Cooperative Data Offloading in Opportunistic Mobile Networks

Zongqing Lu, Xiao Sun, and Thomas La Porta

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. In contrast to 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 intermittently connected infrastructure within a given time constraint, which is referred to as the cooperative offloading problem. Unfortunately, the estimation of data delivery probability over an opportunistic path is difficult and cooperative offloading is NP-hard. To this end, we first propose a probabilistic framework that provides the estimation of such probability. Based on the proposed probabilistic framework, we design a heuristic algorithm to solve cooperative offloading at a low computation cost. 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. Experimental results show that the probabilistic framework can accurately estimate the data delivery probability, cooperative offloading greatly improves the delivery probability, the heuristic algorithm approximates the optimum, and the performance of both the 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 recent work on such networks focuses on data forwarding [30][8][9] and data caching [12][40][10]. Other research work focuses on exploring opportunistic mobile networks for various applications. Shi et al. [29] proposed to enable remote computing among mobile devices so as to speedup computing and conserve energy. Liu et al. [21] explored the practical potential of opportunistic networks among smartphones. Lu et al. [22] built a system to network smartphones opportunistically using WiFi for providing communications in disaster recovery. Han et al. [13] proposed to migrate the traffic from cellular networks to opportunistic communications between smartphones, coping with the explosive traffic demands and limited capacity provided by cellular networks.

In this paper, we focus on exploiting opportunistic communications for offloading data which is to be transmitted from a mobile device to intermittently connected infrastructure (e.g., a remote server or data center). Specifically, besides directly transmitting data to infrastructure, the mobile device can also transfer data to other devices and then let these 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 the cloud via cellular networks and the decision of offloading is mainly based on energy consumption or application execution time [4][3][35], 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 [30] and data redundancy [16] 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 and NP-hard. Although cooperative offloading admits PTAS (Polynomial Time Approximation Scheme), PTAS is still not affordable for mobile devices due to its incurred computation overhead of data delivery probability.

To deal with this, we design a centralized heuristic algorithm based on the proposed probabilistic framework to solve coop-
erative offloading. To cope with the lack of global information, we further propose a distributed algorithm. Through extensive simulations on synthetic networks and experiments on real traces, we demonstrate that cooperative offloading can greatly improve the data delivery probability, the heuristic algorithm approximates the optimum, and the performance of both the centralized heuristic algorithm and distributed algorithm outperforms other approaches. The main contributions of this paper can be summarized as follows.

- We propose a probabilistic framework that provides the estimation of data delivery probability over the opportunistic path, considering both data size and contact duration. To the best of our knowledge, this is the first work that gives such estimation without any restrictions.
- We design a heuristic algorithm that performs path allocation and data assignment by carefully considering opportunistic contact probability and path capacity. The algorithm approximates the optimum at a much lower computational cost than PTAS.
- We design a distributed algorithm which employs criterion assignment, real-time adjustment and assignment update to make data offloading decisions at runtime. Although it only exploits paths no more than two hops for data forwarding, its performance is comparable to that of the heuristic algorithm.

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

II. RELATED WORK

Mobile opportunistic networks have been studied mainly for 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 costs. Many forwarding schemes have been proposed from different perspectives, such as representative strategies including [7][8][31], contact capability based schemes including [35][9], and social concepts based schemes including [5][4][38][15][24][25]. In addition, 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 on forwarding in mobile opportunistic networks is based on an assumption that nodes can completely transfer a data item during a node 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 relax this 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) has stimulated research on exploring mobile opportunistic networks for various applications. These applications can be generally classified into three categories: mitigating cellular traffic load, offloading computational tasks and information dissemination. The potential of mobile opportunistic networks in terms of reducing cellular traffic load was investigated in [21]. The incentive mechanism was studied in [39] and some solutions were proposed in [13][34][28][27]. Computational offloading among intermittently connected mobile devices was investigated in [29][18]. Information dissemination in mobile opportunistic networks was considered based on mobility and community in [32] and [26], respectively. Unlike the existing work, in this paper, we investigate application scenarios in which mobile devices cooperatively transmit data to intermittently connected infrastructure 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, in addition to transmitting a data item to the infrastructure along one particular path, the node can also select multiple paths to the infrastructure, and each path can be used to carry part of the data item.

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 (taken from the experiments in Section VII), 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 motivation, 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 $\sum_{n=1}^{4} (0.23)^n (0.77)^{4-n} = 0.407$, 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 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 $\sum_{n=1}^{4} (0.23)^n (0.77)^{4-n} > 0.504$, 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.

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 network 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 different path (i.e., each path at most carries one segment). Then, cooperative offloading can be mathematically formulated as

\[
\text{max} \quad \prod_{i=1}^{m} \prod_{j=1}^{n} P(S_i, j)^{x_{i,j}} \\
\text{s.t.} \quad \sum_{i=1}^{m} S_i = S, \quad m \in \{1, 2, \ldots, n\}, \quad \sum_{j=1}^{n} x_{i,j} = 1, \quad \sum_{i=1}^{m} x_{i,j} \in \{0, 1\}, \quad x_{i,j} \in \{0, 1\},
\]

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, $\sum_{i=1}^{m} x_{i,j} = 1$ ensures that each segment is assigned once, and $\sum_{j=1}^{n} x_{i,j} \in \{0, 1\}$ 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 on which path to transmit each segment so as to maximize the successful delivery probability of the entire data item before its deadline. Thus, it is a joint optimization of the number of data segments, the size of each segment, and the selection of path for each segment. Although the formulation of cooperative offloading is straightforward, it turns out to be NP-hard.

### D. NP-hardness

The NP-hardness of cooperative offloading can be proven by reduction to the minimization knapsack problem. In cooperative offloading, maximizing $\prod_{i=1}^{m} \prod_{j=1}^{n} P(S_i, j)^{x_{i,j}}$ is equivalent to minimizing

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} -\log P(S_i, j)x_{i,j},
\]

where $-\log P(S_i, j) > 0$ since $0 < P(S_i, j) < 1$. For a pair of segment $i$ and path $j$, it has a positive value of
can be rewritten as:

\[
\min \sum_{k=1}^{m} v_k x_k \\
\text{s.t.} \sum_{k=1}^{m} s_k x_k \geq S, \quad x_k \in \{0, 1\},
\]

where \(v_k\) and \(s_k\) respectively correspond the positive value and the size of a pair of segment and path. This is the minimization knapsack problem, which is NP-hard, and thus cooperative offloading is also NP-hard. Although cooperative offloading also admits PTAS (Polynomial Time Approximation Scheme) as the minimization knapsack problem, PTAS for cooperative offloading requires the knowledge of the value (the probability) for any pair of arbitrary data segment and path. However, in opportunistic mobile networks, the estimation of such probability requires high computation as we will discuss in Section [IV]. Since mobile devices have limited computation capabilities, PTAS is not affordable for mobile devices and thus we design better algorithms.

### IV. Probabilistic Framework

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 propose a probabilistic framework to estimate such probability based on node contact pattern. The existing work [16][40] simplifies the estimation by considering only one node contact along an opportunistic path before the deadline and thus severely underestimates the data delivery probability. To cope with this, we consider an arbitrary number of node contacts, and thereby our framework can accurately estimate the probability.

In opportunistic mobile networks, contact patterns between nodes have been well analyzed. Similar to [1][11], we model the contact process between nodes as the independent Poisson process. This modeling has been experimentally validated in [11]. Moreover, unlike [20][19] that assume only a fixed amount of data can be transferred during a node contact, or [36][11] that assume an arbitrary size of data can be sent during a node contact, we do not make such assumptions; i.e. the data amount that can be transmitted during a node contact depends on the contact duration as in [16][40]. In [16][40], 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 an estimation is not technically sound. Therefore, to accurately estimate the delivery probability, we consider the data amount that is brought by any arbitrary number of contacts between nodes within a time constraint. Moreover, as Wang et al. [33] validated that contact duration in opportunistic mobile networks follows the Pareto distribution, we also model contact duration between nodes as the Pareto distribution.

In the following, we first introduce opportunistic contact probability and data transfer probability, and then based on these, we show how to calculate data delivery probability.

#### A. 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 an Exponential distribution with rate parameter \(\lambda_1\), i.e. \(T_1 \sim \text{Exp}(\lambda_1)\), where \(\lambda_1\) is the rate parameter of the Poisson process between \(u\) and \(v\). Similarly, we have \(T_2 \sim \text{Exp}(\lambda_2)\) 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 \(P(T \leq T)\), where \(T = T_1 + T_2\). Note that the sequence of node contact is already captured by \(P(T \leq T)\). Assuming the probability density functions (PDF) of \(T_1\) and \(T_2\) are \(f_1(t)\) and \(f_2(t)\), respectively, \(P(T \leq T)\) can be calculated through the convolution \(f_1(t) \otimes f_2(t)\). However, a data item may not be completely transmitted during one contact between neighboring nodes, may 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 \(T_i, i = 1, \ldots, a\) and \(T_j, j = 1, \ldots, b\), we have \(T_1 \sim \text{Gamma}(a, \lambda_1)\), where \(T_1 = \sum_{i=1}^{a} T_i\). Similarly, for \(T_j, j = 1, \ldots, b\) and \(T_j \sim \text{Exp}(\lambda_j)\), we have \(T_j \sim \text{Gamma}(b, \lambda_j)\), where \(T_2 = \sum_{j=1}^{b} T_j\). Then the PDF of \(T = T_1 + T_2\) can be represented as

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

For a \(k\)-hop opportunistic path with the rate parameter \(\lambda_k\) and the contact number \(n_i\) at each hop \(i\) (we also call it \(k\)-hop opportunistic contact path), the PDF of \(T = T_1 + \cdots + T_k\) can be easily generalized as

\[
f(t) = f(t; n_1, \lambda_1) \otimes f(t; n_2, \lambda_2) \otimes \cdots \otimes f(t; n_k, \lambda_k) = \frac{t^{n_1-1}e^{-\lambda t}}{\lambda_1^{n_1} \Gamma(n_1)} \otimes \frac{t^{n_2-1}e^{-\lambda t}}{\lambda_2^{n_2} \Gamma(n_2)} \otimes \cdots \otimes \frac{t^{n_k-1}e^{-\lambda t}}{\lambda_k^{n_k} \Gamma(n_k)}.\]

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

\[
f(t) \cong \frac{t^{\gamma-1}e^{-\delta t}}{\delta \gamma \Gamma(\gamma)},
\]

where

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

Thus, \(T\) can be approximated by a single gamma random variable \(T \sim \text{Gamma}(\gamma, \delta)\). Then, the available probability of the opportunistic contact path with time constraint \(T\), denoted as \(P(T \leq T)\), can be easily calculated.
B. Data Transfer Probability

As the 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 shown in [40]. 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 $\alpha$ and scale parameter $\beta$ ($\beta$ is the minimum possible value of $D_1$), i.e. $D_1 \sim \text{Pareto}(\alpha, \beta)$. For a collection of i.i.d. random variables $\{D_i, i = 1, \ldots, c\}$ and $D_i \sim \text{Pareto}(\alpha, \beta)$, let $D = \sum_{i=1}^{c} D_i$. Then, the probability that a data item with size $D$ can be transferred with the number of contacts $c$ between node $u$ and $v$ is represented as $P(D \geq D)$ and can be derived from the PDF of $D$. However, since $D$ is the sum of an arbitrary number of random variables, its PDF cannot be easily approximated by stable distributions. Therefore, we choose to approximate $D$ by the maximum value of $\{D_i, i = 1, \ldots, c\}$ denoted as $M$, which relies on the observation that $M$ has the same order of magnitude as the sum $D$ since the Pareto distribution is a heavy-tailed distribution.

To capture such intuition, let us define $R = D/M$. Then, $D$ can be approximated by $M$, considering the ratio $R$. First let us consider the ratio of $D_1/M$. Then the Cumulative Density Function (CDF) of $D_1/M$ can be expressed in terms of the conditional distribution $F(x|y)$ of $D_1$ on the fixed maximum $M = y$ [37] as following

$$G(z) = P(D_1/M < z) = \int_{0}^{\infty} P(D_1 < z|y) df_c(y) = \int_{0}^{\infty} F(yz|y) df_c(y).$$

By taking the derivative with respect to $z$, then

$$g(z) = \int_{0}^{\infty} \frac{zf(yz|y) df_c(y)}{df_c(y)}.$$  \hspace{1cm} (4)

The conditional density $f(x|y)$ corresponding to the distribution $F(x|y)$ is given by

$$f(x|y) = \frac{\delta(y - x)}{c} + (1 - \frac{1}{c}) \frac{f(x) H(y - x)}{F(y)},$$  \hspace{1cm} (5)

where $H(x) = 1$ for $x > 0$, otherwise $H(x) = 0$. The first term on the rhs of (5) corresponds to the case $M = D_1$, whose probability is $1/c$, while the complementary event $D_1 < M$ occurs with probability $(1 - 1/c)$. By plugging (5) into (4), we then have

$$g(z) = \int_{0}^{\infty} \left( \frac{\delta(1 - z)}{c} + \frac{c - 1}{c} \cdot \frac{f(yz)H(y - yz)}{F(y)} \right) df_c(y) = \frac{\delta(1 - z)}{c} + (c - 1) \int_{0}^{\infty} yf(yz)f(x)F^{-2}(y)dy.$$  \hspace{1cm} (6)

Then, the expectation of the ratio $R = D/M$, denoted as $\bar{R}$, can be calculated as

$$\bar{R} = c \int_{0}^{1} zg(z)dz\hspace{1cm} = 1 + c(c - 1) \cdot\int_{0}^{\infty} yf(y)F^{-2}(y)\int_{0}^{1} zf(yz)dydz = 1 + c(c - 1) \cdot \int_{0}^{\infty} yf(y)F^{-2}(y) \left( \frac{F(y)}{y} - \frac{1}{y^2} \int_{0}^{y} F(x)dx dy \right) = 1 + (c - 1) \int_{0}^{\infty} df_c(y) - c \left( \frac{1}{y} \int_{0}^{y} F(u)du \right) df_c^{-1}(y) = c \left( 1 - \int_{0}^{\infty} \frac{1}{y} \int_{0}^{y} F(u)du \right) df_c^{-1}(y).$$

For Pareto distribution $F(x) = 1 - \left( \frac{\beta}{y} \right)^\alpha$, then

$$\bar{R} = c - c \int_{\beta}^{\infty} 1 - \frac{\beta}{y} \int_{1}^{\infty} d \left( 1 - \left( \frac{\beta}{y} \right)^\alpha \right)^{-1}. $$

By replacing the integral by beta function $B(\cdot, \cdot)$, finally we have

$$\bar{R} = \frac{1 - cB(c, \alpha - 1)}{\sum_{i=1}^{\infty} i^{-\alpha}, \hspace{0.5cm} \alpha \neq 1} = \frac{1}{\alpha}. \hspace{1cm} (6)$$

Thus, $P(D \geq D)$ can be approximated as

$$P(D \geq D) \approx P(M\bar{R} \geq D) = P(M \geq \frac{D}{\bar{R}})$$

$$= 1 - \prod_{i=1}^{c} P(D_i < \frac{D}{\bar{R}}) = 1 - \left( 1 - \left( \frac{\beta}{\bar{R}} \right)^\alpha \right)$$

where $\bar{R}$ can be easily calculated using (6).

C. 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 $P(T, D)$, based on the opportunistic contact probability and data transfer probability.

First, we consider the case of a one-hop path. The data item needs at most $l = \lceil \frac{T}{r} \rceil$ 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

$$P(T, D) = \sum_{i=1}^{l} \bar{P}_{i - 1} \cdot P(T_i \leq T - T') \cdot P(D_i \geq D),$$  \hspace{1cm} (7)

where

$$\bar{P}_i = \left\{ \begin{array}{ll} \prod_{j=1}^{i} P(T_j \leq T - T') \cdot P(D_j < D) & i > 0, \\
1 & i = 0, \end{array} \right.$$  \hspace{1cm} (8)

$T' = \frac{D}{r}$ denotes the transmission time of the data item and $r$ is the data rate between these two nodes. As in [7], the probability that $i$th contact completes the data transfer can be
interpreted as the probability that the data can be transferred within 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
\[
\langle n_1, \ldots, n_i, \ldots, n_k \rangle, \quad 1 \leq n_i \leq \lceil \frac{D}{\beta} \rceil, \quad i \in \mathbb{N}, \quad 1 \leq i \leq k \tag{8}
\]
which falls into one of the \( \prod_{i=1}^{k} \lceil \frac{D}{\beta} \rceil \) 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 the one-hop path, we need to consider the failure probability of tuples which have one less contact number at one of the hops. 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, and the computational complexity is \( \prod_{i=1}^{k} \lceil \frac{D}{\beta} \rceil \).

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.

V. HEURISTIC ALGORITHM

In this section, based on the proposed probabilistic framework, we give the design of the heuristic algorithm.

A. The 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 a certain amount of data to 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 prior knowledge of how much data will be assigned at each path, it is difficult to quantify the 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, \( \beta \) is the size of data that can be guaranteed to be transferred (according to the 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., \( \min(\beta, i = 1, \ldots, k) \)). 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 a direct path to \( v \), \( Q_i = 0 \)), then we assign the data amount of \( C^i_{uv} \) to path \( i \), i.e., \( S_i = C^i_{uv} \) if \( C^i_{uv} < S - \sum_{i \in \mathcal{P}} C^i_{uv} \), otherwise \( S_i = S - \sum_{i \in \mathcal{P}} C^i_{uv} \), and add \( i \) into the set of allocated paths.

Algorithm 1: Heuristic Algorithm

Input: \( u, v, S, T \)
Output: \( \mathcal{P}, S \)

while DijkstraMaxQ(\( u, v, S \)) \( \neq \emptyset \) do

\( p \leftarrow \) DijkstraMaxQ(\( u, v, S \))

\( Q_p \leftarrow Q \)

break
end

ExcludeAllocPath(p)

\( \mathcal{P} \leftarrow \mathcal{P} \cup \{ p \} \)

\( S \leftarrow S \cup \{ S_p \} \)

do

while \( S - \sum_{i \in \mathcal{P}} S_i \geq 0 \) do

\( p \leftarrow \arg \max_{i \in \mathcal{P}} P^i_{uv}(T, S_i) \)

\( q \leftarrow \arg \max_{j \in \mathcal{P} \setminus \{ p \}} P^j_{uv}(T, S_i) \)

while \( P^i_{uv}(T, S_p) \geq P^j_{uv}(T, S_q) \) \&\& \( S - \sum_{i \in \mathcal{P}} S_i > 0 \) do

\( S_p \leftarrow S_p + \Delta ; // \Delta \) is the increment of assigned data

\( S_j \leftarrow S_j - \Delta \)

end

\( S \leftarrow S \setminus \{ S_p \} \)

\( \mathcal{P} \leftarrow \mathcal{P} \setminus \{ p \} \)

\( S \leftarrow S \)

end

while \( S - \sum_{i \in \mathcal{P}} S_i \geq 0 \) do

\( \mathcal{P} \leftarrow \mathcal{P} \setminus \{ j \} \)

end

end

while \( S \neq \emptyset \) do

\( \mathcal{P} \leftarrow \mathcal{P} \setminus \{ j \} \)

\( S \leftarrow S \setminus \{ S_j \} \)

end

\( \mathcal{P} \leftarrow \mathcal{P} \)

end

\( S \leftarrow S \)

break

end

end
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: (i) \( Q_i < Q \); (ii) node \( v \) cannot be reached by Dijkstra’s algorithm from \( u \); (iii) \( \sum_{i \in \mathcal{P}_A} S_i = S \). If the searching process ends 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 \( S \).

The path searching process is from line 1 to 9 in Algorithm [1].

To assign the remaining data of \( S \) to \( \mathcal{P}_A \), we first we rank \( \mathcal{P}_A \) according to the delivery probability of the assigned data at each path, i.e., \( P_u(T, S_i), i \in \mathcal{P}_A \). 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. The capacity of the remaining data is from line 10 to 16 in Algorithm [1].

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 (from line 17 to 34 in Algorithm [1]). If the reallocation of \( S_i \) improves the delivery probability of \( S \), path \( i \) is included in \( \mathcal{P}_A \) 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.

B. Discussion

In the algorithm, we exclude the allocated paths from the subsequent path searches, because it is extremely hard to calculate the data delivery probability if multiple data segments are transmitted along an edge which is shared by multiple allocated paths. Therefore, we choose to avoid the difficulty by regulating an edge to be occupied by only one allocated path. Moreover, this also limits the maximum number of allocated paths to the minimum node degree (number of neighbors) of source and destination, and thus path allocation using Dijkstra’s algorithm can terminate quickly.

Intuitively, multi-hop paths should be considered for data transmission when its available probability is higher than a one-hop path. Therefore, \( Q' \) is employed as a criterion to allocate paths. As indicated in Section [V] the computational complexity of data delivery probability over multi-hop paths increases with the number of hops. However, due to the adoption of \( Q' \), allocated paths are usually limited to paths with fewer hops and thus we can avoid the high computational complexity of delivery probability when data that is more than the path capacity is assigned to a multi-hop path.

Data assignment at each allocated path is initially set to the path capacity, and thus the data delivery probability at each path is the same as \( Q \). For the case where data assigned more than the path capacity can severely decrease the delivery probability, this is obviously a better choice. However, this choice also works for other cases since it is unknown which path is sensitive to the increase of assigned data beforehand. After path allocation and initial data assignment, the number of data segments is the same as that of allocated paths. Then, data is reallocated from paths of lower delivery probability to ones of higher probability so as to improve the delivery probability of the data item by reducing the number of allocated paths. This complies with the design intuitions that the number of data segments should be small and each allocated path should have a high delivery probability.

Moreover, the computational complexity of the heuristic algorithm is much lower than PTAS for cooperative offloading. PTAS that needs to search all possible paths (they can be very long) between source and destination; the heuristic algorithm only needs to locate the paths, the available probability of which is more than \( Q' \). In addition, PTAS has to calculate the delivery probability of each path when carrying all possible amounts of data. This incurs the most of the computation. Unlike PTAS, the heuristic algorithm only needs to calculate certain amounts of data for the few allocated paths. Due to the uncertainty of the number of paths, path length and path capacity, it is hard to give the mathematical comparison of the computational complexity between the heuristic algorithm and PTAS. However, based on the reasoning stated above, it is easy to conclude that the computation of the heuristic algorithm is much less that of PTAS, and its performance is close to the optimum as we will demonstrate in Section [VII].

VI. DISTRIBUTED ALGORITHM

Global information is required for the heuristic algorithm presented above. However, such information might not be available or cost too much to collect 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 two-hop information of contact frequency and contact duration between nodes; i.e., each node maintains contact frequency and contact duration with its neighbors, and when two nodes encounter, they will exchange such information and then they will have the two-hop information. The goal 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 the collected information so as to improve the delivery probability.

With the two-hop information maintained locally, the source node can initially construct paths to the destination (infrastruc-
The following shows 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 $P^{d}\left(\mathcal{T}_{c_i}(T, S_j, S_d + S_j) \cdot P_{cv}^{d}(T, S_j) \cdot P_{cv}^{d}(T, S_e + S_j)\right)$ and $P^{e}\left(\mathcal{T}_{c_i}(T, S_d + S_j) \cdot P_{cv}^{e}(T, S_d) \cdot P_{cv}^{e}(T, S_e + S_j)\right)$ respectively. If $P^{d}\left(\mathcal{T}_{c_i}(T, S_d + S_j) \cdot P_{cv}^{d}(T, S_d) \cdot P_{cv}^{d}(T, S_e + S_j)\right)$ 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 $P^{d}\left(\mathcal{T}_{c_i}(T, S_j) \cdot P_{cv}^{d}(T, S_d) \cdot P_{cv}^{d}(T, S_d + S_j)\right)$ and $P^{e}\left(\mathcal{T}_{c_i}(T, S_d + S_j) \cdot P_{cv}^{e}(T, S_d) \cdot P_{cv}^{e}(T, S_d + S_j)\right)$. 

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 $\mathcal{S}_c$, 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 $\mathcal{S}_c$ should be excluded.
from the assignment $\mathcal{A}_n$ by sequentially removing the data assigned at the path with the lowest delivery probability. Similarly, for node $c$, the amount $S_c - S_c'$ needs to be removed from $\mathcal{A}_c$.

D. Discussion

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.

In the design of the distributed algorithm, nodes are only required to maintain two-hop information. One motivation for this is that, as stated above, nodes opportunistically contact each other and hence locally maintained multi-hop information cannot be promptly updated. Another reason is that by regulating to two-hop paths, the data delivery probability over the paths can be easily calculated and thus nodes can quickly make decisions at runtime, e.g. at the phase of real-time adjustment.

Although the distributed algorithm requires that the source node must have a path with no more than two hops to infrastructure, it can easily establish these paths through its neighbors since most nodes in the network intermittently connect with infrastructure. Moreover, intermediate nodes, e.g. node $c$ in Figure 3B, will also exploit two-hop paths to forward data. Therefore, there are enough paths to be explored for data transmission.

VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the heuristic algorithm and 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-know benchmark [17]. 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, $\mu_w$; the mixing parameter for the topology, $\mu_t$; the exponent for the weight distribution, $\xi$; the average node degree, $d$; and the maximum node degree, $d_m$. The settings of these parameters are shown in Table I.

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., $\alpha, \beta$ and $\lambda$) are shown in Table II, where $\alpha = [3, 4]$, for example, means that $\alpha$ between pair of nodes is set to a random number between 3 and 4, which follows the uniform distribution.

When a node transmits data to the 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 the 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.

![Fig. 4: Data Delivery probability based on estimation and simulation for Individual and Cooperative in synthetic networks, where $d = 10$, $d_m = 15$, $\alpha = [6, 10]$, $\lambda' = [0.001, 0.1]$.](image)

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

TABLE I: Parameter settings for benchmark

| Parameter | Value | Meaning |
|-----------|-------|---------|
| $n$       | 100   | number of nodes |
| $\mu_w$   | 0.3   | mixing parameter for weights |
| $\mu_t$   | 0.3   | mixing parameter for topology |
| $\xi$     | 2     | exponent for weight distribution |
| $d$       | 5, 10 | average node degree |
| $d_m$     | 8, 15 | maximum node degree |

TABLE II: Parameter settings for node contacts

| Type         | Parameter | Value         |
|--------------|-----------|---------------|
| Between nodes| $\alpha$  | $[3, 4]$, $[6, 10]$ |
| $\beta$      |           | $[2, 3]$ |
| $\lambda$    |           | $[0.001, 0.1]$ |
| Between node and infrastructure | $\alpha'$ | $[3, 4]$ |
| $\beta'$     |           | $[2, 3]$ |
| $\lambda'$   |           | $[0.001, 0.1]$ |
nodes according to their distributions, then data is transmitted between nodes based on these generated contact information.

**Cooperative does help.** 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 delivery 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. Figure 4b shows the probability of data transmissions with varying deadlines. Similarly, Cooperative and Individual start at the same probability. The difference between them expands and then narrows when the deadline further loosens.

**Delivery probability estimation.** As shown in Figure 4a for Individual, the estimated delivery 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 approximation of sum of Pareto random variables as discussed in Section IV. When the data size increases, more node contacts are required to transfer a data item between neighboring nodes, which may increase the deviation of the approximation. As the deadline is irrelevant to the approximation, the increase of the deadline does not impact the difference between the estimated and simulated probabilities, and they are almost identical as shown in Figure 4b.

**Cooperative approximates the optimum.** We also compare Cooperative to the optimal solution of the cooperative offloading problem, denoted as Optimal. Due to the high computational complexity of Optimal, we choose a problem of small size (i.e., we select one node in a generated synthetic network to send data to infrastructure) and use brute-force search to find the optimum.

Figures 5a and 5b depict the delivery probabilities of Cooperative and Optimal in terms of varying data size and deadline, respectively. Cooperative tightly approximates Optimal in both cases and different data sizes and deadlines only slightly affect their disparities.

**When data offloading benefits.** Figure 5 gives the relation between percentage of offloaded data transmissions (over all the data transmissions with different data sizes and deadlines) and contact pattern with infrastructure at particular nodes, where contact pattern is captured by expected contact duration and inter-contact duration between a node and infrastructure. Intuitively, nodes that frequently contact with infrastructure or have long contact duration with infrastructure do not need to offload the data as much as other nodes. Consistent with this intuition, as shown in Figure 6a the shorter the contact duration, the more transmissions are offloaded. Moreover, as depicted in Figure 6b the more frequently a node contacts infrastructure, the less transmissions are offloaded.

**Networking settings affect data offloading.** Figure 7 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 7a 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 7a for the network with $\alpha = [6, 10]$ and $\lambda = [0.001, 0.01]$, most of the transmissions are offloaded (i.e., most of area is 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 7b, 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 7b. 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 7c), compared to Figure 7a, 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 a network, e.g., $d = 5$ and $d_m = 8$ as in Figure 7d, compared to Figure 7a, the reduced node degree undermines the offloading capability of the network; i.e., the offloaded transmissions are generally less than that of Figure 7a. Compared with Figure 7d, inter-contact duration between node and infrastructure is decreased ($\lambda = [0.01, 0.1]$) in Figure 7c. This dramatically changes the pattern of the offloaded transmissions. As shown in Figure 7c, offloading can improve the probability of most transmissions but not for the transmissions with large data sizes and long deadlines. That is because when nodes frequently contact infrastructure, it is easy for nodes to transmit a data item to infrastructure by multiple contacts and offloading the data item to other nodes may not yield a high delivery probability. Last, let us compare Figure 7f with Figure 7c. 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, through the extensive experiments on synthetic networks, we can conclude that the proposed probabilistic framework can accurately estimate data delivery probability over opportunistic paths, cooperative data offloading can significantly improve data delivery probability in various network settings and contact patterns, and the data delivery probability achieved by the heuristic algorithm is close to the optimum.

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 [6] and DieselNet [11]. 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.

| 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 two-hop or one-hop to infrastructure are excluded from the traces (note that only very few nodes are eliminated). In the simulation, nodes send data items 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, 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.

Figures 8a and 8b give 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 8a, 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 8b, 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 the heuristic algorithm performs 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 that only exploits paths no more than two hops for data transmission is comparable to that of the heuristic algorithm.

VIII. 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 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 the probabilistic framework accurately estimates the data delivery probability, cooperative offloading greatly improves the delivery probability, the heuristic algorithm approximates the optimum, and the performance of the heuristic algorithm and distributed algorithm outperforms other approaches.

Acknowledgment

This work was supported in part by Network Science CTA under grant W911NF-09-2-0053 and CERDEC via contract W911NF-09-D-0006. A preliminary version of this work was accepted and will appear in the Proceedings of IEEE INFOCOM 2016 [23].

References

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In Proc. ACM SIGCOMM, 2007.
[2] N. Banerjee, M. D. Corner, and B. N. Levine. An energy-efficient architecture for dtn throwboxes. In Proc. IEEE INFOCOM, 2007.
