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

Yi-zhuo Cai, Bo Lei, Qian-ying Zhao, Jing Peng, Min Wei, Yu-shun zhang, Xing Zhang 2022. Communication Efficiency Optimization of Federated Learning for Computing and Network Convergence of 6G Networks. *Front Inform Technol Electron Eng*, 23(4):617-629.
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 subcarrierize 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. This is particularly well suited for the transmission of model parameters in federated learning.

2 Simulation experiments under peer-to-peer architecture

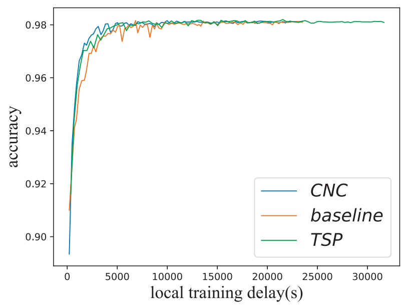
2.1 Simulation environment parameter setting

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 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:

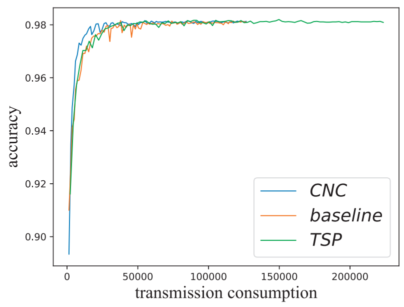
1. All of the eight clients participate in the training. The transmission problem is transformed into a total suspended particulate problem (TSP);
2. The eight clients are divided into two parts, and the main part includes six clients (CNC);
3. Each global training randomly selects six clients (baseline).

2.2 Experiment results and analysis

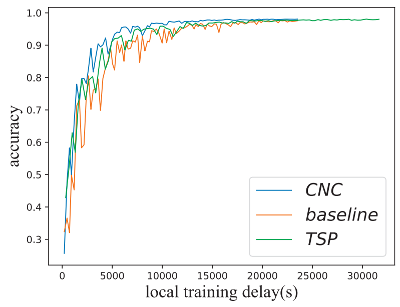
The results of the second experiment under the peer-to-peer architecture, and its analysis is as follows. In Fig. S1(a) and S1(c), we compare algorithm performance in terms of local training time delay. The global model of optimization of CNC converges faster. Moreover, our proposed method maintains good performance when measured in terms of transmission consumption, as shown in Fig. S1(b) and S1(d). The optimization under peer-to-peer structures can be advantageous when there are fewer nodes involved in



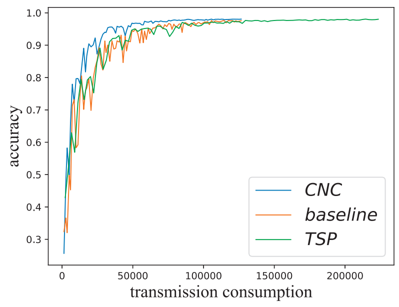
(a) local training delay(IID)



(b) transmission consumption(IID)



(c) local training delay(non-IID)



(d) transmission consumption (Non-IID)

Fig. S1 Results of test accuracy comparison under peer-to-peer architecture in experiment 2 (the horizontal axis is the different communication consumption)

federated learning. Solving the TSP problem 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 and local training delay, the gradient information obtained by the model per global training 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 transmissions ensures not only model the convergence rate and performance but also reduces local training latency. 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.

3 Table S1 Parameter value setting under traditional architecture

In the experimental simulation under the traditional architecture, we use random number seeds 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, $num\_clients$ represents the total number of participating clients, $cfraction$ is the sampling ratio for each global training, and $local\_epoch$ is the number of local training rounds for each client. Similarly, $batch\_size$ represents the number of samples in one client-side training, lr stands for the learning rate, and $global\_epoch$ denotes the number of global training iterations.

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.

Table. S1 Parameter value setting under traditional architecture

Parameter	Value	Parameter	Value
N_0	-174 dBm/Hz	m	0.024 dB
B^U	1 MHz	$num\_clients$	[100, 60]
P	0.01 W	$cfraction$	[0.1, 0.2]
I	U $(10^{-8}, 1.1 \times 10^{-8})$	$local\_epoch$	[1, 5]
d	U (0,500)	$batch\_size$	10
$Z(g)$	0.606 MB	lr	0.01
o	1	$global\_epoch$	[300, 250]

4 Fig. S2 System operation flow chart

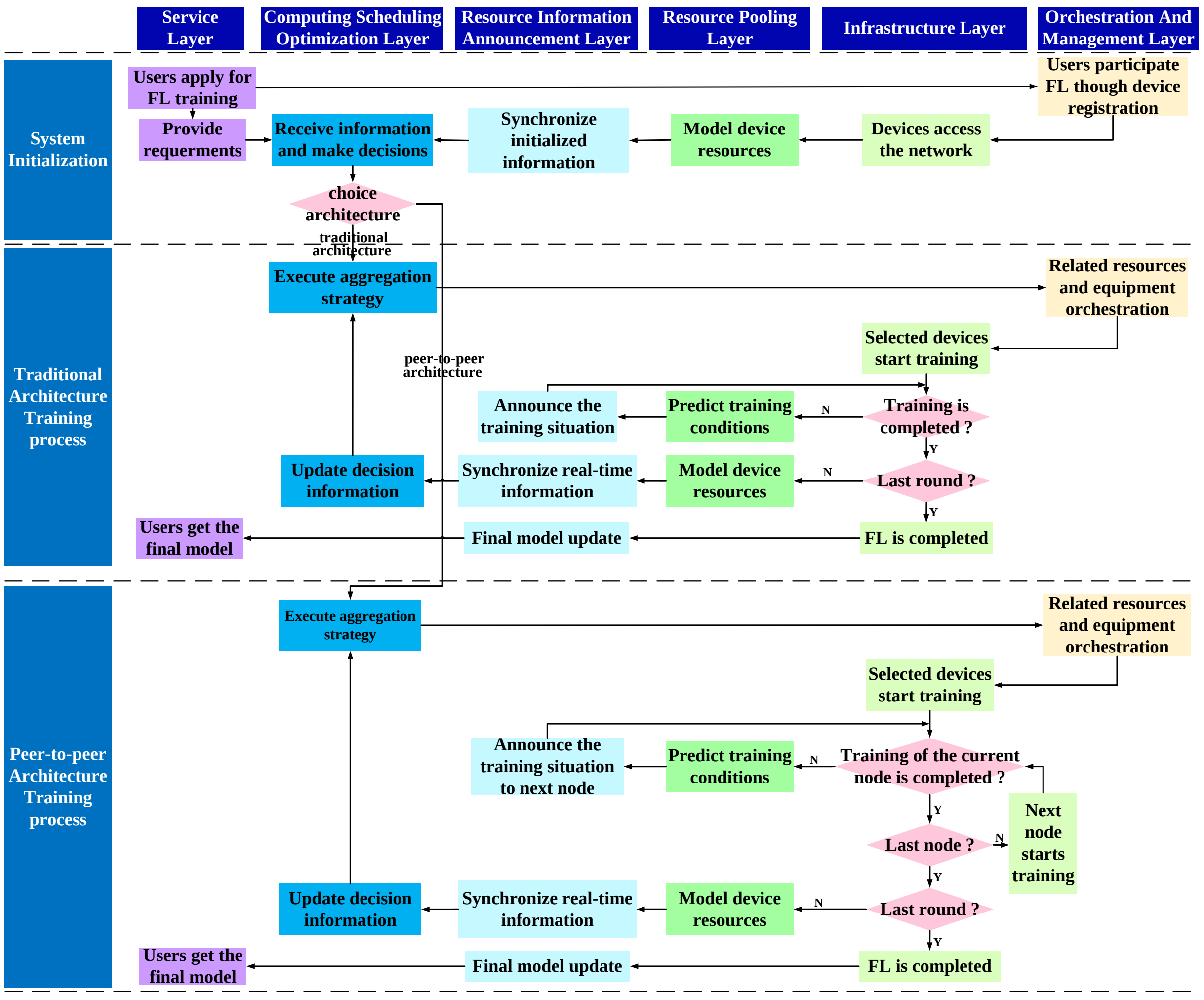

Service Layer
Computing Scheduling Optimization Layer
Resource Information Announcement Layer
Resource Pooling Layer
Infrastructure Layer
Orchestration And Management Layer
System Initialization
Users apply for FL training
Users participate FL though device registration
Provide requerments
Receive information and make decisions
Synchronize initialized information
Model device resources
Devices access the network
choice architecture
traditional architecture
peer-to-peer architecture
Traditional Architecture Training process
Execute aggregation strategy
Related resources and equipment orchestration
Selected devices start training
Announce the training situation
Predict training conditions
Training is completed ?
N
Y
Update decision information
Synchronize real-time information
Model device resources
Last round ?
Users get the final model
Final model update
FL is completed
Peer-to-peer Architecture Training process
Execute aggregation strategy
Related resources and equipment orchestration
Selected devices start training
Announce the training situation to next node
Predict training conditions
Training of the current node is completed ?
Next node starts training
Last node ?
Update decision information
Synchronize real-time information
Model device resources
Last round ?
Users get the final model
Final model update
FL is completed

Fig. S2 System operation flow chart

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.

5 Algoritem S1 Optimal transmission path selection strategy

Algorithm S1 Optimal transmission path selection strategy

```
Require: The consumption matrix of $S_{te}$:$G_e$.
Ensure: The optimal transmission path for$S_{te}$:$trace\_path$.
 1: for $i$ in $S_{te}$ do
 2:   Initialize array variable $trace$, $trace = [[i]]$
 3:   while $trace$ do
 4:     Get the last feasible path $current\_trace$, extract the last client $current\_point$ in $current\_trace$
 5:     for $j$ in $S_{te}$ do
 6:       if client $j$ has been traversed or the client is at infinite distance from the $current\_point$ then
 7:         continue
 8:       end if
 9:       Save the next client $j$ and its distance to the client $current\_point$
10:     end for
11:     if no current saved next client then
12:       Remove the current path
13:       Continue
14:     else
15:       Select the shortest distance of the client connected to $current\_point$ as the next client and output
          the latest transmission path
16:       if all clients are traversed then
17:         Get $trace\_path$ of client $i$
18:         Break
19:       end if
20:       $trace$ stores the current feasible paths
21:     end if
22:   end while
23: end for
24: Select the path with the shortest sum of transmission consumption in $S_{te}$ as $trace\_path$
25: return $trace\_path$
```

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 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.