Accepted Manuscript

GoSharing: An intelligent incentive framework based on users' association for cooperative content sharing in mobile edge networks

Shuyun Luo, Zhenyu Wen, Xiaomei Zhang, Weiqiang Xu, Albert Y. Zomaya, Rajiv Ranjan

PII: S0167-739X(18)32006-5
DOI: https://doi.org/10.1016/j.future.2019.01.013
Reference: FUTURE 4706

To appear in: *Future Generation Computer Systems*

Received date : 20 August 2018
Revised date : 19 November 2018
Accepted date : 9 January 2019

Please cite this article as: S. Luo, Z. Wen, X. Zhang et al., GoSharing: An intelligent incentive framework based on users' association for cooperative content sharing in mobile edge networks, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.01.013

GoSharing: An Intelligent Incentive Framework based on Users' Association for Cooperative Content Sharing in Mobile Edge Networks

Shuyun Luo[a], Zhenyu Wen[b], Xiaomei Zhang[c], Weiqiang Xu[a], Albert Y Zomaya[d], Rajiv Ranjan[b]

[a]*School of Information Science and Technology, Zhejiang Sci-tech University, China*
[b]*School of Computing, Newcastle University, United Kingdom*
[c]*Department of Mathematics and Computational Science, University of South Carolina Beaufort, USA*
[d]*School of Information Technologies, University of Sydney, Australia*

Abstract

In most metropolises, commuters spend a considerable amount of time on public transport, and many of them entertain themselves with the content (like music or videos) on their mobile devices to alleviate boredom. Currently, the content, usually shared in co-located wireless networks to avoid huge monetary cost of using cellular data, is delivered from single host (resource owner) to single request user, which brings low transmission quality, due to the uncertainty of mobile edge networks in public transport environments.

In this paper, we present an intelligent incentive framework called GoSharing which encourages multiple hosts to share content collaboratively to improve delivery quality, by taking advantage of users' association and consideration of network Quality of Service(QoS) requirements. The highlight of GoSharing is the novel Association-based Intelligent incentive mechanism that consists of three key components. First, a Fast Candidate Generation algorithm discovers users' association according to their stored content and QoS requirements and filters the candidate groups from large host groups. Second, a Host Selection algorithm finds a near-optimal solution among candidate groups within an

Email addresses: `shuyunluo@zstu.edu.cn` (Shuyun Luo), `Zhenyu.Wen@newcastle.ac.uk` (Zhenyu Wen), `xiaomei@uscb.edu` (Xiaomei Zhang), `wqxu@zstu.edu.cn` (Weiqiang Xu), `albert.zomaya@sydney.edu.au` (Albert Y Zomaya), `raj.ranjan@ncl.ac.uk` (Rajiv Ranjan)

approximate factor of $F(d)$, where d denotes the maximum size of completed tasks when any candidate group is selected. Last but not least, a Payment Determination algorithm determines the payment of resource contributors while guaranteeing the truthfulness of their bids based on the procurement auction. Both theoretical analysis and extensive simulations demonstrate that GoSharing not only effectively motivates hosts' collaborative sharing, but also achieves the properties of *truthfulness, individual rationality, high computational efficiency, low overpayment ratio*, and high download ratio.

Keywords: GoSharing, Cooperative System, Users' Association, Intelligent Incentive Mechanism, Mobile Edge Networks

---

1. Introduction

In many dense crowded metropolises in Asia and Europe, as well as some US cities like New York city and San Francisco, driving fails to be a good option for daily commuting because of traffic jams and the unavailability of parking. Public transport such as buses, subways and trains becomes the best choice for urban citizens. As a result, a considerable proportion of the people in these metropolises tend to choose public transport for their daily commute. A study from the Singapore Management University reported that the average one-way commute time in Singapore is about 26 minutes [1], and other surveys show that the average commuting time of the urban citizens is very long, e.g. 40 minutes for New York City [2], 66 minutes for Tokyo [3], and 97 minutes for Beijing [4]. Entertainment, such as watching videos, becomes the first choice for the commuters to kill the long commute time. Although the recent popularization of cellular networks (e.g. 3G/LTE) provides mobile users with ubiquitous Internet access, high cellular data cost and network latency prevent the cellular networks from being a good way for video downloading. To address this problem, a promising solution is to utilize short-range wireless network interfaces, such as WiFi and Bluetooth, to exchange the media content in neighboring devices.

The current research on short range communications usually focuses on data transmission from single host (resource owner) to single request user called *single-host* model [5, 6, 7, 8]. However, based the single-host communication model, thus reliability of communications cannot be guaranteed because users randomly pop-in and pop-out. For example, mobile users usually have unpredictable mobility, if one of the communication nodes moves beyond the transferring range, the content will not be delivered successfully. To address this problem, we envision that multiple co-located devices can share content cooperatively to enhance the reliability of content sharing among commuters.

In addition, sharing media content requires hosts to contribute not only their content but also hardware, especially battery. To stimulate hosts to share their resources, incentives like monetary rewards should be provided to the hosts. In the literature, various incentive mechanisms have been proposed in mobile networks [9, 10, 11, 12, 13, 14, 15, 16].

Some mechanisms, e.g.[9, 10, 11, 12, 13], are designed for tasks that only require a single user (host) to perform, referred to as *simple tasks*. In our content sharing scenario, every downloading task needs the cooperation of multiple users (hosts), referred to as *cooperative tasks*. There is no existing incentive mechanism that is designed for rewarding the participates with *multiple cooperative tasks*, which comes up with new challenges, especially how to combine users' association for efficient cooperation in such complex scenario.

In this paper, we propose GoSharing, an intelligent incentive framework which motivates resource owners to share their stored videos cooperatively in mobile edge networks. GoSharing is able to achieve the goals of encouraging commuters to share their content cooperatively with the minimum incentive cost based on users' association and guaranteeing the Quality of Service(QoS) of the task sharing. The main intellectual contributions of this work are summarized as follows:

1. In order to improve the reliability of content sharing in mobile edge networks, we present a multi-host communication model to allow multiple resource owners to share their content collaboratively.

2. We measure the factors that impact the quality of data delivery from hosts to the request users on public transport. Based on the experimental results, we formalize a network QoS model to describe the tradeoff between reliability and download time.

3. To motivate hosts to share their content collaboratively, we design an intelligent incentive framework, GoSharing, whose highlight, Association-base Intelligent (AI) incentive mechanism composed of candidate generation, host selection and payment determination, which has four desirable properties: a) truthfulness, b) individual rationality, c) computational efficiency, d) low overpayment ratio, as well as high download ratio.

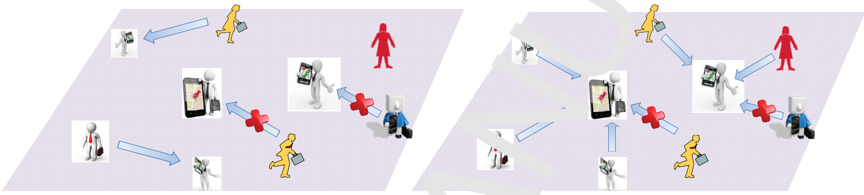
The rest of this paper is organized as follows. Section 2 provides the experimental observations and results to verify the efficiency of the multi-host model. Section 3 presents the overview of GoSharing framework and system model. In Section 4, we present the design of AI incentive framework and prove its desirable properties. Section 5 evaluates the performance of our proposed mechanism. Finally, Section 6 reviews related work and Section 7 concludes this paper as well as outlining future work.

2. Motivation and Preliminary Results

In this section, we first illustrate the unreliability problem of the single-host model in mobile edge networks, then demonstrate the motivation of the GoSharing system model, i.e. the multi-host model. Finally, we have the measurements and experiments in real scenarios to verify the motivation and analyze the factors that influence the QoS communications.

2.1. Motivation

Most content sharing applications are based on the single-host model, as shown in Figure 1 (a). In this model, once the sender or receiver move out of communication range during the video sharing period, the downloading task will fail. Thus, the task has to be re-started from the beginning. In order to mitigate this unreliability problem of the single-host model, we propose the multi-host model, shown in Figure 1 (b) in which multiple hosts are sharing the downloading file simultaneously. Furthermore, if any provider (host) moves out of communication range, the rest of the hosts can continuously provide the required sources until the completion of the task. To confirm the above assumption, we made some real world measurements for both single-host model and multi-host model in §2.2 and §2.3.



(a) a single-host model (b) a multi-host model

Figure 1: A motivating scenario of GoSharing

2.2. The Unreliability of the Single-host Model

In this section, a number of real world experiments are conducted for detecting the download ratio in mobile opportunistic networks and capturing the factors which influence transmission rates.

Taking wifi-P2P (wifi-direct) as an example, the IEEE standard claims its theoretic maximum transmission rate is 250 $Mbps$, and the maximum transmission range is 200m [17]. However, in practical environments, such an upper bound cannot be reached. Hence, we conduct the experiments to detect the real transmission rates of wifi-direct in a real environment. Two Samsung Note3 smartphones with 3G RAM are used on buses and subways to measure the transmission rate of wifi-direct with various peer distances (the distance from single host to single requester) in three different conditions, i.e., crowded, normal and almost empty.

(a) Empty (b) Normal (c) Crowded

Figure 2: Three statuses on the subway

file size(M)	time(s)	rate(M/s)
12.3	1.96	6.28
39.8	8.35	4.77

Less than one carriage (17m)

file size(M)	time(s)	rate(M/s)
39.8	52	0.77
123	100	1.23

One and a half carriages (30m)

file size(M)	time1(s)	time2(s)
39.8	26	65
123	[illegible]	fail

Two carriages (40m)

Figure 3: The transmission rate on the subway when the carriage is almost empty

On the subway, we monitor the transmission rates under three statuses: empty, normal and crowded. Figure 2 shows the status of the circumstances throughout conducting the experiments. From Figure 3, the transmission rate changes dramatically at different peer distances when the subway is almost empty. This illustrates that the transmission rate is relatively stable at the range of 4 to $6.5MBps$ when the peer distance is less than the length of one carriage, about $17m$. However, when the peer distance increases to $30m$, the transmission rate is vastly reduced, whereas the connection is broken when the peer distance extends to the length of two carriages, about $40m$.

Figure 4 shows that the transmission rate fluctuates dramatically with the same peer distance when the carriage status is normal. This implies that the maximum transmission range is limited to one carriage when the status is crowded. The experimental results indicate that the transmission rate depends on not only the peer distance and crowdedness of the carriage, but also some unexpected factors, such as users' behaviors and wireless interference. Hence,

file size(M)	time1(s)	time2(s)	rate(M/s)
39.8	52	26.1	1.02
123	60	108	1.46

Normal + One carriage (20m)

file size(M)	time(s)
39.8	350
123	fail

Crowded + One carriage (20m)

Figure 4: The transmission rates on the subway when the carriage is normal or crowded.

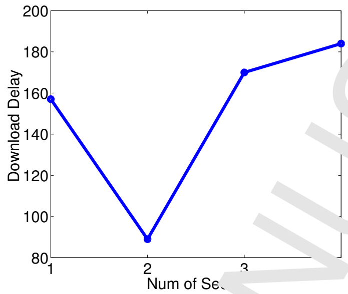


Figure 5: Impact on download time

the single-host model is not applicable for sharing video among devices on pubic transport.

2.3. The Performance of the Multi-host Model

Since the single-host model fails to provide a reliable network to share video among the commuters an alternative multi-host model can reduce the chance of the failures caused by the unavailability of hosts. This subsection shows the performance of the multi-host model. BitTorrent Sync[1], a BT protocol based file sharing tool is used to create a multi-host network.

We have conducted the experiments on Android testbeds which contain one Galaxy Note3, two Sony and two HuaWei running Andriod 5.0. We evaluate the relationship between the number of hosts and download time in the multi-host mode. The Sony smartphone is a receiver, and we record the download time of a media file with 350M under different numbers of hosts with the tool of BitTorrent Sync, as shown in Figure 5. The observations show that the download time reaches the lowest point when there are two hosts, and then

[1] https://www.getsync.com/

ascends with the increase of hosts' size. This is because in the BT protocol each pair of receiver and sender has to create a channel, and a new channel will cost some bandwidth. Therefore, when the total transmission rate (sum of the transmission rate of created channels) is higher than the throughput of the device (receiver), the transmission rate of each channel will reduce. As shown Figure 5, when there are three hosts, the throughput of the receiver is lower than the counterpart in the case of one host.

As the measurement results show the multi-host model can efficiently improve the communication reliability in mobile edge networks. However, the multi-host model requires hosts for cooperative sharing, which needs to select host groups for multiple tasks completion. In order to encourage hosts to share content with efficient cooperation, it is vital to find hosts' association based on their stored content and condition of link quality, such that we can obtain the filtered host groups for further host selection within the design of the new incentive mechanism. In this paper, we propose the intelligent incentive framework, GoSharing which can be adapted to the multi-host model with the benefit of hosts' association for efficient cooperation, the goal of minimizing incentive cost, as well as ensuring the QoS.

3. System Model and Problem Formulation

In this section, we give an overview of the GoSharing framework, QoS model, system model and problem formulation.

3.1. Framework

The GoSharing framework supports cooperative systems with multiple tasks, and can be applied to scenarios where many people gather for a transient period, such as public transport, conference, supermarket checkout and hospital queuing. etc. GoSharing consists of a set of hosts and request users, also a local-based server, such as the near base station. To solve the time-consuming problem of neighbor discovery in co-located networks [1], a local-based server

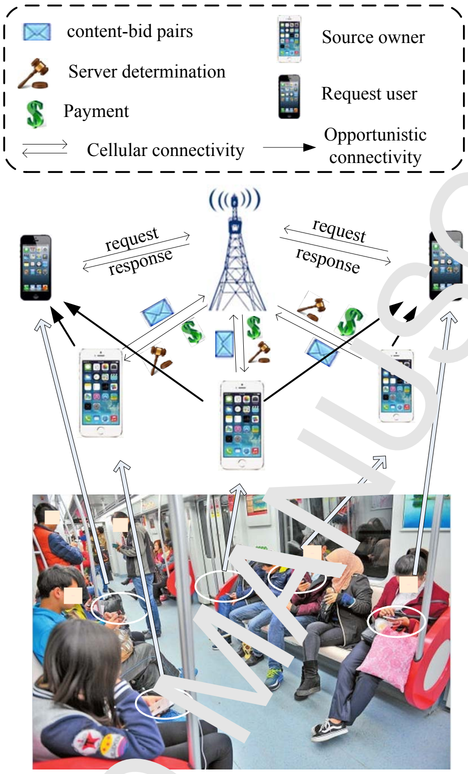


Figure 6: One realistic scenario of GoSharing

uses cellular networks to collect hosts' information in the covered area, such as content list, host's location and behavior histories, which generates little or even negligible data traffic and cellular cost. Figure 6 illustrates one of the realistic scenarios of using our GoSharing framework, i.e. the commuters want to share content with neighbors on the subway. The server acts as the buyer who offers the monetary payment to the hosts who are willing to share their content. The host plays the role of seller who is encouraged to submit a list, listing the content that s/he is willing to share; and make a bid for the cost of sharing their content.

3.2. Network QoS Model under Hosts' Mobility

From the measured results in section 2, the dynamic connection between two mobile devices significantly influences data transmission in terms of download time and download ratios, defined as follows.

- Download time: The download time is recorded as the average time starting from the local network being established to the media files being downloaded.
- Download ratio: The download ratio of media files is computed as the number of downloaded tasks to allocated tasks.

Firstly, we develop a link quality (LQ) model that represents the probability of a request user i successfully downloading files from a single host, defined as

$$LQ(i) = \alpha(dis_i) * \beta(tim_i) * \gamma, \ (0 \leq \alpha, \beta, \gamma \leq 1) \tag{1}$$

where α is a parameter linked to the proper distance from host i to the request user, denoted as dis_i, and β is determined by the host's mobile behavior, which can be predicted by historical information, such as his commute behavior (the simple way is to record a host's entry time tim_i which can be known from collected historical information and we can predict the probability of users' departure), and γ is a factor to depict the unexpected wireless interference. Note that the functions of α, β and γ can be defined specifically under various scenarios, which is beyond the scope of this paper.

Since the reliability of media download depends on how many links work, the download ratio can be calculated by

$$\rho_{download} = 1 - \prod_{i \in G_{t_j}} (1 - LQ(i)), \tag{2}$$

where G_{t_j} is the candidate groups (Definition 1) for task t_j.

Next, we record the download time under both models, in which LQs are assumed to obey the Poisson distribution with $\lambda = 2$. Based on the measured results of wifi-direct transmission in Figure 4 and sync BT in Figure 5 (a), the

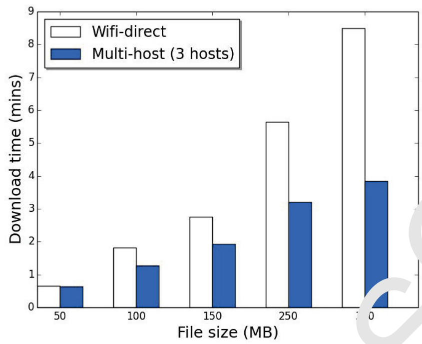


Figure 7: single-host model vs. multi-host model (when any host interrupts during the file download, single host model adopts the rule of retransmission from the beginning, while the remaining host(s) will continue to keep transmitting the remaining data in multi-host model)

transmission rates are set to be 1.46 MBps and 1.3 MBps for the single-host model and the multi-host model, respectively. Since the link quality dynamically changes, the communication may be interrupted during the file download. The rule of retransmission is adopted from the beginning in the single host model, while in the multi-host model, when any host interrupts their transmission, the remaining host(s) will continue to keep transmitting the remaining data. From Figure 7, it could be deduced that the download time has been reduced significantly with the media file in larger size in the multi-host model, compared with the case in the single host model. Therefore, the multi-host model is an efficient method to improve communication reliability under network uncertainty, especially in the case of sharing large files.

From the above analysis, we can draw up the rule that the download ratio is increased by the number of hosts, while the download time decreases as the number of hosts increase until the total transmission rate (sum of the transmission rate of created channels) exceeds the wireless capacity. To solve the tradeoff, we assume that the candidate groups are the ones with two constraints: 1) satisfying the requirement of the download ratio; 2)minimizing download time, defined as follows.

Definition 1 (Candidate Group). *Combined with the QoS requirements of* *download task t_j, the definition of candidate group is the host group with two constraints: 1) whose link qualities satisfy $\rho_{download} \geq \rho_{th}$, where ρ_{th} is the threshold of download ratio; 2) with the minimum number of hosts.*

Therefore, the problem of discovering all candidate groups can be decomposed into two subproblems:

1. Find all host groups that can satisfy QoS requirements, i.e. the first constraint in Definition 1.

2. Prune the large set of host groups with the second constraint in Definition 1, and generate the candidate groups.

3.3. System Model

Users first send their download requests to server s, and each requested media file represents a task. This set of tasks is denoted as $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$, thus M is the total number of tasks, i.e., $|\mathcal{T}| = M$. According to the host information (locations and resource lists) collected by cellular networks, the server can detect request users' neighboring hosts and corresponding available resources.

All detected hosts are symbolized as $\mathcal{U} = \{1, \ldots, N\}$, where N is the size of $\mathcal{U}$. If the request user launches the download task t_j, and downloads successfully from local neighbors, the server can therefore obtain a revenue r_{t_j}, which depends on the popularity of t_j. Each host i has stored a set of media files (music or videos). When request users launch a set of download tasks T_i, $T_i \subset \mathcal{T}$, the hosts can share their content with them. Accordingly, host i has an associated cost c_i, which is private and only known to itself. Traditionally, c_i is related to the number of request users that hosts support [12]. However, in our scenario, a host cannot detect how many users he can provide resource to, thus each host bids its cost based on its available time period, stored content and remaining energy. Although the local-based edge server can require the information about how many users access its resource, it does not know the real-time wireless link qualities between the host and request users. Actually, the link qualities can be very dynamic because of the frequent user movements. As a result, the number of users that actually access the host is also dynamic in real-time and unknown to both the edge server and the host. In addition, each host announces the content-bid pair (T_i, b_i) as well as the location information to the server, where b_i is the reserve price that host i wants to charge for sharing its content.

Table 1 lists the frequently used notations.

Table 1: Notation List

Notation	Description
$\mathcal{T}$	the set of all allocated tasks
M	the total number of tasks
N	the total number of hosts
T_i	the tasks host i has capacity to participate in
$\mathcal{S}$	the host winners
$\mathcal{S}_{curr}$	the current selected hosts
$\mathcal{S}_k$	the selected hosts in the κ-th iteration
Λ	the set of current completed tasks
Λ_k	the set of new completed task in the k-th iteration
$SubU_{t_j}$	users able to perform task t_j
S_{t_j}	the hosts with the capacity to perform task t_j
G_{t_j}	for task t_j, the candidate groups
G	for all tasks, the candidate groups
r_{t_j}	the revenue that the server obtained by finishing task t_j
p_i	the payment for each host i
g	one candidate group

3.4. Utility Functions

Based on the content-bid pairs received from each host, the server selects a subset of hosts $\mathcal{S}_{winner} \subset \mathcal{U}$ as winners and computes the payment p_i for each host i. How to select hosts and determinate the payment will be discussed in

§4. The utility of each winner i can be defined as:

$$u_i = p_i - c_i \tag{3}$$

The revenue is related to various customers' information, including their preference of watching videos, behavior histories etc., which is hard to measure. Since each task completion helps the server to collect more efficient customer information, the revenue R is defined as the sum of revenue obtained by completing tasks. Similarly, the total payment P is the accumulation of the payment for each winner. Hence, we have the equations as follows.

$$\begin{cases} R(\Lambda) = \sum\limits_{t_j \in \Lambda} r_{t_j} \\ P(\mathcal{S}_{winner}) = \sum\limits_{i \in \mathcal{S}_{winner}} p_i \end{cases} \tag{4}$$

where Λ is a set of completed tasks, and t_j is one of completed tasks. In addition, p_i is the payment for each host i and r_{t_j} is the revenue that the server obtained by finishing task t_j.

3.5. Payment Minimization Problem

From a practical business perspective, the server needs to obtain a lower bounded revenue such that its basic operating costs can be covered. Meanwhile, corporations want to reduce the incentive cost to the minimum for their benefit. Hence, we give the definition of the payment minimization problem in the following.

Definition 2. *Payment Minimization (PM) problem: Given a set of hosts $\mathcal{U}$, the server selects a subset of hosts $\mathcal{S}_{winner}$ as providers to share their content with local neighbors such that the server's total payment is minimized, subject to a given revenue target of the server.*

It is easy to deduce that the total payment of the server is minimized with $p_i = c_i$. The PM problem can be formalized as an optimization problem in the following.

Objective: Minimize $\sum_{i \in \mathcal{S}_{winner}} c_i$

$$s.t.\ R(\Lambda) \geq R_{th} \tag{5}$$

where R_{th} is the minimal revenue that a server has to obtain.

3.6. Association-based Intelligent Auction Method

Auction is a good way to determine the value of a commodity or service that has an imponderable and dynamic price, which has been applied to many fields [18]. The ordinary auctions are the forward auction which involves multiple buyers and a single seller, where the buyers send bids to compete for the offered commodity or service; the one with the highest bid will win the competition. However, our GoSharing has multiple hosts to share their content and the server has no idea about the exact cost of each user, which inspires use of a procurement auction, namely reverse auction. In other words, multiple hosts tell the server their required reward for sharing their content. After that the buyer selects a group of sellers with the minimum incentive cost.

In this paper, the AI incentive mechanism is common knowledge among hosts, and contains three steps. In the first step, *Candidate Generation*, we present a fast generation candidate algorithm to discover hosts' association based on their stored content and QoS requirements, which makes it possible for hosts to cooperate efficiently. In the second step, *Host Selection*, the server decides which host groups among filtered candidate groups to share their content with request users. Since hosts have selfishness, they have the intention to lie a higher bid price for their content to obtain a higher utility. In the third step, *Payment Determination*, the pricing algorithm is presented to avoid cheating behaviors. The server computes the actual payment for each winner. Next, the server returns to the hosts the auction outcome which contains the matches of hosts and the request users as well as the payment of each host. Finally, hosts share their designated content to request users via the local wireless network. The details of the proposed auction mechanism will be illustrated in §4.1.

The incentive mechanism aims to satisfy the following properties:

- *Computational Efficiency*: the solution can be computed in polynomial time.
- *Individual Rationality* (IR): each sharing host will have a non-negative utility.
- *Incentive compatibility* (IC): also called truthfulness, each host prefers to report his private information truthfully to the server rather than make any potential lie, i.e. the host will get the maximum utility when he bids his cost truthfully.

4. Main Design of GoSharing

In this section, we illustrate the details of the AI incentive mechanism that can be applied for general cooperative systems.

4.1. Association-based Intelligent Incentive Mechanism

The main challenge of the host selection process is that the exhaustive search by checking all possible combinations of hosts makes it impossible for the server to match host groups with tasks effectively. Fortunately, there is a stable association among hosts based on their stored content lists and QoS requirements, which can largely compact the search space. Therefore, before the auction mechanism is presented, it is critical to design smart data filtering method to discover hosts' association and filter candidate groups from host groups.

AI incentive mechanism consists of three parts: candidate generation, host selection and payment determination. First, it exploits hosts' association to filter candidate groups from host groups. Second, with searching among the candidate groups, we present a greedy host selection method with a feasible approximate ratio. Last but not least, we design a corresponding pricing algorithm to make sure AI mechanism has the property of truthfulness.

The AI auction mechanism relies on Myerson's well-known characterization [19], illustrated in theorem 1.

Theorem 1. *Based on the theorem in [20][21], an auction mechanism is truthful if and only if:*

1. *The Host Selection (HS) algorithm is monotone: If host i wins the auction by bidding b_i, it also wins by bidding $b_i' \leq b_i$.*
2. *Given the HS algorithm, there is a unique truthful mechanism associated with this selection algorithm. The pricing algorithm pays each winner the critical value: the highest bid the host could claim and still win under the condition of all other hosts' bids being fixed.*

4.1.1. Candidate Generation

We propose a Fast Candidate Generation (FCG) algorithm, illustrated in Algorithm 1, which can filter candidate groups efficiently in coordinating hosts' association that can be induced by the content stored in hosts' devices and QoS constraints. In terms of collected hosts' information, the server can calculate the link quality of each host who belongs to $SubU_{t_j}$. For simple description, we call the number of hosts in a host group its size, and call a host group of size k a k-host group, noted as H_k. Hosts within a host group are kept in decreasing order by their LQ values for each task. The notation $H_1[1], H_1[2], ..., H_1[\eta_{t_j}]$ is used to represent the 1-host groups for task t_j, where η_{t_j} is the number of hosts able to perform task t_j and $LQ(H_1[1]) \geq LQ(H_1[2]) \geq ... \geq LQ(H_1[\eta_{t_j}])$.

The algorithm contains three steps: in the first *join* step, we join host groups of a particular size k. In general, $k = 1$ and orders H_1 decreasingly by their LQ values. Next, in the *filter* step, QoS function filters H_k with QoS constraints and the filtered groups are namely the candidate groups with size k, G_k. G_k found in the k-th round are used to generate the host groups H_{k+1}. In the *prune* step, we delete all the superset of G_k with size $k+1$ from the original H_{k+1}. The principle of prune rule is that the *cpr* (an important metric defined in the process of host selection) of G_k is always larger than the counterpart of all the superset of G_k, which is proved in Lemma 1. Note that G is the combination of G_{t_j} for all task $t_j \in \mathcal{T}$, which provides the choice candidate groups as the input

---

Algorithm 1 Fast Candidate Generation ($\mathcal{U}$, $\mathcal{B}$, V, R_{th})

Input: Hosts set ($\mathcal{U}$), request task ($\mathcal{T}$), user's association($\mathcal{L}$).

Output: Candidate groups (G).

H_k: Set of k-host groups; G_k: Set of k-candidate groups.
$\mathcal{L} \Leftarrow \{SubU_{t_j} | \left|SubU_{t_j}\right| = \eta_{t_j}\}$
for all task $t_j \in \mathcal{T}$ **do**
    $H_1 = \{$1-host groups $|LQ(H_1[1]) \geq LQ(H_1[2]) \geq ... \geq LQ(H_1[\eta_{t_j}])\}$
    for $(k = 1; H_k \neq \varnothing; k++)$ **do**
        $G_k =$QoS(H_k)
        $H_{k+1} = H_{k+1} \setminus$ superset(G_k)
    $G_{t_j} = \bigcup_k G_k$
$G \leftarrow$ combination of G_{t_j} for all $t_j \in \mathcal{T}$.

---

for the following host selection algorithm.

In the following, a walk-through example of the FCG algorithm is illustrated in Figure 8. Assume that host A, B, C and D store the content that task t_j requested and they have the relations that $LQ(A) \geq LQ(B) \geq LQ(C) \geq LQ(D)$. After the join step, $H_1 = \{(A), (B), (C), (D)\}$. In the next filter step, it is found G_1, the 1-candidate groups that satisfy QoS constraints. Let $G_1 = \varnothing$, then in the prune step $H_2 = H_2 \setminus superset(G_1) = H_2$, since $superset(\varnothing) = \varnothing$. In the next iteration, let $G_2 = \{(AB), (BC)\}$, noted as real line circles in Figure 8, then $H_3 = H_3 \setminus superset(G_2) = \{ACD\}$, where $superset(G_2) = \{(ABC), (ABD), (BCD)\}$. In the final iteration, assume $G_3 = QoS(H_3) = \{(ACD)\}$, namely host group (ACD) satisfies QoS constraints, then $H_4 = H_4 \setminus superset(G_4) = \varnothing$ and the iteration terminals.

Note that there is no necessary to search the 1-host group $(B), (C), (D)$ when it is found 1-host group (A) can not satisfy QoS constraint. Because $LQ(A) \geq LQ(B) \geq LQ(C) \geq LQ(D)$, if host (A) can not satisfy QoS constraints, let alone other 1-host group. The same as the 2-host groups $\{(AD), (CD)\}$, shown as the green dotted circles in Figure 8. Therefore, the search speed can actually

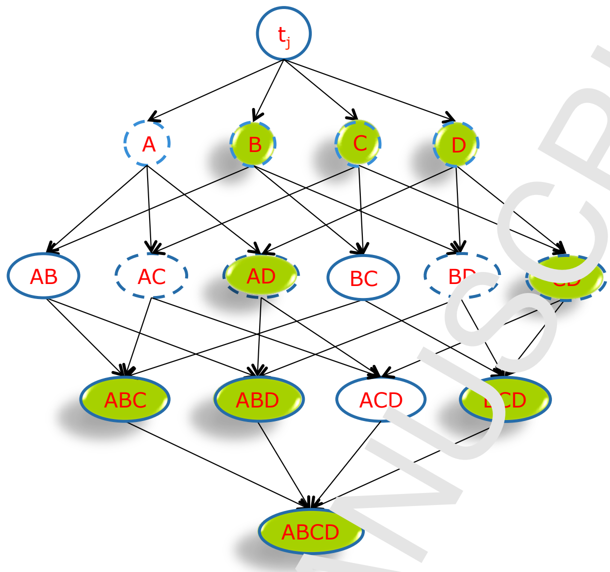


Figure 8: Fast Candidates Generation Algorithm

be further improved.

Lemma 1. *For each task t_j, any given different host groups G_A and G_B satisfying $\rho_{download} \geq \rho_{th}$, if $G_B \in superset(G_A)$, then $cpr(G_A) \leq cpr(G_B)$.*

Proof. Since G_A and G_B satisfy $\rho_{download} \geq \rho_{th}$, the server can obtain the revenue r_{t_j}, no matter G_A or G_B is chosen. Since $G_A \subseteq G_B$, $\sum\limits_{i \in G_A} c_i \leq \sum\limits_{i \in G_B} c_i$, $cpr(G_A) \leq cpr(G_B)$. □

4.1.2. Host Selection

The objective is to design an incentive mechanism that selects hosts to minimize the server's payment under the condition that the server can earn the targeted revenue, i.e. the targeted sharing tasks. In § 3.5, the Payment Minimization (PM) problem is formalized as an optimization problem, which can be reduced to a Weighted Multiple Set Cover (WMSC) problem, proved to be NP-hard in [22]. The reduction process is similar to our previous work [23]. Therefore, we put forward Theorem 2 below.

Theorem 2. *The PM problem is an NP hard problem.*

Unfortunately, the PM problem fails to be solved by exploiting the well-known Vickrey-Clarke-Groves (VCG) mechanism that ensures each host reveals its cost truthfully. The reason is that VCG requires the selected set of users with the lowest cost all the time. However, when the scale of the problem is increased, it is hard to find a solution in polynomial time regarding the PM problem is NP-hard. Moreover, [20] also proves that a non-optimal user selection algorithm with the VCG mechanism could not guarantee truthfulness. Hence, an alternative non-VCG auction mechanism is desired to ensure the truthfulness of hosts while minimizing the payment subject to a server's revenue target.

To solve the PM problem, we propose a host-selection greedy algorithm summarized in Algorithm 2. The basic idea is to select the most cost-efficient host group which has the smallest total bid but makes the server obtain the most revenue, by iterating the selection until the given revenue target has been reached. To this end, we combine these two criteria into the single metric as follows:

$$\frac{\sum_{i \in \mathcal{S}_k} b_i}{\sum\limits_{j \in \Lambda_k} r_{t_j}}. \tag{6}$$

The metric represents the "cost per revenue" (cpr), where Λ_k means the task(s) that can be completed by selecting the host group $\mathcal{S}_k$. The total bid of $\mathcal{S}_k$ is $\sum\limits_{i \in \mathcal{S}_k} b_i$, where b_i is host i's bid. It is assumed that selected hosts will not accept unallocated download requests. Thus, we maintain the set $\mathcal{S}_{curr}$ of the current selected hosts and the set $\mathcal{T}_{uncom}$ for the remaining unallocated download tasks. The host set $\mathcal{S}_k$ is the candidate group with the minimum marginal *cpr* in the k-th iteration, defined as

$$cpr(\mathcal{S}_k) = \frac{\sum\limits_{i \in \mathcal{S}_k \setminus \mathcal{S}_{curr}} b_i}{\sum\limits_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j}}. \tag{7}$$

In each *while-loop*, the server selects the host set $\mathcal{S}_k$ with the minimum marginal *cpr* from G in the k-th iteration.

Algorithm 2 Host Selection ($\mathcal{U}$, $\mathcal{B}$, V, R_{th})

Input: Candidate groups (G), hosts' bids ($\mathcal{B}$), request task ($\mathcal{T}$), revenue from task completion ($\mathcal{R}$) and server's revenue target (R_{th}).

Output: Host winners ($\mathcal{S}_{winner}$) and social cost (C).

Initialization: $T_{uncom} = \mathcal{T}$, $\mathcal{S}_{curr} = \varnothing$, iteration round $k = 0$ and revenue $r = 0$
while $r < R_{th}$ **do**
  Select the set $\mathcal{S}_k = \arg\min cpr(g)$, where $g \in G$.
  $\mathcal{S}_{curr} = \mathcal{S}_{curr} \cup \mathcal{S}_k$, $G = G \setminus \mathcal{S}_k$
  $r = r + \sum_{t_j \in \Lambda_k \cap T_{uncom}} r_{t_j}$
  $T_{uncom} = T_{uncom} \setminus \Lambda_k$
  $k = k + 1$
$\mathcal{S}_{winner} = \mathcal{S}_{curr}$
$C = \sum_{i \in \mathcal{S}_{winner}} b_i$

4.1.3. Payment Determination

After host winners are selected, combined with the HS algorithm, we develop the PD algorithm summarized in Algorithm 3 to encourage hosts to bid honestly, which follows Theorem 2.

In Algorithm 3, the outside *for-loop* (Lines $2-11$) is to compute the critical bid for each winner $i \in \mathcal{S}$. Each *while-loop* aims to calculate host i's maximum bid that can still be selected in this iteration. Given the current selected hosts $\mathcal{S}_{curr}$ and remaining download tasks T_{uncom}, we first select the set $\mathcal{S}_k$ and $S_{k\setminus\{i\}}$ with the minimum *cpr* from the group set G and $G_{\setminus\{i\}}$, respectively (Lines 4 and 6), where $G_{\setminus\{i\}}$ the set of candidate groups that do not contain i. The maximum bid in each iteration is the sum of host i's bid and the *cpr* difference between $\mathcal{S}_k$ and $S_{k\setminus\{i\}}$. In the end, the maximum of these bids among the *while* loops is set to be critical bid p_i, which can promise that host i will be selected in at least one iteration.

Algorithm 3 Payment Determination

Input: Host winners ($\mathcal{S}$), candidate groups (G) and hosts' bids ($\mathcal{B}$)

Output: Critical payments ($\mathcal{P}$)

$p_i = 0$ for all hosts $i \in \mathcal{U}$, $T_{uncom} = \mathcal{T}$, $\mathcal{S}_{curr} = \emptyset$ and $r = 0$
for all host $i \in \mathcal{S}_{winner}$ **do**
    while $r < R_{th}$ **do**
        Select the set $\mathcal{S}_k = \arg\min cpr(g)$, where $g \in G$
        $G_{\setminus\{i\}} = \{g' \in G | i \notin g'\}$
        Select the set $S_{k\setminus\{i\}} = \arg\min cpr(g_{\setminus\{i\}})$, where $g_{\setminus\{i\}} \in G_{\setminus\{i\}}$.
        $\mathcal{S}_{curr} = \mathcal{S}_{curr} \cup S_{k\setminus\{i\}}$
        $r = r + \sum_{t_j \in \Lambda_{k\setminus\{i\}} \cap T_{uncom}} r_{t_j}$
        $T_{uncom} \setminus T_{k\setminus\{i\}}$
        $p_i = \max\{cpr(\mathcal{S}_{k\setminus\{i\}}) \times \sum_{t_j \in \Lambda_k \cap T_{uncom}} r_{t_j} - B(\mathcal{S}_k \setminus \mathcal{S}_{curr}) + b_i, p_i\}$
        k=k+1
    $\mathcal{P}.add(p_i)$
Return $\mathcal{P}$

4.2. Properties of AI Incentive Mechanism

How good is the AI auction mechanism? In the following we will analyze the above mechanism according to the four desirable properties as performance metrics.

4.2.1. Individual Rationality

In Algorithm 2, Line 4 aims to find the subset $\mathcal{S}_k$ including host i with the minimum *cpr*, while Line 6 tries to find a subset $\mathcal{S}_{k\setminus\{i\}}$ with the minimum *cpr* exclusive of host i. If and only if $cpr(\mathcal{S}_k) \leq cpr(\mathcal{S}_{k\setminus\{i\}})$ is true, host i will be selected in the host selection period. Thus, we can have $\frac{B(\mathcal{S}_k \setminus \mathcal{S}_{curr})}{\sum\limits_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j}} \leq cpr(\mathcal{S}_{k\setminus\{i\}})$. Based on this inequality, we therefore have the payment p_i of host i:

$$p_i = \max\{cpr(\mathcal{S}_{k\setminus\{i\}}) \times \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j} - B(\mathcal{S}_k \setminus \mathcal{S}_{curr}) + b_i, p_i\} \geq b_i \tag{8}$$

Hence, GoSharing can guarantee that all hosts' utility is non-negative.

4.2.2. Truthfulness

As long as the conditions listed in Theorem 2 are satisfied, it can promise that GoSharing can make truth-telling a weakly dominant strategy for each host, such that each host reports its bid honestly. For the first condition, the monotonicity of the HS algorithm is easy to prove since host i bidding a smaller value could increase the *cpr* value of the subset with host i. Thus, host i must win in the current or an earlier iteration.

For the second condition, we should prove that p_i is the critical value for host i, i.e. bidding higher p_i could prevent host i from winning the auction otherwise host i must become a winner. Suppose that host i is selected in the k-th iteration. On the one hand, if $b_i > p_i$, i cannot be selected in this iteration, because there exists another subset without i having smaller *cpr* value or $r \geq R_{th}$, i.e. the loop meets the termination condition. On the other hand, if $b_i < p_i$, host i must be selected in the k-th iteration, because *cpr* value of the subset with host i is lower than that with critical value p_i:

$$cpr(\mathcal{S}_k) = \frac{b_i + B(\mathcal{S}_k \setminus (\mathcal{S}_{curr} \cup \{i\}))}{\sum\limits_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j}} < \frac{p_i + B(\mathcal{S}_k \setminus (\mathcal{S}_{curr} \cup \{i\}))}{\sum\limits_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j}} \tag{9}$$

4.2.3. Computational Efficiency

First, we analyze the complexity of Algorithm 1. Set $\eta = \max |SubU_{t_j}|, t_j \in \mathcal{T}$, and we can get $k \leq \eta$. The time complexity of sorting $SubU_{t_j}$ in descending order of LQ values is $O(\eta \log \eta)$. The *for-loop* is at most η, since $k \leq \eta$. Therefore, by searching candidate groups for M tasks, the TCG algorithm runs in $O(M\eta \log \eta)$.

Next, given the candidate groups G, the time complexity of finding the group with minimum *cpr* in each iteration is $|G|$. Since there are M download tasks and each *while-loop* will contribute at least one download task, the number of *while-loop* is at most M. Hence, the HS algorithm runs in $O(M|G|)$ time.

After the set of host winners $\mathcal{S}_{winner}$ is selected, we compute the running time of the PD algorithm. In each round of finding the minimum *cpr* group (Lines 4 and 6), the process similar to Line 6 of Algorithm 1 is realized. Thus, the time complexity of finding G with minimum *cpr* is $O(|G|)$. Moreover, the number of *while-loop* is at most M since each *while-loop* will complete at least one task. Therefore, the PD algorithm takes $O(|\mathcal{S}_{winner}| \cdot |G| \cdot M)$, which dominates the whole auction. It is obtained that the running time of the GoSharing auction mechanism is bounded by $O(|\mathcal{S}_{winner}||G|M)$.

Realistic scenario: Generally speaking, the capacity of a bus is set as 100. It is assumed that 50% of commuters are GoSharing users, and the number of simultaneous launched hosts is less than 25% of total users. Thus, $|\mathcal{S}_{winner}| < 50$ and $M \leq 12.5$. Since at most 10% hosts have stored the same common content, $p \leq 5$. Figure 5 shows when a device has more than four connections, download time will rise to quite long. Thus, the time complexity is $O(|\mathcal{S}_{winner}||G|M) < 50 \cdot 2^5 \cdot 12.5 = 2 \times 10^4$, which is feasible in our real scenario.

Specifically, when the bus is moving between two cell towers with request calling, the soft hand-off technology is applied. That is a cell phone simultaneously connected to two or more cells during a request, such that server switch will not influence the normal operation of our GoSharing for content sharing.

4.2.4. Approximate Ratio Analysis

It is supposed to analyze the approximation ratio achieved by the proposed algorithm 1.

Theorem 3. *The HS algorithm can obtain the approximate solution with a factor of F(d), where* $F(d) = \sum_{p=1}^{d} (\frac{1}{\sum_{j=p}^{d} r_{t_j}} \times r_{t_p})$, $d = \max|\Lambda_k|$ *denotes the maximum size of completed tasks when any candidate group* $\mathcal{S}_k$ *is selected.*

Proof. We assume that $\mathcal{S}$ is the selected hosts by HS algorithm then

$$\sum_{\mathcal{S}_k \in \mathcal{S}} \sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i = \sum_{t_j \in \Lambda} cpr_{t_j} \times v_{t_j} \tag{10}$$

where cpr_{t_j} is the cpr value when t_j is completed, and Λ is a set of completed tasks.

The key of the analysis is to find out the upper bound of $\sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i$ with the corresponding obtained value $\sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \times r_{t_j}$, when candidate group $\mathcal{S}_k$ is selected.

Thus, we need to give an upper bound on the ratio

$$\frac{\sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(S_k) \times r_{t_j}}{\sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i}.$$

To simplify the notation, we assume that the set of tasks Λ_k can be completed when the candidate group S_k is selected, that is $\Lambda_k = \{t_1, \ldots, t_d\}$, where $d = |\Lambda_k|$. Furthermore, it is assumed that these tasks are labeled in the order of cpr_{t_j} computed by GoSharing, i.e. $\{cpr_{t_1} \leq cpr_{t_2} \leq \ldots \leq cpr_{t_d}\}$. In the p-th iteration, t_p will be labeled completed, and where $p \leq d$. Before t_p is labeled as completed, there are at least $t_p, t_{p+1}, \ldots, t_d$ tasks that are uncompleted, noting $\{t_p, t_{p+1}, \ldots, t_d\} \subseteq T_{uncom}$, i.e.

$$\sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} r_{t_j} \geq \sum_{j=p}^{d} r_{t_j}.$$

So we can have:

$$\frac{\sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i}{\sum_{t_j \in \Lambda_k \cap T_{uncom}} r_{t_j}} \leq \frac{\sum_{i \in S_k \cap \mathcal{S}_{curr}} b_i}{\sum_{j=p}^{d} r_{t_j}}.$$

In this iteration, HS selects a candidate group $\mathcal{S}_p$ with the minimum cpr, and so we have

$$cpr_{t_p} = \frac{\sum\limits_{i \in S_p \cap S_{curr}} b_i}{\sum\limits_{t_j \in \Lambda_p \cap T_{uncom}} r_{t_j}} \leq \frac{\sum\limits_{i \in S_k \cap S_{curr}} b_i}{\sum\limits_{t_j \in \Lambda_k \cap T_{uncom}} r_{t_j}} \leq \frac{\sum\limits_{i \in S_k \cap S_{curr}} b_i}{\sum\limits_{j=p}^{d} r_{t_j}}.$$

Here we add up these inequalities for all tasks $\in \Lambda_k$.

$$\begin{aligned} \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \times r_{t_j} = \sum_{p=1}^{d} cpr_{t_p} \times r_{t_p} \leq \\ \sum_{p=1}^{d} \frac{\sum\limits_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i}{\sum\limits_{j=p}^{d} v_{t_j}} \times r_{t_p} = \sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i \cdot \sum_{p=1}^{d} \left(\frac{1}{\sum\limits_{j=p}^{d} r_{t_j}} \times r_{t_p}\right). \end{aligned} \tag{11}$$

With the replacement of $F(d) = \sum\limits_{p=1}^{d} \left(\frac{1}{\sum\limits_{j=p}^{d} r_{t_j}} \times r_{t_p}\right)$, we can obtain the

$$\sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i \geq \frac{1}{F(d)} \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \cdot r_{t_j}.$$

Let $\mathcal{S}^*$ denote the optimum selected winners, so that

$$\begin{aligned} \sum_{i \in \mathcal{S}^*} b_i &= \sum_{\mathcal{S}_k \in \mathcal{S}^*} \sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i \\ &\geq \sum_{\mathcal{S}_k \in \mathcal{S}^*} \frac{1}{F(d)} \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \cdot r_{t_j} \\ &= \frac{1}{F(d)} \sum_{\mathcal{S}_k \in \mathcal{S}^*} \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \cdot r_{t_j}. \end{aligned} \tag{12}$$

Because in every iteration, HS always selects the candidate group with the minimum cpr, we have

$$\sum_{S_k \in \mathcal{S}^*} \sum_{t_j \in \Lambda_k \cap T_{uncom}} cpr(S_k) \cdot v_{t_j} \geq \sum_{t_j \in \Lambda} cpr_{t_j} \cdot v_{t_j}.$$

Finally combined with equations (10) and (12), we get the desired bound,

$$\begin{aligned} \sum_{i \in \mathcal{S}^*} b_i &\geq \frac{1}{F(d)} \sum_{\mathcal{S}_k \in \mathcal{S}^*} \sum_{t_j \in \Lambda_k \cap \mathcal{T}_{uncom}} cpr(\mathcal{S}_k) \cdot v_{t_j} \\ &\geq \frac{1}{F(d)} \sum_{t_j \in \Lambda} cpr_{t_j} \cdot v_{t_j} = \frac{1}{F(d)} \sum_{\mathcal{S}_k \in \mathcal{S}} \sum_{i \in \mathcal{S}_k \cap \mathcal{S}_{curr}} b_i \\ &= \frac{1}{F(d)} \sum_{i \in \mathcal{S}} b_i. \end{aligned} \tag{13}$$

□

5. Performance Evaluation

To evaluate the performance of the AI incentive mechanism, we exploit the following metrics through the simulation experiments.

1. Social cost (C): The total cost of selected hosts. In the host selection period, we aim to choose the hosts to minimize a server's total payment, subject to the given server's revenue target. Note that social cost is the minimum payment by a non-truthful mechanism [12].

2. Approximation ratio (R): This is the main metric demonstrating the performance of the HS algorithm. It illustrates how the HS greedy algorithm approaches the optimal solution (denoted by OPT). $R = \frac{C_{\mathcal{M}}}{OPT}$ where $C_{\mathcal{M}}$ is the obtained social cost by using mechanism $\mathcal{M}$.

3. Overpayment ratio: It is computed as $\gamma = \frac{P-C}{C}$, where P denotes the total payment by applying our truthful mechanism. Hence, the overpayment ratio characterizes the cost that the server overpays to guarantee truthfulness.

4. Utility of all hosts: We record the utility of all hosts to show the property of Individual Rationality (defined in §3.6).

5. Execution time: The total time of auction execution is the time cost to find hosts plus the time cost of determining the payment to each host winner.

5.1. Simulation Setup

It is assumed that 50% of users on the bus or subway have launched the Go-Sharing application. Let δ denote the average fraction of hosts who can share the same media content in each auction period. Since only a small portion of hosts have the same media content, δ is expected to be relatively small and set to be $\delta = 0.2$ in the following simulations. All simulations ran on a PC with 2.9GHZ CPU and 4GB memory. Each simulation is repeated 100 times, and the average values are reported as statistical results.

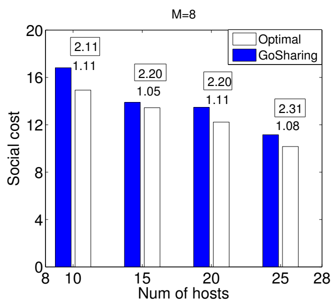
5.2. Case Study:

To evaluate the performance of GoSharing in the bus case, the revenue of each task completion (r_{t_j}) and the sharing cost of each host (c_i) are uniformly distributed over $[5, 10]$ and $[1, 5]$, respectively. If the capacity of the bus is 100, the maximum number of hosts is 50. When the bus is in normal status, N follows the uniform distribution over [20,30]. When the number of passengers is over 50, it is considered as a crowded state. Therefore, we set N distributed among [40, 50] uniformly when the bus is crowded. LOs of hosts to request users are followed by Poisson distribution with $\lambda = 2$.

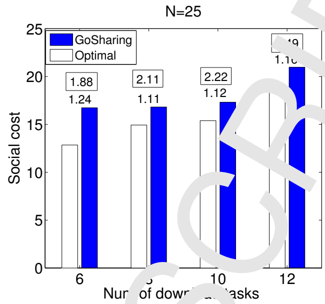
5.2.1. Evaluation of Approximation Ratio

We first evaluate the performance of the HS algorithm of AI incentive mechanism. Since the HS problem is NP-hard, it is time consuming to obtain the optimal solution with the general approach, i.e. brute force search. Hence, the approximate ratio of GoSharing is only evaluated in settings with a small scale, i.e. the bus is in normal status. Specifically, the total number of hosts N is less than 25, while the number of tasks M increases from 6 to 12 with a step of 2. Moreover, we set $\delta = 0.2$ to define the average fraction of hosts who can involve the same media sharing and the target revenue R_{th} is set as the total revenue of all task completion minus 5.

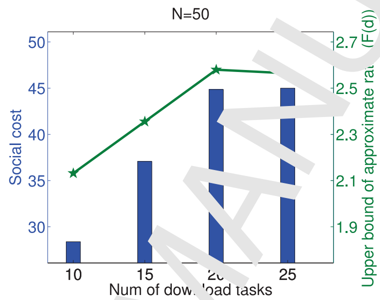
Figure 9(a) shows the approximate ratios of AI mechanism in various settings. The numbers located over bars inside black boxes mean its upper bound, calculated by $H(d)$ function, while the numbers without black boxes represent the practical approximate ratio. It is clear that the social costs of the GoSharing method are very close to its corresponding optimal solutions. With the expanded scale of hosts, the social cost has a declining trend. The reason is that the augment of hosts resource can make the server have better choices. With the augment of M, the social cost increases dramatically, shown in 9(b). This is because the server needs to recruit more hosts to share more media files. From Figure 9(b), it is also observed that the upper bound of AI mechanism approximate ratio increases along with the expanded size of download tasks.



(a) Number of hosts



(b) Number of tasks

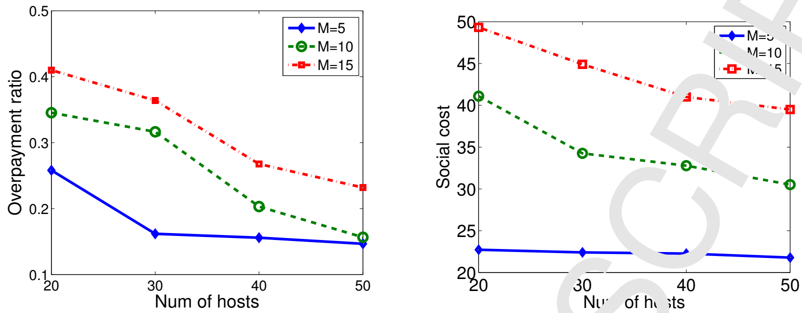


(c) Approximate ratio when the bus is crowded

Figure 9: Approximate ratio under various conditions. (a) and (b) The impact of N and M on approximate ratio; (c) Approximate ratio when the bus is crowded .

This is due to the fact that the candidate group can complete more tasks, d is therefore increased.

When the bus is crowded, the upper bound of approximate ratio for AI mechanism is calculated by the function $f(d)$, as plotted in Figure 9(c). The social cost tends to keep stable when the number of download tasks is over 20, the same with the upper bound of approximate ratio of AI mechanism. The reason is that there are not enough hosts to complete the given tasks when the number reaches 20.



(a) Impact on overpayment ratio

(b) Impact on social cost

Figure 10: The impact of N and M on overpayment ratio and social cost

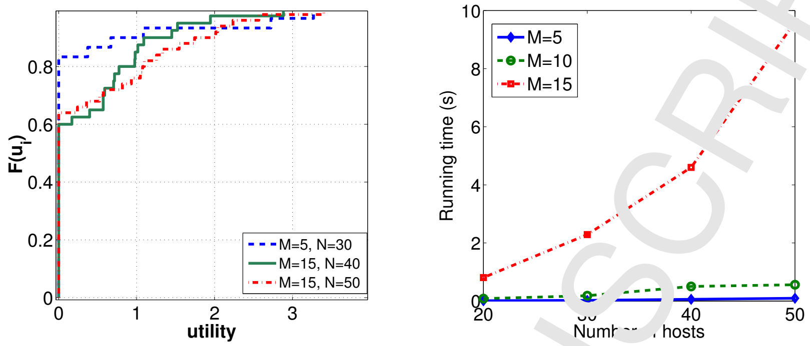
5.2.2. *Evaluation of Overpayment Ratio*

We investigate the impact of the number of hosts (N) on the overpayment ratio. N is varied from 20 to 50 with the increment of 10, and M from 5 to 15 with the step of 5. As shown in Figure 10(a), the overpayment ratio of the AI auction keeps below 0.5 under different M and N, indicating the AI auction with low overpayment cost for the truthful property. With the increase of N, the overpayment ratio is descending. The reason is that the difference of the cost of the candidate groups with the minimum cost and second minimum cost is suppressed with the expanding number of candidate groups. In addition, with the increase of M, the overpayment ratio rises accordingly. That is because the number of host winners increases for sharing more media files.

Figure 10(b) shows that the social cost decreases with the rising number of hosts but increases along with the increasing number of tasks. Also, the social cost is not significantly impacted by the host numbers when M is in small scale.

5.2.3. *Evaluation of Individual Rationality*

In order to show all users have non-negative utility, we depict the empirical CDF (Cumulative Distribution Function) of the utility for all hosts under various settings. From Figure 11(a), it is observed that the proportion of hosts with negative utility is zero. The utility with zero is corresponding to the proportion



(a) Empirical CDF of utilities

(b) Evaluation of running time

Figure 11: Individual rationality and computational efficiency of GoSharing mechanism

of unselected hosts in y-axis in Figure 11(a). All hosts have non-negative utility, and the AI auction mechanism achieves the property of individual rationality (see §3.6).

5.2.4. *Evaluation of Computational Efficiency*

Figure 11(b) demonstrates the computational efficiency of the AI mechanism with different settings, and shows the execution time of all cases is under 10 seconds. The study in [24] shows that users will keep their patience when the response time in man-computer conversational transactions is less than 10 seconds. Therefore, the AI auction mechanism has high computational efficiency in the bus scenario.

6. Related Work

The contribution of our work lies in the intersection of two important cutting-edge research topics. (1) Cooperative mobile opportunistic systems; (2) Incentive mechanisms. Combining the above cases, a fundamentally new incentive mechanism is proposed to solve the cooperative allocation of multiple tasks in this paper.

6.1. Cooperative Mobile Opportunistic Systems

Mobile users usually have temporal and spatial correlations, which can be exploited for task allocation to improve communication quality. Taking the geographical proximity into account, [12] presents a collaborative sensing system for mobile crowdsourcing. Based on the virtual opportunistic community associated with an event, [25] presents several event detection methods toward real-time and cooperative mobile visual sensing and sharing. In order to handle the contradiction between dynamic user traffic and fixed data plans, [26] builds a collaborative sharing system of data plans to make users help neighbors for data download. Authors of [6] consider a scenario in which a group of smartphone users in proximity are interested in the same video and propose a MicroCast system to use the resource on groups of smartphones in a cooperative way for a better streaming experience. Under the assumption of packets being spatial-temporal correlated, [27] presents a cooperative sensing and data forwarding framework to tradeoff delivery delay and transmission overhead. Although the above applications make use of the spatial information for data offloading or media sharing, they are not suitable for the scenario of transient get-together, such as urban transport for its special requirements. While some works have shared similar scenarios as this paper [5, 1], none of them consider the cooperative approach to improve the download quality of media content.

Furthermore, there are many cooperation strategies among mobile devices for content dissemination or resource sharing in delay tolerant and opportunistic networks, based on social ties [28, 7]. However, they use the single-host delivery model, which cannot solve the download problem of poor quality. More importantly, we exploit a multi-host model, as opposed to the single-host model, to improve the reliability of the GoSharing system.

6.2. Incentive Mechanisms

[9] presents incentive mechanisms for both platform-centric and user-centric models. However, on the one hand, in its platform-centric model, it assumes that users and the platform have knowledge of users' costs, which is neither practical

in most mobile sensing systems nor feasible for the cooperative wireless system. On the other hand, in its user-centric model, it designs an auction mechanism for tasks without taking users' cooperation into consideration. Authors of [29, 30] design feasible recruitment models for piggyback crowdsensing under the constraints of coverage quality. Introducing a novel metric, users' quality of information (QoI) into mobile crowdsensing systems, both the single-minded and multi-minded combinatorial auction models are proposed to incentivize user participation [31]. Some research pays attention to the incentive mechanisms based on social networks or social cloud systems [8, 32], which fails to be applied directly for our cooperative content sharing system. The authors of [10] consider the cooperative task individually, thus it can not be directly extended to the cooperative system with multiple correlated tasks.

In addition, [33] and [34] study the online incentive mechanisms for multiple opportunistic users and the real-time requirement, which can not handle the uncertainty of public transport environments. [35] presents a bargaining game theoretic method for virtual resource allocation in cellular networks, which ignores the mobile edge networks.

To the best of our knowledge, this is the first paper to undertake comprehensive research on the truthful incentive mechanism for cooperative systems to share content in mobile edge networks. In this paper, we propose a novel GoSharing framework which uses the stored resources on mobile devices within proximity to share popular content cooperatively. Furthermore, a corresponding AI auction mechanism is proposed for motivating media hosts to share their resources based on QoS requirements, while minimizing the payment of the server as well as keeping users giving their truthful bids.

7. Conclusion

The edge storage of mobile devices and costly charge of cellular network leads to the necessity of content exchange among neighboring commuters. Moreover, the short-range wireless network interface provides the technical support. In this paper, we propose GoSharing to encourage a group of hosts within proximity to share content cooperatively. Our GoSharing is objective to find the effective solution which can minimize incentive cost, subject to the target revenues.

To this end, we first develop a network QoS model based on real measurements to solve the tradeoff between download time and download ratio. To handle the tradeoff and exploit users' association, a smart data filter method, namely a Fast Candidate Generation algorithm is presented. After the candidate groups filtered, a new Host Selection algorithm, which is to find a set of candidate groups with minimum social cost to share content. Furthermore, a novel Payment Determination algorithm is developed to guarantee the truthfulness of each host. Eventually, both theoretical analysis and extensive simulations demonstrate that the GoSharing incentive framework achieves not only truthfulness, individual rationality, high computational efficiency in real scenarios and low overpayment ratio, but also high download delivery and acceptable download time.

An interesting further extension of this work is to consider both the strategies of hosts and request users, such that we can obtain a better match for content sharing. The online scenario and the impact of users' mobility will be deeply analyzed for the future.

Acknowledgment

This research was supported in part by the National Natural Science Foundations of China (No. 61701444 and No. U1709219), Key Research and Development Program of Zhejiang Province (No. 2018C01093) and in part by the Science Foundation of Zhejiang Sci-Tech University (No.16032182-Y).

References

[1] N. Zhang, Y. Lee, M. Radhakrishnan, R. K. Balan, Gameon: P2p gaming over public transport, in: Proc. ACM MobiSys, 2015, pp. 105–119.

[2] N. G. Aaron M. Renn, Commuting in new york city, 2000-2010., http://www.newgeography.com/content/002691-commuting-new-york-city-2000-2010/,2012.

[3] N. Bakki, Commuting time of workers in japan, 2011, http://nbakki.hatenablog.com/entry/2014/02/20/124942/,2014.

[4] chinadaily.com.cn., Commuting time in capital averages 97 minutes, 2014., http://www.chinadaily.com.cn/china/2014-11/24/content_18967891.htm/.

[5] L. McNamara, C. Mascolo, L. Capra, Media sharing based on colocation prediction in urban transport, in: Proc. ACM MobiCom, 2008, pp. 58–69.

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, A. Markopoulou, Microcast: cooperative video streaming on smartphones, in: Proc. ACM MobiSys, 2012, pp. 57–70.

[7] X. He, H. Dai, P. Ning, R. Dutta, A multi-player markov stopping game for delay tolerant and opportunistic resource sharing networks, in: Proc. IEEE INFOCOM, 2016.

[8] W. S. Lin, H. V. Zhao, K. Liu, Incentive cooperation strategies for peer-to-peer live multimedia streaming social networks, IEEE Transactions on Multimedia 11 (3) (2009) 396–412.

[9] D. Yang, G. Xue, X. Fang, J. Tang, Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones, IEEE/ACM Transactions on Networking 24 (3) (2016) 1732–1744.

[10] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, J. Walrand, Motivating smartphone collaboration in data acquisition and distributed computing, IEEE Transactions on Mobile Computing 13 (10) (2014) 2320–2333.

[11] L. G. Jaimes, I. Vergara-Laurens, M. A. Labrador, A location-based incentive mechanism for participatory sensing systems with budget constraints, in: Proc. IEEE PerCom, 2012, pp. 103–108.

[12] Z. Feng, Y. Zhu, Q. Zhang, M. L. Ni, A. V. Vasilakos, Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing, in: Proc. IEEE INFOCOM, 2014, pp. 1231–1239.

[13] K. Ota, M. Dong, J. Gui, A. Liu, Quoin: Incentive mechanisms for crowd sensing networks, IEEE Network 32 (2) (2018) 114–119.

[14] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, J. Song, Comon: cooperative ambience monitoring platform with continuity and benefit awareness, in: Proc. ACM MobiSys, 2012, pp. 43–56.

[15] A. Archer, E. Tardos, Frugal path mechanisms, ACM Transactions on Algorithms (TALG) 3 (1) (2007) 3.

[16] X. Liu, K. Ota, A. Liu, Z. Chen, An incentive game based evolutionary model for crowd sensing networks, Peer-to-Peer Networking and Applications (2015) 1–20.

[17] Compare wifi-direct with bluetooth 4.0., http://www.makeuseof.com/tag/the-differences-between-bluetooth-4-0-and-wi-fi-direct-you-need-to-know/.

[18] X. Zhuo, W. Gao, G. Cao, S. Hua, An incentive framework for cellular traffic offloading, IEEE Transactions on Mobile Computing 13 (3) (2014) 541–555.

[19] R. B. Myerson, Optimal auction design, Mathematics of operations research 6 (1) (1981) 58–73.

[20] N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, Algorithmic game theory, Vol. 1, Cambridge University Press Cambridge, 2007.

[21] Y. Singer, Budget feasible mechanisms, in: Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on, IEEE, 2010, pp. 765–774.

[22] J. Yang, J. Y.-T. Leung, A generalization of the weighted set covering problem, Naval Research Logistics (NRL) 52 (2) (2005) 142–149.

[23] S. Luo, Y. Sun, Z. Wen, Y. Ji, C2: Truthful incentive mechanism for multiple cooperative tasks in mobile cloud, in: Proc. IEEE ICC, 2016, pp. 1–6.

[24] Response times: The 3 important limits, http://www.nngroup.com/articles/response-times-3-important-limits/.

[25] H. Chen, B. Guo, Z. Yu, Q. Han, Toward real-time and cooperative mobile visual sensing and sharing, in: Proc. IEEE INFOCOM, 2016, pp. 1–9.

[26] T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu, S. Lu, Indapson: An incentive data plan sharing system based on self-organizing network, in: Proc. IEEE INFOCOM, 2014, pp. 1545–1553.

[27] D. Zhao, H. Ma, S. Tang, X.-Y. Li, Coupon: A cooperative framework for building sensing maps in mobile opportunistic networks, IEEE Transactions on Parallel and Distributed Systems 26 (2) (2015) 392–402.

[28] X. Zhang, G. Cao, Transient community detection and its application to data forwarding in delay tolerant networks, IEEE/ACM Transactions on Networking 25 (5) (2017) 2829–2843.

[29] D. Zhang, H. Xiong, L. Wang, G. Chen, Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint, in: Proc. ACM UbiComp, 2014, pp. 703–714.

[30] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint, in: Proc. IEEE PerCom, 2015, pp. 55–62.

[31] H. Jin, L. Su, D. Chen, K. Nahrstedt, J. Xu, Quality of information aware incentive mechanisms for mobile crowd sensing systems, in: Proc. ACM MobiHoc, 2015, pp. 167–176.

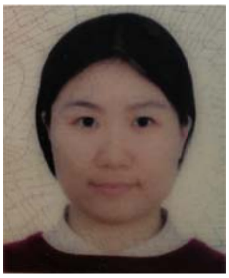
[32] Y. Zhang, M. van der Schaar, Incentive provision and job allocation in social cloud systems, IEEE Journal on Selected Areas in Communications 31 (9) (2013) 607–617.

[33] D. Zhao, H. Ma, L. Liu, Frugal online incentive mechanisms for mobile crowd sensing, IEEE Transactions on Vehicular Technology 66 (12) (2017) 3319–3330.

[34] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, X. Li, Free market of crowdsourcing: Incentive mechanism design for mobile sensing, IEEE transactions on parallel and distributed systems 25 (12) (2014) 3190–3200.

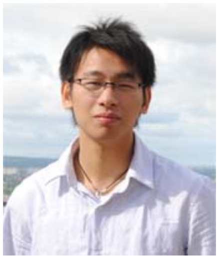
[35] G. Zhang, K. Yang, K. Xu, Y. Dong, A bargaining game theoretic method for virtual resource allocation in LTE-based cellular networks, Science China Information Sciences 58 (12) (2015) 1–9.

{Shuyun Luo}



received his B.S. degree in electronic information engineering and Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications, China, in 2008 and 2016, respectively. From 2012 to 2014, she studied at Pennsylvania State University in computer science and engineering department as a visiting student. She is currently an assistant professor in the College of Computer Science and Technology, Zhejiang Sci-tech University, China. Her research interests include edge computing, sensor networks and network economics.

{Zhenyu Wen}



is currently a postdoc researcher with the School of Computing, the Newcastle University, Edinburgh, UK. He received the M.S and Ph.D. degree in computer science from Newcastle University, Newcastle Upon Tyne, UK in 2011 and 2015 respectively. His current research interests includeMulti-objects optimization, Crowdsources, AI and Cloud computing.

{Xiaomei Zhang}

is an assistant professor of computational science at the University of South Carolina Beaufort. She received the bachelor degree in electrical engineering from University of Science and Technology of China in 2010 and the PhD degree in computer science and engineering from the Pennsylvania State University in

2016. Her research interests mainly include mobile computing, mobile social networks, wireless communications and data science. Her research focuses on design, analysis and implementation of mobile network systems to achieve efficient data access and computation in mobile networks,. Other than this, she also does researches on data mining and data-centric computing based on online social networks.

{Weiqiang Xu}

received his M.Sc. degree in Communications and Information System from Southwest Jiao-Tong University, China, and his Ph.D. degree in Control Science and Engineering from Zhejiang University, China, in 2003 and 2006, respectively. He also was a postdoctor at Zhejiang University and visiting scholar at Columbia University. He is currently a professor with the School of Information Science and Technology, Zhejiang Sci-tech University, Hangzhou, China. He is currently a committee membership of the Youth Committee of China Institute of Communications, and the Committee of Chinese Institute of Electronics on Information Theory. His research interests include multi-cell networks, Ad Hoc networks, wireless sensor networks, etc. He is a senior member of IEEE.

{Albert Y. Zomaya}

a is currently the Chair Professor of High Performance Computing and Networking in the School of Information Technologies, The University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. Professor Zomaya published more than 500 scientific papers and articles and is author, co-author/ editor of more than 20 books. He is currently Editor in Chief of the IEEE Transactions on Sustainable Computing. He served as the Editor in Chief of the IEEE Transactions on Computers (2011-2014). Currently, Professor Zomaya serves as an associate editor for 22 leading journals, such as, the ACM Computing Surveys, IEEE Transactions on Computational Social Systems, IEEE Transactions on Cloud Computing, and Journal of Parallel and Distributed Computing. He delivered more than 150 keynote addresses, invited seminars, and media briefings and has been actively involved, in a variety of capacities, in the organization of more than 600 national and international conferences. Professor Zomaya is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET (UK). Professor Zomayas research interests are in the areas of parallel and distributed computing and complex systems

{Rajiv Ranjan}

is a Reader in Computing Science at Newcastle University, United Kingdom. Before moving to Newcastle University, he was Julius Fellow (2013-2015), Senior Research Scientist and Project Leader in the Digital Productivity and Services Flagship of Commonwealth Scientific and Industrial Research Organization (CSIRO C Australian Governments Premier Research Agency). Prior to that he was a Senior Research Associate (Lecturer level B) in the School of Computer Science and Engineering, University of New South Wales (UNSW). He has a PhD (2009) from the department of Computer Science and Software Engineering, the University of Melbourne.

In this paper, we propose GoSharing, an intelligent incentive framework which motivates resource owners to share their stored videos cooperatively in mobile edge networks. GoSharing is able to achieve the goals of encouraging commuters to share their content cooperatively with the minimum incentive cost based on users' association and guaranteeing the Quality of Service (QoS) of the task sharing.

The highlights of this work are summarized as follows:

1) In order to improve the reliability of content sharing in mobile edge networks, we present a multi-host communication model to allow multiple resource owners to share their content collaboratively.

2) We measure the factors that impact the quality of data delivery from hosts to the request users on public transport.

Based on the experimental results, we formalize a network QoS model to describe the tradeoff between reliability and download time.

3) To motivate hosts to share their content collaboratively, we design an intelligent incentive framework, GoSharing, composed of candidate generation, host selection and payment determination, which has four desirable properties: a) truthfulness, b) individual rationality, c) computational efficiency, d) low overpayment ratio, as well as high download ratio.