Opportunistic Networks: Minimizing Expected Latency

Shohreh Shaghaghian and Mark Coates
Department of Electrical and Computer Engineering, McGill University
3480 University St., Montreal, QC, Canada H3A 0E9
shohreh.shaghaghian@mail.mcgill.ca, mark.coates@mgcill.ca

Abstract—**Even in the absence of a physical network infrastructure, it is important that mobile devices can take advantage of the opportunities presented to them to maintain communication. Opportunistic networks are a class of mobile networks that must rely on unscheduled sporadic meetings between nodes to achieve communication. The main challenge in these networks is to route messages towards a destination with high delivery probability, low average latency and efficient usage of network resources. Despite the numerous proposals for opportunistic routing protocols, there are relatively few analytical characterizations of their behaviour. In this paper, we propose both centralized and decentralized message forwarding algorithms that, under relatively strong assumptions about the network's behaviour, minimize the expected latency to a destination. After presenting proofs of the optimality of our proposed algorithms, we confirm their improvement in system performance through numerical simulation.**

I. Introduction

In Delay/Disruption Tolerant Networks (DTNs), a communication path between any two nodes is frequently unavailable and nodes are only intermittently connected. Due to the dynamic nature of DTNs, routing is usually performed based on a store-carry-forward mechanism. The source transmits a message to a node it meets; this intermediate node stores the message by buffering it, then carries it (the mobility of the node is exploited), before forwarding it to another node it meets. The term *opportunistic networks* is used to refer to DTNs where the meetings between nodes are unpredictable. Opportunistic networks have been studied in the context of non-intrusive wildlife tracking (e.g. ZebraNet [1], SWIM [2]), provision of data communication to remote/rural areas (e.g., DakNet [3]) and offloading of mobile data traffic (e.g., [4]).

Routing methods in opportunistic networks can be classified as replication-based, history-based and social-based algorithms. In replication-based routing algorithms such as Epidemic [5] and Spray and wait [6], a node forwards messages stored in its buffer to all or a fraction of encountered nodes without assessing their capability to deliver them. The only parameters in these algorithms are the extent of replication and the nodes that are allowed to make new copies. The replication approaches can achieve a higher probability of message delivery since more nodes have a copy of each message, but they can produce significant network congestion.

History-based routing algorithms assume that nodes' movement patterns are not completely random and that the rate of future contacts between two nodes depend on the frequency and number of past encounters and possibly the time gaps between them. *PRoPHET* [7] assigns a "delivery predictability" metric to each node for each destination. This metric is updated each time two nodes meet, and thus takes into account the history of meetings in the network. The metric is intended to indicate "how likely it is that this node will be able to deliver a message to that destination" [7] (perhaps via other nodes). *MaxProp* [8] and *MEED* [9], proposed for vehicular DTNs, are other examples of this class of algorithms.

Social-based algorithms incorporate the principle that mobile devices are usually carried by members of a society and therefore the contact patterns depend on their social interactions. This suggests that social network analysis concepts and community formations can be used to better predict the node meeting patterns and identify superior forwarding rules. In *Simbet* [10], a node decides to forward a message to other nodes based on their betweenness centralities and a social similarity metric. *BubbleRap* [11] makes use of community affiliation and centrality to make forwarding decisions. The contact graph based routing algorithm in [12] uses information derived from a weighted contact graph to make forwarding decisions based on the neighbourhood, community and degree centrality of a node.

More recent studies have focused on modeling the network with more realistic features such as limited buffer size [13], constrained message generation rate [14], and selfish mobile nodes [15], [16]. Most of the studies cited above use experimental analysis to demonstrate the efficiency of their proposed method (primarily simulations based on traces recorded from opportunistic networks). While the experimental analysis is valuable and takes into account more practical considerations, it can leave us with an incomplete understanding of how an algorithm operates and how it will perform in other untested network conditions. For example, the behaviour of PRoPHET has been shown to be very sensitive to parameter choice [17]. It is also useful to design an optimal algorithm under somewhat unrealistic modeling assumptions, and then consider how it can be adapted to address more practical considerations, without completely losing its desirable features. There are a few studies that derive a forwarding process whose optimality (in some sense) can be mathematically proved under relatively strong assumptions about the behaviour of the network. In [18] the expected delivery time to the destination has been considered as a metric to define the best set of candidate relays in two hop relay strategy introduced in [19]. By increasing the number of relaying steps recursively, a centralized single-copy opportunistic routing scheme is proposed for sparse DTNs. The algorithm is shown to be loop free because messages are forwarded only to the nodes that have less expected delivery time to the destination. It is also proved that the recursive process has a polynomial running time and the number of steps is finite.

In this paper, we propose centralized and distributed versions of an algorithm for assigning forwarding rules to each node in an opportunistic network so that the sum of average latencies from all the nodes of the network to a particular destination is minimized. We prove the optimality of our suggested methodology under certain modeling assumptions and assumed node knowledge. We then explore through simulation how the relaxation of some of these modeling assumptions impacts performance. The paper is organized as follows. In Section II, we define the system model and state the problem. In Section III, we propose a forwarding algorithm and prove that it achieves a form of optimality under the network modeling assumptions. We present numerical simulation results in Section IV and make concluding remarks in Section V.

II. System Model

We consider a closed network of N mobile nodes which aim to send messages to a particular destination node d. The set of nodes is denoted by $\mathcal{N}$. We assume that the random intermeeting times of nodes are independent and exponentially distributed with parameter λ_{ij} for nodes i and j. We associate with the network a *contact graph* which is formed by adding a link between any two nodes that meet. We assume that the contact graph is connected and denote the set of neighbors of node i in this graph by $\mathcal{S}_i$. fSince the contacts between nodes are not pre-scheduled, we cannot identify end-to-end paths ahead of time. Hence, solving the routing task is equivalent to identifying the forwarding decisions that nodes should make when meeting each other. We assume that buffer size is unlimited, message Time To Live (TTL) is infinity and that any number of messages can be forwarded during each meeting.

We consider only algorithms that do not involve replication. In the class of algorithms we consider, each time node i meets one of its neighbors $j \in \mathcal{S}_i$, it forwards a message destined for d with probability p_{ij}. Considering the matrix $\mathbf{P}_{N\times N}$ comprised of all pairs i and j, we set $p_{ij} = 0$ if nodes i and j never meet and are thus not neighbors in the contact graph. We denote the forwarding probabilities of node i by the vector $\mathbf{p}_i$; this is the i-th row of the matrix $\mathbf{P}$.

The expected latency from node i to destination d is a function of the probability decision matrix $\mathbf{P}$ and we denote it by $L_{id}(\mathbf{P})$. Our goal is to find the matrix $\mathbf{P}^*$ such that the sum of the expected latencies of all the nodes in the network to the specified destination d is minimized. Let us call this utility function $\mathcal{U}(\mathbf{P}) = \sum_{i\in\mathcal{N}} L_{id}(\mathbf{P})$. We assume that the network topologies and meeting rates are such that the solution $\mathbf{P}^*$ is unique. If not, our algorithms guarantee that we reach one of the optimal matrices, but the proofs are more complicated.

Lemma 1 provides an expression for $L_{id}(\mathbf{P})$ in terms of $\mathbf{P}$ and the expected latencies of its neigbours.

Lemma 1. *The expected latency of a node $i \in \mathcal{N}$ to the destination d is*

$$L_{id}(\mathbf{P}) = \frac{1 + \sum_{j\in\mathcal{S}_i} p_{ij}\lambda_{ij}L_{jd}(\mathbf{P})}{\sum_{j\in\mathcal{S}_i} p_{ij}\lambda_{ij}} \tag{1}$$

Proof: When node i commences in its routing of a packet, it must first wait a time T_w before it meets one of its neighbors. The amount of time before node i meets a specific neighbor j is an exponentially distributed random variable with parameter λ_{ij}. The time T_w is equal to the minimum of the exponentially distributed random variables corresponding to all neighbours $k \in \mathcal{S}_i$ and its expected value is

$$E(T_w) = \frac{1}{\sum\limits_{k\in\mathcal{S}_i} \lambda_{ik}} \tag{2}$$

The probability that j is the first node that i meets is $\frac{\lambda_{ij}}{\sum_{k\in\mathcal{S}_i}\lambda_{ik}}$. Hence $L_{id}(\mathbf{P})$ is

$$L_{id}(\mathbf{P}) = E(T_w) + \sum_{j\in\mathcal{S}_i} \frac{\lambda_{ij}}{\sum\limits_{k\in\mathcal{S}_i}\lambda_{ik}}[p_{ij}L_{jd}(\mathbf{P}) + (1-p_{ij})L_{id}(\mathbf{P})] \tag{3}$$

The last term in (3) follows from the memoryless property of the distributions. Subsituting (2) into (3) leads to (1). ■

Before continuing to propose algorithms for finding $\mathbf{P}^*$, we make two points about the structure of $\mathbf{P}^*$ through the following theorem.

Theorem 1. *Suppose $\mathbf{P}^* = \arg\min_{P\in[0,1]^{N\times N}} \sum_{i=1}^{N} L_{id}(\mathbf{P})$. Then:*

1. *$\mathbf{P}^*$ is a binary matrix (its components are either 0 or 1).*
2. *For any $i \in \mathcal{N}$, the matrix $\mathbf{P}^*$ also minimizes $L_{id}(\mathbf{P})$:*

$$\forall i \in \mathcal{N}: \quad \mathbf{P}^* = \arg\min_{P\in[0,1]^{N\times N}} L_{id}(\mathbf{P}) \tag{4}$$

Proof: Let us assume that the nodes, excluding d, are labelled in ascending order of their expected latency under $\mathbf{P}^*$, i.e., $L_{1d}(\mathbf{P}^*) < L_{2d}(\mathbf{P}^*) < \cdots < L_{N-1d}(\mathbf{P}^*)$. For a given matrix $\mathbf{P}$, we denote by $\mathbf{P}_{\bar{i}}$ all rows of $\mathbf{P}$ except i. If we fix $\mathbf{p}_i$ and L_{kd} for $k \in \mathcal{S}_i, k \neq j$ for some $j \in \mathcal{S}_i$, then L_{id} is monotonically increasing with respect to L_{jd} (see (1)). This implies that if we commence with any $\mathbf{P}$ and change only $\mathbf{p}_j$ to decrease L_{jd}, then all other L_{id} such that $j \in \mathcal{S}_i$ either decrease or remain the same. The matrix $\mathbf{P}^*$ must therefore satisfy $\mathbf{p}_j^* = \arg\min L_{jd}(\mathbf{P}_{\bar{j}}^*, \mathbf{p}_j)$ for all j. Otherwise we could choose an alternative $\mathbf{p}_j'$ that reduces L_{jd} and hence achieves $\mathcal{U}(\mathbf{P}') < \mathcal{U}(\mathbf{P}^*)$.

We can examine the partial derivative of L_{id} with respect to p_{ij} at $\mathbf{P}' = (\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$:

$$\frac{\partial L_{id}}{\partial p_{ij}} = \frac{\lambda_{ij}[\sum_{k\in\mathcal{S}_i}\lambda_{ik}p_{ik}(L_{jd}(\mathbf{P}') - L_{kd}(\mathbf{P}')) - 1]}{(\sum_{k\in\mathcal{S}_i}\lambda_{ik}p_{ik})^2} \tag{5}$$

This derivative has the same sign as: $L_{jd}(\mathbf{P}') - \frac{1+\sum_{k\in\mathcal{S}_l}\lambda_{ik}p_{ik}L_{kd}(\mathbf{P}')}{\sum_{k\in\mathcal{S}_i}\lambda_{ik}p_{ik}}$, or equivalently $L_{jd}(\mathbf{P}') - L_{id}(\mathbf{P}')$.

This expression for the derivative, together with the requirement that $\mathbf{p}_i^* = \arg\min L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$, implies that $\mathbf{p}_{ij}^* = 0$ if $L_{id}(\mathbf{P}^*) < L_{jd}(\mathbf{P}^*)$ and $\mathbf{p}_{ij}^* = 1$ if $L_{id}(\mathbf{P}^*) > L_{jd}(\mathbf{P}^*)$. Our assumption that the solution is unique implies that $L_{id}(\mathbf{P}^*) \neq L_{jd}(\mathbf{P}^*)$. Otherwise, from (3), it is clear that we could choose any p_{ij}^* between 0 and 1 and achieve the same $L_{id}(\mathbf{P}^*)$, without affecting any other $L_{jd}(\mathbf{P}^*)$. This establishes statement (1) of the theorem.

Although we have established that $L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i^*) = \min L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$, we have not yet shown that $\mathbf{P}^*$ globally minimizes L_{id}. We establish this by contradiction. Suppose $\mathbf{P}^*$ does not minimize the expected latency for some non-empty set of nodes $\mathcal{N}' \subset \mathcal{N}$. In other words, denoting the minimum expected latency achieved via the minimization in (4) for node i by L_{id}^*, we have

$$\forall i \in \mathcal{N}' : \quad L_{id}^* < L_{id}(\mathbf{P}^*) \tag{6}$$

Let node s be the node in $\mathcal{N}'$ such that $L_{sd}^* < L_{kd}^*$ for all $k \in \mathcal{N}', k \neq s$. Denote by ℓ the ranking of the node with greatest expected latency under $\mathbf{P}^*$ such that $L_{\ell d}(\mathbf{P}^*) < L_{sd}^*$. Based on the discussion above, for each node $i \in \{1, 2, \ldots, \ell\}$, $p_{ik}^* = 0$ for all $k > \ell$ and hence $p_{is}^* = 0$. Node s must have at least one neighbour in the set $\{d, 1, 2, \ldots \ell\}$. Otherwise, it could not achieve an expected latency under $\mathbf{P}^*$ that is less than all nodes $\ell + 1, \ldots, N - 1$ (observe from (1) that $L_{sd}(\mathbf{P}^*) > \min_{p_{sj}>0} L_{jd}(\mathbf{P}^*)$).

The matrix $\mathbf{P}'$ that achieves the minimum L_{sd}^* must satisfy $p'_{sk} = 0$ for all $k \in \mathcal{N}'$, since for any matrix $\mathbf{P}'$ we have $L_{kd}(\mathbf{P}') \geq L_{kd}^* > L_{sd}^*$. We also have $p'_{sk} = 1$ for $k \in \mathcal{S}_s \cap \{d, 1, 2, \ldots \ell\}$ if $L_{kd}(\mathbf{P}') < L_{sd}(\mathbf{P}')$. For a fixed choice of $\mathbf{p}'_s$ the value $L_{sd}(\mathbf{P}')$ decreases if we can reduce $L_{kd}(\mathbf{P}')$ for any k such that $p'_{sk} = 1$. The matrix $\mathbf{P}^*$ minimizes L_{kd} for all $k \in \{1, 2, \ldots, \ell\}$, implying that $\mathbf{p}'_k = \mathbf{p}_k^*$ for all $k \in \{1, 2, \ldots, \ell\}$. Since $L_{kd}(\mathbf{P}') = L_{kd}(\mathbf{P}^*)$ for all $k \in \mathcal{S}_s \cap \{d, 1, 2, \ldots \ell\}$, it follows that $\mathbf{p}'_s = \mathbf{p}_s^*$. For node s, the values of $\mathbf{p}'_j$ for $j \notin \{1, 2, \ldots, \ell, s\}$ have no impact on L_{sd}, so we have $L_{sd}(\mathbf{P}^*) = L_{sd}(\mathbf{P}') = L_{sd}^*$. This contradicts the original assumption that $\mathbf{P}^*$ does not minimize the latency for all nodes $s \in \mathcal{N}'$, and thus establishes statement (2) of the theorem. ■

Theorem 1 showed that the minimization problem is actually a binary problem. Each time node i meets one of its neighbours $j \in \mathcal{S}_i$, it either forwards the message or keeps it. From now on, we change our notation and use the binary indicator matrix $\mathbf{B}$ instead of $\mathbf{P}$ to capture this binary decision. Therefore, the optimization takes the form:

$$\mathbf{B}^* = \arg \min_{\mathbf{B} \in \{0,1\}^{N \times N}} \sum_{i=1}^{N} L_{id}(\mathbf{B}) \tag{7}$$

In the next section, we state the algorithms we have proposed for solving this optimization problem and prove that they find the optimal solution.

III. Algorithm

In the first part of this section, we try to to find $\mathbf{B}^*$ in a centralized fashion where the whole topology and meeting rates of the network are available at a central unit. This unit calculates $\mathbf{B}^*$ and informs the nodes about which neighbours they should forward their buffered messages to. In the second part of this section we introduce a decentralized algorithm and prove that it converges to the same global solution. The advantage of the decentralized approach is that no node needs global knowledge of the network and each node can learn its own optimal forwarding decisions. The only piece of information a node needs to know is its meeting rates with its own neighbours. A step forward towards improving the decentralized partial knowledge approach is that each node estimates its meeting rates with other nodes it meets and modifies the estimations each time a contact occurs. This step is discussed in the third part of this section where we assume that nodes have no a-priori knowledge of any meeting rates.

Algorithm 1 Centralized Greedy Latency Minimization

```
// Initialization
$\mathcal{A} = \{d\}$, $\mathbf{B} = \mathbf{0}_{N \times N}$, $L_{dd} = 0$, $L_{jd} = \infty$ for $j \neq d$
// Iteratively add nodes to the set
while $\mathcal{A} \neq \mathcal{N}$ do
    for each node $i \notin \mathcal{A}$ do
        Identify $\mathcal{S}_{i\mathcal{A}} = \mathcal{S}_i \cap \mathcal{A}$
        Calculate $G_{id}$ for $\mathbf{m}_i^* \in \{0,1\}^{|\mathcal{S}_{i\mathcal{A}}|}$
        $G_{id} = \min_{\mathbf{m}_i} \frac{1+\sum_{j \in \mathcal{S}_{i\mathcal{A}}} m_{ij}\lambda_{ij}L_{jd}}{\sum_{j \in \mathcal{S}_{i\mathcal{A}}} m_{ij}\lambda_{ij}}$
        Denote $\mathcal{D}_i = \{j | j \in \mathcal{S}_{i\mathcal{A}}, m_{ij}^* = 1\}$
    end for
    Identify $v = \arg\min_{i \in \mathcal{N}/\mathcal{A}} G_{id}$
    Set $L_{vd} = G_{vd}$
    Set $b_{vj} = 1$ for all $j \in \mathcal{D}_i$
    Set $\mathcal{A} = \mathcal{A} \cup \{v\}$
end while
```

A. Centralized Approach with Global Knowledge

Suppose for each node, $i \in \mathcal{N}$ the set of neighbours $\mathcal{S}_i$ and their meeting rates $\lambda_{ij}, j \in \mathcal{S}_i$ are known at a central caluculation unit. Algorithm 1 presents an iterative procedure to identify $\mathbf{B}$. Theorem 2 demonstrates that the iterative optimization procedure expressed in Algorithm 1 solves the minimization task in (7).

Theorem 2. *Suppose all meeting rates are different and there exists a unique solution $\mathbf{B}^*$ for the optimization problem* (7).

1) *After each iteration of Algorithm 1,*
    a) $\forall i \in \mathcal{A}, \forall j \in \mathcal{N} : \quad b_{ij} = b_{ij}^*$
    b) $\forall i \in \mathcal{A} : \quad L_{id}(\mathbf{B}) = L_{id}(\mathbf{B}^*)$
    c) $\max_{i \in \mathcal{A}} L_{id}(\mathbf{B}^*) < \min_{i \notin \mathcal{A}} L_{id}(\mathbf{B}^*)$
2) *Upon completion, Algorithm 1 identifies a labelling $\mathbf{B}$ and associated expected latencies L_{id} such that $\mathbf{B} = \mathbf{B}^*$*

Proof: We observe that for all $i \in \mathcal{N}$, $G_{id} \geq L_{id}^*$ (since the optimizations are the same). Based on Theorem 1 and its proof, the equality holds only if $j \in \mathcal{A}$ for all $j \in \mathcal{S}_i$ such that $L^*_{jd} < L^*_{id}$. The statements in the theorem follow based on an induction argument.

Suppose, without loss of generality, that the nodes are labelled in ascending order of expected latency under B^*. For node 1, the only neighbour with lower expected latency is the destination. In iteration 1, the destination is included in $\mathcal{A}$ and must be in $\mathcal{S}_1$. Recall that $L^*_{1d} < \min_{b^*_{1j}=1} L^*_{jd}$. Node 1 has the minimum expected latency according to the chosen labelling and Theorem 1, except for the destination itself. The relationship thus implies that $d \in \mathcal{S}_1$. We therefore have $G_{1d} = L^*_{1d} < L^*_{jd} \leq G_{jd}$, and node 1 is selected to be added to $\mathcal{A}$, with $b_{1d} = 1$ and $b_{jd} = 0$ for all $j \neq d$. Statements 1a)-c) in the theorem clearly hold one iteration, i.e. after the addition of node 1 to $\mathcal{A}$.

Assume the same statements hold after the addition of node $k-1$ to $\mathcal{A}$. Then, for node k we must have $j \in \mathcal{A}$ for all $j \in \mathcal{S}_k$ such that $L^*_{jd} < L^*_{kd}$. Again this implies that $G_{kd} = L^*_{kd} < L^*_{jd} \leq G_{jd}$ for all $j > k$. Thus, node k is correctly selected for addition to $\mathcal{A}$ and the statements 1a)-c) hold at the end of iteration k.

It follows that the statements hold for all iterations of the algorithm, and after completion, when $\mathcal{A} = \mathcal{N}$, the second statement follows. ■

B. Decentralized Approach with Partial A-priori Knowledge

As opposed to the previous part, suppose no central unit exists and each node is just aware of its own $\mathcal{S}_i$ and the meeting rates $\lambda_{ij}, j \in \mathcal{S}_i$. Algorithm 2 demonstrates how nodes can make their binary forwarding decisions based on this local information. We denote by $L_{id}(j)$ the estimate at node j of the latency from node i to the destination. Theorem 3 proves that this decentralized approach results in the same global optimum solution.

Algorithm 2 Decentralized Greedy Latency Minimization

```
1:  // Initialization
2:  $\mathbf{B} = \mathbf{0}_{N \times N}$
3:  $\forall i \in \mathcal{N}/d, \forall j \in \mathcal{N}:\quad L_{dd}(j) = 0,\ L_{id}(j) = \infty$
4:  while Nodes continue to meet do
5:      // Nodes $i$ and $j$ meet at time $t$
6:      Set $L_{id}(j) = L_{id}(i)$
7:      Set $L_{jd}(i) = L_{jd}(j)$
8:      Update $L_{id}(i) = \min_{\mathbf{m}_i \in \{0,1\}^{|\mathcal{S}_i|}} \frac{1+\sum_{k\in\mathcal{S}_i} m_{ik}\lambda_{ik}L_{kd}(i)}{\sum_{k\in\mathcal{S}_i} m_{ik}\lambda_{ik}}$
        and identify the minimizing $\mathbf{m}^*_i$
9:      Update $L_{jd}(j) = \min_{\mathbf{m}_j \in \{0,1\}^{|\mathcal{S}_j|}} \frac{1+\sum_{k\in\mathcal{S}_j} m_{ik}\lambda_{ik}L_{kd}(i)}{\sum_{k\in\mathcal{S}_j} m_{ik}\lambda_{ik}}$
        and identify the minimizing $\mathbf{m}^*_j$
10:     Set $\mathbf{b}_i = \mathbf{m}^*_i$ and $\mathbf{b}_j = \mathbf{m}^*_j$
11: end while
```

Theorem 3. *The decision matrix* $\mathbf{B}$ *identified by Algorithm 2 converges to* $\mathbf{B}^*$ *with probability* 1.

Proof: Assume that the nodes are labelled in order of ascending expected latency under $\mathbf{B}^*$, i.e., $L_{1d}(\mathbf{B}^*) \leq L_{2d}(\mathbf{B}^*) \leq \cdots \leq L_{N-1d}(\mathbf{B}^*)$. Denote by T_1 the moment of time at which node 1 meets the destination node. For $k = 2, \ldots, N$ denote by T_k the earliest time by which node k has met all nodes in the non-empty set $\{1, \ldots, k-1\} \cap \mathcal{S}_k$ in the time period $(T_{k-1}, T_k]$. Due to the assumption that the inter-meeting times are exponentially distributed, T_N is finite with probability 1.

At T_1, node 1 knows its meeting rate with the destination (λ_{1d}). Since the latencies are initialized to ∞ and due to the update equations in Algorithm 2, the estimation that node 1 has at T_1 of the latencies of its neighbors $i \in \mathcal{S}_1$ are upper-bounds, i.e. $L_{1d}(i) \geq L_{id}(\mathbf{B}^*)$. As discussed in the proof of the previous theorems, the minimizer $\mathbf{b}^*_1$ has $b_{1d} = 1$ and $b_{1j} = 0$ for all $j \neq d$. At time T_1, since the term involving d in the update equation of Algorithm 2 has its minimum value, the vector $\mathbf{m}^*_1 = \mathbf{b}^*_1$ identifies the same minimum latency $L_{1d}(1) = L_{1d}(\mathbf{B}^*)$. Hence, immediately after time T_1 we are guaranteed that $\mathbf{b}_1 = \mathbf{b}^*_1$.

At T_k, node k is aware of the minimum expected latencies $L_{sd}(k) = L_{sd}(\mathbf{B}^*)$ for the nodes in the set $\mathcal{V}_k = \{d, 1, ..., k\} \cap \mathcal{S}_k$. All other expected latencies are upper bounds, i.e. $L_{jd}(k) \geq L_{jd}(\mathbf{B}^*)$ for $j \notin \mathcal{V}_k$. The solution $\mathbf{b}^*_k$ takes value 1 only for nodes in $\mathcal{V}_k$. The minimizer $\mathbf{m}^*_k$ at time T_k is thus equal to $\mathbf{b}^*_k$ and achieves $L_{kd}(k) = L_{kd}(\mathbf{B}^*)$. Therefore, imediately after T_k we will have $\mathbf{b}_k = \mathbf{b}^*_k$.

This argument applies until just after T_N, at which point we have $\mathbf{B} = \mathbf{B}^*$. Since T_N is finite with probability 1, the statement of the theorem follows. ■

Regarding the computational complexity of finding the minimum expected latency with the greedy decentralized approach of Algorithm 2, we should note that the optimizations in lines 8 and 9 of this algorithm are linear-fractional programs and can thus be solved very quickly using variants from linear programming.

C. Decentralized Approach with No A-Priori Knowledge

In part III-B, we assumed that as soon as a node meets another node, it has a perfect knowledge of its meeting rate with that node. In practice, a node will need to estimate its meeting rates with the neighbours and periodically revise the estimation as meetings occur (or fail to occur).

Consider an arbitrary pair of nodes that meet each other with rate λ. We denote the i^{th} intermeeting time which is the time between i^{th} and $i + 1^{\text{th}}$ meetings by $x_i \geq 0$. For this specific pair of nodes, x_i is an independent sample of an exponentially distributed random variable with parameter λ. Using the maximum likelihood approach we can estimate the parameter λ after n samples. The likelihood function $\mathcal{L}(\lambda | x_1, .., x_n) = \lambda^n e^{-\lambda \sum_{i=1}^n x_i}$ is maximized by $\widehat{\lambda} = \frac{n}{\sum_{i=1}^n x_i}$ Hence, under the exponential model, a node only needs to remember the last time it met its neighbour and the number of times it has met that neighbour. With these two pieces of information, it can update its estimation of the meeting rate ($\widehat{\lambda^n}$) from the previously estimated value ($\widehat{\lambda^{n-1}}$) using the following equation.

$$\widehat{\lambda}^n = \frac{n\widehat{\lambda}^{n-1}}{n-1+\widehat{\lambda}^{n-1}x_n} \quad (8)$$

In the next section we investigate how this estimation procedure affects the convergence of Algorithm 2.

IV. Simulation Results

We now investigate the efficiency of our proposed approach in modeling and solving the routing problem in different opportunistic network scenarios. We use three different networks to model the contacts between $N = 41$ mobile nodes.

In the first network (*Net I*), we construct a contact graph using an evolving undirected network model based on the preferential attachment mechanism. We start with a small fully connected graph of $m_0 = 5$ vertices and add vertices to it one by one until the graph consists of $N = 41$ nodes. At each step, the new vertex is connected to $m = 5$ previously existing vertices. The probability that the new vertex is connected to vertex i is $\frac{k_i}{\sum_j k_j}$ where k_i is the degree of i up to this stage. After building the contact graph, we assign a parameter λ_{ij} to each pair of nodes i and j which are connected in the contact graph and assume that they meet with exponentially distributed intermeeting times with parameter λ_{ij}. We choose the parameters λ_{ij} from a uniform distribution $U[0, 7.5\times10^{-5}]$ per second; this leads to the same average meeting rate between nodes as the empirically-derived networks discussed below.

The characteristics of the second and third networks (*Net II* and *Net III*) are derived from the Infocom05 data set [20]. This data set is based on an experiment conducted during the IEEE Infocom 2005 conference in Miami where 41 Bluetooth enabled devices (Intel iMote) were carried by attendees for 3 to 4 days. The start and end times of each contact between participants were recorded. The average time between node contacts in the Infocom05 data set is 1.3×10^4 seconds (3.6 hours). This is relatively high because of the night intervals.

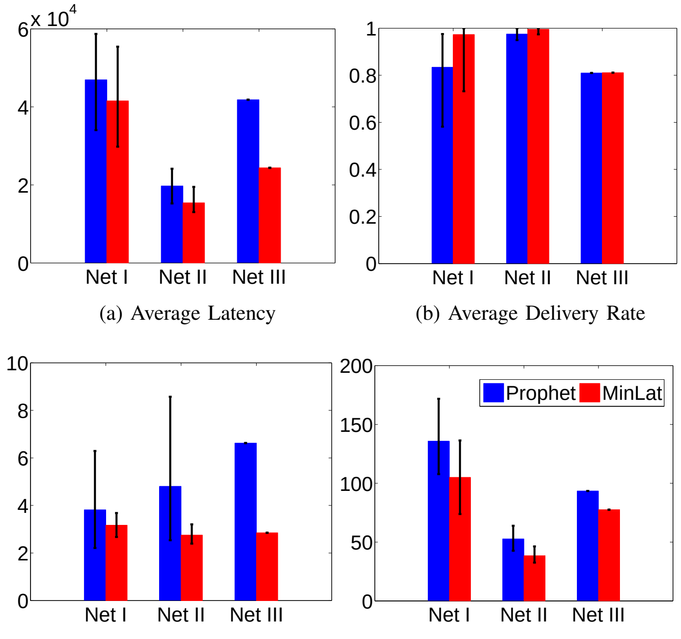
In our processing, we only consider the contacts in which both devices recognized each other so that an acknowledged message could be transfered between them. We are interested in the behaviour of the algorithms in relatively sparse networks, so we limit the number of neighbours of each node: node i is only connected to node j in the contact graph if the meeting rate λ_{ij} is among the largest $K = 10$ meeting rates of either node i or node j.

In *Net II*, we set λ_{ij} to be equal to the inverse of the average intermeeting time between nodes i and j in the sparsified Infocom05 dataset. In our simulations, the intermeeting times between nodes i and j are then chosen from an exponential distribution with parameter λ_{ij}. In *Net III*, we use the actual meeting times recorded in the Infocom05 dataset. In contrast to our model, which assumes exponential distributions, the recorded intermeeting times can be reasonably well approximated by a power-law distribution [21]. Table I summarizes the properties of the test networks.

TABLE I: Test Network Properties

Network	Contact Graph	Parameters	Intermeeting Times	Simulation Length (s)
I	Preferential Attachment	$m_0 = 5$ $m = 5$	Exponential λ: uniform	$100,000$
II	Sparsified Infocom	$K = 10$	Exponential λ: data-set	$100,000$
III	Sparsified Infocom	$K = 10$	Data-set times	$247,031$

We call our proposed decentralized greedy latency minimization algorithm (Algorithm 2) MinLat, and compare it with the history-based algorithm PRoPHET. The parameters of PRoPHET are set to those suggested in [7], i.e., $P_{init} = 0.75$, $\beta = 0.25$, $\gamma = 0.98$, and time step=1. We randomly choose one of the network nodes as the destination and send $M = 10000$ messages from all other nodes of the network to this destination. We conduct 100 simulation runs for networks *Net I* and *Net II*; the destination node, contact graph, and meeting rates remain fixed for all runs, but the intermeeting times differ. We evaluate the average latency, delivery rate, and hop count of these M messages as well as the average buffer occupancy of the network nodes.



(a) Average Latency

(b) Average Delivery Rate

(c) Average Hop Count

(d) Average Buffer Occupancy

Fig. 1: Comparison Metrics in Test Networks with $N = 41$

Figure 1 shows the four comparison metrics in the three test networks using PRoPHET and MinLat algorithms. The error bars in the graphs represent the 95% confidence intervals. The results indicate that MinLat achieves a smaller average latency than PRoPHET for the test networks. For *Net I* and *Net II*, where the assumption of exponentially distributed intermeeting times holds, this is expected from Theorem 3. But we observe that the average latency reduction holds also

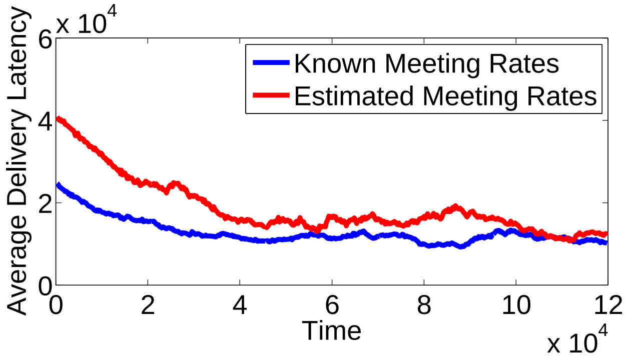


Fig. 2: Evolution of average delivery latency with time (both in seconds).

for *Net III*, where a power-law distribution provides a better fit to intermeeting times. The better performance of MinLat over PRoPHET in terms of average latency is due to our primary goal in developing Algorithm 2 which is minimizing the sum of expected latencies of all nodes to the destination node. Also, reductions in hop count and buffer occupancy are side effects of this minimization.The delivery rates of the algorithms are less than 1 because the simulation is terminated before all generated messages are successfully delivered

We conducted further simulations with a different message generation scenario on *Net II* to explore how the incorporation of meeting rate estimations in (8) affects the performance of MinLat. Figure 2 shows the average delivery latency as a function of time. Messages are generated every t seconds at randomly chosen source nodes where t is uniformly distributed in $[0, 1]$. Each point on the curves represents the average latency of the most recent 1000 packets. As we see in Figure 2, the average time it takes for a message to be delivered decreases as time goes by and the forwarding rules discovered by MinLat get closer to the optimum forwarding rules. This decreasing trend is observed in both known and estimated meeting rates scenarios, but the convergence to the optimum point is slower when nodes need to estimate the meeting rates.

V. Conclusion

In this paper, we have proposed an analytical model to develop a forwarding algorithm which leads to the minimum expected latency in opportunistic data transfer among mobile devices. We proved that under relatively strong assumptions, our proposed forwarding algorithm MinLat can achieve the minimum of sum of expected latencies from all the nodes of the network to a particular destination. Simulation results showed that the performance is improved compared to PRoPHET, the most similar existing approach. We also made our model more realistic by assuming that nodes do not know their meeting rates with other nodes and suggested an approach to estimate the meeting rates at the same time that the message forwarding is performed. Simulation results showed that our proposed method takes longer to converge but eventually results in almost the same average message delivery latency as when the meeting rates are known. This work ignores many practical challenges in opportunistic networks, but provides a useful foundation for the design of a principled and practical forwarding algorithm.

References

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet,” *ACM Sigplan Notices*, vol. 37, no. 10, pp. 96–107, 2002.

[2] T. Small and Z. J. Haas, “The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way),” in *Proc. ACM Int. Symp. Mob. Adhoc Net. and Comput. (MobiHoc)*, Annapolis, MD, USA, Jun. 2003.

[3] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: Rethinking connectivity in developing nations,” *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[4] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation,” *IEEE Trans. Mob. Comput.*, vol. 11, no. 5, pp. 821–834, 2012.

[5] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Tech. Rep. CS-200006, Duke University, Tech. Rep., 2000.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. ACM SIGCOMM Workshop on Delay-tolerant networking*, Philadelphia, PA, USA, Aug. 2005.

[7] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE Mob. Comput. and Comm. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks.” in *Proc. IEEE Infocom*, Barcelona, Spain, Apr. 2006.

[9] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, “Practical routing in delay-tolerant networks,” *IEEE Trans. Mob. Comput.*, vol. 6, no. 8, pp. 943–959, 2007.

[10] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proc. ACM Int. Symp. Mob. Adhoc Net. and Comput. (MobiHoc)*, Montreal, QC, Canada, Sept. 2007.

[11] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks,” *IEEE Trans. Mob. Comput.*, vol. 10, no. 11, pp. 1576–1589, 2011.

[12] D. A. Sharma and M. Coates, “Contact graph based routing in opportunistic networks,” in *IEEE Global Conf. Sig. and Info. Proc. (GlobalSIP)*, Austin, TX, USA, Dec. 2013.

[13] H. Chen and W. Lou, “GAR: Group aware cooperative routing protocol for resource-constraint opportunistic networks,” *Computer Comm.*, vol. 48, pp. 20–29, Jul. 2014.

[14] Q. Li, W. Gao, S. Zhu, and G. Cao, “To lie or to comply: Defending against flood attacks in disruption tolerant networks,” *IEEE Trans. Dependable and Secure Comput.*, vol. 10, no. 3, pp. 168–182, May/June 2013.

[15] Y. Li, G. Su, D. O. Wu, D. Jin, L. Su, and L. Zeng, “The impact of node selfishness on multicasting in delay tolerant networks,” *IEEE Trans. Vehicular Tech.*, vol. 60, no. 5, pp. 2224–2238, 2011.

[16] P. Sermpezis and T. Spyropoulos, “Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks,” *Computer Comm.*, vol. 48, pp. 71–83, 2014.

[17] S. Grasic, E. Davies, A. Lindgren, and A. Doria, “The evolution of a DTN routing protocol-PRoPHETv2,” in *Proc. ACM Workshop on Challenged Networks*, Las Vegas, NV, USA, Sept. 2011.

[18] V. Conan, J. Leguay, and T. Friedman, “Fixed point opportunistic routing in delay tolerant networks,” *IEEE Journal on Sel. Areas in Comm.*, vol. 26, no. 5, pp. 773–782, 2008.

[19] M. Grossglauser and D. Tse, “Mobility increases the capacity of ad-hoc wireless networks,” *Trans. on Net.*, vol. 10, no. 4, pp. 477–486, 2002.

[20] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD data set cambridge/haggle (v. 2006-01-31),” Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006.

[21] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Trans. Mob. Comput.*, vol. 6, no. 6, pp. 606–620, Jun. 2007.