Joint Placement of UPF and Edge Server for 6G Network

Yuanzhe Li, *Graduate Student Member, IEEE*, Xiao Ma, *Member, IEEE*, Mengwei Xu, Ao Zhou, *Member, IEEE*, Qibo Sun, *Member, IEEE*, Ning Zhang, *Senior Member, IEEE*, and Shangguang Wang, *Senior Member, IEEE*

***Abstract*—The emerging 6G network will make it possible for cybertwin, which relies deeply on the low latency and powerful computation provided by the edge network. To this end, the convergence of computing and network has been attached great importance. Most existing work study either placing edge servers or deploying user plane functions (UPFs), seldom considers the two processes jointly. In this article, we study how to minimize the latency with cost limitation by means of jointly deploying edge servers and UPFs in 6G scenario. We have shown that the problem is NP-hard. Then, we simplify the problem by analyzing the placement relationship between edge servers and UPFs and prune the solution space of the problem. To solve the problem effectively, a UPF and edge server placement algorithm is proposed. Massive experiments are conducted based on real-world data set and an edge core network emulator. The evaluation results show that our algorithm outperforms the benchmark algorithms.**

***Index Terms*—6G, cybertwin, edge server, user plane function (UPF).**

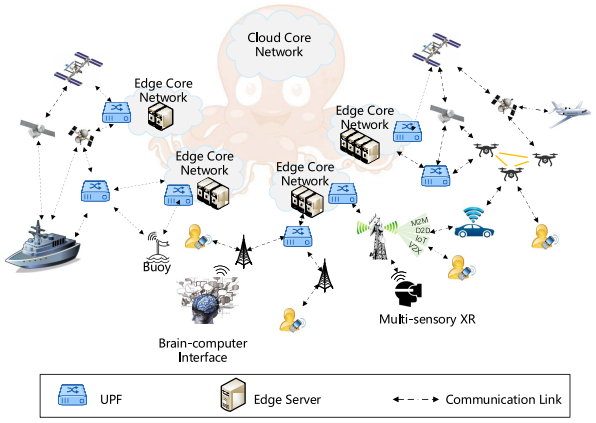


Fig. 1. 6G edge core network architecture.

I. Introduction

THE 6G network is expected to realize the Internet of Everything [1], [2]. With the introduction of satellite communication, unmanned aerial vehicle communication, and maritime communication as supplements, user equipments can get access to the Internet services from anywhere with reliable low latency and mobile broadband. Besides, 6G network is on course for a higher peak data rate ($>$ 100 Gb/s), higher traffic density ($>$ 100 Tbps/m^2), lower latency ($<$ 1 ms), and better reliability ($>$ 99.999%) with much larger coverage area (for more than 99% of the Earth) [1]. Such a network upgrade makes it possible to realize cybertwin [3], [4], which locates at edge clouds and serve as a digital representation of human or IoT devices. Cybertwin relies heavily on a low-latency network environment to achieve full capability and meet the requirements of latency-sensitive and computation-intensive applications at user ends.

In order to adapt to such complex scenarios and requirements, the current core network architecture will further evolve into a network where most network control and service provision are conducted at the network edge, i.e., edge core network. As is shown in Fig. 1, such a network works like the nervous system of octopus where most of the nerve cells locate in the arms and only a the tiny fraction of nerve cells is in the central brain [5]. As a result, most complex motor skills are decided and conducted by the arms themselves. Likewise, leveraging edge servers deployed at the proximity of users, 6G network provides low-latency services with high efficiency and flexibility [6]. The cloud core network, playing a role of the central brain, will not directly participate in the communication. Most tasks are offloaded to edge servers via user plane functions (UPFs) to satisfy the latency-sensitive and computation-intensive service requirements.

This network structure puts forward higher requirements for edge infrastructures. As most user equipments rely on edge

Manuscript received January 31, 2021; revised May 27, 2021; accepted July 3, 2021. Date of publication July 6, 2021; date of current version November 5, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0205503, and in part by NSFC under Grant 62032003, Grant 61922017, and Grant 61921003. *(Corresponding author: Ao Zhou.)*

Yuanzhe Li, Xiao Ma, Mengwei Xu, Ao Zhou, and Qibo Sun are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: buptlyz@bupt.edu.cn; maxiao18@bupt.edu.cn; mwx@bupt.edu.cn; aozhou@bupt.edu.cn; qbsun@bupt.edu.cn).

Ning Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: ning.zhang@uwindsor.ca).

Shangguang Wang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Network Communication Research Center, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: sgwang@bupt.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3095236

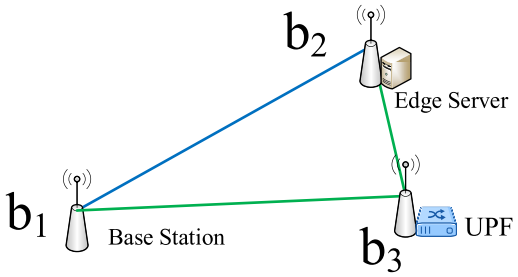


Fig. 2. Influence of UPF on backhaul communication path.

servers to acquire low-latency services, the deployment of edge infrastructures should be carefully considered [7]. The most intuitive factor that will influence the latency is the distance between user equipments and edge servers. However, taking the locations of UPFs into consideration is also indispensable. UPFs are in charge of steering user traffics [8], [9]. Every user data packet must pass through UPF before arrive at edge clouds. Take Fig. 2 as an example. b_1, b_2, and b_3 are three base stations, and any of the two base stations are connected directly. An edge server is placed at b_2 and a UPF is deployed at b_3. If a user at b_1 wants to offload tasks to the edge server, its data packets must pass through the UPF at b_3. As a result, the communication path of data packets is the green one instead of the blue path between b_1 and b_2, even if the blue path is the shorter one. This example clearly indicates that the location of UPF has a great impact on communication path and latency. In the 6G network, the existence of cybertwin relies highly on the computation and communication resources provided by the edge network system. Improper placement of either edge servers or UPFs will inevitably result in the deterioration of quality of service.

Most existing work studies the placement of edge servers and UPFs separately. References [10]–[15] study the placement of network gateway purely from the perspective of network without considering the location of edge servers. References [16]–[20] focus on optimizing the key parameters in edge server placement, neglecting the influence of UPFs on latency. Although [21] considered both the placement of edge servers and UPFs, the two placement is still conducted separately. The interdependence between the two processes is neglect. In this article, the influences of UPF and edge server on latency are considered simultaneously, and we formulate it as a joint placement problem.

Solving the joint deployment problem is challenging. The first challenge is the interaction between edge servers and UPFs when calculating the latency. The latency in this work refers to the time for a data packet to travel from a base station to the edge server via UPFs. Therefore, changing the location of either edge servers or UPF both results in different latency. Considering the two subproblems, edge server placement and UPF placement, are both NP-hard, the complexity of the joint placement problem is significantly increased. To tackle this challenge, we propose a two-level algorithm. First, base stations are divided into several groups by means of periodically merging operation. Then, the location of edge servers and UPFs is optimized in turn.

The second challenge comes from the scale of the problem. The placement involves choosing locations for edge servers and UPFs as well as assigning base stations to edge servers via one UPF. Considering the large quantity of base stations, the search space would be tremendous. To simplify the problem, we analyze the placement relationship between edge servers and UPFs. By proving some solutions are unrealistic and can be ruled out, we prune the search space.

The contribution of this article is threefold.

1) We formulate the problem of joint placing UPF and edge server in 6G scenario aiming to minimize latency with cost and bandwidth limitation. To the best of our knowledge, we are the first to study the placement of edge servers and UPFs jointly instead of splitting it into two processes.
2) We analyze the complexity of the problem and prove that the problem is NP-hard. To solve the problem efficiently, we first simplify it by pruning the search space via discussing the location relationship between edge servers and UPFs. Then, an effective UPF and edge server placement algorithm (UEPA) is proposed.
3) Extensive simulations are conducted leveraging real-world data set and network emulator. The evaluation results show that our proposed algorithm stands out in terms of achieving low latency with limited total cost.

II. System Model and Problem Formulation

A. System Model

The topology of the edge network system is considered as an undirected graph $G = (\mathcal{B} \cup \mathcal{E} \cup \mathcal{U}, \mathcal{A})$. $\mathcal{B} = \{b_1, \ldots, b_i, \ldots, b_{n_b}\}$ represents the set of base stations, where $b_i (i = 1, 2, 3, \ldots, n_b)$ denotes base station i. $\mathcal{E} = \{e_1, \ldots, e_j, \ldots, e_{n_e}\}$ denotes the edge clouds, where $e_j (j = 1, 2, 3, \ldots, n_e)$ denotes edge cloud j. In each edge cloud, several edge servers are placed to form a cluster. $\mathcal{U} = \{u_1, \ldots, u_j, \ldots, u_{n_u}\}$ denotes the set of UPFs, where $u_k (k = 1, 2, 3, \ldots, n_u)$ denotes UPF k. n_b, n_e, and n_u denote the total number of base stations, edge clouds, and UPFs, respectively. Edge clouds and UPFs are assumed to be deployed at base stations. Therefore, $n_b \geq n_e$ and $n_b \geq n_u$. $\mathcal{A}$ denotes the set of physical network links among base stations, edge clouds, and UPFs. Let $\alpha_{ij} \in \{0, 1\}$ denote the assignment relationship between base station and edge cloud. If $\alpha_{ij} = 1$, base station b_i is assigned to edge cloud e_j. Similarly, let $\beta_{ik} \in \{0, 1\}$ denote the assignment relationship between base station and UPF. $\beta_{ik} = 1$ represents UPF. u_k is the default UPF of b_i.

B. Latency Model

When user equipments offload tasks to edge clouds, the latency mainly comes from three parts: 1) the transmission delay between the user equipment and base station through the wireless connection; 2) the backhaul delay between the base station and edge clouds via wired links; and 3) the processing delay for edge servers to process the task [22]. Generally, changing the deployment scheme of edge servers and UPFs mainly affects the topology of backhaul wired links, thus the

latency in this article is defined as the backhaul round-trip time between the base station and edge server. Let $d(\varphi_i, \varphi_j)$ denote the latency between two network devices, where φ_i and φ_j represent the two ends of one communication link, respectively. For an offloading data stream, the latency from the base station to the edge server can be split into two parts: 1) the latency between the base station and UPF and 2) the latency between UPF and edge server. As a result, the latency of offloading tasks to edge servers can be defined as follows:

$$d(b_i, e_j) = \sum_{k=1}^{n_u} \beta_{ik}\big[d(b_i, u_k) + d(u_k, e_j)\big]. \tag{1}$$

If $\beta_{ik} = 1$, the data stream of base station b_i is steered by UPF u_k, and vice versa. $d(b_i, u_k)$ denotes the latency between b_i and u_k. $d(u_k, e_j)$ denotes the latency between u_k and e_j.

1) Latency of Edge Node: The latency of edge cloud e_j is denoted as $D(e_j)$, which is defined as the maximum latency between e_j and all the base stations assigned to it

$$D(e_j) = \max \alpha_{ij} d(b_i, e_j) \quad (\forall i, 1 \le i \le n_b) \tag{2}$$

where $\alpha_{ij} \in \{0, 1\}$ indicates whether base station b_i is assigned to edge cloud e_j.

2) Average Latency: For a placement scheme, the average latency of all base stations is defined as follows:

$$\bar{D} = \frac{\sum_{i=1}^{n_b} \sum_{j=1}^{n_e} \alpha_{ij} d(b_i, e_j)}{n_b}. \tag{3}$$

C. Cost Model

1) Bandwidth Cost: The UPF serves as the upload classifier of the area. Large quantities of data merge at UPF. This includes data offloaded to local edge clouds and data targeted to the remote cloud data centers. Therefore, UPFs are faced with big pressure on bandwidth. To prevent being overloaded, enough UPFs should be deployed and the placement locations should be decided according to the distribution of user workloads. The bandwidth cost refers to the deployment cost of UPFs, which is defined as follows:

$$C_B = p_b B_u n_u \tag{4}$$

where p_b denotes the price per bandwidth. B_u denotes the bandwidth of each UPF. n_u denotes the total number of UPFs.

2) Computation Cost: The computation cost consists of two parts: 1) the equipment cost and 2) the construction cost. The equipment cost comes from the purchase of edge servers. The construction cost is used to build the edge cloud. The computation cost is defined as follows:

$$C_{\text{comp}} = \sum_{j=1}^{y} \left(p_e m_j + p_c\right) \tag{5}$$

where p_e denotes the price of one edge server. m_j denotes the number of edge servers deployed at edge cloud j. p_c denotes the construction cost of one edge cloud.

3) Total Cost: The total cost is acquired by adding bandwidth cost and computation cost together, which is defined as follows:

$$C = C_B + C_{\text{comp}}. \tag{6}$$

D. Problem Statement

Latency is the key performance indicator of the 6G network. The deployment problem is to minimize the latency with cost limitation. The problem is formulated as follows:

$$\text{Minimize} \quad \bar{D} \tag{7}$$

$$\text{s.t.} \quad C \le C_{\max} \tag{8}$$

$$D(e_j) \le D_{\max} \tag{9}$$

$$\sum_{i=1}^{n_b} \alpha_{ij} w(b_i) \le m_j w_s \quad (\forall j, 1 \le j \le n_e) \tag{10}$$

$$\sum_{i=1}^{n_b} \beta_{ik} B(b_i) \le B_u \quad (\forall k, 1 \le k \le n_u) \tag{11}$$

$$\sum_{j=1}^{n_e} \alpha_{ij} = 1 \quad (\forall i, 1 \le i \le n_b) \tag{12}$$

$$\sum_{k=1}^{n_u} \beta_{ik} = 1 \quad (\forall i, 1 \le i \le n_b) \tag{13}$$

where $C_{\max}$ denotes the total cost limitation and $D_{\max}$ denotes the maximum edge latency. Constraint (8) indicates that the total cost is limited. Constraint (10) ensures that each edge cloud is not overloaded, where $w(b_i)$ denotes the workload of base station b_i. Constraint (11) prevents network congestion at UPFs as a result of total data stream exceeding the bandwidth of UPF. $B(b_i)$ denotes the bandwidth that is needed for users at base station b_i. Constraint (12) states that one base station can only offload tasks to one edge cloud. Constraint (13) guarantees that data streams of one base station are all forwarded by one UPF.

Above all, the joint edge server and UPF deployment problem can be formally stated as follows.

1) Find an optimal joint deployment solution.
2) Maximizing the total profit in (7).
3) Subject to constraints in (8)–(13).

E. Complexity Analysis

1) Edge Server Placement Only: In this situation, we only consider the placement of edge servers and neglect the placement of UPFs. Thus, the problem is transformed as follows:

$$\text{Minimize} \quad \bar{D}$$

$$\text{s.t.} \quad (8),\ (9),\ (10) \text{ and } (12).$$

Theorem 1: The edge server placement problem with cost constraint is NP-hard.

Proof: We prove the edge server placement is NP-hard by a reduction from the set cover problem. Given a universe set $\mathcal{Z} = \{z_1, z_2, \ldots, z_n\}$, a size constraint K, and a subset of $\mathcal{Z}$ denoted as $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m\}$, the set cover problem is how to find a collection C from the subsets so that $|C| < K$ and $\bigcup_{i \subseteq C} \mathcal{S}_i = \mathcal{Z}$.

We reduce the set cover problem to the edge server placement problem as follows. First, we map each $z_i \in \mathcal{Z}$ to a base station b_i by a one-to-one mapping. Then, another one-to-one mapping is constructed that maps each subset $\mathcal{S}_j$ to an edge

cloud coverage area. In the coverage area, there is only one edge cloud. If $z_i \in \mathcal{S}_j$, the corresponding b_i is in the coverage area and assigned to the edge cloud. Here, we set $K = n_b$. It is clear that an edge server placement scheme contains several edge server coverage areas whose corresponding subsets are also a cover of $\mathcal{Z}$. As the set cover problem has been proven to be NP-complete [23], the edge server placement problem is NP-hard. ■

2) UPF Placement Only: Here, we consider the special case where edge servers have been deployed, and we only need to place UPFs at base stations. The problem in Section II-D will be change into

$$\text{Minimize} \quad \bar{D}$$
$$\text{s.t.} \quad (8),\ (9),\ (11) \text{ and } (13).$$

Theorem 2: The placement of UPF with cost constraint is NP-hard.

Proof: We prove the NP-hardness of the problem by reducing the 0-1 knapsack problem to it. There is a set $\mathcal{Z}$ containing η items. Each item has a value v_i and weight $w_i (i = 1, 2, \ldots, \eta)$. For a knapsack with a given size Ω, the 0-1 knapsack problem is to select a subset $\mathcal{S} \subseteq \mathcal{Z}$ satisfying $\sum_{i \in \mathcal{S}} w_i \leq \Omega$ while $\sum_{i \in \mathcal{S}} v_i$ is maximized.

First, map z_i of $\mathcal{Z}$ to b_i of the base station set $\mathcal{B}$. If the base station b_i is chosen to deploy UPF, the relevant z_i is placed in the knapsack. The weight w_i of z_i is mapped to the cost of placing one UPF which is denoted as $p_b B_u$. Next, map the value v_i to the negative value of adding all the latency of base stations that forward data stream via this UPF, i.e., $-\sum_{i=1}^{n_b} \sum_{j=1}^{n_e} \alpha_{ij} \beta_{ik} d(b_i, e_j)$, where k is the id of UPF placed at b_i. The size Ω of knapsack is mapped to the total cost $C_{\max}$. Every UPF deployment scheme provides a solution to the knapsack problem. Specifically, with no UPF placed, the latency of each base station is set to infinity. This state corresponds to the case of an empty knapsack whose total value equals negative infinity. Then, for a deployment scheme, the shorter the average latency it has, the larger the total value the corresponding knapsack will get. According to [23], the 0-1 knapsack problem is NP-complete. Therefore, the UPF deployment problem is NP-hard. ■

Based on the above analysis, the joint placement of UPF and edge server is NP-hard.

III. Algorithm Design

A. Problem Simplification

The joint placement is quite complicated. One of the reasons is that the search space is extremely large. However, some placement schemes, although satisfying the constraints, can be excluded because there are better solutions.

1) Base Case: We first consider the scenarios in one coverage area where only one edge cloud and one UPF are deployed.

Lemma 1: If there is only one edge cloud and one UPF in a coverage area, the average latency is the shortest when UPF and edge cloud are deployed at the same location.

Proof: Consider two scenarios. In the first scenario, edge cloud and UPF are deployed at the same base station [shown in Fig. 3(a)]. In the second scenario, keep edge cloud at the same location and move UPF to another base station [shown in Fig. 3(b)]. After changing the location of UPF, the coverage area is split into three sets: 1) the UPF *set* consists of base stations locate far away from edge cloud (denoted as set 1 in red circle); 2) the *edge set* consists of base stations on the one side of edge cloud far away from UPF (denoted as set 3 in orange circle); and 3) the *between set* locates between UPF and edge cloud (denoted as set 2 in blue circle). Changing the location of UPF brings different influence on the three sets. The latency of base stations in UPF *set* does not change because the paths to reach edge cloud are the same. However, latency rises for base stations in *edge set* and *between set*. Specifically, the *edge set* suffers a latency increase of $2d(e_0, u_0)$, where e_0 denotes the edge cloud and u_0 denotes the UPF. The latency increase Δd of *between set* satisfies $0 < \Delta d < 2d(e_0, u_0)$, depending on the location of base station. For example, for base stations in the *between set* of Fig. 3(b), the latency increases $2d(b_5, b_4)$.

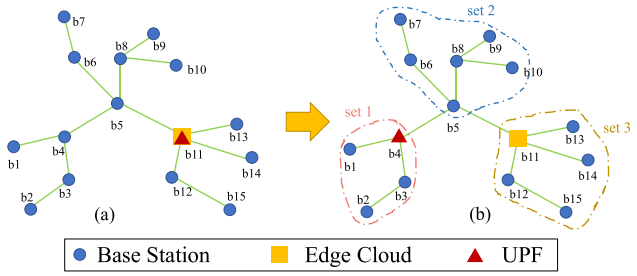


Fig. 3. Base case with one edge cloud and one UPF.

In summary, moving UPF to other location from edge cloud will bring extra latency. Therefore, place edge cloud and UPF at the same location will have the shortest latency in the base case. ■

2) Multiple Edge Clouds With One UPF: Each coverage area is deployed with multiple edge clouds, all served by the same one UPF. This deployment scheme deals with scenarios where the demand for computing resources is relatively large.

Lemma 2: Deploying more edge servers at the edge cloud colocated with UPF is better than deploying multiple edge clouds at different locations.

Proof: As there is only one UPF in the coverage area, denoted as u_0, data stream from all base stations within the area should be forwarded to the UPF first. For a base station b_i, transmitting data packets from b_i to u_0 introduces $d(b_i, u_0)$. Then, if the target edge cloud is colocated with the UPF, $d(b_i, u_0)$ will be the final latency. However, if data stream is forwarded to edge clouds at different location, there will be an extra latency between UPF and the edge cloud. Therefore, expanding the computing resource capacity of the edge cloud that is colocated with the UPF is more applicable. ■

3) Multiple UPFs With One Edge Cloud: In each coverage area, only one edge cloud is deployed. Multiple UPFs forward data stream to it. Such a placement scheme results from a bandwidth shortage of one UPF. In this scenario, one UPF colocates with the edge cloud. Other UPFs are distributed to other base stations and only in charge of base stations in its

UPF *set*. This is a practical and applicable scheme. By deploying multiple UPFs, the bandwidth pressure of one UPF is relieved and distributed to other UPFs. The distribution brings other benefits, as well. First, the reliability of the system is improved. Deploying multiple UPFs can prevent whole coverage area network failure because of the only one UPF break down. Second, it prevents the unnecessary extra referred in the second case.

4) Summary of All Cases: According to the above analyzes, each edge cloud should colocate with one UPF. Other UPFs can be deployed at base stations if the UPF at the edge cloud is faced with bandwidth pressure. According to this rule, the problem can be simplified by excluding inapplicable placement schemes.

B. UPF and Edge Server Placement Algorithm

The proposed UEPA runs as Algorithm 1. It consists of five key operations, which are listed as follows.

Merge Operation: All the base stations are divided into several groups. Each group consists of at least one base station. The total workload of the group is kept no larger than the max workload of one edge cloud. Every time the merge operation is conducted, each group is paired up with its nearest group. Let $D_g(b_i)$ denote the maximum latency between base station b_i and other base stations in the same group. Then, the group with higher $[\sum w(b_i)/\max(D_g(b_i))]$ in the pair will be maintained and the other will be removed. The base stations of the removed group are faced with three choices. First, they will be added into the maintained one if the constraints are satisfied. Second, they will be tried to add into other groups. Third, if no group is acceptable, a new group will be formed.

Choose Edge Cloud Location: After the merge operation, each group will choose one base station as the location of edge clouds. The base station with the minimal $D_g(b_i)$ will be chosen as edge cloud.

Choose UPF: The next step is deploying UPFs. As mentioned above in Section III-A, the edge cloud will colocate with a UPF by default. Then, for each base station in the group, if there is one existing UPF satisfying all the constraints, this UPF will be set as the default UPF of the base station. Otherwise, a new UPF will be deployed at the base station.

Adjustment: The above steps will inevitably result in the overdevelopment of UPFs. Therefore, an adjustment will be conducted iteratively. In each iteration, the UPF with the highest $(\tilde{d}(u_k)/[\sum \beta_{ik}B(b_i)])$ will be removed, where $\tilde{d}(u_k) = ([\sum \beta_{ik}d(b_i, u_k)]/\sum \beta_{ik})$ denotes the average delay of UPF u_k. Its base stations will be assigned to other UPFs. The iteration ends if the removed UPFs are the same between two adjacent iterations.

Decide Final Scheme: The above four steps consist of all the operations in one loop. In each loop, the scheme is recorded and a Q value is calculated to evaluate the scheme. The Q is defined as follows:

$$Q = \frac{D_{\max}}{\bar{D}} - \frac{C}{C_{\max}}. \tag{14}$$

Algorithm 1: UEPA

Input: data set of base stations; $D_{\max}$; $C_{\max}$; algorithm iteration number $\mathcal{T}$;
Output: placement scheme of edge clouds and UPFs

```
initialize groups and choose core node for each group;
while t < T do
    merge the nearest groups pair by pair;
    foreach group do
        deploy edge cloud at base station with the minimal D_g(b_i);
        choose UPF locations from base stations in the same group;
        assign base stations to the edge cloud of the group;
        decide quantity of servers deployed at edge cloud;
    flag ← True;
    while flag do
        calculate d̃(u_k)/(Σ β_ik B(b_i)) of each UPF;
        if the UPF with the highest d̃(u_k)/(Σ β_ik B(b_i)) is the same as that of the last loop then
            flag ← False;
        else
            remove UPF with the highest d̃(u_k)/(Σ β_ik B(b_i));
            assign its base stations to other UPFs;
    calculate Q value of the scheme got in this loop;
return deployment scheme with the highest Q value.
```

A deployment scheme with lower average latency $\bar{D}$ and total cost C will have a higher Q value. At the end of the algorithm, the deployment scheme with the highest Q value will be chosen as the final scheme.

IV. Performance Evaluation

A. Experimental Setup

We build an edge core network emulator based on mininet[1] and libgtp5gnl[2] to measure latencies. The emulator implements a real backhaul network of UPFs to enable real connection for user equipments and edge servers. The emulator will set up general packet radio system tunneling protocol user plane (GTP-U) tunnel [24] to carry user data packets, which starts from base stations, passing through a UPF and finally end at edge servers. We use ping in this system to measure the round-trip time for different network connections. Besides, we use the real-world base station data set[3] from Shanghai Telecom to generate topology and simulate the joint deployment of edge servers and UPFs in the whole city of Shanghai [25], [26].

The price of one edge server p_e is set to 8000 RMB.[4] The construction cost of one edge cloud p_c is set to 50 000 RMB.

[1] http://mininet.org/
[2] https://github.com/PrinzOwO/libgtp5gnl
[3] http://sguangwang.com/TelecomDataset.html
[4] https://item.jd.com/100007288408.html

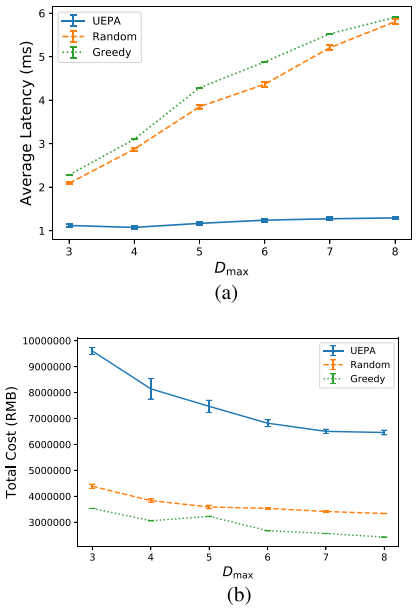


Fig. 4. Performance comparison with different $D_{\max}$. (a) Average latency. (b) Total cost.

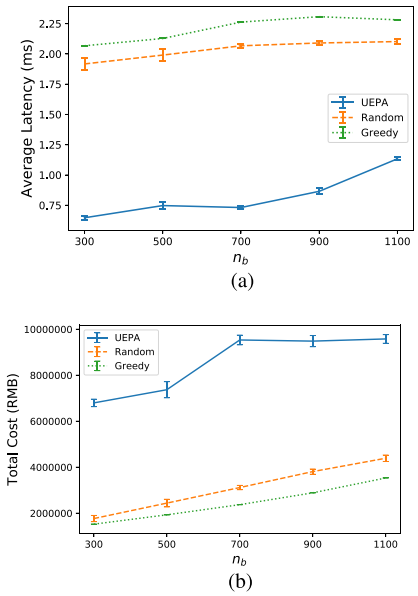


Fig. 5. Performance comparison with different n_b. (a) Average latency. (b) Total cost.

The maximum bandwidths of each UPF are the same, therefore, the placement cost per UPF is set to 10 000 RMB.[5] To acquire the bandwidth requirement of each request, we introduce γ which denotes the ratio of required bandwidth to workload. The large value of γ refers to bandwidth-exhausted applications while the small value means the application is computation intensive.

B. Benchmark Algorithms

To evaluate the performance of our proposed UEPA, two algorithms are introduced as a benchmark, which are listed as follows.

1) *Greedy:* Base stations with heavier workload have higher priority to be chosen as edge clouds. Other base stations satisfying the constraints are assigned one by one. If there is no available UPF, the current base station being assigned will deploy a UPF.
2) *Random:* Placement locations for edge servers and UPFs are selected randomly from all the base stations. Other base stations are assigned to the nearest edge cloud via the nearest UPF if constraints are not violated.

C. Performance With Varying Maximum Delay

Fig. 4(a) and (b) shows the performance of algorithms as the maximum latency increases from 3 to 8 ms, with the total base station number kept as 1100 and $\gamma = 0.0001$.

The value of maximum latency $D_{\max}$ limits the size of the coverage area per edge cloud. In Fig. 4(a), we can observe that as $D_{\max}$ increases, the latency of all algorithms grows up. Different from the obvious increase of Random and Greedy, our UEPA has a very light increment and keeps the lowest latency. When $D_{\max} = 3$ ms, the latency of UEPA is 46.44% and 50.96% less than Random and Greedy, respectively. As $D_{\max}$ increases, the gap enlarges sharply. When $D_{\max} = 8$ ms, the latency of UEPA is 1.29 ms, while the average latencies of Random and Greedy have become 5.80 and 5.90 ms, respectively. The lower latency comes from the higher cost. As is shown in Fig. 4(b), UEPA has the highest total cost. Because it deploys more edge clouds and UPFs. As $D_{\max}$ increases, the coverage area per edge cloud gets larger, and the total number of edge clouds and UPFs decreases. As a result, the total cost of all the three algorithm goes down. However, UEFA witnesses the most dramatic decrease. The cost of UEPA, Random, and Greedy reduces 32.72%, 23.95%, and 31.26%, respectively.

[5]https://item.jd.com/100007288380.html

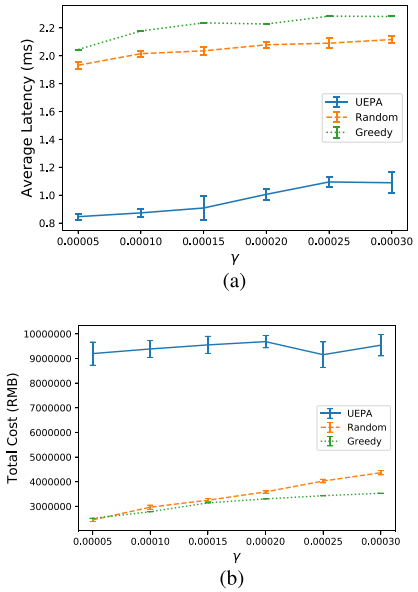


Fig. 6. Performance comparison with different γ. (a) Average latency. (b) Total cost.

D. Performance With Varying Base Station Number

Fig. 5(a) and (b) shows the performance of algorithms when the number of base stations varying from 300 to 1100. The maximum latency $D_{\max}$ is set to 3 ms and γ is set to 0.0001.

As n_b increases from 300 to 1100, edge clouds are faced with heavier workload and UPFs have to process larger data streams. In this process, the average latency of the three algorithms all goes up. Our UEPA keeps the lowest latency and stays far less than the other two algorithms. When $n_b = 300$, UEPA provides services with an average latency of 0.65 ms, which is 66.27% and 68.71% less than Random and Greedy, respectively. When n_b reaches 1100, UEPA still outperforms the other two algorithms. Its average latency is merely 53.94% of the latency gained by Random. On the other hand, in order to guarantee the latency under an acceptable level when the base station number increases, more edge clouds and UPFs are deployed, which results in the increment of total cost. Fig. 5(b) reveals that the lowest average latency of UEPA is achieved by means of paying the highest cost. However, it is worth noting that the total cost of UEPA goes up quickly when $n_b < 700$. Afterward, its increment is very slight. The former is because the extra base stations are covered by adding edge clouds and UPFs. The latter results from the coverage area getting larger to cover the extra base stations.

E. Performance With Varying Bandwidth Ratio

Fig. 6(a) and (b) shows the performance of algorithms when the bandwidth ration γ increases from 0.00005 to 0.00030. The maximum latency $D_{\max}$ is set to 3 ms and n_b is set to 1100.

Increasing the value of γ means UPFs are faced with heavier bandwidth pressure but the total workloads of edge server are not changed. Heavier bandwidth pressure leads to the increase of average latency. As γ increases, UEPA keeps the lowest latency but witnesses the largest increment. It increases 28.74%. Meanwhile, Random increases 9.53% and Greedy increases 11.58%. As for total cost, UEPA still has the highest cost and goes up bumpily in the process. The increment of total cost mainly comes from deploying more UPFs. The total cost of the three algorithms all rises. In specific, the cost increment of Random is the most significant, which increases 78.14%. The total cost of UEPA stays almost unchanged with a slight increase of 3.70%.

V. Related Work

A. Placement of Gateway

Kiess and Khan [10] studied the centralized and distributed deployment of network gateway in the 5G architecture. They propose easily tractable and formal cost model for gateway location and conduct nation-wide simulations. However, latency is not considered in this work. Costa-Requena *et al.* [11] realized 5G UPF components leveraging SDN. They implement different data transport strategies to reducing latency. Besides, how to evolve from legacy 4G gateways to 5G UPF is also discussed. Peters and Khan [12] focused on the session management in the 5G network. A three-stage learning-based approach is adopted to endow the 5G core network with anticipatory functionality. Such a functionality is utilized to select and place intermediate UPFs. Leyva-Pupo *et al.* [13] studied optimal UPF placement configuration, including the number of UPFs and the mapping relationship between users and UPFs. Multiple cost components are considered and an optimal stopping theory-based scheduling method is proposed, which places UPF according to latency and QoS thresholds. Taleb *et al.* [14] studied data anchor gateway placement for carrier cloud with two goals: 1) minimizing path between users and data anchor gateways and 2) optimizing performance in session mobility management. The authors point out that the two goals are conflicting: the former requires gateways deployed closer to the user while the latter needs a far enough deployment. The problem is solved by introducing an increasing log function and transforming it into a convex optimization. In [15], the UPF placement is formulated as a mixed-integer linear programming which is supposed to determine the number and locations of UPFs. The target is to reduce cost as well as guarantee the satisfaction of latency and reliability.

B. Placement of Edge Cloud

The placement of edge cloud, also called edge node or cloudlet, is widely studied. Santoyo-Gonzalez and Cervello-Pastor [16] put forward key parameters in the evaluation of edge cloud placement. They have shown that a poor placement scheme will inevitably result in low resource efficiency. Xu *et al.* [17] proposed a fast scalable heuristic to place heterogeneous edge clouds in a large-scale wireless metropolitan area network and reduce the latency of the system. Zhao *et al.* [18] focused on how to leverage software-defined networking in cloudlet placement. A ranking-based method is proposed to minimize latency with a low computational complexity. Fan and Ansari [19] focused on the tradeoff between latency and deployment cost. As the deployment cost is mainly affected by the number of edge servers, their algorithm seeks low-latency placement schemes with less edge servers. Chantre and da Fonseca [20] studied edge cloud placement in ultra-dense 5G network to achieve high reliability with low cost. The problem is formulated as a capacitated reliable facility location problem and a multiobjective evolutionary algorithm is proposed.

C. Summary

The aforementioned works are effective, however, these cannot be applied directly in the deployment of 6G edge infrastructures. Works mentioned in Section V-A study the placement of UPFs/gateways from the perspective of the network. The locations of edge clouds are not considered. On the contrary, works listed in Section V-B mainly focus on the deployment costs and quality-of-service parameters, especially latency, but neglect the influence brought by deploying UPFs at different locations. This work considers network and computation resources as a whole and jointly place UPFs and edge servers. The 6G network will be faced with multiple complicated scenarios with high quality of service demands. Only

by scheduling network and computation resources jointly, can these requirements be satisfied.

The most similar work is [21], in which the placement of edge servers and UPFs is jointly considered. However, the difference lies in two folds: first, the authors assume that UPFs are all placed at edge clouds, that is, the placement of UPF is choosing target location from edge clouds. This assumption results in the second difference. The authors adopt a two-stage method, which deploys edge clouds first and places UPFs based on the deployment scheme of edge clouds. In this article, the deployment relationship between edge clouds and UPFs is discussed in Section III-A, and we have shown that edge cloud should colocate with UPF but UPFs can be deployed individually. Besides, considering the interdependence between the placement of UPFs and edge clouds in terms of calculating latency, our proposed algorithm considers the two processes simultaneously instead of separating it into two stages.

VI. Conclusion

In this article, we study the problem of joint deployment of edge servers and UPFs in 6G scenarios to deal with challenges brought by cybertwin. We formulate it as an optimization problem and prove the NP-hardness. The network topology is considered in the placement. By analyzing the location relationship between edge clouds and UPFs, we simplify the problem by pruning the solution space. Then, we propose an effective algorithm which jointly considers the location of edge clouds and UPFs in every iteration. Simulations based on real-world data set and real data stream network emulator show that our algorithm outperforms the benchmark algorithms in terms of reducing average latency. In future work, we will focus on the service continuity in the highly dynamic 6G scenario with multiple types of access networks, including satellite communication, unmanned aerial vehicle communication, maritime communication, etc.

References

[1] S. Chen, Y.-C. Liang, S. Sun, S. Kang, W. Cheng, and M. Peng, "Vision, requirements, and technology trend of 6G: How to tackle the challenges of system coverage, capacity, user data-rate and movement speed," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 218–228, Apr. 2020.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," Jul. 2019. [Online]. Available: arXiv:1902.10265.

[3] Q. Yu, J. Ren, Y. Fu, Y. Li, and W. Zhang, "Cybertwin: An origin of next generation network architecture," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 111–117, Dec. 2019.

[4] Q. Yu, J. Ren, H. Zhou, and W. Zhang, "A cybertwin based network architecture for 6G," in *Proc. 6G Wireless Summit (6G SUMMIT)*, Mar. 2020, pp. 1–5.

[5] B. Hochner, "An embodied view of octopus neurobiology," *Current Biol.*, vol. 22, no. 20, pp. R887–R892, 2012.

[6] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security, and intelligence," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 126–132, Oct. 2020.

[7] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, "A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 668–695, 2nd Quart., 2021.

[8] *System Architecture for the 5G System*, 3GPP Standard (TS) 23.501, Dec. 2019. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/

[9] *GPRS Tunneling Protocol (GTP) Across the Gn and Gp Interface*, 3GPP Standard (TS) 29.060, Mar. 2020. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/29_series/29.060/

[10] W. Kiess and A. Khan, "Centralized vs. distributed: On the placement of gateway functionality in 5G cellular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 4788–4793.

[11] J. Costa-Requena, A. Poutanen, S. Vural, G. Kamel, C. Clark, and S. K. Roy, "SDN-based UPF for mobile backhaul network slicing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2018, pp. 48–53.

[12] S. Peters and M. A. Khan, "Anticipatory session management and user plane function placement for AI-driven beyond 5G networks," *Procedia Comput. Sci.*, vol. 160, pp. 214–223, Jan. 2019.

[13] I. Leyva-Pupo, C. Cervelló-Pastor, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic scheduling and optimal reconfiguration of UPF placement in 5G networks," in *Proc. 23rd Int. ACM Conf. Model. Anal. Simulat. Wireless Mobile Syst. (MSWiM)*, Nov. 2020, pp. 103–111.

[14] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3879–3884.

[15] I. Leyva-Pupo, C. Cervelló-Pastor, and A. Llorens-Carrodeguas, "Optimal placement of user plane functions in 5G Networks," in *Wired/Wireless Internet Communications*, (Lecture Notes in Computer Science), M. D. Felice, E. Natalizio, R. Bruno, and A. Kassler, Eds. Cham, Switzerland: Springer, 2019, pp. 105–117.

[16] A. Santoyo-Gonzalez and C. Cervello-Pastor, "Edge nodes infrastructure placement parameters for 5G networks," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, Paris, France, Oct. 2018, pp. 1–6.

[17] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *Proc. IEEE Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 570–578.

[18] L. Zhao, W. Sun, Y. Shi, and J. Liu, "Optimal placement of cloudlets for access delay minimization in SDN-based Internet of Things networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1334–1344, Apr. 2018.

[19] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[20] H. D. Chantre and N. L. S. da Fonseca, "Multi-objective optimization for edge device placement and reliable broadcasting in 5G NFV-based small cell networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2304–2317, Oct. 2018.

[21] I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "A framework for the joint placement of edge service infrastructure and user plane functions for 5G," *Sensors*, vol. 19, no. 18, p. 3975, 2019.

[22] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[23] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Boston, MA, USA: Springer, 1972, pp. 85–103.

[24] *Packet Radio System (GPRS) Tunneling Protocol User Plane (GTPv1-U)*, 3GPP Standard (TS) 29.281, Dec. 2019. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/29_series/29.281/

[25] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2018, pp. 66–73.

[26] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A Reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

Yuanzhe Li (Graduate Student Member, IEEE) received the Bachelor of Engineering degree in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology.

His research interests include mobile-edge computing, cloud computing, and service computing.

Xiao Ma (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2018.

She is currently a Postdoctoral Fellow with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing. From October 2016 to April 2017, she visited the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests include mobile cloud computing and mobile-edge computing.

Mengwei Xu received the bachelor's and Ph.D. degrees from Peking University, Beijing, China.

He is an Assistant Professor with the Computer Science Department, Beijing University of Posts and Telecommunications, Beijing. His research interests cover the broad areas of mobile computing, edge computing, and operating systems.

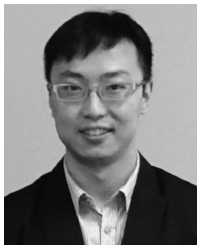
Ao Zhou (Member, IEEE) received the Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2015.

She is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, BUPT. She has published 20+ research papers. She played a key role at many international conferences. Her research interests include cloud computing and edge computing.

Qibo Sun (Member, IEEE) received the Ph.D. degree in communication and electronic system from Beijing University of Posts and Telecommunication (BUPT), Beijing, China, in 2002.

He is currently an Associate Professor with BUPT. His research interests include services computing, Internet of Things, and network security.

Dr. Sun is a member of the China Computer Federation.



Ning Zhang (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2015.

He was a Postdoctoral Research Fellow with the University of Waterloo and the University of Toronto, Toronto, ON, Canada. He is an Associate Professor with the University of Windsor, Windsor, ON, Canada.

Dr. Zhang received the Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, and *Journal of Communications and Information Networks* in 2018, IEEE ICC in 2019, IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and IEEE ICCC in 2019, respectively. He serves as an Associate Editor for IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE ACCESS, *IET Communications*, and *Vehicular Communications* (Elsevier), and a Guest Editor for several international journals, such as IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He also serves/served as a track chair for several international conferences and a co-chair for several international workshops.

Shangguang Wang (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011.

He is currently a Professor with the School of Computing, BUPT, where he is a Vice-Director of the State Key Laboratory of Networking and Switching Technology. He is also with the Network Communication Research Center, Peng Cheng Laboratory, Shenzhen, China. He has published more than 150 papers. His research interests include service computing, cloud computing, and mobile-edge computing.

Dr. Wang served as the General Chair or the TPC Chair for IEEE EDGE 2020, IEEE CLOUD 2020, IEEE SAGC 2020, IEEE EDGE 2018, and IEEE ICFCE 2017, and the Vice-Chair for IEEE Technical Committee on Services Computing from 2015 to 2018. He has been serving as the Executive Vice-Chair for IEEE Technical Committee on Services Computing since 2021 and the Vice-Chair for IEEE Technical Committee on Cloud Computing since 2020.