Communication, Computation, and Caching Resource Sharing for the Internet of Things

Ming Tang, Lin Gao, and Jianwei Huang

Abstract

The Internet of Things connects a large number of smart mobile devices with the Internet, where these devices are embedded with often limited communication, computation, and caching resources. To address the heterogeneity of these devices and achieve efficient overall system resource utilization, researchers have proposed various device-to-device resource sharing models, enabling mobile devices to form device-to-device connections and to share their resources for cooperative task execution. Most of these existing works, however, considered scenarios where mobile devices can share one or two types of resources, and hence inadequately explore the potential of resource sharing among mobile devices. In this article, we introduce a general framework where mobile devices can share any combination of the three types of resources, and it can generalize many existing device-to-device resource sharing models. In addition, it can achieve more efficient resource allocation by offering mobile devices more flexibility in terms of resource sharing. Based on the proposed framework, we focus on discussing two issues: the optimization issue, regarding how to schedule resources among devices; and the economic issue, regarding how to motivate the device owners to share their resources. We introduce the challenges and potential solutions to these two issues. We further outline several open issues and future directions for the proposed general resource sharing framework.

Introduction

The Internet of Things (IoT) connects a large number of smart mobile devices (e.g., wearable devices, smartphones) with the Internet, where these devices are embedded with often limited communication, computation, and caching (3C) resources. These mobile devices can communicate and interact with each other to perform tasks (e.g., data sensing, analysis) without always requiring human-to-human or human-to-computer interactions. The IoT framework can effectively monitor human behaviors and environmental circumstances, and perform smart decision making [1].

IoT is a large-scale distributed system, and the tasks and devices in the system are always highly heterogeneous [1]. To accomplish various complicated tasks that require effective 3C resource coordination, it is important to exploit the device-to-device (D2D) connections of mobile devices, and to propose D2D resource sharing models that enable resource sharing among devices for cooperative task execution. Note that this is different from the mobile edge computing scenario [2], where mobile devices offload their tasks to edge servers, under which many existing works have studied joint 3C resource optimization (e.g., [3]).

For D2D resource sharing, many existing works proposed 1C resource sharing models. In these models, one type of resource is shared among devices. Tang *et al*. [4] considered the sharing of downloading resources for mobile video streaming services. Xu *et al.* [5] considered the sharing of computation resources through D2D connections, and proposed an incentive mechanism to motivate the sharing. In [6], Feng *et al.* considered content sharing among nearby devices. Some studies focused on 2C resource sharing models, where devices share two types of resources. Chen *et al*. [7] proposed a model that shares communication and computation resources. In [8], Destounis *et al*. proposed a caching and computation resource sharing model for cooperative data processing.

In this work, building upon these 1C and 2C models, we propose a 3C framework for D2D resource sharing, where mobile devices can share any combination of the three types of resources. This framework is an extension of our earlier work [9] in terms of divisible computation module. Figure 1 shows an instance in a smart medical healthcare application scenario: family members monitor their health using wearable medical devices (devices A1, A2, and A3), a smartphone (device B), and a laptop (device C). The health monitoring task contains three main processes: retrieve real-time and historical data, perform data analysis and summarization, as well as store the summarized data (locally and in the cloud). In the 3C framework, these devices communicate with each other through D2D connections. By considering the capability of each device, the monitoring can be achieved as follows: devices A1, A2, and A3 provide real-time measurement data (cached content sharing); device B downloads historical data from the Internet (communication sharing), performs data analysis and summari-

The authors introduce a general framework where mobile devices can share any combination of the three types of resources, and it can generalize many existing device-to-device resource sharing models. In addition, it can achieve more efficient resource allocation by offering mobile devices more flexibility in terms of resource sharing.

Ming Tang is with the University of British Columbia; Lin Gao is with the Harbin Institute of Technology; Jianwei Huang is with the Chinese University of Hong Kong, Shenzhen, the Shenzhen Institute of Artificial Intelligence and Robotics for Society, and the Chinese University of Hong Kong, Hong Kong.

Digital Object Identifier: 10.1109/MCOM.001.1900354

 0163-6804/20/$25.00 © 2020 IEEE

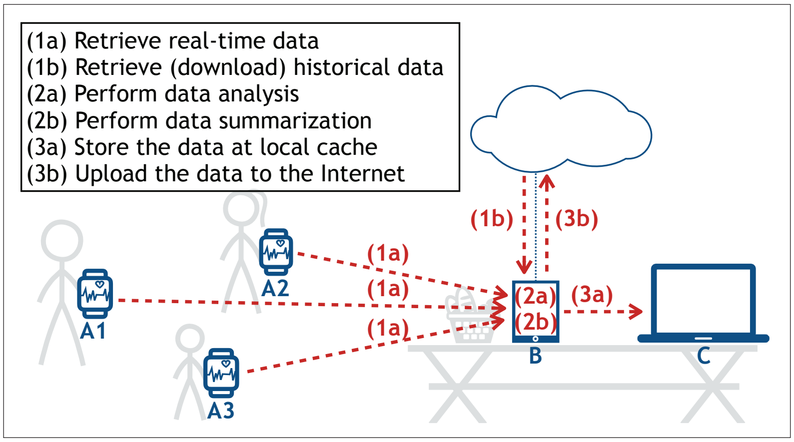

Figure 1. An application instance of the 3C framework: devices share their 3C resources for smart medical healthcare.

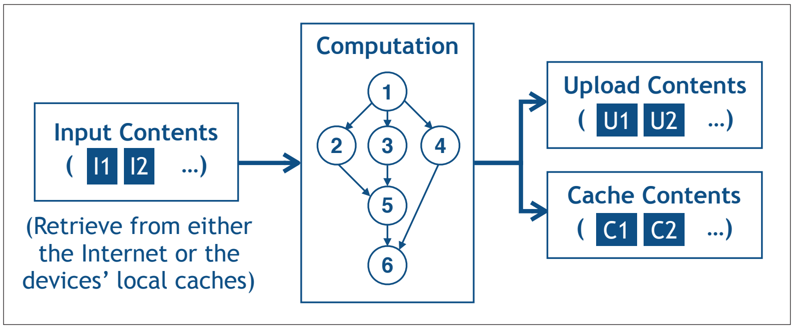

Figure 2. The general resource-centric task model.

zation (computation sharing), and uploads the summarized data to the Internet (communication sharing); device C stores the summarized data at its local cache. We want to emphasize that such a 3C framework is not limited to any particular application. Instead, it enables effective cross-application coordination in terms of the joint 3C resource sharing and scheduling among mobile devices.

There are two key advantages of implementing the proposed 3C framework. First, each task in the framework is modeled in a general resource-centric way, so the tasks belonging to different applications (with heterogeneous resource requirements) can share the resources of multiple devices together. Second, when scheduling resources, the framework can jointly consider all the 3C resources, so it can achieve more efficient resource allocation when compared with the previously mentioned 1C and 2C models.

The understanding of two key issues is required for implementing the 3C framework: the optimization issue, regarding how to schedule the resources of mobile devices efficiently for the cooperative task execution; and the economic issue, regarding how to motivate resource sharing among the devices of different users. Studying these two issues, however, is challenging in the 3C framework due to the coupling of the 3C resources.

The key contributions of this article are listed as follows:

- 3C framework: We propose a framework for 3C resource sharing. It can include many existing 1C and 2C models (e.g., [4–8]) as well as 3C models (e.g., [9]) as special cases.
- Optimization and economic issues: We outline the main challenges of solving the optimization and economic issues in the 3C framework, and propose several online scheduling algorithms and incentive mechanisms dealing with the challenges.
- Future research directions: We identify some future research directions for the 3C framework.

In the rest of this article, we first introduce the proposed 3C framework. Then we discuss the optimization and economic issues. Finally, we outline future research directions.

A General 3C Framework

We first introduce the idea of the 3C framework. Then we show its generalization of the existing 1C or 2C models.

3C Framework

In this framework, we aim to enable mobile devices with tasks requesting any combination of the 3C resources to form cooperative groups through D2D connections to accomplish their tasks. To achieve such joint sharing of the 3C resources, the most crucial part is to design a general resource-centric task model that can accommodate as many types of tasks as possible.

To achieve this, we propose a general task model, which is an extension of the task model in [9]. This task model is illustrated in Fig. 2. Note that by specifying the resources requested, this task model can include the existing task models (e.g., [4–8]) as special cases. Specifically, the task model is centered around a computation module, which needs some input contents and has some output contents. The input content specifications include the required contents but not the content sources, so the contents can be either downloaded or retrieved from the caches of other devices (depending on the available 3C resources). The computation module corresponds to a computational process, which consists of a set of computational subtasks. The dependence among the subtasks is represented by a directed acyclic graph, where each node represents a subtask, and the directed edge from one node to another means that the former subtask has to be executed prior to the later subtask. The output contents can be cached at the owner of the task or uploaded to the Internet. Moreover, any part of this task model can be none or zero; hence, it includes the previously mentioned 1C and 2C models as well as the task model in [9] as special cases.

This task model specifies the contents requested, and hence it is a content-based model. This is different from many existing computation sharing models (e.g., [5]) and 2C models (e.g., [7, 8]), which considered data-based models that specify the volume as well as the source and destination of the requested data. The content-based feature provides flexibility of the content retrieving and forwarding, because the tasks do not need to specify the source and destination of the contents. The flexibility further allows the sharing of downloading resources and cached contents, as the framework can automatically determine where to retrieve contents (either from the Internet or from a device's local cache) through joint optimization.

In the 3C framework, to achieve the joint 3C resource sharing, mobile devices specify their tasks using the general resource-centric task model without specifying the detailed applications (e.g., video streaming, virtual reality). As a result, from the system's point of view, the 3C framework can focus on the 3C resources requested by the tasks and those owned by the devices, and properly schedule the resources to accomplish the tasks. The scheduling in the 3C framework mainly involves the allocation of the input and output contents as well as the computational subtasks of the computation module, and the transmission of the contents among the devices. The scheduling optimization is addressed later.

To show the benefit of applying the proposed framework, Fig. 3 presents an example with five devices and two tasks. Device 1 wants to cache and watch video content H; device 4 wants to transcode video content H to video content L, edit video content L to video content E, cache and watch the edited content, and upload the content to the Internet. Although the tasks of the two devices are quite different, in the 3C framework, both tasks can be modeled using the general task model, as shown in Fig. 3. As a result, the devices need to consider only the resources requested by the tasks, and they can jointly optimize the scheduling of the requested resources (regardless of the tasks requesting them) to avoid repeated content downloading and forwarding. For example, device 3 can be responsible for retrieving content H (from either the Internet or its local cache), which is forwarded to device 1 for caching and transcoding. The transcoded video content L is then forwarded to device 2 for editing. The edited video content is forwarded to device 4 for caching and to device 5 for uploading. Such scheduling is obviously more efficient (e.g., in terms of energy consumption) than scheduling the two tasks respectively.

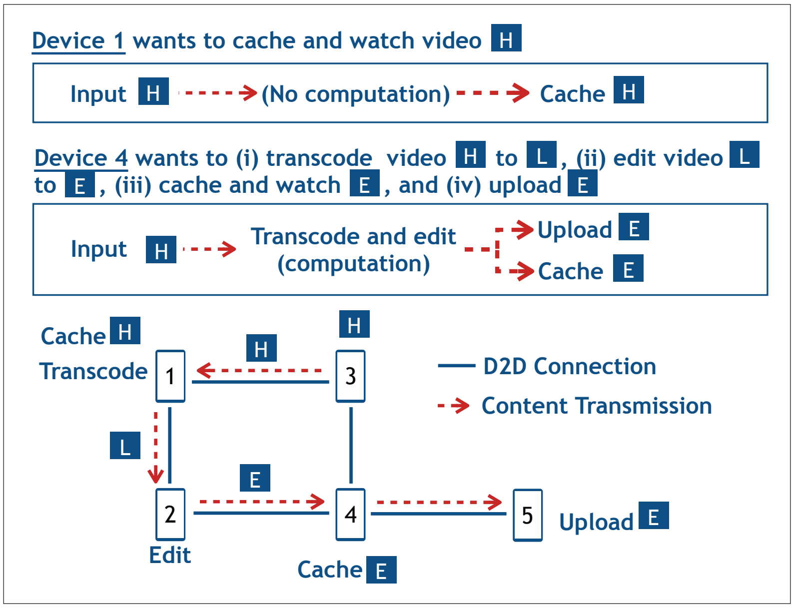

Figure 3. An example of the 3C framework.

Generalization of Existing 1C and 2C Task Models

By specifying the resources requested, the proposed task model can accommodate many 1C and 2C task models.

Communication Resource Sharing: The proposed task model can include the content downloading task model in [4] as a special case by specifying the requested resources as follows: the input contents and cached contents both correspond to the contents to be downloaded; the task owner is the one who wants the contents; there is no computation process and no content to be uploaded. In the case when no device in the system has cached the required contents, the system will schedule devices to download these contents as well as schedule the content transmission among the devices from the content downloaders to the task owner. In addition, the proposed task model can include the content uploading task model as a special case as follows: the input and upload contents both correspond to the contents to be uploaded; there is no computation process and no content involved in the caching process.

Computation Resource Sharing: To model a computation task [5], the general task model can be specified as follows. The input contents are the contents required to be input for computation, the cache contents are the output contents of the computation, and the computation process is specified according to the task requirements. Note that the computation process could contain multiple computational subtasks, as in the task model in [10].

Cached Content (Caching) Sharing: To model the task of retrieving contents from other devices [6], the general task model can be specified as follows. The input contents and cache contents both correspond to the contents to be retrieved; the task owner is the one who wants the contents; there is no computation process and no content for uploading. In the case when some devices have the requested contents, the system can optimize the devices that share the contents, and optimize the content transmission from the devices sharing the contents to the task owner.

Communication and Computation Sharing: If no device has the input contents requested by the tasks in the system, the proposed task model (Fig. 2) degrades to a communication and computation sharing model. However, although this degradation process is straightforward, it is hard to upgrade a communication and computation sharing model to a 3C sharing model. This is because in most works on communication and computation sharing (e.g., [7]), their task models are data-based, specifying only the amount of computation and communication resources that each task requests. These 2C models cannot effectively address the cached content sharing problem without specifying the sources and destinations.

Caching and Computation Sharing: If there is no content to be downloaded or uploaded, the framework degrades to a caching and computation sharing model. However, for the existing caching and computation sharing models (e.g., [8]), it is difficult to directly transform them into a 3C model, because these existing studies focused on data-based models. When further considering the communication resource dimension, a data-based model does not allow us to easily decide whether to retrieve contents using cached content sharing (from devices' caches) or downloading sharing.

Optimization in the 3C Framework

In this section, we discuss the optimization and economic issues under the 3C framework. For each issue, we introduce its problems, challenges, and candidate solutions.

Optimization Issue

The optimization issue aims to understand how to schedule the resources of mobile devices efficiently for cooperative task execution. Specifically, which devices should be responsible for downloading, retrieving (at local caches), and uploading contents, as well as performing the subtasks of the computation module? How should the contents be transmitted among devices through D2D connections? This issue contains both offline and online optimization.

Offline Optimization: Offline optimization aims to optimize the resource scheduling in a centralized manner under complete system information, assuming that all the task requirements and device capacities are known. Such offline optimization can characterize the performance (e.g., energy consumption) of the 3C framework and serve as a benchmark for online optimization.

The offline optimization aims to optimize the resource scheduling in a centralized manner under complete system information, assuming that all the task requirements and device capacities are known. Such an offline optimization can characterize the performance (e.g., energy consumption) of the 3C framework and serve as a benchmark for the online optimization.

Offline optimization involves two sets of binary decision variables. The first set corresponds to the allocation decisions, indicating whether a process (e.g., the retrieving, downloading, or uploading of a content, the execution of a computational subtask) of a task is performed by a device or not. The second set corresponds to the multihop transmission decisions, indicating whether an input or output content of a task (or an intermediate content between dependent computational subtasks) is forwarded from one device to another or not. Due to the 3C resource coupling and the involved binary decision variables, offline optimization always leads to a nonconvex integer programming problem. To address this, auxiliary variables may be needed to transform the problem into a linear integer programming problem (if possible), and linear relaxation may be applied to reduce the computation complexity.

As an example, based on the energy minimization problem in [9], we compare the performance of the proposed 3C framework with that of the 1C/2C models. As shown in Fig. 4, we evaluate the normalized energy consumption (with respect to the energy consumption without any cooperation) of the 1C/2C models and the 3C framework under different variance coefficients (for generating the task and user parameters). A larger variance coefficient implies a higher degree of heterogeneity among the devices and the tasks. The figure shows that the proposed framework can significantly reduce the energy consumption when compared to 1C/2C models. Moreover, the energy reduction is more significant under a larger variance coefficient, as the larger heterogeneity provides devices more opportunities for cooperation.

Online Scheduling Algorithm: In practice, however, the number of devices could be quite large, and the system information may vary over time. This makes offline optimization, requiring complete system information, inapplicable. Hence, it is important to propose a distributed online algorithm that allows devices to make their scheduling decisions in a decentralized fashion without knowledge of future system information. Since it can be costly to let mobile devices communicate with devices that are multiple hops away in a decentralized fashion, we focus on task scheduling within one hop in this section.

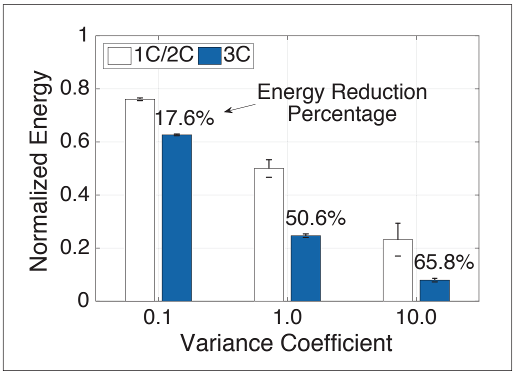

Figure 4. Energy consumption comparison between the 3C framework and the 1C/2C models.

However, it is challenging to design an online algorithm operating in a decentralized fashion. First, the tasks always request a combination of the 3C resources, so the scheduling decisions for these tasks should take into account each of the resources together with the involved D2D resources among mobile devices. Second, from a device's point of view, the scheduling performance (e.g., energy consumption, task delay) depends on the scheduling decisions of not only the device itself but also the other devices, while the decisions of the other devices cannot be known beforehand. Third, the system information (e.g., device connectivity, task requirements of the devices) may change over time and hence be unknown to each device beforehand. In the following, we discuss how to address these challenges using two candidate methods: a Lyapunov-based algorithm and a deep reinforcement learning (DRL)-based algorithm.

A Lyapunov-based scheduling algorithm is one candidate method that can handle the practical scenario where the system information is unknown beforehand. This algorithm can stabilize the queues (of the tasks or subtasks) in the system while optimizing the performance objective (e.g., minimizing energy consumption). This idea of the Lyapunov-based algorithm is inspired by [11] (for a 1C model) with some modifications addressing the 3C resources. In this algorithm, each device is modeled as an entity with 3C resources and four queues, where the queues correspond to downloading, uploading, computation, and D2D transmission, respectively. When a mobile device has a task, the device requests information on the 3C resources and queues of its nearby devices, and determines its task offloading decisions (e.g., downloading/uploading, computation, or cached content retrieving). The offloading decision is determined by minimizing a weighted sum of the

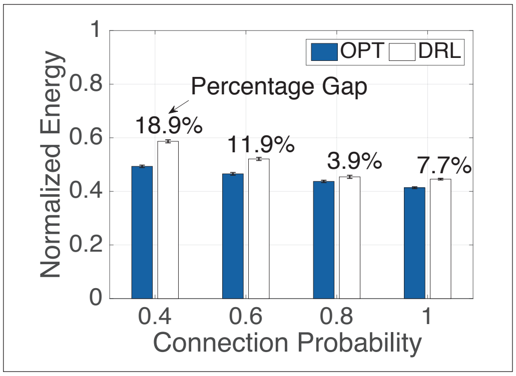

Figure 5. Energy consumption comparison between the optimal solution and the DRL-based algorithm.

energy consumption (if aiming at energy minimization) and the queue length changes of all the neighbor devices. Intuitively, the consideration of the changes of queue length is used to reduce the chance that a queue becomes empty (which implies that the capacity for processing the queue is not fully utilized) or overflows. It can be proved that such a Lyapunov-based scheduling algorithm would converge to the system optimal performance asymptotically, with an approximation error bound that is controllable by the weights chosen in the algorithm.

A model-free DRL-based scheduling algorithm is another candidate method, where the algorithm is based on the DRL technique proposed in [12]. This algorithm can operate without knowing the dynamics of the system (i.e., the way that the environment reacts to the actions of mobile devices), and hence can address the challenges resulting from the coupled 3C resources and the strategic interactions of the mobile devices. Specifically, when a device has a task to process, the device decides its action (i.e., how to offload the input, computation, and output subtasks of the task to nearby devices) based on its locally observed state, such as its task requirements and the capacity information of its nearby devices. Given the observed state, the selected action will induce the device with a reward (e.g., which can be decreasing in energy consumption or task delay). The key idea of the DRL-based algorithm is to learn from the history and obtain a neural network that can provide a mapping from each state to a set of Q-values, where each Q-value reflects the expected long-term reward of the device if the device chooses an action given the state. With such a neural network, the optimal policy of a device is to choose the action with the maximum Q-value under its observed state. Note that the neural network training can be performed offline, so the algorithm does not require large computational resources from the mobile devices.

Figure 5 shows the normalized energy consumption comparison between the proposed DRL-based algorithm and the optimal solution (denoted by OPT) under different device connection probabilities. A larger connection probability means that devices can observe more information through one-hop information exchange. The percentage gap shown in the figure is defined as the energy consumption difference between DRL and OPT normalized by that of OPT. As shown in Fig. 5, when the connection probability is large (e.g., 0.8, 1), the percentage gap is smaller than 10 percent.

Economic Issues

The 3C resources may be shared among the devices owned by different users, but the cost of sharing may discourage it. Hence, it is important to design a mechanism to provide proper incentives for users to join the collaboration and share resources. Designing the incentive mechanism in the 3C framework, however, can be challenging, as it is not always easy to quantify the level of contributions of different devices involved in completing the same task but contributing different resources.

One candidate incentive mechanism is Nash bargaining, which applies to the scenario where users coordinate to achieve a social objective (e.g., social energy consumption minimization). With the help of the Nash bargaining solution, users can reach an agreement that specifies how their devices perform the resource allocation and share the benefits. This mechanism addresses the aforementioned challenges by collecting the system information and solving a Nash bargaining problem in a centralized fashion. The solution can always ensure fairness; that is, the scheduling and payment results are the same for users who contribute the same amount of the same types of resources.

Another candidate mechanism is multi-dimensional auction, inspired by the work in [13] for downloading resource sharing. Such an auction mechanism applies to the scenario where each user has private information (e.g., the cost of executing a task or the gain of having a task being executed), and a properly designed auction mechanism can ensure that users truthfully reveal their private information and achieve efficient resource allocation. Specifically, when a device has a task to process, it can initiate an auction, and its neighbor devices can bid for the opportunities of helping that mobile device. Different from the traditional single-dimensional auction where each bidder bids with a price only, with the multi-dimensional auction, each bidder bids with multi-dimensional bids. The bids include which resources the bidder wants to contribute, how much it wants to contribute, and how much the bidder wants to be compensated. Due to the submission of multi-dimensional bids, such an auction can address the challenge resulting from the coupled 3C resources.

Future Challenges and Open Issues

In addition to the optimization and economic issues, we further outline several future challenges and open issues.

Software and protocols should be carefully designed to enable practical implementation of the 3C framework. Specifically, the key idea of the proposed 3C framework is to generalize various tasks into a unified structure, and pool the resources of the mobile devices to accomplish these tasks jointly. Such a generalization and resource pooling should be supported by new software and protocols embedded in not only mobile devices but also network devices (e.g., base stations).

A model-free DRL-based scheduling algorithm is another candidate method, where the algorithm is based on the DRL technique proposed in. This algorithm can operate without knowing the dynamics of the system, and hence can address the challenges resulting from the coupled 3C resources and the strategic interactions of the mobile devices.

Security and privacy are important in wireless networks, especially under the mentioned cooperation scenario. Authentication and monitoring schemes can help to address the security and privacy issues. It is also possible to design a blockchain system to ensure the security of the transactions of the resources.

Device mobility makes the cooperation groups in D2D resource sharing time-varying, and frequently disconnected cooperation links may harm the system performance by introducing significant signaling overhead. Thus, it is important to propose effective and robust scheduling algorithms that can address the uncertain mobility of the devices. Machine learning techniques are candidate methods for addressing the device mobility problem through analyzing the mobility behaviors and trajectories of the mobile devices.

Conclusion

To address the device heterogeneity in IoT systems, we introduce a joint 3C resource sharing framework. This framework allows mobile devices to run different applications to effectively utilize the 3C resources through the coordination of D2D transmission. It generalizes a wide range of existing models on D2D resource sharing, and it improves the 3C resource allocation efficiency through exploiting the heterogeneities of the devices and the tasks. We elaborate the optimization and economic issues of the 3C framework, discussing both the key challenges and potential solutions. The proposed solutions enable the implementation of the proposed framework in the real world, as they provide practical methods for resource optimization and resource sharing motivation. We further outline several important directions for future research.

Acknowledgments

This work is supported by the General Research Fund CUHK 14219016 from Hong Kong UGC, the Presidential Fund from the Chinese University of Hong Kong, Shenzhen, and the Shenzhen Institute of Artificial Intelligence and Robotics for Society. It is also supported by the National Natural Science Foundation of China (Grant No. 61972113), the Basic Research Project of the Shenzhen Science and Technology Program under Grant JCYJ20180306171800589, and the Guangdong Science and Technology Planning Project 2018B030322004.

References

[1] F. C. Delicato, P. F. Pires, and T. Batista, *Resource Management for Internet of Things*, Springer, 2017.

[2] P. Porambage *et al.*, "Survey on Multi-Access Edge Computing for Internet of Things Realization," *IEEE Commun. Surveys & Tutorials*, vol. 20, no. 4, 4th qtr. 2018, pp. 2961–91.

[3] A. Ndikumana *et al.*, "Joint Communication, Computation, Caching, and Control in Big Data Multi-Access Edge Computing," *IEEE Trans. Mobile Comp.*, 2019 (Early Access).

[4] M. Tang *et al.*, "Optimizations and Economics of Crowdsourced Mobile Streaming," *IEEE Commun. Mag.*, vol. 55, no. 4, Apr. 2017, pp. 21–27.

[5] J. Xu *et al.*, "Designing Security-Aware Incentives for Computation Offloading via Device-to-Device Communication," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, Sept. 2018, pp. 6053–66.

[6] L. Feng *et al.*, "Resource Allocation for 5G D2D Multicast Content Sharing in Social-Aware Cellular Networks," *IEEE Commun. Mag.*, vol. 56, no. 3, Mar. 2018, pp. 112–18.

[7] X. Chen *et al.*, "Exploiting Massive D2D Collaboration for Energy-Efficient Mobile Edge Computing," *IEEE Wireless Commun.*, vol. 24, no. 4, Aug. 2017, pp. 64–71.

[8] A. Destounis, G. S. Paschos, and I. Koutsopoulos, "Streaming Big Data Meets Backpressure in Distributed Network Computation," *Proc. IEEE INFOCOM*, San Francisco, CA, 2016.

[9] M. Tang, L. Gao, and J. Huang, "Enabling Edge Cooperation in Tactile Internet via 3C Resource Sharing," *IEEE JSAC*, vol. 36, no. 11, Nov. 2018, pp. 2444–54.

[10] S. Guo *et al.*, "Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing," *IEEE Trans. Mobile Comp.*, vol.18, no. 2, Feb. 2019, pp. 319–33.

[11] L. Gao *et al.*, "Multi-User Cooperative Mobile Video Streaming: Performance Analysis and Online Mechanism Design," *IEEE Trans. Mobile Comp.*, vol. 18, no. 2, Feb. 2018, pp. 376–89.

[12] V. Mnih *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, Feb. 2015, pp. 529–33.

[13] M. Tang *et al.*, "Multi-Dimensional Auction Mechanisms for Crowdsourced Mobile Video Streaming," *IEEE/ACM Trans. Networking*, vol. 26, no. 5, Oct. 2018, pp. 2062–75.

Biographies

Ming Tang [S'16] (mingt@ece.ubc.ca) is a postdoctoral research fellow at the Department of Electrical and Computer Engineering, University of British Columbia. She received her Ph.D. degree from the Department of Information Engineering, Chinese University of Hong Kong. Her research interests include wireless communications and network economics.

Lin Gao [SM'16] (gaol@hit.edu.cn) is an associate professor in the School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China. He received his Ph.D. degree in electronic engineering from Shanghai Jiao Tong University in 2010. He won the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2016. His research interests are in the area of network economics and games.

Jianwei Huang [F'16] (jianweihuang@gmail.com) is a Presidential Chair Professor and the associate dean of the School of Science and Engineering, Chinese University of Hong Kong, Shenzhen. He is also the Vice President of the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS) and a professor in the Department of Information Engineering, Chinese University of Hong Kong. He is the coauthor of nine Best Paper Awards, including the IEEE Marconi Prize Paper Award in Wireless Communications 2011. He has co-authored six books, including the textbook *Wireless Network Pricing*. He has served as the Chair of the IEEE ComSoc Cognitive Network Technical Committee and Multimedia Communications Technical Committee. He has been an IEEE ComSoc Distinguished Lecturer and a Clarivate Analytics Highly Cited Researcher. More information at http://jianwei.ie.cuhk.edu.hk/.

It is important to propose effective and robust scheduling algorithms that can address the uncertain mobility of the devices. Machine learning techniques are candidate methods for addressing the device mobility problem through analyzing the mobility behaviors and trajectories of the mobile devices.