Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Communication efficiency optimization of federated learning for computing and network convergence of 6G networks*#

Yizhuo CAI[1], Bo LEI[†‡2], Qianying ZHAO[2], Jing PENG[3],
Min WEI[2], Yushun ZHANG[1], Xing ZHANG[†‡1]

[1]*Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[2]*Research Institute of China Telecom Co., Ltd., Beijing 102209, China*

[3]*Beijing Telecom, Beijing 100011, China*

[†]E-mail: leibo@chinatelecom.cn; zhangx@ieee.org

Received Feb. 28, 2023; Revision accepted Oct. 17, 2023; Crosschecked Apr. 26, 2024

Abstract: Federated learning effectively addresses issues such as data privacy by collaborating across participating devices to train global models. However, factors such as network topology and computing power of devices can affect its training or communication process in complex network environments. Computing and network convergence (CNC) of sixth-generation (6G) networks, a new network architecture and paradigm with computing-measurable, perceptible, distributable, dispatchable, and manageable capabilities, can effectively support federated learning training and improve its communication efficiency. By guiding the participating devices' training in federated learning based on business requirements, resource load, network conditions, and computing power of devices, CNC can reach this goal. In this paper, to improve the communication efficiency of federated learning in complex networks, we study the communication efficiency optimization methods of federated learning for CNC of 6G networks that give decisions on the training process for different network conditions and computing power of participating devices. The simulations address two architectures that exist for devices in federated learning and arrange devices to participate in training based on arithmetic power while achieving optimization of communication efficiency in the process of transferring model parameters. The results show that the methods we proposed can cope well with complex network situations, effectively balance the delay distribution of participating devices for local training, improve the communication efficiency during the transfer of model parameters, and improve the resource utilization in the network.

Key words: Computing and network convergence; Communication efficiency; Federated learning; Two architectures

https://doi.org/10.1631/FITEE.2300122 **CLC number:** TP393

1 Introduction

1.1 Context

In federated learning, machine learning models are trained collaboratively on multiple devices without the need for third parties to share training data. Therefore, federated learning with a distributed architecture can effectively solve the problem of data

‡ Corresponding authors

* Project supported by the National Natural Science Foundation of China (Nos. 62271062 and 62071063)

\# Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2300122) contains supplementary materials, which are available to authorized users

ORCID: Yizhuo CAI, https://orcid.org/0000-0002-6662-3767

silos. Moreover, compared with traditional machine learning, it reduces data transfer costs while ensuring fairness among participants. However, there are still many challenges to federated learning. The main challenges are manifested in several major aspects such as data privacy, communication consumption, devices' training latency, and statistical heterogeneity during the training process (Li T et al., 2020; Kairouz et al., 2021; Wahab et al., 2021). Among these, communication consumption is a huge challenge. In traditional federated learning, all clients simultaneously train their respective local models and then upload them to a predetermined central server. In this scenario, multiple copies of model data are simultaneously transmitted to the same server, and a large number of clients are present competing for communication resources with each other, leading to reduced communication efficiency and increased energy consumption. Meanwhile, due to variations in the computational resources of individual client devices, each client consumes uneven training periods, which diminishes the efficiency of federated learning. Traditional federated learning lacks resource scheduling for the system and struggles to address the aforementioned challenges, leading to significant shortcomings in real-world applications.

But today, computing and network convergence (CNC) of sixth-generation (6G) networks has been proposed as a new network architecture and paradigm. Its various characteristics can fit well with federated learning. In detail, CNC can rely on powerful heterogeneous resource scheduling capability and ubiquitous network connection to achieve real-time, flexible, and efficient resource allocation, while ensuring synchronous information sharing of the whole network. Its efficient information-sharing mechanism, high utilization of computing power resources of nodes in the network, and the ability to select and schedule network data traffic paths can effectively improve the data and information communication efficiency of participating clients in the federated learning and optimize the whole training process. This shows that the combination of federated learning and CNC is necessary and effective.

1.2 Related works

Solutions to the communication problem in federated learning focus on several aspects, such as reducing the amount of data for transmitting model parameters, reducing the frequency of communication, changing the type of communication, changing the aggregation method, using routing strategies, and employing knowledge distillation. As the model parameters or gradients passed in federated learning exist in the form of matrices, some scholars have used compression, quantization, and sparsification to reduce the amount of data while ensuring the validity of the data. Based on this idea they adopted two methods, i.e., random number seeding and sketch updating, to reduce the amount of data for the model parameters (Konečný et al., 2016).

Meanwhile, McMahan et al. (2017) reduced the communication frequency by performing a global aggregation only after the participating clients have performed local training several times. Moreover, using sampling to select some clients can reduce the communication consumption required to transfer the model parameters in each global training round (McMahan et al., 2017; Fraboni et al., 2021). However, Fraboni et al. (2021) divided the clients into different categories according to their local data distribution. Then, they sampled clients from different categories for each global training round, which is better than the former method.

There is also a problem of dropouts in communications challenges. Wu et al. (2021) proposed to change the form of communication into semi-asynchronous communication. Moreover, there is one way to select participating clients for each round using a proportional fair scheduling strategy to reduce the possibility of dropouts (Yang HH et al., 2020). Another effective way to address communication is to change the approach of global aggregation (Liu et al., 2020; Deng et al., 2021; So et al., 2021). So et al. (2021) focused on secure aggregation, while others introduced aggregation servers at the edge to reduce the communication consumption (Liu et al., 2020; Deng et al., 2021). Moreover, as the popularity of knowledge distillation grows, knowledge distillation is used for federated learning. Based on the idea of knowledge distillation, only the predicted logits values or the parameters of the student model are transmitted in federated learning (Li DL and Wang, 2019; He et al., 2020; Lin et al., 2020). This significantly reduces the amount of data to be transferred.

Dinh et al. (2021) derived the convergence analysis and resource allocation problem of federated learning in wireless networks. On the other hand, Chen et al. (2021) accelerated model convergence by reducing clients' data transmission errors of federated learning in a wireless environment. Qin et al. (2021) proposed specific scenarios for federated learning in wireless environments. Using a multi-layer aggregation approach to reduce the consumption of bandwidth resources in a specific tree structure can reduce the latency of federated learning model aggregation (Cao et al., 2021). In addition, a novel machine learning enabled wireless multi-hop federated learning framework can greatly mitigate the adverse impact of wireless communications on the federated learning performance metrics (Pinyoanuntapong et al., 2020).

Solely optimizing for either communication energy consumption or latency is insufficient to significantly improve the communication efficiency of federated learning. Yang ZH et al. (2022) provided a comprehensive overview of federated learning applications in 6G wireless networks. They suggested that features such as large-scale ultra-reliable low-latency communication, scalable architectures, and the ability to train and deploy human-centric services make federated learning highly promising in 6G. These point out that in 6G wireless networks, federated learning should be considered from multiple perspectives, leveraging its advantages, considering joint design of communication and computation, as well as asynchronous communication, to optimize federated learning performance, especially communication efficiency.

Therefore, aside from communication resource scheduling, the allocation of device's computational resources becomes a crucial aspect. Yang ZH et al. (2021) proposed an iterative algorithm for solving the minimum total energy consumption problem under delay constraints in wireless transmission networks. This algorithm provides closed-form solutions for time allocation, bandwidth allocation, power control, computation frequency, and learning accuracy at each step, demonstrating the feasibility of communication and computational resource scheduling in wireless networks. Sun W et al. (2023) leveraged the comparison between clients' computational energy consumption and communication energy consumption to make decisions regarding communication frequency in wireless networks and model depth in federated learning, resulting in reduced overall energy consumption for system training. This underscores the importance of perceiving and scheduling computational resources in the future development of federated learning.

However, traditional networks often lack sensitivity to information such as the computing power of underlying devices. Sun YK et al. (2022) investigated a future network paradigm capable of connecting distributed computing nodes. It is able to dynamically and timely sense user needs and multi-dimensional resources such as applications, network resources, computing power resources, and storage resources. The increased efficiency of federated learning communications can exactly take advantage of these properties.

1.3 Contributions

Inspired by Sun YK et al. (2022), we propose communication efficiency optimization of federated learning for CNC of 6G networks. The CNC of 6G networks is a new network architecture and paradigm with greater computing-measurable, perceptible, distributable, dispatchable, and manageable capabilities. Relying on the ability to sense the computing power of client devices in real time and synchronize various resource information in the network, CNC can flexibly schedule the clients in federated learning and make better decisions for the network topology and the allocation of the resources. This guides the training process of federated learning and improves communication efficiency.

The main contributions of this paper are summarized as follows:

1. We propose a federated learning system of CNC and its communication efficiency optimization methods. Combined with the available research, we divide CNC into different layers. When the layers interact with each other, CNC senses the network condition and client device resources and makes decisions to optimize the federated learning training process.

2. As two architectures exist for federated learning training, we adopt corresponding optimization methods. Both of them improve the performance of federated learning.

3. We focus on the computing power heterogeneity of client devices. By scheduling the clients in each global training round, the computing power difference between clients is balanced. Under a traditional architecture, the average delay difference of

the proposed system per training round is one-fifth of that of the federated averaging (FedAvg) algorithm (McMahan et al., 2017). Its maximum value is about 46.6% of that of FedAvg. Under the peer-to-peer architecture, by scheduling the computing power of the devices, the model accuracy of our system converges faster than that of the baseline with the same number of training times.

4. Federated learning communication efficiency optimization for CNC of 6G networks could improve the resource utilization of the whole network. During federated learning, optimization methods can update the information and schedule the resources about client devices or the network in real time. With this support, in the traditional architecture of federated learning, the transmission delay and energy consumption per round of model training are reduced by 46.96% and 19.38% respectively compared to FedAvg (McMahan et al., 2017). In the peer-to-peer architecture, the performance of the system regarding transmission problems varies with the network environment but is similar to, and sometimes better than the performance of the baseline.

2 Optimization system

2.1 Two architectures of federated learning

There are two training architectures in federated learning, as shown in Fig. 1. In each global training round, clients in the traditional architecture receive the global model from the server and train it with their local data. Then they will transmit the updated model to the server. After receiving the local models from all clients, the server performs an aggregation algorithm to obtain a new global model. Thus, federated learning moves on to the next round of global training, which does not end until the global model reaches a certain accuracy.

In the peer-to-peer architecture, federated learning is suitable for direct collaborative training among multiple data providers, where there is no central server. All interactions occur directly between the participating clients. The most rudimentary scenario entails the presence of merely two client devices. These two clients individually possess their own datasets and possess the capacity to engage in both model training and aggregation. Regarding scenarios involving multiple client participants, typically in each round, clients collectively designate one node as the aggregation node. All clients use their local data for training local models, which are subsequently submitted to the aggregation node for model aggregation. The new model is then propagated back to various client nodes, and this process continues, including the selection of a new aggregation node for the subsequent training round.

Moreover, due to the inherent connectivity between client nodes, considering the training methodologies of only two clients, there exists another feasible approach under the peer-to-peer architecture, as illustrated in Fig. 1b. Each round of global training involves selecting a subset of clients and establishing the transmission order between clients. The global model is initially sent to the starting node, and according to a predefined strategy, it is trained and transmitted to the final node. When all selected nodes have completed their training, a round of global training is formed. At each individual node, upon receiving the encrypted model parameters from the previous client, the recipient node performs decryption, followed by local training operations to update the model. Subsequently, the model is encrypted once again before being transmitted to the designated next client. After multiple rounds of global training, the final model is obtained.

2.2 Communication efficiency optimization system for CNC of 6G networks

Based on the two architectures, we propose communication efficiency optimization for CNC of 6G networks. The communication efficiency in federated learning training can be improved by exploiting computing power resource modeling, information synchronization, and scheduling capabilities of CNC. Fig. 2 shows its architecture.

We stratify CNC into an infrastructure layer, a resource pooling layer, a resource information announcement layer, a computing scheduling optimization layer, a service layer, and an orchestration and management layer. As devices of the infrastructure layer, the aggregation servers and client devices involved in federated learning are scheduled and controlled by CNC. Equipment in the resource pooling layer models the network resources, computing power resources, etc., of the underlying devices. The information announcement layer, on the other hand, consists of specialized equipment. Downward, it

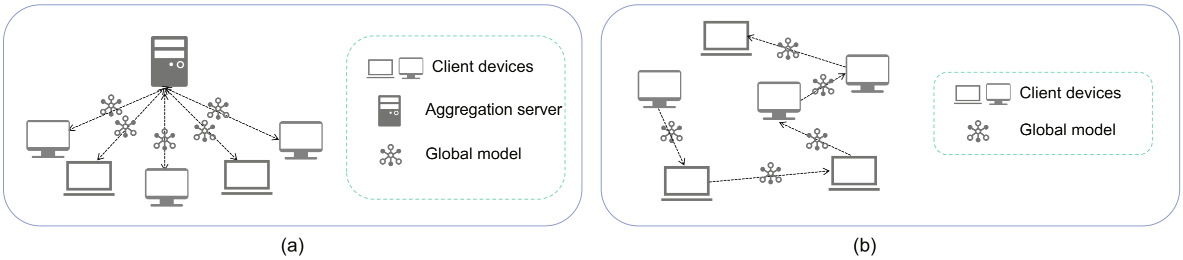

Fig. 1 Two architectures of federated learning: (a) traditional architecture; (b) peer-to-peer architecture

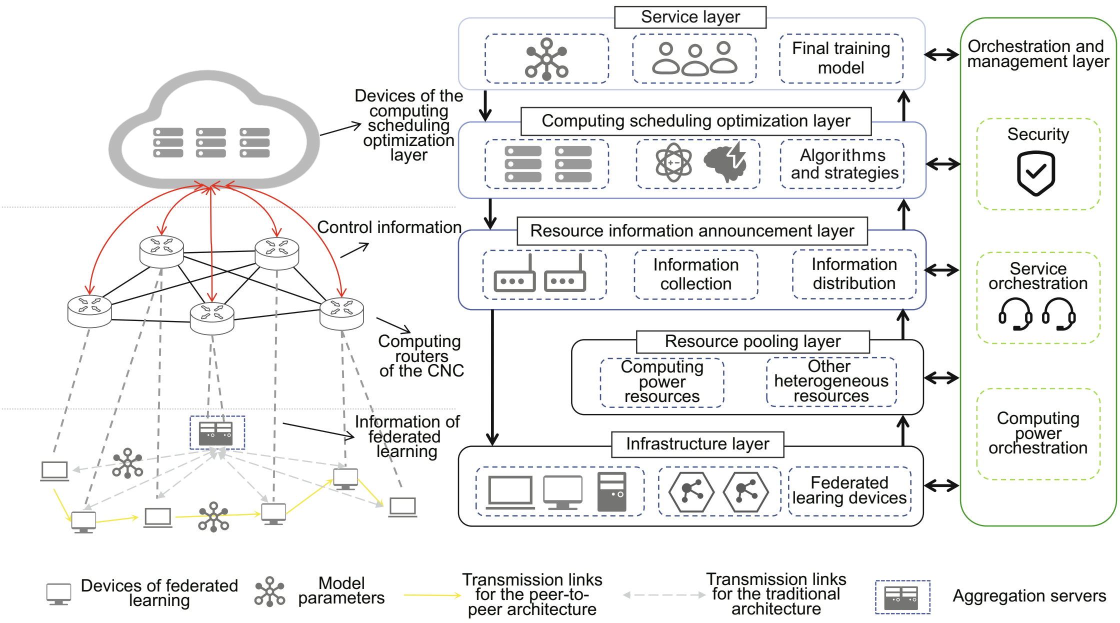

Fig. 2 Communication efficiency optimization system for computing and network convergence (CNC) of 6G networks (References to color refer to the online version of this figure)

collects various sorts of information from the participating devices or publishes training strategies. Upward, it forwards information about the clients to the scheduling optimization layer or obtains decision information from the upper layer. The computing scheduling optimization layer is responsible for optimizing the federated learning scheduling algorithms and topological decisions based on the information from the underlying layer. Then, it outputs control information for federated learning training in real time. The orchestration and management layer has control of the entire system of CNC. It is responsible for orchestrating and scheduling the various resources used in federal learning, as well as managing the various devices in the other layers. The security and service orchestration are also under its control.

The core of the whole structure is the coordinated operation of the upper equipment in CNC, which optimizes the training process of federated learning based on the network, computing power resources, and other information. Equipment of the computing scheduling optimization layer, like cloud servers, exchanges the decision information of federated learning through the red lines as shown in Fig. 2. The routers of the resource information announcement layer are responsible for synchronizing training information. They connect the whole network. Moreover, the infrastructure layer is divided into two cases. The traditional architecture requires the aggregation of server clusters and client devices to participate in training, and the logical transmission topology is shown by the gray dashed line. In the peer-to-peer architecture, only client devices are required to participate in training. The logical transmission topology of model parameters is given by the yellow line in Fig. 2. Fig. S2 in the supplementary materials provides an elaborate representation of the system's operational workflow.

3 Mathematics problems

The communication efficiency optimization of federated learning for CNC of 6G networks contains optimization for several problems. This section introduces specific mathematical problems based on the two architectures of federated learning.

3.1 Convergence problem

Assume that there are U participating clients. For client i, we use D_i to denote the original data and $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{in_i}]$ to denote the feature vector in D_i, that is, the input of the model. Here, n_i is the amount of data of client i. $\boldsymbol{Y}_i = [y_{i1}, y_{i2}, \ldots, y_{in_i}]$ indicates the corresponding labels of client i for the model output, and $\boldsymbol{w}_i$ denotes the local model parameters of client i. Federated learning is a process of continuously training models with data and iteratively aggregating them to find the best global model $\boldsymbol{w}^*$. As the number of iterations increases, the global model gradually converges, approaching $\boldsymbol{w}^*$. The goal of our optimization is to find $\boldsymbol{w}$ so that it satisfies the following:

$$\min_{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_U} \frac{1}{K} \sum_{i=1}^{U} \sum_{n=1}^{n_i} f(\boldsymbol{w}_i, \boldsymbol{x}_{in}, y_{in}), \tag{1}$$

where $K = \sum_{i=1}^{U} n_i = \sum_{i=1}^{U} |D_i|$ and $\boldsymbol{w}_1 = \boldsymbol{w}_2 = \ldots = \boldsymbol{w}_U = \boldsymbol{w}$. For the loss function $f(\cdot)$, it can be defined according to the actual situation, such as the mean squared difference loss function.

3.2 Transmission problem

The network topology and data transmission types of the clients under different architectures of federated learning determine the respective transmission problems.

3.2.1 Transmission problem under a traditional architecture

In this study, we design a traditional architecture in which clients of federated learning are in a wireless network environment. Communication between the client and the server uses the orthogonal frequency division multiple access (OFDMA) transmission technology. In line with the ideas presented in Chen et al. (2021), we consider the minimum schedulable resource to be the resource block (RB). Each client occupies one RB. Based on this, the uplink rate at which the user transmits data to the central server can be expressed as follows:

$$r_i^{\mathrm{U}} = B^{\mathrm{U}} \mathbb{E}_{h_i} \left(\log_2 \left(1 + \frac{P_i h_i}{I_k + B^{\mathrm{U}} N_0} \right) \right), \tag{2}$$

where B^{U} is the bandwidth of each RB, P_i is the transmitting power of client i, $h_i = o_i d_i^{-2}$ is the channel gain between client i and the central server, d_i is the distance between client i and the central server, o_i denotes the Rayleigh fading parameter, $\mathbb{E}_{h_i}(\cdot)$ is the expectation function for h_i, N_0 is the noise power spectral density, and I_k represents the interference arising from users using the same RB k under different base stations. When the system operates in different environments, the interference generated varies accordingly. Additionally, the interference received by users undergoes changes over time.

We assume that the consumption originates only from the data transmission process of the client device. So, the transmission problem needs only to consider the uplink transmission delay and energy consumption of the clients:

$$l_i^{\mathrm{U}} = Z(\boldsymbol{w}_i) / r_i^{\mathrm{U}}, \tag{3}$$

$$e_i = P_i l_i^{\mathrm{U}}, \tag{4}$$

where l_i^{U} denotes the transmission delay, e_i is the transmission energy consumption, and $Z(\boldsymbol{w}_i)$ is the amount of data for the model parameters.

In each global training round, a portion of U clients will be selected as the training set S_{t}. Then the transmission problem can be interpreted as finding the best RB allocation method that satisfies the following expression:

$$\min \sum_{i \in S_{\mathrm{t}}} e_i, \tag{5}$$

or

$$\min \left(\max \left(\left\{ l_i^{\mathrm{U}} \mid i \in S_{\mathrm{t}} \right\} \right) \right). \tag{6}$$

3.2.2 Transmission problem under the peer-to-peer architecture

In the peer-to-peer architecture, the transmission formula in the wireless environment is not applicable because there is no central server. We consider that all clients are connected to each other in the network. The transmission consumption or delay between clients i and j is denoted by cost_{ij}. The transmission problem becomes as follows:

$$\min\left(\sum_{(i,j)\in\mathrm{trace\_path}} \mathrm{cost}_{ij}\right), \tag{7}$$

where trace_path is the transmission order for the selected clients.

3.3 Local training related problem

For participating clients, the local training delay is given according to the computing power of their devices. Here, c_i is the maximum computing power that can be output by client i, $|D_i|$ is the amount of local data, and epoch_local represents the number of local training times per global training round. With the same model for each client, the local training delay can be roughly given as follows:

$$t_i = \alpha \cdot \mathrm{epoch\_local} \cdot |D_i| / c_i, \tag{8}$$

where α is the conversion factor. For federated learning, the local training problem can be solved by finding a suitable subset S_t satisfying the following expression:

$$t_{\max} - t_{\min} < \varepsilon, \tag{9}$$

where $t_{\max} = \max\{t_i \mid i \in S_\mathrm{t}\}$, $t_{\min} = \min\{t_i \mid i \in S_\mathrm{t}\}$, and ε indicates an acceptable time difference.

4 Communication efficiency optimization of federated learning for CNC of 6G networks

According to the network topology and user requirements, we provide the optimization methods under the two architectures of federated learning, as described in the following text.

4.1 Optimization under the traditional architecture

This optimization method uses all layers of CNC to operate efficiently. Based on FedAvg, we focus on the computing power problem and the model transfer problem. The system gives optimization based on these two points.

In federated learning, the clients register their local devices through the platform of CNC to obtain a great global model. The devices of them are then scheduled by the devices in the orchestration and management layer for unified orchestration. Finally, the devices of clients operate as the infrastructure layer devices that support federated learning.

If the local data distribution of clients is similar, the choice of S_t has little effect on Eq. (1). However, its effect on inequality (9) is large. In our proposed algorithm, under the CNC of 6G networks, resource pooling layer devices will model the heterogeneous resources of client devices, and upper-level equipment will group them based on the computing power. In this case, the clients selected for each global training round have similar number of local training times and participating users can have a better experience. In each global training round, the resource pooling layer facilities obtain resource information of all devices and perform resource modeling. Devices of the resource information announcement layer forward these messages to devices in the computing scheduling optimization layer. Then, the decision about S_t will be made. The steps are given in Algorithm 1.

At the same time, we draw on the idea of Chen et al. (2021). By considering the consistent transmitting power of the clients, we make the optimization problem of communication performance an RB allocation problem. Devices in the computing scheduling optimization layer of CNC execute Algorithm 1 to obtain S_t and then optimize the allocation strategy of RBs using the Hungarian algorithm. Different

Algorithm 1 Client scheduling strategy based on computing power

Input: the number of participating clients (U), the amount of data for peer client i ($|D_i|$), the maximum computing power that can be output by client i (c_i), number of local training times per global training round (epoch_local), and conversion factor (α)

Output: the set of participating clients for each global training round S_t

```
for i = 1 to U do
    t_i = α · epoch_local · |D_i| / c_i
end for
Sort {t_i | i = 1, 2, ..., U} by the descending order
Divide U clients into m parts, and each part is denoted by U_k, k = 1, 2, ..., m
Sample clients from m parts, P_k = N_k / Σ_{k=1}^{m} N_k, where N_k = Σ_{i∈U_k} |D_i|
Sample n clients from U_k as S_t, P_i = |D_i| / N_k
return S_t
```

clients that are allocated different resources transfer data with different consumptions. We construct the consumption matrix with the energy consumption generated by client i transmitting data on RB k. The Hungarian algorithm often solves for a good solution about Eq. (5).

Later, devices in the computing scheduling optimization layer hand the selected client information and the allocation policy of RBs to resource information announcement layer devices for forwarding. The clients in federated learning can share transmission performance, local training performance, etc., of other devices predicted by CNC.

Orchestration and management layer devices are based on algorithms to schedule and orchestrate federated learning devices and various resources. Then, the participating clients receive the algorithm information output from CNC to perform federated learning to obtain the converged global model by using the idea of weighted average. After getting the new model, the system moves to the next iteration.

4.2 Optimization under the peer-to-peer architecture

Similarly, client devices operate as the infrastructure layer devices to support federated learning. Algorithm 2 shows the entire process.

Under the peer-to-peer architecture, all clients are trained once as a global training session. Before each global training round, devices in the resource pooling layer and resource information announcement layer cooperate to obtain information on computing power resources and network conditions of clients. Devices in the computing scheduling optimization layer assign subsets $S_{\rm te}$ based on c_i and D_i, where $e=1,2,\ldots,E$ (E is the total number of subcategories of client devices determined based on the computing power and data information). For each $S_{\rm te}$, the sum of local training delay is similar. Then, they abstract the consumption matrix $\boldsymbol{G}$ according to the network connectivity.

For clients in $S_{\rm te}$, the shortest transmission path over all clients needs to be found to solve Eq. (7). With the powerful support of the CNC, we can generate the submatrix of $S_{\rm te}$ from $\boldsymbol{G}$ and then solve it by the optimal transmission path selection strategy.

After outputting the transmission path for each subset $S_{\rm te}$, devices of the resource information announcement layer forward the transmission policy to

Algorithm 2 Optimization under the peer-to-peer architecture

Input: the number of participating clients (U), the data for peer client i (D_i), total number of global training rounds (T), and data captured in one training session (B)

Output: the final global model $\boldsymbol{w}$

Participating clients and aggregation servers join the CNC of 6G networks as node devices
for $t=1$ to T **do**
  Devices in each layer of the CNC collaborate to divide the participating clients into E parts with $S_{\rm te}$ being the subset ($e=1,2,\ldots,E$)
  Calculate the transmission path for each subset $S_{\rm te}$ according to the optimal transmission path selection strategy
  Forward the transmission strategy and initial model
  for client i in trace_path of $S_{\rm te}$ parallelly **do**
    if client i is the first client **then**
      Receive model $\boldsymbol{w}$ from the devices of the CNC, $\boldsymbol{w}_i=\boldsymbol{w}$
    else
      Receive $\boldsymbol{w}_{S_{\rm te}}$ from the previous client, $\boldsymbol{w}_i=\boldsymbol{w}_{S_{\rm te}}$
    end if
    for B in D_i **do**
      for $(\boldsymbol{x},y)$ in batch B **do**
        $\boldsymbol{w}_i=\boldsymbol{w}_i-\eta\cdot\nabla f\left(\boldsymbol{w}_i,\boldsymbol{x},y\right)$, where η is the learning rate
      end for
    end for
    $\boldsymbol{w}_{S_{\rm te}}=\boldsymbol{w}_i$
    Send $\boldsymbol{w}_{S_{\rm te}}$ to the devices of the CNC if i is the last client; otherwise, send $\boldsymbol{w}_{S_{\rm te}}$ to the next client in $S_{\rm te}$
  end for
  $\boldsymbol{w}=\sum_{e=1}^{E}\frac{N_{\rm te}}{\sum_{e=1}^{E}N_{\rm te}}\boldsymbol{w}_{S_{\rm te}}$, $N_{\rm te}=\sum_{i\in S_{\rm te}}|D_i|$
end for
return $\boldsymbol{w}$

all clients and pass the model to each initial client. The clients receive the information and start training and then transmit the parameters of the trained global model according to the policy. After each part is trained, the E sub-models are aggregated by the specific devices in the computing scheduling optimization layer of the CNC to obtain the new global model and perform the next round of training. Until the global model reaches a certain accuracy, other devices from CNC, for instance, routers of the resource information announcement layer, broadcast the final global model to all clients.

5 Simulation results and analysis

In our simulations, we used the MNIST dataset, which can be independent and identically distributed (IID) or non-independent and identically distributed (non-IID) after processing. We cut the datasets equally based on the total number of clients to obtain the clients with local data. Then, we used a simple neural network as the training model for training.

5.1 Simulations under the traditional architecture

5.1.1 Parameter settings of the simulation environment

In simulations under the traditional architecture, we set up different cases to compare the performance of our approach with that of FedAvg (McMahan et al., 2017). For convenience, Table 1 records the parameter settings for different cases. Detailed parameter values are given in Table S1 in the supplementary materials.

Table 1 Parameter settings for different cases

Case	Parameter setting
Pr 1	U: 100, cfraction: 0.1, epoch_local: 1
Pr 2	U: 100, cfraction: 0.1, epoch_local: 5
Pr 3	U: 100, cfraction: 0.2, epoch_local: 1
Pr 4	U: 100, cfraction: 0.2, epoch_local: 5
Pr 5	U: 60, cfraction: 0.1, epoch_local: 1
Pr 6	U: 60, cfraction: 0.1, epoch_local: 5

U: total number of participating clients; cfraction: sampling ratio for each global training round; epoch_local: number of local training times per global training round

5.1.2 Simulation results and analysis

A significant challenge persistently encountered in federated learning is data heterogeneity. Therefore, in our simulations, we typically configure clients' data distributions for the two scenarios. Under the IID dataset, each client's data label types are evenly distributed, resulting in stable accuracy curves for the global model with a high convergence upper bound. In contrast, under the non-IID dataset, where clients possess significantly different data category distributions, the gradients' update direction is highly correlated with data distribution when local model training is updated. As a consequence, different sub-models do not update toward the global optimal direction but rather shift toward the local optimal direction that fits their own data. This leads to a slower increase in global model accuracy after aggregation, a lower upper bound, and significant fluctuations in the accuracy curve. In our simulations, the curves under both data distributions also exhibited the same patterns or regularities.

Fig. 3 shows the global model convergence curve of the CNC optimization method under the traditional architecture. In Fig. 3a, comparing the optimization of CNC in Pr 1 with Pr 2, the accuracy improvement is to be accelerated when the number of local training times is increased. However, this is at the expense of more local training times. Similarly, comparing Pr 1 with Pr 3, when the sampling ratio

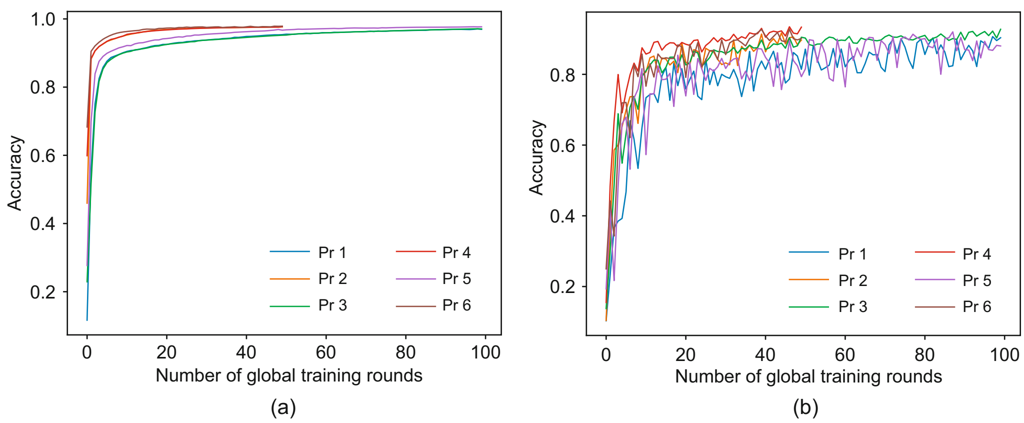

Fig. 3 Global model test accuracy with communication efficiency optimization for CNC under different cases: (a) IID; (b) non-IID (CNC: computing and network convergence; IID: independent and identically distributed. References to color refer to the online version of this figure)

increases, the global model accuracy in Pr 3 rises faster than in Pr 1 as the number of global training times increases. This is because under the setting of Pr 3, there are more clients involved in training. The global model gains more gradient information in each global training round. Under settings of Pr 5, although there are fewer sampled clients than in Pr 1, there is a higher amount of local data in each client. Richer information about the gradient of each global training results in a higher convergence rate than in Pr 1.

However, in Fig. 3b, the curves exhibit some distinct patterns. The performances under cases Pr 5 and Pr 6 are poorer than that of the others. Under the non-IID dataset, both models in Pr 5 and Pr 6 show signs of overfitting. Furthermore, when comparing Pr 2 and Pr 4, or Pr 1 and Pr 3, under the IID condition, the two curves converge similarly. However, under the non-IID condition, the higher sampling ratio in Pr 4 or Pr 3 demonstrates superior performance. This is because a higher sampling ratio, involving more models in aggregation, allows the global model to acquire richer gradient information, resulting in faster convergence. Under the non-IID dataset, where local data distributions are uneven, an increased participation in model aggregation helps balance the disparities among different sub-models. Conversely, under the IID dataset, the impact is relatively minor. Hence, the differences between the curves are more pronounced in Fig. 3b.

Fig. 4 shows the variation of the federated learning communication performance metrics under our proposed algorithm. Similar to the aforementioned description, accelerating the increase of model accuracy with increasing the number of global training rounds is achieved at the expense of some communication performance. Increasing the sampling ratio will exacerbate the local training energy consumption and the energy consumption for transmitting model parameters. Moreover, increasing the number of local training times per global training round will result in higher local training delays. With the optimization of federated learning for CNC, we can adjust the parameter set to achieve the results we want, depending on the objectives of the federated learning training.

Compared with other algorithms, the performance of our proposed algorithm is significant. Our proposed approach initially schedules the clients, selecting them for each round based on their computing power information. This reduces the time delay disparity between client local training during each global training round, consequently reducing the number of local training times per round. Additionally, the communication resources allocatable to clients are intelligently scheduled by CNC, thereby reducing client communication latency and energy consumption. As shown in Fig. 5, compared to FedAvg (McMahan et al., 2017), our method's

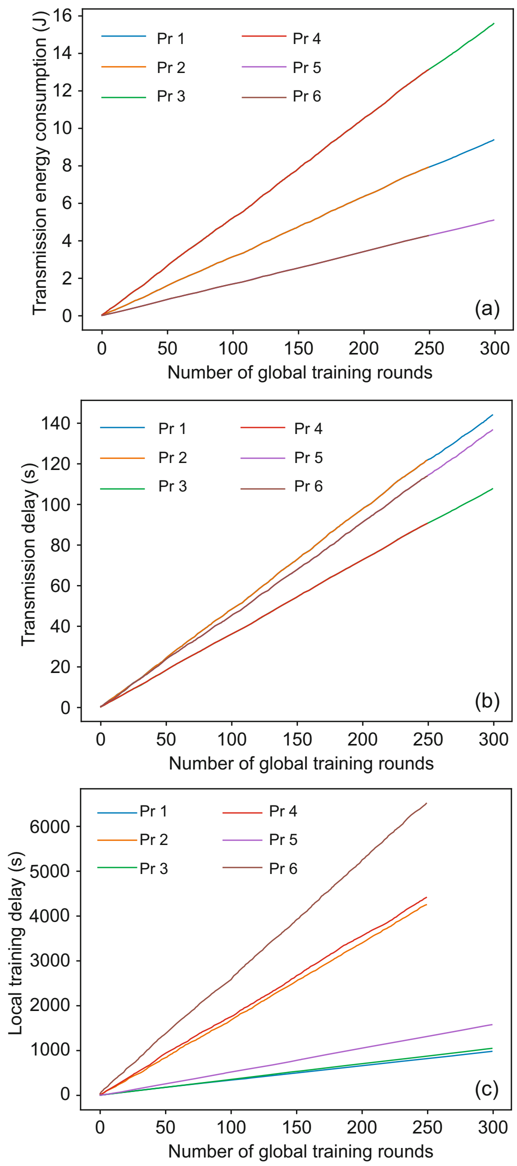

Fig. 4 Communication performance with communication efficiency optimization for computing and network convergence (CNC) under different cases: (a) transmission energy consumption; (b) transmission delay; (c) local training delay (References to color refer to the online version of this figure)

advantages are readily evident in these three metrics. In three settings of Pr 1, Pr 2, and Pr 3, optimization of CNC requires lower local training delay, lower delay to transmit the model, and lower energy consumption for the whole training. Specifically, taking the setting of Pr 1 as a reference, our method can save 19.38% of transmission energy consumption, 46.96% of transmission delay, and 28.41% of local training delay in each global training round compared to FedAvg.

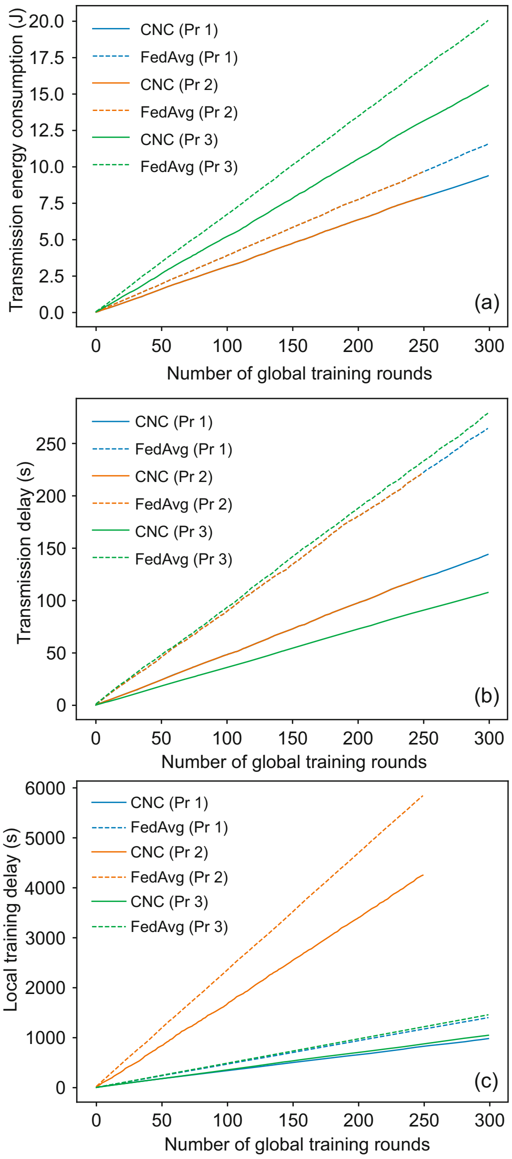

Fig. 5 Communication performance comparison under the two algorithms: (a) transmission energy consumption; (b) transmission delay; (c) local training delay (CNC: computing and network convergence; FedAvg: federated averaging. References to color refer to the online version of this figure)

For a further comparison, we plot the variation curves of global model accuracy. Communication consumption includes local training delay, transmission energy consumption, and transmission delay for the three cases based on the non-IID and IID datasets, as shown in Fig. 6.

In federated learning, energy consumption is highly dependent on the number of clients participating in each round and the communication frequency. In Figs. 6a and 6d, under the Pr 2 setting, lower client number and lower communication frequency result in lower energy consumption compared to other cases. Additionally, more rounds of local training can accelerate both local and global model convergence, ultimately leading to the best performance indicated by the orange curves. However, when comparing Pr 1 to Pr 3, under the IID dataset, Pr 1 with fewer clients participating in each round (represented by the blue lines) converges faster, whereas in the non-IID dataset, Pr 3 demonstrates better performance. This demonstrates that while resource scheduling in federated learning training can optimize communication efficiency, addressing the issue of data heterogeneity remains a pressing concern.

Similarly, when using transmission delay as the metric, as shown by the orange lines for Pr 2 in Figs. 6b and 6e, the reduced communication frequency with clients results in lower accumulated transmission delay over multiple global training rounds, leading to better performance. Regarding the other curves, Pr 3 outperforms Pr 1 due to a larger client base, allowing our method to achieve more optimal resource allocation, and the global model is less susceptible to the effects of data heterogeneity. Moreover, the non-IID dataset can impact model training, especially under the settings of Pr 2. In this scenario, involving more clients in each global training round effectively mitigates this issue without adding unnecessary communication latency, as demonstrated by the green line of Pr 3.

Finally, when considering local training delay as the metric, an increase in the number of local training times implies a proportional increase in local training delay. In Figs. 6c and 6f, Pr 2 (represented by the orange lines) exhibits the poorest convergence performance, while the green curves with a higher sampling ratio perform the best. This difference becomes more pronounced when data distribution is non-IID.

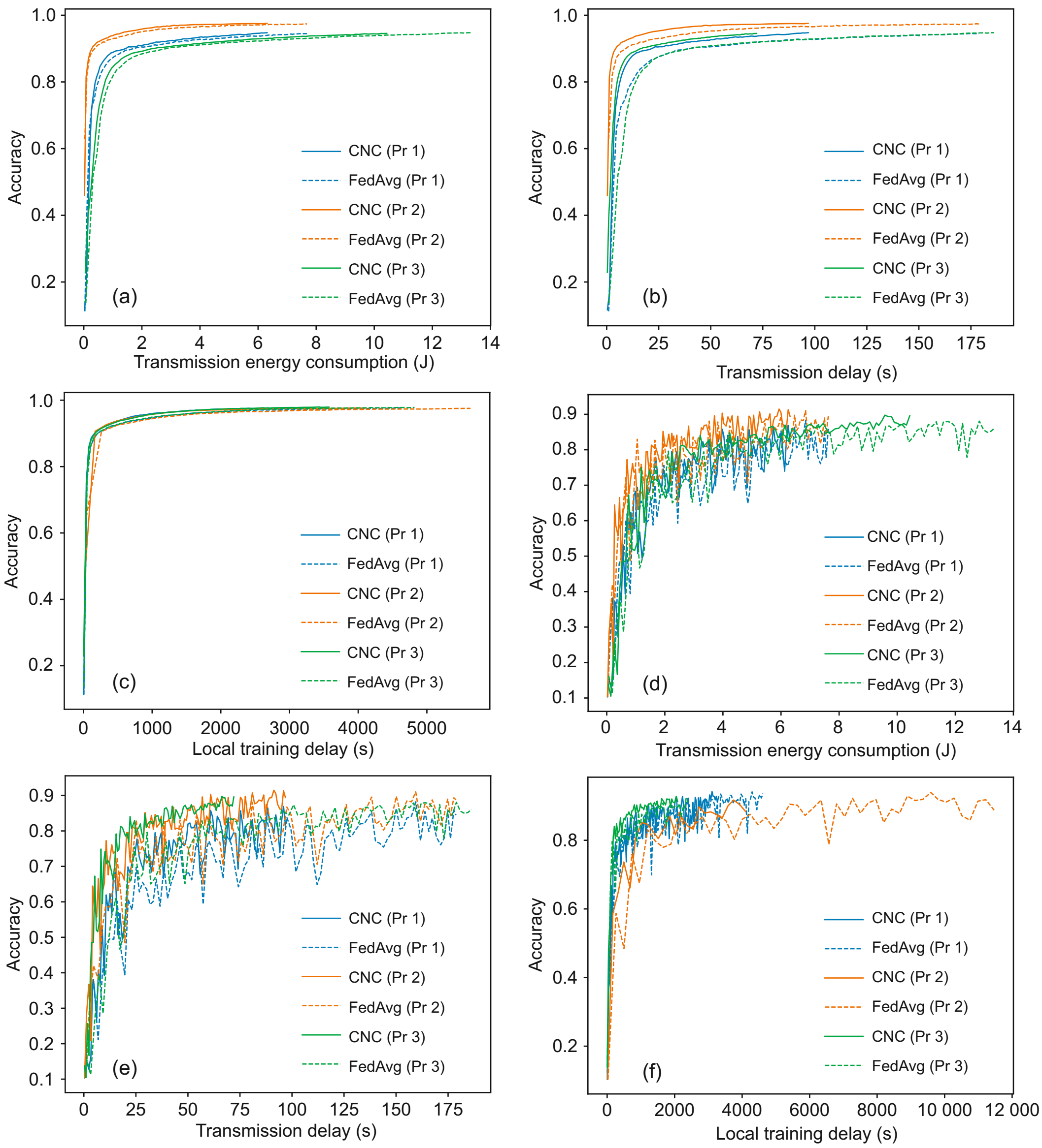

Fig. 6 Test accuracy comparison under the two algorithms: (a) transmission energy consumption (IID); (b) transmission delay (IID); (c) local training delay (IID); (d) transmission energy consumption (non-IID); (e) transmission delay (non-IID); (f) local training delay (non-IID) (CNC: computing and network convergence; FedAvg: federated averaging; IID: independent and identically distributed. References to color refer to the online version of this figure)

Taking into account the entirety of Fig. 6 and comparing the optimization of federated learning for CNC with FedAvg, our proposed method consistently demonstrates superior performance. It does not exhibit weakness in the presence of data heterogeneity, sampling ratios, or communication frequencies compared to FedAvg. Under the same allocation of communication and computational resources, our method is capable of training a superior global model. CNC integrates resources across the entire network, optimizing scheduling effectively from the perspectives of communication and computing power. This improves the communication efficiency and model performance in federated learning.

Fig. 7 provides a comparison of local training delay disparities under the Pr 1 configuration, presenting our method's performance in the form of box plots. After scheduling clients based on their computing power information for each round of training, the mean difference in the number of local training

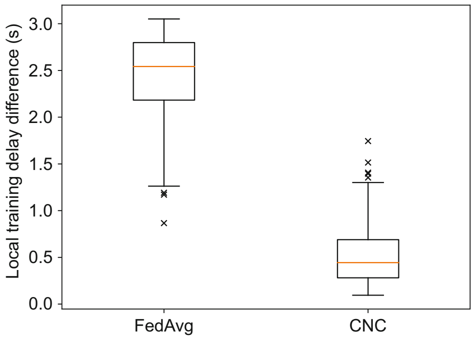

Fig. 7 Local training delay differences in the two algorithms under the setting of Pr 1 (CNC: computing and network convergence; FedAvg: federated averaging)

times among the client clusters simultaneously participating in each global training round is lower. Additionally, the boxes in the figure are narrower, indicating more stable delay disparities and higher efficiency.

5.2 Simulations under the peer-to-peer architecture

5.2.1 Parameter settings of the simulation environment

Due to the chained transmission of model parameters in the peer-to-peer architecture, the communication time cost is huge. Time delay includes local training time and transmission time. Based on this, we designed two experiments. Experiment 1 compares mainly the effect of scheduling based on computing power under optimization of CNC, while experiment 2 compares mainly the performance of different path selection strategies under a few points. The content of experiment 2 can be found in the supplementary materials.

In the first experiment, we designed the transmission consumption matrix of 20 clients. The numerical value represents the relative size, and the local training delay follows the assumption under the traditional architecture. All the transmission paths selected are based on the optimal transmission path selection strategy. Four settings are simulated in the experiment, as follows:

1. According to Algorithm 2, in each global training round, 20 clients are divided into four parts on average;

2. In each global training round, 20 clients are divided into two parts on average;

3. In each global training round, 15 clients are randomly selected and a global model is output;

4. In each global training round, 20 clients are selected and a global model is output.

5.2.2 Simulation results and analysis

The results of the first experiment are analyzed first. As shown in Figs. 8a and 8c, under the peer-to-peer architecture our proposed optimization method has a higher convergence rate. The larger the number of subsets is, the better the method performs. In a more detailed explanation, the greater number of subsets implies a reduction in the significant local training delay accumulation caused by chained transmissions, and the reduction in the model convergence rate is relatively minor. Therefore, it exhibits better performance. Under the non-IID dataset, training multiple sub-models with more subsets can also prevent the global model from experiencing overfitting, which would otherwise affect model accuracy.

In Figs. 8b and 8d, regarding the performance comparison related to transmission consumption, our proposed method exhibits slightly inferior performance, but the difference is not significant. When determining the order of transmission paths, the baseline method relies on a network topology with the presence of more clients, offering a broader perspective for optimized decision-making. In contrast, our method operates on a subset for determining the transmission paths, which, within a narrower network perspective, can sometimes lead to local optima and slightly higher transmission cost. All these four simulation methods use the output of the optimal transmission path selection strategy. The disadvantages in terms of transmission consumption are to be expected.

Overall, our proposed optimization method greatly reduces the local training delay during training without excessive transmission or latency consumption. But in reality, the cost of local training delay is often higher than transmission consumption. From this point of view, our proposed communication efficiency optimization of federated learning for CNC is more practical.

In addition, optimization of CNC under the peer-to-peer architecture can output models with higher accuracy than the traditional architecture. The average accuracy of the former is close to 92%, while that of the latter is as high as about 97%.

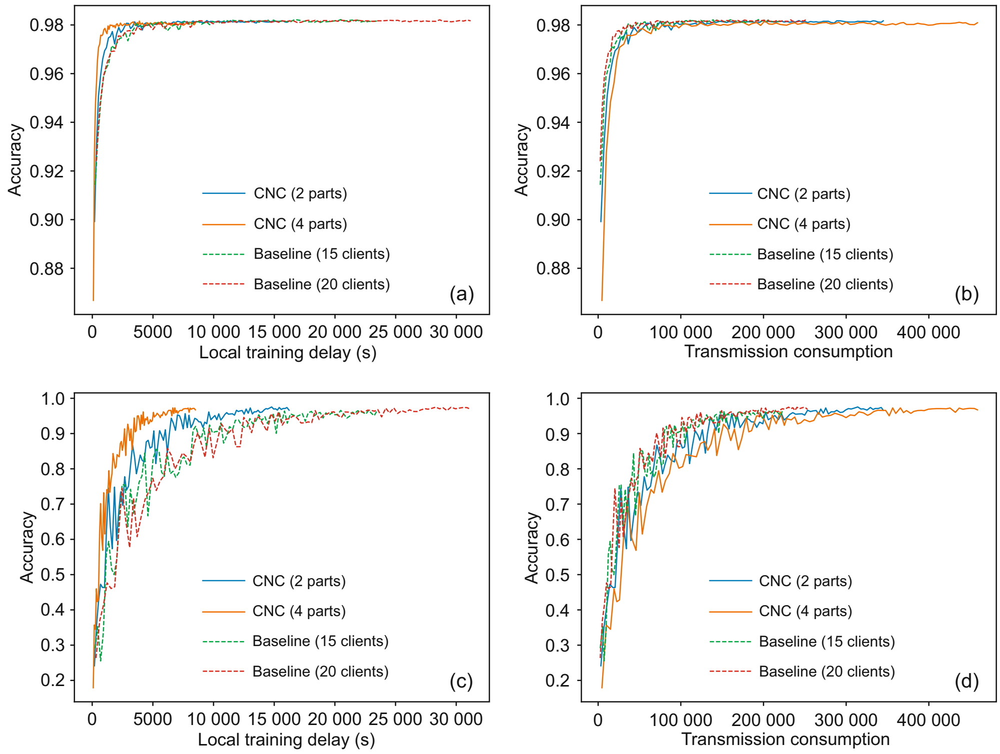

Fig. 8 Test accuracy comparison under the peer-to-peer architecture in experiment 1: (a) local training delay (IID); (b) transmission consumption (IID); (c) local training delay (non-IID); (d) transmission consumption (non-IID) (CNC: computing and network convergence; IID: independent and identically distributed. References to color refer to the online version of this figure)

Finally, we qualitatively study the variation of the average global training delay with the number of clients in the peer-to-peer architecture. Fig. 9 shows the result. Compared to other methods, optimization of federated learning for CNC of 6G networks can guarantee a lower delay rise rate. This ensures very high communication efficiency during federated training.

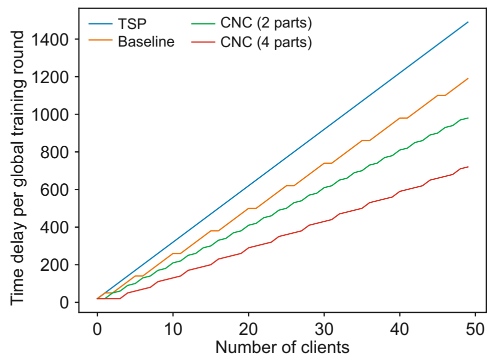

Fig. 9 Variation of the average global training delay with the number of clients (CNC: computing and network convergence; TSP: traveling salesman problem. References to color refer to the online version of this figure)

6 Conclusions

CNC of 6G networks is a new network architecture and paradigm. Federated learning can show better performance with its support, especially in terms of communication efficiency. In this case, we propose communication efficiency optimization of federated learning for CNC of 6G networks. As the simulation results shown, it has great potential to improve the federated learning process, especially in balancing heterogeneous computing power and improving communication efficiency.

Contributors

Yizhuo CAI designed the research. Yizhuo CAI and Yushun ZHANG processed the data. Yizhuo CAI and Jing PENG drafted the paper. Qianying ZHAO and Min WEI helped organize the paper. Bo LEI and Xing ZHANG revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

References

Cao QM, Zhang X, Zhang YS, et al., 2021. Layered model aggregation based federated learning in mobile edge networks. IEEE/CIC Int Conf on Communications in China, p.1-6. https://doi.org/10.1109/ICCC52777.2021.9580403

Chen MZ, Yang ZH, Saad W, et al., 2021. A joint learning and communications framework for federated learning over wireless networks. *IEEE Trans Wirel Commun*, 20(1):269-283. https://doi.org/10.1109/TWC.2020.3024629

Deng YH, Lyu F, Ren J, et al., 2021. Share: shaping data distribution at edge for communication-efficient hierarchical federated learning. IEEE 41st Int Conf on Distributed Computing Systems, p.24-34. https://doi.org/10.1109/ICDCS51616.2021.00012

Dinh CT, Tran NH, Nguyen MNH, et al., 2021. Federated learning over wireless networks: convergence analysis and resource allocation. *IEEE/ACM Trans Netw*, 29(1):398-409. https://doi.org/10.1109/TNET.2020.3035770

Fraboni Y, Vidal R, Kameni L, et al., 2021. Clustered sampling: low-variance and improved representativity for clients selection in federated learning. Proc 38th Int Conf on Machine Learning, p.3407-3416.

He CY, Annavaram M, Avestimehr S, 2020. Group knowledge transfer: federated learning of large CNNs at the edge. Proc 34th Int Conf on Neural Information Processing Systems, p.14068-14080.

Kairouz P, McMahan HB, Avent B, et al., 2021. Advances and open problems in federated learning. *Found Trends Mach Learn*, 14(1-2):1-210. https://doi.org/10.1561/2200000083

Konečný J, McMahan HB, Yu FX, et al., 2016. Federated learning: strategies for improving communication efficiency. https://doi.org/10.48550/arXiv.1610.05492

Li DL, Wang JP, 2019. FedMD: heterogenous federated learning via model distillation. https://doi.org/10.48550/arXiv.1910.03581

Li T, Sahu AK, Talwalkar A, et al., 2020. Federated learning: challenges, methods, and future directions. *IEEE Signal Process Mag*, 37(3):50-60. https://doi.org/10.1109/MSP.2020.2975749

Lin T, Kong LJ, Stich SU, et al., 2020. Ensemble distillation for robust model fusion in federated learning. Proc 34th Int Conf on Neural Information Processing Systems, p.2351-2363.

Liu LM, Zhang J, Song SH, et al., 2020. Client-edge-cloud hierarchical federated learning. IEEE Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC40277.2020.9148862

McMahan HB, Moore E, Ramage D, et al., 2017. Communication-efficient learning of deep networks from decentralized data. Proc 20th Int Conf on Artificial Intelligence and Statistics, p.1273-1282.

Pinyoanuntapong P, Janakaraj P, Wang P, et al., 2020. FedAir: towards multi-hop federated learning over-the-air. IEEE 21st Int Workshop on Signal Processing Advances in Wireless Communications, p.1-5. https://doi.org/10.1109/SPAWC48557.2020.9154266

Qin ZJ, Li GY, Ye H, 2021. Federated learning and wireless communications. *IEEE Wirel Commun*, 28(5):134-140. https://doi.org/10.1109/MWC.011.2000501

So J, Güler B, Avestimehr AS, 2021. Turbo-aggregate: breaking the quadratic aggregation barrier in secure federated learning. *IEEE J Sel Areas Inform Theory*, 2(1):479-489. https://doi.org/10.1109/JSAIT.2021.3054610

Sun W, Li ZJ, Wang QBJ, et al., 2023. FedTAR: task and resource-aware federated learning for wireless computing power networks. *IEEE Int Things J*, 10(5):4257-4270. https://doi.org/10.1109/JIOT.2022.3215805

Sun YK, Lei B, Liu J, et al., 2022. Computing power network: a survey. *China Commun*, early access. https://doi.org/10.23919/JCC.ja.2021-0776

Wahab OA, Mourad A, Otrok H, et al., 2021. Federated machine learning: survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Commun Surv Tut*, 23(2):1342-1397. https://doi.org/10.1109/COMST.2021.3058573

Wu WT, He LG, Lin WW, et al., 2021. Safa: a semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans Comput*, 70(5):655-668. https://doi.org/10.1109/TC.2020.2994391

Yang HH, Liu ZZ, Quek TQS, et al., 2020. Scheduling policies for federated learning in wireless networks. *IEEE Trans Commun*, 68(1):317-333. https://doi.org/10.1109/TCOMM.2019.2944169

Yang ZH, Chen MZ, Saad W, et al., 2021. Energy efficient federated learning over wireless communication networks. *IEEE Trans Wirel Commun*, 20(3):1935-1949. https://doi.org/10.1109/TWC.2020.3037554

Yang ZH, Chen MZ, Wong KK, et al., 2022. Federated learning for 6G: applications, challenges, and opportunities. *Engineering*, 8:33-41. https://doi.org/10.1016/j.eng.2021.12.002

List of supplementary materials

1 Additional information on orthogonal frequency division multiple access (OFDMA)

2 Simulations under the peer-to-peer architecture

Table S1 Parameter settings under the traditional architecture

Fig. S1 Test accuracy comparison under the peer-to-peer architecture in experiment 2

Fig. S2 System operation flowchart

Algorithm S1 Optimal transmission path selection strategy