Traffic Data Scheduling of Frequent Application Sets for Task Offloading in Multi-access Edge Computing

Yifeng Hu
College of Intelligence and Computing
Tianjin University
Tianjin, China
cardistry@tju.edu.cn

Tie Qiu*
College of Intelligence and Computing
Tianjin University
Tianjin, China
qiutie@ieee.org

Jiancheng Chi
College of Intelligence and Computing
Tianjin University
Tianjin, China
chijiancheng.tju@gmail.com

Xuan Yang
State Grid Electronic Commerce Co., Ltd.
Beijing, China
xyzt_2021@163.com

Zimu Wang
State Grid Ecommerce Technology Co., Ltd.
Tianjin, China
wzzbm@163.com

Wenguang Li
State Grid Ecommerce Technology Co., Ltd.
Tianjin, China
mclwgmc@163.com

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838928

***Abstract*—Frequently task offloading in the Multi-access Edge Computing (MEC) system takes up a lot of network resources, which leads to serious network congestion problems. To deal with such problems, applying data pricing strategies in task offloading to schedule traffic data is perceived as a promising solution for Internet service providers (ISPs). However, the traditional application-oriented data pricing strategies do not consider user satisfaction of the task executing process, resulting in lower ISP profit. In this paper, a novel scheme named App-set Usage Patterns-Aware task offloading scheme (AUPA) is proposed to alleviate the tension between traffic data supply and user satisfaction. We mine the sequential patterns in frequent application sets to extract temporal association rules. Then these association rules are used to design a smart data pricing strategy to guide the task offloading decision. Finally, we formulate the ISP's profit maximization problem as a nonlinear programming (NLP) problem based on partial offloading, and we simulate the scheduling processuse using the Stackelberg game model. The performance of our solution is evaluated in terms of ISP's profit, consumers' surplus, capacity utilization, and traffic efficiency. The results show that our scheme significantly improves the ISP's profit by about 20% while ensuring capacity constraints compared with other baseline schemes.**

***Index Terms*—task offloading, multi-access edge computing, data pricing, sequential pattern mining**

I. Introduction

With the explosive growth of smart devices on the Internet-of-Things (IoT) networks, computation-intensive applications such as speech recognition, natural language processing and reality augmentation [1]–[5] are gradually fading into our sight. Multi-access Edge Computing (MEC) offloads tasks of IoT devices to the nearby edge servers, and provides low-latency as well as flexible computing and communication services [6], [7]. Along with the growing popularity of IoT applications, task offloading requests in MEC systems are generating additional data exchanges (including task executing information and note state information, etc.) between users and servers, which would cause unbalanced network load and trsaffic congestion [8].

The task offloading requests generated by IoT users can be affected by many dynamic factors such as time and location. In order to solve the network congestion problem and improve system performance, previous works have optimized the network resource allocation of MEC systems from three aspects: intelligent spectrum utilization [9], dedicated spectrum [2] and traffic scheduling for Internet Services Providers (ISPs) [10]. However, a MEC system can be a small-scale scenario with insufficient users and usage records like a factory or a community, which means a lack of data to analyze application usage behaviors. Another question is that these works ignore the impact of user satisfaction on system performance, such as user habits and characteristics of applications. As a result, a small-scale task offloding scenario can lead to incorrect estimation of application usage expectations.

In this paper, a novel scheme named App-set Usage Patterns-Aware task offloading scheme (AUPA) is proposed to optimize the ISP's profit without exceeding the capacity constraints of a MEC system. The main contributions of this paper are summarized as follows:

- We design a MEC task offloading framework based on app-set usage behavior. We mine the sequential patterns in frequent application sets to extract temporal association rules and design associatd-app usage expectations to quantify user satisfaction.

*Corresponding author

This work is supported by the Joint Funds of the National Natural Science Foundation of China (No. U2001204), Tianjin Science Foundation for Distinguished Young Scholars (No. 20JCJQJC00250) and Key R&D Program of Tianjin (No. 20YFZCGX01150).

- We propose a smart data pricing strategy based on Stackelberg game to maximize ISP's profit and we formulate the ISP's profit maximization problem as a nonlinear programming (NLP) problem.
- Simulation experiments based on a real world dataset are conducted to validate the performance of our proposed scheme. ISP's profits, consumers' surplus, capacity utilization, and traffic efficiency are evaluated to prove the significant improvement of our scheme.

II. Related Work

In this section, the related work is introduced from two perspectives of traffic data scheduling: task offloading and data pricing.

In a MEC system, ENs offload computing tasks to the MEC server to reduce the execution delay and energy consumption, which take up a lot of network resources, and it can lead to serious network congestion problems. In order to optimize the network load of the MEC system, previous works have extensively studied the netword load balancing problem in the task offloading process to schedule the system traffic data. Feng *et al.* [11] jointly considered task offloading, sub-carrier allocation, and power allocation to maintain the arrival rate of the task buffer and keep the amount of network communication data at a stable level. The problem of minimizing service traffic load of edge caching was studied in [12], combined with user distribution prediction algorithm and service caching strategy, it can reduce the traffic load of service migration. To address the lack of communication resources during task offloading processing, some works have turned to the study of intelligent spectrum utilization in edge computing. Saleem *et al.* [2] considered the spectrum sharing between cellular network and D2D devices, instead of cellular network, they schedules traffic data in D2D spectrum to reduce network congestions. However, these works did not pay attention to the influence of users' own behaviors in traffic data scheduling process, the network load still suffers from heavy latency due to the tight relationship between supply and demand of traffic data during peak hours and hotspots.

In order to take user behavior into account for traffic data scheduling and managing traffic congestion, economic mechanisms have been adopted to study the impact of dynamic factors on the demand for traffic data [9], [13]. Ma *et al.* [13] proposed a time and location aware mobile data pricing scheme, which divides the entire network into several regions and optimizes the prices independently for each region. Yin *et al.* [9] further proposed a two-dimensional data pricing scheme based on time and application category awareness, which divides application categories according to the core functions of applications, optimizes users' satisfaction and ISPs' profits respectively.

Although the previous works in [2]–[5], [9], [11]–[14] have been devoted to study the network load balancing problem during the traffic data scheduling process, there are still several deficiencies in the research on the optimization of traffic data allocation and user satisfaction with limited network resources.

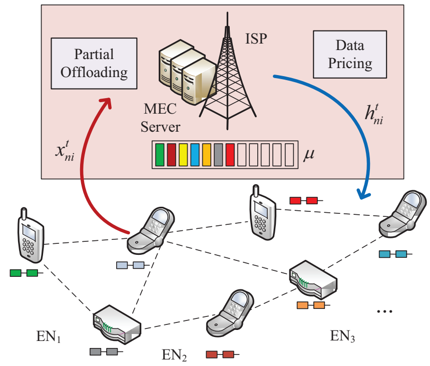


Fig. 1. Illustration of partial offloading and data pricing in the MEC system.

First, ENs in different MEC systems are heterogeneous and the consumption of traffic data can be affected by many dynamic factors. Second, user satisfaction of task offloading is related to ENs' application usage habits and task types. Taking these issues into account can reduce the estimation errors of task offloading expectations. Therefore, the MEC system must study app-set usage behavior to derive a more fine-grained task offloading scheme based on traffic data pricing.

III. System Model and Problem Formulation

This section describes the task offloading behavior and the partial offloading model in the MEC system. Then we formulate the task offloading problem as an ISP's profit maximization problem.

A. Task Offloading Behavior

In order to make full use of the time information in the task offloading dataset, we consider the sequence of applications in the execution process as important note state information. According to the research in [15], we can learn that there may be a specific correlation between different applications, which could promote or restrict the usage of each other. In fact, in MEC systems that enable task offloading, some computation-intensive tasks (such as pedestrian recognition [16]) require multiple applications with a progressive relationship to complete. To correctly discover and quantify these associations, first, we need to identify the user's complete process of each task offloading behavior and the combination of applications included in one task offloading behavior.

We assume that the behavior of each user using a smart device once is regarded as a complete task offloading behavior, and mining the task execution relationship from the applications involved in each task offloading behavior. Although the dataset does not give an accurate division of smart device usage behavior, we can use phone sessions(i.e., separate times that people actually pick up their mobile phone to use it) to divide usage records in the dataset based on the survey in [17]. We estimate the average session duration of users based on the average daily minutes spent on the mobile phone and sessions they use their phones per day in the survey, and thus classify each user's app usage records in a week into different mobile sessions. In particular, long intervals or location shifts are also considered as different mobile sessions.

B. Partial Offloading Model

We consider a MEC system integrated with a base station and a MEC server. The ISP provides traffic data through the base station and let μ be the capacity constraint of the base station. Define the set $\mathcal{N} = \{1, 2, \ldots, N\}$ as the set of ENs with limited computation resources and delay sensitive computation intensive tasks to perform, let $n \in \mathcal{N}$ represents an EN in the set. In order to release the computation burden, ENs can offload computing tasks to the MEC server, which can benefit from parallelism between local and remote executions [18]. Similar to the previous works in [19] and [20], we consider the data partitioned oriented applications [21]. For such applications, the capacity of task input data is proportional to the amount of computation and can be arbitrarily partitioned for parallel processing due to bit-wise independence [22]. In addition, it is assumed that the return data of a task is small and negligible.

In our model, we treat applications called by the task as the smallest unit in the task execution process and we expand the task sequence to the application sequence. The task execution process adopts the time slot model. In each time slot t, the set $\mathcal{I} = \{1, 2, \ldots, I_n\}$ is defined as the application sequence executed by EN n, let $i \in \mathcal{I}$ represents an application in the set, and ϕ_{ni}^t is the capacity of input data when EN n executes the i th application at time slot t , h_{ni}^t is the unit price for EN n to offload the i th application at time slot t. Assuming that the input data of application i can be partitioned, we introduce the parameter ω_{ni}^t to represent the offloading ratio:

$$\omega_{ni}^t = \frac{x_{ni}^t}{\phi_{ni}^t}. \tag{1}$$

Where $\omega_{ni}^t \in [0, 1]$, x_{ni}^t represents the amout of actual offloading data. Therefore, part of the input data is computed locally without consuming traffic data in the network, and the other part needs to be offloaded to the MEC server for remote execution with a certain price.

C. Profit maximization problem

In our work, ENs consume traffic data in the MEC system to offload computation-intensive tasks to the MEC server for remote execution. In contrast, the MEC server must ensure that the total data consumption in each time slot t cannot exceed the capacity constraint μ. The data consumption of the MEC system satisfies the following constraint:

$$\sum_{n=1}^{N}\sum_{i=1}^{I_n} x_{ni}^t \leq \mu. \tag{2}$$

In this paper, the ISP regulates user's task offloading decision by adjusting traffic data price for offloading tasks. Our goal is to maximize the ISP's profit while balancing the traffic load of the MEC system. Therefore, we formulate the offloading decision problem as a profit maximization problem. Based on (2), the ISP maximizes profit by solving the following optimization problem:

$$\begin{aligned} \max_{h_{ni}^t} \quad & \prod = \sum_{n=1}^{N}\sum_{i=1}^{I_n} h_{ni}^t x_{ni}^t, \\ s.t. \quad & \sum_{n=1}^{N}\sum_{i=1}^{I_n} x_{ni}^t \leq \mu \quad \forall t. \\ & 0 \leq x_{ni}^t \leq \phi_{ni}^t \quad \forall x_{ni}^t \end{aligned} \tag{3}$$

A simple illustration of considered scenario is shown in Fig. 1, which depicts that EN n pay price h_{ni}^t to the ISP and the MEC server compute partial task data x_{ni}^t without exceeding capacity constraint μ. So far, we have described the task offloading dataset and the partial offloading model in the MEC system, and discussed the ISP's profit maximization problem. The following section proposes a data pricing strategy based on a two-stage Stackelberg game to adjust users' offloading decisions and describe our task offloading scheme to solve the ISP's profit maximization problem.

IV. App-set Usage Patterns-Aware Task Offloading Scheme

According to our analysis, in addition to the time and application category, offloading requests are related to the sequential patterns of the task execution process. Therefore, in this section, we mine the sequential patterns of time sequences through EN application usage history and extract association rules. Then we design a task offloading scheme by considering the influence of association rules between the applications called by the offloading task.

A. Sequential Pattern Mining and Temporal Association Rules

We use Cspade algorithm [23] to acquire temporal assocation rules from the task offloading dataset. Compared with other frequent pattern mining algorithms (such as Apriori and FP-Growth), the Cspade algorithm considers the order of transactions. It is used to extract sequential patterns for computation-intensive tasks involving multiple applications. The Cspade algorithm uses sequence ID (SID) to indicate the offloading behavior of each task. Each SID is divided into multiple element IDs (EIDs) by time slots. In one SID, applications with the same EID are in a parallel relationship, while the other is in a progressive relationship. The evaluation criteria of association rules include support, confidence and lift.

B. App-set Offloading Expectation

We define the app-set offloading exception of ENs in the MEC system based on the user service evaluation model in [9], [14]. Assume that each EN has an evaluation for a specific application, i.e. the offloading expectation. In each task offloading behavior, EN will use one or more applications in several time slots. During every time slot, EN decides whether to offload the application input data and the offloading ratio based on the offloading expectation and the actual traffic data price h. Only when the expectation of the application is higher than or equal to the actual traffic data price, ENs choose to offload the application, otherwise the application will be executed locally. We define the time slot $[t-1, t] (t = 1, \ldots, T)$ to be the unit in which the traffic data price of an application remains unchanged. Before introducing the offloading expectation model, based on [14], we first define the offloading expectation of EN n for the unit traffic data of application i in time slot t:

$$\lambda_{ni}^t = \rho_1 \sigma_{ni} + (1 - \rho_1 \sum_m s_{ni}^m). \tag{4}$$

Where ρ_1 is the weighting factor, σ_{ni} is the frequency of EN n using application i, and s_{ni}^m represents the support of association rule m related to EN n and application i, which satisfies $l_{ni}^m > 1$, l_{ni}^m is the lift of the association rule m related to EN n and application i. The offloading expectation of unit traffic data depends on the usage frequency of the application itself and association rules related to the application. After normalization, the importance of the two factors can be determined by the weighting factor ρ_1. It should be noted that only an association rule with a lift greater than 1 can increase the offloading expectation, which means the applications promote each other. Next, we define the satisfaction function of EN n for an application i in time slot t:

$$f_{ni}^t(\omega_{ni}^t) = (\omega_{ni}^t)^{\rho_2 \beta_i + (1-\rho_2) tanh(\delta_{ni}^t)}, \tag{5}$$

$$\beta_i = \frac{\overline{\phi_i}}{\phi_{max}}, \tag{6}$$

$$\delta_{ni}^t = \sum_m \theta_{ni}^m s_{ni}^m c_{ni}^m l_{ni}^m. \tag{7}$$

Here, ρ_2 in (6) is the weighting factor. In (7), β_i represents the traffic-consumption speed of application i and is positively related to the computation complexity of the application itself. In partial offloading, we define it as the ratio of $\overline{\phi_i}$ and ϕ_{max}, where $\overline{\phi_i}$ is the average traffic consumption of the application i in the task offloading dataset, and ϕ_{max} is the maximum value of average traffic consumption of all applications. δ_{ni}^t represents the association factor, where $\theta_{ni}^m \in \{0, 1\}$ is a binary indicator variable used to represent whether there is an association rule m involving the current application i and applications that have been used in the previous time slot, c_{ni}^m is the confidence of association rule m related to EN n and application i.

In our satisfaction function, EN's satisfaction of executing an application is related to the offloading ratio. In contrast, the speed at which the satisfaction decreases is related to the computational complexity of the application and applications that EN has just used in the same SID. The higher the computational complexity of the application, the faster the satisfaction decreases. In addition, we use the association factor to indicate the degree of dependence between applications in the association rules. The more significant the association factor, the more likely EN to execute this set of applications. Therefore, when association rules include the currently used application and the previously used application, EN's satisfaction with the current application will also be affected. We use an activation function to normalize the correlation factor, and then use the weighting factor ρ_2 to balance the weight between it and the computational complexity of the application.

From (4-7), we define EN n's offloading expectation of application i in time slot t as follows:

$$Y_{ni}^t = \lambda_{ni}^t \phi_{ni}^t f_{ni}^t(\omega_{ni}^t). \tag{8}$$

Where ϕ_{ni}^t indicates that EN n's input data of application i in time slot t, and it is also the maximum offloading capacity.

C. Frequent Application set Data Pricing for Task Offloading

In this work, our goal is to design a data pricing scheme to adjust the offloading decision of ENs in the MEC system, and maximize the profit of the ISP under the premise that the total data consumption of any time slot does not exceed the capacity constraint μ. According to the ISP's profit maximization problem mentioned in section III, we need to model the relationship between the capacity of offloading data x in the system and the data price h during time slot t.

A two-stage Stackelberg game model is adopted in [9] and [14] to simulate the relationship between data pricing and application usage. First, the ISP, as the leader, sets a price for traffic data. Second, ENs decide whether to use the offloading service for the given price. However, they did not consider a system with multiple independent users and continuous offloading requests. From (8), we define the utilization maximization function of EN n at time slot t in the MEC system:

$$\begin{aligned} \max_{x_{ni}^t} \quad & U_n(x_{ni}^t) = \sum_{i=1}^{I_n} \lambda_{ni}^t \phi_{ni}^t f_{ni} \left(\frac{x_{ni}^t}{\phi_{ni}^t} \right) - \sum_{i=1}^{I_n} h_{ni}^t x_{ni}^t, \\ s.t. \quad & 0 \le x_{ni}^t \le \phi_{ni}^t, \quad 1 \le i \le I_n. \end{aligned} \tag{9}$$

From (9), we use backward induction to obtain the relationship between the capacity of offloading data and the data price and provide a new profit-maximizing offloading scheme:

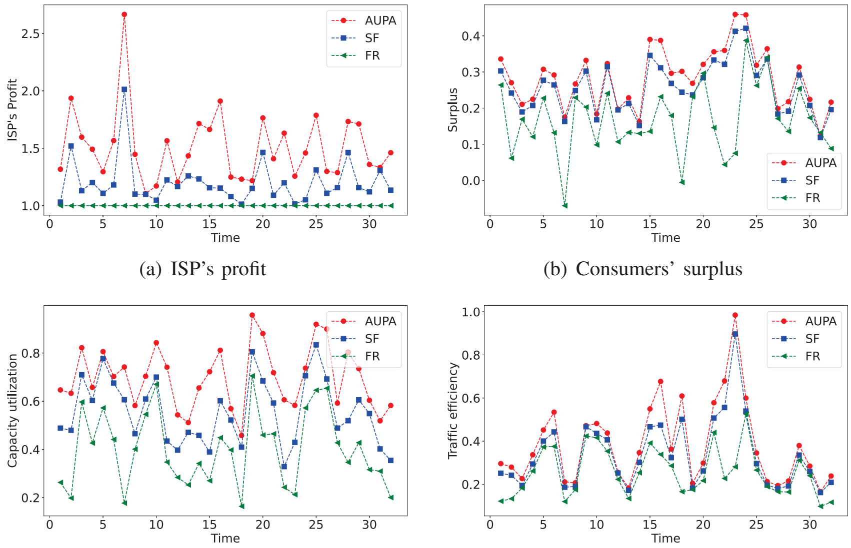
$$\begin{aligned} \max_{h_{ni}^t} \quad & \prod = \sum_{n=1}^{N} \sum_{i=1}^{I_n} h_{ni}^t x_{ni}^{t*}, \\ s.t. \quad & \sum_{n=1}^{N} \sum_{i=1}^{I_n} x_{ni}^{t*} \le \mu \quad \forall t. \end{aligned} \tag{10}$$

This is a nonlinear programming problem and can be solved using efficient nonlinear programming techniques.

V. Numerical Results

In this section, we evaluate the performance of our proposed scheme by analyzing the numerical results and designing the following two benchmark schemes for comparison:

- **Single-app usage Frequency aware (SF):** The scheme adopted in a single-server-single-user system. Regardless of the association rules between tasks and the heterogeneity of ENs. Offloading expectations and satisfaction are



(a) ISP's profit

(b) Consumers' surplus

(c) Capacity utilization

(d) Traffic efficiency

Fig. 2. The performance evaluation of AUPA, SF and FR schemes.

only related to the single-app usage frequency and the computation complexity of the application.

- **Flat-Rate (FR):** The price of traffic data is uniform within the same time slot, and the pricing scheme does not distinguish between application categories.

Specifically, we evaluate the performance of our scheme from the following four aspects:

- **ISP's profit** is the ratio of the total profit of the ISP to the total profit of the FR scheme over a period of time.
- **Consumers' surplus (Cs)** is the difference between consumers' average valuation of services and the service fee charged by the ISP in time slot t [14], in our work it is defined as

$$Cs^t = \frac{\sum_{n=1}^{N}\sum_{i=1}^{I_n}(\lambda_{ni}^t - h_{ni}^t)}{\sum_{n=1}^{N} I_n}. \tag{11}$$

- **Capacity utilization (Cu)** is the ratio of the total offloading volume to the capacity constraint in time slot t, defined as

$$Cu^t = \frac{\sum_{n=1}^{N}\sum_{i=1}^{I_n} x_{ni}^t}{\mu}. \tag{12}$$

- **Traffic efficiency (Te)** is the ratio of the ENs' average offloading expectation to the average offloading capacity in time slot t, defined as

$$Te^t = \frac{\sum_{n=1}^{N}\sum_{i=1}^{I_n} \lambda_{ni}^t \phi_{ni}^t f_{ni}^t(\omega_{ni}^t)}{\sum_{n=1}^{N}\sum_{i=1}^{I_n} x_{ni}^t}. \tag{13}$$

The MEC system is considered within the range of a base station. We randomly select task offloading records of $N = 25$ ENs from the task offloading dataset and extract their application usage association rules based on their historical offloading records. In order to make full use of sufficient usage records, we take the 8 most active time periods (from 9 a.m. to 5 p.m.) in a day, and use 15 minutes as a time unit to simulate 32 sets of task offloading process. The capacity constraint μ is set to 0.8 times the maximum time period of traffic data consumption μ_{max} to ensure the existence of peak hours.

We use the Tsinghua App Usage Dataset provided in [24] to simulate ENs' task execution process in the MEC system. We assume that each application request represents a task offloading request, and the data packet length represents the capacity of task input. In the case of complete offloading, the data packet size requested by the application is equal to the task's traffic data demand. Therefore, we established a connection between the application usage dataset and the task offloading dataset.

We compared the performance indicators of these schemes based on real-world datasets, including the ISP's profit, consumers' surplus, capacity utilization, and traffic efficiency. As shown in Fig. 2(a), the APUA and SF schemes acquire higher profit of ISP compared with the FR scheme, which shows that designing data pricing schemes based on application types is an effective way to increase ISP's profit. The AUPA scheme has been further improved ISP's profit compared to the SF scheme, which means that designing task offloading schemes while considering ENs' habits of using certain applications can incentive ENs to pay higher price for traffic data.

Consumers' surplus is the difference between the maximum price a consumer is willing to pay and the actual price they do pay. If a consumer is willing to pay more for a unit of a good than the current asking price, they are getting more benefit from the purchased product than they would if the price was their maximum willingness to pay. According to the analysis of consumers' surplus in Fig. 2(b), the AUPA scheme performs better when more related association rules are applied. Generally speaking, there are more types of applications used, the correlation between applications is more obvious. In extreme cases, there is no assocition rule between application. The consumers' surplus of AUPA is still higher than SF because it computes the application usage frequency of each user separately, which means more precise offloading expectations.

Compared with SF and FR, the AUPA scheme can make full use of system capacity because AUPA considers app-set usage patterns and the single application usage frequency. As shown in Fig. 2(c), during the peak hours(i.e. the capacity utilization is near 1), the AUPA scheme can adjust the price for each EN in a fine-grained manner while the SF scheme can only adjust the price of a certain application for all ENs, which has significantly improved capacity utilization. When the capacity is sufficient, the AUPA scheme can offer incentive strategies for different ENs separately, making full use of the capacity and increasing the ISP's profit.

A common definition of traffic efficiency can be met as the extent to which a certain traffic data input can meet the data demand of user in a MEC system. In Fig. 2(d), the ISP increases data prices to obtain higher profit and reduces network load using AUPA scheme. On the other hand, SF and FR ignore the heterogeneity of ENs, which is not conducive to improving traffic efficiency.

According to the results of the simulation experiments, the AUPA scheme is more suitable for the MEC system than the SF and FR schemes. Under the premise of ensuring the system capacity constraint, our proposed scheme improves the total ISP's profit by about 20%. Especially in peak hours and when many types of applications cooperate, it performs better.

VI. Conclusion

In this paper, data pricing strategies are combined with task offloading of the MEC system to make full use of system traffic capacity and optimize the ISP's profit. The task execution pattern mining is used to guide the decision process, and the task offloading scheme based on the pricing of the frequent application set is designed. Numerical results show that the task offloading scheme based on frequent application set data pricing improves ISP profits without exceeding the capacity constraint.

References

[1] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[2] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for d2d-enabled partial computation offloading in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4472–4486, 2020.

[3] G. Hu, Y. Jia, and Z. Chen, "Multi-user computation offloading with d2d for mobile edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[4] Y. He, J. Ren, G. Yu, and Y. Cai, "D2d communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1750–1763, 2019.

[5] R. Chai, J. Lin, M. Chen, and Q. Chen, "Task execution cost minimization-based joint computation offloading and resource allocation for cellular d2d mec systems," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4110–4121, 2019.

[6] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22, 2013.

[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] J. Yin, Y. Fan, T. Xia, Y. Li, X. Chen, Z. Zhou, and X. Chen, "Mobile app usage patterns aware smart data pricing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 4, pp. 645–654, 2020.

[10] J. Yan, S. Bi, L. Duan, and Y.-J. A. Zhang, "Pricing-driven service caching and task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4495–4512, 2021.

[11] J. Feng, L. Zhao, J. Du, X. Chu, and F. R. Yu, "Computation offloading and resource allocation in d2d-enabled mobile edge computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[12] Q. Xie, Q. Wang, N. Yu, H. Huang, and X. Jia, "Dynamic service caching in mobile edge networks," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2018, pp. 73–79.

[13] Q. Ma, Y.-F. Liu, and J. Huang, "Time and location aware mobile data pricing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2599–2613, 2016.

[14] L. Zhang, W. Wu, and D. Wang, "Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 700–708.

[15] J. Huang, F. Xu, Y. Lin, and Y. Li, "On the understanding of interdependency of mobile app usage," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2017, pp. 471–475.

[16] S. Sthapit, J. Thompson, N. M. Robertson, and J. R. Hopgood, "Computational load balancing on the edge in absence of cloud and fog," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1499–1512, 2019.

[17] Putting a Finger on Our Phone Obsession. [Online]. Available: {https://blog.dscout.com/mobile-touches}

[18] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[19] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, "Partial offloading for latency minimization in mobile-edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[20] U. Saleem, Y. Liu, S. Jangsher, and Y. Li, "Performance guaranteed partial offloading for mobile edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[21] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.

[22] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[23] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine learning*, vol. 42, no. 1, pp. 31–60, 2001.

[24] D. Yu, Y. Li, F. Xu, P. Zhang, and V. Kostakos, "Smartphone app usage prediction using points of interest," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, p. 174, 2018.