Maximized Cellular Traffic Offloading via Device-to-Device Content Sharing

Jingjie Jiang, Shengkai Zhang, Bo Li, *Fellow, IEEE,* Baochun Li, *Fellow, IEEE*

***Abstract*—In next-generation LTE-advanced cellular networks, Device-to-Device (D2D) communication has emerged as an effective way to offload cellular traffic and improve system performance. Conventionally, a device exclusively relies on cellular communication to retrieve the content it desires. With D2D communication, however, if the same piece of content is available in the vicinity of the device, the content can be directly retrieved from one of its neighbouring devices. Naturally, the key problem becomes how to maximize content sharing via D2D communication. Existing works on content sharing are mainly concerned with a multi-hop communication setting, while works on D2D communication have primarily focused on the communication aspects, including interference avoidance and energy efficiency. In this paper, we study the problem of maximizing cellular traffic offloading with D2D communication, by selectively caching popular content locally, and by exploring maximal matching for sender-receiver pairs. Specifically, we consider an interference-aware communication model and formulate selective caching as a Knapsack problem, and sender-receiver matching as a maximum weighted matching problem in a bipartite graph. We propose decentralized algorithms to solve both problems, and our simulation results demonstrate that our algorithms are effective in maximizing cellular traffic offloading.**

***Index Terms*—Device-to-Device Communication, Content Sharing, Collaborative Caching**

I. Introduction

The demand for bandwidth in cellular networks is expected to surge exponentially over the next several years [1]. However, traditional ways of improving throughput within the context of cellular networks have suffered from two major challenges. *First*, increasing the physical-layer capacity of wireless links becomes difficult, since the physical-layer technologies used in 4G/LTE (Long Term Evolution) networks, including MIMO-OFDM ((Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing) with capacity-achieving codes and interference coordination, have approached their theoretical limits. *Second,* decreasing the cell size by using femto-cell networks is a viable, yet *costly*, approach for improving throughput, as the cost of providing backhaul connectivity of these small base stations needs to be taken into account.

Manuscript received May 21, 2013; revised November 15, 2013.

This research is supported in part by a grant from Hong Kong Research Grant Council under the contract 615613 and by a grant from ChinaCache Int. Corp. under the contract CCNT12EG01.

Jingjie Jiang, Shengkai Zhang and Bo Li are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Their email addresses are {jjiangaf, bli}@cse.ust.hk. Shengkai's email address is szhangaj@connect.ust.hk. Baochun Li is with the Department of Electrical and Computer Engineering, University of Toronto. His email address is bli@eecg.toronto.edu.

Device-to-Device (D2D) communication has recently attracted a substantial amount of attention from both industry and academia, mainly due to its unique advantages to offload cellular traffic, better system throughput, higher energy efficiency and robustness to infrastructure failures. In the industry, with its recent announcement of a new open-source cross-platform development platform called *AllJoyn*, Qualcomm has targeted to address many challenges that exist in enabling device-to-device communication. With the new Message Queue Telemetry Transport (MQTT) protocol designed specifically for D2D communication, the industry is well positioned to its real-world deployment.

Even with the recent focus on D2D communication in the industry, the existing academic literature on D2D communication (*e.g.,* [2], [3], [4], [5]) has still largely focused on how D2D communication can run efficiently as an underlay to cellular networks. In this paper, we wish to study how D2D communication can be used effectively to offload cellular traffic from the Internet, as devices request different pieces of content, referred henceforth as *messages*, over cellular networks. Being requested from the Internet, both the size and the popularity of these messages, ranging from breaking news to stock quotes, can vary widely. With the assistance of D2D communication, if a neighbouring device happens to have the same message cached locally, it can be retrieved without incurring the cost of using cellular bandwidth. As an example, if a piece of breaking news has been requested by one device, it is likely that a nearby device may be interested in the same piece of news in the near-term future.

With a limited amount of storage on each device, the main challenge is how cellular traffic can be maximally offloaded by using D2D communication to satisfy requests for content, and to share messages between neighbouring devices. There are two aspects to this challenge. *First,* since the amount of local storage is limited, it is not feasible to cache all messages that a device has requested. The decision on whether or not a message should be cached needs to be made in a decentralized fashion, such that cached messages are most likely to be requested by neighbouring devices in the future. *Second,* for each message being requested, there may be multiple devices that are able to send this message; for each device receiving requests, there may be multiple messages it is able to send. In order to maximally offload cellular traffic, one would need to design a matching algorithm that optimally matches senders to receivers of messages in a decentralized fashion.

In this paper, we propose to address both of these challenges with decentralized algorithms in an interference-aware D2D communication environment. With respect to caching, by exchanging the interest for messages among neighbouring devices periodically, the popularity of messages are estimated locally, based on which caching decisions are made in a decentralized manner. With respect to matching senders to receivers, we formulate it as a maximum weighted matching problem, with D2D link rates as the weights assigned to each sender-receiver pair. Though the Hungarian algorithm [6] can be used to solve the matching problem, it is centralized in nature and requires global information. Rather, we propose to use an asynchronous and distributed algorithm [7] to solve this problem, which is at least 0.5 of the optimal matching. As compared to most related work on collaborative caching in multi-hop wireless networks, the upshot of our work is in its *best-effort* nature with decentralized algorithms. In our work, devices will not proactively cache any content to improve the cache hit ratio, and message exchanges are completely local, with no messages relayed over multiple hops. We believe that such best-effort decentralized algorithms are simpler and much easier to be implemented, and are therefore feasible in practical real-world D2D communications.

The remainder of the paper is organized as follows: In Sec. II, we present our system model and a high-level description of our problem. In Sec. III and IV, we present our decentralized solutions to the caching and matching problems, respectively. We demonstrate the effectiveness of our algorithms through simulations in Sec. V. The extensions and implementation details of our algorithms are discussed in Sec. VI. We discuss our contributions in the context of related work in Sec. VII before the conclusion in Sec. VIII.

II. System Model and Problem Overview

In this section, we present our system model and give a high-level overview of the problem that we wish to solve in this paper. In general, we wish to offload cellular traffic by allowing devices to retrieve a piece of content, called a *message* in this paper, from a nearby device rather than from the Internet. In order to assist nearby devices, a device will need to cache selected messages locally, taking advantage of its limited storage capacity. Our objective is to maximize such cellular traffic offloading using fully decentralized algorithms.

A. System Model

For the sake of simplicity, we only consider the scenario of a single cell in a cellular network, and ignore the influence from devices in adjacent cells. A central base station (BS) is reachable from any device within the cell, and is responsible to relay requests from each device to the Internet, in order to retrieve messages of interest to the device. Each device makes independent decisions when it requests for messages.

We assume that there are n devices independently and uniformly distributed within the cell. Let $\mathcal{D} = \{D_1, ...D_n\}$ denote the set of devices and $\mathcal{M} = \{M_1, ..., M_m\}$ the set of messages potentially desired by devices. Any two devices with a distance at most R apart can communicate with each other. For a device D_i, we denote the set of its neighbouring devices as N_i in Eq. 1, where R is the communication range. The signaling mechanism proposed by Choi *et al.* in [8] can be employed to discover the neighbouring devices of any given device and acquire its neighbour set. Each device can communicate with its neighbouring devices directly, *i.e.,*

$$N_i = \{D_j \in \mathcal{D} : ||D_i - D_j|| < R\}. \tag{1}$$

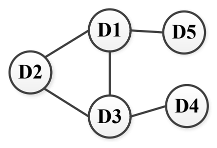
Effective mechanisms exist to allocate radio resources for both cellular and D2D communication links [5]. We can thus suppose that no interference exists between cellular and D2D communication. In addition, the Orthogonal Frequency Division Multiple Access (OFDMA) technique proposed by the 802.16 Task Group e (TGe) and other chunk-based approaches [9] [10] can be utilized to schedule D2D links and avoid mutual-interference. Nevertheless, it is still possible that two concurrent links at a single device will conflict with each other. Fortunately, Choi *et al.* in [11] designed a reliable transceiver to enable full-duplex communication. Based on this work, we can suppose that each device can transmit and receive data simultaneously, but at most one outgoing link from the device and one incoming link to the device can exist. It follows that a device can act as a sender and receiver at the same time with the limitation that the sending and receiving procedure are both dedicated to another end host respectively.

Furthermore, we stipulate that a device does not behave proactively, in that it will not *prefetch* any of the messages that have been requested by its neighbours but is not desired by itself. A device only assists its neighbours in a way that the messages retrieved by itself earlier can be reactively cached and later shared to others. In other words, the cached messages are actually a subset of its requested messages. In addition, we impose the restriction that each message is cached and transmitted in its entirety, and may not be split.

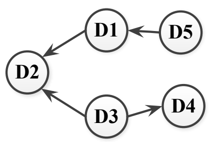
B. Problem Description

With D2D content sharing, each device can act both as a message sender and receiver. For a receiver, it first inquires its neighbours whether they have the desired message. If some idle neighbours have the message, the receiver can select one of them as its sender. Otherwise, the request is resent to the BS After the new message is completely received, the receiver needs to decide whether to cache this message or not. As a sender, the device may receive multiple requests from its neighbouring devices at the same time. But each time it can only send data to one receiver. Therefore, it has to select one of the potential receivers to send data to.

It is clear that for each device, there are two types of decisions to be made: (1) Which message needs to be cached; and (2) which device it should communicate with. Our problem is to find the effective criteria for making the decisions in order to meet our objective of maximizing cellular traffic offloading. With respect to which message should be cached, we should estimate the potential *value* of each message to the neighbouring devices, and predict the amount of benefit the local neighbourhood can obtain if a message is cached. With such a prediction, each device may then cache the most valuable messages within its limited storage. With respect to selecting neighbouring partners to communicate with, we further illustrate this problem with an example of five devices in a D2D communication network, as shown in Fig. 1.



(a) A D2D Network



(b) Communication Graph

Fig. 1. (a) An example of D2D communication networks; (b) The communication graph in the D2D network.

In Fig. 1(a), two devices are connected by an edge if they are neighbours. A directed edge from D_i pointing to its neighbour D_j exists in the communication graph if D_j holds the message requested by D_i. In Fig. 1(b), D_1 holds a message desired by D_5. D_2 holds the messages requested by D_1 and D_3. D_4 can also satisfy the same demand from D_3. We next show that finding the optimal matching for sender-receiver pairs to maximize the concurrent communication links is non-trivial, even in such a simple scenario. For now, we assume all the potential links in the graph are of the same quality and thus D_2 arbitrarily chooses D_1 or D_3 as its receiver. If D_2 selects D_1, D_3 can turn to D_4 for help and the request from D_5 can be satisfied by D_1. As a result, there are three concurrent D2D links, which is optimal in this graph. However, if D_2 happens to select D_3, the demand of D_1 cannot be fulfilled by any other device and thus only two links are set up in the communication graph.

III. Interference-aware Collaborative Caching

In the caching mechanism, each device reactively caches messages. Therefore, the cached messages have all been read by the device at an earlier time. To contribute the most to the neighbourhood, the device would have to cache all the messages it has retrieved. However, the storage capacity of a device is limited. A device needs to select a subset of its retrieved messages to be cached in its local storage.

A. Problem Formulation

We formulate the problem to select messages as a 0-1 Knapsack problem. Given a set of items, each with a weight and a value, the Knapsack problem tries to determine whether to include each item into a collection, so that the total weight is no more than the capacity of the knapsack and the total value of the selected items is as large as possible. In the context of our caching problem, the value of each message is the expected traffic that can be offloaded if the message is transmitted locally using D2D communication links. The weight of a message is its size and the cache capacity of each device is the weight limit. Since devices only have limited storage capacities, it is impossible to cache all the messages it has received. The devices have to make local decisions to select some messages to cache. In this case, each device will choose the more valuable messages and drop other messages.

For a more formal treatment of the caching problem, we define a set of 0-1 variables X_{ik}, where X_{ik} equals to one when the message M_k is cached by the device D_i. The value of P_{ik} indicates the probability that one of D_i's neighbours requests for M_k. The value of message M_k is thus $P_{ik}|M_k|$, where $|M_k|$ is the size of M_k. Denote the cache capacity of D_i as c_i. The optimization problem to maximize the aggregated caching value in device D_i can be defined as:

$$\begin{aligned} \max \ & \sum\nolimits_{M_k \in \mathcal{M}} X_{ik} P_{ik} |M_k| \\ \text{subject to } & \sum\nolimits_{M_k \in \mathcal{M}} X_{ik} |M_k| \leq c_i, \quad \forall 1 \leq i \leq n \\ & X_{ik} \in \{0,1\} \end{aligned} \tag{2}$$

The problem above is NP-hard, even if the value of each message is *priori* knowledge. In practice, however, the value of a message cannot be known beforehand due to the limited scope of information and the time-dependent nature of message values. It follows that the optimal solution to the current system is unnecessarily the best to serve the upcoming requests. To tackle this problem, we need to estimate the value of each message. It is equivalent to predicting the probability that a message will be requested in the future. As we consider the behaviours of mobile devices only in a short time period, historically statistical data can provide a good reference on the trend of the future requests.

To better estimate the probability a message is desired in the local neighbourhood, each device D_i records the local request information with respect to each message. Denote the number of times a message M_k is requested by some device in N_i as t_{ik}. Since every request generated by a neighbouring device will reach D_i, D_i can keep track of all the requests in its neighbourhood. Based on such statistical information, we can define the local popularity P_{ik} of a message in Eq. 3, which is the most important indicator of the local trend. $M(N_i)$ is the set of messages that have been requested in the neighbourhood.

$$P_{ik} = \frac{t_{ik}}{\sum_k t_{ik}}, \ \forall M_k \in M(N_i) \tag{3}$$

Although the trend of local requests is usually consistent, the local interest may change abruptly. For instance, a piece of breaking news might become remarkably popular and incur a burst of local requests. To better respond to such cases, we allow devices to collect the global popularity information of a message when the message is transmitted from the base station. For a globally popular message, most of the devices in the local area might not have cached it. Nevertheless, once a local device retrieves the message from the base station, the global popularity will be attached to it and this piece of information can spread very rapidly in the neighbourhood through frequent information exchanges. As a result, most local devices will consider the message to be valuable and will cache it with high probability. Later requests to the same message are likely to be satisfied locally.

To summarize, the estimation of the local popularity of each message works as follows. If message M_k is requested by one of the neighbours, the device D_i increases t_{ik} by 1. When D_i receives a new message from the neighbour D_j, the local request count t_{jk} of M_k in D_j's neighbourhood is attached to the message. Then D_i updates its local request count t_{ik} as

$$t_{ik} = \alpha t_{ik} + (1-\alpha) t_{jk}, \ \alpha \in [0,1] \tag{4}$$

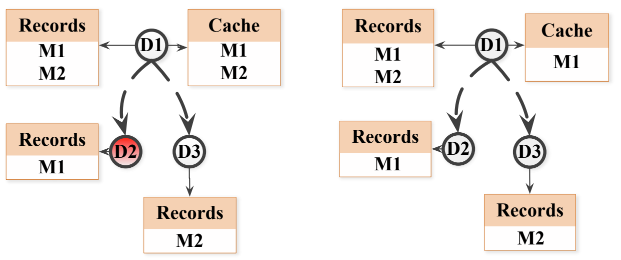


Fig. 2. The motivation for interference-aware caching: an example.

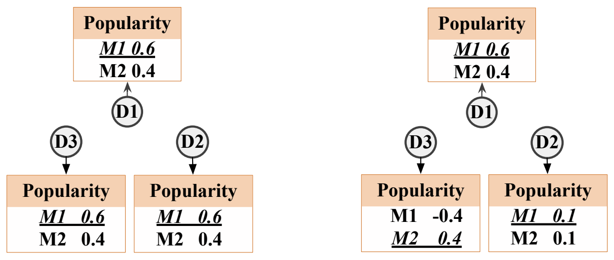


Fig. 3. The motivation for collaborative caching: an example.

When D_i receives a new message from the base station, the global popularity P_k of the new incoming message is attached and D_i updates its local request count t_{ik} as:

$$t_{ik} = \alpha t_{ik} + (1-\alpha) P_k |N_i| \quad (5)$$

where $|N_i|$ is the number of D_i's neighbours. After recording the number of local request for each message, D_i can calculate the local popularity P_{ik} for each message M_k.

B. Interference-Aware Estimation of Popularity

With wireless communication, each device cannot receive multiple messages simultaneously. In other words, if a device is currently a receiver, it cannot generate any more requests unless the current receiving process finishes. This is of great importance for a device to make caching decisions. In Fig. 2, D_1 can only cache one of the messages M_1 and M_2. Through historical requests, D_1 finds out that D_2 has retrieved M_1 and D_3 has retrieved M_2. The local popularities of the two messages are thus both 0.5. In an interference-unaware caching mechanism, D_1 believes that caching either message makes no difference, and it will therefore randomly choose M_1 or M_2 when making its caching decision.

Nevertheless, a neglected fact is that D_2 is already a receiver and cannot generate any request for a rather long period. It follows that only the potential request for M_1 is valid and caching M_2 would be much less valuable. In our mechanism, whenever a device makes a cache decision, it will inquire the current states of its neighbors. The device then realizes that no more requests can be sent from D_2 and the potential request from D_3 is more valuable. As a result, D_1 will cache M_1 in the interference-aware mechanism.

The key to enable the interference-aware estimation is to predict the desired messages of the receiving devices. Such messages are defined to be *weakly desired* in our framework. In contrast, messages that are not desired by receiving devices are defined to be *strongly desired*. As the neighboring devices share common interest, the popular yet unread messages are likely to attract a device. Based on this observation, we affirm that a message is weakly desired if it has not been requested by a receiving device. To avoid caching a weakly desired message instead of a strongly desired message, a weakly desired message is marked and will be replaced first if its popularity is similar to a strongly desired candidate. In this way, the weakly desired messages become less competitive when they contend for caching positions.

C. Enabling Collaborative Caching

Information exchange can help devices to identify popular messages timely and facilitate the devices to cache the most popular messages that are likely to be requested by neighbouring devices. Nevertheless, since the devices in a neighbourhood share similar popularity information, they tend to cache the same set of popular messages. The following example exposes the defect of an independent caching policy.

Consider Fig. 3, where three devices with unit cache capacity reside in the local network and the popularities of the messages M_1 and M_2 with unit size are 0.6 and 0.4, respectively. Initially, the caches of the three devices are all empty. In Fig. 3(a), the devices make their own caching decisions independently and all cache M_1. As a result, the local message availability is low: if a device desires M_2, none of the local devices can satisfy this request. The reason for this phenomenon is the imbalance between the demand for a message and the proportion of storage used to cache this message. Two out of three devices caching M_1 is sufficient to satisfy future requests, which implies that the third replica of M_1 is of little value. In contrast, the probability to ask for M_2 is as high as 40%, whereas the storage used to cache M_2 is zero. The gap between demand and supply decreases local message availability and caching value.

To tackle this problem, collaborative caching is integrated into our mechanism. Instead of caching the most popular messages, we enable devices to collect the information of its neighbours' caches and choose the messages with the biggest gap between its popularity and caching proportion. In Fig. 3(b), when device D_1 starts to collect information, the caches of the other two devices are still empty. It finds out the cache proportions of the two messages are both zero, which means that the difference between the popularity and the cache proportion is 0.6 for M_1 and is 0.4 for M_2. D_1 chooses to cache M_1. At the time D_2 collects information, D_1 has cached M_1 and the cache of D_3 is empty. Then the cache proportion of M_1 becomes 1. Suppose that within a short time period, the popularities of the messages remain unchanged. The gaps of M_1 and M_2 become -0.4 and 0.4. D_2 decides to cache M_2 accordingly. Later when D_3 tries to cache a message, it will discover that the cache proportions of the two messages are both 0.5 and it will cache M_1. After this caching cycle, the overall cache proportions of M_1 and M_2 are 0.67 and 0.33, which matches the distribution of local popularity much better.

Each time a device D_i makes a caching decision, it will generate a broadcast message to all of its neighbours, querying about the messages currently cached in their storage. Once the caching information is collected, the device is able to calculate the proportion of storage used to cache each message M_k, denoted as CP_{ik}. The gap between the local popularity and the caching proportion, denoted as G_{ik}, can be calculated as $P_{ik} - \text{CP}_{ik}$. We revise the *value* of M_k as $G_{ik}|M_k|$, indicating the benefit D_i will get if it caches M_k locally.

D. Distributed Caching Algorithm

A device makes separate caching decisions under two circumstances: 1) When the cache is not full, always cache the received message; 2) when the cache is already full, apply the cache replacement algorithm to select a subset of the messages to cache. For the latter cases, the devices should run the cache replacement algorithm to remove less valuable messages and to make room for more beneficial messages. Suppose there are c messages currently cached in the device D_i. After retrieving a message (denoted as message M_{c+1}) from the base station or some neighbouring device, D_i updates the caching information and communication states of all its neighbouring devices. After collecting all the necessary information, the new popularities of all the $(c+1)$ messages can be computed. D_i can then determine the caching gaps of all the candidates and run the replacement algorithm to determine the messages to be cached.

Here, we apply the classical greedy algorithm for the Knapsack problem and sort the messages in decreasing order of their unit values. Equivalently, the messages are sorted according to the local caching gaps. The complexity of this algorithm is dominated by its sorting process, which is $O(c \log c)$.

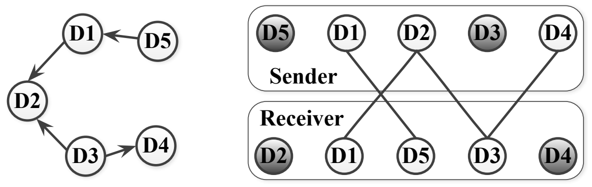
Algorithm 1 Greedy Replacement

```
1: $H_k \leftarrow G_{ik}$ ▷ Monitor the cache gaps (value) of messages
2: Sorting messages in decreasing order of $H_k$
3: $W_i \leftarrow c_i, w_k = |M_k|$, $C_i \leftarrow \Phi$
4: for $k = 1$ to $c + 1$ do
5:     if $w_k < W_i$ then
6:         $W_i = W_i - w_k$
7:         $C_i = C_i \cup M_k$
```

With our collaborative and interference-aware caching mechanism, the devices always cache the messages that will potentially contribute most to the system.

IV. Matching Senders to Receivers

Based on the messages provided in D2D network, a device can find its desired message in its neighbouring devices and satisfy the request locally. However, as we have previously mentioned, if multiple neighbours can send this message, the device has to select one of them as its sender. Similarly, once a device starts sending data, it cannot set up another outgoing communication link. It is thus necessary for the device to scrutinize the incoming requests and to select one of them to construct a communication link. This observation inspires us to formulate the problem of selecting communication partners as a maximum weighted matching problem, with link rates as the weights assigned to each sender-receiver pair.



(a) Communication graph (b) Bipartite system view

Fig. 4. Conversion from Communication Graph to Bipartite Graph

A. Matching for Communication Pairs

Each sender-receiver pair corresponds to a directed edge in the communication graph. In the context of content sharing, senders and receivers are both local devices. Therefore, the communication graph $G = (V, E)$ is non-bipartite. But we can reconstruct an undirected bipartite graph $G' = (V', E')$ to depict the same system, where V' consists of two disjoint S_v and R_v. S_v contains the nodes representing all the senders, and R_v represents all the receivers. As each device can be a sender or a receiver at any time, $|S_v| = |R_v| = n$. Furthermore, the undirected edges in G' are constructed as follows:

$$e = [D_i, D_j] \in E', \text{ iff } (D_i, D_j) \in E, \ \forall D_i \in R_v, D_j \in S_v \quad (6)$$

Fig. 4 illustrates the transition from a communication graph, as shown in Fig. 1(b) previously, to a bipartite graph of a D2D network. Let's revisit the demand and supply of messages in the communication graph: D_1 holds a message desired by D_5. D_2 holds the messages requested by D_1 and D_3. D_4 can also satisfy the same demand from D_3. In this case, D_1, D_2 and D_4 act as the message senders, whereas D_1, D_3 and D_5 are message receivers. The sender set actually consists of three devices, whose incoming degrees in the communication graph are nonzero and the other two devices can be viewed as isolated nodes in the bipartite graph. The same rule applies for the receiver set. To meet the interference restrictions, at most one of the edges incident to a node in the bipartite graph can be selected. We observe that the selection procedure yields a maximum matching problem. Apart from maximizing the number of requests satisfied locally, user experience is another important factor to select the communication links. Among all the senders a receiver can retrieve its desired message, the receiver would like to select the sender with the highest transmission rate on the link between them. In accordance with this requirement, each edge in the bipartite graph is further associated with a weight, which is the transmission rate of the corresponding link. We can then formulate the problem of selecting communication partners as:

$$\begin{aligned} \max \ & \sum_{i=1}^{n}\sum_{j=1}^{n} L_{ij} \cdot r_{ij} \\ \text{subject to } & \sum\nolimits_{j=1}^{n} L_{ij} \le 1, \qquad \forall 1 \le i \le n \\ & \sum\nolimits_{i=1}^{n} L_{ij} \le 1, \qquad \forall 1 \le j \le n \\ & L_{ij} \in \{0, 1\}, r_{ij} \ge 0 \end{aligned} \quad (7)$$

Here, L_{ij} indicates whether a communication pair (D_i, D_j) is actually selected and r_{ij} represents the available rate of the link between D_i and D_j. Note that if a device is receiving a message, its downlink rate is zero. Similarly, the uplink of a sending device is also zero. The optimization problem above is to find the maximum weighted matching in a bipartite graph. We can add dummy edges $e = [D_i, D_j]$ with zero weight if there is no edge between D_i and D_j in G'. In this way, we can convert the bipartite graph with n vertices in each partition into a complete bipartite graph. Correspondingly, the optimization problem is to find the optimal matching in this graph.

B. Matching as an Auction

Denote the maximum matching problem as the primal, then its dual problem defines a non-negative variable u_i for each device. We consider the procedure of mutual selections among senders and receivers as a multiple-goods auction, where the goods are D2D links. Each potential buyer D_j has its own valuation v_{ij} for the potential link sold by D_i. Essentially, the valuation for a link from D_i to D_j is the rate of this link, denoted as r_{ij}. The payoff of D_i is the price it successfully sells its link, namely u_i. If the receiver D_j wins the link sold by D_i, which means this link is selected in the optimal matching, its payoff u_j is defined as $v_{ij} - u_i$. It is clear to see that the maximization of overall payoffs is the same as the maximization of the weighted matching, as shown in Eq. 8.

$$\sum_{i=1}^{n} u_i + \sum_{i=1}^{n}\sum_{j=1}^{n} L_{ij}(v_{ij} - u_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} L_{ij} v_{ij} \tag{8}$$

Both the sender and receiver intend to maximize their payoffs. Initially, all the prices are set to be zero. Each receiver selects the senders that can maximize its payoff as the preferred senders. The edges between a receiver and its potential senders are built in the *Preferred Graph*, denoted as G_p. According to the *Weak Duality Theorem*, if a perfect matching can be found in G_p, then the matching is optimal. Otherwise, a *Constricted Set* of receivers exists in the current G_p. The senders that are neighbours of some receiver in the constricted set would raise their prices. The next round of auction then begins with new prices. The iterative procedure above is proved to terminate in polynomial time [6]. We can then obtain an optimal solution to the original maximum weighted matching problem.

However, the global information of a bipartite graph is needed in the Hungarian algorithm, which means that the optimal solution can only be found in a centralized fashion. It follows that the optimal construction of communication links can only be manipulated by the base station. Not only the computation of maximum matching in the global bipartite graph will consume a considerable amount of computation resources at the base station, but also the control and assignment procedures will generate extra traffic between the devices and the base station, adding additional burden to the cellular network. To make things worse, control packets being exchanged between the local devices and the base station may be lost or delayed when the congestion of cellular networks is severe. For these reasons, a distributed algorithm is desirable to solve the problem locally, without involving the base station.

C. Distributed Matching Algorithm

Matching is a classical problem in graph theory and has been widely studied [6]. We leverage the asynchronous and distributed algorithm in [7] to find the maximum weighted matching between senders and receivers. Through information exchange among neighbouring devices, the algorithm tries to reach a consensus between senders and receivers. The negotiation procedure goes as follows in our system.

Each device maintains two lists L_c and L_w. For a sender, devices in L_c are its neighbours in G' with unsatisfied requests. Similarly, for a receiver, devices in L_c are its neighbours in G' without involving in any sending procedure. D_i selects its candidate partner as Eq. 9 illustrates, sends a communication request to it, and then waits for its response. Devices in L_w are the neighbours who have selected D_i as the candidate partner, and are waiting for D_i to respond.

$$\text{candidate}(D_i) = D_j, \text{ iff } v_{ij} \geq v_{ik}, \ \forall D_k \in L_c \tag{9}$$

If the candidate device also selected D_i, namely, the two devices are the most valuable partner to each other, the negotiation succeeds and a link can be constructed. Both devices then send connection failure messages to all the other devices that are still waiting for their responses. If the candidate device prefer some other device, D_i will receive a failure message and exclude this device from its list of available neighbours. D_i then has to select the next valuable device as its new candidate, and repeat the procedure above.

Algorithm 2 Distributed Matching Algorithm for D_i

```
1:  $L_c \leftarrow N_i^{'}$, $L_w \leftarrow \Phi$
2:  Send $\langle$Req$\rangle$ to $c$
3:  while $L_c \neq \Phi$ & $\exists D_j \in L_c$, s.t $r_{ij} > 0$ do
4:      $c \leftarrow$ candidate$(D_i)$
5:      Receive $\langle$Msg$\rangle$ from $D_j$
6:      if $\langle$Msg$\rangle = \langle$Req$\rangle$ then
7:          $L_w \leftarrow L_w \cup D_j$
8:      if $\langle$Msg$\rangle = \langle$Fail$\rangle$ then
9:          $L_c \leftarrow L_c \setminus D_j$
10:     if $c \in L_w$ then
11:         for $D_k \in L_w \setminus c$ do
12:             send $\langle$Fail$\rangle$ to $D_k$
13:         $L_c = \Phi$
```

$N_i^{'}$ is the neighbor set of D_i in G'. Recall that the available uplink rate of a sending device is zero. Therefore, the sending devices will not be further considered as possible senders. As there is always a sender-receiver pair with the highest link rate, the two devices will first reach a consensus and establish a communication link successfully. Then the devices connected by the second fastest link will go through the same procedure. It is now clear to see that this algorithm actually imitates the global greedy algorithm precisely and thus can achieve an approximation ratio of 0.5. A proof of this conclusion can be found in [7]. After the lists of available neighbours of all the devices become empty, the algorithm terminates and we will have a solution to the weighted matching problem as the guidance for matching senders to receivers.

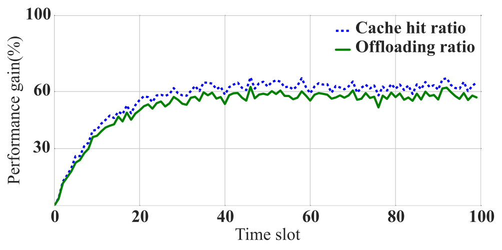


Fig. 5. System performance with an average message size of 10, a standard deviation of 1, and a storage capacity of 100.

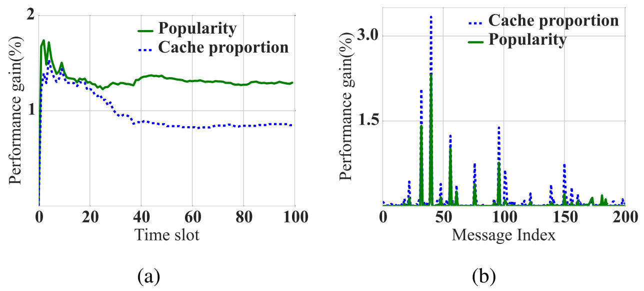


Fig. 6. (a) The popularity and cache proportion of a popular message. (b) Comparison of the popularity and cache distribution of 200 messages.

V. Simulation Results

In this section, we evaluate the performance of our system based on the implementation in a time-slotted simulator using C++. We assume that no mobility of users occurs during the simulation period. 1000 devices are uniformly randomly distributed in one cell. For simplicity, we assume devices have equal storage capacities and the rates of D2D links are identical. Each non-receiving device independently sends a message request with a probability of 50% in each time slot. The popularity of messages in the repository follows a *Zipf*-like distribution as previous studies (*e.g.* [12], [13]) have demonstrated. We do not restrict our system to any specific fading model for D2D communication. Instead, we examine our system under both the UDG (Unit Disk Graph) radio model [14] and the Quasi-UDG radio model [15] to depict the irregular communication area caused by multipath induced fading and shadow fading. Furthermore, the size of the messages follows a normal distribution. We examine the *cache hit ratio* and the *offloading ratio* of our system. The *cache hit ratio* indicates the percentage of requests that result in cache hits in neighbouring devices, and the *offloading ratio* is the ratio between D2D traffic and overall traffic for transmitting all the messages requested within the simulation period.

A. Overall Performance

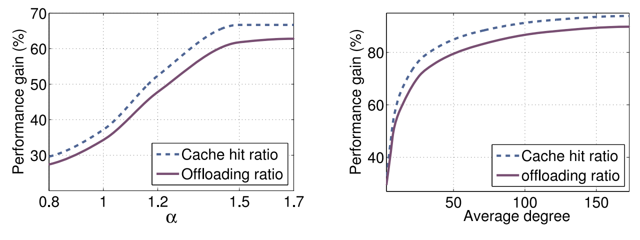
Fig. 5 gives a first glance of the system performance over 100 time slots. Initially, the caches of all devices are empty and all requests are responded to by the base station. After 10 time slots, the caches of most devices are saturated and the cache replacement algorithm comes into effect. It can be seen that our system can offload cellular traffic significantly: 60% network traffic is transmitted through D2D communication. This figure also reveals that the convergence of the system performance is fast. After 20 time slots, the cache hit ratio and the offloading ratio stabilizes around 60%. Such a high D2D response rate demonstrates that our system can effectively share the burden of the base station through D2D communication. In what follows, the cache capacities of devices are fixed to be 100 if no explicit specification is made.

In Fig. 6(a), we focus on the transient response of the content sharing system to a message that has recently become popular. The top curve illustrates the varying percentage of requests generated for the message, and the bottom curve presents the changing proportion of devices that has this message cached. The result demonstrates that our design is very agile to a surge in popularity, and can satisfy most of the requests for a popular message locally. We further make a comparison between the popularity distribution of the messages and their caching proportions in all devices. We sample 200 messages and collect their caching and request information after the system stabilizes. From Fig. 6(b), we can see that although some gaps exist between the two distributions, their tendencies match very well, which is consistent with our design objective to mitigate the gap between content demand and local supply and thus efficiently use the cache capabilities of devices to serve local requests.

B. Influence of the Request Commonality

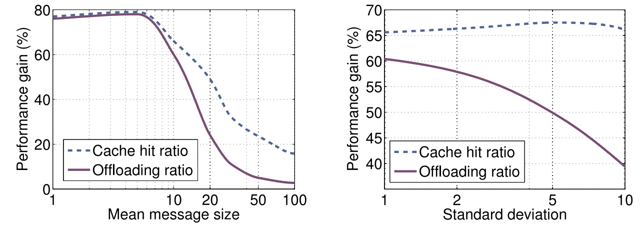
The commonality of local content requests depends both on the popularity distribution of messages and the communication range of devices. We first evaluate the system performances under different popularity distributions in Fig. 7(a). *Zipf*-like distributions have relative probability of a hit for the i^{th} most popular message proportional to $i^{-\alpha}$. It is clear to see that cache hit ratios and offloading ratios highly depend on the popularity distribution. The two metrics vary from from 30% to 60% with different values of α. Trace-based analysis has found that α typically ranges from 0.9 to 0.97 [16] in common university and enterprise networks. In the context of D2D networks, we believe that α could be larger with the prevalence of location-based information services. Even if the commonality in a neighbourhood is limited and the redundancy of network traffic is small, the content sharing system can still offload the redundant traffic effectively.

We explore the system performance under different communication ranges in Fig. 7(b). Obviously, with a larger communication range, a device has more neighbouring devices. As a result, it is more likely to find the desired messages in the neighbourhood. When the average number of a device is 5.5, there are 15 isolated devices in the network. Such isolated devices will always send requests to the base station. The cache hit ratio and offloading ratio are only about 40%. When the average number of neighbors reaches 8.3, we encounter the threshold of keeping the network connected. In this situation, the offloading ratio reaches 50.9%. When the network becomes denser, the cache hit ratio and the offloading ratio keep increasing significantly. However, the larger communication range implies a higher possibility to introduce mutual-interference. The largest possible communication range depends on the underlay scheduling scheme and devices' battery life.



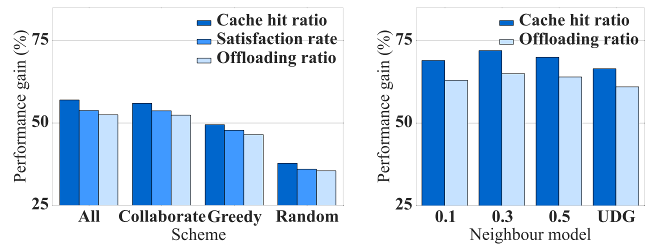
(a) (b)

Fig. 7. (a) System performance with different popularity distributions. (b) System performance under various communication ranges, with an average message size of 10, a standard deviation of 1.



(a) (b)

Fig. 8. (a) System performance with varying mean values. The standard deviation is fixed as 1. (b) System performance with varying standard deviations. Average message size is fixed as 10.



(a) (b)

Fig. 9. (a) System performance with varying mean values. The standard deviation is fixed as 1. (b) System performance with varying standard deviations. Average message size is fixed as 10.

C. The Size Distribution of Messages

We conduct several groups of experiments under different standard deviations and mean values of the size distribution in Fig. 9(a). As the storage capacity is fixed, the mean size of messages actually represents the relative storage capacity of each device. For a larger mean value, the cache hit ratio and offloading ratio are lower due to the relatively small storage capacity of devices. In contrast, the smaller mean value implies a relatively stronger storage capacity to cache more valuable messages in the system. Consequently, the cellular offloading ratio is much higher. When the average size is 20, namely, each device caches 5 messages on average, the cache hit ratio is still 49% and the offloading ratio is 24%.

Fig. 9(b) shows the system performance under different standard deviations of message sizes. Interestingly, the larger the standard deviation, the greater the gap between the cache hit ratio and the offloading ratio. As the content availability is restricted by the storage capacity of devices, it is easier to fit the small messages into the cache, whereas the larger messages are more frequently dropped since they are at a higher risk of overflowing the storage. With a large standard deviation, the average size of messages cached in the devices is smaller than the average size of all the requested messages. Therefore, although the cache hit ratio remains high, the offloading ratio decreases with larger standard deviations since most of the locally unsatisfiable messages are usually large and incur substantial amount of cellular traffic.

D. Comparison of Different Algorithms

We conduct a series of experiments to compare the performance gains under different caching algorithms. The first group of data in Fig. 9(b) is collected using our interference-aware collaborative caching strategy. The second group only uses collaborative caching strategy without considering interference. The third group applies naive popularity-based approach, which greedily caches the locally popular messages. Devices in the last group randomly kick out a message. From Fig. 9(b) we can see that our caching approach improves the system performance significantly. The cache hit ratio and offloading ratio increase by 19.3% and 17.1% respectively compared to the baseline method, and increase by 7.3% and 5.7% in comparison with the popularity-based method. Taking possible interference into consideration, however, do not have obvious influence on the system. We can eliminate this part from the caching algorithm to save computation overhead.

To examine the approximation ratio of the matching algorithm, we further define *satisfaction rate* to indicate the percentage of requests that are satisfied through D2D communication. It is obvious that the cardinality of the optimal matching is less than or equal to the number of cache hits. It follows that the cache hit ratio acts as an upper bound of the satisfaction rate essentially. From Fig. 9(b) we can see that the satisfaction rates are all greater than 90% of the corresponding cache hit ratios in all the four groups of data. We can conclude that the approximation ratio of the matching algorithm is much larger than the theoretical bound (*i.e.* 0.5).

E. Impact of Radio Models

To highlight the robustness of our method, we evaluate the system with a more realistic radio model, Quasi-UDG, which has a tunable parameter $0 < \alpha < 1$ to control the level of irregularity of the radio range. In the Quasi-UDG model, a link between D_i and D_j exists with probability Pr_{ij} as

$$\text{Pr}_{ij} = \begin{cases} 1, & |D_i, D_j| \leq (1-\alpha)R \\ \eta, & (1-\alpha)R < |D_i, D_j| \leq (1+\alpha)R \\ 0, & |D_i, D_j| > (1+\alpha)R \end{cases} \tag{10}$$

where $0 < \eta < 1$ and $|D_i, D_j|$ denotes the Euclidean distance between D_i and D_j. The larger the α, the more uncertain that a link between two nearby nodes is likely to exist, which mimics the irregular radio patterns found in reality.

We can see that our system is insensitive to the change of the radio models and can achieve satisfying performance under different circumstances, which makes our system strong enough to be extended to real-world applications.

VI. Discussions

Although our solution is discussed under the single-cell assumption, it can be extended to a multi-cell environment without major modifications. We only need to focus on the devices that reside near the boundary of a cell. When such devices try to discover their neighbours, it is possible that some of the neighbors currently subscribe to a different base station in an adjacent cell. As the D2D communication utilizes separate radio resources from the cellular communication, a D2D link between two devices subscribed to different base stations will not influence the cellular communication. The D2D link between them, however, should be allocated to one of the cells so that the corresponding wireless scheduler in the assigned cell will treat this link as the other links in the cell equally. Obviously, this situation will not influence our caching mechanism and matching algorithm, and thus no change is needed for the multi-cell extension.

To implement the D2D content sharing system, devices need to implement the caching and the matching algorithm locally. The caching mechanism can collect and record popularity and caching proportion information of messages in the neighbourhood, and help devices to make cache decisions accordingly. The matching algorithm guides devices to find their partners and set up D2D links. Devices without the caching and matching mechanism cannot utilize D2D communication, and only communicate with the base station directly.

VII. Related Work

Caching is a traditional approach to improve content accessibility in wireless networks. Yin and Cao in [17] designed a scheme to enable distributed caching. Based on the size of the passing-by message, an intermediate node caches the small passing-by messages and caches the path to the nearest node that stores the large messages. As their concern is how to route the request to the nearest node, their perspective is completely different from ours, in that we focus on which subset of messages should be cached in each device, rather than where to find these messages. Tang *et al.* in [18] presented an approximation algorithm to solve the cache placement problem and maximize the reduction of message access costs. However, they assumed implicitly that the access frequency of each node to a message is *priori* knowledge, but did not provide any mechanism to estimate such access frequency. In contrast, we effectively estimate the access frequency through exchanging the local interest for messages, and use the estimated probabilities to guide caching decisions.

Golrezaei *et al.* in [19] proposed an architecture for caching popular video content without any deployment of additional infrastructure. By identifying the conflict between collaboration distance and interference, they derived a scaling law analysis of the optimal distance, and gave a closed-form expression as a function of the modelling parameters. This work is largely theoretical without providing any mechanism to enable content sharing. Golrezaei *et al.* in [20] proposed a centralized caching mechanism in cellular networks. By introducing *helpers* with low-rate backhaul and high storage capacity, requests from devices can be satisfied by the nearby helpers. Due to the need for extra auxiliary facilities, this solution may be very costly. Through message sharing among neighbouring devices, we utilize the limited storage capacity of each device, and maximize D2D traffic without additional infrastructure. Besides, the centralized caching mechanism in [20] suffers from the performance bottleneck of a central controller and is thus not scalable, whereas our caching mechanism is completely distributed.

Interference can occur not only between two communication links with different sender-receiver pairs [21] [22] [23], but also among concurrent links at each sender or receiver. Existing work circumvents the difficulty of avoiding interference among concurrent links by introducing dedicated cache servers. It is nontrivial, however, to avoid interference in pure D2D networks. We propose a distributed algorithm to match message senders and receivers to tackle this problem. Our strategy to match the senders and receivers is to select the set of links to be constructed and scheduled. In other words, we deal with the self-interference problem rather than the mutual-interference problem, and as a result our work is orthogonal to existing works on wireless link scheduling.

VIII. Conclusion

In this paper, we designed and evaluated a distributed caching mechanism to effectively enable content sharing among mobile devices and maximize the offloading traffic from the cellular networks. Specifically, we consider an interference-aware environment and incorporate the collaboration among devices into the caching mechanism to improve local content availability. Furthermore, the strategy for matching senders to receivers subject to self-interference constraints is formulated as a classical maximum weighted matching problem, to which the optimal solution can be derived when network-wide information is known, and also an effective distributed algorithm with bounded approximation ratio can be applied. Our simulation results have shown that the proposed mechanism can offload traffic from the cellular networks significantly with quick convergence to a stable state. In addition, the system is very agile to a burst of popular content.

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.

[2] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 42–49, 2009.

[3] K. Doppler, C.-H. Yu, C. B. Ribeiro, and P. Janis, "Mode selection for device-to-device communication underlaying an LTE-advanced network," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2010.

[4] C.-H. Yu, O. Tirkkonen, K. Doppler, and C. Ribeiro, "Power optimization of device-to-device communication underlaying cellular communication," in *Proc. IEEE International Conference on Communications (ICC)*, 2009.

[5] P. Janis, V. Koivunen, C. Ribeiro, J. Korhonen, K. Doppler, and K. Hugl, "Interference-aware resource allocation for device-to-device radio underlaying cellular networks," in *Proc. IEEE Vehicular Technology Conference (VTC)*, 2009.

[6] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[7] J.-H. Hoepman, "Simple distributed weighted matchings," *arXiv preprint cs/0410047*, 2004.

[8] B. S. C. Choi and M. Gerla, "Wireless interrupt: Inter-device signaling in next generation wireless networks," in *Proc. IEEE INFOCOM*, 2010.

[9] H. Zhu and J. Wang, "Chunk-based resource allocation in OFDMA systems-part I: chunk allocation," *IEEE Trans. Communications*, vol. 57, no. 9, pp. 2734–2744, 2009.

[10] ——, "Chunk-based resource allocation in OFDMA systems-part II: joint chunk, power and bit allocation," *IEEE Trans. Communications*, vol. 60, no. 2, pp. 499–509, 2012.

[11] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. ACM Mobicom*, 2010.

[12] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, 1999.

[13] 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 Measurement Conference (IMC)*, 2007.

[14] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs ," *Discrete Mathematics*, vol. 86, no. 1, pp. 165–177, 1990.

[15] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad hoc networks beyond unit disk graphs," *Wireless Networks*, vol. 14, no. 5, pp. 715–729, 2008.

[16] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: findings and implications," in *Proc. ACM SIGMETRICS*, 2009.

[17] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.

[18] B. Tang, H. Gupta, and S. R. Das, "Benefit-based data caching in ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 3, pp. 289–304, 2008.

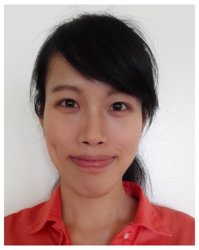
[19] N. Golrezaei, A. Dimakis, and A. Molisch, "Wireless device-to-device communications with distributed caching," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2012.

[20] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, 2012.

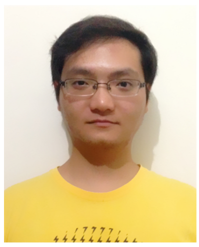
[21] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM Mobicom*, 2006.

[22] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM Trans. Networking*, vol. 17, no. 6, pp. 1846–1859, 2009.

[23] O. Goussevskaia, R. Wattenhofer, M. M. Halldórsson, and E. Welzl, "Capacity of arbitrary wireless networks," in *Proc. IEEE INFOCOM*.



Jingjie Jiang received the B.Engr. degree from the Department of Automation, Tsinghua University, China, in 2012. Since 2012, she has been with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, where she is currently a PhD candidate. She is now visiting the Department of Electrical and Computer Engineering at the University of Toronto. She is a member of IEEE. Her current research interests include: Device-to-Device communication, computing offloading, content distribution, datacenter networking and scheduling.



Shengkai Zhang received his B.Engr. degree in College of Physical Science and Technology, Central China Normal University, 2009. He received the M.S. degree from Huazhong University of Science and Technology in 2012, and the MPhil. degree from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology in 2014. He is currently a research assistant in the Department of Electronic and Computer Engineering of Hong Kong University of Science and Technology. His research concerns wireless networking, Wi-Fi localization and multi-sensor fusion for mobile robot applications.

Bo Li is a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He is a Fellow of IEEE. He was the Chief Technical Advisor for ChinaCache Corp.(NASDAQ CCIH), the largest CDN operator in China. He was a Cheung Kong Visiting Chair Professor in Shanghai Jiao Tong University (2010-2013) and an adjunct researcher in Microsoft Research Asia (1999-2007) and in Microsoft Advance Technology Center (2007-2009). His current research interests include: multimedia communications, the Internet content distribution, datacenter networking, cloud computing, and wireless sensor networks.

He made pioneering contributions in the Internet video broadcast with the system, Coolstreaming, which was credited as the world first large-scale Peer-to-Peer live video streaming system. The work appeared in IEEE INFOCOM (2005) received the IEEE INFOCOM 2015 Test-of-Time Award. He has been an editor or a guest editor for over a dozen of IEEE journals and magazines. He was the Co-TPC Chair for IEEE INFOCOM 2004.

He received five Best Paper Awards from IEEE. He received the Young Investigator Award from Natural Science Foundation of China (NFSC) in 2005, the State Natural Science Award (2nd Class) from China in 2011. He received his B. Eng. (summa cum laude) in the Computer Science from Tsinghua University, Beijing, and his PhD in the Electrical and Computer Engineering from University of Massachusetts at Amherst.

Baochun Li received the B.Engr. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995 and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, in 1997 and 2000.

Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently a Professor. He holds the Nortel Networks Junior Chair in Network Architecture and Services from October 2003 to June 2005, and the Bell Canada Endowed Chair in Computer Engineering since August 2005. His research interests include large-scale distributed systems, cloud computing, peer-to-peer networks, applications of network coding, and wireless networks.

Dr. Li has co-authored more than 290 research papers, with a total of over 13000 citations, an H-index of 59 and an i10-index of 189, according to Google Scholar Citations. He was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000. In 2009, he was a recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society, and a recipient of the University of Toronto McLean Award. He is a member of ACM and a Fellow of IEEE.