TGPOA: Task and Geography Privacy-Preserving Offloading Algorithm in 6G Network*

Aoran Huang and Huachun Zhou †

Beijing Jiaotong University, Beijing, China
{24110088,hchzhou}@bjtu.edu.cn

Abstract

Mobile network technology not only provides cloud-side technical support for 6G networks, but also brings a large number of complex and changeable tasks. In the face of frequent node switching and task updates, it is necessary to relieve the pressure of computing, storage and communication through task offloading and cache enablement. However, due to the untrustworthiness of mobile access nodes, the task data privacy and geographic location privacy at the edge are in danger. How to achieve high-performance offloading while protecting edge privacy has become an urgent need to find a solution. To this end, this paper proposes a task and geographic privacy protection offloading algorithm (TGPOA) for 6G network scenarios. By classifying tasks and selecting the offloaded server location, TGPOA calculates the privacy entropy of location and task respectively. The experimental results of constrained Markov decision process (CMDP) show that TGPOA has significant advantages in resource utilization, time overhead and privacy regardless of how the task volume and cache hit rate change.

Keywords: 6G network, offloading, cache, privacy-preserving, mobile network, constrained Markov decision process (CMDP).

1 Introduction

The development of mobile network technology provides rich network scenarios and task processing capabilities for 6G networks [1], and also brings variable access nodes and devices [2]. These changeable scenarios, complex tasks and diverse and frequently switched nodes put forward higher requirements for the computing power and processing time of 6G networks [3]. By deploying cloud edge nodes to offload the tasks of accessing devices, the computing and delay load of the network can be effectively alleviated [4]. In this process, it is necessary to prevent the random unloading of too many or too few tasks to edge nodes for processing, because this will lead to high time costs and inefficient use of edge node resources [5]. This violates the expected purpose of placing cloud edge nodes.

In recent research, the design of offloading algorithm based on network performance index can efficiently realize the efficient utilization of 6G network resources [6]. Taking network performance such as delay and resource utilization as optimization indicators [7], a certain amount of workload is offloaded to the edge nodes, and the remaining tasks are still processed by the cloud, which can effectively improve the network's ability to deal with the changeable tasks brought by mobile nodes. In addition, considering the frequent node switching brought by mobile network technology, it is inefficient to re-analyze each new access task and load resources [8], because a large number of new access nodes request tasks with high similarity. By

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 1, December 16-18, 2025, Sapporo , Japan. © The copyright of this paper remains with the author(s).

[†] Corresponding author

leveraging cached results of prior executions, the processing time of each task can be significantly reduced [9]. Furthermore, enabling caching throughout the entire offloading pipeline further accelerates execution and lowers latency [10].

However, when offloading tasks in cache-enabled 6G networks, task data privacy and device location privacy will be violated because the deployed or randomly accessed edge nodes are untrusted [11]. If the entire task is offloaded, all data will be exposed to the edge nodes, increasing the risk of data leakage. If the nearest edge node is always selected for offloading to reduce the transmission cost, it is easier for the adversary to determine the location of the device. Privacy protection issues cannot be ignored [12], especially considering the frequency of use of mobile nodes in public [13] and shared [14] scenarios in 6G networks. To this end, two problems need to be solved: 1. Design an appropriate offloading strategy to maximize the use of network resources and improve network performance; 2. Protect task privacy and geographic location privacy during offloading.

In this paper, we propose a task and geographic location privacy protection offloading algorithm (TGPOA) to solve the privacy protection problem in the offloading process of cache-supported 6G cloud-side networks. Firstly, three unloading paths are designed and the unloading model is given. Then, a resource utilization model is constructed based on task and cache hit rate. Based on the time cost model of transmission between cloud, edge and equipment, the time cost model is constructed to optimize the joint goal of network resources and network performance. Based on the probability of geographical location and task type being leaked, a privacy entropy model is constructed, and the task and geographical location privacy entropy values are introduced into the optimization process as constraints to achieve the goal of privacy-preserving offloading. Finally, the constrained Markov decision process (CMDP) is used to solve the problem with privacy entropy as the constraint condition, and the final offloading strategy is obtained. The contributions of this paper are as follows :

- Aiming at the problem of frequent node switching and task update caused by mobile network technology in 6G network, an offloading algorithm combining resource utilization and transmission delay is proposed. Three offloading paths are designed, and a resource utilization model and a time cost model are constructed for the cache-enabled scenario.

- According to the leakage probability of node location and task type under different offloading paths, the privacy entropy model is constructed, and the privacy entropy is solved by CMDP as a constraint, and the final offloading strategy model is obtained.

- The proposed algorithm is simulated, and the performance comparison with other unloading methods and unloading path selection is analyzed in detail. The experimental results show that the proposed method can efficiently utilize network resources and has excellent time cost, and can effectively guarantee task and geographical location privacy.

The subsequent structure of this article is as follows. Chapter 2 reviews the recent work on privacy-preserving offloading. In Chapter 3, the detailed construction method of the system model is introduced. Chapter 4 gives a privacy-preserving offloading algorithm based on the constructed model. Chapter 5 has carried on the simulation experiment. Finally, in Chapter 6, the full text is summarized.

2 Related Work

This summary reviews the recent work on offloading strategies and privacy protection. With the development of AI technology, more and more work uses AI technology to design offloading strategies. In order to support the use of generative AI meta-universe applications in the IoT, W. Zeng et al. [15] proposed a multi-modal parallel offloading framework that distributes multi-modal content to multiple servers to adapt to the limitations of communication and computing resources in IoT. H. Li et al. [16] transformed the offloading process into a Markov decision process through deep learning method in the dual dynamic scenarios of task generation and vehicle movement, and realized the offloading strategy model in the multi-access vehicle networking scenario.

However, in the above work, the cache demand problem caused by the change of mobile nodes is not discussed. L. Liu et al. [17] constructed a multi-user computing offloading and wireless cache resource allocation problem for this problem. In order to minimize the delay, the deep deterministic policy gradient is used to solve the problem, and the reasonable allocation of cache resources is realized. Taking into account the different timeliness of caching and offloading in the joint optimization process. J. Zhang et al. [18] constructed a two-time scale resource allocation problem, and proposed a computing strategy including a service caching strategy and a hierarchical action value function, which realized that the edge cached service provides computing resources to the request task immediately.

Although this part of the work adds the consideration of caching, it does not consider the privacy security problem caused by the frequent switching of mobile nodes in the 6G network scenario. D. Han et al. [19] proposed a swarm intelligence model in the 6G IoT scenario. Based on blockchain technology, through intelligent collaboration and expansion of terminal, edge and cloud resources, the privacy security in the 6G IoT interaction process is guaranteed. Z. Wang et al. [20] proposed a task offloading framework suitable for multi-server access for geographical location privacy, and designed a location perturbation mechanism to ensure the optimal offloading strategy in the perturbation area. It is worth noting that there is no common consideration of offloading, caching, task privacy and geographic location privacy in the above work. Compared with the above work, this paper considers four aspects at the same time. The joint optimization of delay and resource utilization machine also adds the consideration of task and geographical location privacy security as a constraint.

3 System Model

As shown in Fig. 1, the system model consists of a central cloud, edge nodes, and edge devices. For the central cloud and edge nodes, there is memory and a server composed of virtual machines (VMs). The central cloud is represented by a set $\mathcal{C}$. Denote the edge nodes collection as $\mathcal{F} = \{F_1, \cdots, F_s\}$, where s is the total amount of edge nodes in $\mathcal{F}$. For edge node F_i, its VMs set is expressed as $\mathcal{V} = \{v_{1_i}, \cdots, v_{m_i}\}$, where m_i is the number of VMs in edge node F_i. edge devices are represented by set $\mathcal{D} = \{D_1, \cdots, D_N\}$, where N is the number of edge devices. Assume that each device has only one task, represented by a set $\mathcal{T} = \{T_1, \cdots, T_N\}$. Denote deployment strategy for each task as $\mathcal{S} = \{S_1, \cdots, S_N\}$, where $S_n \in \mathcal{F} \cup \mathcal{C} \cup \emptyset$ indicates that the task is offloaded to the edge node or the central cloud or does not need to be offloaded.

3.1 Privacy-preserving Offloading Model

The offloading path is segregated into three types (i.e., P_1, P_2 and P_3) to preserve the device's geographic location's privacy. P_1: The device chooses the closest edge node to send an offload request to, and the edge node checks its cache. If it does not hit, the edge node asks the device for input from the device, which then sends the task to the edge node for processing and provides the result to the device. Instead, the device receives the result straight from the edge

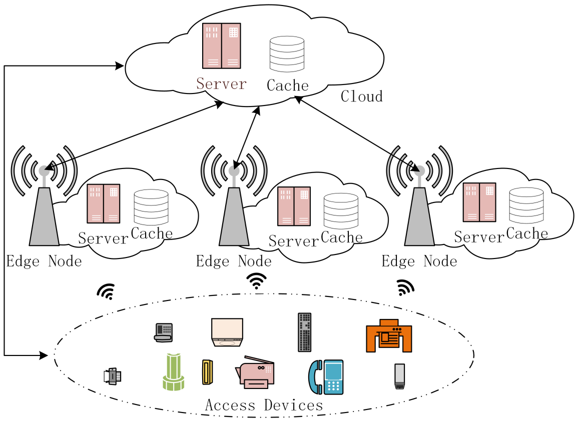

Figure 1: Three-layer system architecture: Tasks are offloaded from the device layer to the edge and cloud. Caching provides task caching, and servers provide computing resources.

node after processing it. P_2: The device chooses edge nodes at random, and the next steps are identical to P_1. P_3: The device directly offloads the task to the cloud for processing and receives the returning result (assuming a hit rate of 1 in the cloud).

3.2 Resource Utility Model

The resource utility is reflected by the VM utilization rate in the edge node. In edge node F_i, it can be expressed as

$$\eta_{F_i} = \sum_{n=1}^{N} \frac{\alpha_i^n}{m_i}, \tag{1}$$

where α_i^n denotes the number of VMs required to process task n in the i-th edge node. For the entire network, the number of VMs used in the edge node is expressed as

$$N_{VM} = \sum_{i=1}^{s} \sum_{n=1}^{N} \alpha_i^n \delta_i^n, \tag{2}$$

where δ_i^n indicates whether the task n is processed in the i-th edge node.

$$\delta_i^n = \begin{cases} 1, & \text{if } S_i = F_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The average utilization rate of VMs in all edge nodes is expressed as

$$\eta_F = \frac{1}{s} \sum_{i=1}^{s} \sum_{n=1}^{N} \frac{\alpha_i^n \delta_i^n}{m_i}. \tag{4}$$

For the central cloud, it is considered that its computing resources are sufficient, regardless of its VMs utilization.

3.3 Privacy Entropy Model

For the edge device n, if the nearest edge node is selected, the geographic area of the device can be inferred, so the entropy is 0. Instead, if the cloud or random edge node is selected, the entropy is 1. Therefore, the geographical location privacy entropy model of device n can be expressed as

$$H_L(n) = \begin{cases} 0, & \text{select nearest edge node,} \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

The average geographic location privacy entropy of all devices can be expressed as

$$H_L = \frac{1}{N} \sum_{n=1}^{N} H_L(n). \tag{6}$$

For the task T_n in the device n, it is divided into K types, and the transmission probability of each type obeys the Poisson distribution with

$$P(T_{n,k}) = \frac{\lambda^{T_{n,k}}}{T_{n,k}!} e^{-\lambda}. \tag{7}$$

Therefore, the privacy entropy of task T_n can be calculated as

$$H_{T_n} = -\sum_{k=1}^{K} P(T_{n,k}) \log P(T_{n,k}). \tag{8}$$

The average privacy entropy of all tasks can be expressed as

$$H_T = \frac{1}{N} \sum_{n=1}^{N} H_{T_n}. \tag{9}$$

3.4 Time Cost Model

The offloading method is P_1 or P_2: Assume that r_1 and r_2, the request's transmission time and cache check time, are both fixed numbers. For device D_n with strategy S_n selected, the time taken for task T_n to be transmitted to the edge node F_i is expressed as

$$\psi_n^{DF}(S_n) = \sum_{k=1}^{K} \frac{d_{n,k}}{C_{D2F}} P(T_{n,k}), \tag{10}$$

where $d_{n,k}$ is the size of task T_n of type k and C_{D2F} is the transmission rate from edge device to edge node. The processing time of the task T_n in the edge node is expressed as

$$\psi_n^{edge}(S_n) = \sum_{k=1}^{K} \frac{|d_{n,k}|}{\beta_i^{n,k} \omega} P(T_{n,k}), \tag{11}$$

where $|d_{n,k}|$ denotes the size of data to be processed by task T_n, $\beta_i^{n,k}$ denotes the number of VMs required to process the k-th type of task T_n, and ω denotes the computing power of each VM. The duration required to return the device n with the processed results is

$$\psi_n^{FD}(S_n) = \frac{\varphi_n}{C_{F2D}}, \tag{12}$$

where φ_n is the size of the data that the task T_n returns to the device n after processing and C_{F2D} is the transmission rate from edge node to edge device. Therefore, the time cost of offloading task T_n to edge node F_i is expressed as

$$\begin{gathered}\psi_n^F(S_n) = r_1 + r_2 + \psi_n^{edge}(S_n) + \psi_n^{FD}(S_n) \\ +(1-\varepsilon_i^n)(r_1 + \psi_n^{DF}(S_n)),\end{gathered} \tag{13}$$

where ε_i^n indicates whether the offloaded task T_n is hit by the caching in the edge node F_i. 1 indicates hit and 0 indicates no hit.

For the offloading method P_3, assuming that the cache hit rate in the cloud is 1, the task can be transmitted directly to the cloud for processing. Meanwhile, the computing resources of the central cloud are sufficient, so the task processing time is set to a fixed value r_3. The time cost of transferring task n to the cloud and returning devices after processing can be expressed as

$$\psi_n^{DC}(S_n) = \sum_{k=1}^{K} \frac{d_{n,k}}{C_{D2C}} P(T_{n,k}), \psi_n^{CD}(S_n) = \frac{\varphi_n}{C_{C2D}}, \tag{14}$$

where C_{D2C} is the transmission rate from edge device to cloud and C_{C2D} is the transmission rate from cloud to edge device. Therefore, the time cost required for task T_n to select the cloud as the offloading destination is expressed as

$$\psi_n^C(S_n) = \psi_n^{DC}(S_n) + r_3 + \psi_n^{CD}(S_n). \tag{15}$$

The time cost of task n is expressed as

$$\psi_n(S_n) = \begin{cases} \psi_n^F(S_n), & \text{if } S_n \in \mathcal{F} \cup \emptyset, \\ \psi_n^C(S_n), & \text{if } S_n \in \mathcal{C}. \end{cases} \tag{16}$$

The total time cost is expressed as

$$\psi(\mathcal{S}) = \sum_{n=1}^{N} \psi_n(S_n). \tag{17}$$

4 Privacy-preserving Offloading Algorithm

4.1 State Space and Action Space

The state space is defined as

$$\mathbb{M} = \mathbb{I} \times \mathbb{Q}, \tag{18}$$

where $\mathbb{I} = \{0, 1, 2\}$ denotes the selection of the nearest edge node, the random edge node and the central cloud as the offloading destination, and $\mathbb{Q} = \{0, 1\}$ denotes whether the device needs to perform task input. The action space is defined as $\mathbb{A} = \{0, 1, 2\}$, which means to offload the task to the nearest edge node, random edge node and central cloud. For sets $\mathbb{M}$, $\mathbb{I}$, $\mathbb{Q}$ and $\mathbb{A}$, exist $M \in \mathbb{M}$, $Q \in \mathbb{Q}$, $I \in \mathbb{I}$ and $A \in \mathbb{A}$ indicates that M, I, Q, and A are the corresponding elements in sets $\mathbb{M}$, $\mathbb{I}$, $\mathbb{Q}$ and $\mathbb{A}$, respectively.

4.2 Transition Probability

The $\mathbb{I}$ of the next stage is determined by the $\mathbb{A}$ of the previous stage. The cache hit rate is different in different servers, and $\mathbb{Q}$ is also affected by $\mathbb{A}$. Therefore, the transition probability from the current M to the next stage M' is expressed as

$$P[M'|M,A] = P[I'|I,A] \times P[Q'|Q,A], \tag{19}$$

where $P[I'|I,A]$ denotes the effect of the current A on the next state I', which can be expressed as

$$P[I'|I,A] = \begin{cases} 1, & \text{if } I' = A, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

It is assumed that the cache hit rate in the edge node is p_f, and the cache hit rate in the cloud is 1. The transition probability from the current Q to the next stage Q' is expressed as

$$P[Q'|Q, A \in \{0,1\}] = \begin{cases} p_f, & \text{if } Q' = 1, \\ 1 - p_f, & \text{if } Q' = 0. \end{cases} \tag{21}$$

$$P[Q'|Q, A = 2] = 1. \tag{22}$$

Considering the state space and action space, the resource utilization of edge node F_i can be expressed as

$$\eta_{F_i} = \begin{cases} \sum\limits_{n=1}^{N} \frac{\alpha_i^n}{m_i}, & \text{if } C \in \{0,1\}. \\ 0, & \text{if } C = 2. \end{cases} \tag{23}$$

The average utilization rate can be expressed as

$$\eta_F(\mathcal{S}) = \frac{1}{s} \sum_{i=1}^{s} \eta_{F_i}. \tag{24}$$

Similarly, the time cost can be expressed as

$$\psi_n(S_n) = \begin{cases} r_1 + r_2 + \psi_n^{edge}(S_n) + \psi_n^{FD}(S_n), \\ \qquad \text{if } C \in \{0,1\}, Q = 1, A \in \{0,1\}, \\ 2r_1 + r_2 + \psi_n^{edge}(S_n) + \psi_n^{FD}(S_n) + \psi_n^{DF}(S_n), \\ \qquad \text{if } C \in \{0,1\}, Q = 0, A \in \{0,1\}, \\ \psi_n^{DC}(S_n) + \psi_n^{CD}(S_n) + r_3, \\ \qquad \text{if } C = 2, Q = 1, A = 2. \end{cases} \tag{25}$$

For geographic location privacy entropy, (5) is expressed as

$$H_L(n) = \begin{cases} 0, & \text{if } I = 0, \\ 1, & \text{if } I \in \{1,2\}. \end{cases} \tag{26}$$

4.3 CMDP-Based Solution

Based on CMDP, the model is constructed to maximize resource utilization and minimize time cost under the premise of following constraints.

$$\max \eta_F(\mathcal{S}), \min \psi(\mathcal{S}). \tag{27}$$

$$\begin{aligned} s.t. \quad & H_T \leq \omega_T, \\ & H_L \leq \omega_T. \end{aligned} \tag{28}$$

Where H_T and H_L denote for the corresponding lower bounds of data privacy entropy and location privacy entropy. Strategy $\mathcal{S}$ can represent a set of state space and action space of all edge nodes in a time slot. As $\mathrm{P}(S, A)$ stands for the stationary probability of state S and action A, The equivalent LP model of CMDP model is defined as

$$\max \sum_S \sum_A \mathrm{P}(S, A) \eta_F(\mathcal{S}), \tag{29}$$

$$\min \sum_S \sum_A \mathrm{P}(S, A) \psi(\mathcal{S}). \tag{30}$$

The constraint is

$$\sum_S \sum_A \mathrm{P}(S, A) H_T \geq \omega_T, \tag{31a}$$

$$\sum_S \sum_A \mathrm{P}(S, A) H_L \geq \omega_L, \tag{31b}$$

$$\sum_A \mathrm{P}(S', A) = \sum_S \sum_A \mathrm{P}(S, A) P[S'|S, A], \tag{31c}$$

$$\sum_S \sum_A \mathrm{P}(S, A) = 1, \tag{31d}$$

$$\sum_{k=1}^{K} P_{n,k} = 1, \tag{31e}$$

$$\mathrm{P}(S, A), P_{n,k} \geq 0, \tag{31f}$$

$$0 \leq \sum_{k=1}^{K} \beta_i^{n,k} \leq m_i. \tag{31g}$$

The (29) and (30) represent the maximum resource utilization and the minimum time cost. Constraints (31a) and (31b) represent the minimum lower bounds of geographic location privacy entropy and task data privacy entropy. Formula (31c) is the constraint condition to satisfy the transition probability. With the constraints in (31d), (31e) and (31f), the probability characteristics are preserved. Formula (31g) guarantees that the number of VMs requested does not exceed the total amount. A universal stochastic policy [21] is used to solve the CMDP model. The optimal strategy $\pi^*(S, A)$ is obtained by the stationary probability $P^*(S, A)$, which is the solution of the equivalent LP model.

$$\pi^*(S, A) = \frac{P^*(S,A)}{\sum_{A'} P^*(S,A')}, \text{ for } S \in \mathcal{S} \quad \sum_{A'} P^*(S, A') > 0. \tag{32}$$

In the above process, the construction process of the whole model and the LP solution process are combined as the strategy solution stage. After the optimal strategy solution is obtained in the solution stage, the strategy execution stage is entered. The tasks of each access device are executed in the order of device state acquisition, action execution and state update, and finally the privacy-preserving offloading strategy of all tasks is obtained.

4.4 Complexity Analysis

The complexity analysis mainly includes two parts: the complexity of the CMDP equivalence and LP solving process in the strategy solving stage and the decision complexity in the strategy execution stage.

For the complexity of the policy solving phase, the resource utilization, time cost and privacy entropy under each state-action are calculated. A simple algebraic operation of N devices, E nodes, and T task types is designed, so the overhead of the model construction process can be expressed as $O(N \times E \times T)$. For the complexity of the constructed LP solution, the number of variables of LP can be expressed as $\tilde{S} \times |\mathbb{A}|$, where $\tilde{S} = N \times S$ represents the total number of states. The interior point method is used to solve the polynomial time. In the process of solving the interior point method, it is necessary to carry out sub-iteration $O(\sqrt{\tilde{S}\,|\mathbb{A}|})$ times, and the calculation amount of each iteration is $O((\tilde{S}\,|\mathbb{A}|)^2\tilde{S})$. Therefore, the complexity of solving the worst case can be expressed as $O(\sqrt{\tilde{S}\,|\mathbb{A}|}(\tilde{S}\,|\mathbb{A}|)^2\tilde{S}) = O((\tilde{S}\,|\mathbb{A}|)^{5}\!/2\tilde{S})$. In the actual processing, $\tilde{S} \times |\mathbb{A}|$ and $\tilde{S}$ are of the same order, so they can be regarded as the same order of magnitude merged into $O((\tilde{S}\,|\mathbb{A}|)^{7}\!/2)$, and the remaining $\frac{1}{2}$ is omitted as a low-order, so the final complexity can be approximated as $O((\tilde{S} \times |\mathbb{A}|)^3)$, representing the cubic polynomial solution time.

For the strategy execution phase, after obtaining the optimal decision strategy, each decision needs to collect the state of the current device, and its complexity is $O(1)$; the optimal action is selected according to the strategy, and the complexity of the process is $O(1)$; finally, the state is updated, and the complexity is still $O(1)$. Therefore, the complexity of the complete policy execution phase is fixed to $O(1 + 1 + 1) = O(1)$.

5 Performance Evaluation

5.1 Experimental Setting

In this section, by comparing with the three schemes, it is proved that TGPOA is excellent. ACloud: All tasks are processed in the cloud. ACFG: All tasks are processed in the closed edge node. ARFG: All tasks are processed in random edge nodes. The default parameter settings of the experiment are as follows: The number of tasks is 500, the hit rate of edge nodes is 0.5, the number of edge nodes is 15, the number of VMs per edge node is 10, the transmission rate from the task to the edge node is inversely proportional to the distance between them, the transmission rate from the task to the cloud is a fixed constant, and the size of the transmission task is within interval $[0.5\text{GB}, 3\text{GB}]$.

5.2 Experimental Results

Fig. (2) shows the trend of resource utilization and the number of tasks. The TGPOA scheme can always maintain the resource utilization at a high level and is positively correlated with the number of tasks. This is because the more tasks there are, the more computing resources are required. The computing resources in scheme TGPOA are provided by edge nodes and cloud servers, so the resource utilization rate of edge nodes slowly increases with the number of tasks. Due to the fact that all tasks in ACFG and ARFG schemes are processed by edge nodes, when the number of tasks reaches a certain level, the resource utilization rate is 1. This situation usually represents insufficient computing resources in the edge node, so the TGPOA scheme

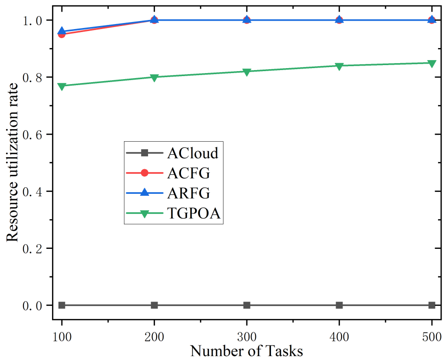

Figure 2: Comparison of resource utilization rates among four schemes with 100-500 tasks

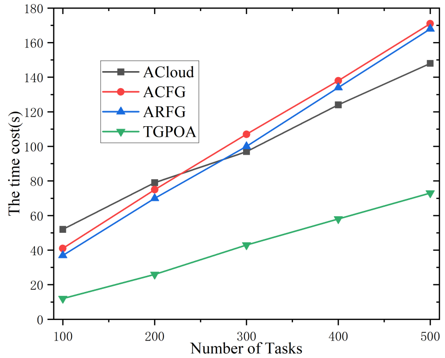

Figure 3: Comparison of time cost among four schemes with 100-500 tasks.

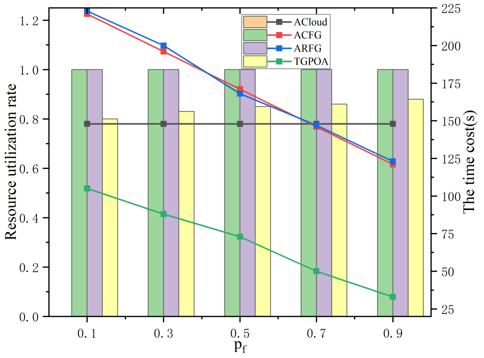

Figure 4: Comparison of resource utilization rates and time cost among four schemes with a hit rate of 0.1-0.9.

has more advantages. All tasks in Scheme ACloud are processed by the cloud, so the resource utilization rate of the edge node is 0.

Fig. (3) shows the trend of time cost under different task numbers. Compared with others, TGPOA scheme has a great advantage in time cost, and the growth rate of time cost with

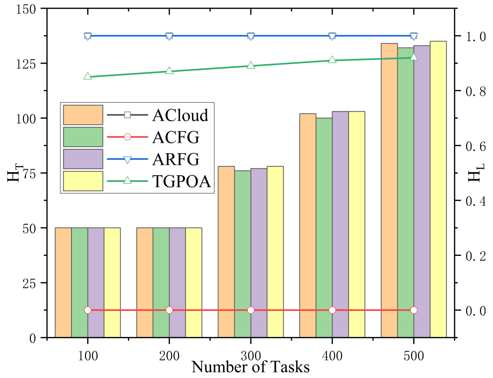

Figure 5: Comparison of privacy entropy H_T and H_L among four schemes with 100-500 tasks.

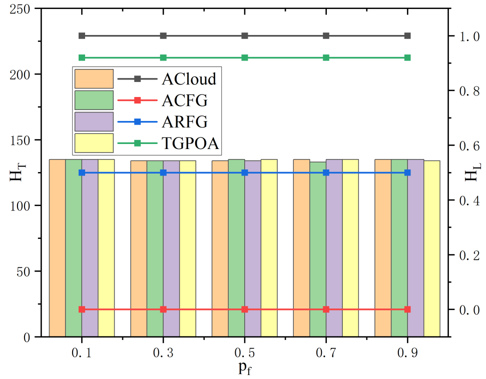

Figure 6: Comparison of privacy entropy H_T and H_L among four schemes with a hit rate of 0.1-0.9.

task number is also lower than that of the three schemes. Since of the fact that the number of tasks dealt with in edge nodes and clouds can be reasonably allocated based on the necessary computing resources and time costs for different task numbers. When there are few tasks, processing in the cloud results in higher transmission costs than it saves in processing time at the cloud center. The processing time advantage provided by the cloud center is more noticeable as the number of tasks rises. As a result, the curve of ACFG and ARFG initially lower than the curve of ACloud but becoming higher as the number of tasks rises.

Fig. (4) shows the trend of resource utilization and time cost under different hit rate. As the hit rate increases, the processing time of edge nodes decreases, and the speed advantage of cloud center processing is indirectly weakened. More tasks are offloaded to edge nodes for processing, so resource utilization increases with an increase in hit rate. For schemes ACFG and ARFG, the resource utilization rate is always 1. For scheme ACloud, the resource utilization rate is always 0. For the time cost, processing more tasks by edge nodes reduces the number of tasks offloaded to the cloud, resulting in a decrease in both processing time and transmission time. Therefore, for schemes TGPOA, ACFG, and ARFG, time cost is negatively correlated with hit rate. For scheme ACloud, its time cost is not affected by the hit rate of edge nodes.

Fig. (5) shows the trend of privacy entropy H_T and H_L under different task numbers.

Obviously, the H_T of all four schemes is positively correlated with the number of tasks. When the number of tasks is 100 and 200, it does not reach the minimum privacy entropy lower limit, so it is a fixed value. Regardless of the number of tasks, the H_L values for both scheme ANFG and ARFG are 1, and the H_L values for scheme ACloud are 0. The H_L of TGPOA is positively correlated with the number of tasks and is remained at a high level. Fig. (6) represents that the change in hit rate does not affect privacy entropy H_T and H_L. Because privacy entropy is related to the distribution probability of task types, and is independent of whether the offloading destination is cloud or edge. Scheme TGPOA always maintains the value of H_L close to 1, indicating that it can provide a high level of geographic location privacy protection.

6 Conclusion

In this paper, we propose an offloading algorithm called TGPOA, which is used to offload tasks and geographic location privacy protection in cache-enabled 6G networks. Based on the minimum threshold of task privacy entropy and geographic location privacy entropy, the access device makes an offloading decision to maximize the resource utilization of the edge node and optimize the time cost. Finally, the problem of policy selection is solved by establishing a constrained Markov decision process. The simulation results show that compared with all-cloud and all-edge processing, the proposed scheme has obvious advantages in time cost and resource utilization, and can maintain a high level of task data and location privacy entropy.

7 Acknowledgments

This paper is supported by National Key R&D Program of China under Grant No.2018YFA0701 604 and NSFC under Grant No.62341102.

References

[1] C. -X. Wang et al., “On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds,” in *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905-974, Secondquarter 2023.

[2] Y. -J. Liu et al., “A Survey of Integrating Generative Artificial Intelligence and 6G Mobile Services: Architectures, Solutions, Technologies and Outlooks,” in *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 3, pp. 1334-1356, June 2025.

[3] C. Sun, X. Wu, X. Li, Q. Fan, J. Wen and V. C. M. Leung, “Cooperative Computation Offloading for Multi-Access Edge Computing in 6G Mobile Networks via Soft Actor Critic,” in *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5601-5614, Nov.-Dec. 2024.

[4] W. Lan, K. Chen, Y. Li, J. Cao and Y. Sahni, “Deep Reinforcement Learning for Privacy-Preserving Task Offloading in Integrated Satellite-Terrestrial Networks,” in *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9678-9691, Oct. 2024.

[5] Xu Chen et al., “Revenue-Oriented Optimal Service Offloading Based on Fog-Cloud Collaboration in SD-WAN Enabled Manufacturing Networks”, in *IEEE Transactions on Network Science and Engineering*, vol.12, no.2, pp.1237-1249, 2025.

[6] X. Wang et al., “Evolutionary Game Caching Resource Allocation Strategy for 6G Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 74, no. 3, pp. 4993-5005, March 2025.

[7] G. Chen and X. Huang, “IRS-Enhanced Parallel Computing and Partial Offloading for Latency Sensitive MEC,” in *IEEE Wireless Communications Letters*, vol. 13, no. 11, pp. 2980-2984, Nov. 2024.

[8] L. Zhao et al., “MESON: A Mobility-Aware Dependent Task Offloading Scheme for Urban Vehicular Edge Computing,” in *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4259-4272, May 2024.

[9] X. Dai, S. Tian, H. Liu, Z. Li, H. Jiang and Q. Deng, “Joint Optimization of Offloading and Caching in Full-Duplex-Enabled Edge Computing Networks,” in *IEEE Transactions on Mobile Computing*, vol. 24, no. 8, pp. 6996-7011, Aug. 2025.

[10] C. Fang et al., “Joint Task Offloading and Content Caching for NOMA-Aided Cloud-Edge-Terminal Cooperation Networks,” in *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 15586-15600, Oct. 2024.

[11] C. N. Hadjicostis and A. D. Domínguez-García, “Trustworthy Distributed Average Consensus Based on Locally Assessed Trust Evaluations,” in *IEEE Transactions on Automatic Control*, vol. 70, no. 1, pp. 371-386, Jan. 2025.

[12] X. Zhu et al., “Enabling Intelligent Connectivity: A Survey of Secure ISAC in 6G Networks,” in *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 748-781, April 2025.

[13] L. Qi, X. Xu, X. Wu, Q. Ni, Y. Yuan and X. Zhang, “Digital-Twin-Enabled 6G Mobile Network Video Streaming Using Mobile Crowdsourcing,” in *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3161-3174, Oct. 2023.

[14] W. Bai and A. Huang, “Privacy-Preserving Collaborative Sharing for Sharing Economy in Fog-Enhanced IoT,” in *IEEE Access*, vol. 11, pp. 95295-95306, 2023.

[15] W. Zeng et al., “Generative AI-Aided Multimodal Parallel Offloading for AIGC Metaverse Service in IoT Networks,” in *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13273-13285, 15 May15, 2025.

[16] H. Li, C. Chen, H. Shan, P. Li, Y. C. Chang and H. Song, “Deep Deterministic Policy Gradient-Based Algorithm for Computation Offloading in IoV,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2522-2533, March 2024.

[17] L. Liu and Z. Chen, “Joint Optimization of Multiuser Computation Offloading and Wireless-Caching Resource Allocation With Linearly Related Requests in Vehicular Edge Computing System,” in *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1534-1547, 1 Jan.1, 2024.

[18] J. Zhang, Y. Shen, Y. Wang, X. Zhang and J. Wang, “Dual-Timescale Resource Allocation for Collaborative Service Caching and Computation Offloading in IoT Systems,” in *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1735-1746, Feb. 2023.

[19] D. Han, Y. Liu, R. Cao, H. Gao and Y. Lu, “A Lightweight Blockchain Architecture with Smart Collaborative and Progressive Evolution for Privacy-Preserving 6G IoT,” in *IEEE Wireless Communications*, vol. 31, no. 5, pp. 148-154, October 2024.

[20] Z. Wang et al., “Location Privacy-Aware Task Offloading in Mobile Edge Computing,” in *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2269-2283, March 2024.

[21] M. M. Moghaddam, M. H. Manshaei, M. N. Soorki, W. Saad, M. Goudarzi and D. Niyato, “On Coordination of Smart Grid and Cooperative Cloud Providers,” in *IEEE Systems Journal*, vol. 15, no. 1, pp. 672-683, March 2021.