A Survey of Opportunistic Offloading

Dianlei Xu, Yong Li, *Senior Member, IEEE*, Xinlei Chen, Jianbo Li, Pan Hui, *Fellow, IEEE*, Sheng Chen, *Fellow, IEEE*, and Jon Crowcroft, *Fellow, IEEE*

***Abstract*—This paper surveys the literature of opportunistic offloading. Opportunistic offloading refers to offloading traffic originally transmitted through the cellular network to opportunistic network, or offloading computing tasks originally executed locally to nearby devices with idle computing resources through opportunistic network. This research direction is recently emerged, and the relevant research covers the period from 2009 to date, with an explosive trend over the last four years. We provide a comprehensive review of the research field from a multi-dimensional view based on application goal, realizing approach, offloading direction, etc. In addition, we pinpoint the major classifications of opportunistic offloading, so as to form a hierarchical or graded classification of the existing works. Specifically, we divide opportunistic offloading into two main categories based on application goal: traffic offloading or computation offloading. Each category is further divided into two smaller categories: with and without offloading node selection, which bridges between subscriber node and the cellular network, or plays the role of computing task executor for other nodes. We elaborate, compare, and analyze the literatures in each classification from the perspectives of required information, objective, etc. We present a complete introductory guide to the researches relevant to opportunistic offloading. After summarizing the development of the research direction and offloading strategies of the current state-of-the-art, we further point out the important future research problems and directions.**

***Index Terms*—Traffic offloading, computation offloading, opportunistic network, device-to-device communication, delay tolerance network.**

Manuscript received October 16, 2017; revised December 23, 2017; accepted February 1, 2018. Date of publication February 21, 2018; date of current version August 21, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61502261, Grant 61572457, and Grant 61379132, in part by the Key Research and Development Plan Project of Shandong Province under Grant 2016GGX101032, and in part the Science and Technology Plan Project for Colleges and Universities of Shandong Province under Grant J14LN85. *(Corresponding author: Jianbo Li.)*

D. Xu is with the Computer Science and Technology College, Qingdao University, Qingdao, China, and the Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China (e-mail: xudianlei916@163.com).

Y. Li is with the Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

X. Chen is with the Department of Electronics and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15289 USA (e-mail: xinlei.chen@sv.cmu.edu).

J. Li is with the Computer Science and Technology College, Qingdao University, Qingdao 266071, China (e-mail: lijianbo@qdu.edu.cn).

P. Hui is with the Department of Computer Science, University of Helsinki, Helsinki, Finland, and the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: panhui@cse.ust.hk).

S. Chen is with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., and also with King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: sqc@ecs.soton.ac.uk).

J. Crowcroft is with the Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, U.K. (e-mail: jon.crowcroft@cl.cam.ac.uk).

Digital Object Identifier 10.1109/COMST.2018.2808242

I. Introduction

With the increasing popularity of smart mobile devices, our lifestyles have been altered dramatically, and we are increasingly relying on cellular networks. Data-hungry applications like video streaming and social sharing are becoming more and more popular, which brings us great convenience but put a huge burden on the cellular network. According to Cisco's forecast [1], global mobile data traffic will increase sevenfold between 2016 and 2021, while over 78% of this mobile traffic will be video by 2021. The constantly increasing traffic is a big problem for cellular operators. According to this increasing trend in mobile data traffic, the cellular network is likely to become more and more congested in the future, and mobile users may experience degraded quality of service (QoS), e.g., missing call, low download speed or even no cellular link, etc. Furthermore, mobile applications are becoming more and more computationally demanding, due to increasingly heavy applications on mobile devices. Resource hungry applications like augmented reality often need to perform tasks that require the computing resource beyond the capability of single mobile device, which is a big problem for application providers. With the emergence of cloud computing, mobile devices can enhance their computing capacity via uploading tasks to the cloud through the cellular network but this will in turn put a big burden on the already overloaded cellular network.

To cope with these two problems, it is urgent to provide a promising solution. The most straightforward way is to update the cellular network to the next-generation network, including the deployment of more base stations (BSs) and/or WiFi access points (APs), to increase the capacity of the cellular network. In this way, we may have enough capacity in the cellular network to support our ever-increasing demand for mobile traffic as well as to upload our heavy tasks that cannot be performed locally to the cloud through the cellular network. However, this solution is not so attractive and may be ineffective. This is because upgrading the cellular network is usually expensive and the financial return can be low. Even if the cellular network is updated to the next generation with higher capacity, the increasing demand will soon surpass the capacity of the new generation of cellular network.

Opportunistic offloading [2], [3], as a promising solution, has been proposed recently to solve the aforementioned two problems. The basic idea of opportunistic offloading is based on green wireless communications [4], which leverages the opportunistic network formed by mobile devices to offload traffic data or computing tasks. Hence, there are two specific types of opportunistic offloading, traffic

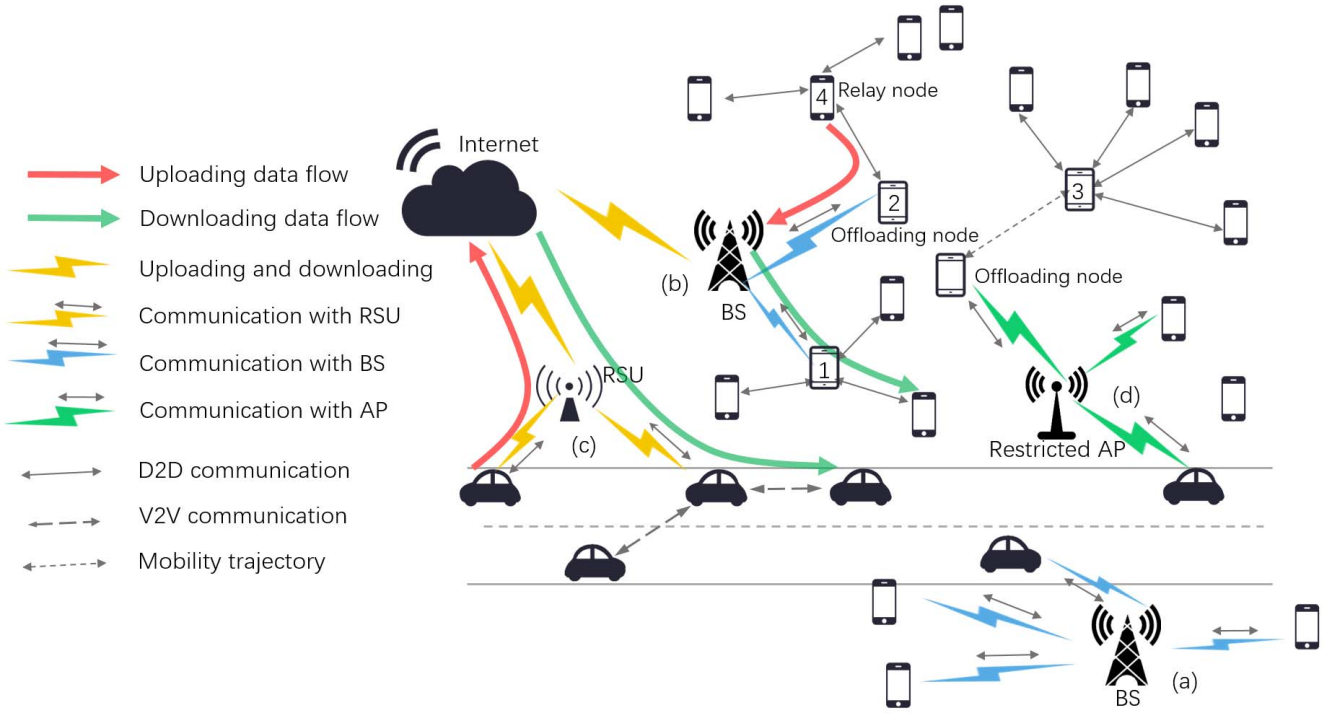


Fig. 1. The contrast of traditional cellular transmission and traffic offloading. In (a), each node downloads its requested content from the BS in traditional manner. In (b), the BS first transmits popular content to a few selected nodes through cellular network. Then, these nodes will deliver the data to the subscriber nodes through D2D manner. In particular, offloading node 1 adopts one-hop delivery while offloading node 2 adopts two-hops delivery. In areas (c) and (d), traffic offloading are assisted by the fixed RSU and other AP, respectively, where opportunistic offloading with the aid of mobile nodes also co-exist.

offloading and computation offloading. In traffic offloading, some mobile nodes[1] download popular contents through the cellular network, and transmit these contents to other subscriber nodes through opportunistic communication. In this way, subscriber nodes can get the requested content without accessing the Internet through the cellular network, which helps to significantly reduce the traffic load on the cellular network. Unlike the existing WiFi offloading, this new type of traffic offloading relies on opportunistic device-to-device (D2D) communication. Computation offloading, or Fog computing, which is first introduced by Cisco in 2012 [5], is used to augment the computational capabilities of mobile devices. It is named mobile edge computing in some works [6], [7]. In computation offloading, a node with limited computing resources can offload computing tasks through opportunistic communication to other mobile devices nearby which have spare computing capacity. After the tasks are finished by these 'helpers', the result will be retrieved back in the same manner.

Opportunistic offloading is feasible and effective due to the following reasons.

- Most of our applications, like podcast, weather forecast, e-mail, etc., are non-real time, and they can tolerate some delay in their delivery. Therefore, cellular operator may send data to a small number of selected users, who will then further propagate data to subscriber users. As long as delay is not too serious, it would not degrade the user experience.
- Popular content downloading causes huge amount of redundant data transmissions in the network. According to statistics, 10% of the top popular videos account for 80% of views in Youtube [8]. In fact, only a small subset of nodes need to download the popular content through the cellular network. Others can obtain the content from these nodes through opportunistic communication.
- Today's smart mobile devices offer large amount of computing resources [9]. Most of these computing resources are idle at most time. A complex task beyong the computing capacity of a single mobile device can be divided into smaller subtasks which are distributed through opportunistic communication to other mobile devices with idle computing resources for completion.
- Opportunistic offloading makes economic sense – there is almost no extra monetary cost involved. Data are directly transmitted among mobile nodes, which 'offloads' huge amount of traffic away from the cellular network and helps to prevent network congestion. Therefore, opportunistic offloading is beneficial to the cellular operator, and the user should not be charged for it.

In effect, there are already some works on opportunistic offloading, which have been applied in practice. Some APPs deployed on smart devices have been developed to facilitate traffic offloading. Some researchers design an application named Cool-SHARE [10], deployed on Android platforms to realize seamless connection. SmartParcel [11] is an APP deployed on Android platform to share delay-tolerant data among spatio-temporally co-existing smart phones, e.g., news,

[1]Mobile node, mobile device and mobile user are interchangeable in this paper.

videos. Han and Ansari [12] design a 'green content broker', driven by solar energy to deliver popular content to requested users nearby. The 'green content broker' not only can decrease accesses to BS, but also can reduce the CO_2 of mobile networks. Existing cloud-based applications will turn to fog-based mode (i.e., opportunistic computation offloading) in the future [16]. Traditional cloud-based applications usually offload computing tasks to remote server, which may lead to significant delay, consisting of application upload to the cloud, result download back and execution time at the cloud. Such a delay makes it inconvenient for real-time applications. Fog computing, characterized by edge location, location awareness, low latency and geographical distribution is a promising choice for these applications [5], [14]. Mobile devices can offload the computation tasks to Fog servers, e.g., cloudlet and smart phones to augment computing capacity and save energy. Moreover, fog computing will be applied in the future 5G network [13], [15].

There are several surveys on offloading [17]–[20], but they are very different from ours. Rebecchi *et al.* [17] provided a survey on traffic offloading solutions, in which mobile devices with multiple wireless interfaces are efficiently used. The authors classified all the solutions that can be used in offloading cellular traffic as infrastructure based offloading and non-infrastructure based offloading. The main idea of infrastructure based offloading is to deploying some infrastructure with the functions of computing, storage and communication to assist data transmission. The infrastructure refers to small cells or APs. Non-infrastructure based traffic offloading means that no extra infrastructure is available, except for the existing BSs. Mobile users either get the requested content from BS, or from other mobile users through opportunistic communications. The focus of [17] is on offloading traffic with the assistance of infrastructure devices. By contrast, we focus on offloading cellular traffic through opportunistic network, consisting of mobile devices, as well as on computation offloading, which was not considered in [17]. More specifically, [17] takes delay as the single matric to classify these related works of traffic offloading based on opportunistic networking into two categories, delayed offloading and non-delayed offloading. Different from [17], we review these existing related works in this area from a multi-dimensional view, e.g., the application goal, realizing approach, offloading direction, etc, and then compare these works in the same dimension in various aspects, e.g., realization method, applicable scenario, advantages and drawbacks.

Similarly, Khadraoui *et al.* [18] reviewed conventional traffic offloading with the assistance of WiFi by coupling architectures of WiFi and cellular network in traffic offloading. They divided these related works into three categories, loose coupling, tight coupling and very tight coupling, in which all users cannot tolerate disconnection between mobile devices and WiFi, or cellular network. These offloading methods are totally different from our work since our focused offloading is non-real time content through opportunistic networks, in which latency occurs naturally.

Chen *et al.* [19] reviewed the existing traffic offloading works and divided them into three categories, traffic offloading through small cells, WiFi networks, or opportunistic communications. While our survey focuses on traffic offloading through opportunistic network, and also include computational offloading. Besides, energy efficiency problem in traffic offloading through small cells is the main discussion point in [19], while we discuss the energy efficiency problem in traffic offloading through opportunistic network, rather than through small cells.

Pal [20] provided a survey on works that extend traditional cloud computing. They divided the existing work on this area into three classifications, Device to cloud (D2C) architecture, Cloudlet to Device (C2D) architecture and D2D architecture, according to the way they deliver services. In D2C architecture, mobile devices connect with remote server through infrastructure (e.g., WiFi AP). The computing tasks are performed on the remote server. C2D architecture uses cloudlets to augment the computation capability of mobile device. In D2D architecture, mobile cloud, formed by mobile devices are used to execute computing tasks for other devices. Different from [20], we provide a classification of computation offloading works based on the offloading modes. That is who, when and where to undertake the task execution job. Strictly speaking, the D2C and C2D modes in [20] are not based on opportunistic network but infrastructure. In contrast, our survey mainly focus on offloading computing tasks to peers with spare computation resources.

The goal of our paper is to offer an introductory guide to the development and the state-of-art in opportunistic offloading. To the best of our knowledge, this is the first effort in this direction. Therefore, our main contributions can be summarized as follows.

- We review the recent advances on opportunistic offloading techniques covering both traffic and computation offloading protocols and techniques working in an opportunistic context or behavior. We categorize existing works from a multi-dimensional view based on application goal, realizing approaches, offloading direction, etc.
- We elaborate, compare and analyze the works in the same category from various aspects, e.g., realization method, applicable scenario, advantages and drawbacks.
- We discuss the open problems and challenges in realizing opportunistic offloading and outline some important future research directions.

The rest of our paper is structured as follow. We provide an overview of our survey in Section II. In Section III, we introduce the development and the state-of-the-art of traffic offloading, while Section IV is devoted to the development and related work on computation offloading. After discussing the future direction and problems in Section V, we conclude this survey in Section VI.

II. Overview

For the convenience of readers, we start by contrasting traditional cellular transmission with traffic offloading in Fig. 1. Traditional cellular transmission is shown in the area (a) of Fig. 1, where each node downloads its content from the BS. In the traffic offloading shown in the area (b), a few selected

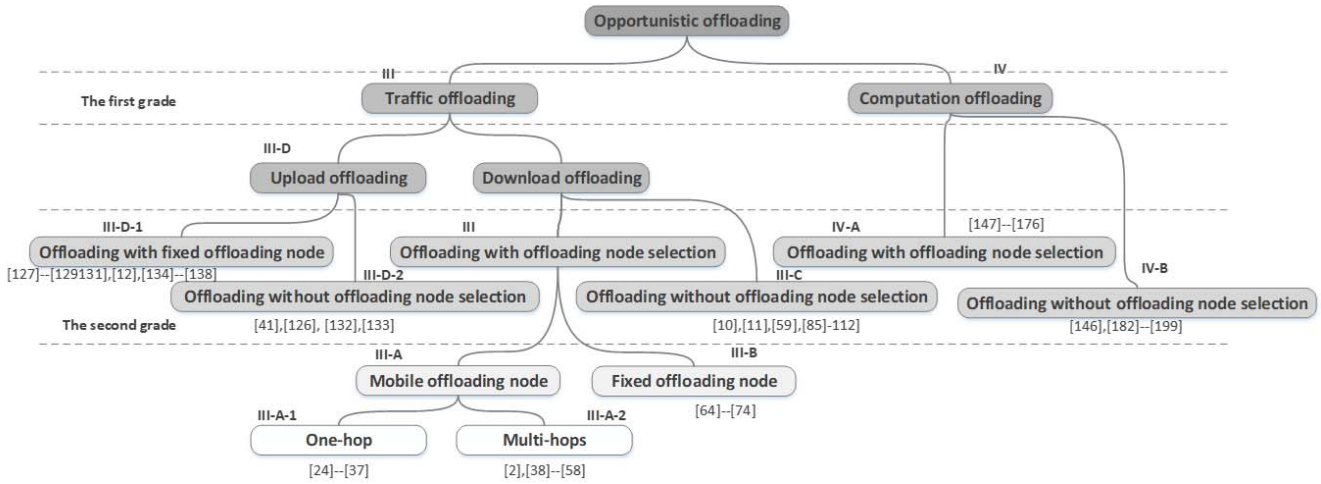


Fig. 2. Hierarchical classification of the opportunistic offloading literature. The first grade is divided according to application goal, and the second grade is divided according to realizing approach. The Roman numeral at each sub-category indicates the section/subsection where the sub-category is discussed.

mobile nodes first download popular content then transmit it to subscriber nodes through D2D based opportunistic communication. Green arrow indicates downloading data flow, while red arrow refer to uploading data flow. In the areas (c) and (d), traffic offloading are assisted by the fixed road side unit (RSU) and other AP, respectively, where opportunistic offloading also co-exist with the assistance of D2D based opportunistic communication.

In this section, we review opportunistic offloading strategies of the existing works and provide a comprehensive classification of them. As mentioned previously, according to different application goals, opportunistic offloading can be divided into two main categories: traffic offloading and computation offloading. The goal of traffic offloading is mainly to transfer the data originally transmitted through the cellular network to an opportunistic network to alleviate the overloaded cellular network. While the goal of computation offloading is to allocate computing tasks to nearby smart devices with idle computing resource to enhance the computing capability of the client device. In both applications, the most important step in the realization of opportunistic offloading is to select a subset of nodes. We refer to these nodes as offloading nodes, which help other nodes to deliver content in traffic offloading or help other nodes to perform computing tasks in computation offloading. Thus, we consider a hierarchical approach in our classification. Specifically, the opportunistic offloading strategies are divided into two sub-categories: offloading with offloading node selection and offloading without offloading node selection. Then, according to the physical characteristics of offloading node, the sub-category of offloading with offloading node selection can be divided into two smaller categories: with mobile offloading node and with fixed offloading node. Since offloading node can adopt one-hop delivery or multi-hops delivery, this sub-category can also be subdivided into two smaller categories: one-hop and multi-hops offloading. The big picture of the proposed hierarchical or graded classification is shown in Fig. 2, together with the associated literature.

A. *Traffic Offloading*

The huge amount of redundant data in the Internet and the increasing popularity of intelligent mobile devices make it necessary and feasible for offloading the traffic to opportunistic network formed by mobile devices.

Instead of every mobile node downloading its required content directly through the cellular network, in download offloading, only some mobile devices download popular content from BS, and then transmit the content to other mobile devices that are interested in the content through opportunistic contact. Some offloading schemes may appoint a subset of nodes as offloading nodes to help offloading. These schemes focus on designing the algorithms for selecting the optimal subset of offloading nodes to achieve the pre-determined objectives. Other offloading schemes do not explicitly select the subset of offloading nodes. These schemes pays attention to the network architecture itself, e.g., how to design incentive mechanism to motivate mobile nodes to participate in traffic offloading and when to re-inject content copies into the network. The offloading nodes may be ordinary mobile devices,[2] or some fixed devices deployed in the network, such as RSUs and functionally restricted WiFi APs.[3] In collaborative scenarios where mobile nodes are willing to forward data to other nodes, the offloading nodes can transmit the data to some relaying nodes which are more likely to contact the subscriber nodes that request for the data. Hence, data delivery adopts a multi-hop mode. When no mobile node is willing to act as relay, an offloading node must directly deliver the data to the subscriber nodes and, consequently, data delivery has to adopt a one-hop mode. Refer to Fig. 1 again. In the area (b), node 1 is an offloading node, which directly delivers the content to subscriber nodes, while offloading node 2 selects node 4 as the relay node and adopts the two-hops mode to deliver the content. Also in the area (c), the RSU acts as a

[2]Mobile devices and smart devices are interchangeable in this paper.

[3]The functionally restricted WiFi AP refers to the AP that only store popular content in advance or can only be used for uploading.

fixed offloading node, while in the area (d), the AP is the fixed offloading node.

Rather than every mobile node uploading data directly to BS, in upload offloading, each mobile node transmits data to other nodes through opportunistic contact to indirectly upload data to BS. Similarly, there are also two basic approaches in upload offloading: selecting a subset of offloading nodes to help offloading and offloading without selecting offloading nodes. An important difference can be observed between upload offloading and download offloading. Unlike download offloading, where data are popular content and many mobile users request the same content, there is no redundancy of data in upload offloading, as each mobile user's data is unique. Most upload offloading schemes rely on fixed devices with Internet access capabilities to upload data, and offloading nodes in upload offloading are mainly RSUs or WiFi APs.

There are two most important features in traffic offloading: redundancy and delay tolerance.

- *Redundancy:* The content to be offloaded must be popular, i.e., many mobile users are interested in the same content or request for the same content. Thus, only a small subset of users directly download the content from the cellular network, while majority of mobile users can get the content from these offloading nodes through opportunistic network. In this kind of scenarios, traffic offloading is most effective and efficient.
- *Delay tolerance:* In opportunistic network, there exists no fixed and stable path among mobile users, and the content delivery totally depends on the mobility of users. Users must tolerate a certain random delay before receive the requested content. In other words, traffic offloading is unsuitable for real-time applications.

B. Computation Offloading

In computation offloading, the mobile node that initiates the computing task offloading is referred to as the client node, while the mobile nodes that perform the tasks for the client node are called the offloading nodes. A client node sends the task to some neighbour mobile nodes with idle computing resources through opportunistic communication. After the task is completed, the result will be retrieved back through opportunistic communication.

Some computation offloading schemes may specify an explicit subset of offloading nodes to perform the computing tasks for the client node based on the computational requirements of the tasks to achieve certain goals, such as minimizing the energy consumption, maximizing the lifetime of devices and/or minimizing the completion time. These schemes must deal with the problem that which node should be selected as an offloading node and how much workload should be allocated to it. Some computation offloading schemes by contrast do not explicitly select the offloading nodes for the tasks. For example, each node in a cluster can perform the tasks for other nodes, and there is no need to specify which node should act as the offloading node. Or the cloudlet consisting of mobile devices has already been formed.

Compared to traditional mobile computation, computation offloading has a distinct advantage, in terms of energy consumption and completion time. In traditional mobile computation, mobile device uploads the whole computing task to remote cloud. The persistent connection to the remote cloud will consume large amount of energy. Moreover, task uploading, task execution and result retrieval all contribute to large delay. By contrast, in computation offloading, a task may be partitioned into several small subtasks, which can be performed in parallel. These subtasks can be transmitted to nearby devices with idle computing resources through opportunistic communication. Parallel execution of the task can significantly reduce energy consumption and completion time, while task uploading and result retrieval from nearby devices cause less delay than the case of remote cloud.

Benefiting from this significant advantage, a large amount of works on computation offloading have been proposed, as indicated in the right part of Fig. 2. Two important references that are not listed in Fig. 2 are elaborated here.[4] The survey by Pal [20] divides all available techniques and solutions for computation offloading into three categories according to offloading schemes employed: device-to-cloud (D2C), cloudlet-to-device (C2D) and device-to-device (D2D). Clearly, only D2D offloading is based on opportunistic network. Tapparello *et al.* [21] survey the state-of-art parallel computing techniques by dividing them into three categories: cluster computing, distributed computing and volunteer computing. Cluster computing is based on a group of co-located computers, and distributed computing is based on the Internet, while only volunteer computing is based on opportunistic network.

Two most important features in computation offloading are idle computing resources and delay tolerance.

- *Idle computing resources:* Most smart devices are underutilized in terms of their computing capability, and most of the time, these smart devices are idle. For example, a typical smart phone such as the Samsung C9 is equipped with a 1.95 GHz eight-core CPU and 6 GB RAM. These idle computing resources can be effectively utilized in computation offloading.
- *Delay tolerance:* In opportunistic network, the connections between client device and offloading devices are intermittent, causing certain random delay in sending tasks to offloading nodes and retrieving results from them. The task completion time by an offloading device is also inherently random. Therefore, tasks in computation offloading must be non-real time. However, this delay is typically smaller than the delay caused by remote cloud computing, because offloading nodes are nearby.

C. Summary

Traffic offloading and computation offloading considered in this survey are both based on opportunistic network. These have recently emerged two most important application areas of opportunistic offloading, and huge volume of researches have been carried out to investigate and realize traffic offloading

[4]They are not listed in Fig. 2 for a specific type of computation offloading, because they discuss both types of computation offloading.

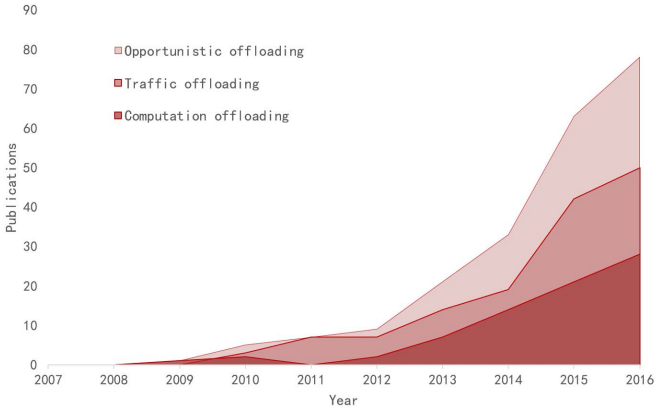


Fig. 3. Publication volume over time. The deep color area shows the trend of the number of publications in computation offloading over time, and the light color area depicts the trend for opportunistic offloading, while the intermediate area indicates the trend for traffic offloading.

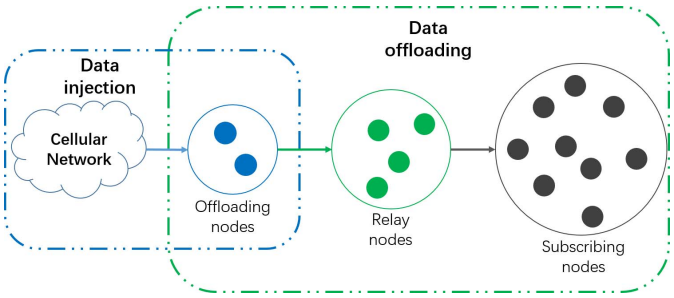


Fig. 4. Download offloading process of traffic offloading. The cellular operator first selects a subset of mobile nodes as offloading nodes, and inject the content to them. Then, offloading nodes transmit the content to relay nodes or directly deliver the content to subscriber nodes. Relay node will transmit the content to other relay nodes or deliver it to subscriber nodes.

and computation offloading. We should note that, significant amount of research on computation offloading has been proposed over past twenty years. For instance, some works leverage 'Cyber Foraging' to augment the computational and storage capabilities of mobile devices [22], [23]. In these works, hardware in wired infrastructure, called *surrogate* is used to perform computing tasks for mobile devices. However, these works are all based on infrastructure. Offloading computing tasks to peers through opportunistic network is a new area. For the convenience of reader, we show the publication statistics on offloading in Fig. 3. Observe from Fig. 3 that the first work on traffic offloading was published in 2010, while the first work on computation offloading was published in 2009. Most strikingly, since 2012, the growth in the publication volume of opportunistic offloading has been dramatic, at about 50% annual growth rate, which indicates that this research area is in its explosive development stage. This can be attributed to the following two reasons.

The first reason is the 'embarrassing' situation of the overloaded cellular network and mobile computation that our world is facing. With the increasing popularity of mobile devices, our demand for mobile Internet service is explosively growing, which puts a big burden on the cellular network. On the other hand, we are increasingly interested in large applications, which require computing power beyond the capability of individual mobile devices. This situation creates the necessity for traffic offloading and computation offloading.

The second reason is that research community and industry have raised to face this challenge. Theory and practice of opportunistic offloading mature very fast. It has passed the stage of theoretical concept proof, and some researches have developed apps or platforms to evaluate the performance of opportunistic offloading, e.g., Cool-SHARE in [10], CoMon in [9], etc. We now have effective means of investigating and realizing traffic offloading and computation offloading.

III. Traffic Offloading

Different from traditional infrastructure-based networks, there exists no fixed end-to-end paths in the opportunistic networks due to the node mobility. However, node mobility can also be utilized to create opportunistic communication paths between nodes. When two non-adjacent nodes move into each other's communication range, they can communicate through short range communication techniques, e.g., WiFi direct and Bluetooth. Opportunistic network is also called delay tolerant network (DTN), which adopts the store-carry-forward mechanism or protocol to deliver data from the source node to the destination node without the need of an end-to-end path. However, there is a price to pay, specifically, the delivery of content will suffer from certain random delay. Therefore, the content to be transmitted through opportunistic network must be delay tolerant. Some content, like e-mail, podcast and weather forecast, etc., do not require real-time, and they are well suitable for this type of offloading. Furthermore, there is no monetary cost involved for the mobile users collaborating in opportunistic communication. For these reasons, D2D based opportunistic network is deemed to offer a promising approach to offload traffic from the overloaded cellular network. The other essential feature of the content that can be offloaded is popularity. Popularity here means that the content is of interest to many users. In other word, the content is redundant in the network. For example, the most popular 10 percent videos account for 80 percent traffic [8]. If a group of nodes are interested in the same video, one of them may download the video through the cellular link, and transmits the content to other group members in proximity via D2D communication. In this way, a large amount of cellular traffic can be saved, and the efficiency of traffic offloading is very high. On the other hand, if the content is of interest to only one node, the offloading efficiency will be zero.

According to flow direction, traffic offloading can be divided into two categories, download offloading and upload offloading. Download offloading process usually consists of 2 steps as shown in Fig. 4. In the first step, cellular operator selects a subset of mobile users as offloading nodes based on some selection strategies, and injects the content to them. To reduce the amount of cellular traffic required, the subset of offloading nodes should be minimized, while meeting other constraints. Offloading nodes usually are those nodes that have the greatest capacity to transmit content to other nodes. In the second step, offloading nodes transmit the content to subscriber nodes

with one-hop or multi-hops. One-hop delivery means that offloading node can deliver the content to subscriber node directly because the latter is within its communication range. By contrast, multi-hops delivery is required, if subscriber node is outside the direct communication range of offloading node. The offloading node could forward the content to other relay nodes, who may eventually deliver the content to the subscriber node via the store-carry-forward mechanism. Multi-hops mode can accelerate the delivery of the content to subscribe nodes, shorten delivery delay and reduce the workload of offloading node. However, one-hop mode has to be adopted if mobile nodes are non-cooperative, i.e., nodes are unwilling to act as relay for other nodes. Similar to download offloading, upload offloading process also involves two steps. In the first step, the mobile nodes with data to upload first deliver the data to offloading nodes selected based on some strategies. In the second step, these offloading nodes then upload the data to cloud or BS, e.g., via WiFi or cellular link, at the expense of their own bandwidths.

In the literature, offloading strategies are classified into three categories based on means different offloading manners: offloading with mobile offloading node, with fixed offloading node and without offloading node selection. In the first category, mobile offloading nodes are selected from mobile subscriber nodes, and they adopt the store-carry-forward mode to exploit nodes' mobility. Most works in this category focus on how to optimally select the subset of offloading nodes from subscriber nodes. In download offloading, in particular, this is equivalent to solve the problem that which node should download through cellular link. The fixed-offloading-node scenarios refer to deploying fixed devices, e.g., RSU and/or functionally limited WiFi AP, which can download the requested content from the Internet through cellular link. A fixed offloading node adopts the store-forward protocol, and it transmit the content to other nodes passing by through short range communication. Most works in this category focus on how to pre-fetch the requested content. In the strategies of offloading without offloading node selection, there is no need to decide which node should be offloading node, that is, every subscriber node can be offloading node. For example, a group of users with the same interests may download bulk data through mutual cooperation to accelerate the transmission and save traffic. Specifically, each user download only a part of the bulk data, and then exchange the part of the content downloaded with other users in the group. Works in this category focus on a higher layer of opportunistic offloading by solving the problem that what content can be offloaded and how to offload, in terms of offloading designing mechanisms, such as energy efficiency, P2P offloading, incentive mechanism, etc. We now discuss all these three strategies in details, according to the literature.

A. Offloading With Mobile Offloading Node

Mobile nodes equipped with multiple radio interfaces may be selected as offloading nodes to download popular content through the cellular network and then to transfer the content to other requesting nodes [26]. An offloading node may transmit the requested content to the end subscriber nodes with one hop or with multi hops. Therefore, the offloading schemes in this category may be divided into two sub-categories: one-hop and multi-hop. Table I summarizes the existing literature for this category.

1) One-Hop: Barbera *et al.* [25], [30] leverage the social attributes of mobile nodes to select socially important mobile nodes as offloading nodes. The authors build a social graph of mobile nodes in a given area over a certain observation period, and analyze the characteristics of contacts between nodes and mobility patterns. They calculate the values of social attributes to quantify the importance of each mobile node in the area with the assistance of the social graph. The social attributes used include betweenness, closeness, degree, closeness centrality and pagerank. The social graph can be divided into several communities through the application of k-clique algorithm. Two different greedy algorithms are proposed to select the socially important mobile nodes as 'very important persons' (VIPs) which play the role of 'bridge' between the cellular network and the mobile users in every community [30]. Different from [25] and [30], Wang *et al.* [33] propose to build the social graph by leveraging social network services to quantify the importance of mobile nodes. Specifically, they leverage the online spreading impact and the offline mobility pattern to select a subset of offloading nodes to download content directly from the cellular network and to share the content with other requesting nodes in proximity.

Barua *et al.* [36] propose an offloading-node selection algorithm, called 'select best' (SB), based on the link quality between downloading nodes and BS and the link quality among nodes. The link quality between a mobile node and BS may be quantified by the flow rate. Consider the scenario where a group of N mobile nodes are interested in the same content. The BS arranges the link quality of the N users in descending orders, and selects the top N_r users as the candidates of offloading nodes, where N_r depends on the density of users in the area. More specifically, the BS sends a segment to the candidates and let these candidates to forward the segment to the subscriber nodes in their communication ranges. The subscriber nodes send feedback with some preset parameters to the BS, which then selects the offloading nodes from the candidates according to feedback information. Trestian *et al.* [34] proposes an analogous method to select offloading nodes by designing the energy-efficient cluster-oriented solution for multimedia (ECO-M), in which mobile nodes interested in the same multimedia content are arranged into clusters. The head of a cluster is selected as the offloading node. Different from [36], the work [34] is restricted to long-term evolution (LTE) based cellular networks, and the ECO-M only selects one offloading node per cluster. In addition to the link quality, battery level of nodes can also be utilized as the selection metric. In the ECO-M, a few heads of the clusters are responsible for the transmission and consume a lot of energy. Therefore, a weighted multiplicative exponential function is designed in [34] to quantify the willingness of a device to be the cluster head. Mota *et al.* [31] propose the so-called OppLite framework based on multi-criteria, including the number of neighbors, battery level and link quality, to select offloading nodes. Specifically, a multi-criteria utility function is used

TABLE I
LITERATURE SUMMARY OF TRAFFIC OFFLOADING WITH MOBILE OFFLOADING NODE SELECTION

Ref.	Hop	Methodology	Scenario	Required information	Objective
[24]	One-hop	Hycloud project	N/A	N/A	Offload cellular traffic
[25]	One-hop	Community detection	Campus-like	Social attribute	Maximize the offloaded traffic
[26]	One-hop	Multi-interfaces	Heterogeneous	N/A	Reduce energy consumption
[27]	One-hop	N/A	Heterogeneous users	Content popularity	Maximize the reduced traffic
[28]	One-hop	Reinforcement learning	SC, MC	N/A	Timely delivery
[29]	One-hop	Erasure coding	VDTN	User interest	Maximize user's interest
[30]	One-hop	Community detection	Campus-like	Social attribute	Maximize the offloaded traffic
[31]	One-hop	OppLite	Crowed	Multi-criterion	Maximize the offloaded traffic
[32]	One-hop	Submodular optimization	Realistic	N/A	Achieve maximum data offloading
[33]	One-hop	Link SNS, MSN	MSNet	Network service	Maximize the reduced traffic
[34]	One-hop	Cluster-orientation	Homogeneous node	Link quality, battery	Improve energy efficiency
[35]	One-hop	White space	N/A	Distance	Maximize the reduced traffic
[36]	One-hop	Feedback	Homogeneous nodes	Link quality	Maximize user and network payoff
[37]	One-hop	Analytical framework	VDTN	N/A	Minimize the load of cellular network
[38]	Two-hops	Submodular optimization	MADNet	Storage assignment	Maximize the reduced traffic
[2]	Multi-hops	Submodular optimization	MSNet	N/A	Maximize the reduced cellular traffic
[39]	Multi-hops	Submodular optimization	MSNet	N/A	Maximize the reduced cellular traffic
[40]	Multi-hops	Push-and-Track	Homogeneous users	Feedback	Guarantee timely delivery
[41]	Multi-hops	MobiTribe	Offload UGC	Contact pattern	Minimize cellular transmission cost
[42]	Multi-hops	Community detection	MSNet	Social community	Maximize the reduced cellular traffic
[43]	Multi-hops	Opp-Off	MSNet	Human mobility	Minimize the traffic over cellular network
[44]	Multi-hops	Push-and-Track	Homogeneous users	Feedback	Minimize the load on cellular infrastructure
[45]	Multi-hops	TOMP	Homogeneous users	Position, velocity	Timely delivery
[46]	Multi-hops	Random Interest Diffusion	N/A	User interest	Reduce peak traffic
[47]	Multi-hops	Time-dependent function	N/A	Content freshness	Maximize the freshness of delivered data
[48]	Multi-hops	Push-and-Track	Homogeneous users	Feedback	Guarantee 100% delivery
[49]	Multi-hops	Push-and-Track	Heterogeneous users	N/A	Minimize the load on cellular network
[50]	Multi-hops	Reinforcement learning	Homogeneous users	Feedback	Guarantee timely delivery
[51]	Multi-hops	NodeRank	Metropolitan area	Human mobility	Maximize the traffic through opportunistic network
[52]	Multi-hops	Reinforcement learning	Homogeneous users	N/A	Minimize the use of cellular infrastructure
[53]	Multi-hops	Gossip-style cascade	Homogenous users	Marginal effect	Maximize the reduced traffic
[54]	Multi-hops	N/A	Heterogeneous users	Content popularity	Minimize the traffic over cellular network
[55]	Multi-hops	PrefCast	Heterogeneous users	User preference	Satisfy the user preference
[56]	Multi-hops	Link SNS, MSN	MSNet	User tags	Maximize the reduced traffic
[57]	Multi-hops	DOPS	N/A	Distance	Offload cellular traffic
[58]	Multi-hops	Pontryagin maximum	Hybrid	Remuneration	Reduce the data dissemination cost

to decide whether a node should get the requested content through cellular network or via opportunistic network as well as whether a node can be selected as offloading node or not.

Li *et al.* [32] consider a realistic scenario where the heterogeneity of data and mobile users are taken into account. As illustrated in Fig. 5, mobile data have different time to lives (TTLs) and sizes, and mobile users have different interests in mobile data, while the buffer of mobile device is not infinite. The optimal offloading nodes selection problem can be described as a sub-modular optimization problem with multiple linear constraints [32], which is NP-complete. Thus three sub-optimal algorithms are designed in [32] according to different application scenarios. The first one is for the general offloading scenario, and the second one is for the scenario having short TTL contents, while the third algorithm is for the scenario with homogeneous contact rate and data. These solutions can be extended to vehicle networks. Li *et al.* [29] propose to leverage the vehicular delay tolerant network (VDTN), consisting of vehicles, to offload the mobile data with the assistance of erasure coding technique. Data is coded into small segments by erasure coding to provide redundancy, and the offloading nodes are selected based on contact rate. Offloading nodes obtain the segment or data from the cellular network and transmit it to subscriber vehicles when they meet each other. Similar to [32], Chen *et al.* [27] consider the scenario where mobile users have different interest

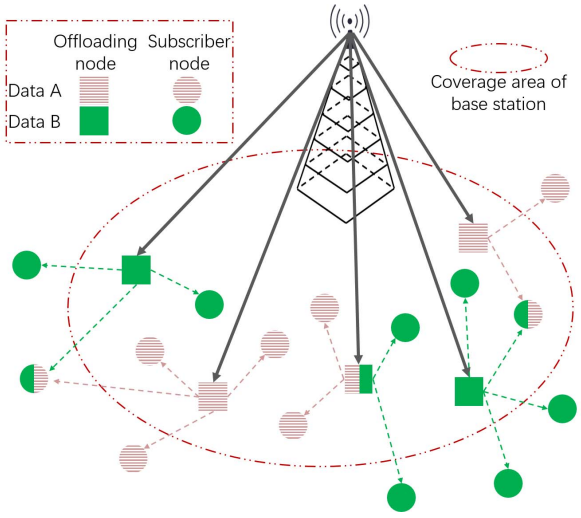
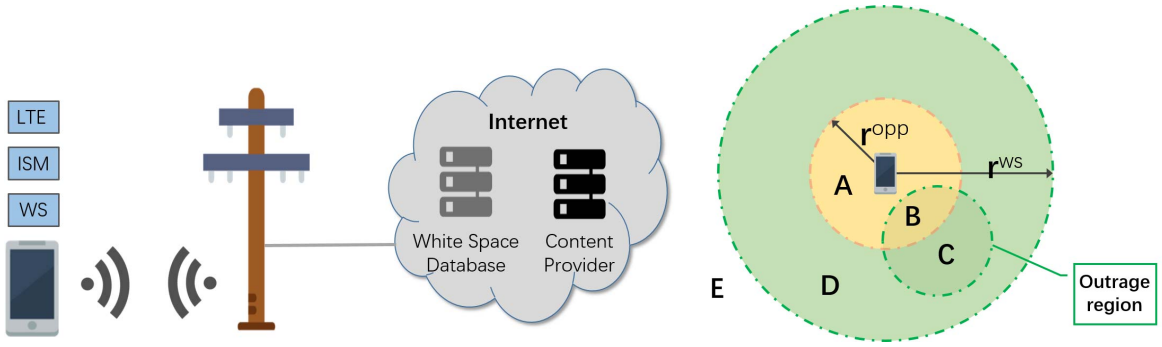


Fig. 5. Illustrations of heterogeneous data offloading through DTN, where there are two different types of mobile data and two different types of mobile nodes. Offloading node downloads popular contents through the cellular network and transmit them to the interested mobile nodes through one-hop.

in mobile content. They propose two offloading-node selection algorithms, termed fully and partial allocation algorithms, to select a subset of offloading nodes for each content according to content popularity. The fully allocation algorithm selects



(a) Reference architecture

(b) Opportunistic offloading in A and B areas; white space offloading in D area; and no offloading in C and E areas.

Fig. 6. Traffic offloading based on the combination of opportunistic communication and white space. Mobile nodes can obtain requested content through white space or opportunistic communications.

offloading nodes to transmit all the content to subscriber users, while the partial allocation algorithm selects offloading nodes to transmit the top several contents via opportunistic network. Different from [32], the schemes of [27] do not take into account the heterogeneity of mobile users.

White space (WS) is the blank band between TV bands to prevent the interference. Considering the congestion of the licensed bands for cellular communications, Bayhan *et al.* [35] proposes to leverage the WS to improve the capacity of cellular network, where the offloading problem is formulated as an NP-hard optimization problem. Several heuristic algorithms are proposed to select offloading nodes. There are three radio interfaces for each mobile node: LTE based cellular interface, short range D2D communication interface based on industrial, scientific and medical (ISM) band as well as WS interface. The BS is connected to content server and a database of white space, as illustrated in Fig. 6(a). When the distance between subscriber node and offloading node is within the short communication range, offloading node transmit the requested content to subscriber node through opportunistic communication. When the distance is longer than the short communicate range but shorter than the white space communicate range, offloading node query the WS database to get an available white space channel for transmitting the requested content to subscriber node, as shown in Fig. 6(b). If the subscriber node could not obtain the requested content before the deadline, content server will transmit the content through cellular link. A common feature of [27] and [35] is that they are both content-centric.

In order to alleviate the overburdened cellular network, Vigneri *et al.* [37] propose to transform vehicles into offloading nodes that can download popular content and transmit the content to other requesting users when they are passing by them. In this architecture, there are three types of nodes: infrastructure node, cloud node and mobile node. Infrastructure nodes are BSs. Cloud nodes are vehicles, such as taxis and buses, which are pre-determined offloading nodes. Mobile nodes are ordinary users with smart devices. Cloud nodes download popular content from infrastructure nodes through cellular link. A mobile node may send request to the nearby cloud node. If the cloud node holds the requested content

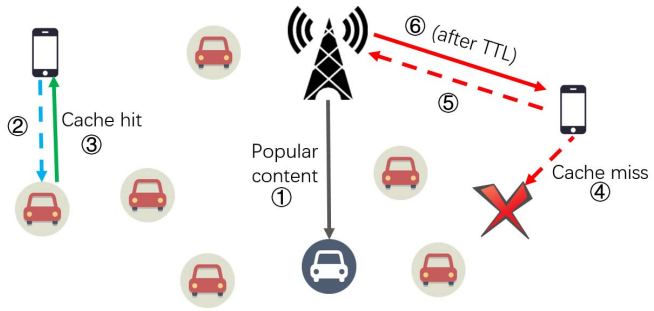


Fig. 7. Illustration of vehicles acting as offloading nodes. ①BS send popular content to the vehicle. ②A user send a request to the vehicle. ③The vehicle deliver the requested content to the user. ④There is no requested content in the vehicle. ⑤The user cannot get the requested content before the TTL, and send a request to the BS through cellular network. ⑥BS send the requested content to the user.

in cache, the requesting node can download the content via short range communicate technique. Otherwise, the mobile node may wait for another cloud node. When the deadline is past, the requesting node has to get the content via cellular link. This offloading process is illustrated in Fig. 7. Different from [29], there is no communication between vehicles, and the buffer of vehicle is not taken into consideration.

2) Multi-Hop: To alleviate the overloaded cellular network, we may select a subset of K offloading nodes. Content provider injects popular content to these offloading nodes via cellular links to initialize offloading. Then offloading nodes transmit the content to other subscriber nodes with opportunistic communication in mobile social network, as shown in Fig. 8. The content has a deadline. When the TTL is past, subscriber nodes that have not received the requested content have to download the content from content provider through cellular link. Note that multi-hop offloading relies on the assumption that mobile nodes are always willing to forward content for other nodes unselfishly, which may not be true. The schemes for delivering content with multi hops may be broadly divided into three classes: schemes that maximize the amount of reduced cellular traffic by offloading, schemes that pre-download popular content before peak time, and schemes that guarantee the timely delivery of content.

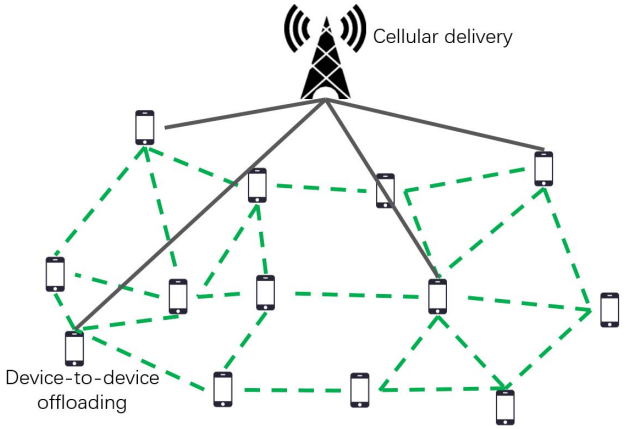


Fig. 8. A contact graph for mobile users, where K users are selected in the graph as offloading nodes. Subscriber users can obtain the requested content from other mobile users or the cellular network.

a) Maximizing the amount of reduced cellular traffic: Han *et al.* [2] study the problem of selecting offloading nodes to maximize the amount of reduced cellular traffic. Since this problem is NP-hard, they design three suboptimal algorithms. They further design Opp-Off [43], a prototype deployed on the smart phones to demonstrate the feasibility of opportunistic communication using Bluetooth to transmit content. Han *et al.* [39] propose the metropolitan advanced delivery network (MADNet) to study the offloading problem in metropolitan area with the cooperation of cellular, WiFi and opportunistic networks. They adopt an offloading-node selection approach similar to [2], but only select offloading nodes based on the contact probabilities among mobile nodes, which is clearly insufficient. For example, when mobile nodes with high contact rates are all concentrated in one community, offloading nodes selected in this manner are unable to achieve a high performance. Furthermore, when the distribution of nodes is sparse in a given area, offloading efficiency will be very low. Similar to [2], Wenxiang *et al.* [53] also select a subset of offloading nodes by maximizing the amount of reduced cellular traffic. However, a gossip-style social cascade model is adopted to simulate the spreading of data in an epidemic manner. The submodularity of the offloading problem is proved in [53] and a greedy algorithm is utilized to select offloading nodes based on marginal effect function.

Wang *et al.* [38] design an enhanced offloading scheme. In this scheme, in addition to selecting offloading nodes, they also select relays which are more likely to encounter subscriber nodes than offloading nodes. Furthermore, there are two types of offloading nodes, mobile offloading nodes and fixed offloading nodes. When requesting for a content, the subscriber node or the cellular operator can appoint several relays to help. A relay downloads the requested content from an offloading node through D2D link when they meet. When the subscriber node encounters an offloading node or a relay that has the content, the content is transmitted to the subscriber node through D2D link. If the subscriber node does not want to wait, it can choose to get the content via cellular link. Upon receiving the requested content, the subscriber node needs to send a notification to the selected relays for them to stop the mission. Mobile offloading nodes can be encouraged by certain incentive mechanism [59], and fixed offloading nodes are provided by the operator. Both of them are predetermined. The key issue is how to select relays for a specific request. Wang *et al.* [38] propose to select the relays according to the encounter patterns, and the scheme can be carried out either in a centralized manner or in a decentralized manner. In the former, the cellular operator can select the top K nodes with the highest encounter rates with subscriber nodes as relays. In the later, the subscriber node sends the record of its preferred top K nodes to the operator along with the request.

Chuang and Lin [42] select offloading nodes based on the nodes' encounter probabilities with disjoint social communities. Cheng and Lin [55] design a preference-aware scheme, called PrefCast, by considering the heterogeneity of user preference, which jointly considers the problems of selecting offloading nodes and forwarding. The nodes belonging to different communities may have different preference on content. PrefCast takes into account the community structure and user preference to select the offloading nodes, by calculating the utility for each node and selecting the top K nodes with the highest utility values as offloading nodes. After offloading nodes transmit the content to other nodes in a community, these nodes must decide whether to forward the content to other nodes in a given time period. Hence, PrefCast also predicts the utility that the forwarding will generate in the future to help the node decide whether to forward the content. A similarity between [42] and [55] is that both select the offloading nodes based on social community information. But the former also designs multi-hops forwarding inside a community. Cheng *et al.* [54] extend th one-hop scheme of [27] to multi-hop scenarios. By evaluating tags on nodes and content, Wang *et al.* [56] extend the work of [33] to enable multi-hop offloading.

Li *et al.* [51] propose NodeRank algorithm, similar to the famous PageRank [60], to select the subset of offloading nodes based on the temporal and spatial characteristics of human mobility. The goal of this offloading scheme is to maximize the number of nodes that obtain the requested content through opportunistic network. Similar to [30], NodeRank builds a contact graph based on history contact information, and then calculates the importance of each node in the graph. NodeRank not only takes into account the contact rates between mobile nodes but also the contact duration and inter-contact time to ensure a full delivery of content.

Lu *et al.* [57] propose an offloading scheme based on distance. When a node requests for a content, the content subscribing server (CSS) gives a deadline. The node can download the requested content from other node that has received the content when they encounter each other. If the node cannot obtain the requested content before deadline, there are two choices for the requesting node: extending the deadline or downloading the content through cellular link. Upon receiving the content, the node can work as an offloading node. Each node periodically uploads its location, and when an offloading node accesses the CSS, it will check the list of requested content. If the offloading node has the requested content, it

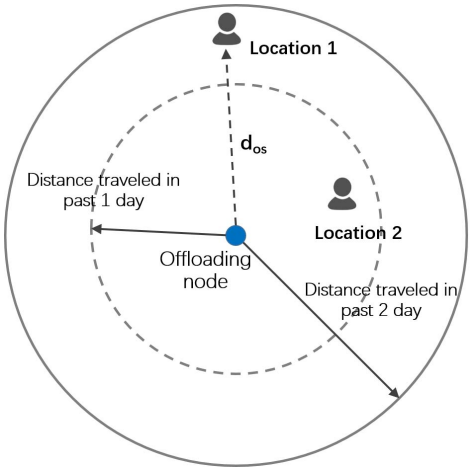


Fig. 9. If the distance between the offloading node and a requesting node is less than the largest distance that it travelled in the past 2 days, the offloading node can transmit the content to the requesting node through opportunistic network, assuming that the remaining time for the content is 2 days.

will check the deadline and the locations of subscriber nodes as well as calculates the remaining time to the deadline. As the example shown in Fig. 9, let us assume that the remaining time for the content is 2 days. if the distance between the offloading node and a subscriber node is shorter than the distance that the offloading node travelled in the past 2 days, the offloading node can decide to transmit the content to the subscriber node through opportunistic network. The work [61] shares the same framework as [57]. The difference between these two works is that a decision mechanism is introduced in [57] for the offloading node to decide whether to send the content to the detected subscriber node through opportunistic network based on its history movement distance, as discussed above.

b) Preload offloading: Proulx and Zhang [46] propose to preload cellular traffic to alleviate the cellular traffic peak. They leverage the social network to predict the traffic demand and pre-download the traffic to a subset of offloading nodes. With this pre-downloading, offloading nodes can transmit the content to other interested nodes through opportunistic network to minimize the usage of cellular network during the peak time. The selection of offloading nodes is based on a greedy preload algorithm. Similarly, nodes can proactively store content for neighbours, as in [62] and [63].

c) Timely delivery of offloading: Whitbeck *et al.* [40] design Push-and-Track with the objective of guaranteeing the delivery delay. A subset of offloading nodes are selected to download the content through cellular link, and they transmit the content to other subscriber nodes in an epidemic manner assuming that all the nodes in the area request for the content. Four strategies are proposed to select offloading nodes: random strategy, entry time strategy, GPS based strategy and connectivity based strategy. A control loop supervises the offloading process. When a node enters the given area, it sends a subscribing message to the control loop, and when a node leaves the area, it sends a un-subscribing message. Subscriber node that has obtained the requested content also sends an acknowledgement to the control loop. The control loop decides how many copies should be injected through offloading nodes, with the reinjection policy for guaranteeing the chosen objective of timely delivery. When the performance of the offloading is lower than the objective, new copies of the content will be injected into the network to guarantee the delivery before deadline. The work [44] extends Push-and-Track framework of [40] to include the float data scenario. Rebecchi *et al.* [48] propose DROid, an offloading framework based on Push-and-Track. In DROid, both the actual performance of infection and the infection-rate trend are taken into account to decide reinjection.

Baier *et al.* [45] design a framework, called traffic offloading using movement predictions (TOMP), to guarantee timely delivery. In TOMP, the area is divided into several sub-areas, and each sub-area has a server in charge. The server injects a content to a subset of offloading nodes, and offloading nodes transmit the content to other mobile nodes in an epidemic way. The authors propose to leverage the location and moving speed of a mobile node to predict its movement in order to estimate its probability of encountering other nodes. Three coverage scenarios, static coverage, free-space coverage and graph-based coverage, are considered to select the subset of offloading node with the highest probability of encountering other nodes correspondingly, as shown in Fig. 10. Similar to [40], there is also a control center in TOMP. Upon obtaining the content from an offloading node or other subscriber node, the receiving node sends an acknowledgement to the control center through cellular link, and the node starts to infect other nodes. In TOMP, the server transmits the content to nodes that does not receive the content after a given time through cellular link to guarantee the 100% delivery, instead of reinjecting more copies of the content to offloading nodes as in [40].

The work [47] proposes to offload cellular traffic through proximity link by considering user topological importance and interest aggregation. A time-dependent function is designed which takes node importance and aggregated interest as the parameters in adjacent graph to quantify the patience of nodes. The importance of a node is calculated based on betweenness centrality, and the aggregated interest of a node for a content is estimated based on the demand for the content by the node and its neighbours. An equation is given to calculate the probability of each node to add the selection of offloading nodes. When a node requests for a content, downloading through cellular network or opportunistic network does not begin immediately. Rather the node first decides whether to download the content through cellular network or through opportunistic network according to this probability equation. If the node downloads the content through cellular network, it becomes an offloading node to infect other interested nodes in an epidemic manner.

Rebecchi *et al.* [50] propose an offloading scheme based on reinforcement learning by combining LTE multicast and D2D communication, aiming at reducing redundant traffic while guaranteeing timely delivery. The scheme is similar to [44] and [45]. The control center selects a subset of all subscriber nodes as offloading nodes based on the channel quality indicator (CQI) in a greedy manner. After the offloading nodes receive the content through multicast, they epidemically infect other nodes. Nodes that fail to receive the content will

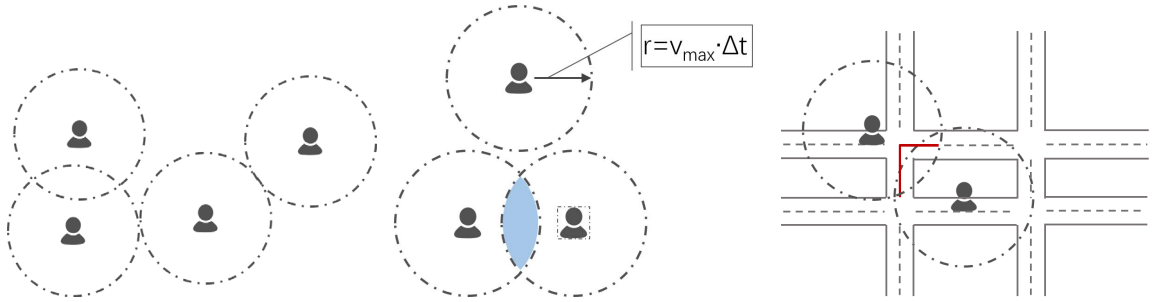


Fig. 10. The scenarios considered in TOMP: (a) Static coverage where all the nodes are static; (b) Free coverage where all the nodes are free to move in any direction, and TOMP calculates the contact probability of two nodes based on the current location and moving speed; and (c) Graph coverage where the real road graph are taken into consideration to predict the contact probability of nodes.

download it through unicast from the cellular network. The work [52] also proposes a scheme based on reinforcement learning. The mechanism of [52] is similar to [44] and [48], except for the algorithm that decides the number of offloading nodes. The central dissemination controller in the scheme of [52] determines the initial number of offloading nodes through reinforcement learning. Two reinforcement learning algorithms, actor-critic algorithm and Q-learning algorithm, are evaluated in [52]. After the offloading nodes downloads the content from the cellular network, the dissemination controller will evaluate the performance of infection at fixed time steps and determine whether to reinject the content to the network based on the available time. The work [49] proposes HYPE, an analogous mechanism with [40], to offload overloaded traffic with heuristic solutions, which applies an adaptive algorithm to achieve the optimal tradeoff between multiple conflicting objectives. Two common features of [28], [40], [44], [45], [48]–[50], and [52] are worth noticing. The first one is that all these schemes adopt epidemic approach to share the content, and the second one is that all of them require the subscriber node to upload feedback information to the control center.

Rebecchi *et al.* [58] define two different types of offloading nodes, leecher and seeder. Both leecher and seeder receive content through the cellular network. Seeder is normal offloading node but leecher cannot forward the content to other nodes. When the offloading performance of D2D transmission is lower than expected, the control originator will promote leecher into seeder to accelerate the D2D transmission. Pontryagin's maximum principle [75] is applied to select leecher and seeder. The scheme is similar to Push-and-Track [40], but it adopts a very different reinjection strategy. The scheme of [58] selects the new offloading nodes in advance, and the content is injected to these nodes at the beginning of offloading process, while the scheme of [40] starts to select the new offloading nodes when it decides to reinject.

3) Discussion: The performance of an offloading scheme depends largely on the selection of offloading nodes. The number of offloading nodes should be small in order to reduce the load on cellular network. But a too small number of offloading nodes will put an unfair burden on the offloading nodes and may result in low offloading efficiency. Therefore, the number of offloading nodes is a tradeoff between these two conflicting requirements. What type of information used to select offloading node is also important to the achievable offloading performance. In the literature, many researches focus on utilizing the encounter pattern between mobile nodes and other metrics, such as energy, fairness and etc. On the other hand, adopting one-hop or multi-hop to delivery the content from the offloading nodes also seriously impacts on the achievable performance of an offloading scheme. Using one-hop is too conservative, as it cannot exploit the full potential of opportunistic communications. Using multi-hop, while better exploiting opportunistic communications, may be unrealistic, as it relies on the assumption that nodes are willing to act as relays. Many researches study human behaviors and investigate how to encouraging mobile nodes, i.e., human beings, to participate in forwarding content for other nodes. This direction of research has great potential to enhancing offloading performance.

B. Offloading With Fixed Offloading Node

Similar to WiFi AP, special fixed devices, which can download and store the content from the Internet and then transmit delay tolerant data to mobile devices passing by, can be deployed to alleviate the overloaded cellular network. These fixed devices can be regarded as fixed offloading nodes. Fixed offloading node plays the role of 'bridge' between cellular network and end users, and it generally has three functions: short range communication, e.g., via Bluetooth or WiFi direct, caching and connecting to cellular network [65]. Note that, these works on offloading with fixed offloading node is different from content offloading (i.e., caching), although they are based on the same principle. The work of caching is to cache some popular content to BS in order to avoid the direct access to mobile core network. From the view of mobile devices, a change of next hop is not strictly required in caching. In contrast, in the works focusing on offloading with fixed offloading node, not only who to download these popular content, but also how to deliver them to requesting users are considered. Moreover, opportunistic network is not strictly required in caching. Most of the works on offloading with fixed offloading node consider vehicular ad hoc networks (VANETs). Table II summarizes the existing literature for this category.

TABLE II
LITERATURE SUMMARY OF TRAFFIC OFFLOADING WITH FIXED OFFLOADING NODE

Ref.	Hop	Required information	Offloading node	Methodology	Objective
[64]	Multi-hops	Qos attributes	AP	Mixed-integer programming	Maximize the amount of offloaded traffic
[65]	One-hop	Trajectory	RN	Time-prediction	Minimize the use of cellular network
[66]	One-hop, two-hops	Multi criteria	RSU	FOSAA	Maximize the content through VANET
[67]	One-hop, multi-hops	Mobility prediction	RSU	Fog-of-War	Maximize the amount of offloaded traffic
[68]	One-hop	Mobility Repetitiveness	Ship	Model-checking	Reduce the cost of communication
[69]	Multi-hops	Video quality	Relay	SSIM	Offloading low bit H.264 streaming
[70]	One-hop	Mobility predication	AP	SMV-BV, SMV-GP	Maximize the amount of offloaded traffic
[71]	One-hop, multi-hops	Mobility predication	RSU	Fog-of-War	Maximize the amount of offloaded traffic
[72]	One-hop	N/A	Station	Linear programming	Offload bulk data
[73]	One-hop	Mobility Repetitiveness	AP	Scripted handoff	Maximize the amount of offloaded traffic
[74]	One-hop	Mobility pattern	Cloudlet	Distributed Caching	Maximize the amount of offloaded traffic

1) Offloading to VANETs: RSU that has dedicate short range communication (DSRC) interface and buffer can connect to BS and play the role of offloading node in VANET. When a vehicle on a road with RSUs deployed sends a request for some content (e.g., a video clip) through cellular, a RSU can prefetch the requested content and waits for the vehicle. When the requesting vehicle is passing by the RSU, the requested content is transmitted to the vehicle [70]. The key issues for offloading to VANET are therefore which content RSU should prefetch from the Internet, which RSU should prefetch the content and how to schedule the transmission between RSU and vehicles. The work [73] proposes to predict the movement of vehicles to enable prefetching the content at appropriate RSU. The daily mobility of vehicles has certain regularity [76], [77], particularly for bus and private car travelling to/from work. But not all vehicles tend to drive in regular pattern, e.g., taxi. With the assistance of intelligent transportation system (ITS), however, the movement of vehicles can be predicted accurately and the requested content can be prefetched to the appropriate RSU which will encounter the requesting vehicular user [70]. More specifically, when a vehicular user sends a request to the server, other route information, such as the destination, the traffic condition and history trajectory, will be sent to the central controller of the ITS to predict which RSU will encounter the requester and the requested content can be prefetched to this RSU.

It is expensive to install RSUs. The work [65] proposes to replace RSUs with cheaper relay nodes (RNs) positioned at fixed locations. Like RSU, RN has DSRC and storage capacities, but unlike RSU, it does not have Internet connectivity. RNs can act as offloading points in VANET. When a vehicular user requests for a content, both the vehicle trajectory and the request are sent to the control center through cellular network. The control center decides which RN is most appropriate to place the content to. When the vehicle meets this RN, the content is transmitted to the vehicle via DSRC. Lee *et al.* [65] present the DOVE algorithm to select the optimal RN that overlapping the trajectory of the requesting vehicle. Since RNs do not connect to the Internet, it is necessary to prefetch the requested content to the offloading RN, but the work [65] does not discuss this important issue. We point out that the content can be offloaded to RNs via cellular link. If as part of the infrastructure, RNs are also connected to the backhaul of the cellular network, the delivery of content to a RN can naturally be done through the backhaul.

For a fast moving vehicle, the contact duration with RSU or RN is too short to complete the transmission of bulk data in a single contact. There are three approaches to enable offloading massive data. The first one is to extend the contact time, and the second one is to reduce the data size, while the third one is to increase the transfer rate. The third approach is related to physical-layer transmission techniques and is beyond our scope. We will discuss the second approach in the next section. Regarding the first approach, Baron *et al.* [72] propose a massive-data transmission framework that turns the electric vehicle charging stations into offloading nodes and the electric vehicles into data carrier. Subsequently, they demonstrate the feasibility of this framework in French road network.

The aforementioned researches only consider RSU/RN to vehicular subscriber communication, namely, one-hop offloading. Malandrino *et al.* [67], [71] jointly consider RSU-to-vehicle transmission and vehicle-to-vehicle (V2V) transferring to enable multi-hop offloading. They propose to predict the mobility of vehicles with different degrees of uncertainty through a fog-of-war model and study the impact of the inaccurate prediction on offloading performance. They further propose two approaches to schedule the vehicular relay selection between the RSU and the end vehicular subscriber. By taking into account the link qualities of both RSU-to-vehicle and V2V communication phases, Zhioua *et al.* [66] propose a model, called flow offloading selection and active time assignment (FOSAA), to analyze the capacity of vehicular network for offloading. The results of [66] indicate that the data size, density of vehicles and hop count between vehicles influence the achievable offloading performance considerably.

Wang *et al.* [64] also investigate multi-hop offloading in VANETs with both fixed offloading nodes and mobile offloading nodes. The authors design an offloading model to calculate the offloading capacities of both RSUs and vehicles through a connectivity graph. In the proposed offloading system, vehicles periodically upload their location and velocity information to RSUs to build the connectivity graph. The weight of an edge in the graph represents the link quality between the two connected nodes. The graph is used to determine which node, RSU or vehicular node, will be selected as the offloading node. The offloading problem is formulated

as a multi-objective optimization problem, considering the heterogeneity of vehicles and the global QoS guarantee.

2) Offloading to Other Systems: Access to the Internet from maritime ship mainly relies on satellites. Communications via satellite network have some serious disadvantages, including limited capacity, high delay and high cost. Mu *et al.* [78] propose a hybrid maritime communication framework that can be used to seamlessly transmit data between different networks. Mu and Prinz [68] further propose to leverage the repetition and the predictability of ship route to opportunistically transmit delay tolerant data which is originally transmitted through satellite network. Ship's onboard gateway, playing the role of offloading node, can opportunistically communicate with the network on the shore. Here, we further envisage the future global oceanic ad hoc network (OANET), where each ship is a 'fixed' offloading node, while sailors and passengers onboard are 'mobile' users. The requested content can be offloaded from the shore or satellite to ship and/or from ship to ship, to reach the end mobile subscriber.

The dream of the 'Internet above the clouds' [79] has fueled the research in the aeronautical ad hoc network (AANET) [80] for supporting direct communication and data relaying among aircraft for airborne Internet access. There appears no work to date on offloading to AANET. This is because the current physical-layer transmission techniques are incapable of providing the high throughput and high bandwidth efficiency communications among aircraft required for this airborne Internet access application. Even the planned future aeronautical communication system, called the L-band digital aeronautical communications system (L-DACS) [81], [82], only offers an air-to-air mode [83] that is capable of providing 273 kbps net user rate for direct aircraft-to-aircraft communication, which cannot meet the high throughput demand of the Internet above the clouds. However, a very recent study [84] has developed a very high throughput and high bandwidth efficiency physical-layer transmission technique for aeronautical communications. With this enabling physical-layer infrastructure in place, we envisage the following future global AANET, in which each jumbo jet is a 'fixed' offloading node, while passengers are mobile users. The requested content can be offloaded from airport to jumbo jet and/or from jumbo jet to jumbo jet, to reach the end mobile subscriber user.

3) Discussion: A RSU must prefetch and hold the relevant content so that when the interested vehicular users are passing by, the content can be directly delivered. The key issue in offloading to VANET is which RSU should prefetch which content and when to prefetch it. The prediction of the vehicular movement is particularly important to address this challenge. History mobility, location, velocity and trajectory of vehicle can all be utilized to serve the prediction. Furthermore, multi-hop delivery relying on both RSU-to-vehicle and V2V opportunistic communications can enhance offloading performance. Except for the special case of [72] where offloading nodes are electric vehicle charging stations and mobile users are electric vehicles parked at charging stations, it is difficult to offload bulk data, because contact during is too short owing to fast moving vehicles. Thus, dividing the bulk data into small fragments may be a good choice.

Most of the works in offloading with fixed offloading node focus on VANETs. In this survey, we envisage the OANET and AANET, and propose to extend the research to offloading to OANETs and to AANETs.

C. Offloading Without Offloading Node Selection

This category of offloading schemes do not select offloading nodes. Rather, mobile nodes collaborate to improve offloading performance. In effect, every node can be an offloading node. Table III summarizes the existing literature for this category. The existing works mainly focus on improving offloading mechanism, and they may be divided into five classes: energy-efficient (EE) offloading, bulk data offloading, peer-to-peer (P2P) offloading, adaptive offloading and incentive mechanism.

1) Energy-Efficient Offloading: Most works assume that all mobile nodes are wiling to exchange content with each other and participate in offloading, which is overly optimistic. It consumes the power for a mobile device to transmit the content to others, and selfish nodes that have the content may refuse to deliver the content to other nodes in order to save their battery power. Selfishness has a great impact on the dissemination of content between mobile nodes [113]. Some works [98], [104] study the impact of selfishness, in terms of energy consumption, in the dissemination of content through opportunistic network, and design a duty-cycling strategy, which allows a node repeatedly switches on and off its radio interface to reduce the energy consumption. The work [96] study investigate the energy consumption problem in periodic contact probing and propose a wakeup scheduling technique to prolong the lifetime of mobile devices. Coordinated multipoint (CoMP) processing can be applied to improve the energy efficiency in wireless communications. Wen *et al.* [114] propose a stochastic predictive control algorithm to obtain the requested content through an optimal BS group. Actually, receiving content also consumes power. Wu *et al.* [115] investigate the joint transmitter and receiver energy efficiency maximization problem and propose a joint optimization algorithm based on Dinkelbach transmission to iteratively solve the problem.

Let us further consider the following scenario, where a group of mobile nodes are interested in a number of contents. When a node sends a request to the Internet server for a certain content through cellular link, the requested content will not be transmitted to the node immediately via cellular link. The node can first obtain the content from other nodes that have the content through opportunistic communications before the content deadline. Only if the node cannot obtain the content before the deadline expires, cellular download will begin. There are two types of nodes in this scenario, data-seeking node that is requesting for one or more contents and data-fulfilled node that has obtained all the requested content. From a pure selfishness consideration, to save energy, the data-fulfilled nodes will switch off their radio interfaces. Consequently, the data-seeking nodes are unlikely to obtain the requested content before the deadline. In order to enhance offloading performance while saving energy, Kouyoumdjieva and Karlsson [91] propose an energy-aware

TABLE III
LITERATURE SUMMARY OF TRAFFIC OFFLOADING WITHOUT OFFLOADING NODE SELECTION

Ref.	Hop	Required information	Mechanism	Solution	Objective
[10]	Multi-hops	Energy consumption	Bulk data offloading	Cool-SHARE	Minimize energy consumption
[85]	One-hop	Battery level	Bulk data offloading	N/A	Minimize required cellular channels
[59]	Multi-hops	Potential, delay tolerance	Incentive mechanism	Reverse auction	Minimize the cost of incentive
[11]	Multi-hop	User interest		Smartparcel	Minimize the traffic over cellular network
[86]	Multi-hops	N/A	Content offloading	Proactive caching	Maximize the amount of offloaded traffic
[87]	One-hop	Node and content profile			Maximize the amount of offloaded traffic
[88]	One-hop	N/A	P2P offloading	Bearer control	Improve offloading efficiency
[89]	One-hop	Data fragmentation	Bulk data offloading	Device-centric cooperation	Reduce overhead and delay
[90]	One-hop	Data fragmentation	Bulk data offloading	Overhearing, network coding	Save cellular bandwidth, improve QoS
[91]	One-hop	Energy-aware	EE offloading	Progressive selfishness	Save energy, maintain throughput
[92]	One-hop	N/A		PPP	Reduce interference, improve spacial reuse
[93]	Multi-hops	Delivery possibility	P2P offloading	Hybrid transmission	Assure delivery
[94]	Multi-hops	Coupon and delay	Incentive mechanism	Reverse auction	Guarantee timely delivery
[95]	Multi-hops	Stackelberg game	Incentive mechanism	Stackelberg game	Reduce the traffic over cellular network
[96]	Multi-hops	Energy consumption	EE offloading	Wakeup scheduling	Reduce energy consumption
[97]	Multi-hops	Pricing scheme	Incentive mechanism	Bidding contest	Minimize network resource usage
[98]	Multi-hops	Energy consumption	EE offloading	Duty-cycling	Reduce energy consumption
[99]	Multi-hops	Data fragmentation	Bulk data offloading	Random Linear coding	Offload bulk data
[100]	Multi-hops	Query history, feedback		MObicache	Maximize operator's interest
[101]	Multi-hops	Selfishness	Incentive mechanism	Network formation game	Evaluate the impact of selfishness
[102]	Multi-hops	Coding scheme			Minimize the usage of cellular network
[103]	Multi-hops	Energy consumption	Bulk data offloading	WeCMC	Maximize energy efficiency
[104]	Multi-hops	Selfishness	EE offloading	Duty-cycling	Save energy
[105]	Multi-hops	Query history, location			Reduce cellular network load
[106]	Multi-hops	Data fragmentation	Bulk data offloading	Music stream service	Offload real-time content
[107]	Multi-hops	N/A		MIRCO	Maintain integrity and reputation
[108]	Multi-hops	Delivery possibility	P2P offloading	Probabilistic framework	Improve delivery possibility
[109]	Multi-hops	User's satisfactory	Incentive mechanism	Contract	Maximize operator's interest
[110]	Multi-hops	User's satisfactory	Incentive mechanism	Contract	Maximize operator's interest
[111]	Multi-hops	N/A	Adaptive offloading	AOM	Get requested content in the fastest manner
[112]	Multi-hops	Energy consumption	Bulk data offloading	N/A	Assure the fairness on energy consumption

algorithm called progressive selfishness, which requires a data-fulfilled node to periodically switch on and off its radio interface. The experiment results of [91] show that the progressive selfishness can save 85% of energy for mobile devices while only reducing the throughput by 1%, compared to the case that all the data-fulfilled nodes must always switch on their radio interfaces.

In addition, some efforts focus on reducing the energy consumption in traffic offloading in heterogeneous cellular networks (HCNs) while preserving the quality of service experienced by users. In these works, small cells are applied to offload the cellular traffic from macro cells. Small cells refer to low-power and short-range access points that can be applied to transmitting contents in a flexible and economical way [116], [117]. Strictly speaking, small cell traffic offloading is not based on opportunistic network. Hence, we give a brief introduction on this area. Interested readers can refer to [19] for further research.

The energy consumption of a small cell is dependent on the system load, which is the average utilization level of radio resources in terms of time and frequency domains [118]. Small cells are either activated for offloading cellular traffic or deactivated for saving energy. Hence, without elaborate design, directly offloading cellular traffic to small cells not only may not reduce the energy consumption of the whole network, but also may deteriorate the congestion of the current cellular network. Saker *et al.* [119] study the implementation of sleep/wake up mechanism in small cells based on the traffic load and user location within small cells with the objective of minimize the energy consumption of the overall network. Chiang and Liao [120] prove that the switch-on and switch-off strategy of a small cell is monotone hysteretic, and then realize the offloading scheme by simple switch-on and switch-off thresholds. These aforementioned works assume that small cells are independent, that is the coupling interference across different cells is not considered. The coupling interference, resulting from the sharing of a common spectrum resource, has a great impact on the throughput of the whole network. Taking the coupling interference into consideration, Chen *et al.* [19] model the energy efficiency problem as a discrete-time Markov decision process and use Q-learning with compact state representation algorithm to make offloading strategies. The offloading strategy is a sequence of actions, consisting of switch-on and switch-off operations on each small cell. Then, they prove the convergence of the algorithm from the points of both theory and practice. Nevertheless, the learning efficiency and the mixing characteristics of the underlying Markov chain are not analyzed in [19].

2) Bulk Data Offloading: With the popularity of multimedia content, the volume of content we like to share is getting larger and larger, which becomes a big challenge to opportunistic offloading: the content cannot be entirely transmitted over single contact between mobile nodes because the contact duration is too short. As mentioned previously, a solution to this problem is to divide the bulk data into small fragments.

Li *et al.* [99] propose a contact-duration-aware cellular traffic offloading scheme, called Coff, which partitions the bulk content into several small segments that can be transmitted in one contact. These segments are encoded with random

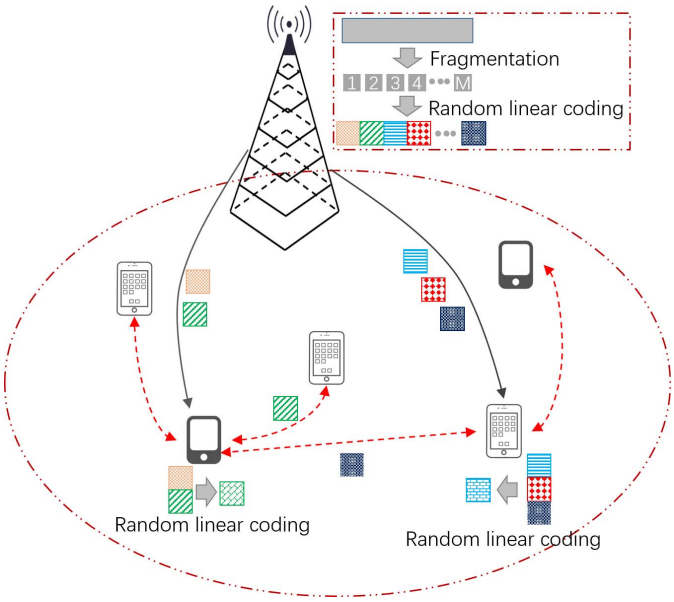


Fig. 11. Illustration of Coff. The bulk data is fragmented into several small segments at the content server. Subscriber nodes download these segments through cellular network and exchange the segments that they obtain though opportunistic communications. The bulk data is recovered at the subscriber nodes through random linear coding when enough segments are collected.

linear networking coding technique at the content provider, and each coded segment contains certain information about the content, so that the original bulk data can be recovered from the coded fragments. The segments are sent to subscribe nodes through cellular network. Subscribe nodes exchange these segments they have received through opportunistic D2D communications. When a subscriber node get all these segments, it recovers the original bulk content. The process is illustrated in Fig. 11. By the deadline, subscriber nodes that have not received all segments can download the segments that they do not have through cellular network. The work [99] also designs a greedy algorithm to optimally allocate load on each subscriber node. Seferoglu and Xing [89] consider the scenario where a group of mobile nodes are interested in a same video content. These nodes cooperatively download the content using both cellular network and opportunistic network. Specifically, each node downloads a segment of the video content through cellular network, and then multicasts the segment it has to the other nodes in the group through opportunistic network. Seferoglu and Xing [89] design a device-centric cooperation scheme for the mobile nodes to decide which segment should each node download. Both [89] and [99] are based on the same idea but they adopt different allocation strategies.

Kouyoumdjieva and Karlsson [106] design a framework to offload bulk multimedia with real-time requirement in urban environment. In this framework, every node maintains a play list sequentially recording the requested contents, and its cache is initially empty. The requesting orders of the play lists for different nodes may be different, owing to the heterogeneous preference of mobile nodes. Each node downloads the first content in its play list through cellular link and places the content in its cache. When two nodes encounter and if they have each other's requested contents, they download the requested contents from each other's caches through opportunistic communications. The results of [106] indicate that the performance of this scheme is sensitive to the density of mobile nodes and the number of requests. Le *et al.* [90] describe the cooperative downloading problem as a network utility maximization problem. Different from [89], Le *et al.* [90] use overhearing and network coding in local WiFi transmission between mobile nodes. Subsequently, they introduce MicroCast, a modular system implemented on Android platform. Their experiment shows that the scheme can significantly enhance offloading performance while imposing a little more battery consumption on each mobile node.

Chang *et al.* [103] design a collaborative mobile cloud (CMC), composed of a certain number of mobile devices that are interested in the same big content. The big content is divided into several fragments as usual. In CMC, a small subset of the mobile devices are selected to separately download a certain fragment from the BS through cellular link. Then these mobile devices can exchange what they have received to form a complete content as well as transmit their fragments to other mobile nodes via D2D manner. It can be seen that unlike the schemes of [89], [90], and [106] most mobile users can receive the complete content without downloading a fragment from BS, but CMC is also an effective way of offloading bulk data. Note that CMC selects a small subset mobile devices to download the fragments of bulk data via cellular link and these devices are responsible to transmit their downloaded fragments to other mobile devices. Therefore, these mobile devices consume much more energy [85], [112], which is clearly unfair. The emerging simultaneous wireless information and power transfer (SWIP) technique can help solving this problem [103]. SWIPT enables a mobile node to charge its battery using the power of the wireless signal transmitted by BS. Chang *et al.* [103] propose the wireless power transfer enabled CMC (WeCMC) to allocate the sub-channels for the transmission between BS and WeCMC. Interested readers can refer to [121]–[124] for further exploration.

In some situations, the contact time is sufficiently long and large data can be delivered in single contact. Ashton and Zhang [10] consider this case, and they design an app called Cool-SHARE to seamlessly share bulk data, e.g., apps, multimedia data. A Cool-SHARE installed on the smart phone can download apps from app store through cellular network. Then it can transmit the apps to other smart phones with Cool-SHARE installed. The authors specifically consider two scenarios: social sharing scenario and opportunistic sharing scenario. The former occurs between acquaintances, e.g., at workplace, while the later occurs between strangers, e.g., on the bus.

3) P2P Offloading: Mayer and Waldhorst [93] propose to offload the traffic generated by a pair of communicating end nodes with the assurance of 100% delivery. When a mobile device sends a content to the other mobile device, the content is transferred in opportunistic network with router strategy initially. Specifically, there is no replicates of the content in the opportunistic network, and the content is sent from the source device to the end device via unicast communication. Not all the devices are connected to the infrastructure. Upon

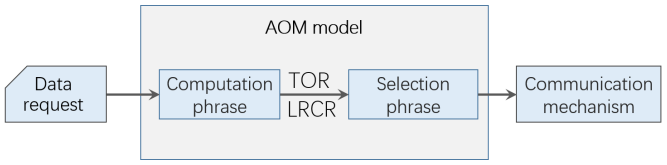


Fig. 12. Two phases of AOM: in computation phase, AOM calculate TOR and LRCR, while in selection phase, it decides whether to adopt cellular network or opportunistic network.

receiving the content, a node must decide whether to transmit the content through opportunistic network or through infrastructure network based on the information obtained from other nodes via opportunistic communication. When the probability of delivery through opportunistic network is lower than a certain threshold, nodes prefer to transmit the content to nodes that can connect to the infrastructure to guarantee the delivery. The idea is to use the opportunistic network to transmit the content as much as possible to alleviate the load of infrastructure-based network, while guaranteeing the delivery. Yang *et al.* [88] consider the scenario where a pair of mobile users connected to the same eNodeB communicate through cellular network. When a pair of mobile nodes connected to the same eNodeB are communicating, the gateway will detect it and informs them. Then they will perform a discovery process to find each other. When they find that they are in proximity to each other, the cellular communication will turn into D2D communication between them. Thus, the traffic generated by the communication between these two mobile users is offloaded.

4) Adaptive Offloading: Hsu *et al.* [111] introduce an adaptive offloading model (AOM) to adaptively switch between cellular network and opportunistic network. The main advantage of downloading directly from the cellular network is that subscribed users can obtain the requested content in a fastest way, while the main advantage of obtaining the requested content through opportunistic network is that subscriber users can obtain the requested content in a cost-effective way at the expense of certain delay. AOM combines the two approaches to adaptively improve the offloading efficiency. There are two phrases in AOM, as illustrated in Fig. 12. When subscribe nodes request for content, AOM calculates traffic offloading rate (TOR) and local resource consumption rate (LRCR) in the first phrase. In the second phrase, AOM decides which mechanism should be adopted by comparing the TOR and LRCR with their given thresholds. If the TOR and LRCR are smaller than their respectively thresholds, the subscriber node should download the content directly through cellular network. Otherwise, the content provider injects the requested content to offloading nodes, and let offloading nodes to propagate the content to subscriber nodes before the deadline.

5) Incentive Mechanism: Opportunistic network can only provide an intermittent connectivity between mobile users. Receiving the content via opportunistic network will inevitably cause certain delay, and not all mobile users are willing to be served in this way. On the other hand, not all mobile users are willing to forward the content for other users [125], because acting as relay will consume their battery energy

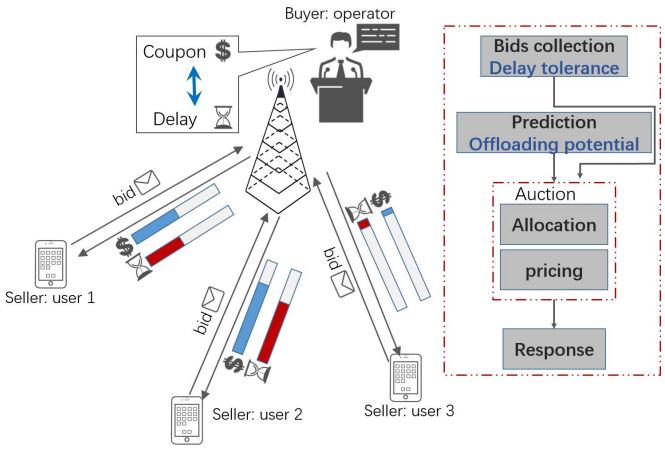


Fig. 13. Incentive scheme based on cellular operator acting as the buyer and mobile users acting as sellers. Each user sends a bid, involving the delay that it can tolerate and the discount that it wants to get. The operator determines who is the winner of the auction that will get the required discount.

and take up their storage space. Wang *et al.* [101] evaluate the impact of selfishness of mobile nodes on opportunistic offloading efficiency. In their evaluation, the offloading scheme is user-centric, and all mobile nodes are 'rational': they are only concerned with their own interest. Each user can decide whether to download the requested content or not, and when mobile nodes have downloaded the content, they are free to decide whether to forward the content to others or not. The authors use a game theory to describe the selfishness of mobile nodes, and the evaluation result indicates that the offloading efficiency is significantly degraded, compared with the ideal offloading where all mobile users are selfless.

It is then making sense to provide incentives for mobile users to encourage them to participate in offloading. For example, if a mobile user is willing to wait for a certain time before receiving its requested content, the cellular operator will give a discount for the service charge. Zhuo *et al.* [59] design an incentive mechanism, named Win-Coupon, to encourage mobile users to offload cellular traffic by leveraging their tolerance for delay and their potential for offloading. The incentive mechanism is based on reverse auction, where mobile users act as sellers and the cellular operator acts as a buyer. The process of auction is shown in Fig. 13. Each mobile user sends a quotation to the operator, including the delay that it is willing to wait and the discount that it wants. The cellular operator takes into account both the delay tolerance and the offloading potential to determine the auction outcome. Specifically, mobile users with higher tolerance to delay should be given less discount if the delay is the same. Similarly, mobile users with higher potential for offloading should undertake more offloading tasks by giving large discount, if the delay is the same. The winner of the auction will receive the requested content from opportunistic network with the contracted delay and coupons. Other users will receive the requested content directly from the cellular network with the original charges. By adopting stochastic analysis, Zhuo *et al.* [59] propose a model based on the data access and mobility pattern of mobile users, to predict the offloading potential of mobile users. In [94],

TABLE IV
LITERATURE SUMMARY OF UPLOAD OFFLOADING

Reference	Hop	Offloading node	Required information	Objective
[41]	Multi-hops	None	Past contact pattern	Minimize the cost of upload
[126]	Multi-hops	None	Congestion degree of cellular network	Reduce peak traffic level
[127]	Two-hops	Vehicle	Time-varying connectivity graph	Maximize the offloaded traffic at peak hour
[128]	Two-hops	Vehicle	Time-varying connectivity graph	Maximize the offloaded traffic at peak hour
[129]	One-hop	AP	Bandwidth allocation	Reduce energy consumption
[130]	One-hop	AP	Deadline	Maximize the amount of offloaded traffic
[131]	Multi-hops	AP	Estimate of traffic of other vehicles	Maximize the amount of offloaded traffic
[132]	Multi-hops	None	Congestion degree of cellular network	Reduce peak traffic level
[12]	One-hop, two-hops	GCB	N/A	Maximize the amount of offloaded traffic
[133]	Multi-hops	None	Service quality	Save energy and reduce delay
[134]	One-hop	AP	Bandwidth allocation	Reduce energy consumption
[135]	One-hop	AP	Bandwidth allocation	Reduce energy consumption
[136]	One-hop	AP	Data need	Improve energy efficiency
[137]	Multi-hops	AP	Mobility and connection prediction	Reduce overall cost
[138]	One-hop	AP	SINR	Maximize minimum energy efficiency

they present an extended work to [59], including the consideration of WiFi case. Li *et al.* [109], [110] present a contract based incentive mechanism to encourage mobile users utilizing their delay tolerance and price sensitivity for offloading. The authors describe the cellular traffic offloading process as a monopoly market, where the cellular operator is the monopolist who signs the contracts with mobile users according to the statistical user satisfaction. To capture the satisfaction of different users, the authors divide the users into different classes according to their delay tolerance and price sensitivity. Each user selects a quality-price contract according to its class to maximize its utility. Different from [94], the works [109], [110] take into consideration not only the delay sensitivity but also the price sensitivity to depict the QoS.

Different from [59], Sugiyama *et al.* [95] propose an incentive mechanism based on reward to encourage mobile nodes to forward the content for other mobile nodes. The cellular operator announces the total reward to be shared among the mobile users that use their surplus resources to forward the content for other mobile nodes. The authors describe the problem as a Stackelberg game, in which each mobile node must select to forward the content for other mobile nodes through opportunistic communication or not to, by balancing the cost and reward. However, the work [95] does not consider the fact that user satisfaction will be affected by long delay.

6) Discussion: Researches in the category of offloading without offloading node selection mainly focus on improving the offloading structure, since structural innovation plays a crucial role in enhancing offloading efficiency. Some works design offloading schemes in a decentralized manner, but these schemes impose considerable overhead for collecting necessary control information. Performing offloading in a centralized manner may be a better option, because the cellular operator already has most of the essential information. Many works design multi-hop offloading schemes but these schemes face a practical challenge – not all mobile users are willing to forward the content for others. Incentive mechanisms may offer effective means for tackling this problem. Surprisingly, the individual privacy has not been involved in any work so far. In real-world scenarios where mobile nodes may not trust each other, the individual privacy must be taken into account, and there is a big scope to study effective trust mechanism in unfamiliar environments, e.g., node authentication.

D. Upload Offloading

As can be seen from the previous three subsections, many existing works concentrate on download offloading. However, with the big change of our habits in the digital world, we become data creators and generate ever-increasingly large amount of data to upload. Consequently, upload offloading is attracting more and more attentions.

Applications like Facebook, Qzone, Instagram and Youtube enable us to upload our data, e.g., text, mp3, photos and videos, to share with our friends at the time of creation. We tend to upload big files like photos and video clips to social networks conveniently through mobile applications, which transforms us into data creators [129]. On the other hand, with the development of mobile cloud computing, many applications on mobile computing have emerged recently. When we perform these applications, we need to upload data to the cloud sever [130]. Vehicles also need to frequently upload data to the cloud side [131]. Furthermore, when vehicles are travelling at a high speed on highway, they generate large amount of information, e.g., traffic condition, vehicle condition, etc, which need to be uploaded to the control center of ITS to guarantee the road safety and improve the road efficiency. These uploads put a big burden on the cellular network. In addition, in cellular network, such as LTE, uploading consumes nearly 8 times more energy than downloading [139]. Table IV summarizes the existing literature for upload offloading.

1) Offloading With Fixed Offloading Node: Considering the fact that large number of high-speed WiFi APs and cellular femtocells are widely deployed, the most simple approach of upload offloading is to directly transfer the data created by mobile users to available WiFi APs and femtocells [140]. The empirical research in [139] indicates that the energy consumption of WiFi is lower than third generation (3G) and LTE cellular networks' uplink. Furthermore, inexpensive WiFi APs are simple to deploy. WiFi APs and femtocells used for upload offloading can be regarded as fixed offloading nodes, which adopts certain caching strategies to get data from other nodes

and upload the data to the Internet. Mobile nodes rely on their mobility to create opportunities of meeting these fixed offloading nodes. When a mobile node has data to upload, it can store the data and carry the data with its movement. When the mobile node meets an offloading node, the data can be transmitted to the offloading node. Alternatively, when the mobile node meets a relay node, it can transfer the data to the relay node who also adopts the store-carry-forward mechanism to help transferring the data to an offloading node.

Because of the aforementioned advantages, WiFi AP strategies have been widely investigated for upload offloading. Sethakaset *et al.* [138] derive a closed-form expression to calculate the number of users who should forward their data to the AP in a given area with one LTE macrocell and one WiFi AP, in order to maximize the minimum energy efficiency. Specifically, the N_w users with the worst signal-to-interference-plus-noise radio (SINR) values in the macrocell should upload their data via the WiFi AP. Gao *et al.* [130] focus on tackling the deadline-sensitive data offloading problem which needs to schedule uploading data items between WiFi AP and cellular network. Taking into account the heterogeneity of data items in terms of size and TTL, the corresponding optimization problem is NP-hard. Two greedy-based algorithms, offline data offloading (OFDO) and online data offloading (ONDO), are presented in [130] to solve this optimization problem. The authors prove that the expected total sizes of data items offloaded to WiFi AP achieved by these two algorithms are no less than half of the expected total sizes attained by the corresponding optimal solutions.

The aforementioned two works only exploit opportunistic communications between mobile nodes and offloading nodes, i.e., they are one-hop schemes. Komnios *et al.* [137] propose cost-effective multi-mode offloading (CEMMO), a mechanism that allows peer node to play the role of relay between source node and AP. Three communication modes are allowed in CEMMO: cellular delivery, delay tolerant delivery and peer-assisted delivery. CEMMO selects the most effective upload offloading mode through the prediction of user mobility and connectivity with WiFi AP. Considering the big volume of data created by vehicles, WiFi AP based upload offloading is a promising solution. However, due to the limited contact time between high-speed vehicle and WiFi AP, the cached data may not be all delivered to the AP during one communication. Kolios *et al.* [131] study this problem and propose a V2V assisted offloading scheme to accelerate upload offloading by exploiting the cooperation between vehicles, as illustrated in Fig. 14. The scheme allows the communication between vehicles to balance the caches among vehicles. When encountering an AP, each vehicle with just enough data can transmit its cached data to the AP in one-go.

As discussed previously, a large number of vehicles constantly create a great deal of information and transmit these information to the control center of ITS. These floating car data (FCD) are extremely valuable in analysing the road traffic conditions and maintaining an efficient ITS. Similar to mobile network, vehicle network can also utilize the V2V mode to offload the upload traffic. Stanica *et al.* [127] propose to offload the upload traffic of FCD with the assistance of V2V

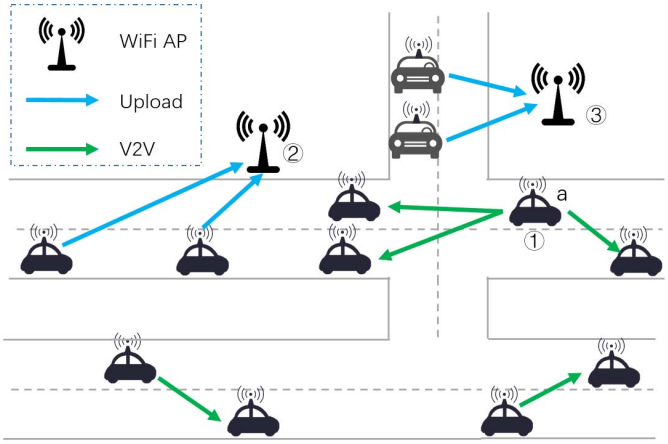


Fig. 14. Vehicle-to-vehicle assisted upload offloading: in ① vehicle a transmits part of its data to other vehicles nearby to balance its cache, while in ② and ③, vehicles with just enough data transmit the data to AP.

opportunistic communications. Fig. 16 contrasts this FCD uploading assisted by V2V opportunistic communications with the traditional FCD uploading approach. By studying the fundamental properties of V2V connectivity, the authors design three heuristic algorithms to select the subset of vehicles, which are responsible for collection, fusion and uploading of the data efficiently. The work [128] studies the best-case and worst-case performance of this FCD offloading approach.

WiFi AP is not the only equipment that can act as fixed offloading node. Han and Ansari [12] introduce a green content broker (GCB), powered by green energy, for acting as the content brokerage to delivery content between the content requester node and the content owner node. The authors formulate the optimization problem of maximizing the amount of offloaded traffic, subject to the amount of green energy available and other constraints. They proposed a heuristic traffic offloading algorithm to find near optimal solution.

Some works consider the offloading strategies based on IP flow mobility (IFOM), which is a technology used in fourth generation (4G) network allowing a user equipment (UE) to maintain two data streams concurrently, one through WiFi AP and the other through LTE [141]. Consider the scenario that a moving UE is uploading a file. When the UE moves into the coverage of a WiFi AP, the file uploading may be shifted to the WiFi network. When the UE moves outside the coverage of the WiFi AP, the file uploading may be seamlessly shifted back to the cellular network. Another example is that the UE maintains a uploading flow and the flow can be divided into two sub-flows, which can be serviced through different access technologies. A question naturally aroused is how the UEs fairly offload part of their data to the WiFi network, or in other words, how to allocate the bandwidth. Carrier-sense multiple access with collision avoidance (CSMA/CA) is applied to fairly share the radio resource in 802.11 Distributed Coordinated Function, which treats all UEs equally. However, there are different data needs for UEs and the conditions of connection with eNodeB for different UEs are different.

Miliotis *et al.* [134]–[136] consider the scenario where N UEs are under the coverage of an eNodeB, and at the

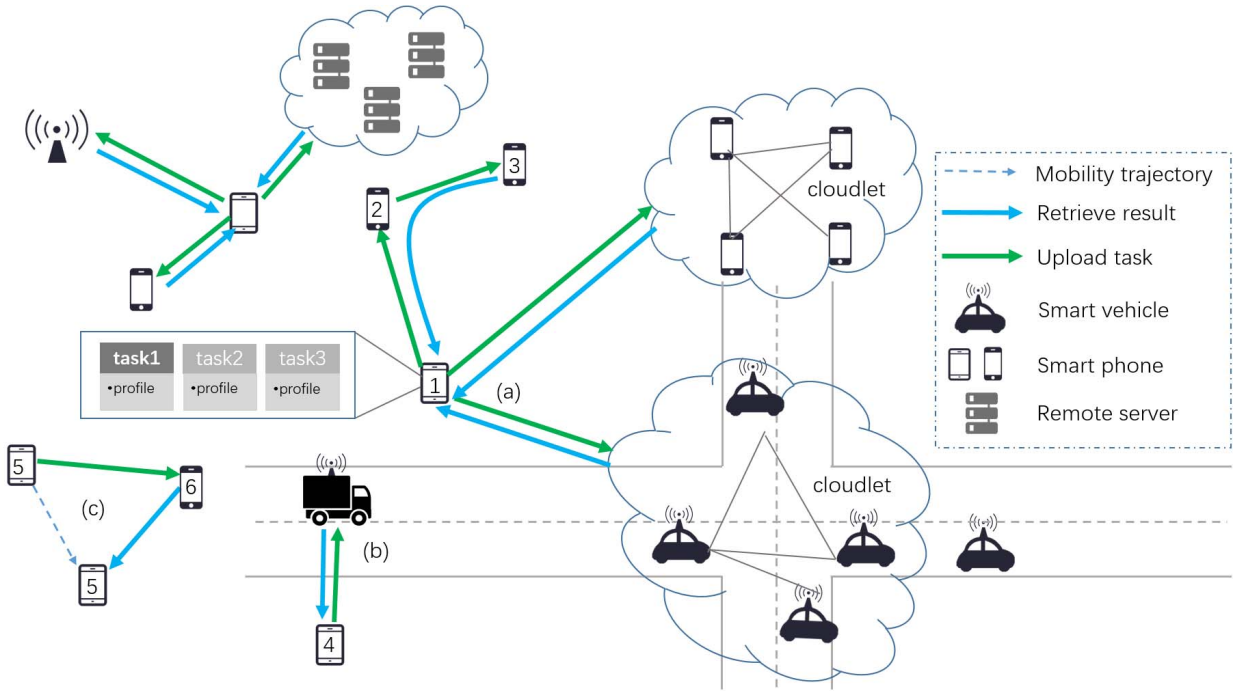
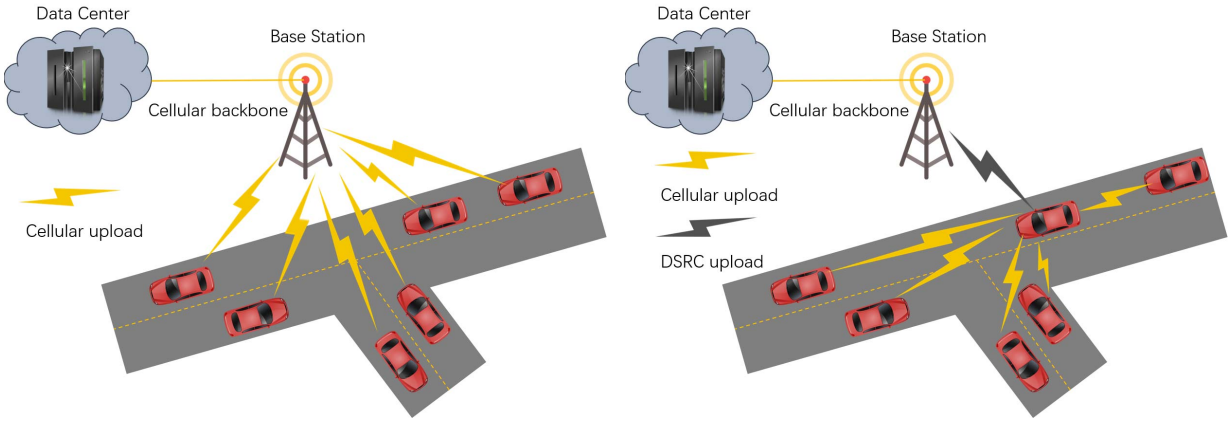


Fig. 15. Some application scenarios of computation offloading: in scenario (a), client node 1 may offload its computation subtasks to offloading nodes 2 and 3, or mobile cloudlet, in scenario (b), smart vehicle acts as offloading node for client node 4, and in scenario (c), client node 5 relies on offloading node 6 for helping its computation task.



(a) FCD upload in traditional approach

(b) FCD upload in opportunistic offloading approach

Fig. 16. Contrast of (a) the FCD uploading in tradition approach, where each vehicle uploads its own data through cellular network, and (b) the FCD uploading with the assistance of V2V opportunistic communications, where a specified vehicle collects the data of other vehicles through opportunistic communications and uploads the collected data through cellular network.

same time they are under the coverage of a WiFi AP. Miliotis *et al.* [136] propose two upload offloading algorithms based IFOM, the first one giving priority to the UEs with high volume of data while the other one giving proportional priority to each UEs. However, only the data needs of UEs are considered in [136]. Subsequently, in [134], Miliotis *et al.* further propose a weighted proportionally fair bandwidth (PFB) algorithm, which not only considers the data needs of UEs but also takes LTE spectrum efficiency into consideration. Miliotis *et al.* [135] evaluate the achievable performance of the PFB algorithm, in terms of energy efficiency and throughput, in the presence of malicious UEs. They also present an approach based on reputation to fight against the malicious UEs. The work [129] provides enhancement to both WiFi access and LTE uplink, aiming to improve the energy efficiency and offloading volume. For the WiFi access, the authors present a scheme to maximize the utilization of WiFi resource, while for the LTE uplink part, they present two pricing algorithms, linear and exponential.

2) Offloading Without Offloading Node Selection: The deployment of large number of WiFi APs is not the best cost-effective way of mitigating the congestion and enhancing the capacity of cellular network. This is because congestion only occurs during the peak time, and cellular network has enough capacity to deal with the data traffic at other times [126]. Off the peak time, the installed WiFi APs are seriously under

utilization, resulting in huge waste of resource. Therefore, some researches also study the strategies of offloading the upload traffic without the assistance of offloading node, e.g., WiFi AP.

Izumikawa and Katto [132] design robust cellular network (RoCNet) to enhance the cellular network in conjunction with opportunistic network. When the congestion of cellular network is detected, UEs may choose the store-carry-forward mode to transmit the data to other less loaded BS. Existing techniques for available bandwidth estimation can be used to estimate the degree of cellular network congestion [142]. Hu and Cao [133] present a quality-aware traffic offloading (QATO) framework to offload the upload traffic in order to save energy consumption and to reduce delay. Obviously, the service quality for different nodes in an given area may be different, and transmitting data with poor QoS will consume more energy [143]. Therefore, when a node with poor QoS has data to upload, it may choose to transmit the data to neighboring nodes with good QoS and to rely on them to upload its data. Hu and Cao [133] further evaluate the performance of QATO on Android phones in a real experiment. The result obtained demonstrates that 70% energy consumption and 88% delay can be reduced in upload offloading. The framework can also be applied to download offloading.

Thilakarathna *et al.* [41] propose MobiTribe, a distributed storage system composed of mobile devices to store user generated content (UGC) and thus to avoid the need for uploading UGC. when a mobile user decides to share its UGC, the application running on the mobile device sends a registration information to the control center who allocates a replication group to the user. The replication of the UGC is opportunistically transmitted to the group through P2P mode. When another user requests for the UGC, the control center orders a device in the replication group to deliver a replication to this user. The authors verify the feasibility of MobiTribe by implementing MobiTribe on Android mobile phones with the integration of Facebook [144]. The content replication problem in MobiTribe is NP-hard [145]. An algorithm is represented in [41] to significantly reduce the amount of replications whilst hardly reducing the availability of content. MobiTribe can also be used in download offloading.

3) Discussion: Unlike the download offloading research, where large volume of works are available, there are fewer works in the upload offloading research. This is because download traffic used to dominate the total traffic. However, as we become big data creators and generate huge amount of data to upload, designing effective upload offloading schemes to alleviate the congestion of cellular uplink become vitally important too. Because of encountering uncertainty between mobile users and fixed offloading node, accurate prediction of users' future movement based on the past data and the knowledge of deployed offloading nodes is critical to achieve high offloading efficiency in the approaches of upload offloading with fixed offloading node. For the schemes of upload offloading without offloading node selection, how to select appropriate relays is the key to success. We observe that in the current research, the selfishness of mobile users and the privacy of user data are hardly taken into account. This is an important research area, requiring researchers to put big efforts in.

IV. Computation Offloading

Although our smart phones are getting increasingly powerful, the capacity of single device, in terms of battery, CPU and memory, is still limited, particularly in comparison with the computation requirements of big applications demanded by us. Cloud computing, e.g., Google AppEngine and Amazon EC2, can effectively solve this problem [146]. Users can upload their computing tasks consisting of programs and data to the remote cloud through cellular network. When these tasks are completed, the results are sent back to the user device again via cellular network. However, cloud computing is impractical or cost-ineffective when mobile devices cannot connect with the Internet or the mobile traffic is expensive. Moreover, uploading tasks and downloading results through cellular network create large volume of traffic, which puts a big additional burden on the already overloaded cellular network. In addition, the latency of uploading and downloading data through cellular network may be large, particularly in peak time.

On the other hand, because almost everyone has a mobile phone, there are always a large number of mobile devices with idle resources in the vicinity. Thus, if a capacity limited mobile device offloads its computing tasks to other mobile devices nearby through opportunistic network, the aforementioned problem will be effectively solved [3]. Not only the computation capacity of individual mobile device is enhanced but also aggravating the overburdened cellular network is avoided. The computation offloading process involves three parts: task upload, task execution and result retrieval. Mobile node that offloads computing tasks to other nodes is called client node, and the node that performs computing tasks for other nodes is called offloading node. Client node needs to upload computing tasks, involving data and program, to a subset of offloading nodes through opportunistic communication. Offloading nodes execute the tasks with their spare computing resource. When the tasks are completed, client node retrieves the results from offloading nodes through opportunistic communications. A big computation task can be partitioned into several subtasks, which can be performed in parallel or sequence according to the task structure. Subtasks that can be executed in parallel can be allocated to different offloading nodes to reduce overall completion time. Fig. 15 illustrates some application scenarios of computation offloading. According to the realizing approaches, computation offloading schemes can be classified into two categories: offloading with offloading node selection and without offloading node selection.

A. Computation Offloading With Offloading Node Selection

Table V summarizes the existing literature. Existing works mainly focus on how to select a subset of offloading nodes to allocate tasks in order to achieve different objectives, e.g., minimizing the energy consumption, minimizing the completion time, maximizing device lifetime, etc. We further divide the existing works into three parts based on single-objective, multi-objectives and incentive mechanism, respectively.

TABLE V
LITERATURE SUMMARY OF COMPUTATION OFFLOADING WITH OFFLOADING NODE SELECTION

Reference	Task division	Objective
Yang et al. [147]	No consideration	Balance workload of mobile users
Banerjee et al. [148]	No consideration	Minimize completion time
Liu et al. [149]	No consideration	Save energy
Chatzopoulos et al. [150]	No consideration	Motivate user to participate in offloading
Xu et al. [151]	No consideration	Maximize the interest of operator
Li et al. [152]	No consideration	Balance workload of mobile users
Chatzopoulos et al. [153]	No consideration	motivate user to participate in offloading
Zhou et al. [154]	No consideration	Provide virtual computing cloud with mobile devices
Khaledi et al. [155]	No consideration	Reducing the overall job completion time
Li et al. [156]	No consideration	Balance workload of mobile users
Lu et al. [157]	No consideration	Minimize the average task response time
Li et al. [158]	No consideration	Balance workload of mobile users
Ghasemi-Falavarjani et al. [159]	Parallel	Minimize task completion time and maximize the device lifetime
Mtibaa et al. [160]	Parallel	Save execution time and consumed energy
Shi et al. [161]	Parallel	Leverage mobile devices to offload computing tasks
Ghasemi-Falavarjani et al. [162]	Parallel	Minimize the energy consumption, task completion time and satisfy the deadline
Trono et al. [163]	Parallel	Minimize individual computational loads
Trono et al. [164]	Parallel	Minimize individual computational loads
Fahim et al. [165]	Parallel	Save execution time and consumed energy
Xiao et al. [166]	Parallel	Minimize the average makespan of all tasks
Xiao et al. [167]	Parallel	Minimize the average makespan of all tasks
Alanezi et al. [168]	Parallel	Minimize the overall cost, energy consumption and task execution time
Flores et al. [169]	Parallel	Motivate user to participate in offloading
Mtibaa et al. [170]	Parallel	Save execution time and consumed energy
Chatzopoulos et al. [171]	Parallel	Minimize the completion time
Shi et al. [172]	Parallel, sequence	Improve the performance of computationally complex jobs
Chen et al. [173]	Parallel	Reduce energy consumption
Chen et al. [174], [175]	Parallel	Minimize energy consumption and tolerate fault
Gao [176]	Parallel	Save overall energy consumption

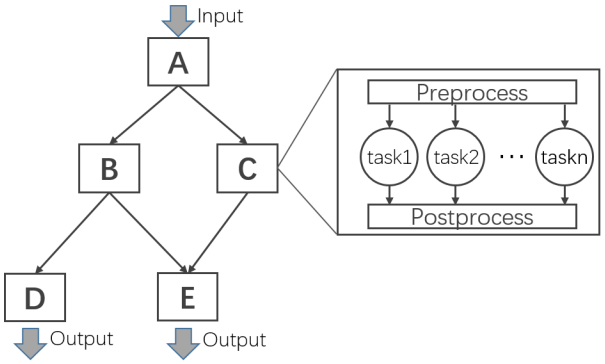


Fig. 17. The working job is described by a direct acyclic graph. The vertices in the graph are PNP-blocks, and each vertex or PNP-block involve three parts: preprocess, n subtasks that can be performed in parallel, and postprocess.

1) Single-Objective: These works select offloading nodes to achieve a single objective, such as minimizing the completion time of tasks or minimizing average energy consumption of all devices or prolonging the lifetime of devices.

Shi *et al.* [172] propose Serendipity, which enables mobile nodes to offload their computing tasks to other mobile nodes through opportunistic network. In this architecture, the computing task to be offloaded consists of several PNP-blocks, and each PNP-block contains three parts: pre-process, n tasks that can be executed in parallel, and post-process, hence the name PNP. Pre-process and post-process can only be executed by client node, while the n tasks are executed in other mobile nodes in parallel. The authors use a directed acyclic graph (DAG) to describe this model, as shown in Fig. 17. Furthermore, in Serendipity, each mobile node maintains a profile that describes the available capacity, i.e., execution speed and energy consumption model. The profile is used to determine whether to allocate the task to the node encountered. In other words, offloading nodes are selected based on the profile. Every task is given a TTL. If client node does not receive the result of the task before TTL, the task is discarded and it is performed locally. Serendipity assumes that the workloads of all tasks are the same, which is clearly a limitation. Shi *et al.* [172] design a water filling algorithm to greedily select offloading nodes with expected minimum completion time to minimize the completion time of the whole task. They further propose Cirrus Cloud, a computation paradigm, to leverage both mobile devices and other available computation resource to offload computing tasks in [161]. The architecture can also be used to deal with video compression problem in practice, as carried out by Chatzopoulos *et al.* [171].

Xiao *et al.* [166] propose an offloading scheme with the focus on average makespan of tasks based on node mobility model in mobile social network. Different from [172], the workloads of all tasks are not assumed to be the same. The authors design two greedy algorithms, offline task assignment (OFTA) and online task assignment (ONTA), to select the offloading nodes. Tasks are sorted in ascending order according to their workloads. The basic idea of OFTA is to select the node who has the smallest expected processing time for finishing the tasks already assigned to it, and to assign the minimum-workload task among the tasks not yet been assigned to this node. By contrast, ONTA makes

the selecting decision for each encountered node. Specifically, when the client node meets a node, it computes the instant processing time of the node, namely, the expected time for this node to return the result, and the expected processing times of other nodes that the client has not met. Similar to OFTA, it always assigns the task with the minimum workload to the node who has the minimum instant processing time or expected processing time. The authors extend the work in [167] by designing largest makespan sensitive online task assignment (LOTA), a greedy algorithm to select the offloading nodes for collaborative tasks. Unlike ONTA which adopts the principle of small-task-first-assignment and earliest-idle-user-receive-task, LOTA prefers large-task-first-assignment and earliest-idle-user-receive-task. A similar work is introduced in [157] by considering data transmission time, processing time and queueing time of tasks. The work [157] selects the offloading nodes to minimize the response time of task under both centralized and distributed settings. In the former setting, the information of all tasks are known in advance, and the offloading node selection becomes an integer linear programming (ILP) problem which can be solved by an offline centralized algorithm. In the later setting, the information of future tasks cannot be handled in advance, and the authors propose to solve the problem with an online distributed algorithm.

Mtibaa *et al.* [170] design a mobile device cloud (MDC) platform consisting of Android mobile devices with application client installed. In MDC, mobile devices communicate with each other through opportunistic network, and a client device divides a task into several subtasks to be executed on other mobile devices. The authors create four types of social graph based on contact history as well as four kinds of information: friendship, interest, combination of friendship and interest, and encountering history between mobile nodes. They propose four algorithms based on the four types of social graphs, respectively, to select the subset of offloading nodes. The goal of MDC is to reduce the execution time of the task and to reduce the energy consumption on mobile devices. Subsequently, the benefit of MDC offloading, in terms of execution time and energy consumption, is assessed in [160]. A similar work is presented in [165].

Considering the heterogeneity of mobile devices' capacity and the diversity of tasks, the tasks should be flexibly allocated to speed up the execution process of tasks and to balance the energy consumption of mobile devices. If the tasks are not appropriated distributed, it will lead to abnormal energy consumption of individual device and low execution speed of the whole job. The works [152], [156], and [158] study extensively load balance in task offloading. Li and Yang [156] propose a computation offloading and task reassignment scheme based on 'ball and bin' theory to balance the workload of mobile devices in mobile social network. When a client node needs to allocate a task to other mobile nodes, it first randomly selects d mobile nodes within communication range. The task is then allocated to the least loaded one among the d nodes. They evaluate the scheme with random walk model, and the result shows that with $d = 2$, the performance of the task balancing is the best. Subsequently, they evaluate the scheme with real data. However, the result indicates that the random 'd-choice' may not lead to well-balanced allocations in real-data, since relatively stable social relationship between mobile nodes leads to the imbalance in assigning tasks. It is seen that social relationship has serious impact on the assignment of tasks. Li *et al.* [158] propose a task assignment algorithm, called iTop-k, based on social relationship to appropriately offload the computing tasks. The iTop-k algorithm selects the top k friends to assign the tasks based on the contact pattern. Compared to the random selection in [156], selecting intimate friends to offload tasks leads to a smaller delay while maintaining the load balance. Using real-trace data, the work [147] validates that the performance of iTop-k is better than that of random 'd-choice'. Unlike [156] and [158], which do not consider the selfishness of mobile nodes, the work [152] is based on the realistic assumption that all mobile nodes are selfish. The authors propose Chance-Choice, a load balancing scheme, to obtain Nash equilibrium among selfish nodes.

Liu *et al.* [149] propose an adaptive method for selecting offloading nodes that can automatically switch between two selecting modes to save energy. The two selecting modes are: centralized selection through cellular network and broadcast selection through opportunistic network. In the first mode, a node first checks whether it has enough resource. If not, it sends a request to the central controller through cellular network. The central controller returns the ID of the node that has the required resource to be the offloading node to the requesting node. In the second mode, a node also first check whether it has enough resource. If not, it broads a request to all the nodes in its communication range. Other nodes that received the request first check whether they have the requested resource. If a node has, it sends its ID to the requesting node, and this node can be the offloading node. If not, the node broadcasts the request to other nodes in its communication range. Thus, the request is spreading through opportunistic network, and the requesting node will receive a definite reply. The central controller collects the statistic information from nodes, reporting their experience during each time slot, at the end of the time slot. The central controller then estimates the energy consumptions of the two modes, respectively, according to the statistic information, and decides which mode each node should use in the next time slot.

Chen *et al.* [173] propose a resource allocation scheme based on the k-out-of-n theory, a famous theory in reliability control. The idea is to allocate the data fragments to n service center nodes in the network but just need to access k out of the n nodes to retrieve the data, so as to reduce the energy consumption. Based on this idea, Chen *et al.* [174], [175] propose a k-out-of-n computation offloading scheme that considers both data storage and data processing. The data generated by applications are encoded and divided into several segments, and these segments are allocated to n nodes in the network. When a client node needs the data, it can download the segments from the nearest k nodes through opportunistic network and recover the data locally. When a client node needs to process the data, the nearest k nodes can process the segments of the data in parallel for the client, and the client node can download the result from these k nodes through opportunistic

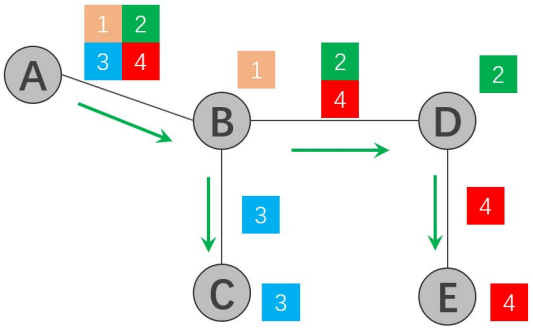


Fig. 18. The task allocation in NCG. Node A offloads the task to Node B, who has higher computing capability. When node B encounters nodes C and D, it offloads part of the task to them, because they have higher computing capacities. Part of the task is also offloaded by node D to node E for the same reason. The result will be retrieved back when the relevant nodes re-contact.

network. This approach can minimize the overall energy consumption in the network. The authors formulate the problem as an ILP. A similar idea is also proposed in [177].

Gao [176] proposes to offload the computing task to mobile nodes with high capacity to save the overall energy consumption. The idea is based on the assumption that it consumes less energy for the mobile device with higher computation capacity to perform the same task. The author also takes into consideration the completion time of a task. A network contact graph (NCG) is built based on the contact history to characterize the opportunistic contact among mobile nodes, where the value of the edge between two nodes in the graph is determined by the inter contact time (ICT) distribution. Thus the value of an edge indicates the probability of re-contact between two nodes, and is used to decide whether to offload the task to a node, to guarantee the timely retrieval of the task result. Moreover, the task can be divided into several subtasks that can be executed in parallel. When the client node encounters a mobile node with a higher computation capacity, the task is transmitted to it, and the node becomes sub-client node. A sub-client node can offload subtasks to other mobile nodes with higher computation capacities. This offloading process is illustrated in Fig. 18.

Some works investigate map generation in disaster area. The cloud service is generally inaccessible in post disaster environment, and DTN can be used to transmit data and to generate map. With limited computing capability and resource, a single mobile device cannot complete the map generation task. Trono *et al.* [164] introduce a distributed computing system, called DTN MapEx, to deal with this problem while minimizing the individual workloads of mobile devices. The devices of rescue workers and volunteers act as sensing nodes. They record the map data of the area that they are passing through with the aid of GPS and transmit the map data and the map integration task to a computing node, which can be pre-deployed in the area after disaster, through DTN. A computing node is an offloading node in this case. When the task is completed, the result is routed back to the DTN. To balance the workload, a computing node periodically broadcasts its load information to the DTN, and sensing nodes can select a proper computing node to offload based on the load information. Trono *et al.* [163] evaluate this system through experiment and simulation. The results indicate that the system can effectively reduce the processing time.

2) Multi-Objectives: These works focus on selecting offloading nodes to simultaneously achieve multiple objectives, e.g., minimizing the completion time and maximizing device lifetime, minimizing overall cost and energy consumption, etc. Three relevant works are compared in Table VI in detail.

Ghasemi-Falavarjani *et al.* [159] design a multi-criteria based optimal fair multi-criteria resource allocation (OFMRA) algorithm to select offloading nodes and to allocate resource, based on the assumption that all the subtasks require the same amount of computation. The goal of OFMRA is to minimize the task completion time and to maximize the lifetime of devices simultaneously. However, in practice, the subtasks are heterogeneous in terms of computational requirements.

With the rapid development of communication, context-aware communication have been applied in the area of communication offloading. Context is the information that characterize the situation of an entity, or a cluster of entities [180], [181]. The entity infers to mobile device in our paper. Context information can be categorized into four classes, device context, user context, network context, service context. Device context refers to the information that describes device profiles, e.g., location, mobility, computing resources, etc. User context is the information that describes users, e.g., gender, age, relationships, etc. Network context is the network condition, e.g., link quality, bandwidth, interference, etc. Service context characterizes network service, e.g., service mode, service rate. Some works incorporate context information into offloading system to improve the system performance.

Ghasemi-Falavarjani *et al.* [162] design and implementation a context-aware middleware, named OMMC, to collect context information (e.g., device context, network context and service context) and manage the offloading process based on proposed offloading nodes selection algorithm. More specifically, the authors first adopts the non-dominated sorting genetic algorithm II (NSGA-II) to locate the Pareto solution set. Then entropy weighting and a technique for order preference by similarity to ideal solution (TOPSIS) method are employed to specify a best compromise solution, which determines the subset of optimal offloading nodes. The design aims to simultaneously optimize three objectives: minimizing the task completion time, minimizing the overall energy consumption of mobile devices and meeting the deadline. Sigg *et al.* [168] consider a more complex and practical scenario, where cloud, cloudlet and mobile devices. They propose Panorama, a context middleware, which is performed in mobile devices and collect various context information, to decide when and where to offload the computing tasks. The heterogeneity of subtasks is taken into account, and Panorama can select a optimal offloading mode to achieve a best tradeoff among different objectives.

3) Incentive Mechanism: None of the aforementioned works considers the selfishness of mobile users, which has serious impact on how computing tasks can be offloaded

TABLE VI
COMPARISON OF MULTI-OBJECTIVE BASED WORKS FOR COMPUTATION OFFLOADING WITH OFFLOADING NODE SELECTION

Reference	Algorithm	Subtasks' loads	Objectives
Ghasemi-Falavarjani et al. [159]	OFMRA	Same load	Minimize the task completion time and maximize the lifetime of devices
Ghasemi-Falavarjani et al. [162]	TOPSIS	Different loads	Minimize the energy consumption, task completion time and satisfy the deadline
Alanezi et al. [168]	Panorama	Different loads	Minimize the overall cost, energy consumption and task execution time

through opportunistic network. Most mobile users are cautious about helping other users to performing computation tasks by consuming their precious resource. Additionally, selfish users would like to offload their computing tasks to others as many as possible while avoiding to help other users. To combat this selfishness, some rewards can be offered to motivate mobile devices to participate in offloading.

Chatzopoulos *et al.* [150] propose an intensive framework combining with a reputation mechanism. Mobile users that help others will receive a number of FlopCoin, a kind of virtual currency. When a client user wants to offload a task, it broadcasts a request to all neighboring users. A neighbor can calculate the number of FlopCoin as a bid based on the resource needed to performance the task. The client user selects a user as the offloading node to perform the task based on the bids by neighbors as well as their 'reliability' or reputation. Through a P2P reputation exchange scheme, each user can record the 'goodness' of other users to quantify their reputation, which indicate the selfishness degrees of users [153]. A similar incentive mechanism is proposed by Flores *et al.* [169] based on reputation and credit. When a user requires to offload a computing task to a offloading node, its credit decreases, and the reduced portion will be added to the offloading node's credit. After the completion of each offloading task, users receive the updated information about reputation and credits of the interacting peers. Xu *et al.* [151] propose an incentive mechanism to maximize the interest of the operator based on the 'less is more theory and considering distributed denial of service (DDoS) attacks.

Banerjee *et al.* [148] propose to select the offloading node by competitive bidding. The authors consider the scenario where the connections between mobile nodes are relatively stable. When a client node has a task to offload, it will invite all the nodes within the communication range to bid as well as broadcasts the deadline for the task. Nodes received the invitation can execute a pre-installed application to estimate the execution time of the task. If the estimated execution time is shorter than the deadline, the node can participate in bidding. Otherwise, it will not. The client node selects one of them as the offloading node according to certain criterion, such as the shortest execution time. The client node will give some reward to the offloading node. However, realistically, topology of opportunistic network is time-varying. To deal with node mobility, Khaledi *et al.* [155] propose to hold multi auctions over adaptive time intervals for selecting the offloading node. The authors adopt an additive increase and multiplicative decrease (AIMD) method to adaptively determine the time interval between auctions.

4) Discussion: The achievable performance of a computation offloading scheme largely depends on the selection of offload nodes. Mobile nodes with high computing capacities may be selected as offloading nodes to perform tasks for other nodes. However, the computing capacity alone may not be able to determine the best choice, and the mobility of nodes also affects the performance of an offloading scheme seriously. For example, a high computing-capacity offloading node may move out before the result can be retrieved, and the task assigned to it has to be performance again by another node. This causes high latency. Therefore, the computing capacity and mobility of nodes are two most important factors that need to be taken into account when selecting offloading nodes.

Mobile nodes with spare computing resource may not be willing to help others for free. Incentive plays a key role in overcoming this selfishness of mobile nodes. Security and privacy of data are always the most important considerations for any offloading application. 'Can I trust the other node to handle my computation task?' will and should be the first question that a client node asks. Again, little research has been done in this critical area.

B. Computation Offloading Without Offloading Node Selection

Again without offloading node selection simply means that there is no need to determine to whom to perform the computation tasks for clients. Table VII summarizes the existing literature for this category, where we have added three more references not indicated in Fig. 2. A group of mobile devices with idle computing resources decide to form a cluster or cloudlet for performing computation tasks through mutual cooperation. Nodes in such a cluster or cloudlet are all willing to collaboratively execute tasks for others, and thus there is no need to decide who should do the job. The task or client may come from inside the group or outside group. Therefore, this category can naturally be divided into two sub-categories.

1) Tasks From Inside: We start the discussion by considering the following scenario. A group of mobile devices adjacent to each other are running the same app. A single device does not have sufficient resource to perform the app. The implementation of the app can be divided into several small tasks which can be performed on different devices. After each small task is finished, the result is transmitted to other devices through short range communication. Compared to the traditional approach in which each device offloads the computing task to the remote cloud through cellular network, such a computation offloading approach has clear advantages.

Jin *et al.* [186] propose a penetration based mobile cloudlet based on computation offloading, PMC^2O for short, to offload tasks within clusters. In PMC^2O, some mobile nodes form an opportunistic cluster through discovery, and each cluster

TABLE VII
LITERATURE SUMMARY OF COMPUTATION OFFLOADING WITHOUT OFFLOADING NODE SELECTION

Reference	Algorithm	System	Execution	Objective
Hasan et al. [182]	N/A	Wearable cloud	Parallel	Enhance computing capability of wearable devices
Wu et al. [183]	FTS	Cluster	N/A	Allocate tasks at peak time
Shi et al. [184]	N/A	Wearable cloud	N/A	Minimize completion time
Liu et al. [185]	ALP	N/A	Parallel	Make use of idle resources
Jin et al. [186]	SA-UM	Cluster	Parallel	Optimize computation capacity of cluster
Zeng et al. [187]	RLNC	Mobile Cloud	Parallel	Minimize the expected completion time
Habak et al. [188]	N/A	Cluster	N/A	Provide virtual computing cloud with mobile devices
Li et al. [189]	DynPredict	Cluster	N/A	Minimize completion time
Mtibaa et al. [190]	Power balancing	Collaborative devices	Parallel	Prolong device lifetime
Zhang et al. [191], [192]	MDP	Cloudlet	Sequence	Minimize the cost of computation
Li et al. [193]	N/A	Cloudlet	Parallel	Decide whether to offload to cloudlet
Truong-Huu et al. [194]	MDP	Cloudlet	Parallel	Minimize the processing cost
Wang et al. [195]	TVG	Cloudlet	N/A	Measure the service level of MVC
Monfared et al. [196]	MHPC	Cloudlet	N/A	Schedule MHPC on demand
Panigrahi et al. [197]	EEOA	Cloudlet	N/A	Optimize the energy usage of cloudlet
Huerta-Canepa et al. [146]	N/A	Virtual cloud	Parallel	Enhance computing capability of mobile devices
Xiang et al. [198]	Merge and split	Virtual cloud	Parallel	Enhance computing capability of mobile devices
Chen et al. [199]	OSCC	Cloudlet	Parallel	Minimize cost and ensure QoE

selects a cluster head after the nodes in the cluster exchange information with each other. The head of the cluster is responsible for managing cluster member and task assignment. Nodes in a cluster are all willing to executing tasks for other nodes. When a node leaves or joins the cluster, the cluster head adjusts the resource and task assignment within the cluster. As illustrated in Fig. 19, the offloading process involves three levels, node, app and component. The components of an app can be performed on other mobile devices in the cluster. The simulated annealing based on user mobility (SA-UM) algorithm is used to optimize the component offloading within the cluster. To balance the workload, the cluster head periodically checks the average usage of resource. If it is higher than a given threshold, the head node deletes the node with lowest resource sharing degree from the cluster. Wu *et al.* [183] consider the scenario that the computation resource within the cluster is insufficient to offload all the tasks at peak time, where a central scheduler in the cluster responsible for task assignment plays the similar role to the cluster head in [186]. The authors design a friendship based algorithm to assign priority to nodes in the cluster by considering both the contribution of nodes and the friendship between nodes.

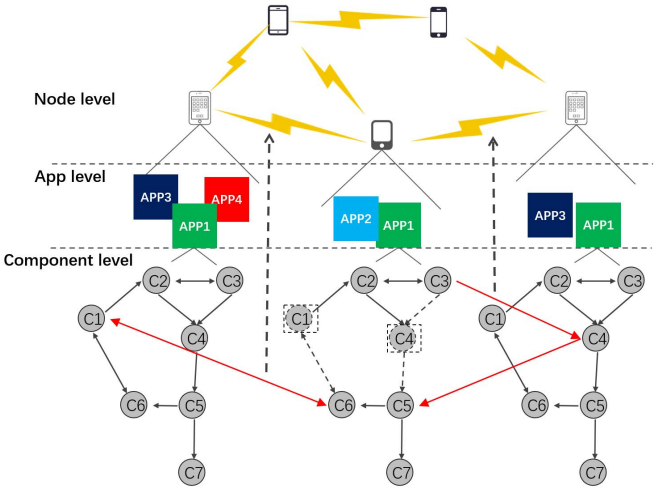


Fig. 19. The structure of PMC^{2}O involving three levels: node level, app level and component level. The components of app 1 are executed in different nodes in the cluster.

Zeng *et al.* [187] propose the intermittent mobile cloud (IMC), a computation offloading scheme based on opportunistic network. The IMC is composed of mobile devices, such as smart phone, tablet and vehicle, with idle computation resources. Mobile devices can transmit data to each other through opportunistic communication. All the devices in an IMC are assumed to be willing to execute the tasks for other devices. The computation offloading process involves three stages: uploading the data, executing the tasks and retrieving the results. The data to be uploaded is encoded with random linear network coding (RLNC) and partitioned into a number of fragments. The fragments are transmitted through the opportunistic network in an epidemic approach. When a service provider node with sufficient computation resource receives enough fragments, it can recover the data and process the data. The result is send to the requesting device in the same way. Liu *et al.* [185] propose an analogous computation offloading architecture. The client node partitions the whole task into several subtasks and transmits the subtasks to the service provider node encountered through opportunistic communication. The service provider node starts execute the subtasks. When the subtasks are finished, the client node will download the result from the service provider node when they contact again. The offloading process is shown in Fig. 20. It can be observed that there are two differences between [185] and [187]. First, the task in [185] is divided into several subtasks which are executed in parallel, and there exists only one copy of the whole task. Second, the data delivery between client and service provider adopts one-hop approach in [185].

Mtibaa *et al.* [190] design a computation offloading scheme for the scenario where mobile devices are highly cooperative, based on the goal of maximizing the lifetime of mobile devices in the offloading scheme through balancing energy consumption. The cooperative mobile devices may belong to one person or to a household. The authors propose an

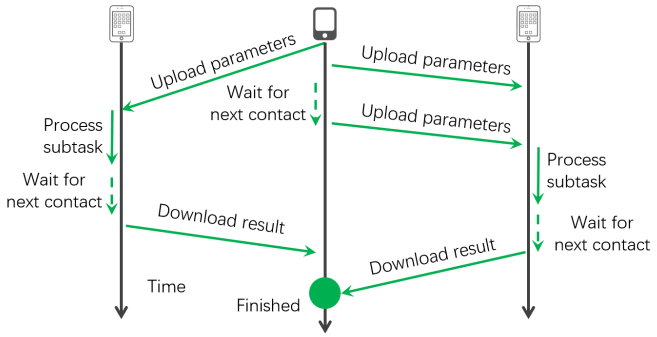


Fig. 20. The opportunistic data flow between the client node and service provider nodes.

energy consumption function based on the computation and data-transfer requirements, and they design an energy balancing algorithm to ensure the fairness of energy consumption in order to prolong the lifetime of the MDC. Hasan and Khan [182] propose a wearable cloud to offload the computing tasks for mobile and wearable devices. The wearable cloud consists of a number of wearable computing devices, e.g., Raspberry Pi. Shi *et al.* [184] offer several guidelines for enhancing the computing capability of wearable devices in a relative stable environment. The computation tasks can be offloaded to mobile devices of the same user to minimize the completing time and to save the power of wearable devices.

Huerta-Canepaand and Lee [146] propose a virtual computing cloud consisting of mobile devices to offload computing tasks by leveraging opportunistic network. Each mobile device in the group acts both as a server and a client. The tasks are uploaded to the virtual cloud through opportunistic network, and when the task is finished, the result is delivered to every device in the cloud through opportunistic network. In this offloading scheme, all the nodes are willing to offload computation tasks for others, without any incentive mechanism. Xiang *et al.* [198] study a similar problem with a coalition game theory. The authors consider the scenario that users are not all altruistic to exchange computation result with each other, and they design a merge and split algorithm to assign users with the same tasks to one coalition, in which users are collaborative to exchange computation result. Users in the same coalition perform the tasks for each other to avoid the use of remote cloud. Like [146], incentive mechanism is not used in [198].

2) Tasks From Outside: A group of mobile devices with idle computing resources may decide to form a cloud to perform the tasks for the devices outside the group who do no have enough computing resources. This is similar to traditional mobile cloud computing, and we may refer to this type of offloading as opportunistic cloud computing. Tasks to be offloaded are mainly from outside the cloud.

Some researches focus on constructing local cloud, consisting of co-located mobile devices with idle computing resources, to provide local cloud service based on opportunistic network. A local cloud can make full use of its idle resources to offload the computationally intensive task at a low price. Habak *et al.* [188] propose a femtocloud to provide local cloud service with some incentive. The control device in a femtocloud has complete information about other mobile devices in the cluster, e.g., leaving time and spare resource, and it is responsible for task assignment. The client deployed on a mobile device in the femtocloud can evaluate the node's computation capacity and maintains a profile based on the usage history, device sensors and input. The control device assigns tasks based on the profiles of the mobile devices in the femtocloud. More specifically, when a new task arrives, the task assignment and scheduling module on the control device will assign it to an appropriate node in the femtocloud based on the information collected by/provided from other modules. This architecture is shown in Fig. 22. In this kind of opportunistic cloud, however, some devices may be 'dishonest'. For example, some devices leave the cluster before the registered leaving time, and the task must be started again on other mobile devices. This problem exists in both the local cloud formed to serving its members and the local cloud formed to serving outside devices. Zhou *et al.* [154] propose a status-aware and stability-aware approach to solve this problem by considering the status and historical characteristics. For opportunistic cloud computing, a same idea is presented in [189]. The client deployed on a mobile device in the cluster will record the status and historical characteristic information and evaluates the stability of the device. The 'stability' here refers to the relationship between the real leaving time and the registered leaving time. The task allocation is based on the stability and computing capability information.

Cloudlet traditionally refers to a cluster formed by fixed devices deployed in the vicinity of mobile users, which offers rich computing resource and is connected to the Internet [200]. The concept of cloudlet can be extended to the case where a group of mobile devices, e.g., smart phone and smart vehicle, with idle computing resources willingly act as a cloudlet to help other mobile users. This cloudlet offloading precess is illustrated in Fig. 21. Client nodes transmit tasks to cloudlets nearby through opportunistic communication and retrieves the results when the job is done. In Table VIII, we survey the existing works on computation offloading with cloudlet.

Offloading computing tasks to cloudlet when accessible can save energy consumption while speeding up task completion. However, offloading may fail due to the mobility of cloudlet and/or client node. If the client node moves outside the communication range of the cloudlet before the task result is retrieved, the offloading fails, and the task must be re-executed locally. When a client node has a complex task, it must decide how to process the task: to offload it to a cloudlet, to use remote cloud, or to process it locally. Several works discuss whether to offload tasks to cloudlet from different perspectives [191]–[194]. Specifically, Zhang *et al.* [191], [192] propose a dynamical offloading algorithm based on Markov decision process (MDP) model to decide whether to offload the task to the cloudlet or to execute it locally, by taking into account both the task workload and the accessibility of cloudlets. The cloudlet considered consists of mobile devices with rich computing resource, and these mobile nodes are connect to each other with Bluetooth or WiFi. The task to be executed is divided into several phases. Client node checks the

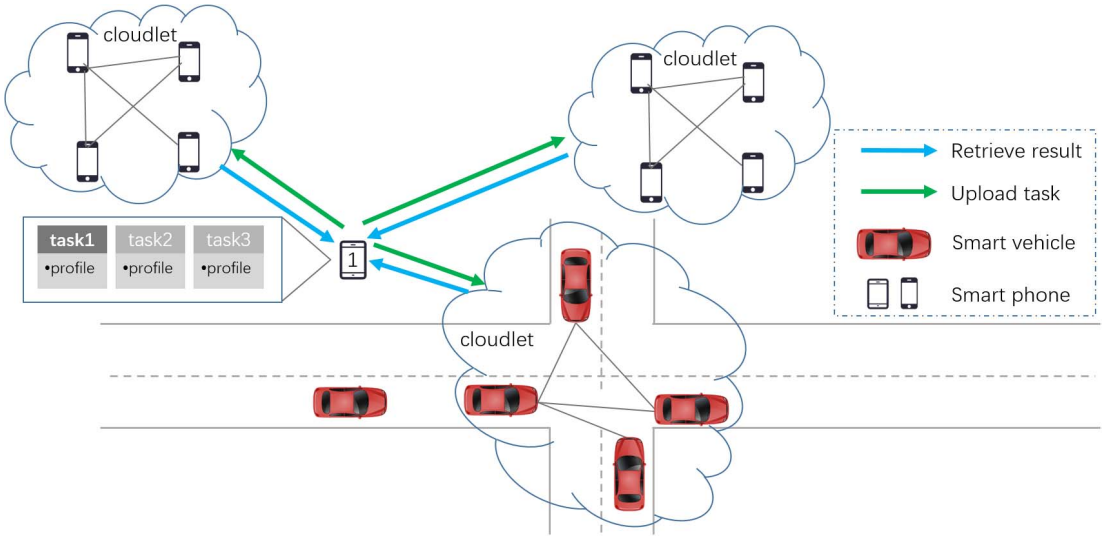


Fig. 21. Computation offloading with cloudlet: client node 1 offload its three subtasks in parallel to three cloudlets nearby formed by smart devices or vehicles with idle computing resources.

TABLE VIII
SURVEYING THE EXISTING WORKS ON COMPUTATION OFFLOADING WITH CLOUDLET

Reference	Cloudlet	Subtask	Main idea
Zhang et al. [191], [192]	Vehicular BS	Several phases of a task	Decide whether to offload to cloudlet based on cost
Li et al. [193]	Mobile devices	Subtasks can be processed in parallel	Decide whether to offload to cloudlet based on capacity
Truong-Huu et al. [194]	Mobile devices	A set of task can be executed in parallel	Decide whether to offload to cloudlet based on cost
Wang et al. [195]	Smart vehicles	No consideration	Measure the service level of MVC based on TVG
Monfared et al. [196]	MHPC	No consideration	Schedule MHPC on demand
Panigrahi et al. [197]	Mobile devices	No consideration	Optimize the energy usage of cloudlet
Chen et al. [199]	Mobile devices	Subtasks can be processed in parallel	Find the tradeoff between remote cloud and cloudlet

local state, in terms of the remaining workload, the task phase and the number of accessible mobile nodes in the cloudlet. Then the dynamical offloading algorithm is performed to make the decision to offload the task phase to the cloudlet or to process it locally. Offloading of the task phase may fail due to node mobility. If it is failed, the task phase will be restarted. A similar offloading decision scheme is presented in [194]. Li and Wang [193] make the offloading decision based on the computing capability of the cloudlet. The authors quantify the computing capability boundary through investigating the cloudlet attributes, i.e., cloudlet size, lifetime and reachable time. The task is divided into several subtasks that can be processed in parallel. For each subtask, if the required capacity is less than the lower boundary of the computing capability of the cloudlet, the subtask may be offloaded to the cloudlet. If the required capacity is larger than the upper boundary of computing capability of the cloudlet, the subtask should be offloaded to the remote cloud through cellular network. Otherwise, the subtask can be executed through either of the two approaches. An analogous work is also presented in [199], which finds the compromise between remote cloud and cloudlet to reduce energy consumption and delay.

Wang *et al.* [195] introduce a new notion of serviceability to measure the service level of mobile vehicular cloudlet (MVC) in large-scale urban environment. By considering the impact of connection and mobility of vehicular nodes over time on serviceability, the authors propose to describe the problem with

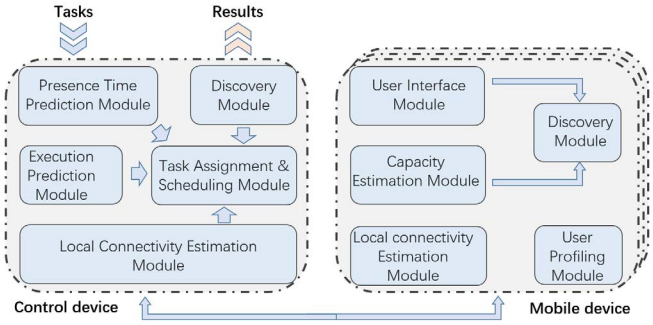


Fig. 22. The structure of femtocloud. The task assignment and scheduling module on control device appoint a proper node for each task based on the feedback information from other modules.

time-varying graph (TVG). An algorithm is designed to calculate serviceability. The authors evaluate serviceability through a real-world vehicular mobility trace, and the results obtained show that serviceability is related to the delay that a computing task can tolerate. Monfared *et al.* [196] propose to deploy powerful computing resource on vehicle to form a mobile high performance computer (MHPC) to ferry computation. The MHPC vehicle can move to the vicinity of mobile users that request for computation resource to offload tasks.

3) Discussion: A group of mobile devices with idle computing resources forming a cluster to help each other can

significantly enhance the computing capability of each device and reduce overall energy consumption. Since offloading is mutually beneficial within the cluster, selfishness of mobile users is less a problem in this offloading approach. The offloading performance largely depends on the stability of the cluster. Thus, user mobility and contact pattern should be accurately characterized. On the other hand, a group of smart devices with idle computing resources can form a cloudlet to act as the local cloud to serve other devices without enough computing capacities. Client user only needs to send tasks to the local cloud, without the need to know the task assignment within the cloud. However, the assumption that the devices forming the cloud are all willing to perform tasks for free is too optimistic. Some incentive mechanism should be employed to reward these devices in the local cloud. We are surprised to find that no work exists to address the security and privacy considerations of client nodes.

C. Computation Offloading With Wired Network

In this section, we discuss computation offloading frameworks that are based on wired network. Although, these frameworks are designed for traditional cloud computing, they can also be adapted to opportunistic environment. In other words, if these cloud servers are located at the edge of network, the computation offloading can be occurred in opportunistic manner. For example, MAUI, Cuckoo, CloneCloud,ULOOF, COSMOS, Odessa, AIOLOS, cloudlet, etc. Note that, the term 'cloudlet' here is not the same with the term 'cloudlet' in Section IV-B2. Cloudlet refer to the wired, trusted resourceful computers deployed in the proximity of mobile users, not mobile device [204].

MAUI [178] is a computation offloading framework designed for Windows phones, which adopts Microsoft .NET to detect computing tasks that can be offloaded to remote cloud. There are three components in mobile device, Solver, Proxy and Profiler. Solver is the decision engine interface. Proxy keeps track of the state of the server side, while Profiler records the information on energy consumption, measurement and data transmission requirement. The server consists of four parts, in which Solver, Proxy and Profiler are corresponding with the mobile side and controller is used to deal with authentication. The architecture of MAUI is shown in Fig. 24. In opportunistic environment, MAUI can measure the offloading cost for each application and continuously detect the existence of MAUI server. If the mobile user move in a MAUI server's range, the MAUI client will determine whether to offload application tasks to the server according to the context information. Similarly, the Cuckoo [179] framework can be deployed on Android smart phones. When encountering Cuckoo server, mobile devices with Cuckoo clients can offload computing tasks to the server to save energy.

CloneCloud is a system that can automatically transmit the unmodified mobile application, which is performed in an application-level virtual machine, to a clone device in the cloud [205], [206]. The computing task are locally partitioned into two parts: the part that can be migrated to the cloud and the part that is remained on the local device. Theoretically, any VM-targeted applications can be partitioned. The partition mechanism of CloneCloud is similar to MAUI. Static program analysis and dynamic program profiling are combined together to determine the task partition. Threads, migrated to the cloud, start to run at the partitioned points in the clone environment. After they are finished, these threads returns back to local mobile device, then merge back to the original process. There is a difference between MAUI and CloneCloud. MAUI mainly focuses on the task partition, while CloneCloud has an efficient thread migration and merging mechanism. The evaluation shows that CloneCloud can reduce 20 fold energy consumption of mobile devices, while increase 20x execution speed of computing tasks. Taken broadly, the term cloud may be remote servers located in data centers, as well as resourceful computers or servers located in the proximity of mobile users. When users move into the communication range of a server, they can directly mitigate the partitioned computing tasks to the server through short range communication techniques, which is different from traditional cloud computing.

The key component of these offloading frameworks is the decision engine, which determines wether offload a task to an external server. The offloading decision is generally based on the prediction of execution time and energy consumption of the task, both local and external execution. However, the changeable wireless network capacity and computation input are not takin into consideration in MAUI and CloneCloud, which may lead to imprecise prediction on execution time.

ULOOF [213], [214] is designed to overcome these aforementioned shortcomings. Energy consumption and execution time are estimated before decision engine makes a choice, and the estimation will be updated after the actual execution locally or externally. Hence, ULOOF can adapt to the variable execution environment. Specifically, ULOOF adopts cost functions, which are based on historical execution results, to estimate the energy consumption and execution time in running the invoked methods. Decision engine makes the choice to offload the method of application to external server, only when the estimated local cost is higher than external. ULOOF supports both remote cloud offloading and offloading to nearby android devices. Real data based evaluation shows that ULOOF can reduce about 50% energy consumption in WiFi offloading scenario with low access latency. ThinkAir [216] is a similar computation offloading, which predicts energy consumption and execution time based on empirical data. However, ThinkAir is specially designed for commercial cloud, which may not available in opportunistic environment.

COSMOS [215] is another computation offloading framework designed for the network of variable connectivity, which makes offloading decisions in a risk-controlled manner. Connectivity waiting time, transmission time and execution time are considered to estimate offloading benefit. When the benefit is larger than a given threshold, task offloading occurs. The estimation will be refined by adjusting connectivity information, each time an execution is finished. There is a task allocation modular in COSMOS, which determines which COSMOS sever should perform the task. Three heuristic methods are designed to allocate task. The first method is that

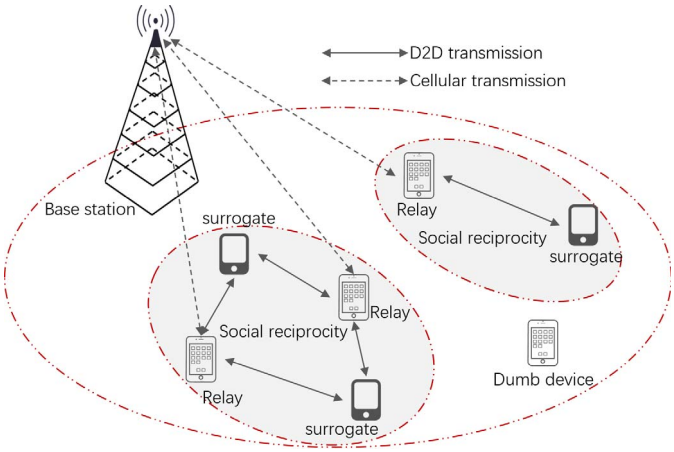


Fig. 23. The architecture of cloudlet. Each device has four parts: application component, Execution Environment, Operating System and Node Agent. 'C1' refers to application component. Ad hoc cloudlet consists of discovered devices. Elastic cloudlet performs on virtual infrastructure. Devices may offload application components to other devices in the same cloudlet through opportunistic communication or other cloudlet.

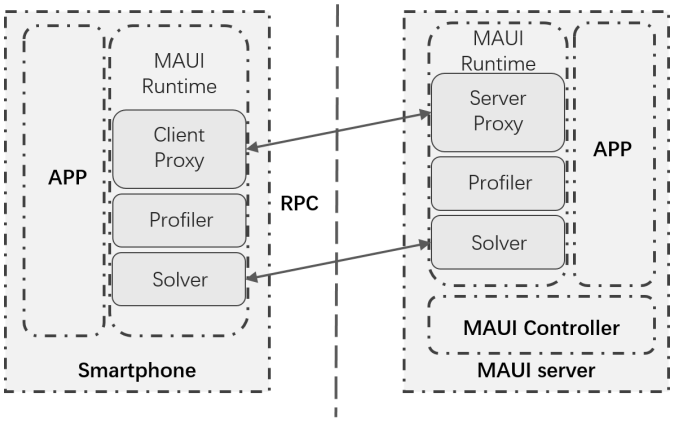


Fig. 24. The architecture of MAUI offloading system.

the COSMOS sever scheduler maintains a request queue, and allocate tasks to a COSMOS server with idle cores by time sequence. The second method is that the task requesting device queries a set of COSMOS severs and randomly select one with low workload. The third method is that the COSMOS server scheduler provides the task requesting device a set of COSMOS severs, as well as the average workload, and then the task requesting device randomly select one. Opportunistic network has a lot in common with the network of variable connectivity. Therefore, theoretically speaking, COSMOS can be directly applied to opportunistic environment.

Some computation offloading frameworks are based on cyber foraging, which extends the computing capacity of mobile devices with wired infrastructures [207], [208]. The wired infrastructure is called surrogate. Balan *et al.* [208] assume that surrogates are available for mobile users, which means that user mobility is not considered. Moreover, tasks execution in parallel is not considered. Odessa is a computation offloading framework designed for interactive perception applications, which has several special requirements on the capabilities of mobile devices. Interactive applications require quick response and the data processing algorithm are compute intensive. Hence, offloading to surrogates is a feasible solution. Odessa can dynamically make task offloading and parallel execution decisions between mobile device and surrogate based on a greedy algorithm. Specifically, Odessa periodically check the bottleneck of the current system. Then the decision maker part check the processor frequency and network history to estimate weather to offload to surrogate or increase parallel execution level of the bottleneck stage. The greedy and incremental based method in Odessa can effectively increase the system throughput. AIOLOS is a similar offloading framework designed for Android platform to reduce the execution time [209]. Both Odessa and AIOLOS are designed without the consideration user mobility. If we consider a dynamical scenario, where all users freely move and are able to communicate through wireless interfaces, these two framework can also be adapted to opportunistic offloading.

Some computation offloading frameworks are based on cloudlet. In a narrow sense, cloudlet refers to fixed infrastructure deployed in the proximity of access point. Mobile users with constraint computing resources may transmit computing tasks to high performance server [210]. In a broad sense, the cloudlet do not have to limited to that. Cloudlet can be opportunistically formed by mobile devices [211], [212]. The cloudlet architecture is shown in Fig. 23. There are two kinds of cloudlets in Fig. 23, ad hoc cloudlet, opportunistically formed by discovered devices and elastic cloudlet, performing on a virtual infrastructure. Each device in the cloudlet is divided into four parts: application component, Execution Environment, Operating System and Node Agent. The applications are managed on a component level. The Execution Environment, performed on the top of Operating System monitors the resource usage and determine to start or stop an application component. All devices are called nodes in cloudlet. Hence, the Node Agent is the control module, which is responsible for monitoring the resource usage of the whole device and task exchanging. Cloudlet Agent is hosted on the device with the most powerful computing resource. Cloudlet Agent can manage all devices in the cloudlet. The application component can either be migrated to other devices in the same cloudlet through opportunistic communication, or to other cloudlets. Since the Execution Environments may be different in different devices, each device in the cloudlet must configure virtual machine (VM) to perform application components for other devices. However, it is difficult to precisely configure the back-end software for VM. Ha *et al.* [217] propose to adopt just-in-time strategy to configure the VM in the cloudlet and design a prototype to demonstrate the effectiveness of the strategy.

Discussion: Although computation offloading frameworks, like MAUI, Cuckoo, CloneCloud, ULOOF, COSMOS, Odessa, AIOLOS, cloudlet, etc, are designed for traditional computation offloading, we find that they can be adapted to opportunistic environment. The goal of MAUI framework is to partition computing tasks to minimize the energy consumption and minimize the burden on programmer. However, MAUI can only offload computing tasks of .NET applications. In addition,

the offloaded partition in application need to be pre-marked, which is not suitable for third-party software. CloneCloud supports task offloading in thread level and can automatically perform tasks on VM. At the same time, it need to pre-configure the execution environment, which would put more burden on task performer. ULOOF makes offloading decision based on algorithm complexity estimation, which may not always be easy for developers. COSMOS only considers the network connectivity and execution time when making offloading decision. Whereas energy consumption should also be considered when offloading to equivalent mobile devices. For other offloading frameworks, they all focus on two key problems. The first is how to partition computing tasks, while the second problem is how to execute tasks in parallel, both in local devices and servers (i.e., surrogates and cloudlets). These two problems are well solved in cellular network environment. However, these two problems must be reviewed carefully in opportunistic environment, due to the user mobility.

V. Future Directions and Problems

We survey the existing state-of-art works on offloading cellular traffic and computing tasks by leveraging opportunistic network consisting of mobile devices. Benefits of opportunistic offloading are clear – it is a realistic technology to meet our exponentially increasing demands on mobile traffic, and applications of opportunistic offloading have increased dramatically. It is worth recapping that mobility of mobile devices is double-edged sword. On one hand, opportunistic offloading exploits mobility of nodes to create opportunistic contacts, where users transmit computing tasks to other users or download the requested content from other users with a low price or even free for resource and traffic. On the other hand, offloading is not always available due to user mobility. Users must be willing to stand for possible disruption and loss of packets, which results in long latency. In general, an opportunistic offloading application has multiple and often conflicting objectives. Since it is impossible to purely focus on achieving one goal without damaging other interests, the design is always a tradeoff between different goals. Through our intensive examination of the existing literature, it is also become evident that there still exist some big problems and challenges in realizing opportunistic offloading. We now outline some important future research directions to provide possible solution for these problems. We organize the discussions in five key areas: algorithm design, incentive mechanism, human behavior utilization, security and privacy, and computation-traffic offloading.

A. Algorithm Design

It is always necessary to design algorithms to deal with different stages/parts of an opportunistic offloading process. For example, an opportunistic offloading application may involve designing an offloading node selection algorithm to meet certain optimization goal, designing a relay selection algorithm to effectively forward data, designing a resource allocation algorithm to perform computing tasks for others, etc. A large amount of algorithms have been proposed from various perspectives. However, most of them have some unrealistic assumptions that are difficult to be met in practice. Some challenges on algorithm design must be addressed before opportunistic offloading can be realized in the real world.

The first challenge in algorithm design is how to deal with the heterogeneity of mobile nodes and contents/tasks. Mobile nodes are heterogeneous in their buffer, battery level, computing capability, etc. Contents/tasks are heterogeneous in their size, deadline, requirement for computing resources, etc. It is challenging to decide how to offload heterogeneous contents/tasks to heterogeneous mobile nodes. In traffic offloading, most works ignore the heterogeneity, and they assume that the buffer and energy of mobile nodes are infinite and the sizes of contents are the same. In computation offloading, most works assume that the workloads of all tasks are the same and all the tasks can be executed in parallel. Li *et al.* [32] are the first to take into account heterogeneity of mobile nodes and traffic content in opportunistic offloading. The authors establish a mathematical model and formulate the problem as a submodule optimal problem with multiple linear constraints. We believe considerable further researches will be carried out in order propose more efficient resolutions to this challenge.

The second challenge in algorithm design is how to achieve multiple and possibly conflicting objectives simultaneously. Most works in opportunistic offloading focus on achieving a single objective, e.g., maximizing the amount of reduced cellular traffic, or guaranteeing timely delivery, or minimizing the completion time of tasks, or improving energy efficiency. Few works focus on achieving multi-objectives at the same time. Actually, it is very challenging to achieve multi-objectives simultaneously in opportunistic offloading. In traffic offloading, various algorithms have been designed to maximize the amount of offloaded cellular traffic at the cost of deliberately delaying the delivery of data. However, subscriber nodes wish to receive the data as soon as possible. The smaller the delay is, the more satisfied the subscriber nodes are. Here there is a paradox between the two objectives. Future works are required in order to design better algorithm for achieving optimal tradeoff between these two conflicting objectives. In computation offloading, most works focus on designing algorithms to minimize the completion time of tasks or to minimize the energy consumption of mobile devices. Minimizing the completion time of tasks can improve the QoS, and minimizing the energy consumption can prolong the lifetime of mobile devices. Both the two objectives are important to mobile users in opportunistic offloading. However, few designs can achieve these two objectives simultaneously. Future works are required by taking into consideration of multiple objectives in order to design better algorithm for computation offloading.

The third challenge is how to practically test/verify the feasibility of the proposed algorithm, which is the step necessary towards implementation. Most works test their algorithms through simulations based on some mobility patterns, such as random walk model. These simulation experiments can only show whether the algorithms are feasible in the given idealized environments. A few works carry out the simulations with real user datasets, which are far better than using

idealized mobility models but are still far from real-world environments. In real world, there are various practical constraints that need to be taken into account, e.g., the interference in heterogeneous network. In traffic offloading, for example, the direct communications between mobile devices share the spectrum resource with cellular transmission, which will seriously affect the achievable performance of an offloading system. Few works have realistically investigated the effect of this interference. Considerable further works are required to realize realistic offloading test beds or systems in order to practically test various design algorithms, leading to eventual real-world implementations of opportunistic offloading systems.

We discuss the challenges in algorithm design, which opens up some future research directions. The heterogeneity of mobile nodes, rarely considered in the literature, has significant impact on the achievable performance of an offloading system, especially when the resource of each device is limited. Single-objective optimization based design cannot meet the true requirements of opportunistic offloading, because there are many different and possibly conflicting goals. Adopting a multi-objectives optimization approach will lead to better design. Simulations based on idealized mobility models are insufficient to test the feasibility of an offloading algorithm, and implementing real offloading test bed or system is necessary to evaluate practical implementation in real world. Future works will focus on addressing these problems in order to achieve better performance in opportunistic offloading, leading to real-world implementations of opportunistic offloading systems.

B. Incentive Mechanism

Opportunistic offloading is based on user collaboration, which requires the participating users to share their resources, e.g., CPU, battery, storage space, etc. Specifically, in opportunistic offloading, offloading-node users download content through cellular network or perform tasks for other mobile users, while relay users forward data to other mobile users, all at the expense of their own battery, storage space and/or computing resource. Wireless interfaces on these devices must always be turned on, which consumes more battery energy, and transmitting data through D2D communications also consumes a great amount of energy. Some works adopt multi-hops model to deliver data with the assumption that all mobile users are willing to act as relay to forward data for other users without any incentive, which is too optimistic. Some works adopt conservative one-hop model to deliver data, which does not utilize the full potential of opportunistic network. Whichever model adopted, multi-hop or one-hop, the critical challenge is how to motivate mobile users to participate in opportunistic offloading, considering the fact that not all mobile nodes with limited resources are willing to serve other mobile users.

Some works have studied using incentive mechanisms to reward mobile users who participate in opportunistic offloading and help other users. Game theories have been utilized to elaborate the relationship between offloading benefit and rewarding. Another possible solution is to establish a mechanism based on the user interest, battery level and buffer to let each mobile node voluntarily decide whether to participate in offloading. If a node with sufficient energy and buffer storage is interested in the data, it may voluntarily choose to download the content through cellular network and transmits the content to other nodes that are interested in it through opportunistic network. If a node with sufficient energy and enough idle computing resources is interested in the tasks, it may voluntarily choose to perform the tasks for other nodes. New communication techniques may be adopted in opportunistic offloading, e.g., Bluetooth 5.0, which can significantly save energy. However, the real challenge is establishing effective incentive mechanisms, which are appropriate for various real-world scenarios, to make it sufficiently attractive for mobile users to collaborating in opportunistic offloading. Considerable future researches are required to address this challenge.

C. User Behavior Utilization

In opportunistic offloading, user mobility is exploited to create communication opportunities to transmit data among mobile users. However, user mobility is a double-edged sword, which not only brings advantages that traditional cellular network and cloud computation do not have, but also creates the instability problem. There exists no stable end-to-end path between mobile users, and the delivery of data totally depends on opportunistic contact between mobile users. The critical challenge is how to effectively manage and utilize the mobility of mobile users. Most existing works focus on characterizing the mobility pattern of mobile nodes with history mobility, to predict when and where the contact will take place. This works reasonably well for the situations where the mobility patterns of mobile users exhibit sufficient degree of regularity. However, the mobility of irregularly moving nodes is unpredictable. In computation offloading, users mobility leads to dynamic changes of resources. A client node needs to send the tasks to an offloading node with idle computing resources nearby through opportunistic communication. However, the offloading node may leave the direct communication range of the client node before the tasks are finished. These tasks must be restarted on some other offloading node, which will lead to high delay and large cost. A possible solution in future research is to establish a stability evaluation mechanism for mobile users. The unpredictable mobile users with low stability are filtered out. Since social relationships, such as social ties, among users significantly affect users' behaviours towards each other, social attributions of users can be exploited to help accurately characterizing users' behaviors. This leads to better designed opportunistic network [201].

In the real world, mobile devices are held and controlled by humans, who have the instinctive and indispensable selfishness nature. Most existing researches in opportunistic offloading unrealistically assume that users will operate cooperatively and unselfishly to transmit data or to perform tasks for others. However, most users more or less behave in a selfish way, which makes user selfishness a key factor that affects the achievable performance of an opportunistic offloading system. We point out that a very recent work [202] has studied the impact of selfishness in D2D communications. Future works can be directed toward this important area.

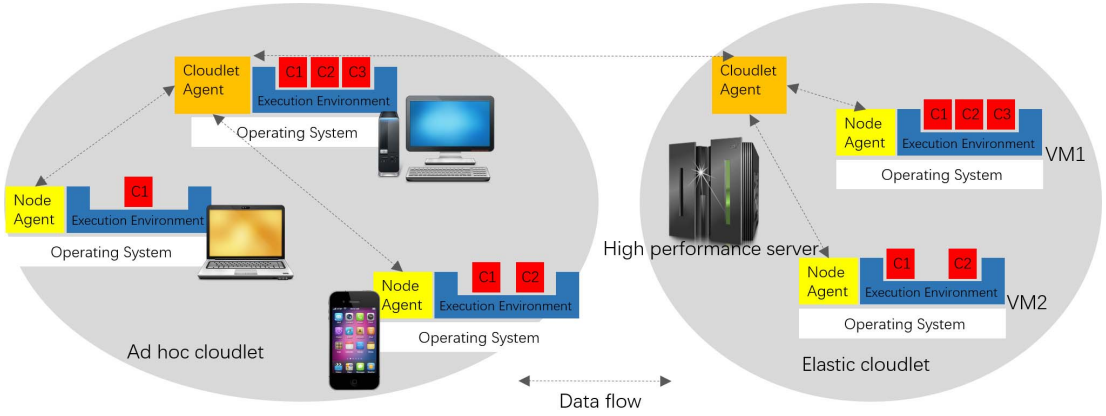


Fig. 25. A joint computation-traffic offloading application: the devices with sufficient computing resources can act as surrogate for the devices with insufficient computing resources with the requirement that the helped device acts as a relay to upload/download data through cellular network for its computing helper.

D. Security and Privacy

In our cellular network, we have placed ever-increasingly extensive security and safety protocols to keep it secure and safe. Mobile users cannot access the data stored on unfamiliar devices and will refuse the accessing from unfamiliar users, owing to security and privacy consideration. However, the situation in opportunistic offloading is rather strange to say the least. Mobile users may be asked to communicate with not only friends but also strangers, and yet the existing researches have not yet touched the security and privacy provisions.

The data transmitted in opportunistic network may contain the privacy information of users, and these privacy information may be handed to other users during the transmission through opportunistic network. Some mobile users may rightly deny communicating with other users who they do not trust, which may lead to the failure of offloading. On the other hand, malicious users can easily infect an offloading node through uploading a task containing virus or through DDoS, due to the lack of unified security protocols. Once infected, the offloading node cannot provide computing service for client nodes and it may further infect other nodes in opportunistic contacts. Since security and privacy must be paramount, this unsatisfactory situation must be resolved quickly. The critical challenge is how to prevent the privacy of mobile users being compromised and to protect mobile users from being attacked by malicious users, in the most open architecture of opportunistic network.

Considerable research efforts must be directed to address this challenge. A possible solution is to establish an authentication mechanism, which is operated in the operator side. Each nodes in the opportunistic offloading system must pass the security certification. The offloading node checks the credit information of the client node before performing task for it. In addition, the data transmitted in opportunistic network can be encrypted, and only authenticated users can obtain the key through cellular network.

E. Computation-Traffic Offloading

As discussed extensively in the previous sections, opportunistic offloading can be divided into two categories: traffic offloading and computation offloading. Specifically, mobile devices with sufficient computing resources may help other mobile devices with insufficient computing resources to perform tasks, while mobile devices with sufficient traffic usage may help other mobile devices with insufficient traffic usage to download/upload data. Most of the existing researches either focus on traffic offloading alone or study computation offloading separately. Few works consider both traffic offloading and computation offloading simultaneously. Clearly, it is beneficial, in terms of resource utilization and achievable performance, to jointly design traffic offloading and computation offloading, which may be referred to as computation-traffic offloading.

Pering and Ballagas [203] propose a computation-traffic offloading scheme based on mutually beneficial cooperation between mobile users in social network. The basic idea is that mobile devices with sufficient computing resources but insufficient traffic usage can form reciprocal groups with devices with sufficient traffic usage but insufficient computing resources to achieve a win-win situation. Devices that help other devices to perform computing tasks are called surrogates, while devices that help other devices to download/upload data through cellular network are called relays. This offloading process is illustrated in Fig. 25. A surrogate can execute task for a relay. At the same time, the relay can download/upload data for the surrogate through cellular network. When a device wants to offload a task to other devices, it may broadcast an offloading request with the description of the task to its neighbours. Devices that meet the capacity request can response. The traffic usage offloading process is similar. The authors design two algorithms to match the different goals of surrogates and relays. Specifically, for the operator, a centralized algorithm is designed based on matching surrogates and relays to minimize the overall energy cost, while for mobile devices, a distributed algorithm is derived based on game theory to minimize the energy cost of each device.

A particular senario that is worth mentioning is computing tasks with redundancy, i.e., many devices require to perform some same tasks. In this case, a few selected devices can offload the tasks to neighboring nodes with idle

computing resource through D2D communications. The results will be delivered through D2D communication not only to the client nodes but also to other nodes that need to perform the same tasks. Computation-traffic offloading is generally much more complex than traffic offloading or computation offloading alone. Considerable further researches are required in order to design effective computation-traffic offloading applications.

VI. Conclusion

In this paper, we survey the existing literature on opportunistic offloading. In particular, we propose a graded classification of the works in this recently emerged research area, by comparing and contrasting a large amount of the existing researches in the relevant categories. This survey therefore provides a comprehensive summary of the development of opportunistic offloading and the existing state-of-the-arts as well as offers the challenges and future research directions. As the underlying explosively increasing trends continue in mobile traffic and big artificial intelligence applications, we expect that research efforts and outputs in this exciting new area will exponentially increase for the following decades.

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," San Jose, CA, USA, Cisco, White Paper, Feb. 7, 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[2] B. Han *et al.*, "Cellular traffic offloading through opportunistic communications: A case study," in *Proc. 5th ACM Workshop Challenged Netw. (CHANTS)*, Chicago, IL, USA, 2010, pp. 31–38.

[3] E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using MapReduce," *Science*, vol. 0389, pp. 1–123, Sep. 2009.

[4] J. Wu, "Green wireless communications: From concept to reality [industry perspectives]," *IEEE Wireless Commun.*, vol. 19, no. 4, pp. 4–5, Aug. 2012.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, 2012, pp. 13–16.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G," Sophia Antipolis, France, ETSI, White Paper, pp. 1–16, 2015.

[7] A. Ahmed, A. Ahmed, and E. Ahmed, "A survey on mobile edge computing A survey on mobile edge computing," in *Proc. 10th IEEE Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 3–6.

[8] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, You Tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. ACM Internet Meas. Conf. (IMC)*, San Diego, CA, USA, 2007, pp. 1–14.

[9] Y. Lee *et al.*, "CoMon: Cooperative ambience monitoring platform with continuity and benefit awareness," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2012, pp. 43–56.

[10] N. B. Ashton and Q. Zhang, "Cool-SHARE: Offload smartphone data by sharing," in *Proc. IEEE Veh. Technol. Conf.*, vol. 2015. Seoul, South Korea, May 2015, pp. 1–5.

[11] B. Kaushik, H. Zhang, X. Fu, B. Liu, and J. Wang, "SmartParcel: A collaborative data sharing framework for mobile operating systems," in *Proc. Int. Conf. Distrib. Comput. Syst.*, Philadelphia, PA, USA, 2013, pp. 290–295.

[12] T. Han and N. Ansari, "Offloading mobile traffic via green content broker," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 161–170, Apr. 2014.

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

[14] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar. 2017.

[15] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[16] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Warsaw, Poland, Sep. 2014, pp. 1–8.

[17] F. Rebecchi *et al.*, "Data offloading techniques in cellular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 580–603, 2nd Quart., 2015.

[18] Y. Khadraoui, X. Lagrange, and A. Gravey, "A survey of available features for mobile traffic offload," in *Proc. 20th Eur. Wireless Conf. Eur. Wireless*, Barcelona, Spain, 2014, pp. 421–424.

[19] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627–640, Apr. 2015.

[20] S. Pal, "Extending mobile cloud platforms using opportunistic networks: Survey, classification and open issues," *J. Univ. Comput. Sci.*, vol. 21, no. 12, pp. 1594–1634, 2015.

[21] C. Tapparello, C. Funai, S. Hijazi, and A. Aquino, "Volunteer computing on mobile devices: State of the art and future research directions," in *Enabling Real-Time Mobile Cloud Computing Through Emerging Technologies*. Hershey, PA, USA: Inf. Sci. Ref., 2015.

[22] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, "The case for cyber foraging," in *Proc. 10th Workshop ACM SIGOPS Eur. Workshop Beyond (PC EW)*, Saint-Émilion, France, 2002, pp. 87–92.

[23] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb, "Simplifying cyber foraging for mobile devices," in *Proc. 5th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2007, pp. 272–285.

[24] A. Noori and D. Giustiniano, "HyCloud: A hybrid approach toward offloading cellular content through opportunistic communication," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, Taipei, Taiwan, 2013, pp. 551–552.

[25] M. V. Barbera, J. Stefa, A. C. Viana, M. D. De Amorim, and M. Boc, "VIP delegation: Enabling VIPs to offload data in wireless social mobile networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops (DCOSS)*, Barcelona, Spain, Mar. 2011, pp. 1–8.

[26] S. Sharafeddine, K. Jahed, N. Abbas, E. Yaacoub, and Z. Dawy, "Exploiting multiple wireless interfaces in smartphones for traffic offloading," in *Proc. 1st Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Batumi, Georgia, Jul. 2013, pp. 142–146.

[27] N.-S. Chen, Y.-F. Chou, R.-G. Cheng, and S.-L. Tsao, "Multiple contents offloading through opportunistic communications," in *Proc. Int. Conf. Telecommun. (ConTEL)*, Zagreb, Croatia, 2013, pp. 65–70.

[28] L. Valerio, R. Bruno, and A. Passarella, "Adaptive data offloading in opportunistic networks through an actor-critic learning method," in *Proc. 9th ACM MobiCom Workshop Challenged Netw.*, 2014, pp. 31–36.

[29] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Coding or not: Optimal mobile data offloading in opportunistic vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 318–333, Feb. 2014.

[30] M. V. Barbera, A. C. Viana, M. D. De Amorim, and J. Stefa, "Data offloading in social mobile networks through VIP delegation," *Ad Hoc Netw.*, vol. 19, pp. 92–110, Aug. 2014.

[31] V. F. S. Mota, D. F. Macedo, Y. Ghamri-Doudanez, and J. M. S. Nogueira, "Managing the decision-making process for opportunistic mobile data offloading," in *Proc. IEEE/IFIP NOMS IEEE/IFIP Netw. Oper. Manag. Symp. Manag. Softw. Defined World*, Krakoów, Poland, 2014, pp. 1–8.

[32] Y. Li *et al.*, "Multiple mobile data offloading through delay tolerant networks," in *Proc. 6th ACM Workshop Challenged Netw. (CHANTS)*, vol. 13. Las Vegas, NV, USA, 2011, pp. 43–48.

[33] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. T. Kwon, "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 2346–2354.

[34] R. Trestian, Q.-T. Vien, H. X. Nguyen, and O. Gemikonakli, "ECO-M: Energy-efficient cluster-oriented multimedia delivery in a LTE D2D environment," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 55–61.

[35] S. Bayhan, G. Premsankar, M. Di Francesco, and J. Kangasharju, "Mobile content offloading in database-assisted white space networks," in *Cognitive Radio Oriented Wireless Networks*, D. Noguet, K. Moessner, and J. Palicot, Eds. Cham, Switzerland: Springer Int., 2016, pp. 129–141.

[36] B. Barua, Z. Khan, Z. Han, A. A. Abouzeid, and M. Latva-Aho, "Incentivizing selected devices to perform cooperative content delivery: A carrier aggregation-based approach," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5030–5045, Jul. 2016.

[37] L. Vigneri, T. Spyropoulos, and C. Barakat, "Storage on wheels: Offloading popular contents through a vehicular cloud," in *Proc. IEEE 17th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Coimbra, Portugal, 2016, pp. 1–9.

[38] W. Wang, X. Wu, L. Xie, and S. Lu, "Joint storage assignment for D2D offloading systems," *Comput. Commun.*, vol. 83, pp. 45–55, Jun. 2016.

[39] B. Han, P. Hui, and A. Srinivasan, "Mobile data offloading in metropolitan area networks," *ACM SIGMOBILE Mobile Comput. Commun.*, vol. 14, no. 4, pp. 28–30, 2010.

[40] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, "Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Lucca, Italy, 2010, pp. 1–10.

[41] K. Thilakarathna, H. Petander, and A. Seneviratne, "Performance of content replication in MobiTribe: A distributed architecture for mobile UGC sharing," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, Bonn, Germany, 2011, pp. 558–566.

[42] Y.-J. Chuang and K. C.-J. Lin, "Cellular traffic offloading through community-based opportunistic dissemination," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, 2012, pp. 3188–3193.

[43] B. Han *et al.*, "Mobile data offloading through opportunistic communications and social participation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 821–834, May 2012.

[44] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M. D. De Amorim, "Push-and-track: Saving infrastructure bandwidth through opportunistic forwarding," *Pervasive Mobile Comput.*, vol. 8, no. 5, pp. 682–697, 2012.

[45] P. Baier, F. Dürr, and K. Rothermel, "TOMP: Opportunistic traffic offloading using movement predictions," in *Proc. Conf. Local Comput. Netw. (LCN)*, Clearwater, FL, USA, 2012, pp. 50–58.

[46] B. Proulx and J. Zhang, "Ameliorating cellular traffic peaks through preloading and P2P communications," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, 2013, pp. 4889–4894.

[47] W. Peng, F. Li, X. Zou, and J. Wu, "The virtue of patience: Offloading topical cellular content through opportunistic links," in *Proc. IEEE 10th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Hangzhou, China, 2013, pp. 402–410.

[48] F. Rebecchi, M. D. De Amorim, and V. Conan, "DROid: Adapting to individual mobility pays off in mobile data offloading," in *Proc. IFIP Netw. Conf.*, Trondheim, Norway, 2014, pp. 1–9.

[49] V. Sciancalepore, D. Giustiniano, A. Banchs, and A. Hossmann-Picu, "Offloading cellular traffic through opportunistic communications: Analysis and optimization," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 122–137, Jan. 2016.

[50] F. Rebecchi *et al.*, "A joint multicast/D2D learning-based approach to LTE traffic offloading," *Comput. Commun.*, vol. 72, pp. 26–37, Dec. 2015.

[51] Z. Li, Y. Shi, S. Chen, and J. Zhao, "Cellular traffic offloading through opportunistic communications based on human mobility," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 3, pp. 872–885, 2015.

[52] L. Valerio, R. Bruno, and A. Passarella, "Cellular traffic offloading via opportunistic networking with reinforcement learning," *Comput. Commun.*, vol. 71, pp. 129–141, Nov. 2015.

[53] D. Wenxiang, C. Jie, Y. Ying, and Z. Wenyi, "Epidemic-like proximity-based traffic offloading," *China Commun.*, vol. 12, no. 10, pp. 91–107, Oct. 2015.

[54] R.-G. Cheng, N.-S. Chen, Y.-F. Chou, and Z. Becvar, "Offloading multiple mobile data contents through opportunistic device-to-device communications," *Wireless Pers. Commun.*, vol. 84, no. 3, pp. 1963–1979, 2015.

[55] H.-H. Cheng and K. C.-J. Lin, "Source selection and content dissemination for preference-aware traffic offloading," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 11, pp. 3160–3174, Nov. 2015.

[56] X. Wang, X. Li, and V. C. M. Leung, "TASA: Traffic offloading by tag-assisted social-aware opportunistic sharing in mobile social networks," in *Proc. IEEE Workshop Local Metropolitan Area Netw.*, vol. 2015. Beijing, China, May 2015, pp. 1–6.

[57] X. Lu, P. Lio, and P. Hui, "Distance-based opportunistic mobile data offloading," *Sensors*, vol. 16, no. 6, p. 878, 2016.

[58] F. Rebecchi, M. D. D. Amorim, and V. Conan, "Should i seed or should i not: On the remuneration of seeders in D2D offloading," in *Proc. IEEE WoWMoM 17th Int. Symp. World Wireless Mobile Multimedia Netw.*, Coimbra, Portugal, 2016, pp. 1–9.

[59] X. Zhuo, W. Gao, G. Cao, and Y. Dai, "Win-coupon: An incentive framework for 3G traffic offloading," in *Proc. Int. Conf. Netw. Protocols (ICNP)*, Vancouver, BC, Canada, 2011, pp. 206–215.

[60] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, Jun. 2005, Art. no. 814818.

[61] L. Xiaofeng, H. Pan, and P. Lio, "Offloading mobile data from cellular networks through peer-to-peer WiFi communication: A subscribe-and-send architecture," *China Commun.*, vol. 10, no. 6, pp. 35–46, Jun. 2013.

[62] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 2014, pp. 2648–2653.

[63] J. Gu, W. Wang, A. Huang, and H. Shan, "Proactive storage at caching-enable base stations in cellular networks," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, U.K., 2013, pp. 1543–1547.

[64] S. Wang, T. Lei, L. Zhang, C.-H. Hsu, and F. Yang, "Offloading mobile data traffic for QoS-aware service provision in vehicular cyber-physical systems," *Future Gen. Comput. Syst.*, vol. 61, pp. 118–127, Aug. 2016.

[65] M. Lee, J. Song, J. P. Jeong, and T. T. Kwon, "DOVE: Data offloading through spatio-temporal rendezvous in vehicular networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, vol. 2015. Las Vegas, NV, USA, Oct. 2015, pp. 1–8.

[66] G. E. M. Zhioua, J. Zhang, H. Labiod, N. Tabbane, and S. Tabbane, "A joint active time and flow selection model for cellular content retrieval through ITS," *Comput. Netw.*, vol. 107, pp. 220–232, Oct. 2016.

[67] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Content download in vehicular networks in presence of noisy mobility prediction," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1007–1021, May 2014.

[68] L. Mu and A. Prinz, "Delay-oriented data traffic migration in maritime mobile communication environments," in *Proc. 4th Int. Conf. Ubiquitous Future Netw. Final Program (ICUFN)*, 2012, pp. 417–422.

[69] E. V. Dias, E. R. A. Vargas, M. C. Q. Farias, and M. M. Carvalho, "Feasibility of video streaming offloading via connection sharing from LTE to WiFi ad hoc networks," in *Proc. Int. Workshop Telecommun. (IWT)*, 2015, pp. 1–6.

[70] S. Yoon, D. T. Ha, H. Q. Ngo, and C. Qiao, "MoPADS: A mobility profile aided file downloading service in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5235–5246, Nov. 2009.

[71] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Offloading cellular networks through ITS content download," in *Proc. Annu. IEEE Commun. Soc. Conf. Sensor Mesh Ad Hoc Commun. Netw. Workshops*, vol. 1. Seoul, South Korea, Jun. 2012, pp. 263–271.

[72] B. Baron, P. Spathis, H. Rivano, and M. D. de Amorim, "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3248–3261, Dec. 2016.

[73] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular WiFi access," in *Proc. 7th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Kraków, Poland, 2009, pp. 263–276.

[74] K. Ezirim and S. Jain, "Taxi-cab cloud architecture to offload data traffic from cellular networks," in *Proc. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Boston, MA, USA, Aug. 2015, pp. 1–6.

[75] A. T. Fuller, "Bibliography of pontryagm's maximum principle," *Int. J. Electron.*, vol. 15, no. 5, pp. 513–517, 1963.

[76] J. Froehlich and J. Krumm, "Route prediction from trip observations," in *Proc. SAE World Congr.*, vol. 2193, 2008, p. 53.

[77] J. Krumm, "A Markov model for driver turn prediction," in *Proc. SAE World Congr.*, 2008.

[78] L. Mu, R. Kumar, and A. Prinz, "An integrated wireless communication architecture for maritime sector," in *Multiple Access Communications*, C. Sacchi *et al.*, Eds. Berlin, Germany: Springer, 2011, pp. 193–205.

[79] O. Grémillet *et al.*, "Aeronautical communications for personal and multimedia services via satellite," in *Proc. 21st Int. Commun. Satellite Syst. Conf. Exhibit (ICSSC)*, Yokohama, Japan, Apr. 2003.

[80] Q. Vey, A. Pirovano, J. Radzik, and F. Garcia, "Aeronautical ad hoc network for civil aviation," in *Proc. Nets4Cars/Nets4Trains/Nets4Aircraft*, Offenburg, Germany, 2014, pp. 81–93.

[81] M. Schnell, U. Epple, D. Shutin, and N. Schneckenburger, "LDACS: Future aeronautical communications for air-traffic management," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 104–110, May 2014.

[82] R. Jain, F. Templin, and K.-S. Yin, "Analysis of L-band digital aeronautical communication systems: L-DACS1 and L-DACS2," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2011, pp. 1–10.

[83] T. Gräupl, M. Ehammer, and S. Zwettler, "L-DACS1 air-to-air data-link protocol design and performance," in *Proc. IEEE Integr. Commun. Navig. Surveilance Conf. (ICNS)*, Herndon, VA, USA, 2011, pp. B3-1–B3-14.

[84] J. Zhang, S. Chen, R. G. Maunder, R. Zhang, and L. Hanzo, "Adaptive coding and modulation for large-scale antenna array-based aeronautical communications in the presence of co-channel interference," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1343–1357, Feb. 2018.

[85] L. Al-Kanj and Z. Dawy, "Offloading wireless cellular networks via energy-constrained local ad hoc networks," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, Kathmandu, Nepal, 2011, pp. 1–6.

[86] L. Valerio *et al.*, "Offloading cellular traffic with opportunistic networks: A feasibility study," in *Proc. 14th Annu. Mediterr. Ad Hoc Netw. Workshop (MED HOC NET)*, Jun. 2015, pp. 1–8.

[87] R. Lan, W. Wang, A. Huang, and H. Shan, "Device-to-device offloading with proactive caching in mobile cellular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[88] M. J. Yang, S. Y. Lim, H. J. Park, and N. H. Park, "Solving the data overload: Device-to-device bearer control architecture for cellular data offloading," *IEEE Veh. Technol. Mag.*, vol. 8, no. 1, pp. 31–39, Mar. 2013.

[89] H. Seferoglu and Y. Xing, "Device-centric cooperation in mobile networks," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Luxembourg City, Luxembourg, 2014, pp. 217–222.

[90] A. Le *et al.*, "Microcast: Cooperative video streaming using cellular and local connections," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2983–2999, Oct. 2016.

[91] S. T. Kouyoumdjieva and G. Karlsson, "Energy-aware opportunistic mobile data offloading for users in urban environments," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, Toulouse, France, 2015, pp. 1–9.

[92] S. Andreev, O. Galinina, A. Pyattaev, K. Johnsson, and Y. Koucheryavy, "Analyzing assisted offloading of cellular user sessions onto D2D links in unlicensed bands," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 1, pp. 67–80, Jan. 2015.

[93] C. P. Mayer and O. P. Waldhorst, "Offloading infrastructure using delay tolerant networks and assurance of delivery," in *Proc. IFIP Wireless Days*, vol. 1. Niagara Falls, ON, Canada, 2011, pp. 1–7.

[94] X. Zhuo, W. Gao, G. Cao, and S. Hua, "An incentive framework for cellular traffic offloading," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 541–555, Mar. 2014.

[95] K. Sugiyama, T. Kubo, A. Tagami, and A. Parekh, "Incentive mechanism for DTN-based message delivery services," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, 2013, pp. 3130–3135.

[96] W. Gao and Q. Li, "Wakeup scheduling for energy-efficient communication in opportunistic mobile networks," in *Proc. INFOCOM*, Turin, Italy, 2013, pp. 2106–2114.

[97] G. B. T. Kalejaiye *et al.*, "Mobile offloading in wireless ad hoc networks: The tightness strategy," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 96–102, 2014.

[98] S. T. Kouyoumdjieva and G. Karlsson, "Energy savings in opportunistic networks," in *Proc. 11th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, 2014, pp. 57–64.

[99] Z. Li, Y. Liu, H. Zhu, and L. Sun, "Coff: Contact-duration-aware cellular traffic offloading over delay tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 5257–5268, Nov. 2015.

[100] S. Zhang, J. Wu, Z. Qian, and S. Lu, "MobiCache: Cellular traffic offloading leveraging cooperative caching in mobile social networks," *Comput. Netw.*, vol. 83, pp. 184–198, Jun. 2015.

[101] T. Wang, Y. Sun, L. Song, and Z. Han, "Social data offloading in D2D-enhanced cellular networks by network formation games," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 7004–7015, Dec. 2015.

[102] G. Mao, Z. Zhang, and B. D. O. Anderson, "Cooperative content dissemination and offloading in heterogeneous mobile networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6573–6587, Aug. 2016.

[103] Z. Chang, J. Gong, Z. Zhou, T. Ristaniemi, and Z. Niu, "Resource allocation and data offloading for energy efficiency in wireless power transfer enabled collaborative mobile clouds," in *Proc. IEEE INFOCOM*, vol. 2015, Aug. 2015, pp. 336–341.

[104] S. T. Kouyoumdjieva and G. Karlsson, "The virtue of selfishness: Device perspective on mobile data offloading," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, New Orleans, LA, USA, 2015, pp. 2085–2090.

[105] A. Samydurai, N. Karunakaran, K. R. R. Priya, and S. Sujitha, "Device to device resource dissemination by social network," *Int. Res. J. Adv. Eng. Sci.*, vol. 1, no. 2, pp. 76–78, 2016.

[106] S. T. Kouyoumdjieva and G. Karlsson, "Device-to-device mobile data offloading for music streaming," in *Proc. IFIP Netw. Conf. Workshops (IFIP Netw.)*, Vienna, Austria, 2016, pp. 386–394.

[107] T. Zimmermann, J. Rüth, H. Wirtz, and K. Wehrle, "Maintaining integrity and reputation in content offloading," in *Proc. 12th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, Jan. 2016, pp. 1–8.

[108] Z. Lu, X. Sun, and T. L. Porta, "Cooperative data offloading in opportunistic mobile networks," in *Proc. IEEE INFOCOM*, vol. 2016. San Francisco, CA, USA, Jul. 2016, pp. 1–9.

[109] Y. Li *et al.*, "Contract-based traffic offloading over delay tolerant networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[110] Y. Li *et al.*, "A contract-based incentive mechanism for delayed traffic offloading in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5314–5327, Aug. 2016.

[111] T. Lei, S.-G. Wang, and F.-C. Yang, "An adaptive data traffic offloading model for cellular machine-to-machine networks," in *Internet of Vehicles—Safe and Intelligent Mobility*, C.-H. Hsu, F. Xia, X. Liu, and S. Wang, Eds. Cham, Switzerland: Springer Int., 2015, pp. 261–272.

[112] L. Al-Kanj, H. V. Poor, and Z. Dawy, "Optimal cellular offloading via device-to-device communication networks with fairness constraints," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4628–4643, Aug. 2014.

[113] H. Zhang, Y. Li, D. Jin, and S. Chen, "Selfishness in device-to-device communication underlaying cellular networks," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., 2015, pp. 675–679.

[114] S. Wen, F. R. Yu, and J. Wu, "Stochastic predictive control for energy-efficient cooperative wireless cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, 2013, pp. 4399–4403.

[115] Q. Wu *et al.*, "Resource allocation for joint transmitter and receiver energy efficiency maximization in downlink OFDMA systems," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 416–430, Feb. 2015.

[116] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, Apr. 2012.

[117] Y. Sun *et al.*, "Traffic offloading in two-tier multi-mode small cell networks over unlicensed bands: A hierarchical learning framework," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 11, pp. 4291–4310, 2015.

[118] I. Siomina and D. Yuan, "Analysis of cell load coupling for LTE network planning and optimization," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2287–2297, Jun. 2012.

[119] L. Saker, S. E. Elayoubi, R. Combes, and T. Chahed, "Optimal control of wake up mechanisms of femtocells in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 664–672, Apr. 2012.

[120] Y.-H. Chiang and W. Liao, "Genie: An optimal green policy for energy saving and traffic offloading in heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Commun.*, Budapest, Hungary, 2013, pp. 6230–6234.

[121] D. W. K. Ng, E. S. Lo, and R. Schober, "Wireless information and power transfer: Energy efficiency optimization in OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6352–6370, Dec. 2013.

[122] R. Zhang and C. K. Ho, "MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 1989–2001, May 2013.

[123] P. Grover and A. Sahai, "Shannon meets tesla: Wireless information and power transfer," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, 2010, pp. 2363–2367.

[124] Q. Shi, W. Xu, J. Wu, E. Song, and Y. Wang, "Secure beamforming for MIMO broadcasting with wireless information and power transfer," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2841–2853, May 2015.

[125] Y. Wu, S. Deng, and H. Huang, "Hop limited epidemic-like information spreading in mobile social networks with selfish nodes," *J. Phys. A Math. Theor.*, vol. 46, no. 26, 2013, Art. no. 265101.

[126] H. Izumikawa, "Spatial uplink mobile data offloading leveraging storecarry—Forward paradigm," in *Proc. 1st ACM Int. Workshop Pract. Issues Appl. Next Gener. Wireless Netw.*, 2012, pp. 33–38.

[127] R. Stanica, M. Fiore, and F. Malandrino, "Offloading floating car data," in *Proc. IEEE 14th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Madrid, Spain, 2013, pp. 1–9.

[128] S. Ancona, R. Stanica, and M. Fiore, "Performance boundaries of massive floating car data offloading," in *Proc. IEEE/IFIP 11th Annu. Conf. Wireless Demand Netw. Syst. Services (WONS)*, 2014, pp. 89–96.

[129] V. Miliotis, L. Alonso, and C. Verikoukis, "Weighted proportional fairness and pricing based resource allocation for uplink offloading using IP flow mobility," *Ad Hoc Netw.*, vol. 49, pp. 17–28, Oct. 2016.

[130] G. Gao, M. Xiao, J. Wu, K. Han, and L. Huang, "Deadline-sensitive mobile data offloading via opportunistic communications," in *Proc. 13th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, London, U.K., 2016, pp. 1–9.

[131] P. Kolios, C. Panayiotou, and G. Ellinas, "ExTraCT: Expediting offloading transfers through intervehicle communication transmissions," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1238–1248, Jun. 2015.

[132] H. Izumikawa and J. Katto, "RoCNet: Spatial mobile data offload with user-behavior prediction through delay tolerant networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, 2013, pp. 2196–2201.

[133] W. Hu and G. Cao, "Quality-aware traffic offloading in wireless networks," in *Proc. 15th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Philadelphia, PA, USA, 2014, pp. 277–286.

[134] V. Miliotis, L. Alonso, and C. Verikoukis, "Energy efficient proportionally fair uplink offloading for IP flow mobility," in *Proc. IEEE 19th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Athens, Greece, 2014, pp. 6–10.

[135] V. Miliotis, L. Alonso, and C. Verikoukis, "Combating selfish misbehavior with reputation based uplink offloading for IP flow mobility," in *Proc. IEEE Int. Conf. Commun.*, vol. 2015. London, U.K., Sep. 2015, pp. 7012–7017.

[136] V. Miliotis, L. Alonso, and C. Verikoukis, "Offloading with IFOM: The uplink case," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Feb. 2015, pp. 2661–2666.

[137] I. Komnios, F. Tsapeli, and S. Gorinsky, "Cost-effective multi-mode offloading with peer-assisted communications," *Ad Hoc Netw.*, vol. 25, pp. 370–382, Feb. 2015.

[138] U. Sethakaset, Y.-K. Chia, and S. Sun, "Energy efficient WiFi offloading for cellular uplink transmissions," in *Proc. IEEE Veh. Technol. Conf.*, vol. 2015. Seoul, South Korea, Jan. 2015, pp. 1–5.

[139] J. Huang *et al.*, "A close examination of performance and power characteristics of 4G LTE networks," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 225–238.

[140] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li, "Cellular traffic offloading through WiFi networks," in *Proc. 8th IEEE Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Valencia, Spain, 2011, pp. 192–201.

[141] A. De La Oliva, C. J. Bernardos, M. Calderon, T. Melia, and J. C. Zuniga, "IP flow mobility: Smart traffic offload for future wireless networks," *IEEE Commun. Mag.*, vol. 49, no. 10, pp. 124–132, Oct. 2011.

[142] E. Bergfeldt, S. Ekelin, and J. M. Karlsson, "A performance study of bandwidth measurement tools over mobile connections," in *Proc. IEEE Veh. Technol. Conf.*, Barcelona, Spain, 2009, pp. 1–5.

[143] A. Schulman *et al.*, "Bartendr: A practical approach to energy-aware cellular data scheduling," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Chicago, IL, USA, 2010, pp. 85–96.

[144] K. Thilakarathna, A. A. A. Karim, H. Petander, and A. Seneviratne, "Mobitribe: Enabling device centric social networking on smart mobile devices," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, New Orleans, LA, USA, Jun. 2013, pp. 230–232.

[145] K. Thilakarathna, H. Petander, J. Mestre, and A. Seneviratne, "MobiTribe: Cost efficient distributed user generated content sharing on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 9, pp. 2058–2070, Sep. 2014.

[146] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop Mobile Cloud Comput. Services Soc. Netw. Beyond (MCS)*, vol. 16, 2010, Art. no. 6. [Online]. Available: http://www.sophos.com/en–us/threat–center/mobile–s

[147] P. Yang *et al.*, "'Friend is treasure': Exploring and exploiting mobile social contacts for efficient task offloading," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5485–5496, Jul. 2016.

[148] A. Banerjee, H. S. Paul, A. Mukherjee, S. Dey, and P. Datta, "A framework for speculative scheduling and device selection for task execution on a mobile cloud," in *Adaptive Resource Management and Scheduling for Cloud Computing*, F. Pop and M. Potop-Butucaru, Eds. Cham, Switzerland: Springer Int., 2014, pp. 36–51.

[149] W. Liu, T. Nishio, R. Shinkuma, and T. Takahashi, "Adaptive resource discovery in mobile cloud computing," *Comput. Commun.*, vol. 50, pp. 119–129, Sep. 2014.

[150] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "Have you asked your neighbors? A hidden market approach for device-to-device offloading," in *Proc. 17th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Coimbra, Portugal, 2016, pp. 1–9.

[151] J. Xu, L. Chen, K. Liu, and C. Shen, "Less is more: Participation incentives in D2D-enhanced mobile edge computing under infectious DDoS attacks," *arXiv preprint arXiv:1611.03841*, pp. 1–15, 2016.

[152] Q. Li, P. Yang, S. Tang, C. Xiang, and F. Li, "Many is better than all: Efficient selfish load balancing in mobile crowdsourcing systems," in *Proc. 3rd Int. Conf. Adv. Cloud Big Data*, Yangzhou, China, 2015, pp. 1–6.

[153] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "OPENRP: A reputation middleware for opportunistic crowd computing," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 115–121, Jul. 2016.

[154] A. Zhou, S. Wang, J. Li, Q. Sun, and F. Yang, "Optimal mobile device selection for mobile cloud service providing," *J. Supercomput.*, vol. 72, no. 8, pp. 3222–3235, 2016.

[155] M. Khaledi, M. Khaledi, and S. K. Kasera, "Profitable task allocation in mobile cloud computing," in *Proc. 12th ACM Symp. QoS Security Wireless Mobile Netw. (Q2SWinet)*, 2016, pp. 9–17.

[156] Q. Li and P. Yang, "STORE: Simple task offloading and reassignment for mobile social network," in *Proc. 8th Int. ICST Conf. Commun. Netw. China (CHINACOM)*, Guilin, China, 2013, pp. 581–586.

[157] Z. Lu, J. Zhao, Y. Wu, and G. Cao, "Task allocation for mobile cloud computing in heterogeneous wireless networks," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Las Vegas, NV, USA, 2015, pp. 1–9.

[158] Q. Li, P. Yang, Y. Yan, and Y. Tao, "Your friends are more powerful than you: Efficient task offloading through social contacts," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 2014, pp. 88–93.

[159] S. Ghasemi-Falavarjani, M. A. Nematbakhsh, and B. S. Ghahfarokhi, "A multi-criteria resource allocation mechanism for mobile clouds," in *Proc. Int. Symp. Comput. Netw. Distrib. Syst.*, Tehran, Iran, 2013, pp. 145–154.

[160] A. Mtibaa, M. A. Snober, A. Carelli, R. Beraldi, and H. Alnuweiri, "Collaborative mobile-to-mobile computation offloading," in *Proc. 10th IEEE Int. Conf. Collaborative Comput. Netw. Appl. Worksharing (CollaborateCom)*, Miami, FL, USA, 2014, pp. 460–465.

[161] C. Shi, M. H. Ammar, E. W. Zegura, and M. Naik, "Computing in cirrus clouds: The challenge of intermittent connectivity," in *Proc. ACM SIGCOMM Workshop Mobile Cloud Comput.*, Helsinki, Finland, 2012, pp. 23–28.

[162] S. Ghasemi-Falavarjani, M. A. Nematbakhsh, and B. S. Ghahfarokhi, "Context-aware multi-objective resource allocation in mobile cloud," *Comput. Elect. Eng.*, vol. 44, pp. 218–240, May 2015.

[163] E. M. Trono *et al.*, "Disaster area mapping using spatially-distributed computing nodes across a DTN," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops) (PerMoby)*, Sydney, NSW, Australia, 2016, pp. 1–6.

[164] E. M. Trono, Y. Arakawa, M. Tamai, and K. Yasumoto, "DTN MapEx: Disaster area mapping through distributed computing over a delay tolerant network," in *Proc. 8th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, 2015, pp. 179–184.

[165] A. Fahim, A. Mtibaa, and K. A. Harras, "Making the case for computational offloading in mobile device clouds," in *Proc. Int. Conf. Mobile*, Miami, FL, USA, Sep./Oct. 2013, pp. 203–205.

[166] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *Proc. IEEE INFOCOM*, vol. 26, 2015, pp. 2227–2235.

[167] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2306–2320, Aug. 2017.

[168] K. Alanezi, X. Zhou, L. Chen, and S. Mishra, "Panorama: A framework to support collaborative context monitoring on co-located mobile devices," in *Mobile Computing, Applications, and Services*, S. Sigg, P. Nurmi, and F. Salim, Eds. Cham, Switzerland: Springer Int., 2015, pp. 143–160.

[169] H. Flores *et al.*, "Social-aware hybrid mobile offloading," *Pervasive Mobile Comput.*, vol. 36, pp. 25–43, Apr. 2017.

[170] A. Mtibaa, K. A. Harras, and A. Fahim, "Towards computational offloading in mobile device clouds," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, vol. 1. Bristol, U.K., 2013, pp. 331–338.

[171] D. Chatzopoulos, K. Sucipto, S. Kosta, and P. Hui, "Video compression in the neighborhood: An opportunistic approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.

[172] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proc. Netw. Comput.*, 2012, pp. 145–154.

[173] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Resource allocation for energy efficient k-out-of-n system in mobile ad hoc networks," in *Proc. 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2013, pp. 1–9.

[174] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage & processing in dynamic networks," in *Proc. 14th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2013, pp. 281–286.

[175] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, Jan./Mar. 2015.

[176] W. Gao, "Opportunistic peer-to-peer mobile cloud computing at the tactical edge," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Baltimore, MD, USA, 2014, pp. 1614–1620.

[177] B. N. Begum and S. Prasanna, "Data storage and precision of dynamism-efficient fault-tolerant in mobile cloud," vol. 1, 2015.

[178] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Jun. 2010, pp. 49–62.

[179] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: A computation offloading framework for smartphones," in *Mobile Computing, Applications, and Services*, M. Gris and G. Yang, Eds. Berlin, Germany: Springer, 2012, pp. 59–79.

[180] J. Wu *et al.*, "Context-aware networking and communications: Part 1 [guest editorial]," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 14–15, Jun. 2014.

[181] Q. Li, H. Li, P. Russell, Z. Chen, and C. Wang, "CA-P2P: Context-aware proximity-based peer-to-peer wireless communications," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 32–41, Jun. 2014.

[182] R. Hasan and R. Khan, "A cloud you can wear: Towards a mobile and wearable personal cloud," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Atlanta, GA, USA, 2016, pp. 823–828.

[183] L. Wu, X. Du, H. Zhang, W. Yu, and C. Wang, "Effective task scheduling in proximate mobile device based communication systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3503–3508.

[184] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, QLD, Australia, 2015, pp. 1271–1274.

[185] W. Liu, R. Shinkuma, and T. Takahashi, "Opportunistic resource sharing in mobile cloud computing: The single-copy case," in *Proc. 16th Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Hsinchu, Taiwan, 2014, pp. 1–4.

[186] H. Jin, S. Yan, C. Zhao, and D. Liang, "PMC^{2}O: Mobile cloudlet networking and performance analysis based on computation offloading," *Ad Hoc Netw.*, vol. 58, pp. 86–98, Apr. 2017.

[187] D. Zeng, S. Guo, I. Stojmenovic, and S. Yu, "Stochastic modeling and analysis of opportunistic computing in intermittent mobile cloud," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Melbourne, VIC, Australia, Jul. 2013, pp. 1902–1907.

[188] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: Leveraging mobile devices to provide cloud service at the edge," in *Proc. IEEE 8th Int. Conf. Cloud Comput. (CLOUD)*, New York, NY, USA, Jun./Jul. 2015, pp. 9–16.

[189] B. Li, Z. Liu, Y. Pei, and H. Wu, "Mobility prediction based opportunistic computational offloading for mobile device cloud," in *Proc. 17th IEEE Int. Conf. Comput. Sci. Eng. (CSE) Jointly 13th IEEE Int. Conf. Ubiquitous Comput. Commun. (IUCC) 13th Int. Symp. Pervasive Syst.*, Chengdu, China, 2015, pp. 786–792.

[190] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," *Comput. Commun. Rev.*, vol. 43, no. 4, pp. 579–584, 2013.

[191] Y. Zhang, D. Niyato, P. Wang, and C.-K. Tham, "Dynamic offloading algorithm in intermittently connected mobile cloudlet systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 4190–4195.

[192] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[193] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1060–1068.

[194] T. Truong-Huu, C.-K. Tham, and D. Niyato, "To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, Singapore, Dec. 2014, pp. 182–189.

[195] C. Wang, Y. Li, D. Jin, and S. Chen, "On the serviceability of mobile vehicular cloudlets in a large-scale urban environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2960–2970, Oct. 2016.

[196] A. Monfared, M. Ammar, E. Zegura, D. Doria, and D. Bruno, "Computational ferrying: Challenges in deploying a mobile high performance computer," in *Proc. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–6.

[197] C. R. Panigrahi, B. Pati, M. Tiwary, and J. L. Sarkar, "EEOA: Improving energy efficiency of mobile cloudlets using efficient offloading approach," in *Proc. Int. Symp. Adv. Netw. Telecommun. Syst. (ANTS)*, vol. 2016. Kolkata, India, Dec. 2016, pp. 1–6.

[198] L. Xiang, B. Li, and B. Li, "Coalition formation towards energy-efficient collaborative mobile computing," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, vol. 2015. Las Vegas, NV, USA, Aug. 2015, pp. 1–8.

[199] M. Chen *et al.*, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Trans. Services Comput.*, to be published.

[200] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A survey of cloudlet based mobile computing," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Shanghai, China, 2016, pp. 268–275.

[201] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen, "Social-aware d2d communications: Qualitative insights and quantitative analysis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 150–158, Jun. 2014.

[202] C. Gao *et al.*, "Impact of selfishness in device-to-device communication underlying cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9338–9349, Oct. 2017.

[203] B. T. Pering and R. Ballagas, "Share communication and computation resources on mobile devices: A social awareness perspective," *Commun. ACM*, vol. 48, no. 9, pp. 53–59, 2016.

[204] M. Satyanarayanan, "Mobile computing: The next decade," in *Proc. 1st ACM Workshop Mobile Cloud Comput. Services Soc. Netw. Beyond (MCS)*, San Francisco, CA, USA, 2010, pp. 1–6.

[205] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proc. 12th Conf. Hot Topics Oper. Syst. (HotOS)*, 2009, p. 8.

[206] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst. (EuroSys)*, 2011, pp. 301–314.

[207] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, Aug. 2001.

[208] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, "The case for cyber foraging," in *Proc. 10th Workshop ACM SIGOPS Eur. Workshop (EW)*, Saint-Émilion, France, 2002, pp. 87–92.

[209] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "AIOLOS: Middleware for improving mobile application performance through cyber foraging," *J. Syst. Softw.*, vol. 85, no. 11, pp. 2629–2639, 2012.

[210] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[211] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Services (MCS)*, 2012, pp. 29–36.

[212] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "Adaptive deployment and configuration for mobile augmented reality in the cloudlet," *J. Netw. Comput. Appl.*, vol. 41, pp. 206–216, May 2014.

[213] J. L. D. Neto *et al.*, "ULOOF: A user level online offloading framework for mobile edge computing," Working Paper, Jun. 2017.

[214] J. L. D. Neto, D. F. Macedo, and J. M. S. Nogueira, "Location aware decision engine to offload mobile computation to the cloud," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Istanbul, Turkey, Apr. 2016, pp. 543–549.
[215] C. Shi *et al.*, "COSMOS: Computation offloading as a service for mobile devices," in *Proc. 15th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Philadelphia, PA, USA, 2014, pp. 287–296.
[216] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 945–953.
[217] K. Ha, P. Pillai, W. Richter, Y. Abe, and M. Satyanarayanan, "Just-in-time provisioning for cyber foraging," in *Proc. 11th ACM Annu. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Taipei, Taiwan, 2013, pp. 153–166.

Dianlei Xu received the B.S. degree from Anhui University, Hefei, China, and is currently pursuing the degree with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China.

His research interests include mobile opportunistic networks, device-to-device communication, mobile social networks, and edge/fog computing.

Yong Li (M'09–SM'16) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012.

From 2012 to 2013, he was a Visiting Research Associate with Telekom Innovation Laboratories and the Hong Kong University of Science and Technology. From 2013 to 2014, he visited the University of Miami, FL, USA, as a Visiting Scientist. He is currently a Faculty Member of electronic engineering with the Tsinghua University. His research interests are in the areas of networking and communications, including mobile opportunistic networks, device-to-device communication, software-defined networks, network virtualization, and future Internet. He has published over 100 research papers, and has ten granted and pending Chinese and International patents.

Dr. Li was a recipient of the Outstanding Post-Doctoral Researcher Award, the Outstanding Ph.D. Graduates Award, and the Outstanding Doctoral Thesis Award from Tsinghua University. His research is funded by the Young Scientist Fund of Natural Science Foundation of China, Post-Doctoral Special Find of China, and industry companies of Hitachi and ZET. He has served as the Technical Program Committee Chair for WWW workshop of Simplex 2013, served as the TPC of several international workshops and conferences. He is also a Guest-Editor for *Mobile Networks and Applications* (ACM/Springer), Special Issue on *Software-Defined and Virtualized Future Wireless Networks*. He is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*.

Xinlei Chen received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, China, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, Carnegie Mellon University, USA. His research interests are in the areas of networking and communications (millimeter wave communications, device-to-device communication, medium access control), mobile embedded system (collaborative drone swarms), and big data.

Jianbo Li received the B.S. and M.S. degrees in computer science from Qingdao University, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer science from the University of Science and Technology of China in 2009.

He is currently a Professor with the Computer Science and Technology College, Qingdao University. His research interests include opportunistic networks, data offloading techniques, and intelligent city.

Pan Hui (M'03–SM'05–F'18) received the Ph.D. degree from Computer Laboratory, University of Cambridge, and the bachelor's and M.Phil. degrees from the University of Hong Kong. He is the Nokia Chair Professor of data science and a Professor of computer science with the University of Helsinki. He is also the Director of the HKUST-DT System and Media Laboratory, Hong Kong University of Science and Technology. He was an Adjunct Professor of social computing and networking, Aalto University from 2012 to 2017. He was a Senior Research Scientist and then a Distinguished Scientist for Telekom Innovation Laboratories, Germany, from 2008 to 2015. His industrial profile also includes his research with Intel Research Cambridge and Thomson Research Paris from 2004 to 2006. His research has been generously sponsored by Nokia, Deutsche Telekom, Microsoft Research, and China Mobile. He has published over 200 research papers and with over 13 500 citations. He has 29 granted and filed European and U.S. patents in the areas of augmented reality, data science, and mobile computing. He has founded and chaired several IEEE/ACM conferences/workshops, and has been serving on the organizing and technical program committee of numerous top international conferences including ACM SIGCOMM, MobiSys, IEEE INFOCOM, ICNP, SECON, MASS, Globecom, WCNC, ITC, IJCAI, ICWSM, and WWW. He is an Associate Editor for the leading journals the IEEE TRANSACTIONS ON MOBILE COMPUTING and the IEEE TRANSACTIONS ON CLOUD COMPUTING. He is an ACM Distinguished Scientist, and a member of the IEEE Computer Society Fellow Evaluation Committee 2018.

Sheng Chen (M'90–SM'97–F'08) received the B.E. degree in control engineering from East China Petroleum Institute, Dongying, China, in 1982, the Ph.D. degree in control engineering from City University, London, U.K., in 1986, and the Doctor of Sciences degree from the University of Southampton, Southampton, U.K., in 2005.

From 1986 to 1999, he held research and academic appointments with the University of Sheffield, Sheffield, U.K., the University of Edinburgh, Edinburgh, U.K., and the University of Portsmouth, Portsmouth, U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, where he holds the post of Professor of intelligent systems and signal processing. He has published over 550 research papers. His research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, and evolutionary computation methods and optimization.

Prof. Chen was a recipient of the ISI Highly Cited Researcher Award in Engineering in 2004. He is a Distinguished Adjunct Professor with King Abdulaziz University, Jeddah, Saudi Arabia. He is a fellow of the United Kingdom Royal Academy of Engineering and IET.

Jon Crowcroft (SM'95–F'04) received the graduation degree in physics from Trinity College, Cambridge University, U.K., in 1979, the M.Sc. degree in computing and the Ph.D. degree from University College London, U.K., in 1981 and 1993, respectively.

He is currently the Marconi Professor of communications systems with Computer Laboratory, University of Cambridge, U.K.

Prof. Crowcroft was a recipient of the ACM Sigcomm Award in 2009. He is a fellow of the United Kingdom Royal Academy of Engineering, ACM, and IET.