Routing on Delay Tolerant Sensor Networks

Michael Keane[1], Evangelos Kranakis[2], Danny Krizanc[2], and Lata Narayanan[3]

[1] Department of Mathematics and Computer Science, Wesleyan University, Middletown CT 06459, USA.
[2] School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada. Supported by NSERC and MITACS grants.
[3] Department of Computer Science and Software Engineering, Concordia University, Montreal, QC, H3G 1M8, Canada. Supported by NSERC grant.

Abstract. Delay (or disruption) tolerant sensor networks may be modeled as Markovian evolving graphs [1]. We present experimental evidence showing that considering multiple (possibly not shortest) paths instead of one fixed (greedy) path can decrease the expected time to deliver a packet on such a network by as much as 65 per cent depending on the probability that an edge exists in a given time interval. We provide theoretical justification for this result by studying a special case of the Markovian evolving grid graph. We analyze a natural algorithm for routing on such networks and show that it is possible to improve the expected time of delivery by up to a factor of two depending upon the probability of an edge being up during a time step and the relative positions of the source and destination. Furthermore we show that this is optimal, i.e., no other algorithm can achieve a better expected running time. As an aside, our results give high probability bounds for Knuth's toilet paper problem [11].

Key Words and Phrases: Ad hoc networks, Delay tolerant, Disruption tolerant, Evolving graphs, Routing, Sensors.

1 Introduction

Delay (or disruption) tolerant networks (DTNs) are characterized by their experiencing frequent, variable and long-duration periods of intermittent connectivity. Examples of such networks include satellite networks, mobile radio networks, energy-constrained sensor networks, etc. Such networks may be captured by Ferreira's *evolving graph* model [6] in which a network is modelled as a sequence of subgraphs of a fixed graph.

Our study is specifically motivated by the case of sensor networks with limited battery power for communication. A standard technique for power saving in such networks is to put sensors to sleep for long periods. Upon waking up the sensor searches among its neighboring sensors for ones that are awake and with which it can establish a communication link. (We assume that the sensors are "location-aware" in that they know their location in some co-ordinate system as well as the co-ordinates of their neighbors.) After transferring what messages it has to send

along these links the sensor goes back to sleep. The time between waking periods is chosen randomly. A network communicating under these circumstances is modelled well by a special class of geometric Markovian evolving graphs [1]. We assume the graph evolves in time steps. At each time step each potential edge (between neighboring sensors) is up with some fixed probability p, independent of the other potential edges. We are interested in the question of routing packets on such networks.

Experience shows that for sufficiently dense sensor networks (i.e., above the connectivity threshold) the geographic greedy path (i.e., each node chooses the edge that minimizes the geographic distance to the destination) generally minimizes the routing delay for routing between any given source and destination [10]. This suggests the following algorithm for routing in the sensor networks described above. A node attempting to deliver a packet to a given destination, upon waking up checks to see if the neighboring node that is closest to the destination is awake. If yes, it passes the packet to this node and goes to sleep. Otherwise it goes to sleep and waits for the next time it is awake. We call this the *greedy* strategy. If each edge is up with probability p and the greedy path (exists and) is of length l then clearly the expected number of steps required to deliver the packet to its destination is l/p. The question arises if one can do better by considering other (non-greedy) paths.

We provide experimental evidence that the following natural algorithm can improve the expected delivery time by as much as 65 per cent depending upon the value of p and the relative positions of the source and destination. Fix a value k. Upon waking up a node sorts its neighbors by their distance to the destination. If there is at least one neighbor among the top k that is awake and it is closer to the destination, it passes the packet to the best among them and goes to sleep. Otherwise it goes to sleep and waits for the next time it is awake. We call this the k-greedy strategy. (Clearly 1-greedy is the greedy strategy we defined above.) Our experiments are performed on sensor networks consisting of 1000 randomly distributed sensors on a unit square with radius of communication sufficiently above the threshold of connectivity to ensure the greedy path correctly delivers a packet with high probability. They show that as one increases k the improvement achieved increases until a maximum at about $k = 8$ but that most of the improvement (more than 60 percent) comes already from $k = 2$. We also observed a noticeable drop in improvement when considering packets whose greedy path is close to the perimeter of the square versus those in the central region of the square.

In order to study this problem analytically we consider the case of routing on a Markovian $n \times n$ grid where on each time step each edge of the grid exists (i.e., is up) with probability p independent of the other edges. Analogous to the greedy algorithm above, imagine that we fix a shortest path and forward the packet along that path whenever edges are available. Without loss of generality consider the case of a packet starting at node (i, j) and destined for $(0, 0)$ where $i \geq j$. In this case the expected time for delivery will be $(i + j)/p$.

We consider the following algorithm analogous to the 2-greedy strategy above. On a given step, if $i > 0$ and $j > 0$ (recall that $i \geq j$), then if the edge to $(i-1, j)$ is up, take it, i.e., pass the packet across the edge. If the edge to $(i-1, j)$ is down but the edge to $(i, j-1)$ is up, take it. Otherwise wait for the next time step. If i or j is 0 then wait for the unique edge on the shortest path to $(0, 0)$ to be up.

Let $r = 1 - (1-p)^2$ be the probability that at least one of the edges to $(i-1, j)$ or $(i, j-1)$ is up. We show that the time required to deliver a packet by this algorithm is

$$\max\left\{ \frac{i+j}{r}, \frac{\max(i,j)}{p} \right\} + O\left(\sqrt{(i+j)\log(i+j)} \right)$$

with high probability, i.e., with probability at least $1 - \frac{1}{i+j}$. Furthermore we show that this is optimal up to lower order terms. Our results indicate that depending on p, packets following paths closer to the center of the grid (e.g., along the diagonal) can achieve as much as a factor of 2 improvement in their expected delivery time over the simple greedy strategy. For packets closer to the perimeter of the grid, the improvement drops until we reach the case of packets whose shortest path is a straight line where no improvement is possible. Simulations confirm that the behavior of the grid algorithm and the 2-greedy strategy on the unit square are qualitatively very similar.

Finally we note that the analytic problem we are studying is a variant of Knuth's toilet paper problem [11] and that our results can be used to show high probability results for that problem as well.

1.1 Related Work

Since traditional routing protocols make the assumption that the network graph is connected they may fail to route packets if a path from the source to the destination does not exist at all times. To deal with the problem of intermittent connectivity several approaches have been proposed by the networking community. For the most part, these approaches either assume that the graph evolves in a predictable (even repetitive) fashion or that the connectivity of the network is the result of a predictable process such as the (possibly random) movement of the nodes in a given region. Examples of such network include LEO satellite networks where the edges of the graph change in a predictable fashion, vehicular networks where nodes (cars) following predictable paths and MANETs, sometimes modeled as networks of mobile nodes that encounter each other randomly inside a bounded region. Experimental studies along these lines include [3, 4, 12, 13, 15, 16, 18, 19]. In the setting where the evolution of the graph is known in advance, the problem can be approached using extensions to standard shortest path calculations. In the random movement setting, most approaches may be described as a variant of controlled flooding, where an effort is made to not overload the network.

The evolving graphs model of delay tolerant networks was introduced by Ferreira [6]. In a series of papers, Ferreira and co-authors experimentally evaluated a variety of standard protocols as well as novel approaches to routing on such graphs [7, 8]. In [2] it is shown that when the graph sequence is given in advance, the "best" path under a number of natural metrics can be computed in polynomial time. Several related papers examine properties of the evolving graphs model itself. [9] looks at connectivity in evolving graphs with geometric properties and [1] examines the cover time of random walks in this model.

Related to the analysis of our algorithm is the following problem (generally referred to as the *toilet paper problem*), first described and solved by Knuth [11]. A toilet stall contains two rolls of toilet paper of n sheets each. The stall is used by people of two types: *big choosers* and *little choosers*. They arrive to use the toilet randomly and independently, the former with probability p and the latter with probability $1 - p$. Big (respectively, little) choosers select exactly one sheet of paper from the roll with the most (respectively, least) number of sheets. What is the expected number of toilet sheets remaining just after one of the two rolls has emptied, defined to be the residue $R_n(p)$?

Knuth [11] uses combinatorial techniques to prove that for fixed p and r, which satisfy the condition $4p(1 - p) < r < 1$, we have that

$$
E[R_n(p)] = \begin{cases}
\frac{p}{2p-1} + O(r^n) & \text{if } p > 1/2 \\[2mm]
2\sqrt{\frac{n}{\pi}} - \frac{1}{4}\frac{1}{\sqrt{n\pi}} + O(n^{-3/2}) & \text{if } p = 1/2 \\[2mm]
\frac{1-2p}{1-p}n + \frac{p}{1-2p} + O(r^n) & \text{if } p < 1/2
\end{cases}
$$

as $n \to \infty$ where the constants implied by the O notation depend on p, r but not on n. The case $p = 1/2$ is identical to Banach's match box problem given in the Scottish book [14]. For a related generalization of this problem using a martingale approach we refer the reader to [17] and [5].

1.2 Outline of the paper

In the next section we present the results of our simulation of the k-greedy strategy for nodes distributed randomly on the unit square. In Section 3 we prove matching upper and lower bounds on the performance of the 2-greedy strategy on the Markovian grid. Due to space limitations the proofs will be appear in the full version of the paper.

2 Experimental Results for the k-greedy Strategy

In this section we present experimental results concerning a natural strategy for routing on a special class of geometric Markovian evolving graphs. For this class of graphs, the node set is a set of points on the plane. An edge potentially exists between two nodes if their Euclidean distance is less than a given distance

R. The graph evolves in time steps. During each time step each potential edge exists with probability p, independently of the other edges. We assume the nodes are "location aware", i.e., their is a fixed co-ordinate system agreed upon by all nodes and the nodes know their position in this system as well as the positions of their potential neighbors.

We are interested in studying the behavior of the following natural routing strategy for such graphs. Fix a value k. Consider a packet with destination t currently located at a given node other than t. On each step, the current node calculates the distance of all its neighbors to the destination and sorts them in increasing order. If among the first k potential neighbors their exists a node whose distance to t is less than the distance from the current node to t and the edge to that node exists, it forwards the packet along the existing edge to the node that is closest to t. If no such node exists, it waits until the next step. We call this strategy the k-greedy strategy and will refer to 1-greedy as simply the greedy strategy.

Experience has shown that for randomly distributed points on a unit square, if R is chosen sufficiently large, then the greedy strategy results in a path that generally minimizes the number of routing steps required to deliver the packet for most source-destination pairs [10]. Intuition suggests that by considering more paths, e.g., by increasing k in the strategy above, one might improve the performance of the algorithm. We ran a series of experiments to test this hypothesis. A standard experiment involved randomly (uniform in both co-ordinates) distributing 1000 points in a unit square and choosing $R = .1$. Experiments were run for a variety of source-destination pairs but we found there was little loss of generality by fixing the destination to be $(0,0)$. All the results reported below are based upon the average of 1000 runs with a single randomly chosen point set and a fixed parameter set of p, k and source node.
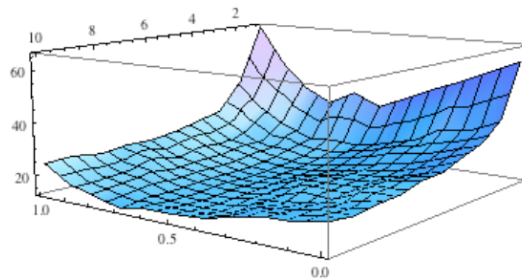
Fig. 1. Effect of k. $p = .2$ and destination is $(0,0)$; k varies from 1 to 10; source is $(x, 1-x^2)$ where x varies from 0 to 1.0 by .1; z-axis gives the number of steps required.

The effect of k Figure 1 shows the effect of the parameter k on the strategy. This example is for the case where p is fixed at .2 and we consider an equidistant set

of source points equidistant from $(0,0)$ on the curve $y = 1 - x^2$ (with x varying from 0 to 1 by .1). Similar results are obtained for other values of p (not to close to 0 or 1) and choices of source points. The general trend for each source point is that the number of steps required to reach the destination drops as k increases until about $k = 8$ at which point the improvement levels off. Over all values of $k > 1$ and all points the average improvement over straight greedy ($k = 1$) is 49.9 per cent. For $k = 8$ the average improvement is 61.4 per cent with a maximum of 65.6 per cent and a minimum of 54.3 per cent. We note that the majority of the observed savings over greedy occurs by $k = 3$ with 61 per cent explained by $k = 2$ and 79 per cent by $k = 3$.
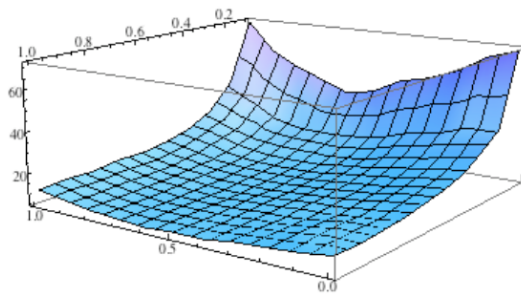
Fig. 2. Effect of p. $k = 2$ and destination is (0,0); p varies from .1 to 1.0 by .1 along the top axis; source is $(x, 1 - x^2)$ where x varies from 0. to 1.0 by .1; z-axis gives the number of steps required.

The effect of p Figure 2 shows the effect of varying p. This example is for the case where $k = 2$, for p ranging from .1 to 1 by .1 intervals and we again consider a set of equidistant source points. Similar results are obtained for other values of k and choices of source points. As expected the number of steps required increases significantly as p is decreased. At the same time the average savings over greedy increases from 8.4 per cent to 40.0 per cent as p varies from .9 down to .1. A slight effect due to the position of the source is visible in Figure 2 in that nodes closer to the edge of the square require longer to reach the destination even though they are the same Euclidean distance away as the nodes in the center. (Along each fixed p, the curve is slightly concave when moving from x equals 0 to 1.)

The effect of source position Figure 3 plots for $p = .2$ and $k = 2$ the time required for a uniform grid of points to route to the destination $(0, 0)$. Similar results are obtained for other values of k and p. While it is clear that the distance to the destination is a major factor in the number of steps required to reach it, there appears to be a slight increase in the time for points with source near the edge
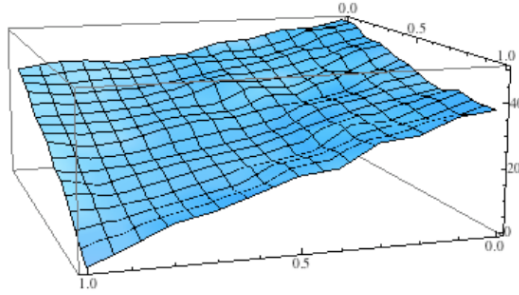
Fig. 3. Effect of source position. $k = 2$ and destination is $(0,0)$; x and y of source vary from 0 to 1.0 by .1; z-axis gives the number of steps required.

of the square as opposed in the center. (This effect is perhaps more noticeable in figure 2.)

3 Routing on the Grid

In an attempt to explain the observations made in section 2, in this section we study greedy routing on a special case of the Markovian evolving grid. (This might be considered a coarse approximation to the above where one divides the unit square into a uniform grid of subsquares and identifies the points inside each subsquare.) The node set of the $n \times n$ grid is $\{(i,j)|0 \le i,j < n\}$ and the potential edge set is $\{\{(i,j),(k,l)\}||i - k| + |j - l| = 1\}$. On each time step each edge exists with probability p independently of all other edges.

Analogous to the greedy strategy above one might consider a routing algorithm that fixes a shortest path from source to destination and then waits for that edge to be up in order to forward the packet. Without loss of generality consider a packet starting at (i,j) and destined for $(0,0)$. It can expect to take $\frac{i+j}{p}$ steps to reach its destination under this scheme.

Clearly the above does not take advantage of the many possible shortest paths between most source-destination pairs. Again, consider the case of a packet starting at (i,j) destined for $(0,0)$ where neither $i = 0$ nor $j = 0$. In this case there are two possible shortest paths, one starting with the edge to $(i-1,j)$ and one to $(i,j-1)$. In analogy to the 2-greedy strategy above, one might consider taking either one of these *distance-reducing* edges if they exist. Let $r = 1-(1-p)^2$ be the probability that at least one of the two distance-reducing edges exists. Assuming that $0 < p < 1$, we see that $1/2p < 1/r < 1/p$, i.e., the waiting time between moves in this case is always less than waiting for a specific edge and maybe up to 50 per cent less depending on the value of p. A question arises as to what to do if both edges exist. For instance, one might consider flipping a coin in this case. In fact, we show below the optimal strategy is to take the edge which reduces $|i - j|$. (If $i = j$ either a coin flip or a fixed choice can be used to decide.) We call this edge the *equalizing* edge or link. See figure 4.
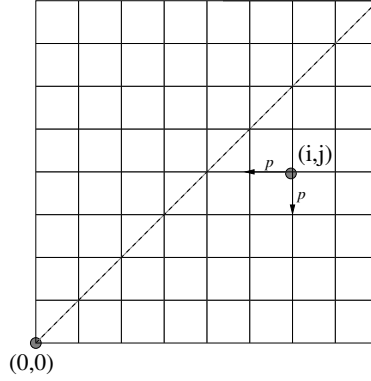
Fig. 4. Either of the two possible distance-reducing links at (i, j) is up independent of the other and with probability p. There are two distance-reducing links at (i, j) while only one is equalizing (the one leading towards the diagonal).

More formally we study the following algorithm for a node (i, j) holding a packet destined for $(0, 0)$:

1. If $i = 0$ and the edge to $(0, j-1)$ exists, forward the packet to $(0, j-1)$.
2. If $j = 0$ and the edge to $(i-1, 0)$ exists, forward the packet to $(i-1, 0)$.
3. If the edge to $(i-1, j)$ exists and the edge to $(i, j-1)$ does not exist, forward the packet to $(i-1, j)$.
4. If the edge to $(i, j-1)$ exists and the edge to $(i-1, j)$ does not exist, forward the packet to $(i, j-1)$.
5. If both edges to $(i-1, j)$ and $(i, j-1)$ exist and $i > j$ then forward the packet to $(i-1, j)$, else to $(i, j-1)$.
6. Otherwise, wait.

Let q be the probability of forwarding a packet on an equalizing link. Consider the event DR that a distance-reducing link exists. Since DR occurs if either exactly one or both distance-reducing edges are up we conclude that $\Pr[DR] = 2p(1-p) + p^2$. Consider the events BL that both distance-reducing links are up, and OL that only the distance equalizing link is up. Then

$$
\begin{aligned}
q &= \Pr[BL \mid DR] + \Pr[OL \mid DR] \\
&= \frac{p^2}{2p(1-p) + p^2} + \frac{p(1-p)}{2p(1-p) + p^2} \\
&= \frac{1}{2-p}.
\end{aligned}
$$

Observe that for $0 < p < 1$, $q = 1/(2-p) > 1/2$.

3.1 Analysis of the algorithm

We want to compute an upper bound on number of steps to route a packet from the source node (i, j) to the destination node $(0, 0)$. It is clear from the description of the algorithm that the route followed by a packet can be divided into two phases.

- Phase 1: The packet travels from the node (i, j) to a node in the first row (or first column) of the grid.
- Phase 2: The packet travels entirely in the first row (or first column) until the node $(0, 0)$ is reached.

Let X be the random variable that is the distance from the origin at which the packet hits the horizontal or vertical axis, that is, the packet enters the horizontal axis at node $(X, 0)$ (or the vertical axis at node $(0, X)$). In the rest of the paper, we use $m = i + j$ to denote the total number of links traversed by the packet. Therefore, the packet uses $m - X$ links in Phase 1, and X links in Phase 2. In the first phase, every intermediate node has two distance-reducing links available and in the second phase, there is exactly one distance-reducing link available at every node. It follows that the packet waits an expected $1/r$ steps to access a link in the first phase, and an expected $1/p$ steps to access a link in the second phase. Thus the expected time taken by the algorithm is $(m - E(X))/r + E(X)/p$. In this section we prove high probability bounds on X and the number of steps taken by the algorithm. Note that the value X is equivalent to the *residue* analyzed by Knuth in the toilet paper problem. The number of sheets on the two toilet paper rolls represent the x and y co-ordinates and choosing an equalizing link is analogous to being a big-chooser. Our bounds on X may be interpreted as high probability bounds on the residue left at the time one roll becomes empty starting with i sheets on one roll and j sheets on the other.

Since q is the probability that a packet is forwarded along an equalizing link it is natural to conjecture that the route of a packet starting at (i, j) with $i > j$ will be characterized by a line L with slope $\frac{1-q}{q}$ passing through the point (i, j). Two events will change the trajectory of the packet. Either the packet reaches the diagonal line $y = x$ or it reaches the horizontal or vertical axis as shown in Figure 5. Which of these two events occurs depends on the starting position of the packet.

Define by R the region $\{(i, j) \mid i \geq j > i\frac{1-q}{q} \text{ or } i\frac{q}{1-q} < i \leq j\}$ delimited by the two lines $y = \frac{qx}{1-q}$ and $y = \frac{(1-q)x}{q}$ (see shaded region in Figure 5). Without loss of generality we consider only starting positions on or below the diagonal of the square. (The case when the starting positions are above the diagonal can be handled in a similar manner.) Also let B be the band delimited by two curves (depicted with dashed lines) surrounding the above lines defined by $y = \frac{q}{1-q}x \pm c_0\sqrt{\frac{x}{q}\log\frac{x}{q}}$ and $y = \frac{1-q}{q}x \pm c_0\sqrt{\frac{x}{q}\log\frac{x}{q}}$ where $c_0 = \sqrt{\ln 2(1 + \ln 8)/2}$.

The thrust of the argument is to show that three basic cases may occur depending on the starting position of the packet:
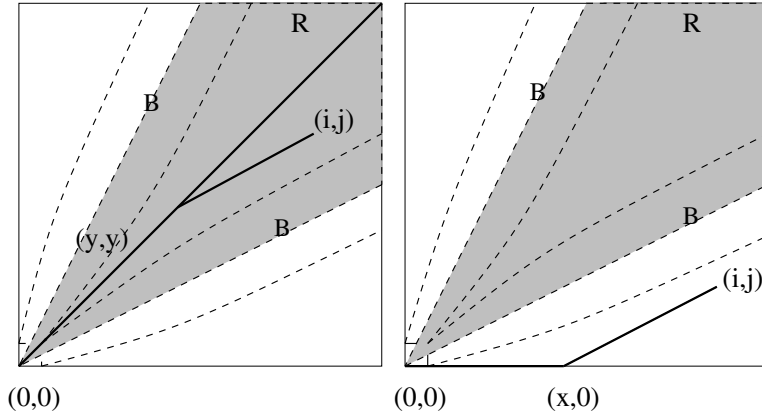
Fig. 5. Possible routes of a packet starting from the source node (i, j), either inside (left) or outside (outside) the shaded region R. When the packet is within the band B it may hit either the diagonal or the horizontal.

$(i, j) \in R \setminus B$: Phase 1 may be divided into two sub-phases for such a packet. In the first sub-phase, it is routed within the region R, staying "close" to the line L until it hits the diagonal. In the second sub-phase, it stays "close" to the diagonal and hits the horizontal axis at distance $O(\log m)$ from the origin with high probability.

$(i, j) \notin R \cup B$: The packet is routed with high probability outside the region R, staying close to the line L and hits the horizontal axis, at distance $O(\sqrt{m \log m})$ away from the point $(i - jq/(1 - q), 0)$.

$(i, j) \in B$: A packet starting in this area may behave either like a packet in the first case or like one in the second case. In other words, the packet may either hit the diagonal, and then continue on to hit the horizontal axis at distance $O(\log m)$ from the origin. Alternatively, it may hit the horizontal axis without hitting the diagonal, but at distance $O(\sqrt{m \log m})$ from the origin.

We make this precise in the following lemma:

Lemma 1. *Consider a packet starting at position (i, j), and let X be the random variable such that the packet enters the horizontal axis at node $(X, 0)$ or the vertical axis at node $(0, X)$. With probability at least $1 - 1/2m$, the value of X is less than:*

$$\begin{cases} i - \frac{jq}{1-q} + c_0\sqrt{m \log m} & \text{if } (i, j) \notin R \cup B \\ c_1 \log m & \text{if } (i, j) \in R \setminus B \\ \max\{c_0 + c_2, c_1\}\sqrt{m \log m} & \text{if } (i, j) \in B \end{cases}$$

for $m = i + j > 2$, where $c_0 = \sqrt{\ln 2(1 + \ln 8)/2}$, $c_1 = -2/\log(1/2 + 2q(1-q)) > 0$ is a constant that depends only on q, and $c_2 = \sqrt{\ln 2(1 + \ln 4)/2}$.

Lemma 1 gives us a bound on the number of links travelled in each phase with high probability. (It also provides high probability bounds on the residue in Knuth's toilet paper problem.) It remains to factor in the number of steps waiting for a link to come up, and compute the number of steps taken by the algorithm. The main result of the paper follows:

Theorem 1. *With probability at least* $1 - 1/(i+j)$ *the number of steps to route a packet from node* (i, j) *to node* $(0, 0)$ *is*

$$
\begin{cases}
\frac{i+j}{r} + O\left(\sqrt{(i+j)\log(i+j)}\right) & \text{if } (i,j) \in R \cup B \\
\frac{\max\{i,j\}}{p} + O\left(\sqrt{(i+j)\log(i+j)}\right) & \text{if } (i,j) \notin R \cup B
\end{cases}
$$

3.2 Lower bound

In this section we show that the result of Theorem 1 is tight.

Theorem 2. *The expected number of steps for any routing algorithm to route a packet from* (i, j) *to* $(0, 0)$ *is at least*

$$
\max\left\{\frac{i+j}{r}, \frac{\max\{i,j\}}{p}\right\}.
$$

4 Conclusion

In this paper we analyzed a natural algorithm for routing on a delay tolerant grid which improves the expected time of delivery by up to a factor of two over a fixed path algorithm, depending upon the probability of an edge being up during a time step and the relative positions of the source and destination. Furthermore we showed that this is optimal, in the sense that no other algorithm can achieve a better expected running time. We also presented experimental evidence that considering multiple (possibly not shortest) paths instead of one fixed (greedy) path can decrease the expected time to deliver a packet on a randomly deployed Markovian sensor network by as much as 65 per cent depending on the probability that an edge exists in a given time interval.

References

1. C. Avin, M. Koucky, and Z. Lotker. How to Explore a Fast-Changing World. *Proceedings of ICALP*, pages 121–132, 2008.
2. B. Bui-Xuan, A. Ferreira, and A. Jarry. Computing Shortest, Fastest, and Foremost Journeys in Dynamic Networks. *International Journal of Foundations of Computer Science*, 14:267–285, 2003.

3. J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. *Proceedings IEEE INFOCOM*, pages 1–11, 2006.

4. E.M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40, 2007.

5. K. Duffy and W.M.B. Dukes. On Knuth's Generalisation of Banach'Match Box Problem. *Mathematical Proceedings of the Royal Irish Academy*, 104(1):107–118, 2004.

6. A. Ferreira. Building a reference combinatorial model for manets. *IEEE Network*, 18:24–29, 2004.

7. A. Ferreira, A. Goldman, J. Monteiro, and A. Ferreira. On the Evaluation of Shortest Journeys in Dynamic Networks. *Sixth IEEE International Symposium on Network Computing and Applications, 1007. NCA 2007.*, pages 3–10, 2007.

8. A. Goldman J. Monteiro and A. Ferreira. Performance Evaluation of Dynamic Networks using an Evolving Graph Combinatorial Model. *Proceedings of WiMob*, pages 173–180, 2006.

9. A. Jarry and Z. Lotker. Connectivity in evolving graph with geometric properties. *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 24–30, 2004.

10. B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *Proceedings of Mobicom*, pages 243–254, 2000.

11. D. Knuth. The toilet paper problem. *American Math Monthly*, 91(8):365–370, 1984.

12. A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *Proceedings of Workshop on Service Assurance with Partial and Intermittent Resources*, pages 239–254, 2004.

13. C. Liu and J. Wu. Scalable routing in delay tolerant networks. *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 51–60, 2007.

14. R.D. Mauldin (editor). The Scottish Book:Mathematics from the Scottish Cafe. *Birkheuser*, 1981.

15. S. Merugu, M.H. Ammar, and E.W. Zegura. Routing in Space and Time in Networks with Predictable Mobility. Technical report, Georgia Institute of Technology, GIT-CC-04-07, 2004.

16. T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. *Proceedings of SIGCOMM*, pages 252–259, 2005.

17. David Stirzaker. A generalization of the matchbox problem. *Mathematical Scientist*, 13:104–114, 1988.

18. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. *Duke University*, 2000.

19. B.D. Walker, J.K. Glenn, and T.C. Clancy. Analysis of simple counting protocols for delay-tolerant networks. *Proceedings of the second workshop on Challenged networks CHANTS*, pages 19–26, 2007.