A Game-Based Incentive-Driven Offloading Framework for Dispersed Computing

Hongjia Wu, Jiangtian Nie, *Member, IEEE*, Zehui Xiong, *Member, IEEE*, Zhiping Cai, Tongqing Zhou, Chau Yuen, *Fellow, IEEE*, and Dusit Niyato, *Fellow, IEEE*

***Abstract*— The popularization of smart Internet of Things (IoT) devices has facilitated the development of fog/edge computing. However, these infrastructure-based service paradigms may fail to complete tasks successfully due to computation and communication overload, or damage in challenging scenarios such as disasters or traffic jams. Noticing that a crowd of devices with considerable idle resources could be available, we investigate the problems of addressing the computation and communication unavailability with peer assistance in this work. To this end, we propose a dispersed service framework for resource-exhausted scenarios that adaptively offloads users' data to available network computation points. However, the users may not be able to achieve the offloading due to geographical hindrances. Consequently, the relay is introduced as a bridge for data offloading between the users and the network computation points. Furthermore, a game-based incentive-driven offloading mechanism is designed by analyzing and balancing the cost and gain factors of three main entities (users, relays, and network computation points). Considering the interactions among the entities, a two-level Stackelberg game is established for efficiently allocating potential computation resource, as well as balancing the utility conflicts due to the data offloading. Given the hierarchical interaction structure, the upper level game involves network computation points as followers and the relay as a leader, while the lower level game includes the relay as a follower and users as leaders. Moreover, to facilitate applicability in large-scale scenarios with multiple relays, we decompose multiple relays into multiple single relay problems using a tripartite matching strategy that assigns appropriate relays to users and network computation points. The simulation results demonstrate the effectiveness of the proposed game-based incentive-driven mechanism and show that it outperforms the baselines in terms of the overall utilities of the involved entities and the average energy consumption of users.**

***Index Terms*— Incentive-driven, IoT, mobile computing, task offloading, network computation points, idle resources, Stackelberg game.**

Manuscript received 29 September 2022; revised 26 February 2023; accepted 31 March 2023. Date of publication 13 April 2023; date of current version 17 July 2023. This work was supported by the National Natural Science Foundation of China (Nos. 62072465, 62172155, 62102425, 62102429, U22B2005), the Science and Technology Innovation Program of Hunan Province (Nos. 2022RC3061, 2021RC2071), and the Natural Science Foundation of Hunan Province (No. 2022JJ40564). The work is supported by the National Research Foundation (NRF) and Infocomm Media Development Authority under the Future Communications Research Development Programme (FCP). The work is also supported by the SUTD SRG-ISTD-2021-165, the SUTD-ZJU IDEA Grant (SUTD-ZJU (VP) 202102), and the Ministry of Education, Singapore, under its SUTD Kickstarter Initiative (SKI 20210204). Furthermore, the work is supported by A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Industry Alignment Fund-Pre Positioning (IAF-PP) (Grant No. A19D6a0053). In addition, the work is supported in part by the DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019), under Energy Research Test-Bed and Industry Partnership Funding Initiative, part of the Energy Grid (EG) 2.0 programme, and under DesCartes and the Campus for Research Excellence and Technological Enterprise (CREATE) programme. The associate editor coordinating the review of this article and approving it for publication was N. Pappas. *(Corresponding author: Zhiping Cai.)*

Hongjia Wu, Zhiping Cai, and Tongqing Zhou are with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: wuhongjia19@nudt.edu.cn; zpcai@nudt.edu.cn; zhoutongqing@nudt.edu.cn).

Jiangtian Nie and Dusit Niyato are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jnie001@e.ntu.edu.sg; dniyato@ntu.edu.sg).

Zehui Xiong is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Chau Yuen is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: chau.yuen@ntu.edu.sg).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2023.3266833.

Digital Object Identifier 10.1109/TCOMM.2023.3266833

I. Introduction

THE growth of communication bandwidth and the popularization of IoT devices have together promoted the emergence of various mobile computing applications with complex tasks and intelligent analysis, including agriculture, healthcare, finance, and other fields. According to the "Data Age 2025" white paper published by International Data Corporation (IDC) [1], the global data amount is predicted to reach 175ZB in 2025, which is 10 times more than today. To explore the value of this data for more scalable computing applications, new computing paradigms have been proposed. Among them, fog/edge computing [2] is envisioned as a promising and feasible technology to alleviate the computation and communication pressure. By migrating computation from the cloud to edge servers, it can provide IoT devices with low latency and context-aware services. However, in real-world scenarios [3], practically challenging situations (e.g., traffic jams, natural disasters) are common. Fixed infrastructures used for communication or computation are often overloaded or even unavailable, making it impossible to effectively perform computation tasks in the corresponding area. For example, when traffic jams occur, cameras at intersections need to capture more information (e.g., video streams) than usual for traffic flow analysis and monitoring [4], resulting in inefficient road traffic management services.

Introducing additional facilities such as edge servers [5], [6], [7] can be a viable option to solve the above problems. However, the surge in traffic is often accidental

and only occurs within a short period, causing computation resource wastage and limiting the scalability of the services. In disaster scenarios, the infrastructure may fail to provide services or even be destroyed, and it is difficult to deploy new network infrastructures in time. There are also some efforts [8], [9], [10], [11] using the delay tolerant network (DTN) to perform small-scale message transmission when infrastructure-based communication is unavailable. However, current common computation tasks, such as AR and real-time video analysis, are computation-intensive and require stringent response latencies. The adoption of DTN will make bandwidth a serious bottleneck and it is difficult to support the offloading and distribution of complex computation tasks for a large number of users. The aforementioned existing schemes are not adequate in providing scalable computation and communication support in many challenging scenarios. Therefore, it is essential to develop a flexible, scalable, and adaptable service framework that can effectively supplement cloud and edge computing to overcome the problem of urgent computation and communication resource exhaustion.

Recently, dispersed computing [12] has gained attention to migrate computation from the edge to local devices. This computing paradigm takes advantage of massive and dispersed IoT devices with idle resources to satisfy the computation demands of nearby devices. Based on the concept of the dispersed paradigm, we find that the increasing number of IoT devices involved in challenging scenarios can provide more available computation and communication resource, implicitly helping to relieve the pitfalls of fog/edge computing mentioned above. Moreover, when fixed service infrastructures and facilities are unavailable, inter-device assistance can be used to support computational applications. Inspired by these insights, this paper attempts to explore the idle computation resources scattered across devices in challenging scenarios. For large-scale outdoor events such as celebrity concerts and festival parties, mobile phones or computers charged at night in nearby residential areas, and even devices that are not in use on-site can be fully or partially utilized to relieve the pressure on infrastructure.

From a high-level view, we propose a dispersed service framework to facilitate computation and communication resource utilization that involves three main entities: network computation points (NCPs), relays, and users. Relays (such as drones or communication vehicles) are introduced as a bridge between the users generating offloading requests and the NCPs with idle resources. However, designing such a framework faces the following challenges:

- NCPs with idle resources are typically self-interested personal devices. Therefore, the challenge is how to incentivize them to actively participate in offloading services to alleviate computational load.
- NCPs exhibit variations in computation capabilities, and the challenge here is how to manage these idle resources in a strategic manner.
- The next subsequent challenge is how to effectively maximize the overall utilities of the three main entities while balancing their conflicts of interest.
- Another significant challenge is how to efficiently allocate appropriate relays to users and NCPs for achieving the best matching in large-scale scenarios with multiple relays.

To address the challenges mentioned above, we propose a novel game-based incentive-driven dispersed offloading mechanism that can handle both single-relay and multi-relay scenarios. For single-relay scenarios, we create a two-level Stackelberg game that incentivizes NCPs to exploit idle resources in providing offloading services to users. For multi-relay scenarios, we introduce a tripartite matching strategy based on a dual matching game to assign appropriate relays to users and NCPs, thus decomposing the complex multi-relay problem into multiple single-relay problems in a tractable way.

In summary, the main contributions of this paper are as follows:

- We propose a dispersed computing framework for offloading that addresses urgent computation and communication resource exhaustion. Compared to existing schemes [5], [6], [7], [8], [9], [10], [11], the proposal focuses on the use of potential idle resources and can respond to the unavailability of infrastructure due to man-made and natural disasters, which is an important and key contribution of this work.
- Compared to existing dispersed computing [13], [14], [15], [16] techniques without considering the selfishness of NCPs, we focus on multi-user and multi-NCP situations, incentivizing NCPs to participate in offloading services in a trading pattern. Furthermore, a two-level stackelberg game is established to balance the conflicting interests of the three entities (i.e., users, relays and NCPs), and the existence and uniqueness of the equilibrium are proved.
- UAVs and unmanned communication vehicles are introduced as relays to cope with large-scale scenarios and long-distance communication. Moreover, matching games are used to achieve efficient and logical relay allocation and a tripartite matching strategy is proposed.
- We evaluate the performance of the proposed algorithms compared with baselines. The results verify the feasibility and advantages of our proposal in terms of overall utilities. Moreover, the proposed scheme achieves the best energy-saving effect on users' average energy consumption.

The rest of this paper is organized as follows. Section II provides the literature review. Section III describes the system model and the game formulation. In Section IV, we analyze the interactions between the three main entities for both single-relay and multi-relay scenarios. Section V presents the performance evaluation and Section VI concludes the paper.

II. Related Work

Mobile computation offloading technology is receiving attention to alleviate the overloaded computation demands. For example, Chen et al. [17] studied task offloading in ultra-dense networks. To minimize the delay and energy consumption of

users, an efficient offloading scheme was proposed. A computing offloading method called COM [18] was designed to support cloud-edge computing of the Internet of Things. Zhao et al. [19] proposed a collaborative method based on mobile edge computing and cloud computing to offload services to automobiles in vehicular networks. However, the technologies mentioned above cannot effectively respond to the sudden increase in computing load caused by accidental emergencies due to their dependencies on infrastructures. To improve the flexibility of offloading, Zhang et al. [20] investigated a UAV-assisted multi-access edge computing system consisting of a flying UAV and a ground base station serving multiple ground mobile devices. Mithun et al. [21] designed a task offloading strategy in a UAV-based MEC system, where the end-user offloads computation intensive tasks to the UAV. The utilization of UAVs has increased the flexibility of edge offloading, yet their expensive cost cannot be overlooked. In addition, the above solutions are limited to dedicated equipment, ignoring the large amount of potentially available idle resources.

For emergency scenarios, **delay tolerant network** (DTN) is applied. For example, Mao et al. [8] designed a routing protocol called scheduling probability, which uses encounter history and transitivity. Taking into account the bandwidth and storage limitations, Wu et al. [9] proposed a photo selection algorithm for DTN to maximize the value of photos delivered to the command center. Datta et al. [10] developed a scheme to dynamically update the POIs list based on the current photo metadata. By sending important photos of POIs, the consumption of bandwidth, energy, and node storage is reduced, which is suitable for disaster recovery scenarios. A delay-tolerant network disaster information system based on a mobile cloud wireless grid was proposed in [11], which has high mobility and multi-platform integration capability. The above studies are feasible in resource-limited environments. However, due to the high intermittency of nodes in the DTN network, as well as the limited bandwidth and storage, the prompt data delivery cannot be guaranteed. Furthermore, it is limited to short-term emergency handling in specific situations and is not suitable for future use in large-scale scenarios (e.g., concerts).

With the enhancement of computation power of smart devices, the idea of **dispersed computing** has drawn attention [13], [14], [15]. Dispersed computing [12] is regarded as a computing paradigm that fully explores and utilizes idle resources in the network, providing services to users as a supplement when the edge is overloaded. For example, a run-time scheduling software worker for decentralized computing was developed in [14]. It can deploy pipeline computation in the form of directed acyclic graphs across multiple geographically dispersed computation points. Ghosh et al. [15] designed a container orchestration architecture for dispersed computing. The system automatically and efficiently assigns tasks among a set of network computation points. Wu et al. [16] proposed a distributed computing offloading framework that considers user interests and network computation points. It contributes to energy saving of mobile devices by making full use of idle and geographically dispersed computation resource through task offloading. These efforts work well to exploit the potential idle computation resource in the network, but the proper incentive scheme is not well addressed to encourage the participation of individual devices. In this paper, we take into account time delays, energy consumption, user satisfaction and close-to-reality transactions that can flexibly respond to sudden and urgent resource exhaustion in challenging scenarios. It is therefore an appropriate complement to existing technologies.

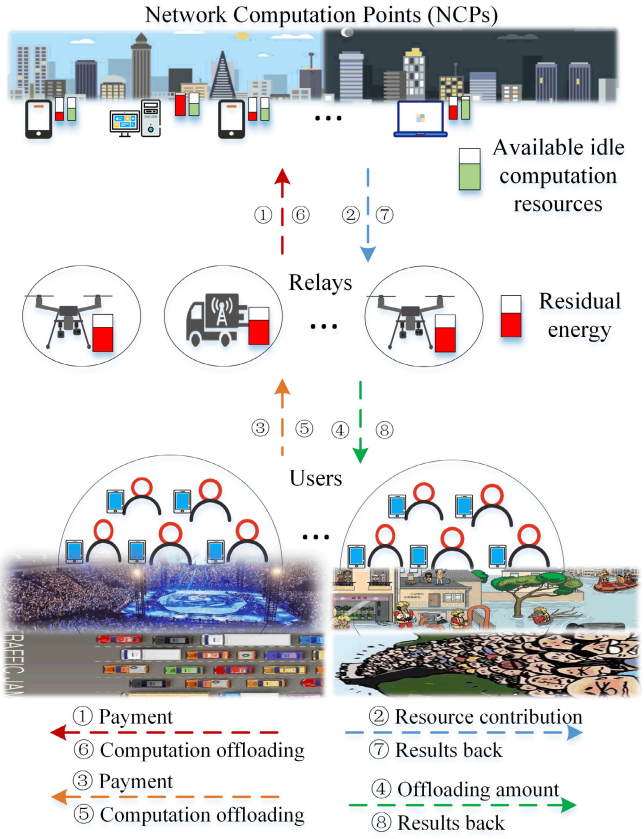


Fig. 1. The dispersed service framework with game-based incentive-driven offloading designs.

III. Offloading Problem Formulation

A. System Overview

As illustrated in Fig. 1, the dispersed service framework defines three main entities. (1) Network computation points (NCPs)[1] have idle computation resource, such as mobile phones charged at night and laptops in standby mode. (2) Relays provide communication relay for computation offloading. In practice, the relays can be drones sent by the government in emergencies or communication vehicles of Internet service providers (ISPs) for mobile communications.

[1]The reliability of NCPs can be ensured through the use of blockchain technology. Specifically, a smart contract could be created to require NCPs involved in the offloading service to submit deposits on the blockchain as the participation fees. If they successfully complete the offloading task, the deposits will be credited to their account along with the rewards. Otherwise, the deposits will be taken away. By doing so, the NCPs are incentivized to effectively ensure their reliability (involving equipment, networks, etc.) for rewards. However, when the deposits are smaller than the incentive rewards, NCPs may have an incentive to provide ineffective offloading services or engage in unreliable activities. To solve this problem, the required deposits should be substantially larger than the rewards offered for a specified period.

TABLE I
SYSTEM MODEL PARAMETER DEFINITIONS AND NOTATIONS

Notation	Definition
$\mathcal{M}$	The sets of NCPs
$\mathcal{K}$	The sets of relays
$\mathcal{N}$	The sets of users
x_i	Amount of offloading offered to user i
y_j	Amount of offloading that NCP j can provide
v_{kj}	Uplink rate between relay k and device j
B_{kj}	Bandwidth allocated to device j by relay k
f_j	NCP j's parallel computation capability
q	Purchase price paid to NCPs by relay k
P_i	Pricing of user i
θ	The ratio of output data size to input data size

(3) Users refer to various IoT devices that perform partially offloadable computation-intensive tasks, i.e. data, but have limited resources. The users are the requesters of offloading services, such as security cameras in concert halls executing face recognition functions, traffic cameras performing video stream analysis during rush hours, and mobile phones playing various games. In Fig. 1, ①-④ describe the incentive-driven flow among three main entities, and ⑤-⑧ represent the actual offloading process. We define the sets of NCPs, the sets of relays and the sets of users as $\mathcal{M} = \{1, \ldots, j, \ldots, M\}$, $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ and $\mathcal{N} = \{1, \ldots, i, \ldots, N\}$, respectively.

Without loss of generality, we assume that the locations of the NCPs and users may change over time, but they are fixed in a specific period [22]. Moreover, each relay supports multiple NCPs and users, while users and NCPs can be associated with only one relay. The location of each node is assumed to be known (e.g., by using GPS [23] or some positioning methods). The important parameters and descriptions are given in TABLE I.

B. Cost Factors

In our communication model, we adopt Orthogonal Frequency Division Multiple Access[2] (OFDMA) technology [26] as the communication mode of data transmission to avoid mutual interference between links. Hence, the uplink rate v_{kj} of the link between relay k and device (NCP or user) j can be expressed as $v_{kj} = B_{kj} log(1 + \frac{p^t g_{kj}}{\sigma^2})$, where $B_{kj} = B_k^{total}/(|M|+|N|)$ is the bandwidth allocated to device j by relay k. Here, B_k^{total} denotes relay k's total bandwidth, and $|M|+|N|$ represents the total number of NCPs and users associated with relay k. $g_{kj} = 1/d_{kj}^{\psi}$ indicates the channel gain, where d_{kj} is the distance between the relay k and device j, and ψ denotes the path loss index. p^t and σ^2 are the transmission power and constant noise power, respectively.

1) Delay: The processing of the offloading data for each user contains the processes of transmission and computation, which consists of five delay periods (see Fig. 3): a. The transmission delay of data offloading from user i to relay k is defined as $\tau_{ik} = x_i/v_{ik}$. Here, user i offloads x_i amount of data to relay k, denoted by vector $\boldsymbol{x} \triangleq (x_i : \forall i \in \mathcal{N})$.

[2]Following [24] and [25], the sub-channel allocation for OFDMA in this paper is flexibly chosen between centralized and distributed approaches depending on the actual situation.

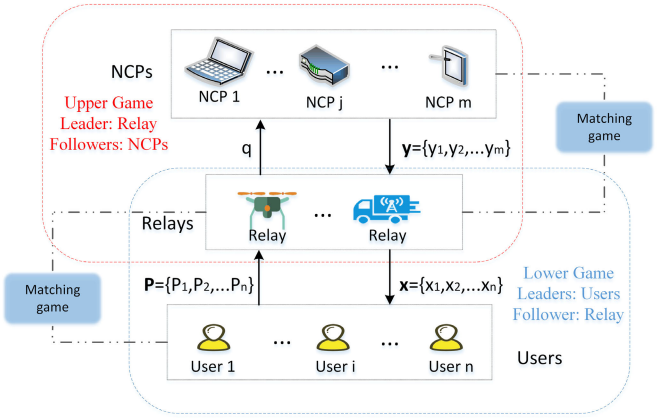


Fig. 2. An illustration to the two-level Stackelberg game combined with matching game process for game-based incentive-driven mechanism.

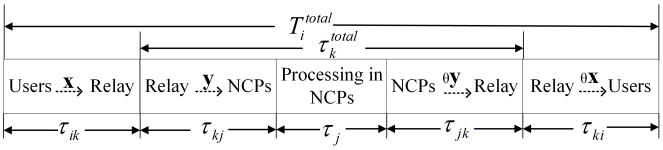


Fig. 3. An illustration of five delay periods.

Note that the computational overload occurs here due to the fact that users are clustered together in the scenarios of interest in this work, such as traffic congestion and festival parties. Hence, for simplicity, we consider that the channel gain g_{ik} between all users and relay k is reciprocal (uplink and downlink channel gains are the same). b. The relay forwarding delay of offloading data to NCP $j \in \mathcal{M}$ for processing is defined as $\tau_{kj} = y_j/v_{kj}$. Here, y_j is the amount of offloading that NCP j would like to provide, defined as vector $\boldsymbol{y} \triangleq (y_j : \forall j \in \mathcal{M})$. It is worth noting that after the relay receives the offloading data from all users, the data needs to be strategically assigned to the NCPs for processing to meet the conditions $y_j \leq y_j^{max}$ and $\sum_{i=1}^{N} x_i \leq \sum_{j=1}^{N} y_j$. c. The computation delay of data in NCP j can be denoted by $\tau_j = \omega y_j/f_j$, where ω is the required CPU cycles per bit [27] and f_j is NCP j's parallel computation capability without queuing time. d. The result-return delay of NCP j transmitting the computation results to relay k can be expressed by $\tau_{jk} = \theta y_j/v_{jk}$, where θ is the ratio of output data size to input data size. e. The result-return delay of relay k transmitting the computation results to user i is defined as $\tau_{ki} = \theta x_i/v_{ki}$. Thus, we can denote the total delay in obtaining results for user i as T_i^{total} shown in Fig. 3.

2) Energy Consumption: Based on the delay model above, the transmission energy consumption and result-return energy consumption can be formulated as $E_{ik} = p^t \tau_{ik}$, $E_{kj} = p^t \tau_{kj}$, $E_{jk} = p^t \tau_{jk}$ and $E_{ki} = p^t \tau_{ki}$. The computation energy consumption of data in NCP j can be defined as $E_j = Power(f_j)\frac{\omega y_j}{f_j}$, where $Power(f_j) = \mu f_j^{\alpha}$ is the speed-power curve function [28] of processors, and μ and $\alpha = 3$ [28] are the device-related parameters.

C. Utility Maximization of Three Main Entities

We first model the interactions among NCPs, a single relay, and users as a two-level Stackelberg game. Then, a dual matching game is introduced for multi-relay scenarios, allowing the multiple relays to be decomposed into multiple independent single-relay problems, as shown in Fig. 2.

1) The Utility of NCPs: Driven by the incentive mechanism, each NCP contributes its idle computation resource to provide users with offloading services. The utility function of NCP j consists of two parts: the reward paid by relay k and the cost due to providing computation and communication services. The reward function is characterized by a logarithmic function [29] as $R_j^{ncp}(y_j, q) = qln(1 + y_j)$, where q is the purchase price determined by relay k. The cost function of NCP j is represented by $C_j^{ncp}(y_j, q) = a(E_j + E_{jk})$, where a is the positive energy cost coefficient. Thus, the optimization problem for NCP j can be formulated as follows:

$$\begin{aligned} &\max_{y_j} U_j^{ncp}(y_j, q) = R_j^{ncp}(y_j, q) - C_j^{ncp}(y_j, q), \\ &s.t.\ C1: 0 \le y_j \le y_j^{max}, \\ &\quad C2: E_j + E_{jk} \le E_j^{max}, \\ &\quad C3: \tau_{kj} + \tau_j + \tau_{jk} \le \tau_k^{max}. \end{aligned} \tag{1}$$

Constraint C1 ensures that the amount of offloading data available from each NCP does not exceed its maximum value y_j^{max}. Moreover, the maximum energy that each NCP consumes is limited, and the computation results must be returned to the relay in time. Thus, constraints C2 and C3 are applied to meet these requirements.

2) The Utility of Relays: Each relay k is designed to act as a broker for offloading between NCPs and users. Specifically, the relay receives payments $R_k^{relay}(\boldsymbol{x}, \boldsymbol{P}) = \sum_{i=1}^{N} P_i ln(1 + x_i)$ from the users, which are used to purchase the NCPs' computation resource and cover its forwarding cost, i.e., $C_k^{relay}(\boldsymbol{x}, \boldsymbol{y}, q) = \sum_{j=1}^{M}[qln(1 + y_j) + aE_{kj}] + \sum_{i=1}^{N} aE_{ki}$. $R_k^{relay}(\boldsymbol{x}, \boldsymbol{P})$ is the total reward given by the users, where P_i is the purchase price of user i, denoted by a vector as $\boldsymbol{P} \triangleq \{P_i : \forall i \in \mathcal{N}\}$. $C_k^{relay}(\boldsymbol{x}, \boldsymbol{y}, q)$ denotes the relay's cost function, where $\sum_{j=1}^{M}[qln(1+y_j)+aE_{kj}]$ represents the total price paid to the NCPs and the total cost of data offloading, while $\sum_{i=1}^{N} aE_{ki}$ describes the cost that the relay incurs to return the final results to users. The relay determines its purchase price q and the amount of offloading $\boldsymbol{x}$ available to users based on the optimal users' pricing $\boldsymbol{P}^*$ and the optimal amount of offloading $\boldsymbol{y}^*$ available from NCPs. By deciding the optimal values of q and $\boldsymbol{x}$ to maximize its utility, we can describe the optimization problem of relay k as follows:

$$\begin{aligned} &\max_{\boldsymbol{x}, q} U_k^{relay}(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{y}, q) = R_k^{relay}(\boldsymbol{x}, \boldsymbol{P}) - C_k^{relay}(\boldsymbol{x}, \boldsymbol{y}, q), \\ &s.t.\ C1: 0 \le x_i \le x_i^{max}, \\ &\quad C2: 0 \le q \le q^{max}, \\ &\quad C3: \sum_{i=1}^{N} x_i \le \sum_{j=1}^{M} y_j, \\ &\quad C4: \sum_{i=1}^{N} E_{ki} + \sum_{j=1}^{M} E_{kj} \le E_k^{max}. \end{aligned} \tag{2}$$

C1 ensures that the amount of offloading provided by the relay do not exceed the user's maximum demand. C2 represents the upper and lower bounds of the purchase price. C3 implies that the total amount of offloading provided by the relay cannot exceed the total amount of offloading provided by the NCPs. C4 means that the relay must have enough remaining energy to maintain its operation.

3) The Utility of Users: The users need to offload the data to NCPs through relays to achieve computation offloading. The utility function for user i consists of two parts: the user satisfaction with the offloading service and the total offloading cost. The satisfaction function of user i can be modeled as a function of the available offloading amount x_i and defined as follows:

$$S_i^{user}(\boldsymbol{x}) = b_i x_i - h_i(x_i^{max} - x_i)^2 + \sum_{i \ne i'} I(l_{i,i'}) x_i x_{i'}, \tag{3}$$

where $S_i^{user}(\boldsymbol{x})$ is monotonically increasing with x_i, and the coefficients $b_i > 0$ and $h_i > 0$ capture the intrinsic demand rate [30]. Specifically, b_i measures the user's sensitivity to the received data x_i. The coefficient h_i evaluates user i's maximum demand satisfaction. The term $\sum_{i \ne i'} I(l_{i,i'})$ denotes the (positive) network effect [31] that user i obtains from other users. It implies that other users in the area receive a high amount of offloading can improve the social satisfaction of the current user. In other words, when users receive better offloading services, there is a positive mutual influence on each other, contributing to overall social satisfaction. $I(l_{i,i'})$ is an increasing function related to the user contact strength parameter $l_{i,i'}$, satisfying $I(\cdot) \ge 0$. However, in fact, this social effect cannot increase without limit. Therefore, based on the literature [32], here we assume $\sum_{i \ne i'} I(l_{i,i'}) < h_i$. Furthermore, the total cost including the price paid to the relay, offloading energy consumption, and local computation energy consumption can be expressed as $C_i^{user}(\boldsymbol{x}, P_i) = P_i ln(1 + x_i) + a(E_{ik} + \mu f_i^2 \omega(x_i^{max} - x_i))$.

Hence, through the game with other users and feedback from relay k, the optimization problem of user i can be defined as follows:

$$\begin{aligned} &\max_{P_i} U_i^{user}(P_i | P_{-i}^*, \boldsymbol{x}) = S_i^{user}(\boldsymbol{x}) - C_i^{user}(\boldsymbol{x}, P_i), \\ &s.t.\ C1: 0 \le P_i \le P_i^{max}, \\ &\quad C2: \tau_{ik} + \tau_{ki} + \tau_k^{max} \le T_{user}^{max}, \end{aligned} \tag{4}$$

where $P_{-i}^* = \{P_1^*, P_2^*, \ldots, P_N^*\}$ denotes the optimal pricing of all users except user i. C1 means that P_i^{max} is the upper bound on P_i. C2 ensures that the computation results should be returned within the user's acceptable delay.

IV. Analysis and Solutions

We analyze the interactions between the three main entities for both single-relay and multi-relay scenarios. For single-relay scenarios, the interactions are formulated as a two-level Stackelberg game. The relay is acted as a bridge between the users and the NCPs. Specifically, an upper level Stackelberg game consisting of the NCPs as followers and the relay as a leader is constructed in IV-A. Then, a lower level

Stackelberg game consisting of the relay as a follower and the users as leaders is constructed in IV-B. For multi-relay scenarios, we decompose the multi-relay problem into multiple independent single-relay problems via a dual matching game as presented in Section IV-C.

A. The Upper Level of Stackelberg Game

The relay guides the NCPs (offloading service providers) in determining the amount of offloading available through its purchase price. The upper level game thus consists of the NCPs as followers and the relay as a leader.

1) NCPs (Determining the Amount of Offloading Available): We can find that the second-order derivative with respect to y_j is $\partial^2(U_j^{ncp})/\partial y_j^2 = \frac{-q}{(1+y_j)^2} < 0$, which proves that U_j^{ncp} is concave. Thus, the optimization problem for NCP j in (1) is a standard convex problem. Then, the problem can be transformed into an unconstrained optimization problem using the Lagrangian duality method, as follows:

$$\begin{aligned} L_j^{ncp}(y_j, q, \lambda_j, \beta_j, \gamma_j) = U_j^{ncp}(y_j, q) + \lambda_j(y_j^{max} - y_j) \\ + \beta_j(E_j^{max} - E_j - E_{jk}) \\ + \gamma_j(\tau_k^{max} - \tau_{kj} - \tau_j - \tau_{jk}), \end{aligned} \quad (5)$$

where λ_j, β_j, and γ_j are the Lagrangian dual variables corresponding to constraints C1, C2 and C3, respectively. When the purchase price q is given, the problem satisfies the Karush-Kuhn-Tucker (KKT) conditions [33]. Then, we set the derivative of L_j^{ncp} with respect to y_j to zero, and the optimal response y_j^* of NCP j can be obtained by

$$\begin{aligned} y_j^* &= \frac{q}{V_j} - 1, \\ V_j &= (a+\beta_j)(\mu f_j^2 \omega + \frac{\theta p^t}{v_{jk}}) + \lambda_j + \gamma_j(\frac{1}{v_{kj}} + \frac{\omega}{f_j} + \frac{\theta}{v_{jk}}). \end{aligned} \quad (6)$$

According to (6), it can be observed that the optimal response y_j of NCP j with a fixed price q depends on the the dual variables λ_j, β_j and γ_j. We can update the dual variables by solving the dual problem of (5), which is described as follows:

$$\begin{aligned} &\min_{\lambda_j, \beta_j, \gamma_j} D(\lambda_j, \beta_j, \gamma_j), \\ &s.t.\ \lambda_j \geq 0, \quad \beta_j \geq 0, \quad \gamma_j \geq 0. \end{aligned} \quad (7)$$

Clearly, the dual problem is a convex optimization problem, and therefore we can use sub-gradient [34] method to solve it. The sub-gradient of the dual equation $D(\lambda_j, \beta_j, \gamma_j)$ can be obtained as follows:

$$\begin{aligned} \frac{\partial D}{\partial \lambda_j} &= y_j^{max} - y_j^*, \\ \frac{\partial D}{\partial \beta_j} &= E_j^{max} - \mu_j f_j^2 \omega y_j^* - \frac{\theta y_j^* p^t}{v_{jk}}, \\ \frac{\partial D}{\partial \gamma_j} &= \tau_k^{max} - \frac{y_j^*}{v_{kj}} - \frac{\omega y_j^*}{f_j} - \frac{\theta y_j^*}{r_{jk}}, \end{aligned} \quad (8)$$

where y_j^* is the optimal solution given q and the dual variables λ_j, β_j, and γ_j.

From (7), the Lagrangian dual variables are updated according to the following formula

$$\begin{aligned} \lambda_j^{(\eta+1)} &= [\lambda_j^{(\eta)} - \alpha_1^{(\eta)}(\frac{\partial D}{\partial \lambda_j})^{(\eta)}]^+, \\ \beta_j^{(\eta+1)} &= [\beta_j^{(\eta)} - \alpha_2^{(\eta)}(\frac{\partial D}{\partial \beta_j})^{(\eta)}]^+, \\ \gamma_j^{(\eta+1)} &= [\gamma_j^{(\eta)} - \alpha_3^{(\eta)}(\frac{\partial D}{\partial \gamma_j})^{(\eta)}]^+, \end{aligned} \quad (9)$$

where η is the number of iterations, $\alpha_1^{(\eta)}$, $\alpha_2^{(\eta)}$ and $\alpha_3^{(\eta)}$ are non-negative steps. The Lagrangian dual variables are iteratively updated until the given convergence conditions are achieved. The detailed procedure for obtaining the best response y_j^* of NCP $j \in \mathcal{M}$ is shown in Algorithm 1.

Algorithm 1 The Best Response y_j^* of NCP $j \in \mathcal{M}$ for Given q

Input: q, ε_1
Output: y_j^*
Initialization: $\eta = 0, y_j^0, \lambda_j^{(0)}, \beta_j^{(0)}, \gamma_j^{(0)}$;
repeat
  Update $y_j^{(\eta+1)} = \frac{q}{V_j} - 1$ in (6);
  $\eta = \eta + 1$;
  Update $\lambda_j^{(\eta)}, \beta_j^{(\eta)}, \gamma_j^{(\eta)}$ according to (9);
until $\left|\lambda_j^{\eta} - \lambda_j^{\eta-1}\right| < \varepsilon_1 \cap \left|\beta_j^{\eta} - \beta_j^{\eta-1}\right| < \varepsilon_1 \cap \left|\gamma_j^{\eta} - \gamma_j^{\eta-1}\right| < \varepsilon_1$;
$y_j^* = y_j^{(\eta)}$;
return y_j^*

2) Relay (Determining the Purchase Price and the Amount of Offloading Offered to Users): We first prove the existence and uniqueness of optimal solutions for optimization problem of the relay.

Definition 1 (Uniqueness Condition): Given that a problem is a concave maximization problem, it is unique if the solution exists [35].

Theorem 1: The optimization problem of relay k has a unique optimal solution $(\boldsymbol{x}, q)$.

Proof: We first observe that the Hessian matrix of $U_k^{relay}(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{y}, q)$ with respect to $\boldsymbol{x}$ and q is

$$H = \begin{bmatrix} H^{\boldsymbol{x}} & 0 \\ 0 & H^q \end{bmatrix}, \quad (10)$$

where

$$\begin{aligned} H^{\boldsymbol{x}} &= [\frac{\partial^2 U_k^{relay}}{\partial x_i \partial x_{i'}}]_{i,i' \in \mathcal{N}} \\ &= -diag\left(\frac{P_1}{(1+x_1)^2}, \ldots, \frac{P_N}{(1+x_N)^2}\right) < \mathbf{0}. \end{aligned}$$

For H^q, substituting $y_j = \frac{q}{V_j} - 1$ into U_k^{relay}, we can derive

$$U_k^{relay} = R_k^{relay} - \sum_{j=1}^{M}[qln(\frac{q}{V_j}) + \frac{ap^t(\frac{q}{V_j} - 1)}{v_{kj}}] - \sum_{i=1}^{N} aE_{ki}. \quad (11)$$

Then, the second derivative of (11) with respect to q is $H^q = \frac{\partial^2 U_k^{relay}}{\partial q^2} = -\frac{|M|}{q} < 0$. Based on $H^{\boldsymbol{x}}$ and H^q, H is negative definite and U_k^{relay} is concave. According to the

convexity of its constraint conditions and ***Definition 1***, it is proved that the optimization problem of relay k has a unique optimal solution. □

Then, the Lagrangian function associated with the relay k can be written as

$$\begin{aligned} &L_k^{relay}(\boldsymbol{x},\boldsymbol{P},\boldsymbol{y},q,\boldsymbol{\lambda},\beta,\gamma,\delta) \\ &= U_k^{relay}(\boldsymbol{x},\boldsymbol{P},\boldsymbol{y},q) - \beta(q-q^{max}) \\ &\quad - \sum_{i=1}^{N}\lambda_i(x_i - x_i^{max}) - \gamma(\sum_{i=1}^{N}x_i - \sum_{j=1}^{M}y_j) \\ &\quad - \delta(\sum_{i=1}^{N}\frac{\theta x_i p^t}{v_{ki}} + \sum_{j=1}^{M}\frac{y_j p^t}{v_{kj}} - E_k^{max}). \end{aligned} \tag{12}$$

Using the KKT conditions, we can derive the optimal offloading amount allocation of relay k as

$$x_i(P_i,P_{-i}) = \begin{cases} 0 & P_i < P_i^{min}, \\ \dfrac{P_i}{V'} - 1 & P_i^{min} \le P_i \le P_i^{max}, \\ x_i^{max} & P_i > P_i^{max}, \end{cases} \tag{13}$$

where $V' = \theta(ap^t + \delta p^t)/v_{ki} + \gamma$, $P_i^{min} = V'$, and $P_i^{max} = V'(1 + x_i^{max})$. Moreover, the purchase price q of relay k can be expressed as

$$q = e^z, \quad \text{where } z = \frac{\sum_{j=1}^{M}[(ln(V_j) - \frac{(a+\delta)p^t}{v_{kj}V_j} + \frac{\gamma}{V_j} - 1)] - \beta}{|M|}. \tag{14}$$

The best response of relay k is jointly affected by the responses of NCPs (Algorithm 1) and users (Algorithm 3). The iterative solution process performed by relay k is shown in Algorithm 2. Note that when the game reaches an equilibrium, the iterations for finding the best responses converge and terminate.

Algorithm 2 The Relay k's Best Response q^* and $\boldsymbol{x}^*$

Input: The prior knowledge: ϵ_2
Output: The optimal purchase price q^* and the optimal offloading amount $\boldsymbol{x}^*$
Initialization: $q_0 \in [0, q^{max}]$, $t = 0$, $U_k^{relay}(t)$;
repeat
  Collect the best response $\boldsymbol{y}^*$ from NCPs using Algorithm 1;
  Collect the best response $\boldsymbol{P}^*$ from users using Algorithm 3;
  Solve $q^* = e^z$ in (14) and $\boldsymbol{x}^*$ in problem (13) using the interior point algorithm [36];
  Calculate $U_k^{relay}(t+1)$ according to (2);
  $t = t + 1$;
until $|U_k^{relay}(t) - U_k^{relay}(t-1)| < \epsilon_2$;
return q^*, $\boldsymbol{x}^*$

B. The Lower Level of Stackelberg Game

The users (offloading service requesters) guide the relay to determine the amount of offloading offered to them through pricing. Thus, the lower level game consists of the users as leaders and the relay as a follower.

1) Users (Pricing): First, all users inform relay k an initial pricing strategy based on their information, and then receive the response $x_i(P_i, \boldsymbol{P}_{-i})$ from relay k in IV-A.2). After that, the user's utility function will be updated as follows:

$$\begin{aligned} U_i^{user}(x_i(P_i,\boldsymbol{P}_{-i}),P_i) &= S_i^{user}(x_i(P_i,\boldsymbol{P}_{-i})) \\ &\quad - C_i^{user}(x_i(P_i,\boldsymbol{P}_{-i}),P_i). \end{aligned} \tag{15}$$

Moreover, the user's optimal pricing depends on the response of relay k and is also affected by other users' pricing strategies. Since the total amount of offloading provided by the relay is limited, there is price competition among users, forming a non-cooperative game $G = \{\mathcal{N}, \{P_i\}_{i\in\mathcal{N}}, \{U_i^{user}\}_{i\in\mathcal{N}}\}$. To obtain the equilibrium solution of the non-cooperative game, the concepts of existence and uniqueness of Nash equilibrium are first introduced.

Theorem 2: The non-cooperative game G between users has a Nash equilibrium [37].

Proof: For user $i \in \mathcal{N}$, $x(P_i, \boldsymbol{P}_{-i})$ obtained from the relay is substituted into the utility function $U_i^{user}(x(P_i,\boldsymbol{P}_{-i}),P_i)$, and is shown as follows:

$$\begin{aligned} &U_i^{user}(x_i(P_i,\boldsymbol{P}_{-i}),P_i) \\ &= b_i(\frac{P_i}{V'} - 1) - h_i(x_i^{max} - (\frac{P_i}{V'} - 1))^2 \\ &\quad + \sum_{i\neq i'} I(l_{i,i'})(\frac{P_i}{V'} - 1)(\frac{P_{i'}}{V'} - 1) + \frac{ap^t}{v_{ik}} \\ &\quad - P_i ln(\frac{P_i}{V'}) - \frac{aP_i p^t}{V' v_{ik}} - a\mu f_i^2 \omega(x_i^{max} - \frac{P_i}{V'} + 1). \end{aligned} \tag{16}$$

According to the formulation above, the first-order derivative of (16) is

$$\begin{aligned} \frac{\partial U_i^{user}}{\partial P_i} &= \frac{(2x_i^{max}h_i + b_i + 2h_i)}{V'} - \frac{2P_i h_i}{(V')^2} - ln(\frac{P_i}{V'}) \\ &\quad + \sum_{i\neq i'} \frac{I(l_{i,i'})(P_{i'} - V')}{(V')^2} - \frac{ap^t}{V' v_{ik}} + a\frac{\mu f_i^2 \omega}{V'} - 1. \end{aligned} \tag{17}$$

Then, we can obtain the second-order derivative as follows:

$$\frac{\partial^2(U_i^{user}(x_i(P_i,\boldsymbol{P}_{-i}),P_i))}{\partial P_i^2} = -(\frac{2h_i}{(V')^2} + \frac{1}{P_i}). \tag{18}$$

Since $h_i > 0$ and $P_i > 0$, the second derivative is negative and the utility function is a strictly concave. Therefore, the Nash equilibrium of non-cooperative game G exists and ***Theorem 2*** is proved. □

Theorem 3: There is a unique Nash equilibrium in non-cooperative game G.

Proof: The Jacobian matrix $\nabla U^{user}(\boldsymbol{P})$ with respect to $\boldsymbol{P}$ is defined by $-(\Lambda - L)$. The matrix $\Lambda = diag(\frac{2h_1}{(V')^2} + \frac{1}{P_1}, \frac{2h_2}{(V')^2} + \frac{1}{P_2}, \ldots, \frac{2h_n}{(V')^2} + \frac{1}{P_n})$ and matrix L is expressed by

$$L = \frac{1}{(V')^2}\begin{pmatrix} 0 & I(l_{1,2}) & \cdots & I(l_{1,n}) \\ I(l_{2,1}) & 0 & \cdots & I(l_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ I(l_{n,1}) & I(l_{n,2}) & \cdots & 0 \end{pmatrix}.$$

According to the assumption $\sum_{i\neq i'} I(l_{i,i'}) < h_i$ in (3), it can satisfy the following constraint,

$$(\Lambda - L)_{n,n} = \frac{2h_n}{(V')^2} + \frac{1}{P_n} > \frac{h_n}{(V')^2} > \sum_{i\neq n} \frac{I(l_{i,n})}{(V')^2} = \left|\sum_{i\neq n} (\Lambda - L)_{i,n}\right|, \tag{19}$$

where $(\Lambda - L)_{i,n}$ denotes the element in the i^{th} row and n^{th} column of the matrix $(\Lambda - L)$.

We find that $(\Lambda - L)$ is strictly diagonally dominant. Since $I(l_{i,i'}) = I(l_{i',i})\ \forall i, i' \in \mathcal{N}$, it is clear that $(\Lambda - L)$ is symmetric. Then, the corresponding $(\Lambda - L)^T$ is strictly diagonally dominant, and $(\Lambda - L) + (\Lambda - L)^T$ is also strictly diagonally dominant. Based on the above, it can be seen that $-(\Lambda - L) - (\Lambda - L)^T$ is negative definite and $\nabla U^{user}(\boldsymbol{P})$ is diagonally strictly concave. Thus, the non-cooperative game G has a unique Nash equilibrium. The proof of ***Theorem 3*** is completed. □

To sum up, after each user $i \in \mathcal{N}$ collects relay k's response, its optimal pricing P_i^* can be derived by solving $\frac{\partial L_i^{user}(x_i(P_i, \boldsymbol{P}_{-i}), P_i, \lambda_i, \beta_i)}{\partial P_i} = 0$, where

$$\begin{aligned} &L_i^{user}(x_i(P_i, \boldsymbol{P}_{-i}), P_i, \lambda_i, \beta_i) \\ &\quad = U_i^{user}(x_i(P_i, \boldsymbol{P}_{-i}), P_i) \\ &\qquad - \lambda_i(P_i - P_i^{max}) - \beta_i(\tau_{ik} + \tau_{ki} + \tau_k^{max} - T_{user}^{max}). \end{aligned} \tag{20}$$

Thus, given $\boldsymbol{P}_{-i}$, the optimal pricing strategy for user i is

$$P_i^* = arg\max_{P_i} L_i^{user}(x_i(P_i, \boldsymbol{P}_{-i}), P_i, \lambda_i, \beta_i). \tag{21}$$

It is intractable to directly obtain the closed-form solution given the above function. To obtain the equilibrium solution, we design a distributed dynamic adaptive algorithm. Each user $i \in \mathcal{N}$ plays multiple iterations of the game to obtain a optimal pricing strategy. When all users in the network can no longer improve their utilities by updating their strategies individually, the non-cooperative game achieves a Nash equilibrium. The specific solution process[3] performed by user $i \in \mathcal{N}$ is shown in Algorithm 3.

Algorithm 3 The Best Response P_i^* of User $i \in \mathcal{N}$ for Given $\boldsymbol{x}$

Input: The prior knowledge: $\boldsymbol{x}$, ϵ_3.
Output: The optimal pricing P_i^* for user i.
Initialization: $t = 0$, P_i^{max}, $P_i(0) \in [0, P_i^{max}]$;
Collect the best response $\boldsymbol{x}^*$ from the relay k;
repeat
  $P_i(t+1) = P_i(t) + \alpha_i \cdot \frac{\partial L_i^{user}}{\partial P_i}\Big|_{P_i(t)}$, α_i is the adjustment step of the pricing strategy;
  $t = t + 1$;
until $|P_i(t) - P_i(t-1)| < \epsilon_3$;
$P_i^* = P_i(t)$;
return P_i^*

[3]The theoretical complexity of the gradient descent method used in Algorithm 3 is $\mathcal{O}(log(\frac{1}{\varepsilon_3}))$, where ε_3 is the target accuracy. Furthermore, the game algorithm satisfies convergence and converges to a unique equilibrium point according to the convergence proof in §2 of [38].

Based on the above analysis and proof, we can conclude with the following theorem.

Theorem 4: The equilibrium solution $(q^*, \boldsymbol{y^*}, \boldsymbol{P^*}, \boldsymbol{x^*})$ of the two-level Stackelberg game exists and is unique.

Proof: Through the analysis of IV-A and the optimal solution shown in (6), it can be concluded that NCPs (the followers in the upper game) have unique optimal responses $\boldsymbol{y^*}$ to the purchase price q^* of relay k. According to ***Theorem 3***, there exists a unique Nash equilibrium $\boldsymbol{P}^*$ in the non-cooperative game of users (the leaders in the lower game). Furthermore, given $\boldsymbol{y^*}$ and $\boldsymbol{P}^*$, relay k (as the leader in the upper game) has a unique optimal purchase price q^*. Meanwhile, relay k (as the follower in the lower game) has a unique optimal offloading amount $\boldsymbol{x^*}$ that can be offered to users (as proved in ***Theorem 1***). Thus, there is a unique equilibrium solution $(q^*, \boldsymbol{y^*})$ for the upper level Stackelberg game and a unique equilibrium solution $(\boldsymbol{P^*}, \boldsymbol{x^*})$ for the lower level Stackelberg game. In conclusion, the equilibrium solution $(q^*, \boldsymbol{y^*}, \boldsymbol{P^*}, \boldsymbol{x^*})$ of the two-level Stackelberg game exists and is unique, and can be obtained by the proposed algorithms (including Algorithms 1, 2, and 3). □

C. Multiple Relays

Multiple relays can expand the coverage of the framework and provide services to a larger number of users. However, they will compete for the limited computation resource of NCPs and influence the users' decisions. Given that various relays have different bandwidths, geographic locations, and limitations on the number of users and NCPs that can be accommodated, it is a challenging problem to achieve a balance among NCPs, users, and relays. To address this problem, we decompose multiple relays into several single-relay problems and solve them using a dual bilateral matching game.

1) Matching Concepts: We model the associations among three main entities as a dual one-to-many matching game with resource constraints, where the players are the NCP set $\mathcal{M}$, the relay set $\mathcal{K}$, and the user set $\mathcal{N}$. Specifically, in the first bilateral matching game, each relay acts as a seller and NCPs act as buyers, while in the second bilateral matching game, each relay acts as a seller and users act as buyers. It is important to note that the matching is bilateral, meaning that a given NCP (or user) is admitted to a relay only if the relay admits that NCP (or user). The matching game (e.g., the first bilateral matching game) is formally defined as follows.

Definition 2: Given two disjoint finite sets of players $\mathcal{M}$ and $\mathcal{K}$, a one-to-many matching function Ψ [39] is defined such that for all $j \in \mathcal{M}$ and $k \in \mathcal{K}$.

(i) $\Psi(j) \in \mathcal{K}$ and$|\Psi(j)| \in \{0, 1\}$,
(ii) $\Psi(k) \subseteq \mathcal{M}$ and $|\Psi(k)| \leq M_k^{max}$,
(iii) $\Psi(j) = k \Leftrightarrow \Psi(k) = j$.

Thus, the matching game Ψ can be defined by a tuple $\{\mathcal{M}, \mathcal{K}, M_k^{max}, \succ_j, \succ_k\}$, where M_k^{max} is the maximum number of NCPs that can be associated with a relay k. The preference relations of NCPs and relays are represented by $\succ_j$ and $\succ_k$, respectively. Condition (i) implies that each NCP can only be associated with one relay, while condition (ii) specifies

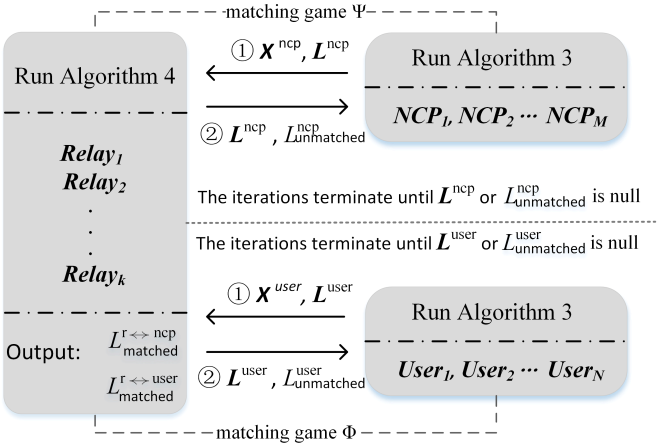


Fig. 4. Tripartite matching strategy.

Algorithm 4 Preference List Generation for Each NCP (or User)

Input: unmatched NCP set $L^{ncp}_{unmatched} = \mathcal{M}$, unmatched user set $L^{user}_{unmatched} = \mathcal{N}$, relay set $\mathcal{K}$, the preference lists of NCPs $\boldsymbol{L}^{ncp}$ (or $\boldsymbol{L}^{user}$), the knowledge of relays: B^{total}_k, M^{max}_k, N^{max}_k, q^{max}_k.
Output: The sets of NCP decisions: $\boldsymbol{X}^{ncp}$ (or the sets of user decisions: $\boldsymbol{X}^{user}$) and the preference lists $\boldsymbol{L}^{ncp}$ (or $\boldsymbol{L}^{user}$).

```
% Take NCPs for example below;
Initialization: X^ncp = 0;
for j = 1 to |L^ncp_unmatched| do
    if L^ncp_j ≠ null then
        go to step 11;
    else
        for k = 1 to |K| do
            Generate L^ncp_j using Pre_j(k) and sort in descending order;
        end
    end
    Find the top ranked k in L^ncp_j and setting X^ncp_kj = 1;
end
return X^ncp, L^ncp
```

the maximum number of NCPs that can be associated with each relay. Condition (iii) states that if an NCP j is matched with a relay k, then k is also matched with j. The output of this game is a set of matching pairs $< j, k >_{j\in\mathcal{M},k\in\mathcal{K}}$ between NCPs and relays. The second bilateral matching game $\Phi = \{\mathcal{N}, \mathcal{K}, N^{max}_k, \succ_i, \succ_k\}$ between relays and users is defined in a similar manner, where N^{max}_k is the maximum number of users that can be associated with a relay k. The same conditions and output apply to the matching game Φ.

2) Preference Profiles of Players: In the proposed game Ψ, each player uses a preference profile to rank the other players, which serves as a reference for subsequent matching. Specifically, each NCP j and relay k can generate their preference lists L^{ncp}_j and L^{ncp}_k based on their respective preference profiles.

Definition 3: The preference profile $Pre_j(k)$ of NCP j for different relays can be mathematically expressed as $Pre_j(k) = \frac{B^{total}_k}{M^{max}_k} + q^{max}_k$. For the NCPs, a relay with a larger total bandwidth, a smaller receivable quantity and a higher purchase price cap is preferred.

Definition 4: The preference profile $Pre_k(j)$ of relay k for different NCPs can be defined as $Pre_k(j) = y^{max}_j - d_{jk}$. For the relays, an NCP that can provide high amounts of offloading while being geographically close to it (i.e. better transmission and lower latency) is preferred.

In the game Φ, users and relays generate preference lists L^{user}_i and L^{user}_k based on the following preference profiles.

Definition 5: The preference profile $Pre_i(k)$ of user i for different relays can be mathematically expressed as $Pre_i(k) = \frac{B^{total}_k}{N^{max}_k}$. For the users, a relay with sufficient bandwidth and less matching capacity is preferred.

Definition 6: The preference profile $Pre_k(i)$ of relay k for different users can be defined as $Pre_k(i) = P^{max}_i - d_{ik}$. For the relays, a user with a higher purchase price cap and geographical proximity is preferred.

3) Tripartite Matching Strategy: As illustrated in Fig. 4, the tripartite matching process combines Algorithm 4 and Algorithm 5 to solve the matching problem among the three main entities, and iteratively achieves the final stable matching. The computational complexity is $O(max(|L_i|_{i\in\mathcal{N}}) \cdot max(|L_k|_{k\in\mathcal{K}})) + O(K \cdot max(|L_i|_{i\in\mathcal{N}})) + O(max(|L_j|_{j\in\mathcal{M}}) \cdot max(|L_k|_{k\in\mathcal{K}})) + O(K \cdot max(|L_j|_{j\in\mathcal{M}}))$. Further details on the analysis process can be found in [16]. In our design, Algorithm 4 addresses the relay selection problem for both users and NCPs, enabling them to select the best relays based on their respective preference profiles. Algorithm 5 solves the reverse selection problem for relays. Given the selection requests from users and NCPs, the relays select users and NCPs that satisfy their preferences and constraints.

Iterative termination conditions are employed to determine when the matching process should stop. Specifically, if the total number of NCPs (or users) allowed to be associated with all relays is less than the actual number of NCPs (or users), the termination condition is when the preference list for each NCP (or user) becomes empty. On the other hand, if the total number of NCPs (or users) allowed to be associated with all relays is greater than or equal to the actual number of NCPs (or users), the termination condition is when there are no unmatched NCPs (or users) left.

For Algorithm 4 implemented by NCPs or users, it has two phases, i.e., network information collection and preference list generation. *Network information collection*: Each relay broadcasts its resource information (involving total bandwidth, purchase price cap, etc.).[4] Then, each user and NCP sort all relays according to their own preferences based on the collected information. This step provides important information for subsequent evaluation. *Preference list generation*: First, each NCP (or user) calculates the preference order for all relays based on the preference profile of NCPs ($Pre_j(k)$ or the preference profile of users $Pre_i(k)$), and sorts them in descending order. Then, we select the top-ranked relays in the preference lists $\boldsymbol{L}^{ncp}$ of NCPs (or $\boldsymbol{L}^{user}$ of users) as the current choice of NCPs (or users). Meanwhile, the selection decisions are sent to the corresponding relays for

[4]We can guarantee the truthfulness of the information based on third-party authority platforms [40].

TABLE II
PARAMETER SETTINGS

Parameters	f_j	d_{kj}	E_{ncp}^{max}	E_k^{max}	$I(\cdot)$	T_{user}^{max}	τ_k^{max}	B_k^{total}	p^t
Values	$[2,5]$GHz	$[1,4]$km	$[30,100]$J	500J	$[0.001,0.01]$	20ms	10ms	$[100,250]$MHz	0.5W
Parameters	ψ	μ	a	θ	α	ω	σ^2	h_i	b_i
Values	4	10^{-28}	0.1	10^{-2}	3	1000	10^{-11}	0.5	1

Algorithm 5 Matching Evaluation for Each Relay

Input: unmatched NCP set $L_{unmatched}^{ncp}$, L_j^{ncp} and X_j^{ncp}, $\forall j \in \mathcal{M}$ (or unmatched user set $L_{unmatched}^{user}$, L_i^{user} and X_i^{user}, $\forall i \in \mathcal{N}$)
Output: $L_{unmatched}^{ncp}$, matched results set $L_{matched}^{k \leftrightarrow ncp}$ for NCPs and relays, $\boldsymbol{L}^{ncp}$ (or $L_{unmatched}^{user}$, matched results set $L_{matched}^{k \leftrightarrow user}$ for users and relays, $\boldsymbol{L}^{user}$)

for $k = 1$ *to* $|K|$ **do**
  % Take NCPs for example below;
  for $j = 1$ *to* $|L_{unmatched}^{ncp}|$ **do**
    if $X_{kj}^{ncp} = 1$ **then**
      if $|L_{matched}^{k \leftrightarrow ncp}| < M_k^{max}$ **then**
        % Reverse Selection Success:
        $L_{matched}^{k \leftrightarrow ncp} = L_{matched}^{k \leftrightarrow ncp} \cup j$;
        $L_{unmatched}^{ncp} = L_{unmatched}^{ncp} \setminus j$;
        $L_j^{ncp} = L_j^{ncp} \setminus k$;
      else
        % Competition & Assessment: Comparison with the existing NCPs in $L_{matched}^{r \leftrightarrow ncp}$ using $Pre_k(j)$;
        $j' = min(L_{matched}^{k \leftrightarrow ncp})$;
        if $L_k^{ncp}(j) > L_k^{ncp}(j')$ **then**
          Performing steps 7-9;
          $L_{unmatched}^{ncp} = L_{unmatched}^{ncp} \cup j'$;
          $L_{matched}^{k \leftrightarrow ncp} = L_{matched}^{k \leftrightarrow ncp} \setminus j'$;
        else
          $L_j^{ncp} = L_j^{ncp} \setminus k$;
        end
      end
    end
  end
end
return $L_{unmatched}^{ncp}$, $L_{matched}^{k \leftrightarrow ncp}$ and $\boldsymbol{L}^{ncp}$

feedback. Based on the above, the NCPs (or users) complete the unilateral selection to maximize their respective benefits.

For Algorithm 5 implemented by relays, it has two phases, i.e., reverse selection and matching evaluation. *Reverse selection*: Upon receiving a selection request from an NCP or user, the relay performs a reverse selection subject to the resource constraints being met. If the resource are sufficient, i.e., the matched NCPs (or users) do not exceed their own constraints, then the reverse selection is successful. Otherwise, the NCP (or user) needs to continue with subsequent assessment to compete for the relay. *Matching assessment*: Each NCP (or user) has competition awareness. When its preferred relay is full of supportable NCPs (or users), it will compare with NCPs (or users) that have already been selected by the relay. If the relay prefers it over the already selected NCPs, it can replace the least preferred existing NCP and thus successfully match with the relay. The eliminated NCP will then seek a new relay for matching.

4) Stability Analysis: The aim of *Tripartite* matching strategy is to find a stable matching among the three main entities, where the stability is the key concept of matching theory, defined as follows:

Definition 7 (Blocking Pair): The pair (j', k') is a blocking pair [39] for the matching Ψ, only if $j' \succ_{k'} j$, $j \in \Psi(k')$ and $k' \succ_{j'} k$, $k \in \Psi(j')$, for $j' \notin \Psi(k')$ and $k' \notin \Psi(j')$. Particularly, the definition of the blocking pair can be presented mathematically as

$$(\forall j \in \mathcal{M}, k \in \mathcal{K})((j, \Psi(j)), (k, \Psi(k))) \Rightarrow (\exists j' \in \mathcal{M}, k' \in \mathcal{K})((j', \Psi(j)), (k', \Psi(k))). \quad (22)$$

In other words, there exists a partnership (j', k') such that j' and k' are not matched with each other under the current matching Ψ but prefer to be matched with each other.

Theorem 5: The matching game Ψ is stable if it admits no blocking pair [41].

Proof: Assume that NCPs j_1 and j_2 match relays k_1 and k_2, respectively. If $k_2 \succ_{j_1} k_1$ ($k_2 \succ_{j_1} k_1$ implies that j_1 prefers k_2 to k_1. From j_1's standpoint, k_2 may bring greater benefits to j_1 than k_1.), then j_1 must have sent a request to k_2. However, k_2 rejected j_1 based on its preference list. Then we can know that $j_2 \succ_{k_2} j_1$, i.e., k_2 prefers j_2 to j_1, which leads to the non-existence of a case that j_1 prefers k_2 and k_2 also prefers j_1. According to ***Definition 7***, there are no blocking pairs. Thus, the matching game Ψ is proved to be stable. The same applies to the matching game Φ. □

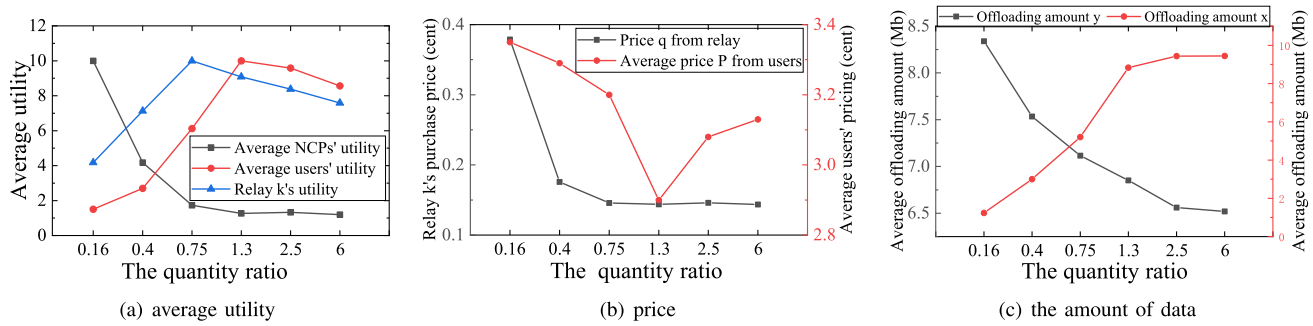
V. EVALUATION

A. Simulation Settings

To verify the effectiveness of the proposed algorithms, we consider a network architecture consisting of NCPs, relays and users. The relays are randomly and uniformly placed among NCPs and users. Each relay has different properties (e.g., bandwidth and purchase price cap). Inspired by [42], we set the number of NCPs and users that can be accommodated in each relay to follow random values of no more than 15 and 10, respectively. We consider that users are clustered in a concentrated area with a radius of no more than 200 meters, and NCPs are randomly distributed in a 4 km × 4 km 2D plane, respectively. The specific parameter settings are shown in Table II.

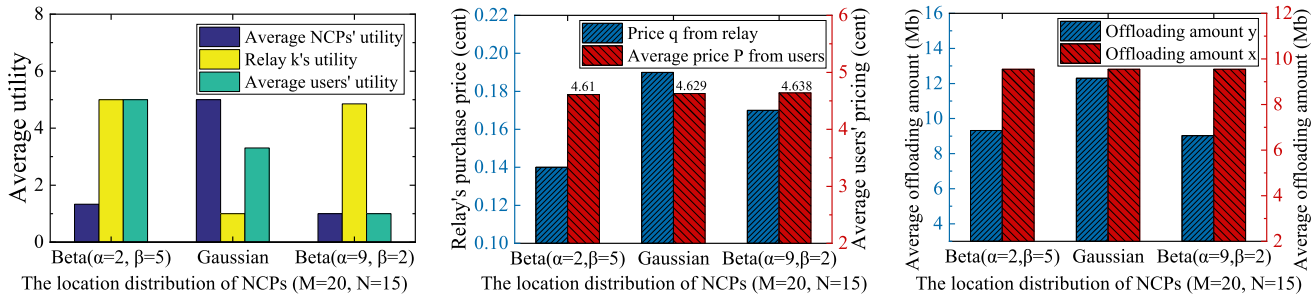
B. Baselines

For single-relay scenarios, to better demonstrate the advantages of the *Two-level game*, we design *Pro-price* and *Blind-matching* inspired by [43] and [44] for comparison in terms of entities' overall utilities and users' average energy consumption. In *Pro-price*, users' random pricing $\boldsymbol{P}$ are customized based on their individual configurations. The relay then assigns higher priority for computation offloading to



(a) average utility (b) price (c) the amount of data

Fig. 5. The impact of quantity ratio.



(a) average utility (b) price (c) the amount of data

Fig. 6. The impact of NCPs' location distribution.

users with higher pricing. For *Blind-matching*, users set prices randomly and the relay allocates offloading amounts $\boldsymbol{x}$ based on these prices. To facilitate comparison of the average energy consumption of users, we also add a *No-offloading* scenario.

For multi-relay scenarios, we design *Demand-first* and *Caps-first* inspired by [45] and [46] for comparison in terms of entities' overall utilities and users' average energy consumption. Moreover, a simplified version of *Tripartite* called *Unilateral matching* is also used as a comparison. *Demand-first*: when matching, users and NCPs prefer relays in proximity, while relays prefer users with high offloading demands and NCPs that can provide more offloading amount. *Caps-first*: when matching, users and NCPs prefer relays with more matches available, while relays prefer users with higher pricing caps and NCPs that can provide more offloading amount. *Unilateral matching*: the NCPs and relays are matched using the strategy proposed in this paper, however the users and relays are matched based on distance.

C. Single Relay

1) Performance Evaluation: We evaluate the impact of the quantity ratio and NCPs' location distribution on the average utility of the three main entities. The quantity ratio refers to the ratio between the number of NCPs and the number of users. There are three NCP distributions, *Beta*($\alpha = 2, \beta = 5$): Most NCPs are concentrated in the area near relay k; *Gaussian*: most NCPs are concentrated in the area which is moderately away from relay k; *Beta*($\alpha = 9, \beta = 2$): Most NCPs are concentrated in the area far from relay k.

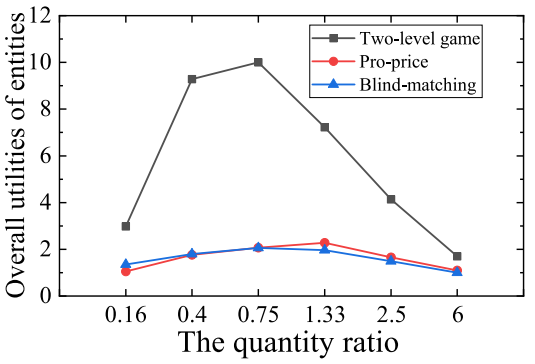
a) The impact of quantity ratio: As shown in Fig. 5(a), the average utility of users initially rises and then slowly decreases as the quantity ratio increases. The highest utility is achieved when the ratio is 1.3, as indicated by the peak in the curve. The reason is that at the beginning, there are fewer resource and more users, and the competition among users is fierce, so the pricing P (see Fig. 5(b)) of users is high and the amount of offloading x (see Fig. 5(c)) available to users is small, resulting in low utility. As the ratio goes up, the competition among users decreases as there are more NCPs in the system. This causes P to decrease and x to increase, resulting in a gradual increase in average utility until reaching a peak. However, when the number of users is far fewer than that of NCPs, the system is in a state of resource surplus. To protect its benefits from great damage, relay k plays a game with users, urging them to raise their prices appropriately, which ultimately leads to a decline in users' average utility after the peak. We can also see from Fig. 5(a) that the NCPs' average utility first declines and then stabilizes as the ratio increases. This is primarily due to the impact of relay k's purchase price q and amounts of offloading y provided by NCPs. As shown in Fig. 5(b), q reaches its maximum when the ratio is 0.16 due to a shortage of resources. However, as the ratio increases, the relay gradually reduces q for its benefit due to the abundant of computation resource, and finally maintains a lower level. Besides q and y, the NCPs' average utility is also affected by their individual energy constraints and idle resources. Thus, based on the above analysis, the NCPs' average utility is consistent with the change of q, and some slight fluctuations are caused by the diversity of NCPs' performance. For relay k, its utility is jointly determined by q, x, y and P. Based on their changing trends, the impact of ratio on relay k's utility can be well explained. When the ratio

is 0.75, the offloading purchased from NCPs are fully utilized. At this moment, supply and demand reach equilibrium, so that relay k's utility is maximized.

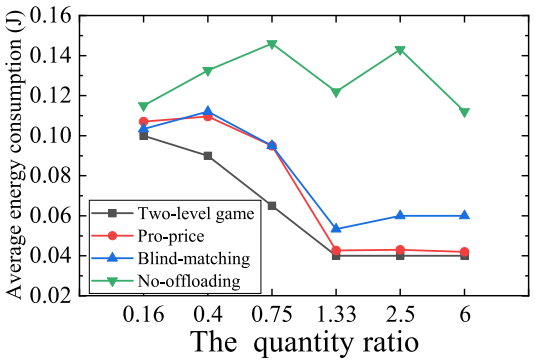
b) The impact of NCPs' location distribution: As shown in Fig. 6(a), three different NCP location distributions have distinct effects on the average utility of the three main entities. Among them, the NCPs' average utility is the lowest in *Beta*($\alpha = 9, \beta = 2$), and the greatest in *Gaussian*. This is because the farther the NCP is from the relay, the higher the cost required to return the computation result. Relay k therefore raises the purchase price q to encourage NCPs to contribute idle resources. We can know from Fig. 6(b) and Fig. 6(c) that within the acceptable cost range, the higher price q is, the more offloading amount y can be obtained. However, q will not be increased indefinitely. When NCP is far away, relay considers its interests and decreases q, and thus y is reduced. Also, we notice that obtaining the same y from a farther NCP requires a higher q than a closer NCP. Combining q and y, NCPs achieve the highest utility when the distribution is Gaussian. For relay k, its utility is higher in the distribution of *Beta*($\alpha = 2, \beta = 5$) and *Beta*($\alpha = 9, \beta = 2$). This is because the resource utilization is high in these two distributions, i.e., the difference between the offloading amount y obtained from NCPs and the offloading amount x provided to users is minimal. In terms of the users' average utility, it is highest in the *Beta*($\alpha = 2, \beta = 5$) and lowest in the *Beta*($\alpha = 9, \beta = 2$). The reason is that the relay gambles with users based on existing costs to maximize its utility as much as possible. Based on the equalization mechanism of the relay, the impact of NCPs' location distribution on the users' average utility can be easily explained. Specifically, if the relay increases price q, the user's pricing P will be raised accordingly (see Fig. 6(b)). Since x is the same (sufficient resource and a fixed number of users), the average utility is directly determined by P.

2) Comparisons and Analysis: In general, the proposed *Two-level game* has higher overall utilities and lower users' average energy consumption, which validates the theoretical results. The overall utilities of *Two-level game* outperform *Pro-price* and *Blind-matching*, which is improved by 67.6% and 72.7% compared with the two baselines on average. For the users' average energy consumption, *Two-level game* is 17.1% and 26.5% on average lower than *Pro-price* and *Blind-matching*, respectively.

a) The impact of quantity ratio on the overall utilities of three main entities: As shown in Fig. 7(a), the overall utilities of all entities under three algorithms all rise first and then fall with the increase of the ratio. The overall utilities of *Two-level game* are 71.9% and 72.6% higher than that of *Pro-price* and *Blind-matching*, respectively. This is because the objective of *Two-level game* is to maximize the utilities of three main entities. In the case of resource shortage or abundance, relay k can play games with NCPs and users, and finally make the utilities of three main entities reach an equilibrium. The *Blind-matching* results in high randomness and the lowest overall utilities due to its arbitrary decision-making. For *Pro-price*, the overall utilities are higher than that of *Blind-matching* when the ratio is greater than 0.75. This is because the offloading service is prioritized for users with high prices, which can only satisfy a small number of users willing to pay high prices, resulting in lower overall utilities.



(a) Overall utilities

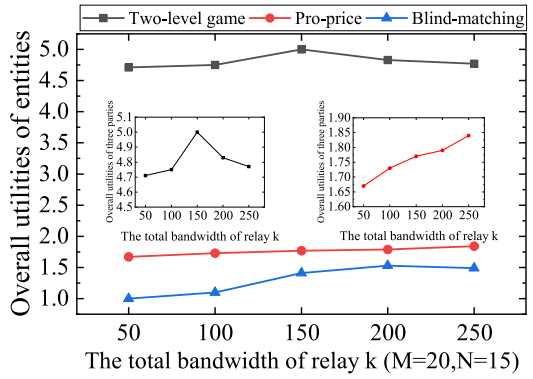


(b) Average energy consumption

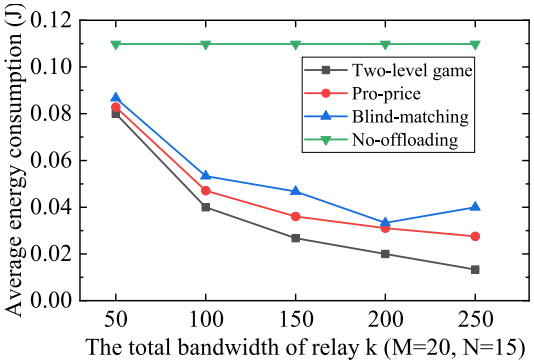
Fig. 7. The impact of quantity ratio.

b) The impact of quantity ratio on users' average energy consumption: We can find from Fig. 7(b) that as the ratio increases, the corresponding average energy consumption obtained from *Two-level game*, *Pro-price*, and *Blind-matching* all show a gradual downward trend. The reason is that with more resource and less demand, each user can utilize more offloading services. Then, the amount of data that needs to be processed by the user is reduced, thereby reducing the user's energy consumption. Among them, the users' average energy consumption of *Two-level game* is 14.6% and 22.4% lower than that of *Pro-price* and *Blind-matching*, respectively. This is because different from *Pro-price* and *Blind-matching*, energy consumption is considered as a cost in the utility function, and thus the energy consumption costs are minimized. Meanwhile, we notice that since all data is processed by users, the users' average energy consumption of *No-offloading* is always the highest. The slight fluctuations are caused by differences in the computation capability of different users.

c) The impact of relay k's total bandwidth on the overall utilities of three main entities: Fig. 8(a) reveals that the three algorithms are impacted differently as the total bandwidth B_k^{total} increases. Specifically, the overall utilities of *Pro-price* and *Blind-matching* show an upward trend as B_k^{total} increases. This is because B_k^{total} directly affects the transmission cost. As B_k^{total} increases, the cost decreases, leading to an increase in overall utilities. However, the overall utilities of *Two-level game* rise first and then fall, with 63.4% and 72.8% more



(a) Overall utilities



(b) Average energy consumption

Fig. 8. The impact of relay k's total bandwidth.

than *Pro-price* and *Blind-matching*, respectively. Besides the transmission cost, B_k^{total} also indirectly affects the decisions of the three entities in *Two-level game*. When B_k^{total} is large enough, the transmission cost is low and the offloading amount purchased by the relay increases. As a result, the supply exceeds the demand, and the overall utilities decrease.

d) The impact of relay k's total bandwidth on users' average energy consumption: As shown in Fig. 8(b), as B_k^{total} increases, the users' average energy consumption of *Two-level game*, *Pro-price*, and *Blind-matching* gradually decreases, while *No-offloading* consumes the most energy and remains unchanged. The reason is that users in *No-offloading* all choose local processing, which does not involve transmission energy consumption. Moreover, the energy consumption for processing is greater than that for transmission, resulting in the highest average energy consumption for users. For *Two-level game*, users consume the least energy on average, which is 19.7% and 30.7% lower than *Pro-price* and *Blind-matching*, respectively. This is because more offloading is available to the users in *Two-level game*. Therefore, as B_k^{total} increases, the average energy consumption of the user in *Two-level game* can be reduced the most.

D. Multiple Relays

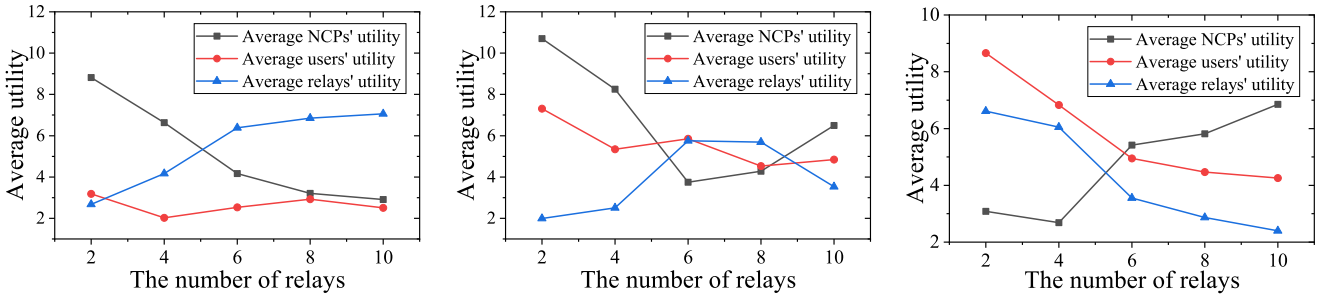
1) Performance Evaluation: We evaluate the impact of the number of relays on the average utility of three main entities for varying numbers of users and NCPs ($|M| = |N| = 120$, $|M| = |N| = 80$ and $|M| = |N| = 40$).

As shown in Fig. 9(a), when the number of users and NCPs exceeds the capacity of all available relays, the average utility of relays increases as the number of relays increases. This is because more relays achieve more offloading services. Furthermore, the relay has the option to select their preferred users and NCPs, leading to a higher utility. However, the average utility of NCPs slowly decreases due to competition for a limited number of relays. This leads to the fact that not every NCP can match their most preferred relay and therefore the best utility is not achieved. We find that the average utility of users does not fluctuate much, indicating that the number of relays has minimal impact on users. This is due to the game between users, which allows the users involved in the offloading to effectively protect their interests.

As shown in Fig. 9(b), the average utility of the relays rises first and then falls as the number of relays increases. The reason is that the number of relays is small at the beginning and the number of NCPs that can be accommodated is limited. However, as the number of relays gradually increases, the NCPs have more advantages and can choose their preferred relays. Then, the average utility of relays decreases, while that of NCPs increases. For users, it slowly decreases and then remains stable. We know that the users' average utility is mainly determined by the offloading amount provided by the relays and its pricing. When the relay's average utility decreases due to NCPs, it reduces the offloading amount provided to users for avoiding utility loss. If a user wants more offloading services at this time, it needs to increase the purchase price. However, users set pricing caps to protect their interests, thus the users' utilities will level off after a certain value.

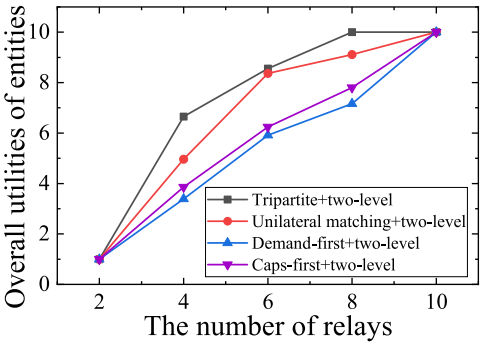
As depicted in Fig. 9(c), when the number of NCPs and users are much smaller than all relays can afford, the average utility of relays tends to decrease as the number of relays increases. This is because there are idle relays that cannot be matched with NCPs, making it impossible to provide offloading services to users. Consequently, relays with higher purchase prices compete for NCPs and provide services to the users. In contrast, NCPs can select their preferred relay and receive higher payment from the relay, leading to a gradual increase in the average utility. However, the users' average utility shows a declining trend. This is because the relay reduces offloading amount provided to the users in order to mitigate the decrease in utility, which causes the users to increase their prices.

2) Comparisons and Analysis: From Fig. 10(a), it is found that the overall utilities of all entities under the four methods show an upward trend as the number of relays increases (see reasons analyzed in Fig. 9(a)). Specifically, *Tripartite* and *Unilateral matching* perform better than *Demand-first* and *Caps-first*. This is because *Tripartite* considers factors such as bandwidth, distance, purchase price and the number of relays that can be matched in the utility function. It allows the three main entities to achieve the best matching results, thus maximizing their utilities. However, *Unilateral matching* only considers the best matching between NCPs and relays, resulting in less utility than *Tripartite*. The *Demand-first* focuses on users' offloading demands and the amount of offloading

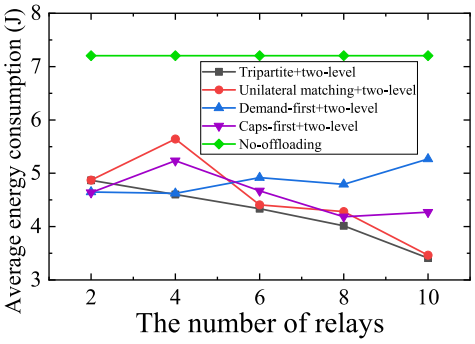


(a) average utility ($|M| = |N| = 120$) (b) average utility ($|M| = |N| = 80$) (c) average utility ($|M| = |N| = 40$)

Fig. 9. The number of relays.



(a) Overall utilities



(b) Average energy consumption

Fig. 10. The number of relays.

available from NCPs, but ignores offloading costs, resulting in the lowest performance. Given the number of relays that can be matched and the pricing caps of users, *Caps-first* is slightly better than *Demand-first*, however its utility remains undesirable compared to the fully considered *Tripartite*.

As shown in Fig. 10(b), the energy consumption of *Tripartite* and *Unilateral matching* decreases as the number of relays increases. Moreover, the energy savings of *Tripartite* are optimal. This is because more users have access to offloading services based on the best matching, and reduce their energy overhead. The offloading energy consumption has been shown to be much lower than the local computing energy consumption. This is due to the fact that users select relays by considering distance, bandwidth and other factors to make their offloading costs as low as possible. However, *Demand-first* and *Caps-first* show instability, i.e., the increase in the number of relays fails to help more users to obtain better offloading services. The reason for this is that incomplete consideration in the selection of relays has resulted in many users or NCPs not being matched to the appropriate relay, or even not being matched.

VI. Conclusion

In this paper, we propose a dispersed service framework to effectively alleviate the urgent computation overload in practically challenging scenarios. A game-based incentive-driven offloading mechanism is developed to encourage the participation of the three main entities. Depending on the number of users, we design solutions for small-scale scenarios with single relay and large-scale scenarios with multiple relays, respectively. For single-relay scenarios, we construct a two-level Stackelberg game by leveraging a single relay as a bridge to achieve a Nash equilibrium among the three main entities. For multi-relay scenarios, we develop a tripartite matching strategy to assign appropriate relays to users and NCPs, allowing the multi-relay problem to be decomposed into multiple single-relay problems. We conduct extensive simulations and the results show that the proposed mechanism provides a feasible solution to an accidental and urgent resource exhaustion in a network.

Acknowledgment

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of A*STAR.

References

[1] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world from edge to core, data age 2025," IDC, Needham, MA, USA, Tech. Rep. 3, Nov. 2018.

[2] T. Q. Dinh, Q. D. La, T. Q. S. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6353–6367, Dec. 2018.

[3] H. Lu, M. Chen, and W. Kuang, "The impacts of abnormal weather and natural disasters on transport and strategies for enhancing ability for disaster prevention and mitigation," *Transp. Policy*, vol. 98, pp. 2–9, Nov. 2020.

[4] H. Chen, B. Guo, Z. Yu, and Q. Han, "CrowdTracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7570–7583, Oct. 2019.

[5] A. Silvestro, N. Mohan, J. Kangasharju, F. Schneider, and X. Fu, "MUTE: Multi-tier edge networks," in *Proc. 5th Workshop CrossCloud Infrastruct. Platforms (CIP Workshops)*, 2018, pp. 1–6.

[6] N. Mohan, A. Zavodovski, P. Zhou, and J. Kangasharju, "Anveshak: Placing edge servers in the wild," in *Proc. Workshop Mobile Edge Commun.*, Aug. 2018, pp. 7–12.

[7] L. Lovén et al., "Scaling up an edge server deployment," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2020, pp. 1–7.

[8] Y. Mao, C. Zhou, Y. Ling, and J. Lloret, "An optimized probabilistic delay tolerant network (DTN) routing protocol based on scheduling mechanism for Internet of Things (IoT)," *Sensors*, vol. 19, no. 2, p. 243, Jan. 2019.

[9] Y. Wu, Y. Wang, W. Hu, X. Zhang, and G. Cao, "Resource-aware photo crowdsourcing through disruption tolerant networks," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 374–383.

[10] S. Datta and S. Madria, "Efficient photo crowdsourcing with evolving POIs under delay-tolerant network environment," *Pervasive Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101187.

[11] J. A. D. Cruz, N. J. Libatique, and G. Tangonan, "Design of a disaster information system using mobile cloud wireless mesh with delay tolerant network," in *Proc. IEEE Global Hum. Technol. Conf. (GHTC)*, Oct. 2019, pp. 1–8.

[12] M. R. Schurgot, M. Wang, A. E. Conway, L. G. Greenwald, and P. D. Lebling, "A dispersed computing architecture for resource-centric computation and communication," *IEEE Commun. Mag.*, vol. 57, no. 7, pp. 13–19, Jul. 2019.

[13] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, "Communication-aware scheduling of serial tasks for dispersed computing," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1330–1343, Aug. 2019.

[14] A. Knezevic et al., "CIRCE—A runtime scheduler for DAG-based dispersed computing: Demo," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, Oct. 2017, pp. 1–2.

[15] P. Ghosh, Q. Nguyen, and B. Krishnamachari, "Container orchestration for dispersed computing," in *Proc. 5th Int. Workshop Container Technol. Container Clouds*, Dec. 2019, pp. 19–24.

[16] H. Wu et al., "Resolving multitask competition for constrained resources in dispersed computing: A bilateral matching game," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16972–16983, Dec. 2021.

[17] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[18] X. Xu et al., "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Gener. Comput. Syst.*, vol. 95, pp. 522–533, Jun. 2019.

[19] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[20] K. Zhang, X. Gui, D. Ren, and D. Li, "Energy–latency tradeoff for computation offloading in UAV-assisted multiaccess edge computing system," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6709–6719, Apr. 2021.

[21] M. Mukherjee, V. Kumar, A. Lat, M. Guo, R. Matam, and Y. Lv, "Distributed deep learning-based task offloading for UAV-enabled mobile edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 1208–1212.

[22] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, Oct. 2015.

[23] C. Li, Y. Fu, F. R. Yu, T. H. Luan, and Y. Zhang, "Vehicle position correction: A vehicular blockchain networks-based GPS error sharing framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 898–912, Feb. 2021.

[24] L. A. Grieco, G. Boggia, G. Piro, Y. Jararweh, and C. Campolo, "Ad-hoc, mobile, and wireless networks," in *Proc. 19th Int. Conf. Ad-Hoc Netw. Wireless (ADHOC-NOW)*. Bari, Italy: Springer, 2020, pp. 19–21.

[25] Y. Su, X. Liu, G. Han, and X. Fu, "A traffic load-aware OFDMA-based MAC protocol for distributed underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10501–10513, Oct. 2021.

[26] Z. Wei, S. Sun, X. Zhu, D. In Kim, and D. W. K. Ng, "Resource allocation for wireless-powered full-duplex relaying systems with nonlinear energy harvesting efficiency," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12079–12093, Dec. 2019.

[27] J. Liu, K. Xiong, D. W. K. Ng, P. Fan, and Z. Zhong, "Optimal design of wireless-powered hierarchical fog-cloud computing networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[28] M. Andrews, A. Fernandez Anta, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *Proc. Conf. Comput. Commun. (INFOCOM)*, Mar. 2010, pp. 1–9.

[29] Y. Wang, Z. Su, and N. Zhang, "BSIS: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3620–3631, Jun. 2019.

[30] Y. Qiao, Y. Li, and J. Li, "An economic incentive for D2D assisted offloading using Stackelberg game," *IEEE Access*, vol. 8, pp. 136684–136696, 2020.

[31] O. Candogan, K. Bimpikis, and A. Ozdaglar, "Optimal pricing in networks with externalities," *Oper. Res.*, vol. 60, no. 4, pp. 883–905, Jul. 2012.

[32] X. Wang, Z. Sheng, S. Yang, and V. C. M. Leung, "Tag-assisted social-aware opportunistic device-to-device sharing for traffic offloading in mobile social networks," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 60–67, Aug. 2016.

[33] G. Haeser and A. Ramos, "Constraint qualifications for Karush–Kuhn–Tucker conditions in multiobjective optimization," *J. Optim. Theory Appl.*, vol. 187, no. 2, pp. 469–487, Nov. 2020.

[34] X.-Q. Pham, T. Huynh-The, E.-N. Huh, and D.-S. Kim, "Partial computation offloading in parked vehicle-assisted multi-access edge computing: A game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10220–10225, Sep. 2022.

[35] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 724–738, Jan. 2019.

[36] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, nos. 1–2, pp. 281–302, Jan. 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042700004337

[37] W. Hao, O. Muta, H. Gacanin, and H. Furukawa, "Dynamic small cell clustering and non-cooperative game-based precoding design for two-tier heterogeneous networks with massive MIMO," *IEEE Trans. Commun.*, vol. 66, no. 2, pp. 675–687, Feb. 2018.

[38] S. Koskie and Z. Gajic, "A Nash game algorithm for SIR-based power control in 3G wireless CDMA networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1017–1026, Oct. 2005.

[39] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 103–122, Nov. 2016.

[40] T. Guo, Z. Zhao, and X. Han, "The discourse analysis of social factors influencing interest contention in business dispute settlement: A perspective of discourse information theory," *Asian Social Sci.*, vol. 15, no. 3, pp. 46–57, 2019.

[41] F. Cooper, "Fair and large stable matchings in the stable marriage and student-project allocation problems," Ph.D. dissertation, College Sci. Eng., Univ. Glasgow, Glasgow, U.K., 2020.

[42] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[43] L. Li, M. Siew, T. Q. S. Quek, and Z. Chen, "Optimal pricing for job offloading in the MEC system with two priority classes," 2019, *arXiv:1905.07749*.

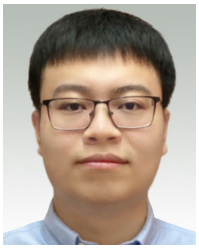
[44] W. Huang, W. Chen, and H. V. Poor, "Pricing for content pushing with request delay information: A Stackelberg game approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[45] S. Yuan, G. Xia, J. Chen, and C. Yu, "Toward dispersed computing: Cases and state-of-the-art," in *Proc. 17th Int. Conf. Mobility, Sens. Netw. (MSN)*, Exeter, U.K., Dec. 2021, pp. 710–717.

[46] Q. Meng and Z. Zhang, "Blockchain-based price priority matching two-stage energy trading algorithm," in *Proc. Asia Conf. Electr., Power Comput. Eng. (EPCE)*, New York, NY, USA, Apr. 2022, pp. 1–6.

Hongjia Wu received the B.S. degree from Liaoning Technical University in 2016 and the M.S. degree from Guangxi University, China, in 2018. She is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT). She was a Visiting Student with the Singapore University of Technology and Design. Her research interests include computation offloading, game theory, resource allocation, and dispersed computing.

Jiangtian Nie (Member, IEEE) received the B.Eng. degree (Hons.) in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from the Interdisciplinary Graduate School, ERI@N, Nanyang Technological University (NTU), Singapore. She is currently a Research Fellow with NTU. Her research interests include network economics, game theory, and crowd sensing and learning.



Zehui Xiong (Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore. He was a Visiting Scholar with Princeton University and the University of Waterloo. He is currently an Assistant Professor with the Singapore University of Technology and Design and also an Honorary Adjunct Senior Research Scientist with the Alibaba-NTU Singapore Joint Research Institute, Singapore. He has published more than 150 research papers in leading journals and flagship conferences and many of them are ESI highly cited papers. His research interests include wireless communications, the Internet of Things, blockchain, edge intelligence, and metaverse. He has won over ten best paper awards in international conferences and is listed in the world's top 2% scientists identified by Stanford University. He was a recipient of the IEEE Early Career Researcher Award for Excellence in Scalable Computing, the IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, the IEEE Internet Technical Committee Early Achievement Award, the IEEE TCI Rising Star Award, the IEEE TCCLD Rising Star Award, the IEEE Best Land Transportation Paper Award, the IEEE CSIM Technical Committee Best Journal Paper Award, the IEEE SPCC Technical Committee Best Paper Award, the IEEE VTS Singapore Best Paper Award, the Chinese Government Award for Outstanding Students Abroad, and the NTU SCSE Best Ph.D. Thesis Runner-Up Award. He is also serving as the Associate Director of the Future Communications Research and Development Program. He is also serving as an Editor or a Guest Editor for many leading journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.

Zhiping Cai received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data. He is a Distinguished Member of the CCF.

Tongqing Zhou received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, in 2012, 2014, and 2018, respectively. He is currently an Assistant Researcher with the College of Computer, NUDT. His main research interests include crowdsensing and data privacy. He was a recipient of the Outstanding Ph.D. Dissertation Award and Outstanding Post-Doctoral in Hunan, China.

Chau Yuen (Fellow, IEEE) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively.

He was a Post-Doctoral Fellow with the Lucent Technologies Bell Laboratories, Murray Hill, in 2005, and a Visiting Assistant Professor with The Hong Kong Polytechnic University in 2008. From 2006 to 2010, he was with the Institute for Infocomm Research, Singapore. From 2010 to 2023, he was with the Engineering Product Development Pillar, Singapore University of Technology and Design. Since 2023, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University. He has published three U.S. patents and published over 500 research papers at international journals or conferences.

Dr. Yuen was a recipient of the Lee Kuan Yew Gold Medal, the Institution of Electrical Engineers Book Prize, the Institute of Engineering of Singapore Gold Medal, the Merck Sharp and Dohme Gold Medal, and twice a recipient of the Hewlett Packard Prize. He received the IEEE Asia–Pacific Outstanding Young Researcher Award in 2012 and IEEE VTS Singapore Chapter Outstanding Service Award in 2019. He currently serves as the Editor-in-Chief for *Computer Science* (Springer Nature), an Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE SYSTEM JOURNAL, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, where he was awarded as an IEEE TNSE Excellent Editor Award and a Top Associate Editor for TVT from 2009 to 2015. He is a Distinguished Lecturer of IEEE Vehicular Technology Society and also a Highly Cited Researcher by Clarivate Web of Science. He also served as the Guest Editor for several special issues, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Wireless Communications Magazine*, *IEEE Communications Magazine*, *IEEE Vehicular Technology Magazine*, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and *Applied Energy* (Elsevier).

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.