Proc. 2nd Annual Conference of International Technology Alliance, ACITA 2008, London, UK, September 2008, pp. 70-77

Minimizing Average Spraying Cost for Routing in Delay Tolerant Networks

Eyuphan Bulut, Zijian Wang and Boleslaw K. Szymanski
Department of Computer Science and Center for Pervasive Computing and Networking
Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA
{bulute, wangz, szymansk}@cs.rpi.edu

***Abstract*—In this paper, we study cost efficient multi-copy spraying algorithm for routing in Delay Tolerant Networks (DTN) in which a source-to-destination path does not exist most of the time. We present a novel idea and the corresponding algorithm for achieving the average minimum cost of packet transmission while maintaining the desired delivery rate by the given deadline. In the presented algorithm, the number of message copies in the network depends on the urgency of meeting the delivery deadline for that message. We find the parameters of efficient copying strategy analytically and validate the analytical results with simulations. The results demonstrate that our time dependent spraying algorithm achieves lower cost of message copying than the standard spraying algorithm while maintaining the desired delivery rate.**

I. Introduction

Delay Tolerant Networks (DTNs) are wireless networks in which at any given time instance, the probability that there is an end-to-end path from the source to destination is low. There are many examples of such networks in real life including wildlife tracking sensor networks [1], military networks [2] and vehicular ad hoc networks [4]. Since the standard routing algorithms assume that the network is connected most of the time, they fail in routing packets in DTNs.

The transient network connectivity needs to be of primary concern to the design of routing algorithms for DTNs. Hence, in recent years, new algorithms using buffering and contact time schedules have been proposed. Since most of the nodes in a DTN are mobile, the connectivity of the network is maintained only through mobile nodes when they come into transmission ranges of other nodes. If a node has a message copy but it is not connected to another node, it stores the message until an appropriate communication opportunity arises. The important considerations in such a design are (i) the number of message copies that are distributed to the network, and (ii) the selection of nodes to which the message is replicated.

In this paper, we study how to distribute the copies of a message among the potential relay nodes in such a way that the predefined percentage of all messages meets the given delivery deadline with the minimum number of copies used. Unlike the previous algorithms, we introduce a time dependent copying scheme which basically considers the time remaining to the given delivery deadline when making copying decisions.

The idea of our scheme is as follows. We first spray smaller number of copies than necessary to guarantee the delivery of the predefined percentage of all messages to the destination before the given delivery deadline. If the delivery does not happen for a certain period of time, then we spray some additional copies of the message to increase the probability of its delivery. Consequently, if an early delivery with small number of copies happens frequently enough, the average number of copies used by each message will be reduced compared to the algorithm with the constant number of messages sprayed.

The remainder of the paper is organized as follows. In Section II, we present the previous work done on this topic and discuss some basic mobility assisted routing concepts. We also comment about the differences between our algorithm and the others. In Section III, we describe our algorithm in detail and provide an analysis of its different variants. In Section IV, we evaluate the presented algorithm using simulations and demonstrate the achieved improvements. We also compare the results of our analysis with the simulation results. Finally, we offer conclusion and outline the future work in Section V.

II. Related Work

Routing algorithms for delay tolerant networks are generally classified as either replication based or coding based [13]. In replication based algorithms, multiple or a single copy of the message is generated and distributed to other nodes (often referred to as relays) in the network. Then, any of these nodes, independently of others, try to deliver the message copy to the destination. In coding based algorithms, a message is converted into a large set of code blocks such that any sufficiently large subset of these blocks can be used to reconstruct the original message. As a result, a constant overhead is maintained and the network is made more robust against the packet drops when the congestion arises. However, these algorithms introduce an overhead of an extra work needed for coding, forwarding and reconstructing code blocks.

Epidemic Routing [3] is an approach used by the replication based routing algorithms. Basically, during each contact between any two nodes, the nodes exchange their data so that they both have the same copies. As a result, the fastest spread of copies is achieved yielding the optimum delivery time. However, the main problem with this approach is the overhead incurred in bandwidth, buffer space, energy and time used by the greedy copying and storing of messages. Hence, this approach is inappropriate for resource constrained networks. To address this weakness of epidemic routing, the algorithms with controlled replication or spraying have been proposed [5], [6], [7], [14]. In these algorithms, only a small number of copies are distributed to other nodes and each copy is delivered to the destination independently of others. Of course, such an approach limits the aforementioned overhead and uses the resources efficiently.

The replication based schemes with controlled replication differ from each other in terms of their assumptions about the network. Some of them assume that the trajectories of the mobile devices are known while some others assume that only the times and durations of contacts between nodes are known. There are also some algorithms which assume zero knowledge about the network. The algorithms which fall in this last category seem to be the most relevant to applications because in most of the examples of delay tolerant networks from real life, neither the contact times nor the trajectories are known for certain. Consider the difficulty of acquiring such information in a wild life tracking application where the nodes are attached to animals that move unpredictably.

The algorithms which assume zero knowledge about the network include the one presented in [9], as well as Max-Prob [12], SCAR [11] and Spray and Wait [8]. In each of these algorithms limited number of copies are used to deliver a message. Yet, the process of choosing the nodes for placing new message copies is different in each of them. In [9] and MaxProb, each node carries its delivery probability which is updated in each contact with other nodes. If a node with a message copy meets another node that does not have the copy, it replicates the message to the contact node only if that node's delivery probability is higher than its own. A similar idea is used in SCAR. Each node maintains a utility function which defines the carrier quality in terms of reaching the destination. Then, each node tries to deliver its data in bundles to a number of neighboring nodes which have the highest carrier quality.

In [8] Spyropoulos et al. propose two different algorithms called Source Spray and Wait, and Binary Spray and Wait, respectively. While in the former, only the source is capable of spraying copies to other nodes, in the latter all nodes having the copy of the message are also allowed to do so. In Binary Spray and Wait, when a node copies a message to another node, it also gives the right of copying the half of its remaining copy count to that node. This results in distributed and faster spraying compared to the source spraying, but once the spraying is done, the expected delivery delay is the same for both. The authors provide the expected delay of message delivery in these two algorithms in [14].

Although there are many algorithms utilizing the controlled flooding approach, the idea of copying depending on the urgency of meeting the delivery deadline for a message has not been used by any of them. To the best of our knowledge this idea is new and it helps to decrease the average number of copies generated in the network. We will describe the details of this idea in the next section.

While designing a routing algorithm for mobile network, an important issue that must be considered is the model of mobility of nodes in the network. Random direction, random walk and random waypoint mobility models are the most often used by the published routing algorithms. Among these models, random direction model is considered more realistic than the others.

Node encounters in a mobility model are characterized by a parameter called expected meeting time (EM). It is assumed that the time elapsing between two consecutive encounters of a given pair of nodes is exponentially distributed with the mean EM. This parameter is specific to each mobility model and can be derived when the network parameters are known [10]. Moreover, as it is discussed in some papers, there is also another parameter called expected hitting time (ET) which is simply the expected inter-meeting time of a mobile node with a stable node in the network. Since in our network model, all nodes are identical and mobile, we do not use this definition.

III. Time Dependent Spraying

In this section, we first list the assumptions of our model and then describe our routing algorithm in detail. Moreover, we also present the analysis of the proposed algorithm with its variants.

We assume that there are M nodes walking on a $\sqrt{N}$ x $\sqrt{N}$ 2D torus according to the random direction mobility model. Each node has a transmission range R and all nodes are identical. Following the assumptions of the standard Spray and Wait algorithm, we assume that the meeting times of nodes are independent and identically distributed (IID) exponential random variables. Furthermore, we also assume the buffer space in a node is infinite (not crucial since we use the number of message copies that is comparable to the standard algorithm), and the communication between nodes is perfectly separable, that is, any communicating pair of nodes do not interfere with any other simultaneous communication. To be consistent with previous research, by L we denote the number of copies distributed to the network.

Given the mobility model, the expected time of delivery in Spray and Wait algorithm is equal to [14]:

$$\sum_{i=1}^{L-1} \frac{EM}{M-i} + \frac{M-L}{M-1} EW$$

This formula assumes that in the first $L-1$ contacts, the source node does not meet with the sink node and thus a wait phase is needed (probability of this happening is $\frac{M-L}{M-1}$). Here, EW is the expected duration of wait phase which is actually exponentially distributed with the mean $\frac{EM}{L}$. Note that, when $M >> L$ (which we enforce by limiting permissible values of L), duration of spraying phase is much shorter than the duration of waiting phase, so we can assume that the expected delivery time in Spray and Wait algorithm is exponentially distributed with the mean $\frac{EM}{L}$.

Figure 1 shows the cumulative distribution function of the expected delay of Spray and Wait algorithm for different L values. Clearly, when L increases, the mean value ($1/\lambda = EM/L$) of exponential cdf decreases and the expected delay shrinks.

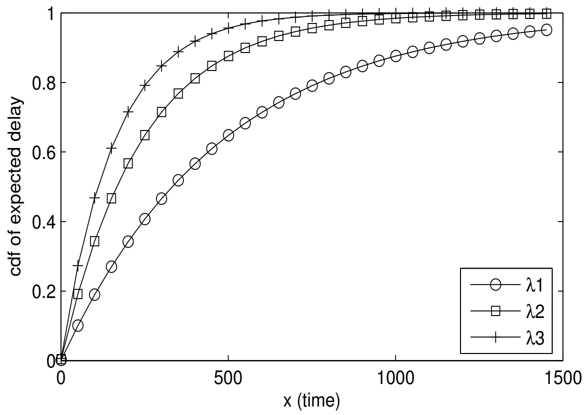

Fig. 1. The cumulative distribution function of probability of meeting the expected delay in the Spray and Wait algorithm for different λ values, where $\lambda_1 > \lambda_2 > \lambda_3$

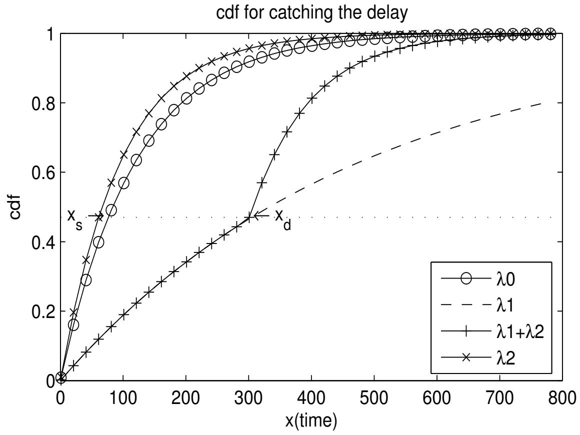

Fig. 2. The cumulative distribution function of delivery time of a message when spraying different number of copies in two different periods.

Our contribution to the spray and wait approach is to control spray of packets to other nodes based on the urgency of meeting the given delivery deadline. More precisely, the algorithm starts with spraying the message copies to fewer nodes than the minimum L needed and then waits for a certain period of time to see if the message is delivered. When the delivery does not happen, the algorithm sprays some additional number of copies and again waits for the delivery. This process repeats until either the message is delivered or the delivery deadline passes. Hence, as the time remaining to the delivery deadline decreases and delivery has not yet happened[1], the number of nodes carrying the message copy increases. To the best of our knowledge, this idea has not been used by any of the previously published algorithms for DTN routing.

Consider the Figure 2. It summarizes what our algorithm is designed to achieve. In this specific version of the algorithm, we allow two different spraying phases. The first one is done at the beginning and the second one is done at time x_d. The main objective of the algorithm is to attempt delivery with small number of copies and use the large number of copies only when this attempt is unsuccessful. With proper setting, the average number of copies sprayed in the network till delivery will be lower than in the case of spraying all messages at the beginning, while the delivery rate by the deadline will remain the same.

To analyze the performance of our algorithm analytically, we need to derive two formulas, one for the average number of copies used by the algorithm, and the second one for the cumulative distribution of the probability of meeting the delivery deadline with mixed number of copies (and therefore mixed λ values).

In our scheme, the term *period* refers to the time duration from the beginning of one spraying phase to the beginning of the next spraying phase. There may be multiple spray phases and the corresponding periods between them, each of different length. We start with the analysis of the two period case to find out the optimal period length and the corresponding copy counts for each period.

1) Partitioning into Two Periods: If there are two periods until the message delivery deadline, the arising questions are when to finish the first period and start the second one and how many copies should be allowed in each. In other words, what should be the value of x_d in Figure 2 to minimize the average number of copies used by the algorithm?

Let's assume that the standard Spray and Wait algorithm uses L copies (including the copy in the source node) of a message to achieve the probability $p_d \approx 1$ of its delivery by the deadline t_d. Let's further assume that the Two Period Delayed Spraying algorithm sprays L_1 copies to the network at the beginning of execution and additional $L_2 - L_1$ copies at time x_d, the beginning of the second period. Then, the cumulative distribution function of the probability of delivering the message at or below time x is:

$$cdf(x) = \begin{cases} 1 - e^{-\alpha L_1 x} & \text{if } x \le x_d \\ 1 - e^{-\alpha L_2 (x - x_s)} & \text{if } x > x_d \end{cases}$$

where, $\alpha = 1/EM$ is the inverse of the expected meeting time of any pair of nodes. x_s is the delay with which the spraying with L_2 copies would need to start to match the performance of our algorithms in the second period. The value of the x_s can be found using the equality of both cdf functions at time x_d:

$$\begin{aligned} 1 - e^{-\alpha L_1 x_d} &= 1 - e^{-\alpha L_2 (x_d - x_s)} \\ x_s &= x_d \frac{L_2 - L_1}{L_2} \end{aligned}$$

The expected delivery ratio when L copies are used in the standard Spray and Wait algorithm is by definition $p_d = 1 - e^{-\alpha L t_d} \approx 1$. Our objective is to match this delivery rate by decreasing the average number of copies below L, the number of copies used in the Spray and Wait algorithm. Hence, at the delivery deadline, t_d, the following inequality must be

[1]We explain how the delivery of a message is acknowledged to other nodes at the end of Section III.

Algorithm 1 FindOptimalsInTwoPeriods(L)

minCost $= L$
for each $0 < L_1 < L$ **do**
  $L_{2floor} = L1 + \lfloor \alpha L_1 t_d (L - L_1) \rfloor$
  for $L_2 = L_{2floor}, L_{2floor} + 1$ **do**
    if $c_2(L_1, L_2)$ <minCost **then**
      minCost $= c_2(L_1, L_2)$
      $[L_{opt_1}$,$L_{opt_2}] = [L_1$,$L_2]$
    end if
  end for
end for
return $[L_{opt_1}$,$L_{opt_2}]$

satisfied:

$$1 - e^{-\alpha L_2 (t_d - x_s)} \geq 1 - e^{-\alpha L t_d}$$
$$L_2(t_d - x_d + x_d L_1 / L_2) \geq L t_d$$

We can use this inequality to bound x_d as $x_d \leq t_d \frac{L_2 - L}{L_2 - L_1}$. Moreover, the larger x_d is the lower the average copy count is with the same L_1 and L_2 values. Since our algorithm aims at decreasing the average copy count while maintaining the delivery rate of the standard spraying algorithm, then the optimal x_d must be the largest possible. Therefore:

$$x_d = t_d \frac{L_2 - L}{L_2 - L_1}$$

We want to minimize the average number of copies, $c_2(L_1, L_2)$ defined as:

$$\begin{aligned} c_2(L_1, L_2) &= L_1(1 - e^{-\alpha L_1 x_d}) + L_2 e^{-\alpha L_1 x_d} \\ &= L_1 + (L_2 - L_1) e^{-\alpha L_1 x_d} \end{aligned}$$

Note that if the message is not delivered in the first period, then the cost becomes L_2 copies. Substituting x_d in the above, we get:

$$c_2(L_1, L_2) = L_1 + (L_2 - L_1) e^{-\alpha L_1 t_d \frac{L_2 - L}{L_2 - L_1}}$$

Taking derivative of c_2 in regard of L_2, and comparing it to zero, we obtain:

$$L_2 = L_1 + \alpha L_1 t_d (L - L_1)$$

so $L_2 - L_1 = \alpha L_1 t_d (L - L_1)$ and therefore:

$$c_2^*(L_1) = L_1[1 + \alpha t_d (L - L_1) e^{-\alpha L_1 t_d + 1}]$$

Taking the derivative of the above function we can obtain the complicated formula on the optimal value of L_1^* as a function of L and t_d, and then taking the floor and ceiling we can compute the corresponding optimal values of L_2^* and again their floors and ceilings can be used to arrive at the result.

Analyzing the derivative of $c_2^*(L_1)$, we proved that there is a unique point $L_1^* > 0$ at which the cost function reaches the minimum if and only if $\alpha L t_d > 1$, or in other words, if and only if the required delivery rate by the deadline is greater than $1 - 1/e$. Succinctly, our methods brings the benefit if and only if $p_d > 1 - 1/e \approx 63\%$, a very reasonable condition for practical solutions. We can find the optimal values of L_1 and L_2 using a simple method which generalizes nicely to cases with more periods, so we will present it here.

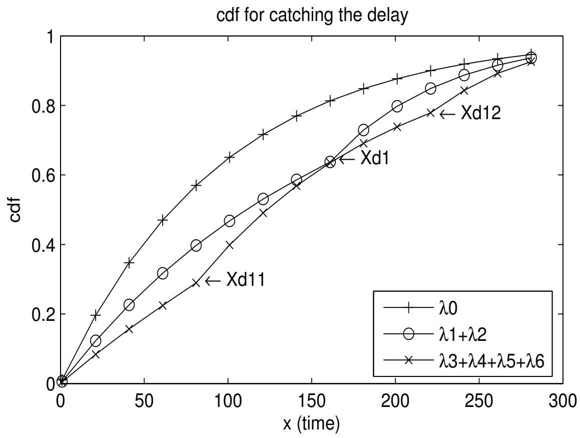

Fig. 3. Recursive partitioning algorithm using more than two periods of spraying to further decrease the total cost of spraying.

From the equation defining $c_2(L_1, L_2)$, it is clear that the average number of copies sprayed by our algorithm is larger then L_1, so for our algorithm to be able to decreased the average number copies below L, L_1 must be smaller than L. As a result, the following boundaries for L_1:

$$0 < L_1 < L$$

Since the possible values for all $L1$ variables are integers, we can use enumeration as explained in Algorithm 1 and obtain the optimal values quickly.

2) More Periods with Recursive Partitioning: In this section, we show that by applying recursive partitioning of each period, more periods can be used to lower the cost of spraying. Consider the example illustrated in Figure 3. From previous section, we know how to partition the entire time interval from the start to the delivery deadline into two periods. However, it is also possible to partition each of these two periods individually to decrease the cost of spraying even further. Although this may not be the optimal partitioning in the resulting number of periods, it still decreases the spraying cost.

Algorithm 2 IncreasePartitions(k,$x_d[\,]$,$L[\,]$)

min $=$ current copy cost with k periods
for each $1 \leq i \leq k$ **do**
  $[x'_d$,$L'] =$ PartitionIntoTwo(i,$x_d[\,]$,$L[\,]$)
  $c = Cost(k+1$,x'_d,$L')$
  if $c <$min **then**
    $p = [x'_d$,$L']$
    min $= c$
  end if
end for
return p

If we want to have three periods until the message delivery deadline, we can either partition the first period (with parameter $\lambda 1$) or the second period (with $\lambda 2$) and select the one which achieves the lowest cost. In other words, we need to select either ($\lambda 3$, $\lambda 4$, $\lambda 2$) or ($\lambda 1$, $\lambda 5$, $\lambda 6$) as the exponential factors in the three corresponding exponential functions. Furthermore, after obtaining the three period spraying, we can even run the same algorithm to find a lower cost spraying with four periods. However, we need to partition each period carefully considering the boundaries of possible L_i values.

Assume that we currently have k periods of spraying. Let L_i denote the copy count after spraying in period i^{th} and x_{d_i} denote the end time of that period. Then, the cumulative distribution function of the probability of delivering the message by the time x becomes:

$$cdf(x) = \begin{cases} 1 - e^{-\alpha L_1 (x - x_{s_1})} & [0, x_{d_1}] \\ 1 - e^{-\alpha L_2 (x - x_{s_2})} & (x_{d_1}, x_{d_2}] \\ ... & \\ 1 - e^{-\alpha L_k (x - x_{s_k})} & (x_{d_{k-1}}, x] \end{cases}$$

where x_{s_i} is the delay with which spraying with L_i copies would have to start to equal the cdf of our algorithm over the i^{th} spraying period, so of course $x_{s_1} = 0$ and for $i > 1$, we have:

$$x_{s_i} = \sum_{j=1}^{i-1} x_{d_j} \frac{L_{j+1} - L_j}{L_i} \quad (1)$$

This expression is easy to derive from the following simple iterative definition of x_{s_i} for $i > 1$ resulting from equality of the respective exponential functions at point $x_{d_{i-1}}$:

$$x_{s_i} = \frac{x_{s_{i-1}} L_{i-1} + x_{d_{i-1}} (L_i - L_{i-1})}{L_i}$$

We want to increase the number of periods to $k + 1$ while decreasing the total cost for spraying with the same delivery rate at the delivery deadline. Algorithm 2 and Algorithm 3 summarize the steps to achieve this goal.

Basically, we partition each period into two periods, one by one, to find the new cost for the current partitioning. Then, from these possible partitions, we select the one that achieves the lowest cost. For each period i, we need to find new numbers of copies L_i^-, L_i^+ to assign to each of the two newly created periods into which the original period is split. The delivery rate at the end of the both periods needs to stay unchanged but the average cost should be smaller than the original average cost of period i.

For each period being split, except the last one, there are the following bounds on those two numbers $L_{i-1} < L_i^- < L_i^+ < L_{i+1}$. The upper bound for the last period, which we will denote for convenience as L_{k+1} is discussed below.

Let x_{split} denote the boundary point in which the second inner period starts (i.e, the start of period for spraying additional L_i^+-L_i^- copies). The value of x_{split} can be found from equality of the probability of message delivery by the ends of the original and the split periods, so

$$\begin{aligned} 1 - e^{-\alpha L_i (x_{d_i} - x_{s_i})} &= 1 - e^{-\alpha L_i^+ (x_{d_i} - x_{s^+})} \\ L_i (x_{d_i} - x_{s_i}) &= L_i^+ (x_{d_i} - x_{s^+}) \end{aligned}$$

Algorithm 3 PartitionIntoTwo(i,x_d[ ],L[ ])

```
f1 = cdf(x_{i-1})
f2 = cdf(x_i)
minCost = L_i(f2-f1) // current cost of period
for each L_{i-1} < L_i^- < L_i do
  for each L_i^- < L_i^+ < L_{i+1} do
    Compute x_split using Eq.3
    Compute x_{s^-} using Eq.2
    internalCost = L_i^-(f2-f1)+L_i^+(f3-f2)
    if internalCost<minCost then
      minCost = internalCost
      x_opt = x_split
      [L_opt^-, L_opt^+] = [L_i^-, L_i^+]
    end if
  end for
end for
x'_d[ ] = [x_{d_1},...,x_{d_{i-1}},x_opt,x_{d_i},...,x_k]
L'[ ] = [L_1,...,L_{i-1},L_opt^-,L_opt^+,L_{i+1},...,L_k]
return [x'_d,L']
```

Substituting x_{s_i} and x_{s^+} by the formula in Eq. 1, which clearly s^- and s^+ must also obey, we obtain:

$$x_{split} = \frac{x_{d_i}(L_i^+ - L_i) + x_{d_{i-1}}(L_i - L_i^-)}{L_i^+ - L_i^-} \quad (2)$$

For the last period k, we need to find an upper bound for values of L_k^+ that we need to consider given L_k^-. The cost of this last period in terms of average number of copies used is slightly different than for other periods. Let p_k denote the probability of message delivery before the period k (and therefore also the first added period) starts. Similarly, let p_{split} denote probability of message delivery before the second added period starts. Of course, $p_k \le p_{split} \le p_d$, where p_d denotes the probability of delivery of the message by the deadline t_d. The cost of the original period k can be simply written as:

$$Cost_k = (1 - p_k) L_k$$

whereas the cost of the split period k is:

$$\begin{aligned} Cost_{ksplit} &= (1 - p_k) L_k^- + (1 - p_{split})(L_k^+ - L_k^-) \\ &\ge (1 - p_k) L_k^- + (1 - p_d)(L_k^+ - L_k^-) \end{aligned}$$

Since $Cost_k \ge Cost_{ksplit}$ for feasible solutions, then we obtain the following inequality:

$$(1 - p_k) L_k^- + (1 - p_d)(L_k^+ - L_k^-) \le (1 - p_k) L_k$$

which yields the following upper bound for feasible values of L_k^+:

$$L_k^+ \le L_k^- + (L_k - L_k^-) \frac{1 - p_k}{1 - p_d} = L_{k+1}$$

Algorithm 3 shows how the optimal partitioning of a single period $0 < i < k + 1$ is found. For convenience, we denote $L_0 = 0$. For each pair of number (L_i^-, L_i^+) such that $L_{i-1} \le$

$L_i^- < L_i^+ \leq L_{i+1}$, the cost of spraying is found and optimal pair which gives the minimum cost is selected.

3) Acknowledgment of Delivery: The designs of most of the routing protocols for delay tolerant networks do not explain in detail how the nodes in the network learn about the delivery of a message to the destination so spraying after the message delivery is not suppressed. Yet, this is a crucial issue in our algorithm because it directly affects the cost of copying of messages. If a message is delivered to destination, but a specific node is not notified about the delivery, this node will continue spraying the message, increasing the average cost of copying.

In this paper, we study two types of acknowledgments for notifying the nodes that the message has been delivered.

TYPE I: When destination receives a message, it first creates an acknowledgment for that message and sends it to other nodes within its range, which is assumed to be same for all the nodes in this case. Then, using epidemic routing, this acknowledgment is spread to all other nodes whenever there is a contact between a node having the acknowledgment and a node without it. Note that, since the acknowledgment packets are much smaller than data messages, the cost of this acknowledgment epidemic routing is small compared to the cost of routing the data packets. More costly is the delay with which all nodes in the network learn about the delivery of the message. During this delay, there may be useless spraying of the message increasing the total cost of copying.

TYPE II: In this type acknowledgment, we assume that the destination uses one time broadcast over the more powerful radio than the other nodes (case often present in practice) so the broadcast reaches all the nodes in the network. Like in the previous case, the acknowledgment message is short, so its broadcast is inexpensive. However, to make the scheme more efficient, we use the following idea. As the destination receives messages, it waits until the closest period change time (x_d) of any of the received message approaches. At that time, the destination broadcasts an acknowledgment message for all received so far messages. As a result, the destination broadcasts acknowledgment relatively infrequently, proportionally to a substantial fraction of the t_d, which is assumed large.

The second case results of course in better performance of delayed spraying than the first one. However, it may require higher energy consumption. In simulations, we compare the performances of both types of acknowledgment by showing how they affect the results of our algorithm.

IV. Simulation Results

In our simulations, we implemented the standard Source Spray and Wait algorithm using a Java based visual simulator. We deployed 100 mobile nodes (including the sink) onto a torus of the size 300 m by 300 m. All nodes (except the sink that has high range of acknowledgment broadcast in TYPE II case) are assumed to be identical and their transmission range is set at R = 10 m.

Nodes move according to random direction mobility model [10]. The speed of a node is randomly selected from the range $[4,\ 13] m/s$ and once the speed of the node is determined, it selects a random direction and then goes in the that direction as long as an epoch lasts. Each epoch duration is again randomly selected from the range $[8,\ 15] s$. With these paramters set, the value of EM can be computed, and accordingly, we set EM to 480, so $\alpha = 1/480$. We selected different values of L for the standard spraying algorithm. Then, for each value of L, using the results provided in [10], we found the corresponding value of t_d (for example, when $L = 6$, $t_d = 256$). In general, the large L is the shorter t_d is. All messages are generated at randomly selected nodes and are addressed to the sink node whose initial location is also decided randomly. With these settings, we collected some useful statistics from the network. The results are averaged over 1000 runs.

Using the Algorithm 1 we first found optimum combination of copy counts (L_1, L_2) for different L values. Then, using the Algorithm 2 and Algorithm 3 we have found optimum L_i combination when there are three periods. Table I shows the values of these optimum L_i's for different L values. In the simulations, we assumed TYPE II acknowledgment which stops any further spraying at the exact time of message delivery to the destination.

We have calculated the average number of copies in each of these optimum L_i combinations with simulations as well. Figure 4 and Figure 5 present the comparison of the results when there are two and three periods, respectively. In the two period case, the results are very close to each other, however in the three period case, the difference gets bigger because in our analysis we ignored the effect of spraying phase. When the number of periods increases, period lengths get smaller, so the effect of spraying phase on the cumulative distribution function increases.

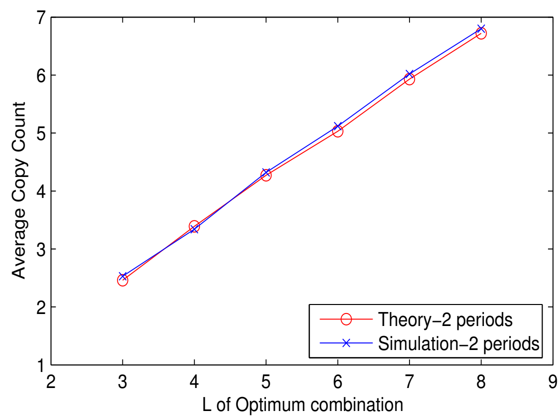

Fig. 4. The comparison of the average number of copies obtained via analysis and simulation for the two period case.

To compare the performance of our algorithm with the standard spraying algorithm, we first compare the average number of copies used in both algorithms when different types of acknowledgment mechanisms are used. Table II shows the average number of copies used by these two algorithms. With

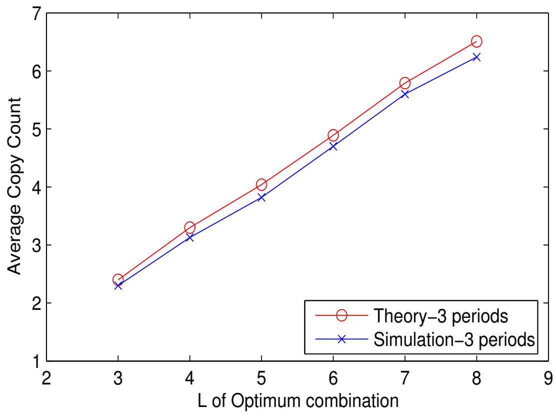

Fig. 5. The comparison of the average number of copies obtained via analysis and simulation for the three period case.

L	3	4	5	6	7	8
2 periods	(2,5)	(3,6)	(3,8)	(4,9)	(5,10)	(6,12)
3 periods	(2,3,6)	(2,4,7)	(3,5,9)	(4,6,10)	(5,7,11)	(5,8,14)

TABLE I
THE L_i COMBINATIONS THAT ACHIEVE THE MINIMUM AVERAGE NUMBER OF COPIES WHILE PRESERVING THE DELIVERY RATE.

both types of acknowledgment mechanisms, our algorithm uses fewer copies on average than standard spraying does. Moreover, in most of the cases, our algorithm with Type I mechanism uses fewer copies on average than the standard spraying algorithm with Type II mechanism does.

It should be noted that in standard spraying algorithm with L copy count, the average number of message copies sprayed to the network is less than L. This is simply because even in standard spraying which does all spraying at the beginning, there is non-zero chance that the message will be delivered before all copies are made.

L	Time-Based Spraying		Standard Spraying	
	Type I	Type II	Type I	Type II
3	2.59	2.53	2.99	2.97
4	3.65	3.43	3.96	3.90
5	4.66	4.38	4.93	4.82
6	5.61	5.23	5.91	5.70
7	6.52	6.05	6.83	6.51
8	7.40	6.81	7.76	7.36

TABLE II
THE AVERAGE NUMBER OF COPIES USED IN OURS AND THE STANDARD SPRAYING ALGORITHMS WHEN DIFFERENT TYPES OF ACKNOWLEDGMENTS ARE USED.

To further compare the performance of our algorithm with the standard spraying algorithm, we have measured via simulations some additional metrics for both of them. In these simulations, we used our algorithm with two periods. Figure 6 shows the the average message delivery delay. Figure 7 shows the average time of spraying completion (time by which the last copy is done). This value does not include the values for the cases when the message is delivered before all potential copies are sprayed.

In Figure 8, we show the success rate which is actually the percentage of all simulations that have delivery time less than or equal to the given deadline t_d. As mentioned earlier, for different values of L, the delivery deadline is different and it gets smaller as L increases.

Inspecting these three graphs, we observe that our time based spraying algorithm incurs higher average delay but it achieves the same delivery rate before the deadline as the standard spraying algorithm. Moreover, since our scheme postpones the spraying of all copies to later times, it finishes spraying later than the standard Spray and Wait algorithm. This results in lower memory usage averaged over time for our algorithm when compared with the usage incurred by the standard spraying algorithm.

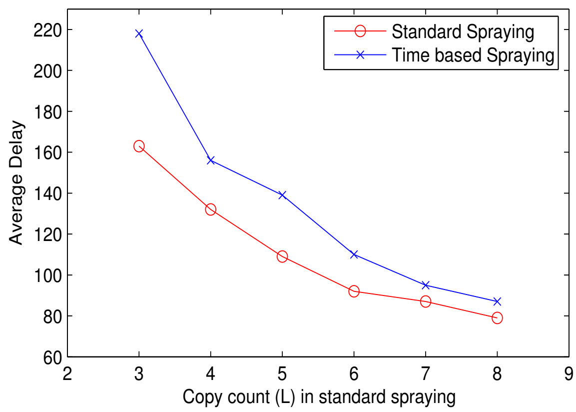

Fig. 6. The average delay comparison for the standard spraying and our algorithm.

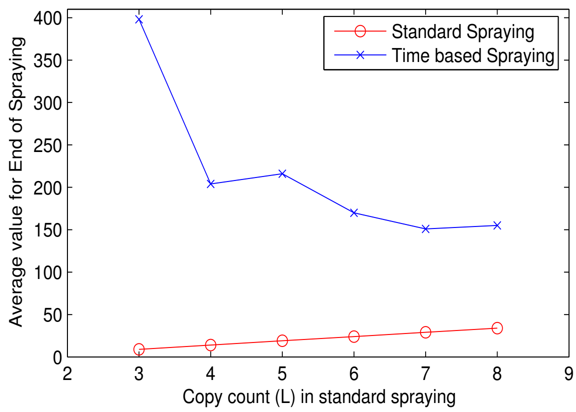

Fig. 7. A comparison of average times for end of spraying in the standard spraying and our algorithm.

Finally, Figure 9 shows the improvement achieved by our algorithm in the average number of copies per message for different values of L. This improvement is defined as the normalized copy gain value with respect to L, so it is the fraction

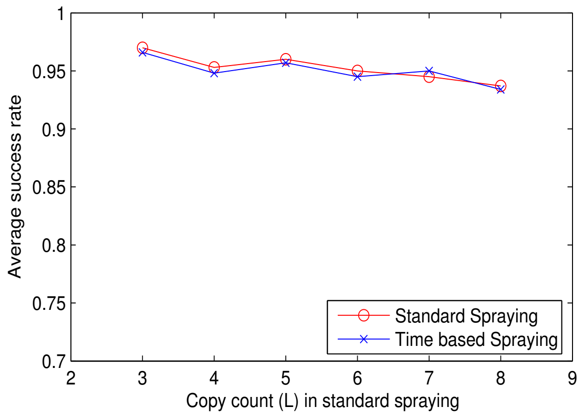

Fig. 8. The delivery rate comparison for the standard spraying and our algorithm.

$(L - L_{avg})/L$. While the two period case achieves about 16% benefit, the three period case shows higher improvement of about 20%. It is interesting that the improvement graph is flat for both cases, meaning that similar gains are obtained in terms of copy count for all L values. Proving this property analytically is the subject of our future research.

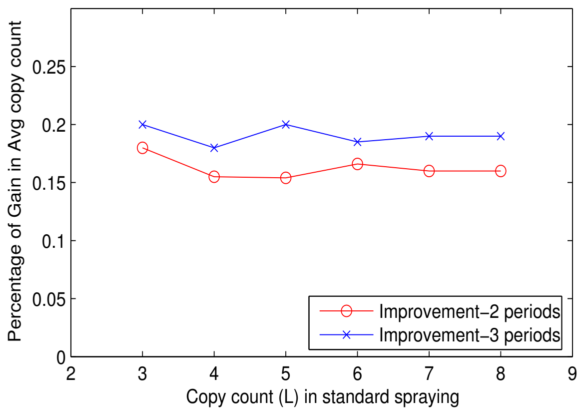

Fig. 9. The improvement obtained by our algorithm in the average number of copies used.

V. Conclusion and Future Work

This paper focuses on routing for Delay Tolerant Networks in which the nodes are disconnected most of the time. We introduce a time dependent spray and wait algorithm and, using simulation, we evaluate its performance. We first show analytically how to partition a standard spraying algorithm into two separate periods. Then, we present a generalization to larger number of periods which reduces the cost even further. Finally, we discuss results of simulations of our algorithm that confirm that the average number of copies used by our algorithm is lower than the average number of copies used by the standard spray and wait algorithm while achieving the same delivery rate by the delivery deadline.

In a future work, we plan to study different aspects of our algorithm. We will also investigate how more realistic radio links and mobility models affect our algorithm. Moreover, we also aim to apply the algorithm to a real test bed such as a disconnected bus network.

Acknowledgment

Authors wish to thank anonymous reviewers for their remarks which helped us to revise the paper.

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

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*, in Proceedings of ACM ASPLOS, 2002.
[2] *Disruption tolerant networking*, http://www.darpa.mil/ato/solicit/DTN/.
[3] A. Vahdat and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Duke University, Tech. Rep. CS-200006, 2000.
[4] J. Ott and D. Kutscher, *A disconnection-tolerant transport for drive-thru internet environments*, in Proceedings of IEEE INFOCOM, 2005.
[5] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, SIGMOBILE Mobile Computing and Communication Review, vol. 7, no. 3, 2003.
[6] Y. Wang, S. Jain, M. Martonosi, and K. Fall, *Erasure coding based routing for opportunistic networks*, in Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.
[7] T. Small and Z. Haas, *Resource and peroformance tradeoffs in delat tolerant wireless networks*, in Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.
[8] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*, ACM SIGCOMM Workshop, 2005.
[9] K. Harras, K. Almeroth, and E. Belding-Royer, *Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks*, In IFIP Networking, Waterloo, Canada, May 2005.
[10] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Performance Analysis of Mobility-assisted Routing*, MobiHoc, 2006.
[11] C. Mascolo and M. Musolesi, *SCAR: Contextaware Adaptive Routing in Delay Tolerant Mobile Sensor Networks*, Proceedings of International conference on Wireless communications and mobile computing, 2006.
[12] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, *MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks*, In Proc. IEEE Infocom, April 2006.
[13] A. Balasubramanian, B. N. Levine, A. Venkataramani, *DTN routing as a resource allocation problem*, ACM SIGCOMM, 2007.
[14] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case*, IEEE/ACM Transactions on Networking, 2008.