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

Supplementary materials for

Yizhuo CAI, Bo LEI, Qianying ZHAO, Jing PENG, Min WEI, Yushun ZHANG, Xing ZHANG, 2024. Communication efficiency optimization of federated learning for computing and network convergence of 6G networks. *Front Inform Technol Electron Eng*, 25(5):713-727.
https://doi.org/10.1631/FITEE.2300122

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

OFDMA is the amalgamation of orthogonal frequency division multiplexing (OFDM) and frequency division multiple access (FDMA) technologies, achieving multi-user channel resource multiplexing by allocating subcarriers to different users and introducing multiple access methods within the OFDM system. This is a transmission technique that involves utilizing OFDM to realize subcarrier allocation of the channel and then loading transmission data on a subset of subcarriers. Users have the option to transmit data on subchannels with favorable channel conditions. Furthermore, OFDMA employs numerous orthogonal narrowband subcarriers to carry data, effectively mitigating multipath effects. It achieves frequency-domain spatial multiplexing, suitable for parallel transmission of small data packets, enhancing channel utilization and transmission efficiency for a single spatial stream, reducing application latency and user queuing, and maintaining stable operational states. It is particularly well suited for the transmission of model parameters in federated learning.

2 Simulations under the peer-to-peer architecture

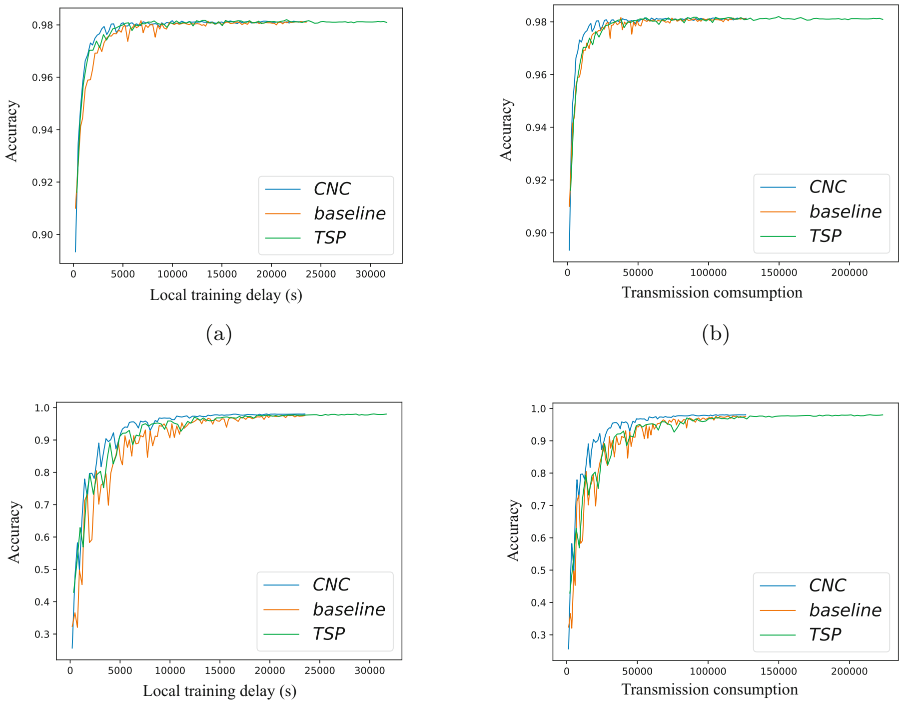
2.1 Parameter settings of the simulation environment

In the second experiment, the number of clients is reduced. To ensure the convergence rate of the model, the optimization method of computing and network convergence (CNC) chooses to divide the whole into two parts, in which the computing power resources of the main part are superior to that of the other part. We designed the transmission consumption matrix of eight clients. Three different parameter settings are simulated, among which the selection of transmission path is slightly different:

1. All of the eight clients participate in the training. The transmission problem is transformed into a traveling salesman problem (TSP);
2. The eight clients are divided into two parts, and the main part includes six clients (CNC);
3. In each global training round, six clients are randomly selected (baseline).

2.2 Simulation results and analysis

The results of the second experiment under the peer-to-peer architecture, and the corresponding analysis are as follows. In Figs. S1a and S1c, we compare algorithm performance in terms of local training time delay. The global model of optimization of CNC converges the fastest. Moreover, our proposed method maintains good performance when measured in terms of transmission consumption, as shown in Figs. S1b and S1d. The optimization under the peer-to-peer structure can be advantageous when there are fewer nodes involved in federated learning. Solving the TSP does not guarantee a small local training time, although it is possible to find the optimal transmission path. In the baseline, although it guarantees smaller transmission consumption



(a) (b) (c) (d)

Fig. S1 Test accuracy comparison under the peer-to-peer architecture in experiment 2: (a) local training delay (IID); (b) transmission consumption (IID); (c) local training delay (non-IID); (d) transmission consumption (non-IID)

and local training delay, the gradient information obtained by the model per global training round is not rich compared to our proposed method.

When fewer clients participate in training during federated learning, our method demonstrates superior performance in both communication metrics. The use of multiple subsets for chained transmission not only ensures model's convergence rate and performance but also reduces local training delay. By making real-time network topology based decisions on client model transmission paths, it really reduces the consumption of communication resources in federated learning, thus enhancing communication efficiency.

In the simulations under the traditional architecture, we use a random seed to generate I and d and then calculate the communication transmission. Table S1 not only contains information regarding parameters like client transmitting power P and uplink bandwidth B^{U} in the wireless network environment but also presents values of certain training parameters in federated learning.

For the scheduling of computing power, the computing power of the local client device is equivalent to the local training time due to the consistent amount of client data as well as model size in the simulation. We tested a local training time of about 4 s for a client. Then, we set up the heterogeneous situation of the computing power resources of the clients and simulate the realistic situation after calculating the local training delay.

Based on the structure of Fig. 2 in the main text, Fig. S2 gives the overall flow of the whole system from initialization to entering the federated learning training process of both architectures and finally completing the training.

Algorithm S1 is the method we proposed to find the optimal transmission path. It follows a greedy approach using the current node as a reference point to search for the nearest next node that is reachable. This next node then becomes the reference point, and the same method is applied in a depth-first traversal. If no nodes are reachable and the current node is not the final destination, it backtracks to the previous node

Table. S1 Parameter settings under the traditional architecture

Parameter	Value	Parameter	Value
N_0	-174 dBm/Hz	U	$\{100, 60\}$
B^{U}	1 MHz	cfraction	$\{0.1, 0.2\}$
P	0.01 W	epoch_local	$\{1, 5\}$
I	$\mathrm{U}(10^{-8},\ 1.1 \times 10^{-8})$	batch_size	10
d	U(0, 500)	η	0.01
$Z(g)$	0.606 MB	T	$\{300, 250\}$
o	1		

$Z(g)$: model parameter size; 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; batch_size: number of samples in one client-side training; η: learning rate; T: number of global training rounds

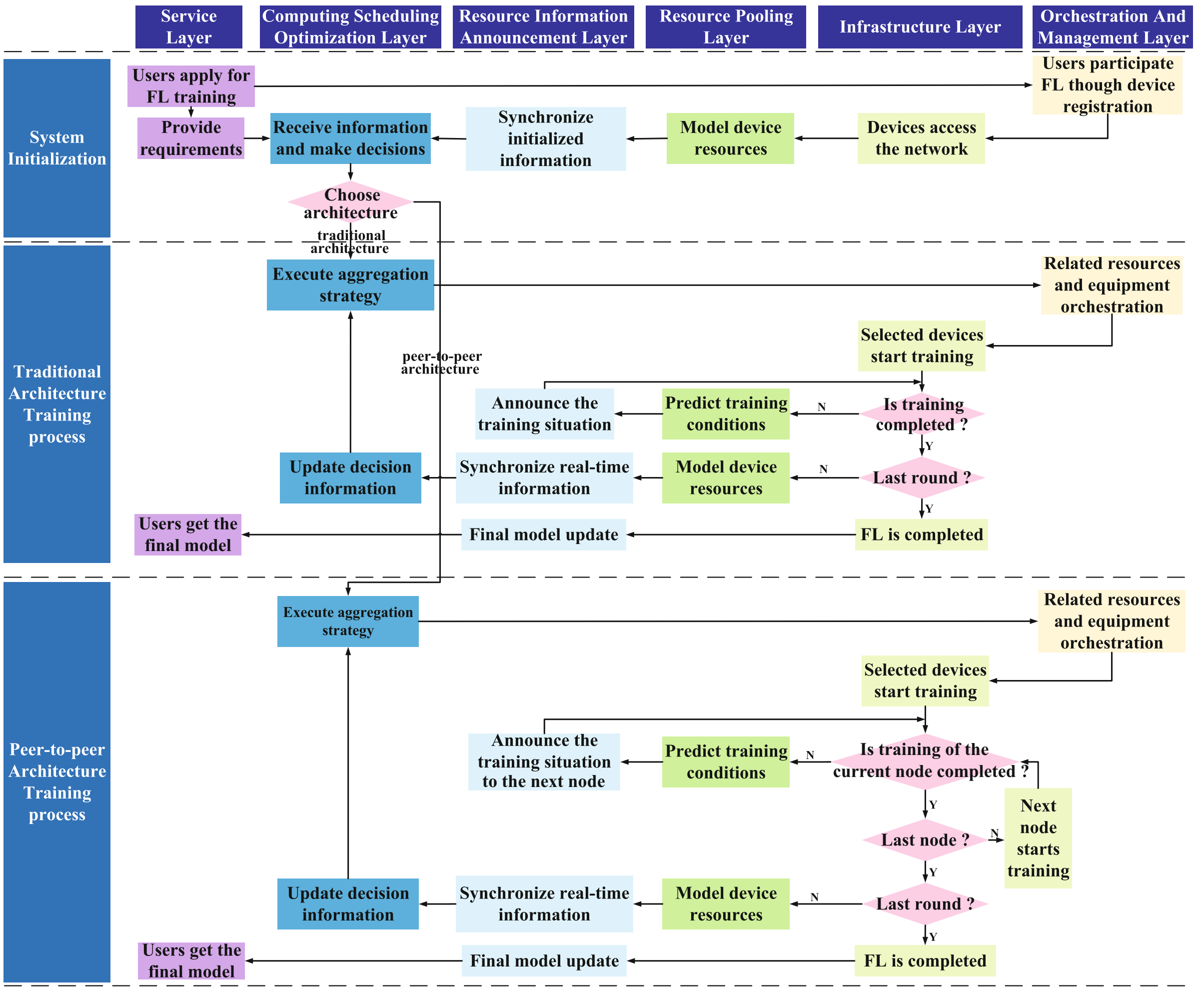


Fig. S2 System operation flowchart

and looks for a suboptimal node to be the next one, repeating this process until the optimal transmission path for all nodes is determined. The detailed steps are as illustrated in Algorithm S1.

Algorithm S1 Optimal transmission path selection strategy

Require: The consumption matrix of S_{te}: $\boldsymbol{G}_e$
Ensure: The optimal transmission path for S_{te}: *trace_path*

for i in S_{te} **do**
  Initialize array variable *trace*, $trace = [[i]]$
  while *trace* **do**
    Get the last feasible path *current_trace*, extract the last client *current_point* in *current_trace*
    for j in S_{te} **do**
      if client j has been traversed or the client is at infinite distance from *current_point* **then**
        Continue
      end if
      Save the next client j and its distance to the client *current_point*
    end for
    if no current saved next client **then**
      Remove the current path
      Continue
    else
      Select the shortest distance of the client connected to *current_point* as the next client and output the latest transmission path
      if all clients are traversed **then**
        Get *trace_path* of client i
        Break
      end if
      trace stores the current feasible paths
    end if
  end while
end for
Select the path with the smallest sum of transmission consumption in S_{te} as *trace_path*
return *trace_path*