Cost-Effective Multiperiod Spraying for Routing in Delay-Tolerant Networks

Eyuphan Bulut, *Member, IEEE*, Zijian Wang, and Boleslaw Karol Szymanski, *Fellow, IEEE*

***Abstract*—In this paper, we present a novel multiperiod spraying algorithm for routing in delay-tolerant networks (DTNs). The goal is to minimize the average copy count used per message until the delivery while maintaining the predefined message delivery rate by the given deadline. In each period, some number of additional copies are sprayed into the network, followed by the wait for message delivery. At any time instance, the total number of message copies distributed to the network depends on the urgency of achieving the delivery rate by the given deadline for that message. Waiting for early delivery in the initial periods with a small number of copies in existence decreases the average number of copies sprayed in the network till delivery. We first discuss two- and three-period variants of our algorithm, and then we also give an idea of how the presented approach can be extended to more periods. We present an in-depth analysis of the algorithm and validate the analytical results with simulations. The results demonstrate that our multiperiod spraying algorithm outperforms the algorithms with a single spraying period.**

***Index Terms*—Cost efficiency, delay-tolerant network (DTN), routing.**

I. Introduction

Delay-tolerant networks (DTNs) [1] are wireless networks in which, at any given time instance, the probability that there is an end-to-end path from the source to the destination is low. There are many examples of such networks in real life, including ecology monitoring [2], [3], peoplenet [4], ocean sensor networks [5], [6], vehicular ad hoc networks [7], and military networks [8]. Since the standard routing algorithms assume that the network is connected most of the time, they fail when applied to routing of messages in DTNs.

The transient network connectivity needs to be of primary concern in 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. Routing of messages is based on *store-carry-and-forward* paradigm. That is, 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: 1) the number of message copies that are distributed to the network; and 2) 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 (desired delivery rate) of all messages are delivered by the given delivery deadline[1] with the minimum number of copies used. Unlike the previous algorithms based on message spraying, we introduce a time-dependent copying scheme that basically considers the time remaining to the given delivery deadline when making copying decisions.

The idea of our algorithm is as follows. It first sprays the number of copies smaller than necessary to guarantee the desired delivery rate of messages to the destination before the given delivery deadline. If the delivery does not happen for a certain period of time, then the algorithm sprays some additional copies of the message to increase the probability of its delivery. As a result, the algorithm partitions the time to the predefined deadline into several, variable-length periods, each composed of a spraying phase followed by the wait for delivery. In the spraying phase, a carefully chosen number of copies of the message are passed to nodes that do not possess one yet. Then, the waiting phase starts in which the delivery of the message with available copies is attempted independently by each copy holder. It is important to note that once the allowed number of copies for the given period has been distributed, no more copies will be distributed to any node until the beginning of next period. If the message is delivered in the early periods of such multiperiod spraying frequently enough, the average number of copies used per message will be reduced compared to the single-period spraying in which all copies of the message are distributed at the beginning of the routing.

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,

Manuscript received November 18, 2008; revised September 30, 2009; accepted February 11, 2010; approved by IEEE/ACM Transactions on Networking Editor S. Das. Date of publication March 29, 2010. Date of current version October 15, 2010. This work was sponsored by the US Army Research Laboratory and the U.K. 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 US Government, the U.K. Ministry of Defence, or the U.K. Government. The US and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

The authors are with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: bulute@cs.rpi.edu; wangz@cs.rpi.edu; szymansk@cs.rpi.edu; szymab@rpi.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2043744

[1]We call the TTL values assigned to messages when they are generated at source nodes as the delivery deadline of the messages.

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 a conclusion and outline the future work in Section V.

II. Related Work

There are various classifications of routing algorithms for DTNs [9], [10]. Here, we divide them into two classes: replication-based algorithms and coding-based algorithms. 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, all of these nodes, independently of others, try to deliver the message copy to the destination. In coding-based algorithms [11], [12], 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. Consequently, 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 extra work needed for coding, forwarding, and reconstructing code blocks.

Epidemic routing [13] 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 the result, the fastest spread of copies is achieved yielding the shortest delivery time.

The performance analysis of epidemic routing is well studied in many articles, including [14] and [15]. The main problem with this approach is the overhead incurred in bandwidth, buffer space, and energy consumption 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 [16]–[19], [31], [32]. 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 provides an efficient utilization of network resources.

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. Moreover, in some of them [20], it is assumed that even the node movements can be controlled. Other than these studies that assume some additional features, there are also some works that assume zero knowledge about the network. The algorithms that fall in this last category seem to be the most relevant to the applications because most often neither the contact times nor the trajectories are known for certain in the applications of DTNs in real life. An example could be a wildlife tracking application where the nodes are attached to animals that move unpredictably.

The algorithms that assume zero knowledge about the network include the one presented in [21], MaxProb [22], SCAR [23], and Spray and Wait [24]. In each of these algorithms, a 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 [21] 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 this node in contact only if that node's delivery probability is higher than its own probability. A similar idea is used in SCAR. Each node maintains a utility function that 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 that have the highest carrier quality.

In [24], Spyropoulos *et al.* propose a single-period spraying algorithm in which all the message copies are given to the other nodes at the beginning, then the waiting phase is entered, and the delivery of the message by any of these copies is expected. Here, note that this algorithm is a specific case of our algorithm in which the number of periods is just one. In that paper, authors also propose two different ways for the distribution of message copies to the network: Source Spray and Wait and Binary Spray and Wait. While only the source is capable of spraying copies to other nodes in the former, all nodes having a copy of the message are also allowed to do so in the latter. In Binary Spray and Wait, when a node copies a message to another node, it also passes the right of copying 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. In [32], an analysis on the expected delivery delay of messages in these two algorithms is provided by the same authors.

Although there are many algorithms utilizing the controlled flooding approach, to the best of our knowledge, the idea presented in this paper is completely new and has not been used by any of the prior work. The main contribution of this work is the distribution of the message copying process over many periods with increasing urgency of meeting the desired delivery rate by the deadline. The resulting adaptivity of the number of copies sprayed to the network is completely different from other adaptive copying strategies [25], [26]. The presented multiperiod spraying algorithm can be considered as a generalization of spraying-based algorithms such as the single-period spray-and-wait algorithm presented in [24]. We show under what conditions such generalization reduces the average number of copies used per message without decreasing the rate of messages delivered by the deadline. The details of this novel approach are given in the next section.

While designing a routing algorithm for mobile networks, an important issue that must be considered is the model of mobility of nodes in the network. There are many mobility models proposed for mobile nodes [27]–[29]. However, random direction, random walk, and random waypoint mobility models are the ones used most often by the published routing algorithms.

In general, node encounters in a mobility model could be characterized by a parameter called expected intermeeting time (EM). In many models, it is assumed that the time elapsing between two consecutive encounters of a given pair of nodes is exponentially distributed with the mean EM. However, the distribution of the intermeeting times of the network nodes is specific to each mobility model, so this parameter can be derived

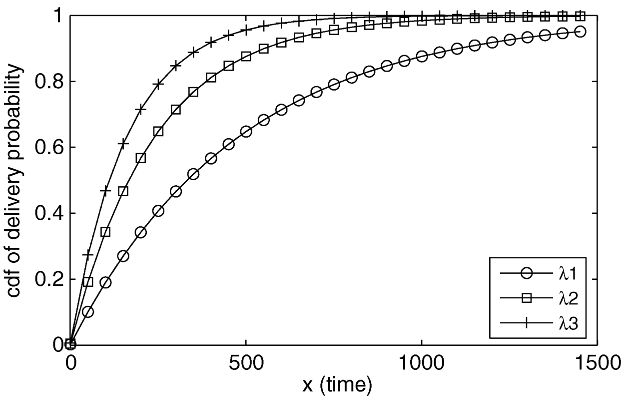


Fig. 1. The cumulative distribution function of delivery probability in the Spray and Wait algorithm for different values of $\lambda = L/EM$, where $\lambda_1 < \lambda_2 < \lambda_3$.

when the network parameters and the assumed mobility model are known [30].

III. Multiperiod 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.

A. Network Model and Assumptions

We assume that there are M nodes moving on a $\sqrt{N} \times \sqrt{N}$ 2D torus according to a random mobility model. Each node has a transmission range R, and all nodes are identical. The meeting times of nodes are assumed to be independent and identically distributed (IID) exponential random variables. Furthermore, we also assume that the buffer space in a node is unlimited (this assumption is not crucial since the presented algorithm uses the predefined number of copies with the maximum number comparable to the single-period spraying algorithm). We also assume that 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 that each message distributes to the network.

In Spray and Wait algorithm [32], the delivery of a message can happen both in spray and wait phases. The probability of message delivery at or before time t when there are L copies of the message in the network is $p_d = 1 - e^{-\alpha L t}$, where $\alpha = 1/EM$ is the inverse of the expected intermeeting time between two consecutive encounters of any pair of nodes. During waiting phase, since L is constant, p_d grows with the same L value. However, since the number of copies increases during the spraying phase, p_d function changes each time a new copy is distributed to other nodes.

To simplify the analysis of message delivery probability, we assume in this paper that $M \gg L$, which is often true in DTNs and which we enforce by limiting permissible values of L. Moreover, for DTNs to be of practical use, the delivery probability p_d must be close to 1, so we assume also that $p_d \geq 0.9$. We will show below that from these two assumptions, it follows that the formula $p_d = 1 - e^{-\alpha L t}$ is a good approximation of the delivery probability at times $t \geq t_d$.

At the ith encounter with another node, the spraying node delivers the message to the destination with probability $1/(M-i)$, so the total probability that the message is delivered during spraying is between L/M and $L/(M-L)$, and since $M \gg L$, L/M is a good approximation of this probability. Binary spraying uses $\log(L)$ steps, each with average time about EM/M (in kth step, 2^{k-1} nodes spray a message copy to 2^{k-1} other nodes), so the total spraying delay is $\log(L)EM/M$. The approximated formula achieves the same delivery probability at the earlier time EM/M, and from that time on, it matches the behavior of the algorithm perfectly. Hence, the average difference between times at which algorithm and formula achieve the same delivery probability is $d = (\log(L) - 1)EM/M$. Thus, the relative error $e(p_d)$ of using the approximate formula for p_d is

$$e(p_d) = 1 - \frac{\left(1 + e^{-\alpha L(t_d - d)}\right)\left(1 - e^{-\alpha L t_d}\right)}{\left(1 - e^{-2\alpha L(t_d - d)}\right)} \approx -(1 - p_d)(\log(L) - 1)\frac{L}{M}.$$

Since $p_d \geq 0.9$, for $L < 2048$ (so much beyond the range of useful values of L), the relative error of approximation is smaller than L/M, which is a small fraction for $M \gg L$.

Fig. 1 shows the cumulative distribution function (cdf) of the delivery probability in a single-period 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 together with the time needed to reach the desired delivery probability shrinks.

Our main contribution is to introduce and analyze the multiperiod spraying algorithm and to show under what condition it is more effective than the single-period spraying. In our algorithm, spraying of message copies is defined by the urgency of meeting the desired delivery probability by the given delivery deadline. More precisely, the algorithm starts with spraying fewer message copies than the minimum L needed by the single spraying algorithm, 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 copies of a message and again waits for the delivery. This process repeats until either the message is delivered[2] or the delivery deadline passes. Hence, as the time remaining to the delivery deadline decreases and delivery has not yet happened, 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.

Fig. 2 summarizes what our algorithm is designed to achieve. In this specific version of the algorithm, we allow two different spraying phases. The first one starts without delay, and the second one starts at time x_d. The main objective of the algorithm is to attempt delivery with a small number of copies and use the large number of copies only when this attempt is unsuccessful. With proper setting, the average number of

[2]At the end of Section III, we explain how the delivered messages are acknowledged to other nodes.

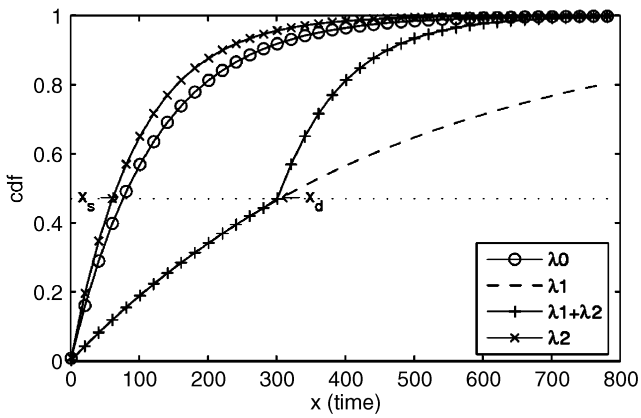


Fig. 2. The cumulative distribution function of delivery probability of a message when different copies are sprayed in two different periods.

copies sprayed until the delivery time can be lower than in the case of spraying all messages without delay, while the delivery probability by the deadline remains 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 delivery with the increasing number of copies (and therefore with the increasing 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 spraying phases and the corresponding periods between them, each of different length. In the next section, we start with the analysis of the two-period case to find the optimal period length and the corresponding copy counts for each period. In subsequent sections, we analyze the three- and multiple-period cases.

B. Two-Period Case

If there are two periods until the message delivery deadline, the questions that need to be answered are "*when should the first period finish and the second one start?*" and "*how many copies should be allowed in each?*" In other words, what should be the value of x_d in Fig. 2 to minimize the average number of copies used by the algorithm, and how many copies should be sprayed in each period?

Let's assume that the single-period spray-and-wait algorithm uses L copies (including the copy in the source node) of a message to achieve the desired delivery probability p_d by the deadline t_d. Let's further assume that the *Two-Period 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 cdf of the probability of message delivery at time x is

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

where x_s is the delay with which the spraying with L_2 copies would need to start to match the performance of our algorithm in the second period (see Fig. 2). The value of the x_s can be found from the equality of respective cdf functions at time x_d

$$1 - e^{-\alpha L_1 x_d} = 1 - e^{-\alpha L_2 (x_d - x_s)}$$
$$x_s = x_d \frac{L_2 - L_1}{L_2}.$$

The expected delivery probability when L copies are used in the single-period spray-and-wait algorithm is by definition $p_d = 1 - e^{-\alpha L t_d}$. Our objective is to match this delivery probability while decreasing the average number of copies below L. Hence, by the delivery deadline t_d, the following inequality must be satisfied:

$$1 - e^{-\alpha L_2 (t_d - x_s)} \geq 1 - e^{-\alpha L t_d}$$
$$L_2 \left(t_d - x_d + x_d \frac{L_1}{L_2} \right) \geq L t_d.$$

We can use this inequality to bound x_d as $x_d \leq t_d((L_2 - L)/(L_2 - L_1))$. As x_d gets larger, the average copy count decreases when L_1 and L_2 values remain constant. Since our algorithm aims at decreasing the average copy count while maintaining the delivery probability of the single-period spraying algorithm at time t_d, the optimal x_d must be the largest possible, and 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

$$c_2(L_1, L_2) = L_1 (1 - e^{-\alpha L_1 x_d}) + L_2 e^{-\alpha L_1 x_d}$$
$$= L_1 + (L_2 - L_1) e^{-\alpha L_1 x_d}.$$

Note that if the message is not delivered in the first period, then the cost (we define cost as the number of copies used per message) 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 dc_2/dL_2, we obtain

$$\frac{dc_2}{dL_2} = \left(1 - \alpha L_1 t_d + \alpha L_1 t_d \frac{L_2 - L}{L_2 - L_1} \right) e^{-\alpha L_1 t_d \frac{L_2 - L}{L_2 - L_1}}.$$

We are only interested in the sign of this derivative, so we can ignore always-positive factor $e^{-\alpha L_1 t_d ((L_2 - L)/(L_2 - L_1))}$. For the same reason, we can also multiply the result by always-positive factor $L_2 - L_1$. As we consider only values $L_2 > L_1$, then we obtain

$$sgn\left(\frac{dc_2}{dL_2} \right) = sgn\left(L_2 - L_1 - \alpha L_1 t_d (L - L_1) \right).$$

We conclude that sign of derivative changes only once at

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

and it changes from negative to positive, so the cost function has the unique minimum at this point. Hence, $L_2^* - L_1 = \alpha L_1 t_d (L - L_1)$, and therefore

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

Again, by taking the derivative of c_2^* in regard of L_1 and comparing it to zero, we can obtain the optimum value of L_1. Let $z = \alpha t_d L_1 - 1$ and $A = \alpha t_d L$, then

$$c_2^*(z)\alpha t_d = 1 + z + \left(A - 1 + (A-2)z - z^2\right) e^{-z}.$$

Setting $f(z) = c_2^*(z)\alpha t_d$, we get

$$f(z) = 1 + z + \left(A - 1 + (A-2)z - z^2\right) e^{-z}.$$

Taking derivative of this function in regard of z, we obtain

$$f'(z) = 1 + e^{-z}(-1 - Az + z^2).$$

Then, by multiplying by an always-positive factor e^z, we obtain

$$g(z) = f'(z)e^z = e^z + z^2 - Az - 1.$$

We notice that because the number and types of extreme points of a function are defined by the number of zeros and the derivative signs near these zeros, functions $g(z)$ and $c_2^*(L_1)$ have the same number and types of extreme points related by equation $z_e = \alpha t_d L_{1e} - 1$.

The first zero of function $g(z)$ is at $z = 0$, and it corresponds to maximum when $A > 1$ because, near zero, $g(z) \approx 1 + z - Az - 1 = -z(A-1)$, so it is positive for $z < 0$ and negative for $z > 0$. If $A < 1$, then of course the point $z = 0$ is at the minimum, and it corresponds to $L_2 = L$, which means that the one-time spraying is optimal in such a case. For $A = 1$, the point $z = 0$ is neither, but then the derivative is nonnegative for $z > 0$. Therefore, one-time spraying is optimal again. However, the condition that $A > 1$ is satisfied in realistic applications because $p_d = 1 - e^{-A} > ((e-1)/e) \approx 63\%$. Usually, the reasonable values for p_d are in the percentage of high 90s, which means that the introduced algorithm will perform better when reasonable delivery probability by the deadline is required.

More formally, for $0 < 4z < \min(A - 1, 24/7)$, we have

$$e^z = \sum_{i=0}^{\infty} \frac{z^i}{i!} < 1 + z + \frac{z^2}{2} + \frac{z^3}{6} \sum_{i=0}^{\infty} \left(\frac{z}{4}\right)^i$$
$$= 1 + z + \frac{z^2}{2} + \frac{2z^3}{12 - 3z} < 1 + z + z^2$$

and $2z^2 - Az < -z$, so $g(z) < 1 + z - z - 1 = 0$. Similarly, for $-1 < z < 0$, we have $e^z > 1 + z$, so $g(z) > 1 + z + z^2 - Az - 1 = z(z - (A-1)) > 0$. This also means that there is at least one more zero point, as $g(z)$ is a continuous function, negative in close positive neighborhood of 0 and positive in $+\infty$. Moreover, considering the equality of two functions

$$e^z + z^2 - 1 = Az.$$

We notice that $e^z + z^2 + 1$ is a convex function that has at most two intersection points with any straight line, including the line Az. Hence, $e^z + z^2 + 1$ and Az intersect at $z = 0$ and, for $A > 1$, exactly once more, at the point corresponding to the minimum, which is the point of interest to us here. Indeed, let $z_{\text{opt}} > 0$ denote the nearest to zero intersection point of these two functions. From the convex property of the first function, it follows that to the right of z_{opt}, the first function is always above the straight line Az. Hence, these two functions cannot intersect for $z > z_{\text{opt}}$. Furthermore, it must be that $z_{\text{opt}} < A$, as for $z \geq A$ we have $e^z + z^2 - 1 > 1 + A + Az - 1 > Az$, so the cost function is already growing.

We conclude that there is a unique optimum point at $z_{\text{opt}} > 0$ if and only if $A > 1$, or in other words, if and only if the required delivery probability by the deadline is greater than $1 - (1/e)$, or succinctly, $p_d > 1 - (1/e) \approx 63\%$, a very reasonable condition for practical solutions. This point can be found in $\lceil \log A \rceil$ steps by bisecting the interval $(0, A)$ until we get the range of the solution within two consecutive integers. Then, we can use the floor and ceiling of the approximation to find an integer solution that we are interested in. Complexity of this algorithm is low, $O(logA)$, and because $A = -\ln(1 - p_d)$, A is the natural logarithm of the inverse of the probability of nondelivery of a message by the end of the deadline. Hence, the complexity is the polylogarithmic function of this inverse.

We can also find the optimal values of L_1 and L_2 by a simpler method, which generalizes nicely to cases with more periods, so we will present it here. From the equation defining $\mathrm{c}_2(L_1, L_2)$, it is clear that the average number of copies sprayed by our algorithm is larger than L_1, so for our algorithm to be able to decrease the average number of copies below L, L_1 must be smaller than L. As a result, the following boundaries for L_1 must hold:

$$0 < L_1 < L.$$

Since the possible values for all L_1 variables are integers, we can use enumeration method as explained in Algorithm 1 and obtain the optimal values relatively quickly in $O(L)$ steps. With constant values of EM and t_d, this is a logarithmic function of the inverse of nondelivery probability, hence it grows faster than complexity of finding a solution via the derivative of the cost function.

Algorithm 1 FindOptimalsInTwoPeriods(L, α, t_d)

opt_cost $= L$; opt_cts $= [L, L]$
for each $0 < L_1 < L$ **do**
  $L_{2\text{floor}} = \max(L + 1, L_1 + \lfloor \alpha L_1 t_d (L - L_1) \rfloor)$
  for $L_2 = L_{2\text{floor}}, L_{2\text{floor}} + 1$ **do**
    if $c_2(L_1, L_2) <$ opt_cost **then**
      opt_cost $= c_2(L_1, L_2)$; opt_cts $= [L_1, L_2]$
    end if
  end for
end for
return opt_cts

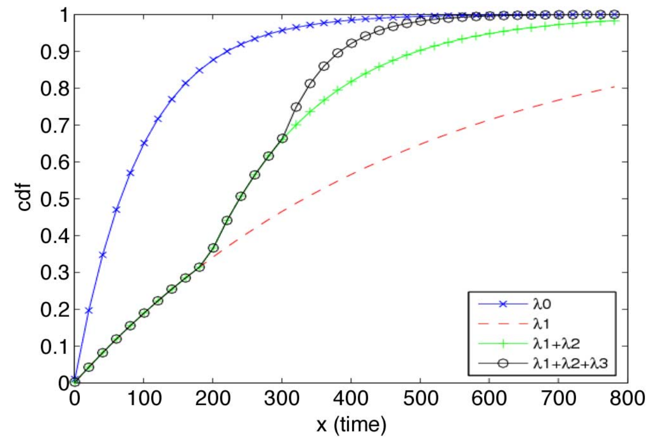


Fig. 3. The cumulative distribution function of delivery probability with copies sprayed in three different periods.

C. Three-Period Case

In this section, we assume that there are three spray-and-wait periods until the delivery deadline. In this case, we need to find two different boundary points that separate these three periods. Let x_{d_1} and x_{d_2} denote these boundary points. While the former stands at the boundary between the first and the second periods, the latter marks the boundary between the second and the third periods. The cdf of the probability of message delivery by the time x

$$\mathrm{cdf}(x) = \begin{cases} 1 - e^{-\alpha L_1 x}, & [0, x_{d_1}] \\ 1 - e^{-\alpha L_2 (x - x_{s_2})}, & (x_{d_1}, x_{d_2}] \\ 1 - e^{-\alpha L_3 (x - x_{s_3})}, & (x_{d_2}, x] \end{cases}$$

where x_{s_2} and x_{s_3} are the delays with which the second (L_2) and the third (L_3) spraying would have to start to equal the cdf of our algorithm over the second and third spraying periods, respectively. As before, using the equality of the functions at times x_{d_1} and x_{d_2}, we can obtain the values of x_{s_2} and x_{s_3}

$$1 - e^{-\alpha L_1 x_{d_1}} = 1 - e^{-\alpha L_2 (x_{d_1} - x_{s_2})}$$
$$x_{s_2} = x_{d_1} \frac{L_2 - L_1}{L_2}$$

and analogously

$$1 - e^{-\alpha L_2 (x_{d_2} - x_{s_2})} = 1 - e^{-\alpha L_3 (x_{d_2} - x_{s_3})}$$
$$x_{s_3} = x_{d_2} \frac{L_3 - L_2}{L_3} + x_{d_1} \frac{L_2 - L_1}{L_3}.$$

Fig. 3 illustrates our approach with three periods. Similar to the two-period case, we want to achieve the same or higher delivery probability p_d at the given deadline t_d while minimizing the average number of copies used. That is, we need to satisfy the following inequality:

$$1 - e^{-\alpha L t_d} \le 1 - e^{-\alpha L_3 (t_d - x_{s_3})}$$
$$L t_d \le L_3 (t_d - x_{s_3})$$
$$x_{d_2}(L_3 - L_2) + x_{d_1}(L_2 - L_1) \le t_d (L_3 - L).$$

Using this inequality, we can eliminate x_{d_2} because as x_{d_2} gets larger, the average copy count gets smaller when all other parameters L_1, L_2, L_3, x_{d_1} are kept constant. Therefore, replacing the above inequality with an equation, we obtain

$$x_{d_2} = \frac{t_d (L_3 - L) - x_{d_1}(L_2 - L_1)}{L_3 - L_2}.$$

Furthermore, the average copy count used in this three-period spraying can be defined as

$$c_3(L_1, L_2, L_3, x_{d_1}) = L_1 + (L_2 - L_1) e^{-\alpha L_1 x_{d_1}} + (L_3 - L_2) e^{-\alpha L_2 (x_{d_2} - x_{s_2})}.$$

When we substitute x_{s_2} and x_{d_2} in $c_3(L_1, L_2, L_3, x_{d_1})$ and take the derivative dc_3/dx_{d_1}, we obtain

$$\frac{dc_3}{dx_{d_1}} = -\alpha (L_2 - L_1) L_3 e^{-\alpha L_1 x_{d_1}} \times \left(\frac{L_1}{L_3} - e^{-\frac{\alpha L_2 \left(t_d (L_3 - L) - x_{d_1} (L_3 - L_1) \right)}{(L_3 - L_2)}} \right).$$

After ignoring the always-positive factors, the sign of the derivative becomes

$$sgn\left(\frac{dc_3}{dx_{d_1}}\right) = -sgn\left(L_1 - L_3 e^{-\frac{\alpha L_2 \left(t_d (L_3 - L) - x_{d_1} (L_3 - L_1) \right)}{(L_3 - L_2)}} \right).$$

We see that if $L_1 < L_3 e^{-(\alpha L_2 t_d (L_3 - L)/(L_3 - L_2))}$, the sign is always positive (note that $x_{d_1} \ge 0$ by definition)—that is, the cost function is always growing with increasing L_3. Therefore, minimum cost is obtained at $x_{d_1} = 0$. Otherwise, we notice that the sign of the derivative changes from negative to positive only once at

$$x_{d_1} = \frac{\alpha t_d L_2 (L_3 - L) + \ln(L_1/L_3)(L_3 - L_2)}{\alpha L_2 (L_3 - L_1)}. \quad (1)$$

For both cases, we obtained the optimum values of x_{d_1}. Then, we can easily obtain formula $c_3^*(L_1, L_2, L_3)$ by substituting x_{d_1} with these optimum values in corresponding conditions. Since $L_1 < L < L_3$ and $L_1 \le L_2 \le L_3$ and all these values are integers, by enumeration explained in Algorithm 2, we can simply find the copy counts (L_1, L_2, L_3) that gives the minimum copy count for a given L.

Algorithm 2 FindOptimalsInThreePeriods(L)

```
opt_cost = L; opt_cts = (L, L, L)
for each 0 < L1 < L do
  L2_Bound(L1 = ⌈(L1 + (L − L1))e^{αL1 td}⌉
  for each L1 < L2 ≤ L2_Bound(L1) do
    if L2 ≥ L and c2(L1, L2) < opt_cost then
      opt_cost = c2(L1, L2)
      opt_cts = (L1, L2, L2)
    end if
    if L1 < (L2 + 1)e^{−αL2 td (L2+1−L)} then
      L3_opt = max(L, L2) + 1
      if c3*(L1, L2, L3_opt) < opt_cost then
        opt_cost = c3*(L1, L2, L3_opt)
        opt_cts = (L1, L2, L3_opt)
      end if
    else
```

```
            R_1 = L_1 + (L_2 − L_1) ln(L_1 + αL_2 t_d(L − L_1)
            R_2 = R − (L_2 − L_1) ln(R)
            L_3opt = Find optimum L_3 by bisecting in [R_2, R_1]
            if c_3^*(L_1, L_2, L_3opt) < opt_cost then
               opt_cost = c_3^*(L_1, L_2, L_3opt)
               opt_cts = (L_1, L_2, L_3opt)
            end if
         end if
      end for
end for
return opt_cts
```

However, to use enumeration, we need to establish bounds on both L_2 and L_3. Using inequality $x_{d_1} < t_d$ that must be satisfied for the second period to start before the deadline for message delivery, we can calculate the upper bound for L_2, denoted as $L_{2_{\text{Bound}}}(L_1)$, as follows:

$$L > c_3^* > L_1 + (L_2 - L_1)e^{-\alpha L_1 t_d}$$
$$L_2 < L_1 + (L - L_1)e^{\alpha L_1 t_d} = L_{2_{\text{Bound}}}(L_1).$$

Now, we have the ranges for L_1 and L_2. Thus, for a given (L_1, L_2), we will try to obtain the optimum L_3 value that makes the cost function minimum. When we take the derivative dc_3/dL_3, we obtain

$$\frac{dc_3}{dL_3} = \left(1 + \frac{\alpha L_2\left(t_d(L_2 - L) - x_{d_1}(L_2 - L_1)\right)}{(L_3 - L_2)}\right) e^m$$

where

$$m = \frac{-\alpha L_2 t_d(L_3 - L)}{(L_3 - L_2)} + x_{d_1}\frac{\alpha L_3(L_2 - L_1)}{(L_3 - L_2)}.$$

Since we are interested in the sign of the derivative, we can ignore the always-positive factor. Then, we have

$$sgn\left(\frac{dc_3}{dL_3}\right) = sgn\left(1 + \frac{\alpha L_2\left(t_d(L_2 - L) - x_{d_1}(L_2 - L_1)\right)}{(L_3 - L_2)}\right).$$

As we consider the values $L_3 > L_2$, we conclude that the sign of the derivative changes from negative to positive only once at

$$L_3^* = L_2\left[1 + \alpha\left(x_{d_1}(L_2 - L_1) - t_d(L_2 - L)\right)\right].$$

When we substitute x_{d_1} in this equation with the optimum x_{d_1} in (1), we have the following equation:

$$L_3 + (L_2 - L_1)\ln(L_3) = L_1 + (L_2 - L_1)\ln(L_1) + \alpha L_2 t_d(L - L_1).$$

Since to find the value of $L_{3_{\text{opt}}}$ in the above function is not so easy, we can instead find a range for $L_{3_{\text{opt}}}$ then by bisecting within the range we can reach the optimum integer value of L_3. It is obvious that $L_{3_{\text{opt}}}$ satisfies the following:

$$L_{3_{\text{opt}}} < L_1 + (L_2 - L_1)\ln(L_1) + \alpha L_2 t_d(L - L_1) = R_1.$$

On the other hand, $L_{3_{\text{opt}}}$ is also bigger than $R_1 - (L_2 - L_1)\ln(R_1)$ because

$$L_{3_{\text{opt}}} = R_1 - (L_2 - L_1)\ln\left(L_{3_{\text{opt}}}\right)$$
$$> R_1 - (L_2 - L_1)\ln(R_1) = R_2$$

Therefore, $L_{3_{opt}}$ lies within the range $[R_2, R_1]$. Since as L_3 increases, the value of $L_3 + (L_2 - L_1)\ln(L_3)$ increases, we can find this integer optimum value of L_3 by bisecting in this range. One can easily see that the complexity of this bisecting search is $O(\log_2(L_2))$.

In Algorithm 2, for each (L_1, L_2) pair, we first find the optimum L_3 minimizing the cost function, then we compare it with the current optimum cost. Here, note that if $L_1 < L_3 e^{-(\alpha L_2 t_d(L_3 - L)/(L_3 - L_2))}$, then $c_3^*(L_1, L_2, L_3)$ is obtained by using the optimum $x_{d_1} = 0$. Otherwise, $c_3^*(L_1, L_2, L_3)$ is computed using the optimum x_{d_1} value given in (1).

To assess complexity of Algorithm 2, we observe that $L_{2_{\text{Bound}}}(L_1)$ can be approximated as follows:

$$L_2 < L_1 + (L - L_1)e^{\alpha L_1 t_d} < L_1 + \max_{x \in (1, L-1)}(xe^{-\alpha t_d x})e^{\alpha L t_d}$$
$$\leq L + \frac{1}{e\alpha t_d(1 - p_d)},$$

because function $f(x) = xe^{-\alpha t_d x}$ has derivative $(1 - \alpha t_d x)e^{-\alpha t_d x}$ and therefore maximum at $x = 1/\alpha t_d$. Hence, the complexity is

$$\sum_{L_1=1}^{L-1} \sum_{L_2=L_1+1}^{L_1 + \frac{1}{e\alpha t_d(1-p_d)}} O\left(\log_2(L_2)\right).$$

In conclusion, the complexity of enumeration in this case is $O(-(L\log_2(e\alpha t_d(1 - p_d))/e\alpha t_d(1 - p_d)))$, so it is inversely proportional to the nondelivery probability times logarithm of the inverse of the nondelivery probability.

D. Increasing the Number of Periods by Recursive Partitioning

In this section, we show that by applying recursive partitioning of each period, more spraying periods can be created in such a way that the total cost of spraying can be decreased even more. An example is given in Fig. 4. From Section III-B, we know how to achieve the optimum partitioning of 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.

If we want to have three periods until the message delivery deadline, we can partition either the first period (with parameter λ_1) or the second period (with λ_2) and select the one which achieves the lower 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 corresponding three exponential functions. Furthermore, after obtaining the three-period spraying, we can 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 ith period and x_{d_i} denote the end time of that period. Then, the cdf of the probability of

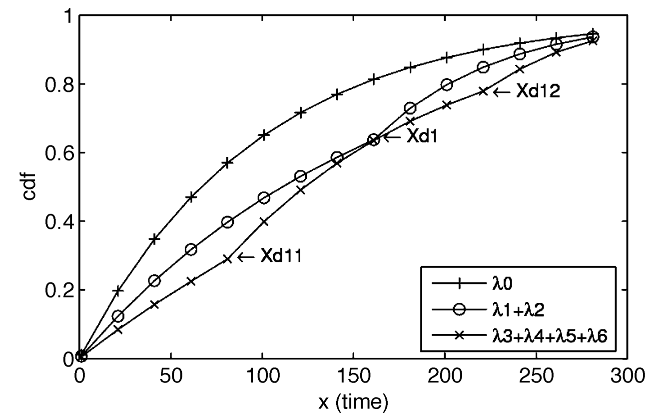


Fig. 4. Recursive partitioning algorithm to define more periods of spraying and further decrease the total cost of spraying.

message delivery by the time x becomes

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

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 ith spraying period. Obviously, $x_{s_1} = 0$, and for $i > 1$, we have

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

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

$$x_{s_i} = \frac{x_{s_{i-1}} L_{i-1} + x_{d_{i-1}} (L_i - L_{i-1})}{L_i}. \tag{2}$$

We want to increase the number of periods to $k+1$ while decreasing the total cost for spraying with the same delivery probability at the delivery deadline. Algorithms 3 and 4 summarize the steps to achieve this goal.

Algorithm 3 IncreasePartitions$(k, x_d[\,], L[\,])$

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

Algorithm 4 PartitionIntoTwo$(i, x_d[\,], L[\,])$

```
f_1 = cdf(x_{i-1})
f_2 = cdf(x_i)
min_cost = L_i(f_2 − f_1) //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 (3)
    Compute x_{s^-} using (2)
    internal_cost = L_i^-(f_2 − f_1) + L_1^+(f_3 − f_2)
    if internal_cost < min_cost then
      min_cost = internal_cost
      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']
```

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 number 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 probability at the end of 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:

$$L_{i-1} < L_i^- < L_i^+ < L_{i+1}.$$

We can also find an upper bound for the last period, which we will denote for convenience as L_{k+1}. 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 the equality of the probability of message delivery by the ends of the original and the split periods

$$1 - e^{-\alpha L_i (x_{d_i} - x_{s_i})} = 1 - e^{-\alpha L_i^+ (x_{d_i} - x_{s^+})}$$
$$L_i (x_{d_i} - x_{s_i}) = L_i^+ (x_{d_i} - x_{s^+}).$$

Substituting x_{s_i} and x_{s^+} by the formula in (2), which clearly s^- and s^+ must also obey, we obtain

$$x_{\mathrm{split}} = \frac{x_{d_i} (L_i^+ - L_i) + x_{d_{i-1}} (L_i - L_i^-)}{L_i^+ - L_i^-}. \tag{3}$$

For the last period k, we need to find an upper bound for the values of L_k^+ with given L_k^-. The cost of this last period in terms of average number of copies used is slightly different than the cost of other periods. Let p_k denote the probability of message delivery before the period k starts. Similarly, let p_{split} denote probability of message delivery before the second added period starts. Of course, $p_k \le p_{\mathrm{split}} \le p_d$, 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

$$\mathrm{Cost}_k = (1 - p_k) L_k$$

whereas the cost of the split period k is

$$\begin{aligned}\mathrm{Cost}_{k\mathrm{split}} &= (1-p_k)L_k^- + (1-p_{\mathrm{split}})\left(L_k^+ - L_k^-\right) \\ &\geq (1-p_k)L_k^- + (1-p_d)\left(L_k^+ - L_k^-\right).\end{aligned}$$

Since we want $\mathrm{Cost}_k > \mathrm{Cost}_{\mathrm{split}}$, then the following inequality must hold:

$$(1-p_k)L_k^- + (1-p_d)\left(L_k^+ - L_k^-\right) < (1-p_k)L_k$$

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

$$L_k^+ < L_k^- + \left(L_k - L_k^-\right)\frac{1-p_k}{1-p_d} = L_{k+1}.$$

Algorithm 3 shows how the optimal partitioning of a single period i (where $0 < i < k+1$) is found. For convenience, we denote $L_0 = 0$. For each pair of numbers (L_i^-, L_i^+) such that $L_{i-1} \leq L_i^- < L_i^+ \leq L_{i+1}$, the cost of spraying is found, and the optimal pair that gives the minimum cost is selected. Clearly, the complexity of this algorithm is $O(L^2)$.

E. Acknowledgment of Delivery

The descriptions of most of the published routing protocols for DTNs do not contain details of how the nodes in the network learn about the delivery of a message to the destination to avoid spraying after the message delivery. 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 a 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 about the delivery of the messages.

Type I: When a 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 the 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 carrying the acknowledgment and a node without it. Note that often the acknowledgment packets (which carry only acknowledged message id) are much smaller than data messages. In such cases, 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 already delivered message, increasing the total cost of copying.

Type II: In this type of acknowledgment, we assume that the destination uses a one-time broadcast over the more powerful radio than the other nodes (the assumption often satisfied 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 epidemiology-inspired idea.

We considered an environment in which, at different times, individuals are infected by different pathogens. Each pathogen has an incubation period during which the infected individual is not contagious. After the incubation period, the sick individual is contagious and able to infect others. We assume that there are effective vaccines for all pathogens, and we want to vaccinate the entire population with the proper mix of vaccines in the most efficient way. The best way to achieve this goal is to wait until the closest end of an incubation period of any infected individual and to apply the vaccines for all observed infections to the entire population at that time. Such a delayed vaccination campaign allows emergence of new infections, possibly with new types of pathogens, before letting sick individuals infect others. This approach minimizes the number of necessary vaccination campaigns, each with all vaccines necessary to stop already started epidemics.

Inspired by this idea, we use the following efficient acknowledgment scheme. As the destination receives messages, it waits until the closest period change time (x_d) of any of the received messages. At that time, the destination broadcasts an acknowledgment of all received messages so far. Hence, the destination broadcasts acknowledgments relatively infrequently, proportionally to a substantial fraction of the t_d, which is assumed large. Even though acknowledgments of some messages are delayed, spraying of any received messages after the delivery time is suppressed.

It is clear that Type-II acknowledgment results in better performance than Type-I acknowledgment in terms of the total number of copies used per message. 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 Model and Results

To evaluate our multiperiod algorithm, we have developed a discrete event-driven simulator in Java. We performed extensive simulations with different parameters that may affect the performance of the proposed algorithm.

First of all, we compare the results of simulations with the analytical results that we have obtained in the previous section. Moreover, we also look at the effects of two different mobility models on the results.

We deployed $M = 100$ mobile nodes (including the sink) onto a torus of size $300 \times 300\ \mathrm{m}^2$. 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 (note that these parameters generate a sparse DTN, which is the most common case in practice). The movements of nodes are decided according to two different mobility models [30].

- Random Walk Model:
  The speed of a node is randomly selected from the range [4, 13] m/s, and its direction is also randomly chosen. Then, each node goes in the selected random direction with the selected speed until the epoch lasts. Each epoch's duration is again randomly selected from the range [8, 15] s.
- Random Waypoint Model:
  First, a new destination inside the network area is chosen randomly. Then, the node moves toward that destination with a randomly selected speed from the range [4, 13] m/s.

TABLE I
OPTIMUM L_i COPY COUNTS THAT MINIMIZE THE AVERAGE NUMBER OF COPIES WHILE PRESERVING THE DESIRED PROBABILITY OF DELIVERY

	Random Walk			Random Waypoint		
t_d	L_{min} in 1p	2p Optimum L_i's	3p Optimum L_i's	L_{min} in 1p	2p Optimum L_i's	3p Optimum L_i's
200	12	7,22	6,12,27	9	5,16	4,8,20
250	9	5,15	5,9,19	7	4,13	3,6,15
300	8	5,14	4,8,18	6	3,11	3,6,14
400	6	4,11	3,6,14	5	3,10	2,4,12
500	5	3,9	2,4,11	4	2,8	2,4,10
600	4	2,7	2,4,9	3	2,5	1,2,6
700	4	2,8	2,4,10	3	2,6	1,2,7
800	3	2,5	1,2,6	2	2,7	1,3,10
900	3	2,6	1,2,7	2	1,4	1,2,5

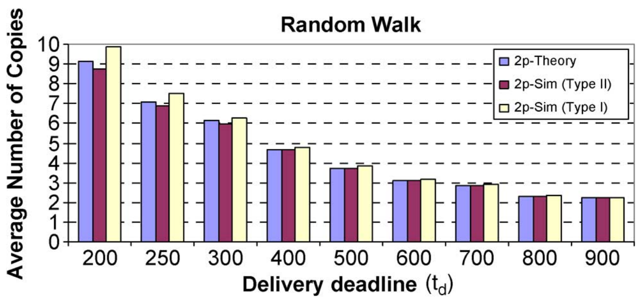


Fig. 5. The comparison of the average number of copies obtained via analysis and simulation for the two-period case when random walk model is used.

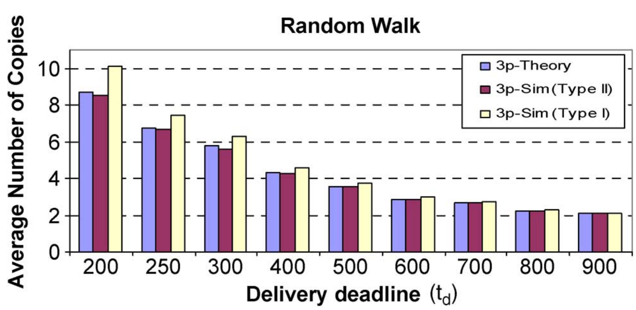


Fig. 6. The comparison of the average number of copies obtained via analysis and simulation for the three-period case when random walk model is used.

When nodes move according to the above models with given parameters, the value of EM in the former and latter becomes 480 and 350 s, respectively (we both computed these values from the given parameters and validated the results by simulations).

Assuming[3] that the desired p_d by the given deadline (t_d) is 0.99, first we have found the optimum copy counts for both two-period (2p) and three-period (3p) cases using Algorithms 1 and 2. Table I shows the values of these optimum L_i's for different t_d values as well as the minimum L value that achieves the desired p_d in the single-period (1p) spray-and-wait algorithm. Clearly, as the deadline decreases, $L_{\min}$ (minimum L achieving p_d by t_d) in 1p increases because more copies are needed to meet the desired p_d by the deadline. Such an increase is also observed for L_i values used in both 2p and 3p algorithms. It is also important to remark that the optimum L_i values are different for random walk and random waypoint models because the EM values generated in these two different settings are different. Although we mentioned in Section II that our algorithms are designed for the environments in which the deadline is not so tight with respect to EM value, in the simulations, we also test our algorithms with tight deadlines (such as 200 and 250 s)[4] to see how they perform in these cases. Moreover, for the optimum L_i values of three periods, we also ran Algorithms 3 and 4 over the result that we obtained with Algorithm 1 and observed that the results closely match the optimum L_i values that we obtained using Algorithm 2.

We started by computing x_{d1}, x_{d2} and the optimum L_i values from theory. Then, we performed simulations to find the average copy count used per message when these computed values are used. We have generated messages from randomly selected nodes to the sink node whose initial location was also chosen randomly. Furthermore, we used binary spraying while distributing the allowed copy counts in each period. All results are the average of 2000 runs.

In Figs. 5 and 6, we show the average copy counts obtained when the optimum L_i values are used in 2p and 3p versions of our algorithm and when the predefined random walk model is used. Our analysis defines the cost function as the average copy counts used per message at the exact delivery time and computes the optimum L_i values that minimize this cost function. Hence, to compare theory with simulations, we obtained the average copy counts in simulations using Type-II acknowledgments. However, we also include the average copy counts obtained in simulations when Type-I acknowledgment is used. From the results in both figures, we observe that analysis results are very close to Type-II results, but as the deadline gets tight, they become an upper bound for Type-II results. This is because for the smaller values of t_d, the number of copies sprayed to the network increases (optimum L_i values in 2p and 3p are large due to large $L_{\min}$ in 1p) so that spraying period takes longer. Besides, this also increases the difference between the average copy counts needed when Type-I and Type-II acknowledgments are used because as L_i values gets larger, more nodes carrying message copies need to be acknowledged about the delivery when Type-I acknowledgment is used.

We also compared the results when random waypoint mobility model is used. Figs. 7 and 8 show the comparison of average copy counts obtained in simulations with those computed analytically. The conclusions are similar to those made above for the random walk model, even though $L_{\min}$ values are different from those used in the random walk model since the settings in this model generate an EM of 350 s. This shows that our analysis holds for different mobility models. It only relies on the EM, the average intermeeting time between nodes for the applied mobility model.

To compare the performance of the proposed algorithms with the single-period (1p) spraying algorithm (which is a special case of our algorithm), we first compare the average number of

[3]We have selected a high desired delivery probability because it is the most likely case in real applications. However, we also look at the effects of different p_d values in later simulations.

[4]These values can surely be considered as tight deadlines because note that direct delivery $(L = 1)$ in single spraying can achieve $p_d = 0.99$ at 2210 and 1610 s in the given random walk and random waypoint models, respectively.

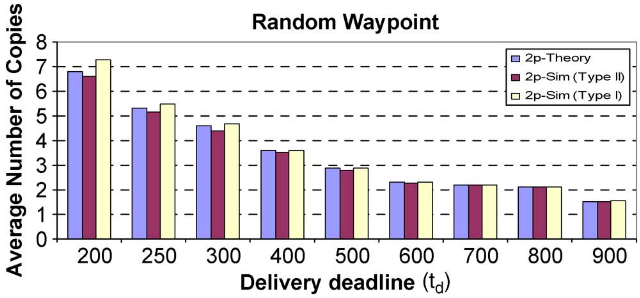


Fig. 7. The comparison of the average number of copies obtained via analysis and simulation for the two-period case when random waypoint model is used.

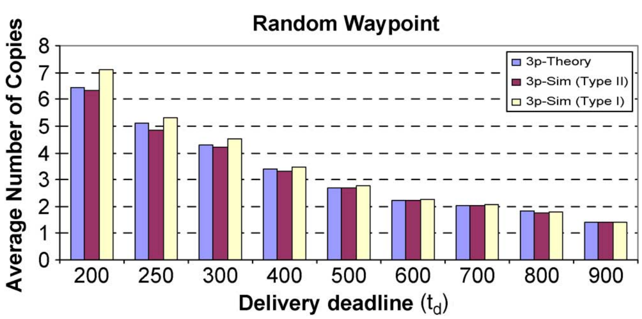


Fig. 8. The comparison of the average number of copies obtained via analysis and simulation for the three-period case when random waypoint model is used.

copies used in both algorithms when different types of acknowledgment mechanisms are used.

In Table II, we present the average copy counts used in three compared algorithms when random walk model is used (we did not include the results when random waypoint model is used because they are similar to the results presented here). From the table, we observe that in both acknowledgment types, 3p algorithm uses fewer copies on average than either 2p or 1p spraying algorithm does. However, when Type-I acknowledgment is used, the saving in the number of copies obtained by 3p algorithm decreases. Moreover, in some cases ($t_d = 200$ s), its performance becomes worse than 2p algorithm. This is because when the deadline gets tight, the number of copies that are sprayed to the network increases so that the number of nodes carrying the message copies increases and the duration of epidemic like acknowledgment is longer. Consequently, more redundant copies are sprayed by the nodes having message copies before they are informed about the delivery. Moreover, we also notice that using the proposed algorithms even with Type-I acknowledgment results in lower average copies used than when using the single-period spraying algorithm with Type-II acknowledgment. It should also be noted that in single-period 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 single-period spraying that does all spraying at the beginning, there is nonzero chance that the message will be delivered before all copies are made.

To further compare the performance of the proposed algorithms to the single-period spraying algorithm, we have measured some additional metrics. Figs. 9 and 10 show the comparison of average message delivery delay and the average time of

TABLE II
AVERAGE NUMBER OF COPIES USED IN SINGLE- (1P), TWO-PERIOD (2P), AND THREE-PERIOD (3P) SPRAYING ALGORITHMS WITH DIFFERENT ACKNOWLEDGMENT TYPES AND DEADLINES

		Type I			Type II		
t_d	L_{min}	1p	2p	3p	1p	2p	3p
200	12	11.61	9.89	10.12	10.92	8.77	8.51
250	9	8.79	7.50	7.44	8.52	6.88	6.65
300	8	7.80	6.28	6.28	7.58	5.94	5.62
400	6	5.87	4.78	4.55	5.78	4.64	4.28
500	5	4.91	3.84	3.72	4.86	3.73	3.54
600	4	3.96	3.18	3.02	3.93	3.10	2.85
700	4	3.96	2.89	2.74	3.93	2.83	2.66
800	3	2.97	2.33	2.31	2.95	2.31	2.24
900	3	2.97	2.24	2.09	2.96	2.23	2.07

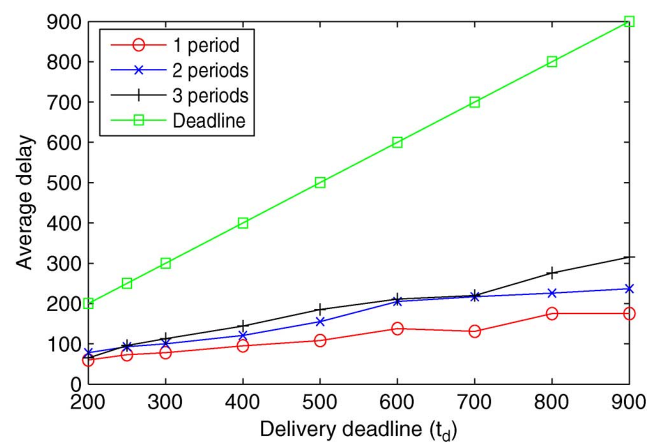


Fig. 9. The comparison of the average delay for the single-period and multiple-period algorithms (random walk model).

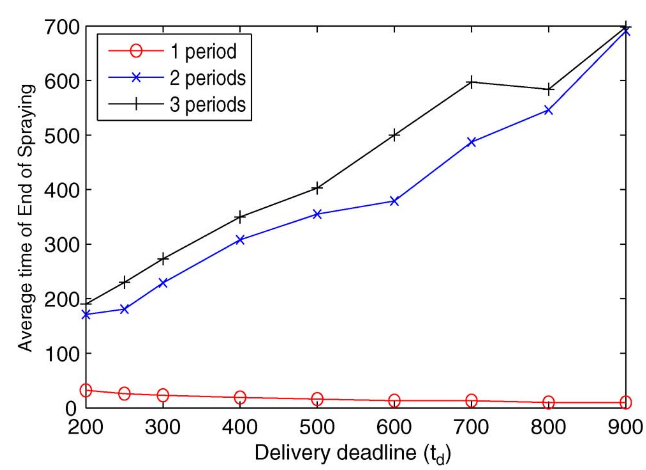


Fig. 10. The comparison of average end of spraying times in the single-period and multiple-period spraying algorithms (random walk model).

spraying completion[5] (time by which the last copy is sprayed) in these algorithms, respectively. Inspecting these two graphs, we observe that the proposed 2p and 3p algorithms incur higher average delay than 1p algorithm, but they achieve the same

[5]The values in Fig. 10 are computed over cases in which the message is delivered after all potential copies are sprayed.

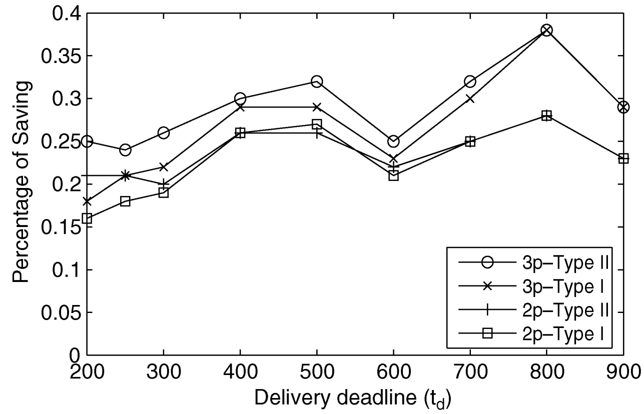


Fig. 11. The percentage of savings achieved by the proposed algorithms with two different acknowledgment schemes (random waypoint model).

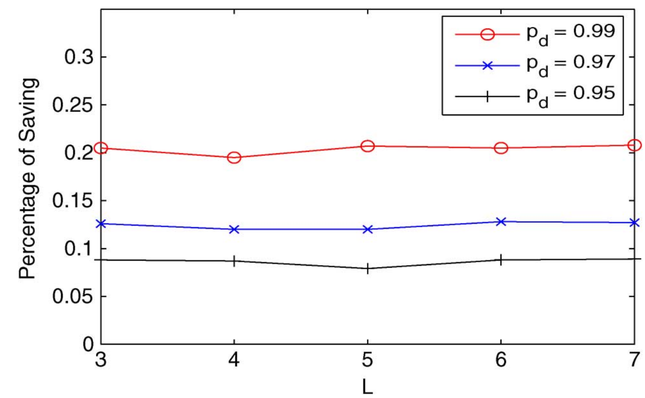


Fig. 12. The percentage of savings achieved by 2p Type-II algorithm with three different p_d values (random waypoint model).

delivery probability[6] before the deadline compared to the 1p spraying algorithm. Moreover, since the proposed algorithms postpone the spraying of all copies to later times, they finish spraying later than the single-period spray-and-wait algorithm does. This results in lower memory usage averaged over execution time of our algorithm when compared to such usage incurred by the single-period spraying algorithm.

We also computed the percentage of the savings achieved in the number of copy counts with the proposed multiperiod algorithms. Fig. 11 charts the fraction $(L - L_{\rm avg})/L$ with the given t_d. Here, L is the average copy count used in single-period spraying, and $\rm L_{avg}$ is the average copy count achieved in the multiperiod spraying algorithm. This time, we present the results when random waypoint model is used (the results with random walk model are similar). From the results shown in Fig. 11, we observe that 3p algorithm provides higher savings than 2p algorithm. Moreover, it is clear that the savings with Type-II acknowledgment are higher than the savings with Type-I acknowledgment in both 2p and 3p algorithms. The difference between the savings of Type-I and Type-II acknowledgments gets smaller as the deadline increases. This is because larger t_d decreases the number of copy counts sprayed to the network, resulting in acknowledgments reaching all nodes carrying message copies earlier. On the other hand, we also observe fluctuations even in the savings of a single algorithm with different delivery deadlines. This is because for some consecutive t_d values (i.e., t_d = 600, 700, 800 s), $L_{\min}$ value in 1p algorithm that achieves the desired p_d is the same (i.e., $L_{\min} = 3$) while L_i values in multiperiod algorithms are different. In these cases, multiperiod algorithms take the advantage of spraying in multiple periods and delay the spraying further when the deadline is larger (for example, in 2p algorithm, when $t_d = 600$ s, then $x_{d1} = 400$ s and the optimum $(L_1, L_2) = (2, 5)$, but when $t_d = 700$ s, then $x_{d1} = 525$ s and the optimum $(L_1, L_2) = (2, 6)$). Hence, multiperiod algorithms can provide more saving over a single-period algorithm in such cases.

We also looked at the effects of the desired p_d on savings achieved by the proposed algorithms. As an example, we plotted the percentage of savings obtained in 2p Type-II algorithm with three different p_d values in Fig. 12. Here, we performed simulations in a different way to show also the flat behavior of percentage of savings with respect to L (instead of t_d). With the given L and p_d values, we first found the minimum t_d value that achieves the given p_d (in random waypoint model), and then obtained the savings provided by 2p Type-II algorithm (when optimum L_i values are used) at that t_d value while maintaining the given p_d.

From Fig. 12, we first observe that the savings are almost the same when plotted according to the L values, where t_d is the minimum time that spraying of L copies achieves the given p_d (proving this property analytically is the subject of our future work). Additionally, we observe that as the given p_d value decreases, the savings provided by multiperiod algorithm decreases. This is because as p_d decreases, minimum t_d value achieving p_d with the given L decreases and the cdf of delivery probability gets more vertical around the t_d value. Because of these two reasons, the chance of saving in multiperiod algorithm decreases with lower values of the desired p_d.

In the above simulations, we always assumed a constant number of nodes $(M = 100)$ in the network. However, the value of M affects the performance of the algorithm as well. For example, in Fig. 13 we plot the simulation and analysis results in random walk model for 2p algorithm with three different M values (where $p_d = 0.99$ and $t_d = 300$ s). It is clear that as M increases, the difference between 2p-Sim (Type II) and analysis gets smaller. This is the result of fast spraying[7] with increasing M. Moreover, the difference between 2p-Sim (Type I) and 2p-Sim (Type II) results decreases because larger M values enable faster acknowledgment process.

In addition to the evaluation of the proposed protocol with random mobility models, we have also looked at its performance on real DTN traces. From the several data sets released so far,

[6]In simulations, we assume that collisions or collision avoidance do not impact message delivery. Indeed, they can only force meeting nodes to communicate sequentially, delaying some pairwise node communications. Yet, the average required communication time (about 0.1 s with 1-Mb/s bandwidth and 100-kb packets) is small compared to the average meeting time of two nodes (1.72 s in random walk setting). Moreover, meeting of four or more nodes is very unlikely (below 1% in our setting). Thus, it is unlikely (below 0.05% in our setting) that a communication delay due to collision or collision avoidance will exceed meeting time, justifying our assumption.

As expected, in simulations, all three algorithms achieve the desired p_d by the deadline (for the sake of brevity, the relevant plot is omitted here).

[7]It should be noted that EM does not change with increasing M. Only the rate of meeting with new nodes increases, which results in the fast spraying of messages.

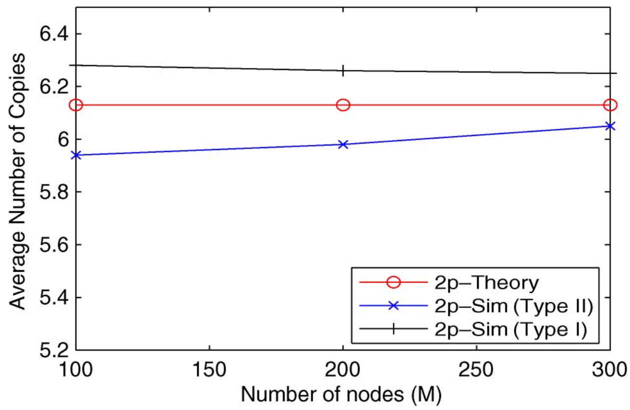


Fig. 13. The effect of number of nodes on the difference between the analysis and simulations results.

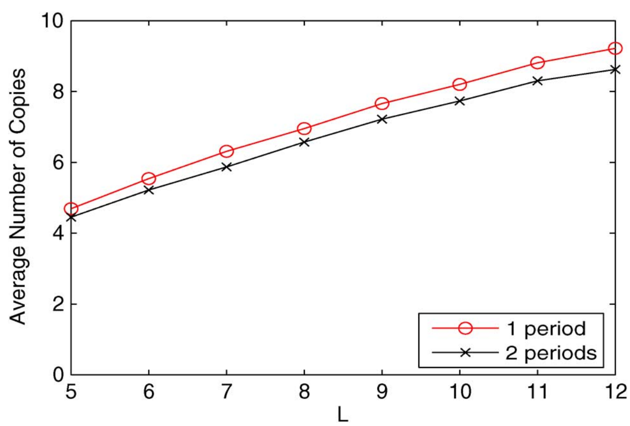


Fig. 14. The average number of copies used per message in the simulations of real traces from RollerNet.

we have selected RollerNet [33] traces thanks to its easy usability (for example, all the meetings between nodes are mutually recorded). RollerNet traces include the opportunistic sightings of Bluetooth devices by groups of rollerbladers carrying iMotes in the roller tours in Paris, France. Since the nodes are not identical, the generated intermeeting times between pairs of nodes vary significantly. Although our protocol is not designed for networks with heterogeneous intermeeting times between nodes, we simply applied our multiperiod spraying idea using the results from our analysis and left the design of an algorithm specifically for heterogeneous networks as a future work.

The RollerNet traces starts at 1156084064s and ends at 1156094040s. In each 10 s starting from the beginning (until the time after which the last message will not have enough time to be delivered), we have generated a message from a random source to a random destination in the network. In single-period routing, for each L (number of copies allowed), we found the delivery times of each message and also the time (we call it discovered t_d) at which 99% (p_d) of all messages are delivered since their generations at the source nodes. Then, we ran the two period routing on the same traces with the same set of messages. In the first period, we allowed the spraying of $L/2$ copies (which is the most frequent case according to the results of our analysis). In the second period, we tried different copy counts and found the necessary copy count that achieves the same delivery rate of all messages by the discovered t_d. Since the intermeeting times between nodes and also the delivery times of messages are different from each other, we computed the start of second period individually for each message. That is, for each message, we used its delivery time in the single-period routing with L messages as the message's own t_d and computed x_d accordingly. In Fig. 14, we show the average copy counts used per message in 1p and 2p spraying algorithms. Clearly, multiperiod spraying idea can reduce the average copy count used in real DTN traces even when the frequencies of node meetings show heterogeneous behavior. The savings in this case are in the range of 6%–8%. However, we believe that a more careful design of multiperiod idea can increase the savings even further. The design of a multiperiod spraying-based routing algorithm for heterogeneous networks will be the subject of our future work.

V. Conclusion and Future Work

In this paper, we introduce a general multiperiod spraying algorithm for DTNs that distributes the message copies depending on the remaining time to delivery deadline, and then, using formal analysis and simulations, we evaluate its performance. We first show analytically how to partition time until deadline in a single-period spraying algorithm into two and three separate periods, each period consisting of a spraying phase followed by the wait phase. Then, we present a generalization of this approach to a larger number of periods to reduce the cost even further. Finally, we discuss the results of simulations of our algorithm confirming that the average number of copies used by our algorithm is smaller than the average number of copies used by the single-period spraying algorithm, while its delivery rate by the deadline matches the performance of the latter.

In the future work, we will investigate how more realistic radio links and mobility models affect our algorithm. Moreover, we also plan to update the proposed protocol for networks in which node meeting behavior varies between nodes.

References

[1] "Delay tolerant networking research group," [Online]. Available: http://www.dtnrg.org

[2] 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 *Proc. ACM ASPLOS*, 2002, pp. 96–107.

[3] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in zebranet," in *Proc. ACM SenSys*, 2004, pp. 227–238.

[4] M. Motani, V. Srinivasan, and P. Nuggehalli, "PeopleNet: Engineering a wireless virtual social network," in *Proc. ACM Mobicom*, Aug. 2005, pp. 243–257.

[5] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in *Proc. ACM WUWNet*, Sep. 2006, pp. 17–24.

[6] A. Maffei, K. Fall, and D. Chayes, "Ocean instrument Internet," in *Proc. AGU Ocean Sci. Conf.*, Feb 2006.

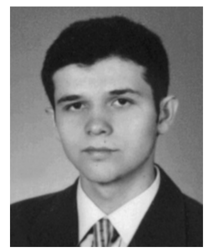
[7] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru internet environments," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 1849–1862.

[8] "Disruption tolerant networking," [Online]. Available: http://www.darpa.mil/ato/solicit/DTN/

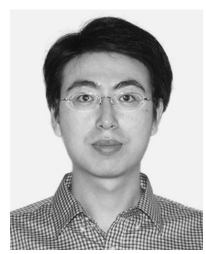
[9] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," *IEEE Commun. Surveys Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[10] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, 2007, pp. 373–384.

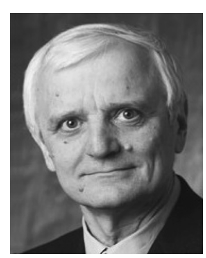
[11] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 229–236.
[12] J. Widmer and J. Boudec, "Network coding for efficient communication in extreme networks," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 284–291.
[13] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, 2000.
[14] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, no. 10, pp. 2859–2891, 2007.
[15] A. Jindal and K. Psounis, "Performance analysis of epidemic routing under contention," in *Proc. Workshop DTMN/IWCMC*, 2006, pp. 539–544.
[16] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.
[17] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 145–158.
[18] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay tolerant networks," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 237–243.
[19] T. Small and Z. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 260–267.
[20] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. MobiHoc*, May 2004, pp. 187–198.
[21] K. Harras, K. Almeroth, and E. Belding-Royer, "Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks," in *Proc. IFIP Netw.*, Waterloo, Canada, May 2005, pp. 1180–1192.
[22] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption- tolerant networks," in *Proc. IEEE INFOCOM*, Apr. 2006.
[23] C. Mascolo and M. Musolesi, "SCAR: Context-aware adaptive routing in delay tolerant mobile sensor networks," in *Proc. Int. Conf. Wireless Commun. Mobile Comput.*, 2006, pp. 533–538.
[24] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 252–259.
[25] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proc. WoWMoM*, 2005, pp. 183–189.
[26] Z. Li, L. Sun, and E. Ifeachor, "Adaptive multi-copy routing for intermittently connected mobile ad hoc networks," in *Proc. IEEE GLOBECOM*, San Francisco, CA, 2006, p. WSN10-5.
[27] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002, Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications.
[28] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Toward realistic mobility models for mobile ad hoc networks," in *Proc. ACM MobiCom*, San Diego, CA, Sep. 2003, pp. 217–229.
[29] A. Jindal and K. Psounis, "Fundamental mobility properties for realistic performance analysis of intermittently connected mobile networks," in *Proc. PerCom Workshops*, 2007, pp. 59–64.
[30] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proc. ACM MobiHoc*, 2006, pp. 49–60.
[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, Feb. 2008.
[32] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.
[33] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on DTN routing," in *Proc. IEEE INFOCOM*, 2009, pp. 1116–1124.



Eyuphan Bulut (M'08) received the B.S. and M.S. degrees in computer engineering from Bilkent University, Ankara, Turkey, in 2005 and 2007, respectively. He is currently pursuing the Ph.D. degree in the Computer Science Department of Rensselaer Polytechnic Institute (RPI), Troy, NY. His interests include design of protocols for wireless sensor and ad hoc networks such as routing protocols for delay-tolerant networks.



Zijian Wang received the B.S. degree in electronics and information engineering from Beihang University, Beijing, China, in 2003. From 2007 to 2009, he was a visiting Ph.D. student at Rensselaer Polytechnic Institute (RPI), Troy, NY. Currently, he is a Ph.D. student in the School of Electronics and Information Engineering, Beihang University. His research interests are wireless ad hoc/sensor networking energy-efficient protocols, including topology control, multipath routing, target tracking, location service, and service discovery.



Boleslaw Karol Szymanski (M'82–F'99) received the Ph.D. degree in computer science from the National Academy of Sciences, Warsaw, Poland, in 1976.

He is the Claire and Roland Schmitt Distinguished Professor of Computer Science and the Founding Director of the Center for Pervasive Computing and Networking at Rensselaer Polytechnic Institute (RPI), Troy, NY. He is an author and coauthor of over 300 publications and an editor of five books. He is also an Editor-in-Chief of *Scientific Programming*. His interests focus on parallel and distributed computing and networking.

Prof. Szymanski is a Member of the Association for Computing Machinery (ACM), for which he was a National Lecturer.