models locally. Also, the device hardware heterogeneity causes edge nodes to complete the training task at different times. In FL, there is a need of each node to efficiently transmit its learned model updates to the server over the uplink communication channel [7]. Often, the throughput of the communication channel is constrained due to issues such as package loss, latency, jitter, etc. Furthermore, there is a large amount of system and data heterogeneity across edge nodes, which will make their learning objectives and capabilities vastly different.

Among the challenges mentioned above, the communication overhead constitutes a major bottleneck in FL systems. In the next generation of wireless networks such as the fifth generation (5G) and sixth generation (6G) networks, a base station essentially serves thousands of devices [8], [9]. An efficient communication protocol together with a FL algorithm need to be addressed. Several works have studied the design of communication-efficient FL algorithms in the literature [10], [11], [12], [13]. The majority of these studies have focused on optimization of FL in a single aspect such as device selection and scheduling [14], [15], FL model parameter updates and transmission [11], [16], or network resource management [17], [18]. In [19], the authors proposed a communication-efficient FL framework that tackles multiple causes for communication delay by jointly optimizing the device selection, FL model parameter transmission, and network resource management. The selection of participating clients is one of the essential considerations in FL. The heterogeneity of the client devices and their limited communication and computation resources can affect the model accuracy because some might not be able to complete the training task in a certain round [20].

In this paper, we deal with the problem of communication network resource allocation in FL training. Specifically, we aim to address three challenges. First, we want to involve as many clients as possible in the training task to increase the accuracy of the ML model. Second, we want to minimize the FL convergence time without sacrificing the training loss of the FL algorithm. Third, we want to reduce the training loss in case the communication is unavailable between the server and clients.

The main contributions of this paper are summarized as follows.

1) We solve the first two challenges by designing an adaptive grouping mechanism with dynamic resource allocation strategy. Given a limited network resource, the idea is to allocate as many clients as possible to participate in FL. With this approach, the global model can acquire more information from the selected clients, improving the convergence time. We utilize a probabilistic client selection mechanism that considers both strengths of the receiver signal and channel gain of each device to elect a leader and participating clients in each group.
2) The third challenge requires the participating clients to establish a new connection with nearby leaders when the communication with existing leader is broken. In our design, we utilize an ordinary hierarchical design with a dynamic group leader mechanism where each group leader is responsible for communicating with the server for local updates submission and global model parameters dissemination. When the leader is unavailable, we allow the clients to establish direct communication with nearby leaders.
3) Experimental results demonstrate that our proposed two-hop communication protocol provides good performance for different scenarios and FL algorithms. The results verify our theoretical findings that the proposed protocol can be applied for real-world application.

The rest of this paper is organized as follows. We discuss the background and related work in Section II and the proposed grouping mechanism, clients and leaders selection are presented in Section III. We present our system design in Section IV, followed by experiment results in Section V. The conclusion is presented in Section VI.

II. BACKGROUND AND RELATED WORK

This section introduces the basic FL phases, its de factor algorithm FedAvg, architecture design and related works in FL client selection, communication optimization and architecture consideration.

A. FEDERATED LEARNING

Consider a general FL framework consisting of one central orchestration server $\mathcal{S}$ and N clients $\{C_1, C_2, \ldots, C_N\}$. Each client i (e.g., a computing device) with a local private dataset D_i and would like to participate in FL process. Due to privacy concern, it is not desirable for the clients to transfer their private dataset to $\mathcal{S}$ or a central repository. Also, $\mathcal{S}$ wants to learn a global model with the data distributed across these clients.

The basic process of FL includes local gradient computation at clients and model weight aggregation by a central

aggregation server. In general, FL involves the following three main phases:

- *Phase 1 (FL Initialization)*: $\mathcal{S}$ first initiates the weight of the global model and the hyperparameters such as the number of FL rounds and local epochs, size of the selected clients for each round, and the local learning rate.
- *Phase 2 (Local Model Training and Update)*: each clients C_i receive the current global weight from $\mathcal{S}$ and updates its local model parameters w_i^t using local datasets, where t denotes the current iteration round. Upon the completion of the local training, all clients send the local weight to $\mathcal{S}$ for model aggregation.
- *Phase 3 (Global Model Aggregation and Update Phase)*: $\mathcal{S}$ aggregates the received local weights and sends back the aggregation result to the clients for the next round of training.

In Phase 2, the goal of each C_i is to obtain the optimal local model parameters $\hat{w}_i^t$ in round t by minimizing the loss function $F_l(w_i^t)$ formulated as follows:

$$\hat{w}_i^t = \arg\min_{\hat{w}_i^t} F_l(w_i^t) \tag{1}$$

In Phase 3, the goal of $\mathcal{S}$ is to obtain the optimal global model parameters $\hat{w}_g^t$ by minimizing the loss function $F_l(w_g^t)$ formulated as follows:

$$\hat{w}_g^t = \arg\min_{\hat{w}_g^t} F_l(w_g^t) \tag{2}$$

such that $F_l(w_g^t) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} F_l(w_i^t)$. The FL will continue until the maximum number of rounds is reached or the accuracy of the global model is greater than the threshold τ. Then, $\mathcal{S}$ completes the FL by aggregating the local updates and distributes the final global model to C_i. Note clients do not need to communicate with each other since the local parameters aggregation is performed by $\mathcal{S}$.

B. FEDERATED LEARNING ARCHITECTURE

Centralized architecture [21] is a commonly used setting in FL where clients are connected directly to $\mathcal{S}$. As shown in FIGURE. 1, $\mathcal{S}$ is responsible for communicating with all clients, aggregating local updates, and deploying the global model. However, due to the heavy communication load with clients, there is a possible communication bottleneck and single-point failure. In particular, $\mathcal{S}$ can only access a limited number of clients lead to inevitable training loss. Because of this, some alternative architecture designs have been proposed in the literature to overcome the issues in the centralized architecture.

For instance, in an ordinary hierarchical FL, several coordinators (with one or more layers) will be added between $\mathcal{S}$ and the clients [22], [23], [24]. All clients will be partitioned and connected directly with these coordinators. This architecture design can reduce data exchange but require permanent coordinators to participate in the learning process. Recently, a client-edge-cloud hierarchical FL has been proposed in [23], where a cloud server is used to support coordinators in processing the massive local updates and allows a quicker model update. However, such a design also required permanent coordinators in the learning process.

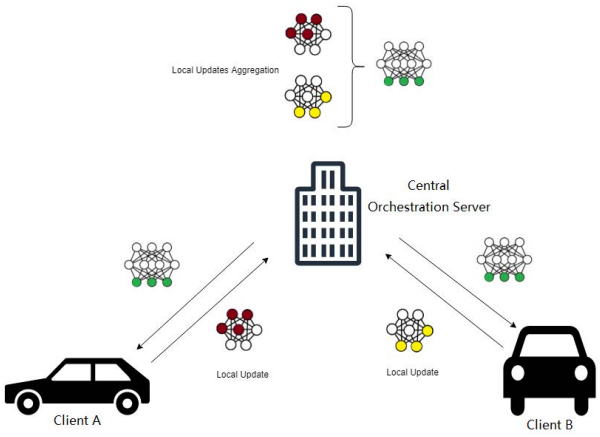


FIGURE 1. A federated learning training model.

C. FEDERATED LEARNING ALGORITHMS

Federated averaging (FedAvg) is the de facto algorithm that allows a subset of devices to performs local iterations in parallel in each round [25]. FedAvg has been successfully deployed in various application domains such as mobile keyboard prediction [21], autonomous driving [26], and payment system [27]. In FedAvg, every participating client first downloads and trains the global model on their local dataset. This process is often known as local or parallel stochastic gradient descent (SGD) where the training is performed for a number of epochs locally. The clients then upload the difference between their initial and final model to $\mathcal{S}$ for local updates averaging.

However, when the data is non-identically distributed (non-IID) across clients and the number of data samples varies significantly from client to client, FedAvg might diverge in realistic scenarios [28]. Several FL algorithms (e.g., FedProx [6], Qffedavg [12], FedFS [29], and FedOptim [30]) have been proposed in the literature to overcome the limitations of FedAvg.

D. RELATED WORK

In recent years, communication efficiency and power management are two active research areas that drive many researchers into the fields. To enhance the availability of FL, several works focus on performance optimization of FL in wireless networks. For instance, Tran et al. [31] analyze the impact of wireless environment for the time of FL task. In [32], the authors optimize the radio resources by scheduling the devices to minimize the convergence time of FL.

Client selection plays an essential role in optimizing the communications of FL systems. For instance, involving more clients in the current FL round reduces the bandwidth

allocated to each client [33]. Most of the recent works assume full client participation in every training round [34], [35], [36] but can be infeasible for FL training with large scale devices. In practice, not all clients can participate in each round due to computation, energy, and bandwidth limitations. Due to this, several mechanisms have been proposed for the selective client to participate in the FL training, such as partial client participation [37], flexible client participation [38], and grouping clients based on network resource and hardware capabilities [39]. In [20], a multi-criteria client selection approach has been proposed by considering the availability of resources, communications overhead, and imbalanced distribution of data. In another work [40], the client selection is based on the long-term average model exchange time.

Zhang et al. [41] proposed a series of schemes for easing the overlarge communication burden in FL systems applied to traffic forecasting tasks using deep models. The proposed clustering-based hierarchical and two-step-updated FL (CTFed) scheme guarantees accurate forecasting performance, circumventing the adoption of any gradient quantization or sparsification approaches that may degrade the performance of collaboratively-trained models when the model's architecture is complex, and the number of involved parameters for feature learning is large. The proposed approach is also orthogonal to the gradient quantization or sparsification approaches. Recently, Lee et al. [42] proposed a solution based on the behaviors of devices' owners. They exploit clustering algorithms to group devices with similar models (i.e., similar gradient updates) and then suppress the training updates of some devices to reduce the communication cost. However, these solutions cannot adapt to a scenario where the clients are changing over time, like in the network traffic.

Unlike the existing solutions, we consider the strength of the receiver signal and channel gain rather than comparing the gradient updates from all clients. Specifically, this work investigates the possibility of bandwidth allocation from a limited network resource to a maximum number of clients to participate in FL. We elect a leader based on its capability to upload the aggregated parameters to the BS successfully. Furthermore, we consider an adaptive environment where the group size and clients can differ at each FL iteration or within a training period.

In an ordinary hierarchical FL, the FL system will fail when one or more coordinators cannot perform their tasks. Furthermore, this will cause clients connected to a failed coordinator cannot join in the later rounds and hence, affect the performance of global model. Although in practice, only a small fraction of clients participate in each learning round, but the exclusion of a group of clients in several rounds will cause biased client participation issue. Inactivity of these clients may be temporary or permanent, depending on their connection with the coordinator. Ultimately, this will restricts the potential availability of training datasets in those inactive clients [43]. Also, it is worth mentioning that the deployment of FL with an ordinary hierarchical design will incur additional costs since the coordinators must be powerful servers with high computational resources. In client-edge-cloud hierarchical FL also required permanent coordinators in the learning process. Furthermore, there is a possible downtime with the cloud server and delay, which can endanger real-time FL applications such as autonomous vehicle systems. The usage of cloud services will further increase the deployment costs of FL in real-world applications.

In view of the limitations in the existing FL architecture design, this paper proposes to utilize an ordinary hierarchical design with a dynamic group leader mechanism (see Section III-B). The group leaders take the role of coordinators in the ordinary hierarchical FL but are not in a permanent fashion. In other words, when an elected group leader is unavailable due to a lost connection or limited transmission range, another client (e.g., the second best client) will replace it by taking responsibility as a new leader. Also, we allow inactive clients to connect to the nearest group leader in case no good candidate is available to serve as the group leader. Unlike the existing works, our architecture design can prevent problems caused by the failure of one or more permanent coordinators and deals with inactive clients during the learning process in FL.

TABLE 1. Summary of key notations.

Notation	Description
$\mathcal{S}$	Central orchestration server
$\mathcal{C}_i$	Participating client i
$\mathcal{L}_i$	Elected leader i
G_i	Partitioned group i
$\hat{w}_i^t$	Optimal local model parameters
$\hat{w}_g^t$	Optimal glocal model parameters
N	Number of clients in the system
K	Number of groups
N_k	Number of clients in k-th group.
(k,j)	j-th client in the k-th group $(k, j \geq 1)$
B	Total system available bandwidth
$B_{k,j}$	Bandwidth allocated to the (k,j)-th client
B_k^{bs}	Bandwidth allocated to the k-th leader to the BS
$\lvert h_k^{bs}\rvert^2$	Channel gain from k-th leader to the BS
$\lvert h_{k,j}\rvert^2$	Channel gain from (k,j)-th client to the its leader
$\lvert h_{m,j}^k\rvert^2$	Channel gain from (m,j)-th client to the k-th leader
N_0	Noise power spectral density
$r_{k,j}$	Achievable rate of the (k,j)-th client
r_k^{bs}	Achievable rate of the k-th leader to the BS
$P_{k,j}$	Transmit power of the (k,j)-th client
P_k^{bs}	Transmit power of the k-th leader to the BS
$\mathcal{K}$	Set of group indices, $\mathcal{K} = \{1, 2, \ldots, K\}$.
$\mathcal{N}_k$	Set of client indices, $\mathcal{N}_k = \{1, 2, \ldots, N_k\}$.

E. NOTATIONS USED

We summarize all key notations in this paper in Table 1.

III. PRELIMINARY

A. GROUPS SELECTION

Selecting a set of groups in each round of update optimally given the limited channel information among clients is a challenging task. This paper proposes a simple round robin method to select a fixed percentage of groups to transmit

FIGURE 2. Group selection: 9 groups out of 81 are selected to upload parameters every round.

updates in each round. To limit the interference amongst groups, we select the groups based on their geometric locations such that they are not adjacent to each other. For example in FIGURE 3, the entire serving area of the BS is divided into $K = N_g \times N_s = 9 \times 9$ groups; that is, there are 81 groups in total. In each update round, $N_g = 9$ groups of the same color (or number) are selected to potentially participate in the update process (the actual participated groups can be less than N_g if the communication resource is limited). We need N_s update rounds such that all groups are potentially participating. Let denote the $\mathcal{K}_i$ are the set of group indices selected in the i-th round ($\mathcal{K} = \cup_{i=1}^{N_g} \mathcal{K}_i$).

B. GROUP LEADER SELECTION

Once the groups are selected, the group leader is selected such that it has the most capability to upload the aggregated parameters to the BS successfully. Hence, it is clear that the client with the highest receiver signal at the BS should be chosen. That is, we will choose

$$i_{ldr} = \arg\max_{i \in \mathcal{N}_k} P_i^{bs} |h_i^{bs}|^2 \tag{3}$$

If all clients can transmit at max power $P_i^{bs} = P^{\max}$, then the clients with the highest channel gain to the BS are selected to be the leader. That is

$$i_{ldr} = \arg\max_{i \in \mathcal{N}_k} |h_i^{bs}|^2 \tag{4}$$

Without loss of generality, we assume the group leader is the first client, or the client with index 0 (if not, we can re-index the client). Note that due to random fading, the channel condition changes after every round, and a new leader may be selected.

The achievable transmission rate in bits-per-second (bps) between the l-th group leader and the BS, in the second time slot, is given by

$$r_k^{bs} = B_k^{bs} \log_2 \left(1 + \frac{P_k^{bs} |h_k^{bs}|^2}{B_k^{bs} N_0}\right) \tag{5}$$

where B is the total bandwidth, B_k^{bs} is the allocated bandwidth for the l-th group leader, N_0 is the baseband noise spectral density, P_k^{bs} is the transmission power, and $|h_k^{bs}|^2$ is the channel gain of the l-th leader. We assume h_k^{bs} follows Rayleigh slow fading channel model, and h_k^{bs}'s are independent. Also, we must have the bandwidth constraint $\sum_{k \in \mathcal{K}} B_k^{bs} \leq B$.

C. CLIENT PARTICIPANTS SELECTION SCHEME

In the first time-slot, the selected clients in all groups reuse the same bandwidth B, which is divided among all clients within the group. Because of the frequency reuse, client of one group can be interfered from clients of other groups if they share the same sub-channels. In general, the bandwidth allocation for each group can be independent; however, it is very complicated to jointly optimize the bandwidth and power allocation amongst groups due to the interference from clients which has partial bandwidth overlap with the main client. In this work, to simplify the problem, we assume the whole bandwidth B is divided onto J sub-channels of bandwidths $B_1, B_2, \ldots, B_J$ such that $B_1 \leq B_2 \leq \ldots \leq B_J$ and $\sum_{j \in \mathcal{J}} B_j \leq B$, where $\mathcal{J} = \{1, 2, \ldots, J\}$. Let assume the clients are reordered/re-indexed such that the j-th client in each group is allocated to the j-th sub-channel of bandwidth B_j. Therefore, the achievable rate from the j-th client to the leader in the k-th group, $\mathcal{L}_k$, is given by

$$r_{k,j} = B_j \log_2 \left(1 + \frac{P_{k,j} |h_{k,j}|^2}{B_j N_0 + \sum_{m=1, m \neq k}^{K} P_{m,j} |h_{m,j}^k|^2}\right) \tag{6}$$

where $P_{k,j}$ and $|h_{k,j}|^2$ are the transmission power and the channel gain, respectively, of the j-th client in the l-th group. Also, $|h_{m,j}^k|^2$ is the channel gain of the (m, j)-th client to the k-th leader who use the same bandwidth. Note that the second term in the numerator in (6) is the interference from clients of other groups who use the same subchannel as (m, j)-th client.

IV. SYSTEM DESCRIPTION

We consider a mobile network with one BS serving N clients. The total available bandwidth is B, which is allocated amongst clients using a frequency-sharing multiple access protocol, such as FDMA or OFDMA [44]. We assume the BS and each client has one antenna for simplicity. Note that the results in this paper can be extended straightforwardly to the scenario of a BS with multiple antennas with a maximum receiver combining (MRC) receiver [44]. Usually, N is large and it is inefficient for all clients to update their local learning parameters in every round. So during each round, only a subset of clients are selected to transmit their local update to the BS.

To support the deployment of FL over a mobile network, we design a collaborative system that involves a resource allocation mechanism, learning nodes partition, and network monitoring role selection. The main idea is to allocate the available block resources to the nearest nodes and help others submit the local model updates to the agency. Furthermore, we aim to include training data from all nodes within the

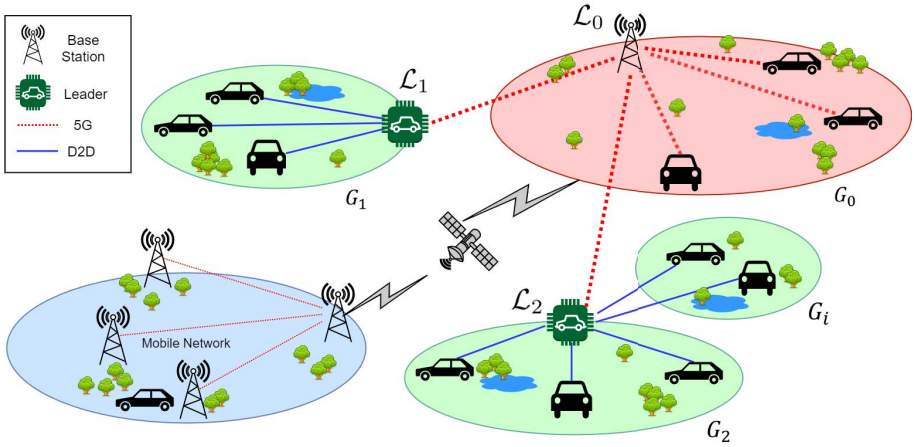


FIGURE 3. Illustrative example of FL system with the proposed architecture design.

target region to ensure the learning accuracy of the FL model. Ultimately, our system can reduce the FL training loss and improve FL convergence time.

A. ARCHITECTURE DESIGN

We illustrate an example of FL system with the proposed architecture design in FIGURE 3. The system consists of N clients that are partitioned into K groups $\{G_0, G_1, \ldots, G_K\}$ based on their geometric locations. In each group, one of the clients will be elected as group leader $\{\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_K\}$. The group leader is responsible for collecting and aggregating local updates from the clients within the group. The second-best client will replace the existing one when an elected group leader is unavailable due to connection loss or high transmission power (e.g., limited transmission range and energy inefficiency). If no good candidate for the leader, all clients should join the nearest group. For example, as illustrated in FIGURE 3, clients in G_i establish a direct connection with $\mathcal{L}_2$. Note that the BS plays the role of $\mathcal{L}_0$ in G_0. Also, all clients connect with leaders via device-to-device (D2D) connection while leaders establish a wireless network connection (5G) with BS.

B. LOCATION-BASED GROUP PARTITIONING

The transmission overhead in FL is much lower than the traditional centralized learning methods since clients only update their local parameters to the BS. However, BS may become a bottleneck if the model involves many learnable parameters or limited bandwidth with clients and BS. There are several methods to mitigate the transmission overhead in an FL-based framework, such as compressed sensing and model compression [45]. For instance, Chen et al. [46] proposed a multi-hop collaborative framework that helps the edge devices to reach the Base station by sending training parameters through their neighbors.

In this work, we propose a location-based clustering method based on multi-hop collaboration combined with the resources allocation optimization algorithm that reduces the transmission overhead and helps the FL algorithm to converge faster. FIGURE 4 shows an example of a real scenario of the Group-based method FL (GFL) applied in vehicular traffic network partitioning. To conserve the communication efficiency of the wireless networks, we propose an adaptive grouping coordination design to group clients based on their computational power, energy, the distance of clients to the base station, and signal strength. New groups will be formed for every fixed number of iterations to adapt to the new network condition.

C. COMMUNICATION MODEL

Since the BS has high power capability and multiple antenna are equipped and the whole bandwidth can be used to broadcast the global models to all clients. We assume that all clients are able to successfully receive the global model update from BS within a short duration. Similar to [47], in this paper we do not consider the optimization for broadcasting the global models to clients. We instead consider the optimization problem for local model updates from clients to the central orchestration server (located at the BS).

In the following, we consider two communication protocols. The first one is single-hop communication, which is similar to many existing works [19], [47], and considered as a baseline. However, as the distance getting longer for some certain clients, the transmission requires high power. A multi-hop routing is preferred in a traditional wireless communication network to improve energy efficiency. In this work, for the first time we propose a two-hop communication model for FL. It is expected not only improving the battery life-time of the clients but also improving the converging time of the overall model.

1) SINGLE-HOP COMMUNICATION

Similar to many existing works, in this protocol, a subset of clients are selected to transmit the local parameters directly to the BS in each updating round. The clients selections can

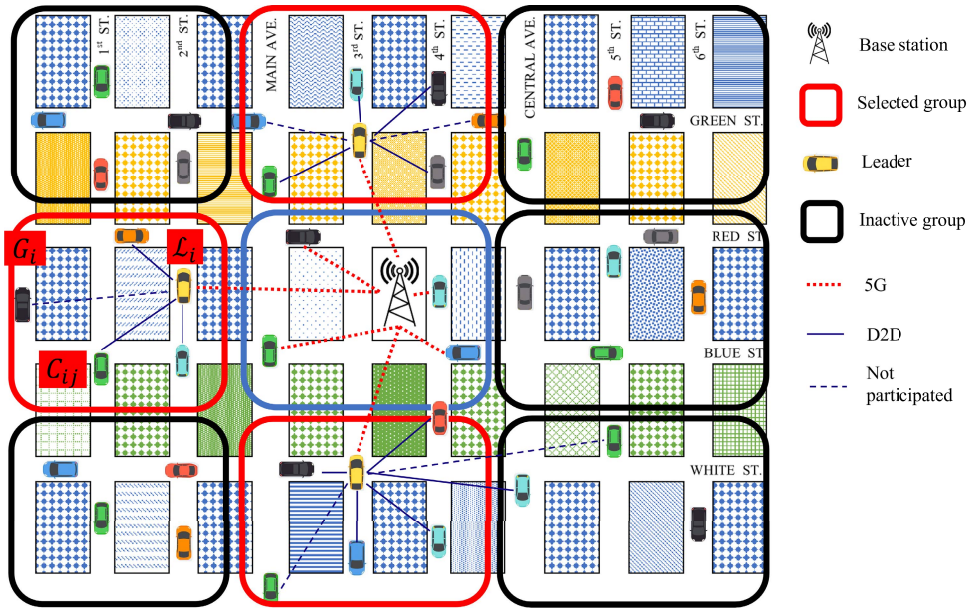


FIGURE 4. Location-based allocation for active, Inactive groups, and leader of active group.

be performed using the methods presented in [19] and [47]. We will use these methods as baselines.

2) TWO-HOP COMMUNICATION

In this subsection, we assume the N clients are divided into K groups, each consisting of N_k clients for $k = 1, \ldots, K$, and $\sum_{k\in\mathcal{K}} N_k = N$. The k-th group has a group leader, called L_k, which is generally selected if it has a strong connection to the BS, see FIGURE 4 and FIGURE 3.

In each updating round, a subset of the groups are selected to participate in the updates. Also, a subset of clients in each selected group are allowed to send the updated local parameters to the BS. Each communication round is divided into two time-slots.

- In the first time slot, all the selected clients in a selected group are allocated bandwidth to transmit the updated local parameters to their group leader. The group leader aggregates the parameters within its group. Note that the clients are grouped together if they are located close to each other. In this time slot, all selected groups are assumed to be located separate from each other, hence, we can reuse the same bandwidth for all groups.
- In the second time slot, all selected group leaders transmit the aggregated parameters to the BS. Each leader is allocated its own bandwidth.

The BS is finally aggregate the parameters from all groups. The BS then broadcast the global updated model back to all clients.

In this work, we assume the base station periodically and accurately estimates the channel gains and locations from all the clients. Similar to a block fading channel model [44], we assume both the channel gains and locations are randomly and independently changing from one time slot to the next time slot. This time slot is predefined and smaller than the coherent time of the fading channels. We note that the mobility of clients could result in changing in clusters they belong to. Our approach is to periodically update the cluster members based on their locations, and the group leaders are also updated based on the procedure described in Section III.B.

D. RESOURCE ALLOCATION PROBLEM FORMULATION

Since the proposed two-hop protocol requires two time slots to transmit update from clients to BS, we need to be able to transmit twice the rate in each time slots compared to the single-hop protocol. Let assume, within an assigned time slot, it requires a transmission rate of at least $2R$, otherwise, the packet is lost. That is, we use information theoretical approach to assume that the packages can be decoded successfully at the receiver if the instantaneous SINR is greater than a threshold, which is equivalent to the instantaneous channel capacity is greater than the required rate to transmit the updated package. Otherwise, the updated package failed to decode and was considered as lost [44]. In this case, the server will skip the aggregation from this clients. Our objective is to maximize the total number of successfully updated clients in each time slot; hence, maximize the overall FL performance.

Due to the fading channel, the outage probability of the (k, j)-th client is given by

$$p_{k,j} = \Pr(r_{k,j} \geq 2R) \tag{7}$$

That is, if the rate $r_{k,j} \geq 2R$, then the (k, j)-th client is successfully upload its parameters to its leader; otherwise, it is considered as failed to update. Similarly, the outage probability of the k-th leader in the second time slot

is given by

$$p_k^{bs} = \Pr(r_k^{bs} \geq 2R) \tag{8}$$

We note that the (k, j)-th client is successfully uploaded its parameter to the BS if and only if both $r_{k,j} \geq 2R$ and $r_k^{bs} \geq 2R$ satisfy.

Similar to [47], to optimize the convergence rate, we need to allocate the resource such that it maximize the number of successful transmission from clients to BS via their leaders.

We assume the channel gains can be accurately estimated at the BS. We denote I_k^{bs} and $I_{k,j}$ to be the indicators that the k-th leader and (k, j)-th client transmit their updated parameters successfully, respectively. We note that the (k, j)-th client's transmission to the BS is successfully if both $I_k^{bs} = 1$ and $I_{k,j} = 1$. Therefore, the resource optimization problem is formulated as follows

$$\begin{aligned} \text{(P1)} \quad \max_{\boldsymbol{B},\boldsymbol{P}} \quad & \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{N}_k} I_k^{bs} \cdot I_{k,j} \\ \text{s.t.} \quad & P_{\min} \leq P_{k,j} \leq P_{\max}, \\ & P_{\min} \leq P_k^{bs} \leq P_{\max}, \\ & \sum_{j\in\mathcal{J}} B_j \leq B, \quad \sum_{k\in\mathcal{K}} B_k^{bs} \leq B, \end{aligned} \tag{9}$$

where $I_{k,j} = 1$ if $r_{k,j} \geq 2R$, and $I_{k,j} = 0$ if otherwise, and $I_k^{bs} = 1$ if $r_k^{bs} \geq 2R$, and $I_k^{bs} = 0$ if otherwise.

Algorithm 1 Groups and Clients Selection

Input: t: update round index, N_g: number of super groups, N_s: number of groups in the super group, $\{G_k\}$: sets of clients, $\{\mathcal{K}_i\}$: the set of all sub-groups.
Output: Sets of selected groups $\mathcal{K}^s$ and sets of clients $\{\mathcal{C}_k^s\}$.

[*Central Orchestration Server*]
Initialize the round-robin index $r = (t \mod N_s) + 1$.
[*Group Selection*]
for each group G_k with index $k \in \mathcal{K}_r$ **do**
  Assign group leader for the k-th group based on equation (3).
end for
Solve for (P2.1), the leader bandwidth allocations, in (10) to obtain a list of participant groups, $\mathcal{K}^s$.
[*Clients Selection*]
for each group $k \in \mathcal{K}^s$ **do**
  Order the channel gains from clients to its leader from high to low.
end for
Solve the bandwidth allocation problem (P2.2) in (11) using the suboptimal method described after (P2.2), we obtain the sets of participant clients in each participant group $\mathcal{C}_k^s$ for $k \in \mathcal{K}^s$.
return $\mathcal{K}^s$ and $\{\mathcal{C}_k^s\}$

We observed that (P1) is a nonconvex optimization problem because $I_{k,j}$ and I_k^{bs} are discrete valued function, which takes value $\{0, 1\}$. To obtain a suboptimal solution, we propose to decompose (P1) into multiple sub-problems. First,

Algorithm 2 Our Proposed Solution

Input: M: Maximum number of rounds, m: the number of clients selected in each round, N_{epoch}: the number of local epochs, and η: the local learning rate
Output: Global Model w_g

[*Central Orchestration Server*]
Initialize global weight w_g^0, M, m, N_{epoch}, and η
for each round t from 1 to M **do**
  Follow the **Algorithm 1** to obtain the set of participant groups $\mathcal{K}^t$ and sets of clients $\mathcal{C}_k^t$ for $k \in \mathcal{K}^t$.
  for each participant group $k \in \mathcal{K}^t$ **do**
    for each client $i \in \mathcal{C}_k^t$, including the leader, **in parallel do**
      $w_{k,i}^t, N_{k,i} \leftarrow$ **LocalTraining**(k, i, w_g^t)
    end for
    [*The k-th Leader*]
    $w_{k,g}^{t+0.5} = \frac{1}{N_k} \sum_{i\in\mathcal{C}_k^t} N_i w_{k,i}^t$, where $N_k = \sum_{j\in\mathcal{C}_k^t} N_{k,j}$ (at the k-th leader)
  end for
  [*Central Orchestration Server*]
  $w_g^{t+1} = \frac{1}{\sum_{k\in\mathcal{K}^t} N_k} \sum_{k\in\mathcal{K}^t} N_k w_{k,g}^{t+0.5}$ (at $\mathcal{S}$)
end for
[*Participating Clients*]
Each client (k, i) divides local dataset $D_{k,i}$ into batches, $B_{k,i}$
for each epoch j from 1 to N_{epoch} **do**
  for each batch $b \in B_{k,i}$ **do**
    $w \leftarrow w - \eta \nabla L(w; b)$
  end for
end for
return the weight w and $N_{k,i} = |D_{k,i}|$

we optimize the total number of leaders that can successfully transmit to the BS, then we optimize the total number of clients that can successfully transmit the update to its leader. That is,

$$\begin{aligned} \text{(P2.1)} \quad \max_{\boldsymbol{B},\boldsymbol{P}} \quad & \sum_{k\in\mathcal{K}} I_k^{bs} \\ \text{s.t.} \quad & P_{\min} \leq P_k^{bs} \leq P_{\max}, \\ & \sum_{k\in\mathcal{K}} B_k^{bs} \leq B \end{aligned} \tag{10}$$

and

$$\begin{aligned} \text{(P2.2)} \quad \max_{\boldsymbol{B},\boldsymbol{P}} \quad & \sum_{k\in\mathcal{K}^s} \sum_{j\in\mathcal{N}_k} I_{k,j} \\ \text{s.t.} \quad & P_{\min} \leq P_{k,j} \leq P_{\max}, \\ & \sum_{j\in\mathcal{J}} B_j \leq B. \end{aligned} \tag{11}$$

Both (P2.1) and (P2.2) are still nonconvex optimization problems because the objective function include the discrete valued variables $I_{k,j}$ or I_k^{bs}.

Note that solving (P2.1) is similar to the baseline method. That is, we order the channel gain $|h_k^{bs}|^2$ from high to low

and allocate the bandwidth such that the respective leader can successfully transmit to the BS. We repeat the process until the total allocated bandwidth, B, is reached.

To optimally solve for (P2.2), it is more complicated, if computationally feasible, due to the interference terms as shown in (6). In this paper, we propose a suboptimal solution approach as follows. In each group, we order the channel gain $|h_{k,j}|^2$ from high to low, and allocate the client with the highest channel gain to the same channel with bandwidth B_1, the second best one to the same channel with bandwidth B_2, and so on. The B_k is chosen such that it is upper bounded by some limit $B_{\max} < B$ (in simulation we choose, $B/B_{\max} = 3$) and as many clients can successfully transmit the updates to its leader. We repeat the process until all available bandwidth, B, can not be further utilized. That is, $\sum_{i=1}^{K'} \leq B$, but $\sum_{i=1}^{K'+1} > B$, where $K' \leq K$ is the actual active group. We can show that the complexity of the proposed algorithm is $\mathcal{O}(K^2)$.

The proposed algorithms to solve the optimization problem (P1) in (9) is summarized in Algorithm 1 and Algorithm 2.

V. EXPERIMENT RESULTS

In this section, we conduct experiments to investigate how the dynamic resource allocation strategy influences the FL algorithms. We use FedAvg as the basic FL algorithm to evaluate our proposed two-hop communication protocol. To show that the proposed protocol is also applies to other FL algorithms, we conduct experiments on FedProx [6], which is a generalization of the FedAvg algorithm to address the heterogeneity of data and systems in FL.

A. DATA DISTRIBUTIONS AND CONFIGURATIONS

We evaluate our results on CIFAR-10 [48] dataset, which consists of 60000 32×32 colour images in 10 classes, with 6000 images per class. To show how the trained models are impacted due to differences in local data distributions, we configure the data distributions in our experiments as follows:

- *Scenario 1*: In the IID setup, data samples from each class are equally distributed across all $N = 500$ clients in the system. Hence, each client has 100 samples and all 10 classes in its local dataset.
- *Scenario 2*: We distribute the initial dataset equally to each client but with a random number of classes.
- *Scenario 3*: Each client has a random number of data samples (at least 50 data samples) and with a random number of classes.

In the non-IID setup (Scenario 2 and 3), the number of classes is drawn randomly from 4 to 10 for each client. We distribute the training set of each dataset to the clients for training and utilize the original test set of each dataset to evaluate the performance of the global model. We consider a system with $N = 500$ clients. For illustration purposes, we show the sample data distributions for Scenario 2 and 3 in FIGURE 5.

TABLE 2. Simulation parameters.

B	20 MHz	α	3.2
N_0	-174 dBm/Hz	P	20 dbm
Total area	$450 \times 450\ m^2$	Group area	$50 \times 50\ m^2$

B. FL MODEL

In our implementation, we use ResNet18 [49] for CIFAR-10 in PyTorch. In addition, we perform data augmentations by using techniques such as Random Horizontal Flip, Random Rotation and Color Jitter. We run and compare the performance of our proposed two-hop communication protocol for Group-based Federated Learning (*GFL*) with the following two baseline approaches:

- *Centralized Machine Learning (CML)*: All data samples are gathered in a place.
- *Centralized Federated Learning (CFL)*: Every client has a participation rate $pr\%$, either full participation ($pr = 100\%$) or partial participation ($pr < 100\%$).

In *GFL*, we divide N clients into 81 groups where each group has a leader and $n-1$ members, all with a participation rate $pr\%$. Any group with $n < 3$ will not be selected at any round. For a fair comparison with *CML*, we use a fixed random seed in our experimental setup. We ran these experiments on 180 rounds with different participation rates, batch size of 16, and SGD with momentum 0.9 and learning rate 10^{-3} and cosine annealing scheduler [50]. In GFL, we randomize the location of all members every 5 round.

C. COMMUNICATIONS PARAMETERS

We consider an area of 450×450 square meters, and 500 clients uniformly distributed in this area. The BS is located at the center of this area. In the simulation, the whole area is divided into 81 small squares, each with size 50×50 square meters as shown in FIGURE 6.a and 6.b. Clients located in each small square form a group as aforementioned. That is, we use 81 groups of clients in our simulation. Similar to [51], we assumed IID Rayleigh fading channels between any two terminals, which can be clients, leaders or the BS. The pathloss (PL) model is assumed to be

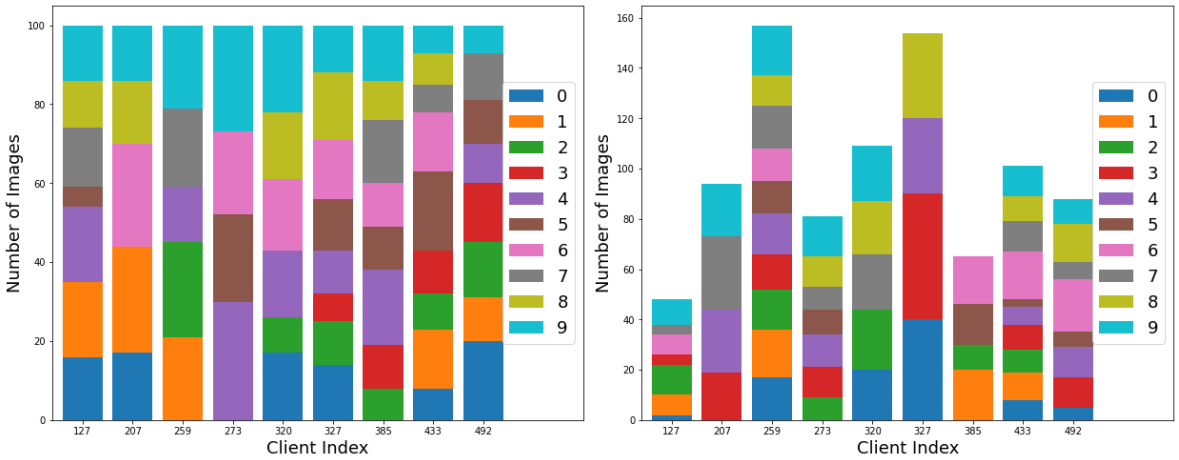
$$PL = -30 + 10\log_{10} d^{\alpha}\ (\text{dB}) \tag{12}$$

where α is the pathloss exponent and d in meter is the distance between two terminals (e.g., between a client and its leader). The other communications parameters are summarized in Table 2.

D. PERFORMANCE EVALUATION

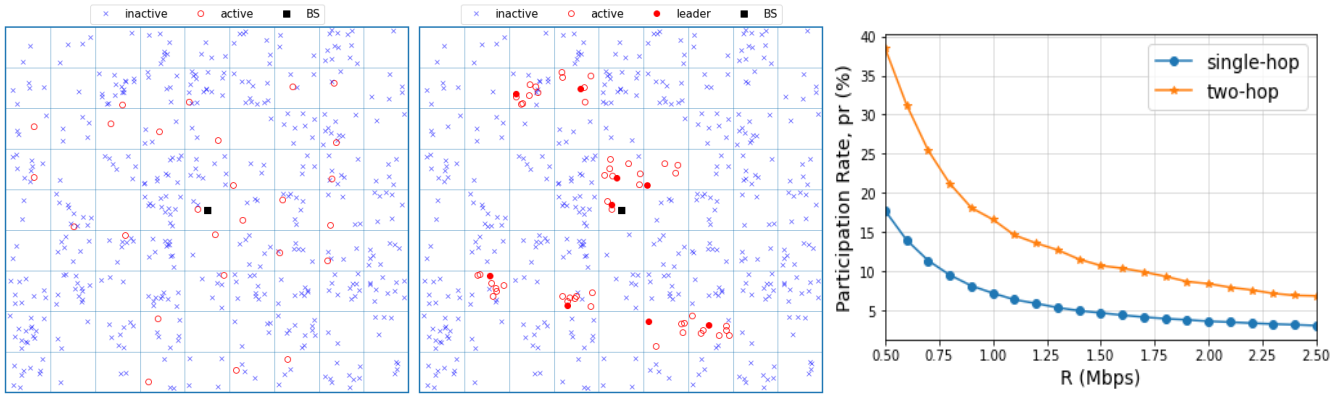
In the simulation, we randomly generated clients' locations and communications channels with parameters as shown in Table 2 with 1000 iterations. The simulation results were then averaged over these 1000 realizations unless otherwise stated.

In FIGURES 6.a and 6.b, we plot the locations of participation clients for a conventional single-hop protocol and our proposed two-hop protocol, respectively (for a particular



(a) Scenario 2 (b) Scenario 3

FIGURE 5. **Sample data distributions.**



(a) Single-hop, 26 active clients (b) Two-hop, 9 leaders and 49 active clients (c) Participation rates

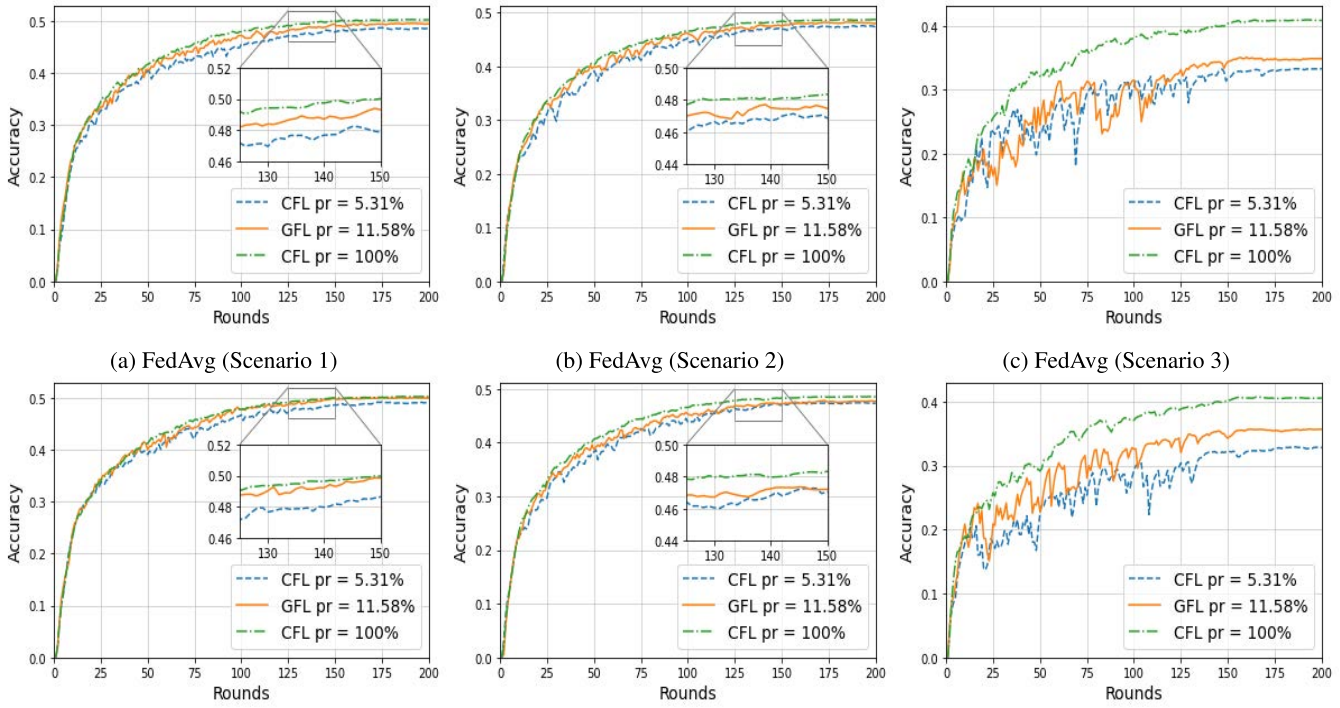
FIGURE 6. **Sample locations of active and inactive clients for $R = 1.4$ Mbps and participation rates.**

clients' locations and channels). As shown in FIGURE 6.a, there are only 26 clients who can participate at a round, while in FIGURE 6.b, for the same communication resources, our proposed two-hop protocol resulted in a higher number of the total active clients, 9 leaders, and 49 clients, which is 58 in total. The gain comes from the more efficient way of exploiting the communications resources such as bandwidth and energy in the two-hop protocol compared to the single-hop protocol. We note that in the two-hop protocol, the D2D communications between clients and its leader are closer in range in the first time slot, thus more energy efficiency, and is more efficient in bandwidth due to the frequency reuse amongst groups.

In FIGURE 6.c, we plot the average participation rates for the conventional single-hop protocol and the proposed two-hop protocol. The participation rate is defined as the ratio between the participated clients (including the leaders) and the total number of clients. We can observe that the proposed two-hop protocol outperforms the conventional protocol. For example, at the rate $R = 1.4$ Mbps, the participation rate for the proposed protocol and the conventional protocol are 11.58% and 5.31%, respectively. In other word, we see about 118.08% gain in participation rate for our proposed two-hop protocol.

In terms of performance, we performed experiments for all three scenarios specified in Section V-A to show the influence of the dynamic resource allocation strategy on model accuracy. As shown in FIGURE 7, the model accuracy of the proposed *GFL* outperform that of the traditional *CFL* over all the considered scenarios. The performance improvement is more pronounced in scenario 3, which is with non-IID. data. Hence, it results a positive insight to applying the proposed protocol for a real-world application.

We also conducted experiment using a modern algorithm, FedProx. We observed that our proposed method *GFL* also outperforms the *CFL* method in all three considered



(a) FedAvg (Scenario 1) (b) FedAvg (Scenario 2) (c) FedAvg (Scenario 3)

(d) FedProx (Scenario 1) (e) FedProx (Scenario 2) (f) FedProx (Scenario 3)

FIGURE 7. The change in global model accuracy with various data distribution scenarios and participating rates. The experiments were implemented using ResNet-18 on CIFAR-10 with one local epoch for up to 200 rounds and different algorithms, FedAvg (first row) and FedProx (second row). The model converged after approximately 150 rounds.

TABLE 3. ResNet-18 on CIFAR-10 with different local epochs and scenarios.

	Participation Rate	Local Epoch	FedAvg Scenario 1	FedAvg Scenario 2	FedAvg Scenario 3	FedProx Scenario 1	FedProx Scenario 2	FedProx Scenario 3
CML	-	-		0.8031			-	
CFL	5.31%	1	0.4873	**0.4767**	**0.3346**	**0.4930**	0.4752	0.3298
		2	**0.5522**	0.5286	**0.3614**	0.5444	**0.5340**	0.3251
		5	0.6064	**0.5857**	**0.3451**	**0.6156**	0.5813	0.3388
	100%	1	**0.5030**	**0.4869**	**0.4101**	0.5026	0.4861	0.4083
		2	0.5701	0.5483	**0.4260**	**0.5711**	**0.5528**	0.4198
		5	**0.6183**	0.5917	0.4672	0.6167	**0.6003**	**0.4736**
GFL	11.58%	1	0.4964	**0.4824**	0.3529	**0.5009**	0.4790	**0.3577**
		2	0.5527	**0.5471**	**0.3339**	**0.5672**	0.5447	0.3338
		5	**0.6069**	0.5877	0.3206	0.6018	**0.5905**	**0.3508**

scenarios. Moreover, the FedProx algorithm results a better performance as compared to the FedAvg algorithm, especially, in scenario 3. In addition, we run the experiments with the same settings but with a different number of local epochs, and the results are summarized in Table 3.

VI. CONCLUSION

In this paper, we have investigated the communication efficiency problem for FL system. Our concern mainly focuses on the possibility of bandwidth allocation from a limited network resource to a maximum number of clients to participate in FL. Based on the connection probability that considers both distance and contribution of each client (to the global model), our proposed solution able to select participating clients and elect a leader in each group. In light of the experiments on our proposed two-hop communication protocol, we found that dynamic resource allocation is crucial for the training performance. In particular, the bandwidth allocation from a limited network resource to a maximum number of clients to participate in FL can improve the accuracy of the global model. The proposed solution is orthogonal to most works on resource allocation for FL systems and can be incorporate together with other approaches such as clustering-based hierarchical, and two-step updated FL (CTFed) scheme.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[3] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[4] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.

[5] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[7] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8851–8855.

[8] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[9] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133995–134030, 2020.

[10] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[11] N. Shlezinger, M. Chen, Y. C. Eldar, H. Vincent Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2021.

[12] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*.

[13] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[14] Q. He, G. Dán, and V. Fodor, "Joint assignment and scheduling for minimizing age of correlated information," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1887–1900, Oct. 2019.

[15] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[16] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, 2021.

[17] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[18] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*.

[19] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 17, 2021, Art. no. e2024789118.

[20] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi, "FedMCCS: Multicriteria client selection model for optimal IoT federated learning," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4723–4735, Mar. 2020.

[21] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[22] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2019, *arXiv:1905.06641*.

[23] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[24] A. Wainakh, A. S. Guinea, T. Grube, and M. Mühlhäuser, "Enhancing privacy via hierarchical federated learning," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Sep. 2020, pp. 344–347.

[25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[26] X. Liang, Y. Liu, T. Chen, M. Liu, and Q. Yang, "Federated transfer reinforcement learning for autonomous driving," 2019, *arXiv:1910.06001*.

[27] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, "Fedcoin: A peer-to-peer payment system for federated learning," in *Federated Learning*. Cham, Switzerland: Springer, 2020, pp. 125–138.

[28] L. T. Nguyen, J. Kim, and B. Shim, "Gradual federated learning with simulated annealing," *IEEE Trans. Signal Process.*, vol. 69, pp. 6299–6313, 2021.

[29] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[30] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

[31] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[32] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[33] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2020.

[34] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local GD on heterogeneous data," 2019, *arXiv:1909.04715*.

[35] R. Pathak and M. J. Wainwright, "FedSplit: An algorithmic framework for fast federated optimization," 2020, *arXiv:2005.05238*.

[36] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," 2020, *arXiv:2002.06440*.

[37] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, *arXiv:1907.02189*.

[38] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3403–3411.

[39] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[40] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[41] C. Zhang, L. Cui, S. Yu, and J. J. Q. Yu, "A communication-efficient federated learning scheme for IoT-based traffic forecasting," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11918–11931, Jul. 2022.

[42] M.-L. Lee, H.-C. Chou, and Y.-A. Chen, "FedSAUC: A similarity-aware update control for communication-efficient federated learning in edge computing," in *Proc. 13th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Nov. 2021, pp. 1–6.

[43] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 10351–10375.

[44] J. Proakis, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2007.

[45] A. M. Elbir, B. Soner, S. Coleri, D. Gunduz, and M. Bennis, "Federated learning in vehicular networks," 2020, *arXiv:2006.01412*.

[46] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 48–54, Dec. 2020.

[47] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless IoT networks with optimized communication and resources," *IEEE Internet Things J.*, 2022.

[48] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 (Canadian institute for advanced research)," 2010, vol. 5, no. 4, p. 1. [Online]. Available: http://www.cs.toronto.edu/kriz/cifar.html

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[50] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2017, *arXiv:1608.03983*.

[51] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. Vincent Poor, "Federated learning over wireless IoT networks with optimized communication and resources," 2021, *arXiv:2110.11775*.

TU VIET NGUYEN (Member, IEEE) received the B.S. degree in electrical engineering from the Post and Telecommunication Institute of Technology (PTIT), Hanoi, Vietnam, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at San Diego, San Diego, La Jolla, CA, USA, in 2010 and 2013, respectively. In 2013, he started working with Broadcom Ltd., San Diego, as a Research and Development System Researcher. He has been working on the research and developments of the current and next generations of the Wi-Fi networks (including IEEE 802.11ag/n/ac/ax/be). From September 2020 to August 2022, he was with the College of Engineering and Computer Science, VinUniversity, Hanoi, as an Assistant Professor. His research interests include multimedia transmissions over wireless networks, cross-layer design optimization, image and video processing, information theory, digital signal processing, multiple antenna systems, resource allocation, Wi-Fi, 5G/6G, intelligent reflecting surface, artificial intelligent, and machine learning applications for wireless communication networks.

NHAN DUC HO received the Engineering degree in mathematics and informatics (advanced program) from the Hanoi University of Science and Technology (HUST), in 2018, and the master's degree in mathematics, cryptology, coding, and applications from the University of Limoges, in 2020. In 2021, he joined the College of Engineering and Computer Science, VinUniversity, Vietnam, as a Research Assistant. His research interests include cryptology and machine learning.

HIEU THIEN HOANG received the B.E. degree in electronics and communications engineering from the Hanoi University of Science and Technology (HUST), Vietnam, in October 2021. He was worked as a Research Assistant at the Communications Theory and Applications Research Group (CTARG), School of Electronics and Telecommunications, HUST. He is currently working as a Research Assistant at the College of Engineering and Computer Science, VinUniversity. His research interest includes the field of resource allocation in telecommunications systems.

CUONG DANH DO (Member, IEEE) received the B.Sc. degree in electronics and telecommunication from Vietnam National University, Hanoi, Vietnam, in 2004, the M.Eng. degree in electronics from Chungbuk National University, South Korea, in 2007, and the Ph.D. degree in electronics from the Cork Institute of Technology, Ireland, in 2012. He had more than four years working experience as a Postdoctoral Researcher at the University of Cambridge, U.K., in both fields MEMS and CMOS circuit for low-power sensor and timing application. He is currently an Assistant Professor at VinUniversity. His research interests including sensor for medical applications and sensor fusion.

KOK-SENG WONG (Member, IEEE) received the degree in computer science (software engineering) from the University of Malaya, Malaysia, the M.Sc. degree in information technology from the Malaysia University of Science and Technology (in collaboration with MIT), and the Ph.D. degree from Soongsil University, South Korea. In addition, he had more than 16 years of teaching record (computer science subjects) at universities in Malaysia, South Korea, Kazakhstan, and Vietnam. He is currently an Associate Professor at the College of Engineering and Computer Science, VinUniversity, Vietnam. His research interests include applied cryptography, secret sharing, information security, data privacy, and machine learning.