On Multicopy Opportunistic Forwarding Protocols in Nondeterministic Delay Tolerant Networks

Cong Liu[†] Jie Wu[‡]
[†]Sun Yat-sen University [‡]Temple University
gzcong@gmail.com jiewu@temple.edu

***Abstract*—*Delay Tolerant Networks* (DTNs) are characterized by nondeterministic mobility and connectivity. Message routing in DTNs usually employs a multi-copy forwarding scheme. To avoid the cost associated with flooding, much effort has been focused on opportunistic forwarding, which aims to reduce the cost of forwarding while retaining high routing performance by forwarding messages only to nodes that have high delivery probabilities. This paper presents two multicopy forwarding protocols, called *optimal opportunistic forwarding* (OOF) and OOF-, which maximize the expected delivery rate and minimize the expected delay, respectively, while requiring that the number of forwardings per message does not exceed a certain threshold. Our contributions in this paper are summarized as follows: We apply the optimal stopping rule in the multi-copy opportunistic forwarding protocol. Specifically, we propose two optimal opportunistic forwarding metrics to maximize delivery probability and minimize delay, respectively, with a fixed number of copies and within a given time-to-live. We implement and evaluate OOF and OOF- as well as several other representative forwarding protocols, i.e., Epidemic, Spray-and-wait, MaxProp*, and Delegation. We perform trace-driven simulations using both real and synthetic traces. Simulation results show that, in the traces where nodes have regular inter-meeting times, the delivery rates of OOF and OOF- can be 30% greater than the compared routing protocols.**

Keywords: Delay Tolerant Networks, Optimal Stopping Rule, Routing, Simulation.

I. Introduction

A *Delay Tolerant Network* (DTN) [1] is a sparse mobile network, where a contemporary source-destination path may not exist between a pair of source-destination nodes, and messages are routed in a store-carry-forward routing paradigm. Due to uncertainty in node mobility, DTN routing algorithms usually spawn and keep multiple copies of the same message in different nodes. The message is delivered if one of these nodes encounters the destination.

The most expensive routing protocol, Epidemic [2], forwards copies of a message to any possible node and guarantees a maximized delivery rate. Effectively flooding the network with every message, Epidemic is impractical in large networks. Recently, much effort has been focused on opportunistic forwarding, which tries to reduce the number of copies of each message while retaining a high routing performance, i.e., a high delivery rate and a low delay. Since only a small fraction of the nodes can obtain the copies of a message to save energy, it is desired that these copies are forwarded by the nodes which have higher delivery probabilities than the other nodes.

In this paper, we improve the message forwarding algorithm in [3], in the calculation of its delivery probability, and name it the *optimal opportunistic forwarding* (OOF) protocol. In the OOF protocol, the optimal forwarding metrics and the optimal forwarding rules are defined. By optimality, we mean that, with a limited number of forwardings (or number of copies) per message, OOF maximizes the expected delivery probability based on a particular knowledge about the network, i.e., the pair-wise inter-meeting times between the nodes.

We further propose a simplified forwarding protocol, named OOF-, which minimizes the expected delay instead of maximizing the delivery rate, as in OOF. OOF- has a significantly smaller computation and storage requirement than OOF, while its routing performance approximates OOF.

In our network model, we assume that the long-term mean inter-meeting times between nodes can be estimated from the contact history of the nodes. Ideally, each node has complete routing information of the mean inter-meeting times between all of the pairs of nodes in the network. We will relax the second assumption and allow the protocol to work with incomplete routing information. Our optimal forwarding metric differs from the existing ones in two important ways:

- It is a *comprehensive* metric which reflects not only the direct (1-hop) expected delivery probability of a message copy or the expected delivery probability of it along a single multi-hop path, but also the joint expected delivery probability of multiple copies of a message being forwarded along multiple paths.
- It is also a *dynamic* metric, which reflects the state of the message. For example, in a hop-count-limited forwarding scheme, our optimal forwarding metric is a function of two important states of the message copy: remaining hop-count and residual time-to-live (Section III).

The basic idea is to model each forwarding as an *optimal stopping rule problem*, in which a forwarding time is deliberately chosen in order to maximize the joint expected delivery probability (or minimize the joint expected delay) of the copies in the forwarding node and receiving node of the copy at each forwarding.

We perform simulations using four Cambridge Haggle traces [4], the National Singapore University (NUS) student trace [5], and the UMassDieselNet [6] trace. We evaluate the routing performance of OOF and OOF- against several representative DTN routing protocols: Epidemic [2], Spray-and-wait [7], MaxProp [8], and Delegation [9], in terms of delivery rate and forwarding number (cost). Simulation results

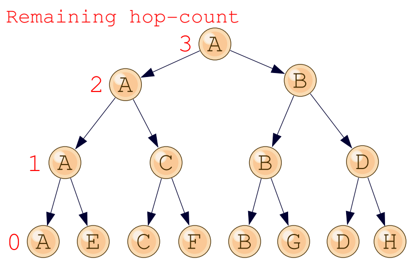


Fig. 1. An illustration of hop-count-limited opportunistic forwarding.

in the Cambridge Haggle traces show that, when nodes have regular inter-meeting times, the delivery rates of OOF and OOF- can be 30% greater than the compared routing protocols. In the simulation results using the UMassDieselNet trace, where regularity in mobility is more observable, OOF and OOF- show even better improvement.

II. Preliminaries and Overview

Our design of the OOF and OOF- protocols is developed on a hop-count-limited opportunistic forwarding scheme, which we will show next. Then, we will briefly present OOF and OOF- together with some representative forwarding protocols.

A. Hop-count-limited Forwarding

In a hop-count-limited opportunistic forwarding protocol, each message maintains a value, called *remaining hop-count*, which indicates the maximum amount of hops that the message can be forwarded over. When a message with a remaining hop-count K is forwarded from one node to another, the remaining hop-count of both copies in the two nodes becomes $K-1$. When $K=0$, the message cannot be forwarded to any node except the destination. That is, if the initial hop-count of a message is H, then the maximum number of forwardings for the message is 2^H, including the one delivered to the destination. In Figure 1, a message is created with $H=3$ in node A, and a tree of message forwarding history is shown.

An advantage of this forwarding scheme is that it has a constant per message forwarding cost (assuming that the forwarding cost is the major cost in the whole communication process), which is necessary to achieve ultimate scalability: with a constant per node message rate, the per node forwarding overhead is kept constant as the network size increases.

B. Motivation and Overview

In most opportunistic forwarding protocols, each node is associated with a forwarding metric for each destination, which signifies the quality of the node as a forwarder. Existing forwarding metrics are usually: (1) direct (1-hop) metrics between the nodes and the destination, such as encounter frequency [10] and the time elapsed since the last encounter [11], [12], [13], [14], or (2) the expected forwarding metric along the expected forwarding path, such as expected cost [8] and expected delay [15]. When node i meets node j, node i forwards a message to node j depending on whether or not the direct forwarding quality of i is better than j.

We found two drawbacks in such strategies. The first drawback is that a forwarding decision based on comparing the direct or multi-hop forwarding qualities of nodes i and j cannot guarantee good forwarding for the following reasons. (1) The forwarding quality of j being better than i does not necessarily mean that j is a good forwarder. (2) Even though the quality of j is high, i might encounter better nodes in the near future. (3) Similarly, even though the quality of j is lower than i, j might still be the best forwarder that i could encounter in the future.

The second drawback is that the forwarding quality of a node is regarded as a constant. However, the forwarding quality of a node may change significantly at different stages. For example, in hop-count-limited forwarding, two important states of the copy are: *remaining hop-count* and *residual time-to-live*. Remaining hop-count is an important factor: a node can be a bad 1-hop forwarder for having a large mean inter-meeting time with the destination, but it can still be an excellent 2-hop forwarder if it has a node that it frequently contacts, which is also a node that frequently contacts the destination. On the other hand, residual time-to-live is important because it affects a node's direct delivery probability, as well as its chance of contacting high quality intermediate nodes.

To rectify these drawbacks, we use a comprehensive forwarding metric, which reflects: (1) not the relative forwarding quality between two nodes (node i and the next node j that would hold custody of a new copy of the message), but the relative forwarding qualities among all possible next nodes j, and (2) not the quality (1-hop or multi-hop delivery probability or delay) of a particular message copy, but the joint delivery probability or delay, of all copies when multiple copies of the message can be forwarded along multiple paths.

We define a delivery probability P_{i,d,K,T_r} for each copy in i and each destination d. This metric is comprehensive because it represents the joint probability of all descendant copies, and it is also dynamic since it is a function of the remaining hop-count K and residual time-to-live T_r. With P_{i,d,K,T_r}, our optimal forwarding rule is presented as follows: we logically regard a forwarding from a node i to another node j as replacing a message copy with two new copies in the two nodes, respectively. Whether i should forward the copy to j depends on whether replacing the copy in i with two logically new copies increases the joint delivery probability: the copy is forwarded only if the joint probability of $P_{i,d,K-1,T_r-1}$ and $P_{j,d,K-1,T_r-1}$ (in case of forwarding) is greater than the probability P_{i,d,K,T_r-1} (in case of no forwarding). Details on our optimal forwarding rule will be discussed in Section III.

It is challenging to calculate the accurate delivery probability P_{i,d,K,T_r} for each K and T_r (Section III-E). We first assume that all nodes have full routing information, which is the mean inter-meeting times between all of the pairs of nodes. We calculate P_{i,d,K,T_r} using *backward induction*, a solution to the optimal stopping rule problem we modeled.

To reduce the computation and storage requirement, we propose another forwarding metric, called *expected delay* $D_{i,d,K}$, which is also comprehensive but reduces the time dimension in the forwarding metric of OOF. The corresponding forwarding protocol is called OOF-. Although without a sense of timing, OOF- approximates OOF in most situations in terms of delivery rate.

C. Protocols in Comparison

We compare OOF and OOF- against several opportunistic forwarding protocols. While OOF and OOF- have well-defined utilities to maximize in each forwarding (the joint expected delivery probability or the joint expected delay of all copies of each message), other algorithms use either heuristic forwarding rules or blind forwarding.

Epidemic [2]. A node sends a copy of the message to every node it encounters that does not have a copy already until its copy of the message times out.

Spray-and-wait [7]. This protocol differs from Epidemic in that it controls the number of copies of each message in the network to be smaller than L.

MaxProp*. We use a variation MaxProp*, which differs from MaxProp [8] in that (1) it incorporates the hop-count-limited forwarding protocol to control forwarding overhead in order to make fair comparison, and (2) it assumes that each node can carry an infinite number of messages. Note that the first modification will effect the performance negatively.

Delegation [9]. Delegation forwarding may use a wide range of forwarding metrics (qualities). We use the mean inter-meeting time $I_{k,d}$ of node k with destination d as the forwarding quality of a node k, and node j has a higher forwarding quality than node i if $I_{j,d} < I_{i,d}$.

D. The Optimal Stopping Rule Problem

Let us briefly review the optimal stopping rule problem [16] with an example. In a stopping rule problem, we may observe a sequence $X_1, X_2, \ldots$ for as long as we wish, where $X_1, X_2, \ldots$ are random variables whose joint distribution is assumed to be known. For each stage $t = 1, 2, \ldots$ after observing $X_1, X_2, \ldots, X_t$, we may stop and receive the known reward y_t, or we may continue and observe X_{t+1}. In the latter case, the bit X_t on day t will not be valid anymore on day $t+1$. The optimal stopping rule is to stop at some stage t to maximize the expected reward.

A stopping rule problem has a finite horizon if there is a known upper bound T on the number of stages at which one may stop. If stopping is required after observing $X_1, \ldots, X_T$, we say the problem has a horizon of T. In principle, such problems may be solved by the method of *backward induction*. Since we must stop at stage T, we first find the optimal rule at stage $T-1$. Then, knowing the optimal reward at stage $T-1$, we find the optimal rule at stage $T-2$, and so on, back to the initial stage (stage 0). Let $V_t^{(T)}$ $(1 \leq t \leq T)$ represent the maximum expected reward one can obtain, starting from stage t. We define $V_T^{(T)} = y_T$ and then inductively for $t = T-1$, go backwards to $t = 0$:

$$V_t^{(T)} = E(max\left\{y_t, V_{t+1}^{(T)}\right\}).$$

The meaning of the above equation is that, at stage t, we compare the reward for stopping, namely y_t, with the best reward $V_{t+1}^{(T)}$ that we expect to be able to get by continuing and using the optimal rule for stages $t+1$ through T. The optimal reward is therefore the maximum of these two quantities, and it is optimal to stop at the earliest t when $y_t \geq V_{t+1}^{(T)}$.

III. Optimal Opportunistic Forwarding (OOF)

In this section, we apply the optimal stopping rule to derive the delivery probability of each message and the optimal forwarding rule in OOF.

A. Assumptions

Each message has a source and a destination and is given a time-to-live at its creation time. A message is deleted only when it expires. Different copies of the same message are forwarded independently without any knowledge of the status of the other copies.

Like other opportunistic forwarding protocols that make use of historical contact information, it is desired that node mobility exhibits long-term regularities, such that some pairs of nodes consistently meet more frequently than other pairs over time. Networks falling into this category include most natural or human-related mobile networks.

Firstly, we assume that each node knows the full routing information: the mean inter-meeting times $I_{i,j}$ between all pairs of nodes $\{i, j\}$. This can be relaxed in practice.

In the calculation of the delivery probability in OOF and the expected delay in OOF-, we assume that the delivery probabilities (or the expected delays) are independent and that the inter-meeting times are exponentially distributed. However, we will not restrict the proposed protocols to these assumptions in our simulations.

B. Discrete Residual Time-to-live

To model our optimal forwarding problem as an optimal stopping rule problem, we need to use a discrete residual time-to-live T_r with a certain fixed time-slot size U. Let T_{max} be the maximum possible discrete time-to-live of any message, and the range of T_r is between 0 and T_{max}. Our forwarding metric (delivery probability) is a function of T_r, and it is calculated using an inductive method. The amount of computation for our forwarding metric is inversely proportional to the length of U, but its accuracy normally decreases as U increases. In the rest of the paper, we use T_r to denote a residual time-to-live or a particular time-slot at T_r interchangeably without causing confusion.

In each time-slot T_r, a node can either meet or not meet with another node. A node has the probability to meet several other nodes during the same time-slot, and we simply assume that all meetings start at the beginning of some time-slot. This assumption holds if we (1) truncate all meeting durations so that their starting times are aligned in the beginning of their respective time-slots; (2) prolong all meeting durations to the end of their respective time-slots; (3) divide long meeting durations into individual time-slots. The meeting probability of two nodes in any time-slot of length U is estimated under the assumption of exponential inter-meeting time [7], [12] by:

$$M_{i,j} = 1 - \exp(-\frac{U}{I_{i,j}}).$$

TABLE I
FORWARDING OPTIONS.

T_r	P_{i,d,K,T_r}	
$T_r - 1$	Not Forward	Forward (becomes $K-1$)
	P_{i,d,K,T_r-1}	$P_{i,d,K-1,T_r-1}$, $P_{j,d,K-1,T_r-1}$

The calculation of $M_{i,j}$ may not rely on the assumption of exponential inter-meeting times. Using more realistic estimations, such as [17], for a network in question is expected to result in a better routing performance.

C. 1-hop Delivery Probability

The 1-hop delivery probability of a message copy is the probability that the hosting node meets the destination directly within its time-to-live. It is only a function of residual time-to-live. We estimate the 1-hop delivery probability, assuming again an exponential inter-meeting time, by:

$$P_{i,d,0,T_r} = 1 - \exp(-\frac{T_r \times U}{I_{i,d}}), \tag{1}$$

where 0 means that the remaining hop-count is 0, $T_r \times U$ is the residual time-to-live of the message (T_r is the number slots of residual time-to-live before the message expires, with the length of the slots being U) and $I_{i,d}$ is the mean inter-meeting time between node i and the destination d. Again, the calculation of $P_{i,d,0,T_r}$ may use an other method, such as [17], without the assumption of exponential inter-meeting times.

D. K-hop Delivery Probability and Forwarding Rule

Our optimal delivery probability and optimal forwarding rule are inter-dependent: (1) the optimal forwarding rule uses the optimal delivery probability in making forwarding decisions, and (2) the optimal delivery probability is calculated assuming that the optimal forwarding rule is used. The optimal delivery probability of a copy in node i, heading for destination d, with a remaining hop-count K ($K > 0$) and with a residual time-to-live T_r, is denoted by P_{i,d,K,T_r}.

We will present our optimal forwarding rule first. When a copy, whose remaining hop-count is K, is in node i, and node i meets node j at time-slot T_r, the decision on whether to forward depends on whether replacing the copy in i with two new copies in i and j, respectively, will increase the joint delivery probability. As shown in Table I, if the message is not forwarded in time-slot T_r, then in the next time-slot, we have the copy with the same remaining hop-count K in node i, and the copy's delivery probability becomes P_{i,d,K,T_r-1}. On the other hand, if the message is forwarded in time-slot T_r, then in the next time-slot, we have two new copies with remaining hop-count $K-1$ in both i and j, whose delivery probabilities are $P_{i,d,K-1,T_r-1}$ and $P_{j,d,K-1,T_r-1}$, respectively. To maximize the delivery probability, we use the optimal forwarding rule which forwards the message only if the joint delivery probability of the two copies (in the case of forwarding) is greater than the single copy (in the case of no forwarding), or:

$$1 - (1 - P_{i,d,K-1,T_r-1}) \times (1 - P_{j,d,K-1,T_r-1}) > P_{i,d,K,T_r-1}.$$

For simplicity, in the above discussion, we assumed that in a sparse DTN, two consecutive forwardings of the same message (i.e., from i to j and then from j to another node) cannot occur in the same time-slot. Also, we only considered uni-cast forwarding. When connected with several nodes at the same time-slot, we forward the copy to node j, which has the largest $P_{j,d,K-1,T_r-1}$.

The optimal delivery probability P_{i,d,K,T_r} depends on the optimal forwarding rule and the meeting probabilities $M_{i,j}$ of i with each node j in time-slot T_r whose delivery probability $P_{j,d,K-1,T_r-1}$ satisfies the forwarding criteria. We will calculate P_{i,d,K,T_r} in the next subsection by modeling a forwarding as an optimal stopping rule problem.

Note that our algorithm allows messages with different initial time-to-lives and hop-counts. T_{max} is the maximum initial time-to-live of all messages, and H is the maximum initial hop-count. For a message, whether it is a newly created, original copy or it is a received copy, our algorithm simply looks up the table for the P_{i,d,K,T_r}s needed. A message is deleted only when it expires, i.e., when $T_r < 0$.

E. OOF as an Optimal Stopping Rule Problem

We model each forwarding as an optimal stopping rule problem as follows: we consider only the next forwarding of a message in node i with remaining hop-count K. At the time of forwarding, the copy is logically regarded as being replaced by two new copies, both of which have a $K-1$ remaining hop-count. A candidate copy receiver j comes in at each time-slot T_r with probability $M_{i,j}$, where T_r also denotes the residual time-to-live of the message. Upon meeting with j, i can either forward the copy to j, or not. Since we assume no consecutive forwardings of the same message (i.e., from i to j and then from j to another node) in the same time-slot, we can calculate the resulting joint delivery probability based solely on the delivery probabilities in the next time-slot, $T_r - 1$.

A node may meet several other nodes in the same time-slot. Forwarding the copy to the node with the highest delivery probability is the optimal strategy to maximize the expected delivery probability. Given the meeting probability $M_{i,j}, M_{i,k}, \ldots$ of node i and nodes $j, k, \ldots$ that i will probably meet with in time-slot T_r and the delivery probabilities $P_{j,d,K-1,T_r-1}, P_{k,d,K-1,T_r-1}, \ldots$ of nodes $j, k, \ldots$, sorted in a decreasing order, the maximum probability that the copy will be forwarded to one of nodes $j, k, \ldots$ in time-slot T_r and then be delivered, is:
$P(delivered|forwarded\ at\ T_r) \times P(forwarded\ at\ T_r) = P(delivered$ **and** $forwarded\ at\ T_r) = M_{i,j} \times P_{i,d,K-1,T_r-1} + (1 - M_{i,j}) \times M_{i,k} \times P_{k,d,K-1,T_r-1} + \ldots$

The expected optimal delivery probability P_{i,d,K,T_r} equals the sum of: (1) the probability that the current node encounters the destination at T_r; (2) the probability that the copy will be forwarded to some node other than the destination at time-slot T_r, and then be delivered, as shown above; (3) $P_{i,d,K,T_r-1} \times M'_{i,N}$, where P_{i,d,K,T_r-1} is the delivery probability if the message is not forwarded at time-slot T_r, and $M'_{i,N} = 1 - M_{i,d} - (1 - M_{i,d}) \times M_{i,j} - (1 - M_{i,d}) \times (1 - M_{i,j}) \times M_{i,k} - \ldots$ is the probability that the message is not forwarded at a time-slot.

Algorithm 1 Calculation of P_{i,d,K,T_r}

1: $P_{i,d,K,T_r} := M_{i,d}$
2: $M'_{i,N} := 1 - M_{i,d}$
3: **for each** (node j, $j \neq i$ **and** $j \neq d$) {
4: $\quad P_{i,j} = 1 - (1 - P_{i,d,K-1,T_r-1}) \times (1 - P_{j,d,K-1,T_r-1})$
5: }
6: $Q :=$ a priority queue of j in decreasing order of $P_{i,j}$
7: **while** ($j :=$dequeue(Q) **and** $P_{i,j} > P_{i,d,K,T_r-1}$) {
8: $\quad P_{i,d,K,T_r} := P_{i,d,K,T_r} + M'_{i,N} \times M_{i,j} \times P_{i,j}$
9: $\quad M'_{i,N} := M'_{i,N} - M'_{i,N} \times M_{i,j}$
10: }
11: $P_{i,d,K,T_r} := P_{i,d,K,T_r} + M'_{i,N} \times P_{i,d,K,T_r-1}$

Algorithm 1 shows the calculation of a single P_{i,d,K,T_r} using the backward induction method. In line 7, the while loop stops when queue Q is empty. Using this algorithm, we calculate P_{i,d,K,T_r} for all K from 1 to H (when $K = 0$, $P_{i,d,0,T_r}$ is calculated by Equation 1) and then for all T_r from 1 to T_{max} (when $T_r = 0$, $P_{i,d,K,0} = P_{i,d,0,0}$).

IV. Removing the time dimension (OOF-)

In OOF, each node needs to calculate a four dimensional table for the delivery probabilities P_{i,d,K,T_r}. Clearly, the size of the table is proportional to the number (T_{max}) of total time-slots, which depends on the maximum time-to-live of the messages and the length of each time-slot. For example, In our simulations using the Cambridge Haggle traces, we set $T_{max} = 100$. This section presents another protocol, called OOF-, which provides a trade-off between a smaller computation and storage requirement ($1/T_{max}$ of that of OOF) and a slightly degraded performance in particular situations.

In OOF-, we define a new forwarding metric, *expected delay* ($D_{i,d,K}$), which is only parameterized by the remaining hop-count. An expected delay $D_{i,d,K}$ denotes the expected time it takes to deliver a message with a remaining hop-count K. Expected delay is also a comprehensive forwarding metric, which considers the joint expected delay of all possible descendant forwarders in the forwarding tree. Assuming an exponentially distributed (memoryless) inter-contact time, the message forwarding condition from node i to node j is: $\frac{1}{D_{i,d,K}} > \frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}$.

A. Deriving $D_{i,d,K}$

The 1-hop (directly) expected delivery delay $D_{i,d,0}$ of a message in node i is simply $I_{i,d}/2$, where $I_{i,j}$ is the mean inter-meeting time between nodes i and j. We assume that inter-meeting times are exponentially distributed: if two copies have expected delays D_1 and D_2, respectively, then their joint expected delay equals $\frac{1}{\frac{1}{D_1}+\frac{1}{D_2}}$.

Suppose that under the optimal forwarding strategy, N ($d \in N$) is a set of nodes that, when encountering any $j \in N$, node i will forward the message to j. Let (1) $W_{i,N}$ be the expected waiting time for i to encounter the first node in N, i.e., $W_{i,N}$ is the expected waiting time for i to forward the message; let (2) $p_{i,j}$ be the probability that j is the first node to encounter i among all nodes in N, and $\sum_{j \in N} p_{i,j} = 1$. Assuming the exponential encountering time, we have $W_{i,N} = \frac{1}{\sum_{j \in N} \frac{2}{I_{i,j}}}$, and $p_{i,j} = \frac{p_{i,j}}{\sum_{j \in N} p_{i,j}} = \frac{\frac{1}{I_{i,j}}}{\sum_{k \in N} \frac{1}{I_{i,k}}} = \frac{2W_{i,N}}{I_{i,j}}$.

Algorithm 2 Calculation of $D_{i,d,K}$

$W_{i,N} := I_{i,d}/2$
$D_{i,d,K} := \infty$
$\sum := 0$
$Q :=$ a priority queue of j in decreasing order of $\frac{1}{D_{j,d,K-1}}$
while ($j :=$dequeue(Q) **and** $D_{i,d,K} > W_{i,N} \times (1 + \sum)$) {
$\quad D_{i,d,K} := W_{i,N} \times (1 + \sum)$
$\quad W_{i,N} := \frac{1}{\frac{1}{W_{i,N}} + \frac{2}{I_{i,j}}}$
$\quad \sum := \sum + \frac{2}{I_{i,j} \times (\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}})}$
}

The expected delay $D_{i,d,K}$ of a message in node i is the sum of: (1) $W_{i,N}$ multiplied by the probability $p_{i,d}$ that i encounters d first; (2) the joint expected delay $W_{i,N} + \frac{1}{\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}}$ of the two new copies in i and j, multiplied by the probability $p_{i,j}$ that i encounters j ($j \in N$ and $j \neq d$) first. Therefore, $D_{i,d,K}$ can be derived as follows: $D_{i,d,K} =$

$$p_{i,d} \times W_{i,N} + \sum_{j \in N \setminus \{d\}} p_{i,j} \times (W_{i,N} + \frac{1}{\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}})$$

$$= W_{i,N} + \sum_{j \in N \setminus \{d\}} p_{i,j} \times \frac{1}{\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}}$$

$$= W_{i,N} + \sum_{j \in N \setminus \{d\}} \frac{2W_{i,N}}{I_{i,j}} \times \frac{1}{\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}}$$

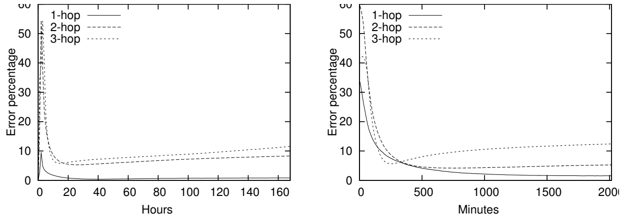
$$= W_{i,N} \times (1 + \sum_{j \in N \setminus \{d\}} \frac{2}{I_{i,j} \times (\frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}})}).$$

Algorithm 2 shows the calculation of $D_{i,d,K}$ using the above equation. The set N of candidate receiver nodes under the optimal forwarding strategy is constructed by adding each node j into N in the decreasing order of $\frac{1}{D_{j,d,K-1}}$, until $D_{i,d,K}$ reaches its minimum value.

B. Comparing $D_{i,d,K}$ to P_{i,d,K,T_r}

The expected delay in OOF- ignores the residual time-to-live T_r, which may introduce incorrect forwarding decisions. In this subsection, we evaluate the percentage of incorrect decisions that OOF- makes by comparing the decisions made by OOF and OOF-. Specifically, we compare the percentage of inconsistency between $P_{i,d,K,T_r} > 1-(1-P_{i,d,K-1,T_r}) \times (1-P_{j,d,K-1,T_r})$ and $\frac{1}{D_{i,d,K}} > \frac{1}{D_{i,d,K-1}} + \frac{1}{D_{j,d,K-1}}$ for all possible permutations of node i, node j, destination d, remaining hop-count K, and residual time-to-live T_r.

We use the inter-meeting times in the UMassDieselNet trace [6], [8] and the NUS trace [5], respectively, to calculate the routing tables for OOF and OOF-. The percentage of inconsistency in forwarding decisions is shown in Figure 2. The



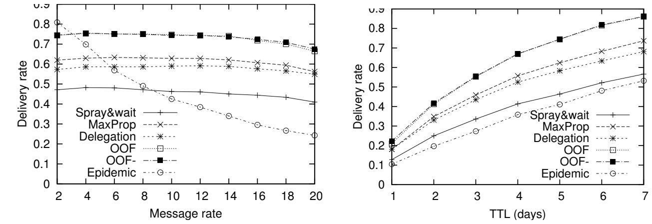
(a) UMassDieselNet trace (b) NUS trace

Fig. 2. Percentage of incorrect OOF- forwarding decisions compared to OOF.

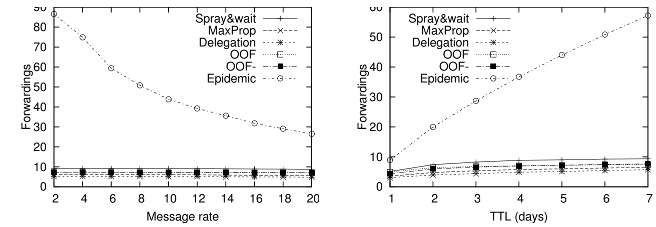
TABLE II
SIMULATION SETTINGS IN THE UMASSDIESELNET TRACE.

parameter name	default	range
number of nodes (N)	92	
tickets in spray-and-wait (L)	10	
initial hop-count (K)	3	
message time-to-live (TTL)	5 days	1∼7 days
length of time-slot (U)	5 minute	
simulation time	5 days	1∼7 days
message rate (total messages)	$10N^2$	$2\sim20N^2$

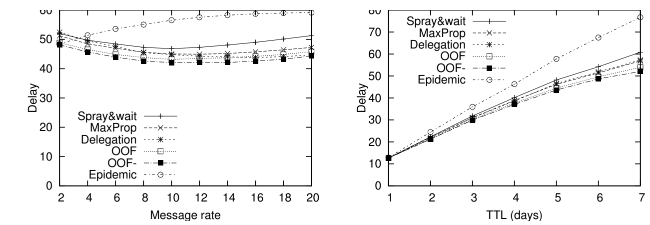
percentage of inconsistent forwarding decisions is generally below 10%, except when the residual time-to-live is less than 10 hours in (a) or less than two hours in (b), where it can be as large as 50%. The reason for this inconsistency is because, as the residual time-to-live becomes smaller, OOF is more ready to spread copies. On the other hand, OOF- is as conservative as usual for being unaware of the upcoming deadline.



(a) Delivery rate versus message rate (b) Delay versus message time-to-live



(c) Forwardings versus message rate (d) Forwardings versus message time-to-live



(e) Delay versus message rate (f) Delay versus message time-to-live

Fig. 3. Delivery rate, delay, and number of forwardings versus message rate and message time-to-live in UMassDieselNet trace.

V. EVALUATION

We evaluate the proposed protocols, OOF and OOF-, against other forwarding protocols using the UMassDieselNet trace and four Cambridge Haggle traces [4]. The routing protocols implemented to compare to OOF and OOF- were listed in Section II-C. Since all of the protocols that we implement aim to compare different delivery probability metrics, other optimizations that have orthogonal effects on the performance of these protocols are not implemented. These optimizations can be added to all of our implemented algorithms and they are expected to improve the routing performance of all of them. They may include buffer management [8], estimation of global message delivery probability [12] and social centrality of the nodes [18], the use of position information [19], as well as acknowledgment mechanism [8], [12]. Note that without an acknowledgment mechanism, even though some copies of a message may have been delivered, the other copies of the same message may still be forwarded in the network. We measure the delay and the number of forwardings (the total number of copies of a message in the network) using their averages on the delivered messages only.

A. *UMassDieselNet Trace*

In the UMassDieselNet bus system consisting of 40 buses [6], [8], the bus-to-bus contacts (the durations of which are relatively short) are logged. Our experiments are performed on traces collected over 55 days during the Spring 2006 semester, with weekends, Spring break, and holidays removed due to reduced schedules. The bus system serves approximately ten routes. There are multiple shifts serving each of these routes. Shifts are further divided into morning (AM), midday (MID), afternoon (PM), and evening (EVE) sub-shifts. Drivers choose buses at random to run the AM sub-shifts. At the end of the AM sub-shift, the bus is often handed over to another driver to operate the next sub-shift on the same route or on another route. Unfortunately, the all-bus-pairs contacts provided in the original traces show no discernible contact pattern among the nodes. We performed the data process in [20] to generate the contacts at a sub-shift level, which exhibit periodic behavior. This process translates the bus-to-bus contacts into contacts between sub-shifts.

The settings of the UMassDieselNet trace simulation are shown in Table II. We further process the traces such that all communication links between the nodes become bidirectional. In these traces, the average number of contacts per sub-shift per day is 8.5, and the contact duration is 12 seconds on average. We restrict the maximum number of messages forwarded in each contact opportunity to a maximum of 100 messages per second of contact duration.

As shown in Figure 3(a), OOF and OOF- have the same delivery rate, which is around 20% higher than those of MaxProp* and Delegation and 60% higher than that of Spary-and-wait. Epidemic, which effectively floods the network with message copies, degrades most significantly as the message

TABLE III
STATISTICS IN FOUR CAMBRIDGE HAGGLE TRACES.

Trace	Contacts	Length(D) (d.h:m.s)	Routing nodes	External nodes
Cambridge	6732	6.1:34.2	12	223
Infocom	28216	2.22:52.56	41	264
Infocom2006	227657	3.21:43.39	98	4519
Content	41587	23.19:50.18	54	11418

TABLE IV
SIMULATION SETTINGS IN THE CAMBRIDGE HAGGLE TRACES.

parameter name	default	range
tickets in spray-and-wait (L)	8	
initial hop-count (K)	3	
message time-to-live (TTL)		$\frac{1}{10}D \sim D$
length of time-slot (U)	$\frac{1}{100}D$	
number of messages	30,000	

rate increases. The effect of the message rate on other protocols with hop-count limitations is minor. Figure **??** shows that OOF and OOF- keep the same delivery rate improvement over the other protocols as the message time-to-live varies.

Firgures 3(c) and 3(d) show that all protocols, except Epidemic, have small per-message forwarding numbers.

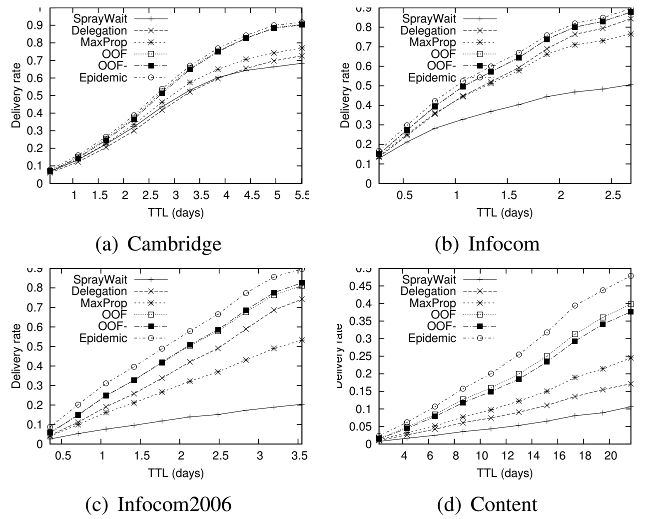
Firgures 3(e) and 3(f) show that OOF and OOF- also have the smallest delay among the compared protocols, although, the improvement is not significant.

B. Cambridge Haggle Trace.

The Cambridge Haggle trace [4] data includes a total of five traces of Bluetooth device connections by people carrying mobile devices (iMotes) for a number of days. These traces are collected by different groups of people in office environments, conference environments, and city environments, respectively. Bluetooth contacts were classified into two groups: iMotes' sightings of other iMotes are classified as *internal contacts*, while sightings of other types of Bluetooth devices are called *external contacts*. Since there is no record of contact between non-iMotes, we only use the iMotes as *routing nodes*. Other nodes, or *external nodes*, can only be assigned as destinations.

Table III shows some of the statistics in the four traces we use. In these traces, we assume infinite forwarding bandwidth and storage in each node. Simulation settings in these four traces are shown in Table IV. We did not include the Intel trace because the trace contains only 9 nodes, where all protocols can flood the network. Each simulation result is averaged over 30,000 randomly generated messages.

Delivery rates of the protocols are compared in Figures 4(a), 4(b), 4(c), and 4(d). From these results, we can see that the compared improvements of OOF and OOF-, in terms of delivery rate, are (1) more significant in the environments where regularity in mobility is more observable, such as campus (Figure 4(a)) and city environments (Figure 4(d)), and (2) less significant in the environments with relatively random mobility, such as conference (Figures 4(b) and 4(c)). In general, the results show that OOF and OOF- have comparable performances in all cases. In Figure 4(c), OOF and OOF- have a 60% of improvement over MaxProp∗, and in Figure 4(d), OOF and OOF- show a 100%+ improvement over Delegation.



(a) Cambridge (b) Infocom (c) Infocom2006 (d) Content

Fig. 4. Delivery rate versus message time-to-live in the Cambridge Haggle trace.

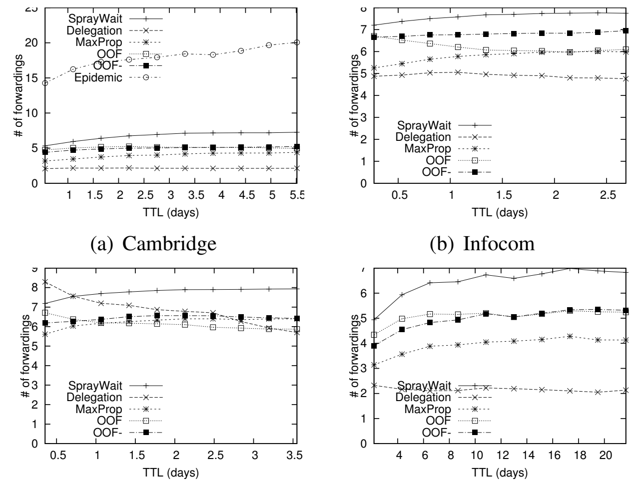
The number of forwardings of all protocols are compared in Figures 5(a), 5(b), 5(c), and 5(d). The results of Epidemic are removed from Figures5(b), 5(c), and 5(d) since they are much larger than those of the other protocols. The results in all of these figures show that Spray-and-wait, which is socially oblivious, has the largest forwarding number and the smallest delivery rate at the same time. The reason that Delegation can have a very low delivery rate is probably due to its over conservative forwarding policy, as can be found in Figure 5(d). From Figures 5(b) and 5(c), we found that OOF, compared with OOF-, can automatically become more conservative in forwarding as time-to-live increases.

VI. CONCLUSION

In this paper, we investigated several forwarding protocols, which maximize the expected delivery rate and minimize the expected delay, while satisfying the constraint on the number of forwardings per message. We proposed the *optimal opportunistic forwarding* (OOF) protocol and a simplification, the OOF- protocol, which make optimal forwarding decisions by modeling each message forwarding as an optimal stopping rule problem. We implemented OOF and OOF-, as well as several other protocols, and performed trace-driven evaluations on several traces.

Evaluation results verify that, compared with other algorithms, OOF and OOF- have higher delivery rates and smaller delays under a bounded number of per message forwardings. In terms of delivery rate, OOF and OOF- perform better in the environments with higher mobility regularities, where they can be up to 60% better than MaxProp∗ and 100% better than Delegation.

Limitations: the proposed protocols suffer from several limitations. (1) Like all of the other routing algorithms in delay tolerant networks that do not rely on infrastructure, the proposed algorithms are unsuitable for a large amount of time-critical applications. (2) The proposed protocols assume regularities in the inter-meeting times to accurately predict delivery probabilities or delays. In networks where there is not much regularity in node mobility, improvement in routing performance becomes less significant. However, as can be found in our simulation results, our algorithms show the best performance improvement as the performances of all of the routing protocols degrade due to the lack of regularity in mobility. (3) The proposed protocols, in their current forms, demand relatively large amounts of computation and storage, which limits their application in larger networks. In practical use, different methods might need to attack the computation and storage problem such as: (a) using only the most socially active nodes as routing nodes, (b) using a compact table for delivery probabilities, which only stores the largest entries for each destination, or (c) using dynamic time-slots, e.g., we can set some several hours during the night as a single time-slot, when connectivity among the network hardly changes.



(a) Cambridge (b) Infocom

(c) Infocom2006 (d) CambridgeHaggleImoteContent

Fig. 5. Number of forwarding versus message time-to-live in the Cambridge Haggle trace.

References

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay Tolerant Networking Architecture. In *Internet draft: draft-irrf-dtnrg-arch.txt, DTN Research Group*, 2006.

[2] A. Vahdat and D. Becker. Epidemic Routing for Partially-connected Ad Hoc Networks. In *Technical Report, Duke University*, 2002.

[3] C. Liu and J. Wu. An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2009.

[4] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2006-09-15). http://crawdad.cs.dartmouth.edu/cambridge/haggle, September 2006.

[5] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proc. of ACM MobiCom*, 2006.

[6] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. of ACM MobiCom*, 2007.

[7] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. of ACM WDTN*, 2005.

[8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. of IEEE INFOCOM*, 2006.

[9] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation Forwarding. In *Proc. of ACM MobiHoc*, 2008.

[10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *Lecture Notes in Computer Science*, 3126:239–254, August 2004.

[11] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proc. of IEEE PerCom*, 2007.

[12] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, 2007.

[13] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proc. of ACM MobiHoc*, 2003.

[14] M. Grossglauser and M. Vetterli. Locating Nodes with Ease: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion. In *Proc. of IEEE INFOCOM*, 2003.

[15] M. Y. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets. A Low-energy, Multi-copy Inter-contact Routing Protocol for Disaster Response Networks. In *Proc. of IEEE SECON*, 2009.

[16] Optimal Stopping and Applications. http://www.math.ucla.edu/∼tom/Stopping/Contents.html.

[17] E. Bulut, S. C. Geyik, and B. K. Szymanski. Efficient Routing in Delay Tolerant Networks with Correlated Node Mobility. In *Proc. of IEEE MASS*, 2010.

[18] E. Daly and M. Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proc. of ACM MobiHoc*, 2007.

[19] U. Acer, S. Kalyanaraman, and A. Abouzeid. Weak State Routing for Large Scale Dynamic Networks. In *Proc. of ACM MobiCom*, 2007.

[20] C. Liu and J. Wu. Routing in a Cyclic MobiSpace. In *Proc. of ACM MobiHoc*, 2008.

Cong Liu Cong Liu is an assistant professor at the Department of Computer Computer at Sun Yat-sen University. He received his Ph.D. degree in computer science from Florida Atlantic University. Before that, he received his M.S. degree in computer software and theory from Sun Yat-sen (Zhongshan) University, China, in 2005, and his B.S. degree in micro-electronics from South China University of Technology in 2002. He is currently working toward his PhD. degree in the department of computer science, Florida Atlantic University, under the supervision of Dr. Jie Wu. He has served as a reviewer for many conferences and journal papers. His main research interests include routing in Mobile Ad hoc Networks (MANETs) and Delay Tolerant Networks (DTNs), Deep Packet Inspection, and Transaction Processing.

Jie Wu Jie Wu is a chair and a professor in the Department of Computer and Information Sciences, Temple University. Prior to joining Temple University, he was a program director at National Science Foundation. His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 500 papers in various journals and conference proceedings. He serves in the editorial board of the IEEE Transactions on Computers, IEEE Transactions on Mobile Computing, and Journal of Parallel and Distributed Computing. Dr. Wu is program cochair for IEEE INFOCOM 2011. He was also general cochair for IEEE MASS 2006, IEEE IPDPS 2008, and DCOSS 2009. He has served as an IEEE Computer Society distinguished visitor and is the chairman of the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the IEEE.