Cooperative Data Offloading in Opportunistic Mobile Networks

Zongqing Lu, Xiao Sun and Thomas La Porta
The Pennsylvania State University
{zongqing, xxs118, tlp}@cse.psu.edu

***Abstract*—Opportunistic mobile networks consisting of intermittently connected mobile devices have been exploited for various applications, such as computational offloading and mitigating cellular traffic load. Different from existing work, in this paper, we focus on cooperatively offloading data among mobile devices to maximally improve the probability of data delivery from a mobile device to an intermittently connected remote server or data center within a given time constraint, which is referred to as the *cooperative offloading* problem. Unfortunately, cooperative offloading is NP-hard. To this end, a heuristic algorithm is designed based on the proposed probabilistic framework, which provides the estimation of the probability of successful data delivery over the opportunistic path, considering both data size and contact duration. Due to the lack of global information, a distributed algorithm is further proposed. The performance of the proposed approaches is evaluated based on both synthetic networks and real traces, and simulation results show that cooperative offloading can significantly improve the data delivery probability and the performance of both heuristic algorithm and distributed algorithm outperforms other approaches.**

I. Introduction

Opportunistic mobile networks consist of mobile devices which are equipped with short-range radios (e.g. Bluetooth, WiFi). Mobile devices intermittently contact each other without the support of infrastructure when they are within range of each other. Most research work on such networks focuses on data forwarding [22][6][7] and data caching [9][26]. Other research work focuses on exploring opportunistic mobile networks for various applications. Shi *et al.* [21] proposed to enable remote computing among mobile devices so as to speedup computing and conserve energy. Liu *et al.* [16] explored the practical potential of opportunistic networks among smartphones. Han *et al.* [10] proposed to migrate the traffic from cellular networks to opportunistic communications between smartphones. Lu *et al.* [17] built a system to network smartphones opportunistically using WiFi for providing communications in disaster recovery.

In this paper, we focus on exploiting opportunistic communications for offloading data which is to be transmitted from a mobile device to an intermittently connected remote server or data center (we also call infrastructure for simplicity). Specifically, besides directly transmitting data to infrastructure, the mobile device can also transfer data to other devices and then let other devices transmit the data to infrastructure so as to improve the probability of successful data delivery. Unlike mobile cloud computing, which assumes mobile devices are always connected to cloud via cellular networks and the decision of offloading is mainly based on energy consumption or application execution time [4][3], we consider the data offloading scenario where mobile devices only have intermittent connections with infrastructure, and mobile devices collaborate to transmit data to infrastructure. There are several application scenarios. For example, in vehicular ad hoc networks, a vehicle has to seek help from other vehicles for transmitting data to roadside units when it will not meet roadside units on its current route or it cannot completely transfer the data to roadside units due to their transient contact. In mobile sensing, to upload a large amount of sensed data, it is better to fragment and distribute the data to other devices nearby, and then let them transmit the data when connected to data collectors. In disaster recovery, due to the very limited coverage and bandwidth of deployed mobile cellular towers, data can be fragmented and sent to multiple users for a better probability to reach the mobile cellular towers.

In opportunistic mobile networks, data replication [22] and data redundancy [11] are normally employed to improve the probability of successful data delivery by increasing the amount of transmitted data. However, both techniques lead to high bandwidth overhead, which can be severe for large data transfer and costly for cellular networks. Unlike existing work, we focus on offloading data segments to other mobile devices so as to improve the probability of successful data delivery. We rely on an important observation that the probability of successful data delivery over the opportunistic path drops dramatically when the size of transmitted data is large. In such scenarios, offloading data segments can yield a better delivery probability than data replication or data redundancy, in addition to not incurring additional bandwidth overhead. However, maximally improving the delivery probability by data offloading, which is referred to as the *cooperative offloading* problem, turns out to be non-trivial but NP-hard. To deal with this, we design a heuristic algorithm based on the proposed probabilistic framework which provides the estimation of the probability of data delivery over the opportunistic path, considering both data size and contact duration. To cope with the lack of global information, we further propose a distributed algorithm which employs *criterion assignment*, *real-time adjustment* and *assignment update* to solve cooperative offloading in a distributed way. Through extensive simulations on synthetic networks and real traces, we demonstrate that

This work was supported in part by Network Science CTA under grant W911NF-09-2-0053 and CERDEC via contract W911NF-09-D-0006.

cooperative offloading can greatly improve the data delivery probability and the performance of both heuristic algorithm and distributed algorithm outperforms other approaches.

The rest of this paper is organized as follows. Section II reviews related work and Section III gives the overview. The probabilistic framework and the heuristic algorithm are presented in Section IV-B, followed by the distributed algorithm in Section V. Section VI evaluates the performance of the proposed approaches and Section VII concludes the paper.

II. Related Work

Mobile opportunistic networks have been studied mainly on data forwarding, where mobile nodes carry and forward messages upon intermittent node contacts. The key problem is how to select appropriate relays such that messages can be forwarded to destinations quickly considering forwarding cost. Many forwarding schemes have been proposed from different perspectives, such as *representative* strategies including [6][23], *contact capability* based schemes including [25][7], *social concepts* based schemes including [18][19]. Moverover, forwarding is addressed as a resource allocation problem in [1] and forwarding capability of nodes under energy constraints is investigated in [2]. However, most research work of forwarding in mobile opportunistic networks is based on an assumption that nodes can completely transfer a data item during their contact. Nevertheless, this assumption is not valid in some cases, for example, when node contacts are transient (e.g., in vehicular ad hoc networks) or when data items are large. Unlike the existing work, in this paper, we release such assumption and take contact duration into consideration. The selection of forwarding path(s) for a data item is determined based on the size of the data item; i.e., we choose the path(s) with the maximum probability of successful delivery of the data item by incorporating data fragmentation (i.e., the data item can be fragmented and the segments can be forwarded over different paths).

The explosive growth of mobile devices (such as smartphones and tablets) stimulates the research on exploring mobile opportunistic networks for various applications. These applications can be generally classified into two categories: mitigating cellular traffic load and offloading computational tasks. The potential of mobile opportunistic networks in terms of reducing cellular traffic load was investigated in [16] and some solutions were proposed in [20]. Computational offloading among intermittently connected mobile devices was investigated in [13]. Different from the existing work, we investigate the application scenarios where mobile devices cooperatively transmit data to an intermittently connected server so as to improve the probability of data delivery.

III. Overview

A. Network Model

Consider a network, where mobile nodes opportunistically contact each other and each node may intermittently connect with infrastructure. Then, mobile nodes and infrastructure together form an opportunistic mobile network. To simplify the notation and the presentation of the paper, infrastructure is seen as a special node, and intermittent connections between nodes and infrastructure are seen as node contacts. In the network, for two nodes, an edge exists between them only if they opportunistically contact each other. With regard to data offloading, each node can exploit other nodes for data transmission. Specifically, besides transmitting a data item to infrastructure along one particular path, the node can also select multiple paths and each path carries only a part of the data item.

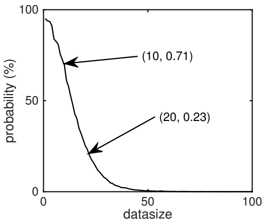
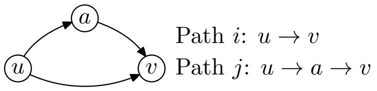


Fig. 1: Relation between data size and successful delivery probability for a given pair of path and time constraint.


a
u
v
Path i: $u \rightarrow v$
Path j: $u \rightarrow a \rightarrow v$

Successful delivery probability

data size	path	
	i	j
$S = 20$	0.23	0.23
$S/2$	0.71	0.71

Fig. 2: A simple scenario where node u and v are connected by two paths. Table shows the delivery probabilities of different data sizes over the paths, taken from Fig. 1

B. Basic Idea

In an opportunistic mobile network, given the pattern of contact frequency and contact duration between nodes, we first investigate the relation between data size and successful delivery probability for given path and time constraint. Intuitively, since the size of data that can be transmitted during a node contact is restricted by the contact duration, data with small size can be easily transmitted over the opportunistic path; data with large size may require multiple node contacts and thus not be completely transmitted. As shown in Figure 1, which is obtained for a given pair of path and time constraint (see Section VI), the probability of successful delivery exponentially decreases when the data size increases. In other words, the delivery probability can exponentially increase if less data is transmitted over the opportunistic path. Based on this important observation, it is concluded that a node may obtain a better delivery probability of a data item if it selects multiple paths to the destination and each path carries a part of the data item.

To better elaborate on this conclusion, let us consider a simple example as shown in Figure 2, where node u needs to send a data item S (we abuse the notation a little and let S also denote the size of the data item) to v. Let us use the real delivery probabilities over an opportunistic path as labeled in Figure 1, and $S = 20$. Assume paths i and j have the same delivery probability for S and $S/2$. If u transmits S to v over path i or j, the delivery probability is 0.23. If each path carries a replication of S, the probability is $\sum_{n=1}^{2} \binom{2}{n} 0.23^n 0.77^{2-n} = 0.407$, considering all failures are Bernoulli. Instead, if u sends $S/2$ along each path, the delivery probability is $0.71 \times 0.71 = 0.504$, which is more than two times of the probability when S is sent over individual path. It may be surprising that sending data segments along multiple paths can achieve a better delivery probability, even better than sending replications along multiple paths. Although increasingly sending replications over multiple paths may eventually yield better performance, this approach is restricted by available paths and incurs a cost of multiplied network resources. In this example, sending S over four paths between u and v will result in the delivery probability of $\sum_{n=1}^{4}\binom{4}{n}0.23^n 0.77^{4-n} > 0.504$, but at the cost of four times of network traffic and thus bandwidth overhead, not to mention requiring four paths between the source and destination. Moreover, applying data redundancy (e.g., erasure coding) also incurs r (replication factor) times of network resources and may or may not be beneficial [11]. Therefore, in this paper, instead of data replication or data redundancy, we focus on maximally improving the probability of data delivery by sending data segments through multiple paths (i.e., offloading data segments to other nodes), which leads to no additional overhead and better performance; i.e., we choose the path(s) that has or have the maximum probability of data delivery, considering sending data fragments via multiple paths.

C. Problem Statement

Suppose node u needs to transmit a data item S to node v within time constraint T. The *cooperative offloading* problem is defined as to maximize the probability of successful delivery of S within T by offloading data among opportunistically connected nodes, where S can be fragmented into different segments and each segment is transmitted over a different path (i.e., each path at most carries one segment). Then, cooperative offloading can be mathematically formulated as

$$\begin{aligned} \max \quad & \prod_{i=1}^{m}\prod_{j=1}^{n} P(S_i, j)^{x_{i,j}} \\ \textit{s.t.} \quad & \sum_{i=1}^{m} S_i = S, \quad m \in \{1, 2, \cdots, n\}, \\ & \sum_{j=1}^{n} x_{i,j} = 1, \quad \sum_{i=1}^{m} x_{i,j} \in \{0, 1\}, \\ & x_{i,j} \in \{0, 1\}, \end{aligned} \tag{1}$$

where n is the number of paths between u and v, m is the number of data segments, $P(S_i, j)$ is the probability that node u can transmit segment i with size S_i to node v within T along path j, $x_{i,j} = 1$ if segment i is assigned to path j, otherwise $x_{i,j} = 0$. Moreover, $\sum_{j=1}^{n} x_{i,j} = 1$ ensures that each segment is assigned once, and $\sum_{i=1}^{m} x_{i,j} \in \{0, 1\}$ ensures that each path carries at most one segment. Thus, the main problem of cooperative offloading is determining how to fragment the data item (including the number of segments and the size of each segment) and which path to transmit each segment so as to maximize the successful delivery probability of the entire data item before deadline. Thus, it is a joint optimization of the number of data segment, the size of each segment, and the selection of path for each segment. Although the formulation of the problem is straightforward, it turns out to be NP-hard, which can be proved by reduction to the minimization knapsack problem.

IV. Probabilistic Framework and Heuristic Algorithm

In opportunistic mobile networks, even the estimation of the delivery probability of a data item along a particular path is hard. Thus, in this section, we first propose the probabilistic framework to estimate the probability of data delivery over the opportunistic path. To simplify the estimation of such probability, the existing work [11][26] only considers single node contact along an opportunistic path and thus severely underestimates the data delivery probability. To cope with this, we consider multiple opportunistic node contacts, and thereby our framework can accurately estimate the probability. Then, based on the proposed probabilistic framework, we give the design of the heuristic algorithm.

A. Probabilistic Framework

In opportunistic mobile networks, contact pattern between nodes has been well analyzed. Similar to [1][8], we model the contact process between nodes as the independent Poisson process. This modeling has been experimentally validated in [8]. Moreover, unlike [15][14] that assume only a fixed amount of data can be transferred during a node contact, or [25][8] that assume an arbitrary size of data can be sent during a node contact, we do not make any assumption on that; i.e. the data amount that can be transmitted during a node contact depends on the contact duration as in [11][26]. In [11][26], the data amount that can be transmitted between nodes is estimated based on single node contact. However, given a time constraint, nodes may contact each other multiple times and thus such estimation is not technically sound. Therefore, to accurately estimate the delivery probability, we consider the data amount that is brought by the number of contacts between nodes within a time constraint. Moreover, as [24] has validated that contact duration in opportunistic mobile networks follows the Pareto distribution, we also model contact duration between nodes as the Pareto distribution.

1) Opportunistic Contact Probability: Since node contacts are independent Poisson processes, let random variable T_1 represent inter-contact duration between node u and v, which follows the Exponential distribution with rate parameter λ_1, i.e. $T_1 \sim \mathrm{Exp}(\lambda_1)$, where λ_1 is the rate parameter of the Poisson process between u and v. Similarly, we have $T_2 \sim \mathrm{Exp}(\lambda_2)$ for node v and w. Then, the available probability of the opportunistic path from u to w via v (i.e., u contacts v first and then v contacts w) within T can be represented as $P(\mathsf{T} \leq T)$, where $\mathsf{T} = T_1 + T_2$. Note that the sequence of node contact is already captured by $P(\mathsf{T} \leq T)$. Assuming the probability density functions (PDF) of T_1 and T_2 are $f_1(t)$ and $f_2(t)$, respectively, $P(\mathsf{T} \leq T)$ can be calculated through the convolution $f_1(t) \otimes f_2(t)$. However, a data item may not be completely transmitted by one contact between neighboring nodes but require multiple contacts. Thus, multiple contacts at each hop should be considered.

Assume a data item can be transferred from u to w by the number of contacts a between u and w, and from w to v by the number of contacts b between w and v. For a collection of independent and identically distributed (*i.i.d.*) random variables $\{T_i,\, i = 1, \dots, a\}$ and $T_i \sim \mathrm{Exp}(\lambda_1)$, we have $\mathcal{T}_1 \sim \mathrm{Gamma}(a, \lambda_1)$, where $\mathcal{T}_1 = \sum_{i=1}^{a} T_i$. Similarly, for $\{T_j,\, j = 1, \dots, b\}$ and $T_j \sim \mathrm{Exp}(\lambda_2)$, we have $\mathcal{T}_2 \sim \mathrm{Gamma}(b, \lambda_2)$, where $\mathcal{T}_2 = \sum_{j=1}^{b} T_j$. Then the PDF of $\mathbb{T} = \mathcal{T}_1 + \mathcal{T}_2$ can be represented as

$$f(t) = f(t; a, \lambda_1) \otimes f(t; b, \lambda_2) = \frac{t^{a-1} e^{-t\lambda_1}}{\lambda_1^{-a}\Gamma(a)} \otimes \frac{t^{b-1} e^{-t\lambda_2}}{\lambda_2^{-b}\Gamma(b)}.$$

For a k-hop opportunistic path with the rate parameter λ_i and the contact number n_i at each hop i (we also call it k-hop opportunistic contact path), the PDF of $\mathbb{T} = \mathcal{T}_1 + \cdots + \mathcal{T}_k$ can be easily generalized as

$$\begin{aligned} f(t) &= f(t; n_1, \lambda_1) \otimes f(t; n_2, \lambda_2) \otimes \cdots \otimes f(t; n_k, \lambda_k) \\ &= \frac{t^{n_1-1} e^{-t\lambda_1}}{\lambda_1^{-n_1}\Gamma(n_1)} \otimes \frac{t^{n_2-1} e^{-t\lambda_2}}{\lambda_2^{-n_2}\Gamma(n_2)} \otimes \cdots \otimes \frac{t^{n_k-1} e^{-t\lambda_k}}{\lambda_k^{-n_k}\Gamma(n_k)}. \end{aligned} \tag{2}$$

However, it is hard to calculate (2) due to the higher order convolution. To simplify the calculation, we have the following by extending Welch-Satterthwaite approximation to the sum of gamma random variables

$$f(t) \approx \frac{t^{\gamma-1} e^{-t\delta}}{\delta^{-\gamma}\Gamma(\gamma)}, \tag{3}$$

where

$$\gamma = \frac{\left(\sum_{i=1}^{k} n_i \lambda_i\right)^2}{\sum_{i=1}^{k} n_i \lambda_i^2} \quad \text{and} \quad \delta = \frac{\sum_{i=1}^{k} n_i \lambda_i^2}{\sum_{i=1}^{k} n_i \lambda_i}.$$

Thus, $\mathbb{T}$ can be approximated by a single gamma random variable $\widetilde{\mathbb{T}} \sim \mathrm{Gamma}(\gamma, \delta)$. Then, the available probability of the opportunistic contact path with time constraint T, denoted as $P(\mathbb{T} \le T)$, can be easily calculated.

2) Data Transfer Probability: As contact duration is modeled as the Pareto distribution, it can be easily concluded that the amount of data that can be transferred during a contact between a pair of nodes also follows a Pareto distribution since the data rate between a pair of nodes is relatively stable as in [26].

Let random variable D_1 represent the amount of data transmitted during a contact between node u and v, which follows the Pareto distribution with shape parameter α and scale parameter β (β is the minimum possible value of D_1), i.e. $D_1 \sim \mathrm{Pareto}(\alpha, \beta)$. For a collection of *i.i.d.* random variables $\{D_i,\, i = 1, \dots, c\}$ and $D_i \sim \mathrm{Pareto}(\alpha, \beta)$, let $\mathcal{D} = \sum_{i=1}^{c} D_i$. Then, the probability that a data item with size D can be transferred with the number contacts c between node u and v is represented as $P(\mathcal{D} \ge D)$ and can be derived from the PDF of $\mathcal{D}$. However, since $\mathcal{D}$ is the sum of an arbitrary number of random variables, its PDF cannot be easily approximated by stable distributions. Therefore, we choose to approximate $\mathcal{D}$ by the maximum value of $\{D_i,\, i = 1, \dots, c\}$ denoted as $\mathcal{M}$, which relies on the observation that $\mathcal{M}$ has the same order of magnitude as the sum $\mathcal{D}$ since the Pareto distribution is a heavy-tailed distribution.

To capture such intuition, let us define $R = \mathcal{D}/\mathcal{M}$. Then, $\mathcal{D}$ can be approximated by $\mathcal{M}$, considering the ratio R. Then, $P(\mathcal{D} \ge D)$ can be approximated as

$$P(\mathcal{D} \ge D) \approx 1 - \left(1 - \left(\frac{\beta \bar{R}}{D}\right)^{\alpha}\right)^{c}, \tag{4}$$

where $\bar{R}$ is the expectation of R. The derivation is omitted due to the page limitation.

3) Data Delivery Probability: Given an opportunistic path, a data item D and a time constraint T, we will show how to calculate the probability of successful delivery of D over the opportunistic path within T, denoted as $P(T, D)$.

First, we consider the case of one hop path. The data item needs at most $l = \lceil \frac{D}{\beta} \rceil$ node contacts to be completely transferred. Then, $P(T, D)$ can be calculated as the sum of the probabilities that the data item is completely transmitted at each particular contact as following

$$P(T, D) = \sum_{i=1}^{l} \widetilde{P}_{i-1} \cdot P(\mathbb{T}_i \le T - T^{'}) \cdot P(\mathcal{D}_i \ge D), \tag{5}$$

where

$$\widetilde{P}_i = \begin{cases} \prod_{j=1}^{i} P(\mathbb{T}_j \le T - T^{'}) \cdot P(\mathcal{D}_j < D) & i > 0 \\ 1 & i = 0, \end{cases}$$

$T^{'} = \frac{D}{r}$ denotes the transmission time of the data item and r is the data rate between these two nodes. As in (5), the probability that ith contact completes the data transfer can be interpreted as the probability that the data can be transferred within i node contacts when $i-1$ contact fails (i.e. the data transferred by former $i-1$ contacts is less than D and the sum of inter-contact duration of i contacts is less or equal to $T - T^{'}$).

Then, we can extend the calculation of $P(T, D)$ to a k-hop path. Given a k-tuple

$$\langle n_1, \dots, n_i, \dots, n_k \rangle,\ 1 \le n_i \le \lceil \frac{D}{\beta_i} \rceil, \tag{6}$$

which falls into one of the $\prod_{i=1}^{k} \lceil \frac{D}{\beta_i} \rceil$ possible combinations, we need to calculate the probability that the data can be transferred through the path by the specified number of contacts at each hop as n_i in the k-tuple. Similar to the calculation for one hop path, we need to consider the failure probability of tuples which have one less contact number at one hop. For example, for 2-tuple $\langle n_1, n_2 \rangle$, we need consider the failure probability of both $\langle n_1 - 1, n_2 \rangle$ and $\langle n_1, n_2 - 1 \rangle$. Finally, $P(T, D)$ on the k-hop path can be calculated as the sum of the probabilities of all the combinations.

In summary, the probabilistic framework explores contact pattern between nodes to provide the estimation of data delivery probability over the opportunistic path. To the best of our knowledge, this is the first work that provides such estimation without any restrictions.

B. Heuristic Algorithm

According to (1), intuitively, a good solution of cooperative offloading should have a small number of data segments while keeping a high delivery probability for each segment. The basic idea of the heuristic algorithm is to first allocate some paths from source to destination (infrastructure) and assign certain amount of data at each path, and then reallocate the assigned data among these paths to improve the delivery probability of the data item.

To choose particular paths for data offloading, we need to measure the capability of paths for data transmission. Since all paths are opportunistic, which are characterized by contact frequency and contact duration at each hop, we can explore these two properties to measure the capability of each path. However, without the prior knowledge of how much data will be assigned at each path, it is difficult to quantify such capability of paths. Therefore, we employ the available probability of the opportunistic path within the time constraint, denoted as Q, as the metric to characterize each path, since the establishment of the opportunistic path within the time constraint is the prerequisite of data transmission between the pair of nodes. Moreover, the available probability of direct path (one hop path between source and destination), denoted as $Q^{'}$, is used as the criterion to choose these paths. After allocating the paths, we need to determine how much data should be assigned to each path initially. When nodes contact each other, β is the size of data that can be guaranteed to be transferred (according to Pareto distribution). Thus, for each path, we allocate the amount of data that can be guaranteed to be transferred along the path with one node contact at each hop (i.e. $\min\{\beta_i, i = 1, \cdots, k\}$). We call this the *path capacity*, denoted as C^i_{uv} for path i between node u and v.

Considering an example where node u needs to transmit a data item S to node v within T, the heuristic algorithm works as follows. First, we adopt Dijkstra's algorithm with the metric of Q to find the path from node u to v that has the minimum $1/Q$. Note that in Dijkstra's algorithm Q is calculated for the path between node u and unvisited node according to (3). If the found path i has $Q_i \geq Q^{'}$ (if node u does not have direct path to v, $Q^{'} = 0$), then we assign the data amount of C^i_{uv} to path i, i.e., $S_i = C^i_{uv}$ if $C^i_{uv} < S - \sum_{i \in \mathcal{P}_A} C^i_{uv}$, otherwise $S_i = S - \sum_{i \in \mathcal{P}_A} C^i_{uv}$, and add i into the set of allocated paths $\mathcal{P}_A$. After allocating path i, the edges along the path will be removed from the network and not considered in subsequent searches. The searching process is iteratively executed until it meets one of the following stop conditions: (1) $Q_i < Q^{'}$; (2) node v cannot be reached by Dijkstra's algorithm from u; (3) $\sum_{i \in \mathcal{P}_A} S_i = S$. If the searching process ends up with $|\mathcal{P}_A| = 0$, there is no need to offload and it is better to send the data item directly from u to v. If $\sum_{i \in \mathcal{P}_A} S_i < S$, the remaining data needs to be assigned to $\mathcal{P}_A$. If $\sum_{i \in \mathcal{P}_A} S_i = S$, data reallocation among $\mathcal{P}_A$ is needed to improve the delivery probability of S.

To assign the remaining data of $S - \sum_{i \in \mathcal{P}_A} C^i_{uv}$ to $\mathcal{P}_A$, first we rank $\mathcal{P}_A$ according to the delivery probability of the assigned data at each path, i.e. $P^i_{uv}(T, S_i)$, $i \in \mathcal{P}_A$. Then, we assign more data to the path with the highest delivery probability among $\mathcal{P}_A$ until the probability is lower than the second highest one. The process is repeated until there is no remaining data. When assigning more data to the selected allocated path, we need to determine the amount of data to be assigned each time. As the path capacity of a k-hop path is $\min\{\beta_1, \beta_2, \ldots, \beta_k\}$, each time we increase the amount of the assigned data to next higher value among $\{\beta_1, \beta_2, \ldots, \beta_k\}$. If the assigned data is more or equal to $\max\{\beta_1, \beta_2, \ldots, \beta_k\}$, each time the assigned data is increased by the path capacity.

After all the remaining data is assigned, we further refine the delivery probability of S by reallocating the data assigned at the path, say path i, which has the lowest delivery probability among $\mathcal{P}_A$, to other paths in $\mathcal{P}_A$ using the same approach described above. If the reallocation of S_i improves the delivery probability of S, path i is excluded from $\mathcal{P}_A$ and the process is repeated, otherwise the reallocation process stops. The heuristic algorithm ends up with the allocated paths and the data assignment at each path.

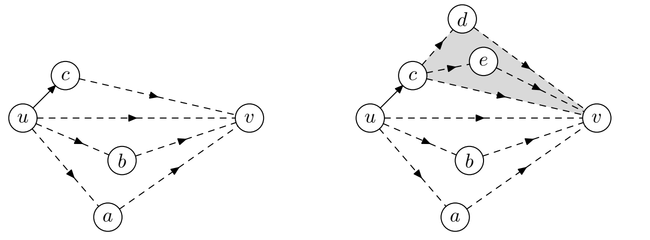
V. Distributed Algorithm

Global information is required for the heuristic algorithm presented above. However, such information might not be available or cost too much in some scenarios. Furthermore, the pattern of contact frequency and contact duration between nodes may vary over time, which will impact the performance of the heuristic algorithm. Thus, in this section we propose a distributed algorithm to address these problems.

In opportunistic mobile networks, it is hard to maintain multi-hop information locally, since nodes only intermittently contact each other and thus the information cannot be promptly updated. Therefore, in the design of the distributed algorithm, it is only required that each node maintains 2-hop information of contact frequency and contact duration between nodes; i.e., each node maintain contact frequency and contact duration with its neighbors, and when two nodes encounter, they will exchange such information and then they will have 2-hop information. The design efforts of the distributed algorithm is to determine whether and how much data should be transmitted when the node carrying the data encounters other nodes based on these local information.

With the 2-hop information maintained locally, the source node can construct paths to the destination (infrastructure). For example, as shown in Figure 3a, there are four paths that can be constructed from node u to node v including one 1-hop path and three 2-hop paths, where node a, b and c are the potential nodes to cooperate for data transmission. Moreover, when the source node encounters a neighbor, it can also have the local information maintained at this neighbor. For example, as shown in Figure 3b, when node u encounters node c, node u can have the information of the 2-hop paths between c and v, and then node u can construct more paths to node v.

The distributed algorithm exploits the locally maintained information for data offloading, which includes three phases: *criterion assignment*, *real-time adjustment* and *assignment update*. The first phase leverages the local information to select the paths for offloading and determines the data assignment for each path; the second phase adjusts the data assignment based on the local information of the encountered node to achieve a better delivery probability; the third phase updates the data assignment at sender and receiver based on the



(a) paths between node u and v based on the information at node u

(b) paths between node u and v based on the information at node u and c

Fig. 3: Paths constructed based on local information

actually transmitted data amount between them. Considering Figure 3 as an example, where node u has data S to transmit to v. First, node u determines the criterion assignment based on the paths in Figure 3a. Then, when node u encounters c, node u needs to adjust the size of data to be transmitted to node c, which can maximally improve the delivery probability of S determined by the criterion assignment, based on the additional paths in Figure 3b. After the data transmission between u and v, both need to update their data assignment.

A. Criterion Assignment

Although the data to be transmitted to a neighbor will be adjusted when the source node meets the neighbor, the source node should provide a reasonable criterion for later adjustment. To obtain a good delivery probability, the criterion assignment is determined as follows. Let $\mathcal{P}_{uv}$ denote the set of paths from node u to v, which only includes the paths that are no more than two hops from node u to v, and let $\mathcal{C}_{uv}$ denote the capacity of $\mathcal{P}_{uv}$, where $\mathcal{C}_{uv} = \sum_{i \in \mathcal{P}_{uv}} C^i_{uv}$. If $\mathcal{C}_{uv} < S$, the data assignment at path i in $\mathcal{P}_{uv}$ is $S_i = S \times \frac{C^i_{uv}}{\mathcal{C}_{uv}}$. If $\mathcal{C}_{uv} \geq S$, we rank $\mathcal{P}_{uv}$ according to the available probability of each path, and then the first k paths of $\mathcal{P}_{uv}$, where $\sum_{i=1}^{k} C^i_{uv} \geq S$, are selected and the assigned data for each path is equal to the path capacity except the last one, the assignment of which is $S - \sum_{i=1}^{k-1} C^i_{uv}$. Let $\mathcal{A}_u$ denote the data assignment at node u, $\mathcal{A}_u = \{S_i, i \in \mathcal{P}_{uv}\}$.

B. Real-time Adjustment

With the criterion assignment, we have the data assignment for each path from node u to v. The data assigned at the path that goes through node c is the amount of data to be transmitted to c when node u and c encounter each other. However, node c may be more likely to deliver a certain amount of data to node v than node u. If so, this amount of data should be transmitted to node v from c instead of u so as to improve the delivery probability. Thus, node u needs to investigate this capability of node c based on c's local information as the shadow part in Figure 3b when they are in contact, and then adjust the amount of data to be transmitted to c accordingly.

The real-time adjustment works as follows. When nodes u and c are in contact, first node u ranks $\mathcal{P}_{uv}$ according to their delivery probability of the assigned data, which is $P^i_{uv}(T_r, S_i)$, $i \in \mathcal{P}_{uv}$ and T_r is the remaining time before deadline, and selects path j from $\mathcal{P}_{uv}$, where $P^j_{uv}(T_r, S_j)$ is minimum (note that $P^i_{uv}(T_r, S_i) = 1$ if $S_i = 0$). After that, the path among $\mathcal{P}_{cv}$, to which the reallocation of S_j can maximally improve the delivery probability of S, will be located, say path k. Then, S_j should be transferred to node v from c instead of u, and thus S_j needs to be transmitted from u to c first. So the data amount to be transmitted from u to c will be increased by S_j and S_j will be assigned at path k. The process is repeated. If none in $\mathcal{P}_{cv}$ with the reallocation of S_j can increase the delivery probability of S, the process stops.

The following details how to compare the improvement of the delivery probability of S when S_j is reassigned to different paths, say path d and e in $\mathcal{P}_{cv}$. Let S_d and S_e denote the amount of data already assigned at path d and path e, respectively, which can be the assignment of data currently carried by node c or zero. To compare their improvement, we only need to calculate the delivery probabilities of $S_j + S_d + S_e$ when S_j is reallocated to path d and path e, respectively, since the assignment of the rest of S is not impacted. For path d and e, we need to calculate $P^d_{cv}(T_r, S_d + S_j) \cdot P^e_{cv}(T_r, S_e)$ and $P^d_{cv}(T_r, S_d) \cdot P^e_{cv}(T_r, S_e + S_j)$ respectively. If $P^d_{cv}(T_r, S_d + S_j) \cdot P^e_{cd}(T_r, S_e) > P^d_{cv}(T_r, S_d) \cdot P^e_{cv}(T_r, S_e + S_j)$, it is better to assign S_j to path d, otherwise path e. Similarly, to determine whether the reassignment of S_j, e.g., from path j to d, can improve the delivery probability, we need to compare $P^j_{uv}(T_r, S_j) \cdot P^d_{cv}(T_r, S_d)$ and $P^d_{cv}(T_r, S_d + S_j)$.

C. Assignment Update

The real-time adjustment ends with the total amount of data to be transmitted from node u to node c, denoted as S_c, and the assignment of S_c at $\mathcal{P}_{cv}$. Together with the assignment of currently carried data at node c, let $\mathcal{A}_c$ denote the total data assignment at node c, $\mathcal{A}_c = \{S_i, i \in \mathcal{P}_{cv}\}$. As the actually transmitted data from u to c, denoted by $S^{'}_c$, may be less than S_c due to the uncertainty of their contact duration, both node u and node c need to update the data assignment after the data transmission. For node u, the amount of $S^{'}_c$ should be excluded from the assignment $\mathcal{A}_u$ by sequentially removing the data assigned at the path with the lowest delivery probability. Similarly, for node c, the amount of $S_c - S^{'}_c$ needs to be removed from $\mathcal{A}_c$.

In summary, based on these three phases, the distributed algorithm works as follows. First, the source node determines a criterion assignment for the data item. Then, when the node that carries the data encounters a neighbor, it first makes sure whether the neighbor is the source node or if it received the data from this neighbor before. If so, no data will be transmitted. If the encountered node does not have the data, the node carrying the data needs to determine how much data is to be transmitted to the encountered node (if both of nodes have the data, they also need to determine who should transmit data to another) by comparing the improvement of the delivery probability by data reallocation. After that, the determined reallocated data is transmitted and then the data assignment at sender and receiver is updated based on the amount of data actually transferred during their contact.

VI. Performance Evaluation

In this section, we evaluate the performance of the heuristic algorithm and the distributed algorithm based on synthetic networks and real traces.

A. Evaluation on Synthetic Networks

First we investigate the performance of the heuristic algorithm on synthetic networks. The synthetic networks are generated by the well-known benchmark [12]. It provides power-law distribution of node degree and edge weight, and various topology control. There are several parameters to control the generated network: the number of nodes n; the mixing parameter for the weights μ_w; the mixing parameter for the topology μ_t; the exponent for the weight distribution ξ; the average node degree d; the maximum node degree d_m. The settings of these parameters are shown in Table I.

TABLE I: Parameter settings for benchmark

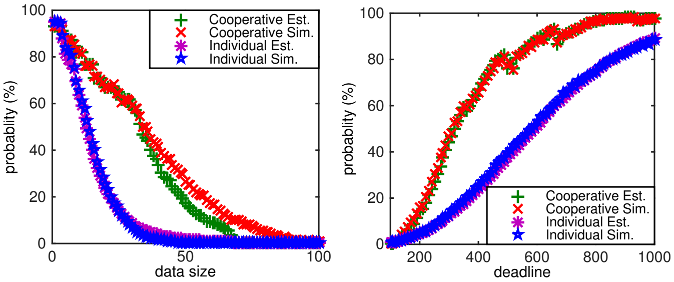
Parameter	Value	Parameter	Value
n	100	ξ	2
μ_w	0.3	d	5, 10
μ_t	0.3	d_m	8, 15

With the synthetic networks, we also need to generate the contact pattern between nodes and between node and infrastructure. The settings of these distribution parameters (i.e., α, β and λ) are shown in Table II, where $\alpha = [3, 4]$, for example, means that α between pair of nodes is set to a random number between 3 and 4, which follows the uniform distribution.

TABLE II: Parameter settings for node contacts

Type	Parameter	Value
Between nodes	α	$[3, 4]$, $[6, 10]$
	β	$[2, 3]$
	λ	$1/$edge weight
Between node and infrastructure	α'	$[3, 4]$
	β'	$[2, 3]$
	λ'	$[0.001, 0.1]$, $[0.01, 0.1]$

When a node transmits data to infrastructure, it needs to decide whether to offload the data to other nodes so as to improve the delivery probability. So, the node goes through the following procedure. First, it calculates the probability that the node directly transmits the data to infrastructure (no offloading, the data is transmitted only when it connects with infrastructure), denoted as *Individual Est.*. Then it employs the heuristic algorithm to calculate the probability if it offloads the data to other nodes, denoted as *Cooperative Est.*. If *Cooperative Est.* is greater than *Individual Est.*, the node offloads the data to the selected paths with the corresponding data assignment determined by the heuristic algorithm. *Individual Est.* and *Cooperative Est.* are compared for data transmissions with different data sizes and deadlines at each node in the networks. We also compare the delivery probabilities based on simulations (denoted as *Individual Sim.* and *Cooperative Sim.*). Specifically, we generate the random numbers for inter-contact duration and contact duration between neighboring nodes according to their distributions, then data is transmitted between nodes based on these generated contact information.



(a) probability vs data size, $T = 400$ (b) probability vs deadline, $S = 20$

Fig. 4: Data Delivery probability based on estimation and simulation for *Individual* and *Cooperative* in synthetic networks, where $d = 10$, $d_m = 15$, $\alpha = [6, 10]$, $\lambda' = [0.001, 0.01]$.

Figure 4 shows the comparison of the delivery probabilities for *Individual* and *Cooperative* in synthetic networks. Figure 4a shows the successful probability of transmissions with varying data sizes, meanwhile Figure 4b shows the successful probability of transmissions with varying deadlines, where the estimated probability is averaged for all the nodes for each transmission, and the simulated probability is averaged for all the nodes with 500 simulation runs (i.e., the simulated probability is the number of successful transmissions divided by the total number of transmissions). As shown in Figure 4a, for both *Cooperative* and *Individual*, the probability decreases with the increase of data size as expected. When the size of data is small, the data can be easily transmitted directly to infrastructure, and thus their probabilities are similar. However, when the size of data increases, the probability of *Individual* drops dramatically from $S = 10$ to $S = 40$ while cooperative offloading significantly improves the probability (e.g., it can be increased from 20% to 70% when $S = 20$). When the data size increases further, the probability of *Cooperative* also decreases, i.e. cooperative offloading cannot improve the delivery probability as much as before. For *Individual*, the estimated probability and the simulated probability are almost the same for different data sizes. Meanwhile, for *Cooperative* there is a little difference between the estimated probability and the simulated probability when the data size is large. That might be incurred by the deviation of the distribution approximations in Section IV-A. Figure 4b shows the probability of data transmissions with varying deadlines. Similarly, *Cooperative* and *Individual* start at the same probability, after that the difference between expands and then narrows down when the deadline further looses. For both *Individual* and *Cooperative*, the estimated probability and the simulated probability are almost identical.

Figure 5 gives the contour of the percentage of facilitated nodes for data transmissions with different data sizes and deadlines in various network settings, where the facilitated node means that data transmission with particular data size and deadline is offloaded at the node. For example, the area enclosed by the green line (80%) in Figure 5a indicates there are more than or equal to 80% of nodes that offload the transmissions with corresponding data size and deadline. As shown in Figure 5a, for the network with $\alpha = [6, 10]$ and $\lambda' = [0.001, 0.01]$, most of the transmissions are offloaded (i.e. most of area are enclosed by green line (80%)) except the transmissions with large data sizes and short deadlines shown as the upper left corner. As shown in Figure 5b, when nodes more frequently contact infrastructure (i.e. $\lambda' = [0.01, 0.1]$), small data can be easily transmitted to infrastructure directly, and thus the transmissions with small data size are not frequently offloaded (less than 40%) as in Figure 5b. On the contrary, the transmissions with large data sizes and short deadlines are mostly offloaded. When contact duration between nodes becomes short (i.e. $\alpha = [3, 4]$ as in Figure 5c), compared to Figure 5a, the delivery probability of the transmissions with large data sizes cannot be greatly improved and thus they are less offloaded at nodes. However, the transmissions with small data sizes are mostly offloaded.

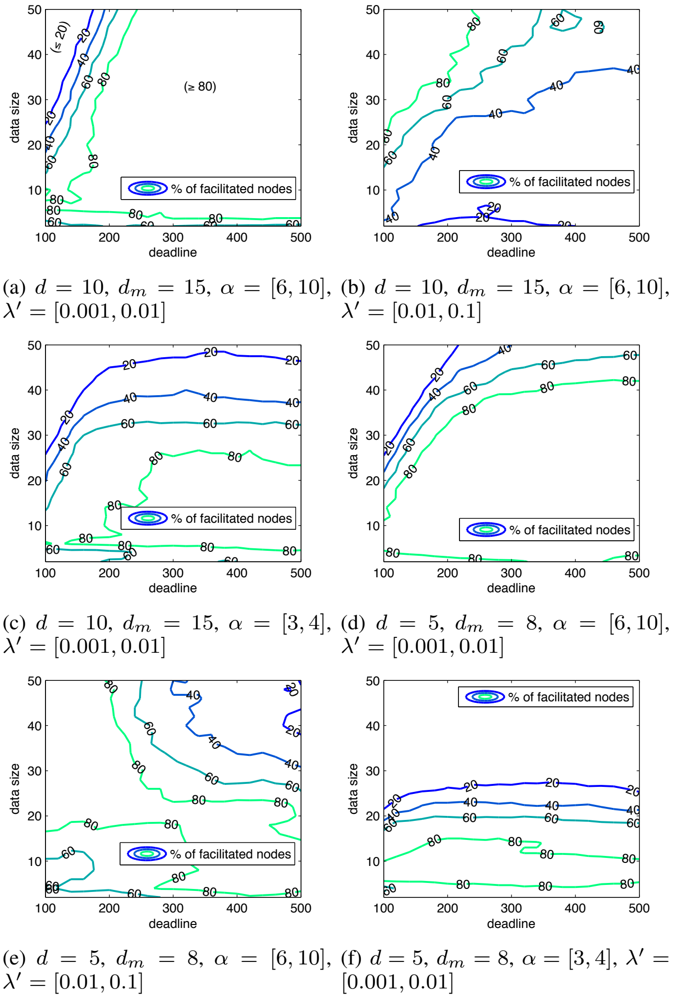
When nodes have a small number of neighbors in network, e.g. $d = 5$ and $d_m = 8$ as in Figure 5d, compared to Figure 5a, the reduced node degree undermines the offloading capability of the network; i.e., the offloaded transmissions are generally less than that of Figure 5a. Compared with Figure 5d, inter-contact duration between node and infrastructure is decreased ($\lambda' = [0.01, 0.1]$) in Figure 5e. This dramatically changes the pattern of the offloaded transmissions. As shown in Figure 5e, offloading can improve the probability of most transmissions but not for the transmissions with large data sizes and deadlines. That is because when nodes frequently contact infrastructure, it is easy for nodes to transmit the data to infrastructure by multiple contacts and offloading the data to other nodes may not yield a high delivery probability. Last, let us compare Figure 5f with Figure 5c. It is shown again that node degree affects the offloading capability of the network, since node degree determines how many nodes nearby that can be explored for data offloading.

In summary, it can be concluded that cooperative offloading can significantly improve the delivery probability in the networks with various topology and contact pattern.

B. Evaluations on Real Traces

Next, we evaluate the performance of the heuristic algorithm and the distributed algorithm based on real traces. We also compare them with other two solutions: *Spread*, where nodes offload the carried data to any encountered node, and *MaxRate*, where nodes only transmit the carried data to infrastructure or the node in its neighbor set that has the maximum contact rate with infrastructure.

The two opportunistic mobile network traces used are MIT Reality [5] and DieselNet [1]. They record contacts among mobile devices equipped with Bluetooth or WiFi moving on university campus (MIT Reality) and in suburban area (DieselNet). The details of these two traces are summarized in Table III.



(a) $d = 10$, $d_m = 15$, $\alpha = [6, 10]$, $\lambda' = [0.001, 0.01]$

(b) $d = 10$, $d_m = 15$, $\alpha = [6, 10]$, $\lambda' = [0.01, 0.1]$

(c) $d = 10$, $d_m = 15$, $\alpha = [3, 4]$, $\lambda' = [0.001, 0.01]$

(d) $d = 5$, $d_m = 8$, $\alpha = [6, 10]$, $\lambda' = [0.001, 0.01]$

(e) $d = 5$, $d_m = 8$, $\alpha = [6, 10]$, $\lambda' = [0.01, 0.1]$

(f) $d = 5$, $d_m = 8$, $\alpha = [3, 4]$, $\lambda' = [0.001, 0.01]$

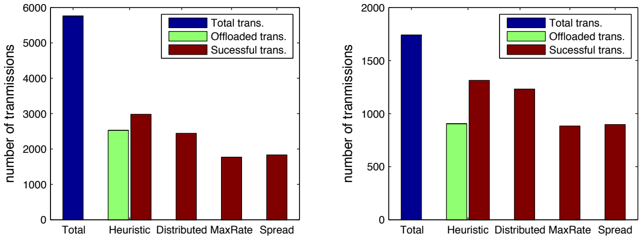
Fig. 5: Percentage of facilitated nodes for data transmissions with different data size and deadline in various network settings.

TABLE III: Trace summary

Trace	MIT Reality	DieselNet
Network type	Bluetooth	WiFi
Contact type	Direct	Direct
No. of devices	97	40
Duration(days)	246	20
No. of contacts	3268	114,046

In the experiment, half of the trace is used as warmup to obtain the distribution information of contact frequency and contact duration between nodes, and other half is used to run data transmission. Since there is no infrastructure in either trace, we choose the node with the maximum degree to act as infrastructure. Nodes that cannot construct 2-hop path to infrastructure are excluded from the traces (note that only very few nodes are eliminated). In the simulation, a node sends data with different sizes and deadlines to infrastructure at a randomly selected timestamp in each simulation run and the results are averaged for 50 runs.

For MIT Reality trace, due to the Bluetooth scan interval,



(a) MIT Reality

(b) DieselNet

Fig. 6: Successful transmissions of *Heuristic*, *Distributed*, *MaxRate* and *Spread* in MIT Reality and DieselNet.

only the contacts with duration of five minutes or more are recorded, where contact duration is at minute level and inter-contact duration is at day level. Thus, for data transmissions, the data size varies from 10MB to 60MB (the data rate is set to 240Kbps) and the time constraint varies from 10 to 100 hours. For DieselNet, contact duration is at second level and inter-contact duration is at hour level. Thus, for data transmissions in DieselNet, the data size varies from 2MB to 10MB (the data rate is set to 3.2Mbps) and the deadline varies from 1 to 10 hours.

Figure 6 gives the total transmissions, the offloaded transmission of *Heuristic*, the successful transmissions of *Heuristic*, *Distributed*, *MaxRate* and *Spread* in MIT Reality and DieselNet, respectively. In MIT Reality, as shown in Figure 6a, there are totally near 6000 data transmissions, and about 40% are offloaded by *Heuristic*. Moreover, *Heuristic* has more successful transmissions than all other three algorithms, and it is about 20% higher than that of *Distributed* and 70% higher than that of *MaxRate* and *Spread*. The number of successful transmissions are higher than the offloaded transmissions because the transmissions that are not offloaded might be completed via direct contact between source node and infrastructure. *Distributed* is better than *MaxRate* and *Spread*, and *Spread* and *MaxRate* are equivalent. In DieselNet, as shown in Figure 6b there are about 1800 total transmissions and more than 50% are offloaded by *Heuristic*. The successful transmissions of *Heuristic* and *Distributed* are comparable, which are 40% more than that of *MaxRate* and *Spread*.

In summary, it can be concluded that heuristic algorithm is better than the distributed algorithm and both are much better than *MaxRate* and *Spread*. In spite of the lack of global information, the performance of the distributed algorithm is comparable to that of the heuristic algorithm.

VII. Conclusion

In this paper, we addressed the problem of cooperatively offloading data among opportunistically connected mobile devices so as to improve the probability of data delivery to infrastructure. We first provided the probabilistic framework to estimate the probability of data delivery over the opportunistic path considering both data size and contact duration and then, based on that, we proposed a heuristic algorithm to solve cooperative offloading. To cope with the lack of global information, we further proposed a distributed algorithm. The evaluation results show that cooperative offloading can greatly improve the data delivery probability and the performance of both heuristic algorithm and distributed algorithm outperforms other approaches.

References

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *ACM SIGCOMM*, 2007.

[2] N. Banerjee, M. D. Corner, and B. N. Levine. An energy-efficient architecture for dtn throwboxes. In *IEEE NFOCOM*, 2007.

[3] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: elastic execution between mobile device and cloud. In *ACM EuroSys*, 2011.

[4] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In *ACM MobiSys*, 2010.

[5] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[6] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *ACM MobiHoc*, 2008.

[7] W. Gao and G. Cao. User-centric data dissemination in disruption tolerant networks. In *IEEE INFOCOM*, 2011.

[8] W. Gao, Q. Li, B. Zhao, and G. Cao. Social-aware multicast in disruption-tolerant networks. *Networking, IEEE/ACM Transactions on*, 20(5):1553–1566, 2012.

[9] P. Gilbert, V. Ramasubramanian, P. Stuedi, and D. Terry. Peer-to-peer data replication meets delay tolerant networking. In *IEEE ICDCS*, 2011.

[10] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. *Mobile Computing, IEEE Transactions on*, 11(5):821–834, 2012.

[11] S. Jain, M. Demmer, R. Patra, and K. Fall. Using redundancy to cope with failures in a delay tolerant network. In *ACM SIGCOMM*, 2005.

[12] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

[13] Y. Li and W. Wang. Can mobile cloudlets support mobile applications? In *IEEE INFOCOM*, 2014.

[14] Y. Lin, B. Li, and B. Liang. Efficient network coded data transmissions in disruption tolerant networks. In *IEEE INFOCOM*, 2008.

[15] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. *Selected Areas in Communications, IEEE Journal on*, 26(5):794–808, 2008.

[16] S. Liu and A. D. Striegel. Exploring the potential in practice for opportunistic networks amongst smart mobile devices. In *ACM MobiCom*, 2013.

[17] Z. Lu, G. Cao, and T. La Porta. Networking smartphones for disaster recovery. In *IEEE PerCom*, 2016.

[18] Z. Lu, X. Sun, Y. Wen, and G. Cao. Skeleton construction in mobile social networks: Algorithms and applications. In *IEEE SECON*, 2014.

[19] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. La Porta. Algorithms and applications for community detection in weighted networks. *Parallel and Distributed Systems, IEEE Transactions on*, 26(11):2916–2926, 2015.

[20] P. Sermpezis and T. Spyropoulos. Not all content is created equal: effect of popularity and availability for content-centric opportunistic networking. In *ACM MobiHoc*, 2014.

[21] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: enabling remote computing among intermittently connected mobile devices. In *ACM MobiHoc*, 2012.

[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM WDTN*, 2005.

[23] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *Networking, IEEE/ACM Transactions on*, 16(1):77–90, 2008.

[24] W. Wang, V. Srinivasan, and M. Motani. Adaptive contact probing mechanisms for delay tolerant applications. In *ACM MobiCom*, 2007.

[25] Q. Yuan, I. Cardei, and J. Wu. Predict and relay: an efficient routing in disruption-tolerant networks. In *ACM MobiHoc*, 2009.

[26] X. Zhuo, Q. Li, W. Gao, G. Cao, and Y. Dai. Contact duration aware data replication in delay tolerant networks. In *IEEE ICNP*, 2011.