Diversity of Forwarding Paths in Pocket Switched Networks

Vijay Erramilli,
Mark Crovella
Dept. of Computer Science
Boston University
(evijay, crovella)@cs.bu.edu

Augustin Chaintreau,
Christophe Diot
Thomson Paris
Paris, France
(augustin.chaintreau,
christophe.diot)@thomson.net

1. ABSTRACT

Forwarding in Delay Tolerant Networks(DTNs) is a challenging problem. We focus on the specific issue of forwarding in an environment where mobile devices are carried by people in a restricted physical space (a conference) and contact patterns are not predictable. We show for the first time a *path explosion* phenomenon between most pairs of nodes. This means that, once the first path reaches the destination, the number of subsequent paths grows rapidly with time, so there usually exist many near-optimal paths. We study the path explosion phenomenon both analytically and empirically. Our results highlight the importance of unequal contact rates across nodes for understanding the performance of forwarding algorithms. We also find that a variety of well-known forwarding algorithms show surprisingly similar performance in our setting and we interpret this fact in light of the path explosion phenomenon.

Categories and Subject Descriptors

H.1 [**Information Systems Applications**]: Models and Principles

General Terms

Algorithms, Measurement, Performance

Keywords

Pocket Switched Networks, DTN, Forwarding, Path Diversity

2. INTRODUCTION

Effective forwarding in a delay tolerant network (DTN) is strongly dependent on particular aspects of the setting, such as the size of the physical space involved, and the degree to which node-to-node contact and node mobility patterns are predictable. We are concerned with understanding forwarding in a setting that has not been extensively studied to date:

human-carried mobiles moving in a restricted space with high node and contact density. This situation corresponds to individuals at conferences, around office spaces, and in social settings. Networks in these environments are examples of Pocket Switched Networks (PSNs) [3, 9], in which both mobility and multihop forwarding can be used to support communication.

We believe that in order to design effective forwarding algorithms for PSNs, it is necessary to start by understanding the opportunities for forwarding that exist. While some work has already studied the performance of various forwarding algorithms in PSNs, there is little understanding to date on the *nature* of the forwarding problem in such settings. In particular, little is known about the kinds of paths (making use of both mobility and multiple hops) that exist in PSNs.

In this paper we study the paths that are available for forwarding in PSNs. We use as examples of the PSN setting measurements of contacts between Bluetooth-enabled Intel iMotes at recent Infocom and CoNext conferences. We develop an efficient method to enumerate all paths of interest between nodes (iMotes) and apply it to these traces to capture all paths of interest for a set of randomly generated messages.

The most striking property of the collection of paths we obtain is the presence of a phenomenon we term *path explosion.* Path explosion refers to the case in which, once the first path reaches the destination the number of subsequent paths grows rapidly with time, so there exist many near-optimal paths. We find that path explosion occurs for the large majority of messages in our datasets.

To explore this effect we build an analytic model describing how paths are created in a homogeneously mixing population and we use it to show how path explosion arises. However our homogeneous model does not explain all aspects of the phenomena observed in our data. In order to understand the nature of forwarding paths more accurately, we show that it is critically necessary to take into account the different contact rates exhibited by different nodes in the population. We find that it is useful to characterize the source and destination as either high contact rate ('in') nodes or low contact rate ('out') nodes. We show that when the contact rates of the source and destination nodes are taken into account, that the empirical properties of paths and the path explosion process can be understood more completely.

Once we understand the nature of paths available for for-

warding, it is easier to interpret the behavior of forwarding algorithms. Using our traces, we evaluate a range of different forwarding algorithms that are appropriate for DTNs. We find the surprising result that forwarding algorithms with very different strategies nonetheless show remarkably similar performance in terms of message delivery success rate and average message delivery delay. In fact we find that the particular choice of routing algorithm does not have a strong impact on delay or success rate for our data. We show that these results can be understood in the context of the path explosion phenomenon. Furthermore, understanding the presence of path explosion, and the particular details of how it occurs between combinations of 'in' and 'out' nodes, allows us to understand how efficient message delivery can take place in these settings.

3. RELATED WORK

Our work studies the set of paths available for forwarding in Pocket Switched Networks. While little work to date has studied the forwarding paths themselves, researchers are beginning to recognize that the movements of individuals carrying mobile nodes create contact patterns that are not well modeled using uniform or homogeneous assumptions [1, 15].

Nonetheless, to date most forwarding algorithms have been analyzed under the assumption that contact rates between nodes are homogeneous [6, 7, 17, 18]. Likewise, the most common mobility model used for forwarding evaluation is random waypoint [2] in which all nodes' speeds and directions are drawn from the same distributions. In contrast, our work shows that it is the *differences* in behavior of nodes in the network that is key to understanding the performance of forwarding in our setting. The closest work to our paper deals with the heterogeneity of inter-contact time distributions and its implications for routing [5]. The authors show that if the distributions of inter-contact times are homogenous and exponential then no forwarding scheme will be able to make practical use of past contact information. However if the rates are heterogenous, then past information can help in predicting future behavior and can thus help improve the performance of forwarding algorithms.

One of the main results in our work is that for our datasets, a wide variety of forwarding algorithms perform equally well in terms of success rate and delay. Among the forwarding algorithms we consider, we use epidemic forwarding [19] to study the path explosion phenomenon as well as to obtain upper bounds on performance. Among the factors we consider in comparing other forwarding algorithms are the use of past contact history, as proposed in [11, 10, 13, 8]. Another factor we consider is whether the algorithm bases decisions on single-hop performance or multi-hop performance. The authors of [11] propose basing forwarding decisions on Minimum Estimated Expected Delay (MEED) where the delay between nodes is calculated using past contact history over a sliding window. In contrast, Grossglauser et al [6] use only the last encounter time between nodes as a basis for forwarding decisions. We consider both these alternatives in our evaluation.

Many papers have looked at the performance of forwarding algorithms based on wireless measurements, but few have looked at forwarding in a dense environment with hard-to-predict contact patterns. Exceptions are [9, 4], which use data similar to ours. However those papers are concerned with studying forwarding algorithms rather than the nature of forwarding paths themselves, which is the focus of our work.

4. DATA

As described in our introduction our study is based on empirically collected data sets. These data sets were collected using short-range Bluetooth enabled devices (iMotes [4]) carried by individuals. These devices perform an inquiry-scan every 120 secs and log each response to the inquiry as a contact. A contact record includes the MAC address of the device that responded, as well as the start time and the end time of the contact. More details about the devices and the datasets, including synchronization issues can be found in [16].

Initial analysis of the data showed that contact activity varied considerably over the course of each day. The analyses in our paper are based on the assumption of approximately constant contact rates over time. Therefore we selected 3 hour periods for which visual inspection indicated that contact rates were relatively stable. Although a contact event need not be symmetric, we assume they are and modify the datasets accordingly. We note that contacts are not synchronized over the entire network, so even though the scanning rate of nodes is set to 120 secs, we observe contact activity every 10 sec.

We also note that most contacts observed in our datasets are very short and we do not consider contact durations in this work. We leave this to future work.

Empirical results in this paper are based on the following data sets:

Infocom 2006: This data set consists of logs taken during during the conference Infocom 2006. There are 98 nodes in total; with 20 nodes are stationary nodes distributed within the conference venue and the remaining nodes are distributed to conference participants who carried them for 4 days from 24th April 2006 to 28th April 2006. The specific time periods we study are 25 April (Tuesday) 9AM-12PM and 3PM-6PM on the same day.

Conext 2006: We collected similar data sets at Conext 2006 which was held from 4 Dec 2006 to 7 Dec 2006. 78 nodes were distributed to conference participants and the remaining 20 nodes were placed around the conference venue. The data sets we use are 4 Dec(Monday), 9AM-12PM and 3PM-6PM.

We replicated portions of our analyses on datasets taken at Infocom 2005 [4]. All of our results in this paper were consistent with the analyses. However we omit details for lack of space.

5. PATH EXPLOSION

A forwarding algorithm solves a decentralized search problem—it searches for a short path between a source and destination node, starting at a given point in time. This path exists in space and time, and its duration is the amount of time between message generation and message delivery. In order to understand how hard the search problem is for PSN settings, we start by studying the solution space.

Given a set of nodes $\mathbb{M}$ with $|\mathbb{M}| = N$, and a continuous time index $0 \leq t < t_{max}$, we define a *path* as a sequence of

tuples:

$$((x_1, t_1), (x_2, t_2), \ldots, (x_k, t_k))$$

where for all i, $t_i \leq t_{i+1}$, $x_i \in \mathbb{M}$, and

$$\ldots, (x_i, t_i), (x_{i+1}, t_{i+1}), \ldots$$

may be present only if x_i is in contact with node x_{i+1} at time t_{i+1}. Each tuple in the sequence is a *hop* and the *length* of a path is the number of hops it contains.

We assume that communication occurs via *messages* which are transmitted in whole from node to node in zero elapsed time. Messages travel along paths, i.e., they are only transmitted between nodes that are in contact. For any source $\sigma \in \mathbb{M}$ and destination $\delta \in \mathbb{M}$ a successful message delivery beginning at time t_1 can occur if there is a path

$$((\sigma, t_1), (x_2, t_2), \ldots, (\delta, t_k)).$$

The most basic goals of a forwarding algorithm are, given σ, t_1, and δ: (a) to find such a path if it exists; and (b) to find the path with shortest-achievable duration if more than one path exists. We use $\mathcal{P}_A(\sigma, \delta, t_1)$ to denote the fact that forwarding algorithm A can find a path from (σ, t_1) to (δ, t_k) for some $t_k < t_{max}$. If forwarding algorithm A finds at least one path, we denote the duration of the shortest-found path $(t_k - t_1)$ as $\mathcal{T}_A(\sigma, \delta, t_1)$.

Our concern centers on the *performance* of forwarding algorithms in the PSN setting. We define the performance of forwarding algorithm A as the *average delay*

$$D_A = \mathbb{E}\left[\mathcal{T}_A(\sigma, \delta, t_1) \mid \mathcal{P}_A(\sigma, \delta, t_1)\right]$$

and the *success rate*

$$S_A = \mathbb{E}\left[\mathbb{I}_{\{\mathcal{P}_A(\sigma, \delta, t_1)\}}\right]$$

when σ and δ are chosen uniformly at random over $\mathbb{M}$ and t_1 is chosen uniformly at random over $[0, t_{max})$.

Note that our formulation of the problem does not consider the *costs* of forwarding: for example, power or storage consumption. Reasonable secondary goals of a forwarding algorithm might be to minimize costs. However our focus is on understanding what alternatives in terms of paths are *available* for use by a forwarding algorithm. After obtaining an understanding of what paths are available, it is easier to concentrate on minimizing costs.

5.1 Path Enumeration

As discussed in Section 3, many previous studies of forwarding algorithms have looked at forwarding performance in specific settings. However, our work takes a different approach by first empirically characterizing the set of paths that are available for use by forwarding algorithms. To accomplish this properly, there are a number of specific considerations.

First, it is important to specify the characteristics that are expected of all forwarding algorithms, because these define the set of paths of interest. We assume that under any reasonable forwarding algorithm, a node holding a message for a destination node will deliver that message whenever it encounters the destination. We call this the assumption of *minimal progress.*

We also restrict our attention only to loop-free paths, i.e., paths in which no node appears more than once. We make this restriction because if looping paths are considered, then an arbitrarily large number of paths may be generated simply by following a loop a varying number of times. One can imagine a scenario where due to high mobility a node is encountered more than once over time and hence can potentially show up in a path multiple times. Enumerating such cases does not expose truly distinct forwarding paths. While looping paths may occur in practice this is not a concern, because disregarding such looping paths means that our counting results are conservative—so in fact even more paths may be present than our results indicate. For a similar reason, we assume nodes have infinite buffers and we do not consider paths that may be created because a node is forced to drop a message. Thus, once a node receives a message it holds the message forever.

A path respects minimal progress and loop avoidance if it does not contain more than one instance of any node, and the destination appears only at the end of the path (if at all). Furthermore, the assumption of minimal progress combined with unlimited message holding implies a more subtle condition, which we call *first preference*. Consider a path

$$P = ((\sigma, t_1), \ldots, (x_i, t_i), \ldots, (x_j, t_j), (\delta, t_k)).$$

If it also happens that x_i encounters the destination δ at time $t' < t_k$, then the following is also a path:

$$P' = ((\sigma, t_1), \ldots, (x_i, t_i), (\delta, t')).$$

Note that no forwarding algorithm respecting minimal progress would take path P rather than path P'. Thus path P is not a *first preference* path. Any path respecting loop avoidance and minimal progress (including first preference) is a *valid* path.

Second, in moderately-large datasets (such as those we work with) a complete enumeration of all valid paths with a given (σ, δ, t_1) is prohibitively expensive. Therefore a key element of our approach is efficient enumeration of the most *important* paths. Our strategy for tackling this problem has two steps:

1. Define the problem as one of path enumeration on a space-time graph (defined below) and

2. at each time step, use dynamic programming to maintain the (up to) k shortest valid paths reaching each node. By 'shortest' we mean the path with the least number of hops.

This strategy allows us to determine the *optimal* path reaching the destination, that is, the path with shortest achievable duration under any forwarding algorithm. We denote the duration of the optimal path $\mathcal{T}(\sigma, \delta, t_1)$ and note that it is the minimum duration path found by epidemic forwarding, i.e.,

$$\mathcal{T}(\sigma, \delta, t_1) = \min_A \mathcal{T}_A(\sigma, \delta, t_1) = \mathcal{T}_{Epidemic}(\sigma, \delta, t_1).$$

This strategy also allows us to determine each subsequent valid path that reaches the destination, up until the point at which k or more valid paths reach the destination in a single timestep.

To recap we want to answer the following questions: How many paths exist between a source-destination node pair? What are the properties of these paths? To organize the search process, we convert the sequence of node contacts into a *space-time* graph, which is a special kind of directed
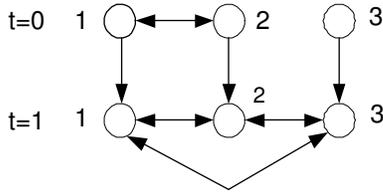
Figure 1: Example space-time graph.

weighted graph. Our use of the space-time graph is based on [14].

Time is discretized in increments of Δ. In all our work we use $\Delta = 10$ sec. Vertices in the space-time graph are pairs (x_i, T) with $x_i \in \mathbb{M}$ and $T = c\Delta$ for $c \in \{1, 2, \ldots, \lfloor t_{max}/\Delta \rfloor\}$. Edges in the space-time graph come in two kinds:

1. There is an edge from vertex (x_i, T) to (x_j, T) iff node x_i was in contact with node x_j at any time during $[T - \Delta, T)$. Such an edge has weight zero.

2. There is an edge from vertex (x_i, T) to $(x_i, T + \Delta)$ for all $x_i \in \mathbb{M}$. Such edges have unit weight.

For example, consider a network with three nodes. Nodes 1 and 2 are in contact during the first timestep, while all three nodes are in contact with each of the others during the second timestep. Then the corresponding space-time graph is as shown in Figure 1. The horizontal edges have weight zero and the vertical ones have weight 1.

Given a space-time graph discretized by Δ and a message defined as (σ, δ, t_1), we enumerate shortest paths using dynamic programming. The algorithm uses as its data structure an $N \times k$ array of paths, denoted P. It maintains the following invariant: at any given timestep $T = c\Delta$ the entry P_{ij} is the jth shortest path reaching from (σ, t_1) to (x_i, t') for some $t' \in [T - \Delta, T)$. The algorithm is given in Figure 2.

5.2 Observations

Using the algorithm described in the previous subsection we are able to enumerate paths for a given message with (σ, δ, t_1) up to the time when k paths reach the destination in a single timestep. For any given message, we will use T_n to denote the time at which the nth path reaches the destination, in order of increasing delivery time.

Our results show that for many messages, the duration of the optimal path can be quite long (thousands of seconds). However the most striking aspect of our results is that in the vast majority of cases, once the first path reaches the destination a very large number of additional paths reach the destination soon thereafter – typically tens or a few hundreds of seconds later. We refer to this phenomenon generically as *path explosion.*

To make the notion concrete, we define the time that path explosion occurs to be the time by which 2000 paths in total have reached the destination, i.e., T_{2000}. We would like to point out that there is nothing sacrosanct about the number 2000; we believe that by enumerating 2000 paths (; a large number of paths) we can develop a concrete picture of the set of paths available. Note that we can always identify this time accurately (to within an error of Δ) as long as we set k in our algorithm to be 2000 or greater. We refer

1. Let $T = c\Delta$ such that $t_1 \in [T - \Delta, T)$.
2. Let $s = i$ such that $x_i = \sigma$.
3. Let $P_{ij} = \varnothing$ for all i, j.
4. Let $P_{s1} = ((\sigma, T))$.
5. While (true)
   (a) Let $X = \varnothing$.
   (b) For $i = 1, \ldots, N$
      i. For $j = 1, \ldots, k$
         • If $P_{ij} \neq \varnothing$:
            A. Let $X' = $ all distinct extensions of path P_{ij} to vertices reachable from $(x_i, T + \Delta)$ via paths of zero weight.
            B. Let $X = X \cup X'$.
   (c) Remove the invalid paths from X.
   (d) Output all paths in X reaching δ.
   (e) If there are k or more paths in X reaching δ, stop.
   (f) For $i = 1, \ldots, N$
      i. Select the (up to) k shortest paths from X that terminate at node x_i and place them in $P_{ij}, j = 1, \ldots, k$.
   (g) Let $T = T + \Delta$.

Figure 2: Algorithm for k shortest paths enumeration.

to the 'time to explosion' (T_E) as the elapsed time between the arrival of the first path and the arrival of the 2000th path, i.e., $T_E = T_{2000} - T_1$. We would also like to point out that there is a possibility that some messages might not have 2000 paths, but may still have a fairly large number of paths between them. We do not consider such messages.

In Figure 3(a) we show the CDFs for optimal path duration in datasets Infocom '06 9-12 and Infocom '06 3-6. The figure shows that optimal path duration can be quite long; in fact, a large fraction of messages (over 25%) require over 1000 seconds before the first path reaches the destination. On the other hand, Figure 3(b) shows CDFs for the time to explosion. Here the story is quite different: almost half the messages see an explosion with little or no delay, and 97% of them have T_E less than or equal to 150 secs.

These figures show the surprising fact that optimal path duration can be an order of magnitude or more greater than time to explosion. That is, in many cases, the first path takes a long time to reach the destination, after which many paths reach the destination in relatively short order. In a phrase, path explosion means that *shortly after the optimal path reaches the destination, there are a large number of nearly-optimal paths to the destination.*

A closer look at the path explosion phenomenon is shown in Figure 4, which is also based on dataset Infocom '06 9-12. Each point in the figure corresponds to (T_1, T_E) for a single message (σ, δ, t_1) (note the difference in scale on the x and y axes). The figure shows that there is no clear relationship between optimal path duration and time to explosion; there are many cases in which the optimal path reaches the destination quickly but the path explosion occurs comparatively
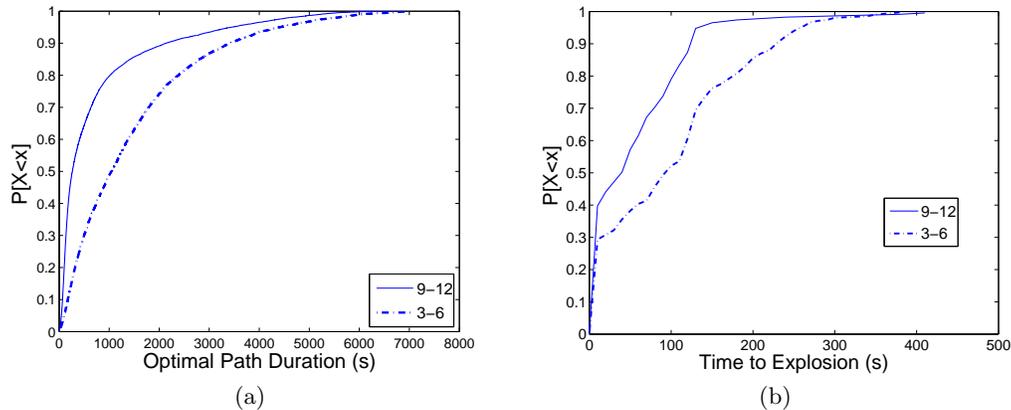
Figure 3: (a) CDFs of optimal path duration Infocom 06 9-12 and 3-6 (b) CDFs of time to explosion Infocom 06 9-12 and 3-6
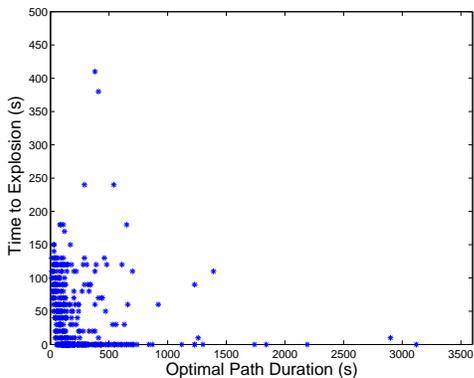

Figure 4: Optimal path duration vs. Time to Explosion, Infocom '06 9-12.

late, and there are many cases in which the optimal path reaches the destination late while the path explosion occurs quickly thereafter.

It is difficult to characterize how the number of successful paths grows in time, in part because path explosion generally occurs quite rapidly and our time measures are rather coarsely discretized. However we can get a rough sense of how the explosion occurs if we look at the slowest cases. In Figure 5 we examine all paths of the messages for which the time to explosion T_E was 150 seconds or greater, again using dataset Infocom '06 9-12. Note that although for each of these messages, the 2000th message arrives at least 150 seconds after the optimal time, nonetheless many messages still arrived during the period between T_1 and T_{2000}. The figure shows that the number of paths arriving over time grows in an approximately exponential fashion.

In summary, we find that once paths are enumerated, a number of surprising properties emerge. While the optimal path duration can be long, there is a path explosion effect that occurs relatively quickly after the first path reaches the destination. However, there is no clear relationship between optimal path duration and time to explosion. Finally, it appears that the explosion process is roughly exponential in time. We seek to explain these observations in the following sections, and we study the implications of these observations

in later sections of the paper.

6. PROPERTIES OF PATHS

In this section we use models and data analysis to understand the questions raised in Section 5.2. In particular the next two subsections ask: (1) How and why does path explosion occur? and (2) What determines the relationship between optimal path duration and time to explosion? To answer the first question we develop an analytic model making the assumption of homogeneity – equal contact rates among all nodes. To answer the second question we incorporate considerations of unequal contact rates (inhomogeneity).

6.1 A Homogeneous Model for Path Explosion

As a first step in trying to understand the path explosion phenomenon, we use a highly simplified model. We consider a setting in which nodes contact other nodes randomly and at constant rate. For a given message, we count the number of paths arriving at each node over time. The model is not restricted to counting just valid paths (which would add considerable complexity) but we seek qualitative results which are presumably insensitive to such details. The main result we show in this section is that the number of paths arriving at any node (such as the destination) grows exponentially in time, with rate given by the (homogeneous) contact rate between nodes.

Our results use known techniques from the study of epidemic process [12, 20]; however we extend those tools to study the set of all forwarding paths, instead of just the first path obtained to each destination.

6.1.1 Assumptions

Our model makes two assumptions, which derive from the uniformity of the setting.

- Poisson contacts: We associate with each node x_n a process of contact opportunities with all other nodes that is a homogeneous Poisson process with intensity λ.

- Homogeneity: whenever a contact opportunity occurs for node x_i, the contacted node x_j is chosen independently from the past, and uniformly among the N
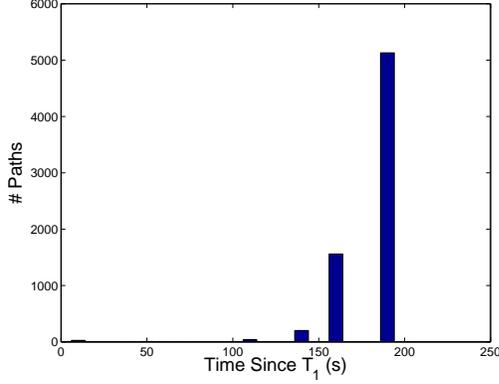
Figure 5: Histogram of number of paths for pairs with time to explosion $\geq$ 150 sec, Infocom '06 9-12.

nodes in $\mathbb{M}$. Note that this model does not include variability of contact rates among pairs, an issue we take up in Section 6.2.

Without loss of generality we study forwarding paths for a message created by a source (denoted by σ) at time $t = 0$.

6.1.2 Evolution and limit for large networks

Our model is adapted from the analysis of population dynamics. The *state* $S_n(t)$ of a node x_n at time t is the number of paths from the source that reach x_n before t. When x_n has an opportunity to contact x_m, the following transition occurs at x_m:

$$S_m(t) \leftarrow S_m(t) + S_n(t).$$

For any $k \geq 0$, we denote

$$U_k(t) = \#\left\{ x_n \in \mathbb{M} \mid S_n(t) = k \right\},$$

for the number of nodes with state k. To focus on the evolution of the population of nodes as a whole we study the evolution of the collection of variables

$$U(t) \in \mathbb{N}^{\mathbb{N}}, \; U(t) = \left\{ U_k(t) \mid k \geq 0 \right\}.$$

Note that we have $\sum_{k \geq 0} U_k(t) = N$, so $U(t)/N$ represents the empirical density of nodes in each states. If one does not differentiate between nodes in the same state, then $U(t)$ entirely characterizes the system at time t.

Note that when node x_n having state $i \neq 0$ contacts node x_m having state j, the collection of variables U is modified as follows.

$$U_{i+j} \leftarrow U_{i+j} + 1 \; ; \text{ and } U_j \leftarrow U_j - 1.$$

We denote by $\lambda_{n,m}$ the intensity of the contacts from node x_n to node x_m. For any fixed $i > 0$ and $j \geq 0$, transitions of the type (i, j) as defined above occur with the following intensity:

$$\beta_{i,j} = \sum \left\{ \lambda_{n,m} \mid n \neq m \text{ s.t. } S_n(t) = i, \, S_m(t) = j \right\}.$$

PROPOSITION 1. *The rate of transition $\beta_{i,j}$ is a function of the density of nodes in each state $U(t)/N$.*

$$\beta_{i,j} = N \cdot \lambda \frac{U_i(t)}{N} \frac{U_j(t)}{N} = N\tilde{\beta}_{i,j}\left(\frac{U(t)}{N}\right)$$

PROOF. Let us fix an n such that $S_n = i$. Then we have

$$\sum_{\{\; x_m \; \mid \; S_m(t) = j \;\}} \lambda_{n,m} = \lambda\frac{U_j(t)}{N},$$

since we have assumed that the contact process for each node x_n has rate λ and is homogeneous with regard to other nodes.

Summing for all nodes x_n having state i we obtain

$$\beta_{i,j} = U_i(t)\lambda\frac{U_j(t)}{N} = N \cdot \lambda\frac{U_i(t)}{N}\frac{U_j(t)}{N}$$

$\square$

We can describe the process $U(t)$ taking on values in $\mathbb{N}^{\mathbb{N}}$ as a Markov jump process, with transitions indexed by $\{\; (i,j) \mid i > 0, \, j \geq 0 \;\}$.

The rate of transition (i, j) is a function $\tilde{\beta}_{i,j}$ only of the density $U(t)/N$ of nodes having states i and states j, multiplied by N. In other words, the process $U(t)$ may be written as

$$U(0) + \sum_{i>0, j\geq 0} (e_{i+j} - e_j) \cdot \int_0^t \mathcal{N}_{i,j}\left(N\tilde{\beta}_{i,j}\left(\frac{U(s)}{N}\right)\right) ds$$

where, for all $i > 0, j \geq 0$, $\mathcal{N}_{i,j}$ denotes a Poisson counting process with intensity 1, and e_k denotes the infinite vector with all entries null except for 1 at position k.

When one considers the density process $U(t)/N$ in the case where N is large, Kurtz's limit theorem [12] shows that one can replace the Poisson counting process in the above expression by the process mean. This means that the trajectory of the density process for large N closely approaches the solution of a deterministic ordinary differential equation.

PROPOSITION 2. *As N goes to infinity, if we assume that $U(0)/N \to u(0) > 0$, then we have for all $K \geq 0$:*

$$\sup_{0 \leq s \leq t, \, 0 \leq k \leq K} \left| \frac{U_k(s)}{N} - u_k(s) \right| \to 0 \; \text{ a.s.}$$

where $u : [0; \infty[\to \mathbb{R}^{\mathbb{N}}$ is the solution of the ODE

$$u(t) = u(0) + \sum_{i>0, j\geq 0} (e_{i+j} - e_j) \cdot \int_0^t \tilde{\beta}_{i,j}\left(u(s)\right) ds$$

PROOF. A priori the process $U(t)/N$ evolves in a space with an infinite number of dimensions $\mathbb{N}^{\mathbb{N}}$, hence Kurtz's limit theorem [12] does not immediately apply to it. However, one can consider for any $K > 0$ a threshold process where all nodes in states $\{\; k \mid k > K \;\}$ are collapsed into a collection of nodes in a single sink state. This threshold process has a finite number of dimensions, and satisfies all the assumptions of the Kurtz limit theorem. To complete the proof we note that the threshold process defines exactly the same dynamics for states in $\{\; 0, \cdots, K \;\}$ as in the infinite process. $\square$

PROPOSITION 3. *For any $k \geq 0$, we have*

$$u_k(t) = u_k(0) + \lambda \int_0^t \sum_{i=0}^k u_i(s)u_{k-i}(s) - u_k(s)ds \qquad (1)$$

PROOF. From the definition of u found in Proposition 2, we obtain that $u_k(t) - u_0(t)$ may be written for all $k \geq 0$ as:

$$\sum_{i>0, j\geq 0, i+j=k} \int_0^t \tilde{\beta}_{i,j}\left(u(s)\right) ds - \sum_{i>0, j=k} \int_0^t \tilde{\beta}_{i,j}\left(u(s)\right) ds$$

which may be rewritten as:

$$\sum_{i=1}^{k} \int_0^t \tilde{\beta}_{i,k-i}\left(u(s)\right) ds - \sum_{i>0} \int_0^t \tilde{\beta}_{i,k}\left(u(s)\right) ds$$

$$\sum_{i=1}^{k} \int_0^t \lambda u_i(s) u_{k-i}(s) ds - \sum_{i>0} \int_0^t \lambda u_i(s) u_k(s) ds$$

After adding $\lambda u_0(s) u_k(s)$ to each of the above terms and using $\sum_{i \geq 0} u_i(s) = 1$ the proposition is proved. $\square$

In other words, from a non trivial initial condition, the average number of nodes with exactly n paths in a large network evolves according to a deterministic equation.

6.1.3 Solution for ODEs, moments

We introduce for all $x \in \mathbb{R}$ the series $\phi_x : t \mapsto \sum_{k \geq 0} x^k u_k(t)$. After multiplying each equation in (1) by x^k and summing all of them we obtain

$$\frac{d\phi_x}{dt} = \lambda(\phi_x^2 - \phi_x).$$

Note that this equation may be rewritten as:

$$\lambda \cdot dt \quad = \frac{d\phi_x}{\phi_x(\phi_x - 1)} = \frac{d\phi_x}{\phi_x - 1} + \frac{d\phi_x}{\phi_x}.$$

This ordinary differential equation can be solved for all $x \in \mathbb{R}$. One needs to distinguish between two cases, depending on the initial value taken by ϕ_x.

- Assume $0 < \phi_x(0) < 1$ (for instance, by choosing x in $[0;1]$). We know that ϕ_x and $1 - \phi_x$ remains positive. The previous equation may then be written as:

  $$\ln(1 - \phi_x(t)) - \ln(\phi_x(t)) - \lambda t \text{ is constant.}$$

  Hence, $\phi_x(t) = \dfrac{\phi_x(0)}{\phi_x(0) + (1 - \phi_x(0))e^{\lambda t}}.$ (2)

- On the other hand, assume $\phi_x(0) > 1$ (for instance, by choosing $x > 1$). We know that ϕ_x and $\phi_x - 1$ remains positive. The previous equation may then be written as:

  $$\ln(\phi_x(t) - 1) - \ln(\phi_x(t)) - \lambda t \text{ is constant.}$$

  Hence, $\phi_x(t) = \dfrac{\phi_x(0)}{\phi_x(0) - (\phi_x(0) - 1)e^{\lambda t}}.$ (3)

The first case above allows us to derive closed form expression for the evolution with time of the mean number of paths per node (see below), as well as the other moments. In the second case, for any $x > 1$ the series ϕ_x becomes infinite in finite time $T_C(x) = \frac{1}{p} \ln \left(\frac{\phi_x(0)}{\phi_x(0) - 1} \right)$. In other words, if the initial distribution for the number of paths per node is light-tailed for a given coefficient, it loses this property within a finite time.

Mean number of paths:.

According to the definition of ϕ_x we can compute the expected number of paths for a node as follows.

$$\mathbb{E}\left[S_n(t)\right] = \mathbb{E}\left[\frac{1}{N} \sum_{m \in \mathbb{M}} S_m(t) \right] = \sum_{k \geq 0} k \cdot u_k(t)$$

thus $\mathbb{E}\left[S_n(t)\right] = \left. \dfrac{\partial \phi_x(t)}{\partial x} \right|_{x=1} = \mathbb{E}\left[S_n(0)\right] e^{\lambda t}.$ (4)

The last equation is obtained from (2) since

$$\frac{\partial \phi_x(t)}{\partial x}(x) = \frac{\frac{\partial \phi_x(0)}{\partial x}(x)}{D(x,t)} - \frac{\phi_x(0) \cdot \frac{\partial \phi_x(0)}{\partial x}(x) \cdot (1 - e^{\lambda t})}{D(x,t)^2},$$

with $D(x,t) = \phi_x(0) + (1 - \phi_x(0)) \cdot e^{\lambda t}$, so $D(1,t) = 1$ for all t.

6.2   The Effects of Inhomogeneity

The analysis developed in Section 6.1 sheds considerable light, but does not explain all the phenomena noted in Section 5.2. The analytic results confirm that path explosion *can* occur and that it should be exponential in nature. However, the analysis in Section 6.1 would also predict that optimal paths should be short, and that path explosion should occur immediately after the first path reaches the destination. These two predictions are not borne out in the data, as already shown in Section 5.2. To explain these phenomena, we need to examine the model assumptions.

The most unrealistic aspect of our analytic model is the assumption that all nodes contact each other at the same rate. In fact, prior work has shown that per-node contact rates in datasets like ours can be highly variable across different nodes. For example, the authors in [9] show that the distributional tails of inter-contact times for such data sets approximately follow a power law.

In Fig. 6 we plot the CDF for the total number of contacts each node has over each three hour period (Infocom '06 9-12, 3-6, Conext 9-12, and 3-6). We refer to the number of contacts a node makes per unit time as the node's *contact rate* or just *rate*. The CDFs suggest that the distribution of contact rates can be approximated as uniform over the range $(0, max)$ where max varies depending on the dataset. The key observation is that some nodes have rates quite close to zero, yielding extremely large average intercontact times. In fact, it is clear that the population consists of some nodes that are quite frequently in contact with a large number of other nodes, while there are also some nodes that rarely encounter other nodes at all.

Our analytic results suggest that for a message generated at time 0, the number of paths reaching node i will grow proportional to $e^{\lambda t}$ for a population in which all nodes have the same rate λ.

To discuss the case where nodes have different rates, we introduce some notation. We define the random variable $T(\sigma, \delta)$ to be equal to $\mathcal{T}(\sigma, \delta, t_1)$ when t_1 is chosen uniformly at random in $[0, t_{max})$. Furthermore we no longer assume equal contact rates, so we define λ_i to be the contact rate of $x_i \in \mathbb{M}$, i.e., $\lambda_i = \sum_k \lambda_{i,k}$. Finally we define the expected time for the first path as

$$H = \mathbb{E}\left[t \mid S_i(t) = 1\right].$$

In the case of the homogeneous model in Section 6.1,

$$H = (t \mid \mathbb{E}\left[S_i(0)\right] e^{\lambda t} = 1) = \frac{\ln N}{\lambda}$$

since $\mathbb{E}\left[S_i(0)\right] = 1/N$.

To understand the case when nodes have different rates, we reason as follows. Assume that at time t_0 a message is held by a node x_i having contact rate λ_i. Then at minimum, we expect a path explosion to occur starting at t_0 with rate
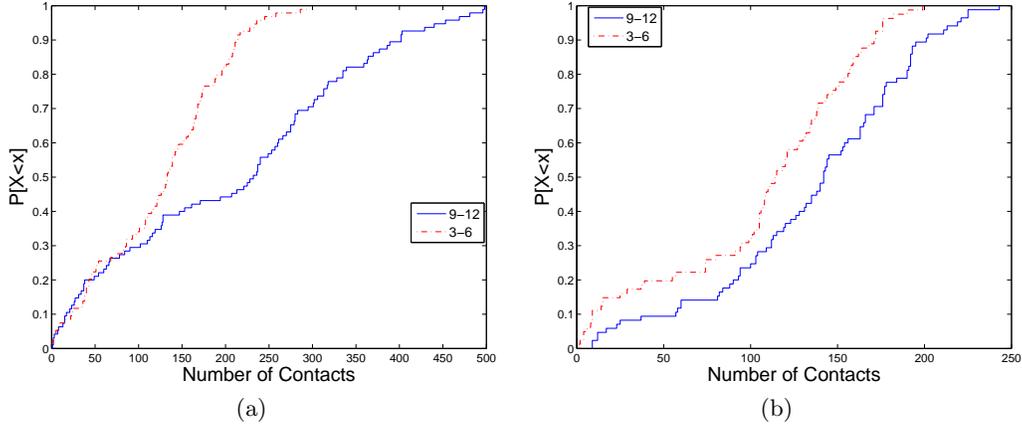
Figure 6: Cumulative Distribution of Number of Contacts (a) Infocom 2006 (b) Conext 2006

λ_i *among the subset of nodes* with contact rates greater than or equal to λ_i. That is, we can infer a lower bound on $\mathbb{E}[S_j(t - t_0)]$ proportional to $e^{\lambda_i(t-t_0)}$ for all j such that $\lambda_j \geq \lambda_i$. This *subset* path explosion means that the number of paths arriving at nodes with rate λ_i or greater grows at least as fast as $e^{\lambda_i t}$.

Now if λ_i is relatively small compared to typical contact rates, then initially the subset path explosion may proceed very slowly. However if at time $t_1 > t_0$, x_i encounters a node x_j with contact rate $\lambda_j > \lambda_i$, then a more rapid path explosion beginning at t_1 will occur with rate λ_j among nodes with contact rates λ_j or greater.

Thus if the source is a low-rate node, then there will be some initial time before it encounters a high-rate node, at which point a high-rate path explosion occurs. This initial time is related to H and so we argue that the time until high-rate path explosion occurs is on the order of $1/\lambda_i$. Thus in this case, T_1 will tend to be larger than is typical.

Furthermore, if the destination is a low-rate node, then path explosion may not be able to reach a high rate. Thus in this case, T_E will tend to be large.

This model suggests that a critically important role is played by the contact rates of the source and destination nodes. Then we can make the following hypotheses concerning the relative sizes of T_1 and T_E for four situations:

- First, when λ_σ is high and λ_δ is high, then path explosion begins immediately and at high rate. So both T_1 and T_E will tend to be small.

- Next, when λ_σ is high and λ_δ is low, then path explosion begins immediately but at a low rate. So T_1 will tend to be small but T_E will tend to be large.

- Next, when λ_σ is low and λ_δ is high, then there is a significant period before path explosion begins (on the order of $1/\lambda_\sigma$) but, once begun, path explosion proceeds at high rate. So T_1 will tend to be large but T_E will tend to be small.

- Finally, when λ_σ is low and λ_δ is low, then both T_1 and T_E will tend to be large.

To explore whether these hypotheses hold in our data, we separated nodes in each dataset into two equal-sized groups. The *in* set are those nodes with contact rates greater than

the median rate. The *out* set are those nodes with contact rates lower than the median rate. Since rate distribution is approximately uniform (as shown in Figure 6), the median rate in each case is approximately half of the maximum rate.

Then each message (σ, δ, t_1) can be placed into one of four cases: *in-in*, where λ_σ and λ_δ are both high; *in-out*, where λ_σ is high and λ_δ is low; *out-in*, where λ_σ is low and λ_δ is high; and *out-out*, where λ_σ and λ_δ are both low.

Using this labeling, we separate the points in Figure 4 into four groups, which are plotted separately in Figure 7. Each plot has the same format as Figure 4, except that only a subset of points are included.

The figures suggest that our hypotheses hold to a certain extent. In particular,

- For all the in-in messages, the optimal path duration T_1 is very low and the time to explosion T_E is less than 150 seconds.

- For the in-out messages, the optimal path duration T_1 is similar to the in-in messages, i.e., small. However the time to explosion T_E is much more variable and can take on relatively large values.

- For the out-in case, the optimal path duration T_1 tends to be larger than in the first two cases, while the time to explosion T_E is relatively small (as in the in-in case).

- Finally, for the out-out case, both T_1 and T_E can take on large values.

Results for other datasets are not shown due to lack of space, but generally showed similar behavior.

In summary, we have shown two key results in this section. First, we have used an homogeneous analytical model to show how path explosion occurs. We have shown that path explosion is an exponential function of both elapsed time and the rate λ at which nodes come into contact. This addresses the first question about our empirical results, namely, how and why does path explosion occur? Second we have shed light on the other set of questions arising from our empirical results by considering the effects of inhomogeneity in rates. We have shown that by looking at the contact rates of the source and destination nodes, one can gain insight into why the optimal path duration and the time to explosion vary, and one can relate these quantities to the relative magnitudes of the two contact rates.
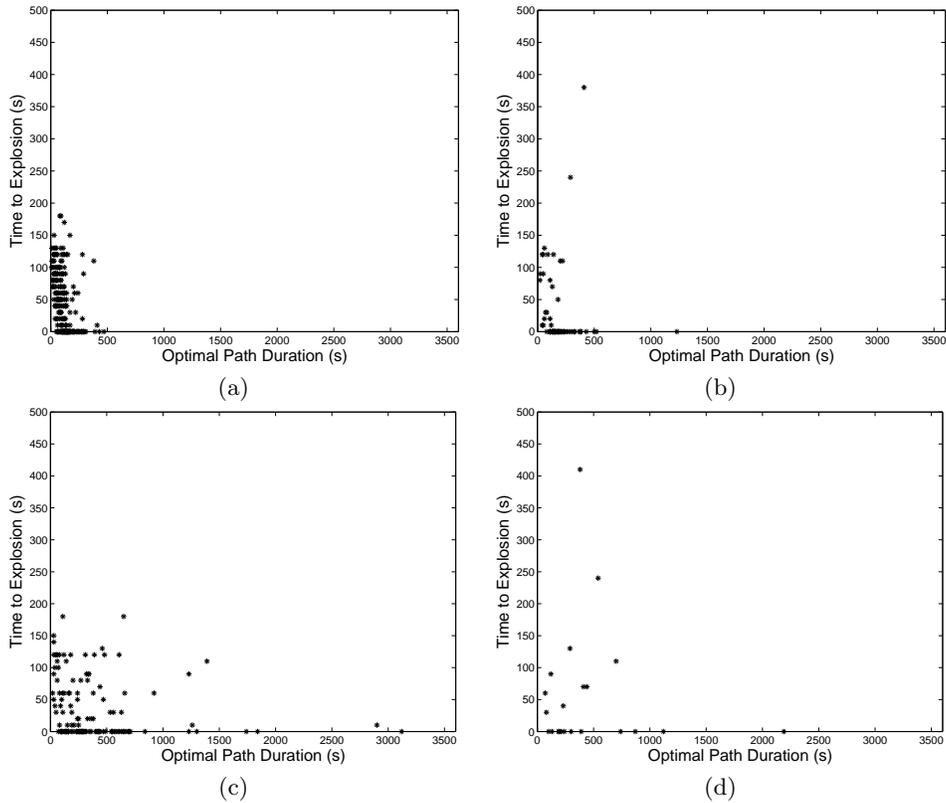
Figure 7: Path Explosion Plots Info 06 9-12 (a) in-in (b) in-out (c) out-in (d) out-out.

7. IMPLICATIONS FOR FORWARDING

In the previous sections we have developed insight into what sorts of paths are available for use by routing algorithms in settings such as ours. This has laid the groundwork for an understanding of how various forwarding algorithms perform in our setting. In this section we study the performance of a wide range of forwarding algorithms and relate their performance to the results developed in previous sections.

7.1 Experiments

We study forwarding by implementing different forwarding algorithms in a trace-driven simulator. We use as input to our simulator the contact traces described in Section 4. Each simulation therefore ran for simulated time of 3 hours. However, to avoid end-effects, only results from the first two hours are used.

For each trace and forwarding algorithm, we generate a set of messages (σ, δ, t_1) uniformly at random. We generate messages according to a Poisson process with rate one message per 4 seconds. As discussed in Section 5.1 we model nodes as having infinite buffers and nodes carry all the messages they receive till the end of the simulation. All our results are averaged over 10 simulation runs. Our metrics are success rate S and average delay D as defined in Section 5.1.

Our goal in this section is not to determine which forwarding algorithm is "best" but to compare the performance of a wide range of algorithms and gain insight into what properties of a forwarding algorithm yield good performance.

Hence we choose a set of algorithms designed to span a range of design choices. The design choices include:

- Destination aware vs. Destination unaware. Destination aware algorithms take the choice of δ into consideration in forwarding, while destination unaware algorithms do not.

- Single hop vs. Multi hop. Single hop algorithms use information about the most recent contact or next expected contact; multi hop algorithms take into account sequences of past or expected future contacts.

- Complete history vs. Recent history vs. Future knowledge. Complete history algorithms take into account the entire past history of other nodes when forwarding. Recent history algorithms taken into account only a limited amount of history (e.g., only the most recent encounter with the destination). Future knowledge algorithms make use of oracles that provide knowledge of future behavior of the nodes. Future knowledge algorithms are not practical but provide useful comparison cases.

To span these design choices we use the following forwarding algorithms. Some of these algorithms are known from the literature and others are modifications or extensions.

Epidemic Forwarding (Flooding): A node forwards all the packets it has to any node it meets, unless that node already has a copy of the packet [19]. As already noted, Epidemic Forwarding achieves the best possible
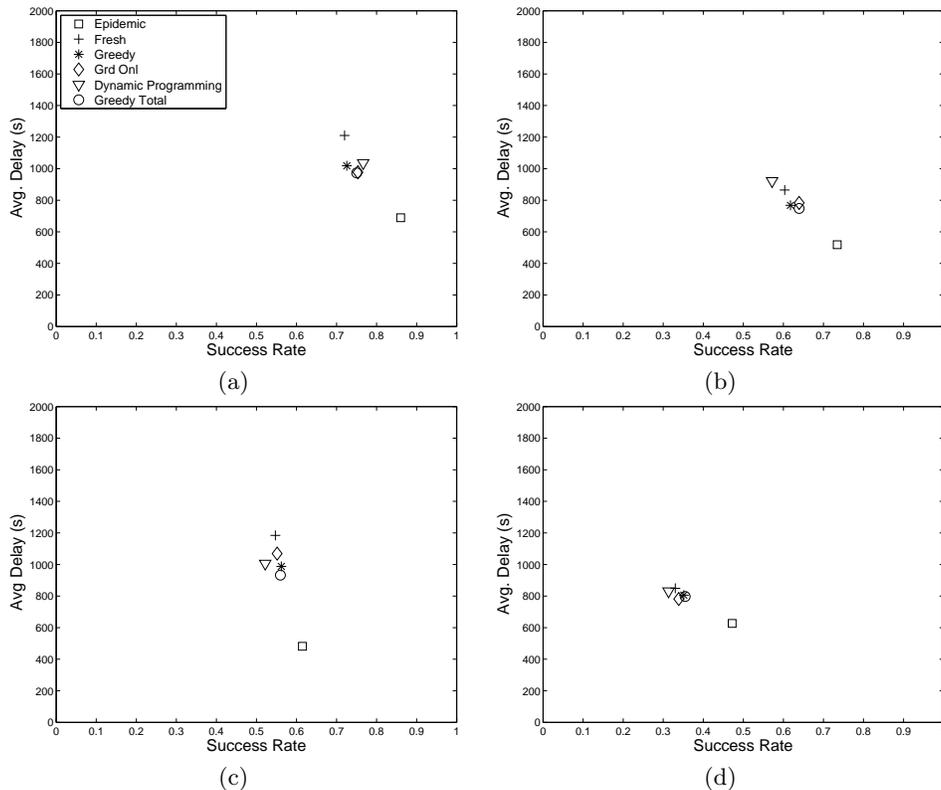
Figure 8: Average Delay (sec) vs Success Rate (a) Infocom 06 9-12 (b) Infocom 06 3-6 (c) Conext 9-12 (d) Conext 3-6

performance, so this algorithm yields upper bounds on success rate and average delay.

FRESH : Node x_i forwards a message to x_j upon contact if x_j has contacted δ more recently than has x_i [6].

Greedy : Node x_i forwards a message to x_j upon contact if x_j has contacted δ more times since the start of the simulation than has x_i. Greedy uses the entire history of contacts in contrast to FRESH which uses only the most recent contact history.

Greedy Total : Node x_i forwards a message to x_j upon contact if x_j has more total contacts (with all other nodes) than x_i. Thus Greedy Total is destination unaware, while FRESH and Greedy are destination aware. Greedy Total uses both past and future knowledge of internode contacts.

Greedy Online : Node x_i forwards a message to x_j upon contact if x_j has more contacts with δ since the start of simulation than does x_i. Greedy Online is destination aware but uses only past knowledge, unlike Greedy.

Dynamic Programming : Node x_i calculates the average delay between all pairs of nodes then finds the optimal path. This is based on the Minimum Expected Delay algorithm [10]. Dynamic Programming is based on both past and future knowledge of internode contacts.

7.2 Observations

7.2.1 Similarity of Performance

The most striking aspect of our results is the similarity in performance of the various different forwarding algorithms. This is illustrated in Figure 8. For each data set and forwarding algorithm we plot the average delay vs success rate.

The figure shows that almost all forwarding algorithms show virtually identical performance. The exception is epidemic routing (shown using square symbols in the plots) which shows somewhat better performance than the others, since it always finds the optimal path if one exists.

While the plots in Figure 8 show the similarity of average delay, a more detailed view is given in Figure 9. This figure shows the entire distribution of delay. The figure shows that the different algorithms show quite similar distributions of delay as well.

To explore reasons behind the similar performance of the different algorithms, we first verify that message delivery is not 'bursty'. That is, we confirm that the phenomenon is not simply due to many nodes all making contact at relatively infrequent times. An example scenario in the conference could be that participants convened at a common area after a session and it was during this time period that many packets were delivered. To confirm that this is not the case, we plot the cumulative totals of in Figure. 10. We plot the cumulative totals of delivery times of all optimal and near-optimal paths. As can be seen the delivery rate is fairly uniform in time. This shows that there are no special periods of time in which most messages are delivered.

Likewise, the fact that epidemic routing does noticeably better than the other algorithms means that similarity in
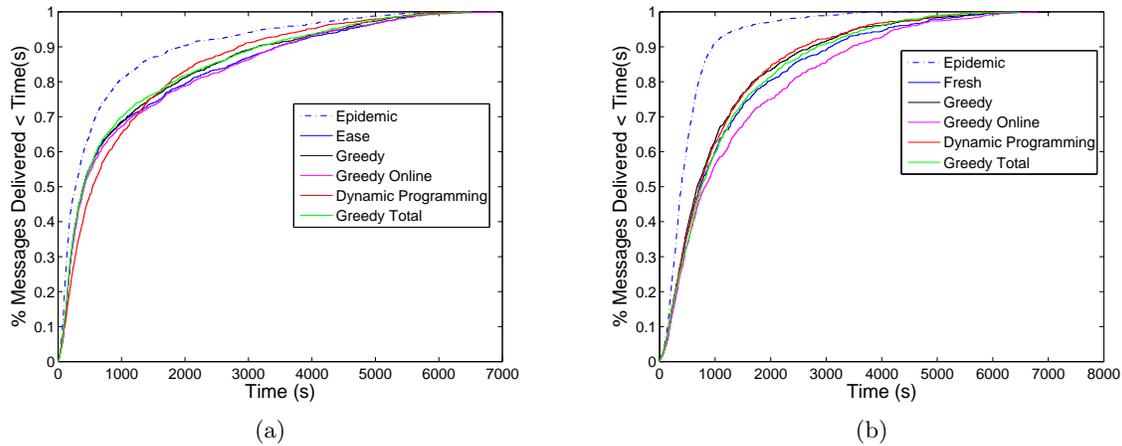
Figure 9: Delay Distributions (a) Infocom 06 9-12 (b) Conext 9-12
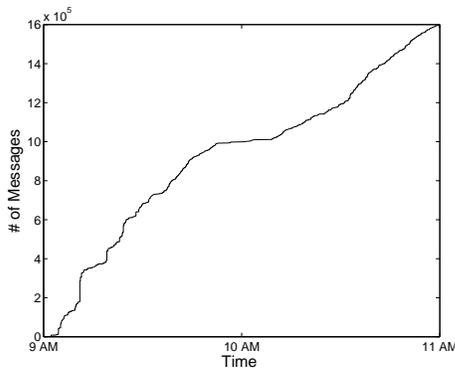

Figure 10: Message Reception Times Infocom 9-12

performance among algorithms does not occur because they are all finding the same, optimal paths. That is, in many cases there are better paths to be found than those chosen by most forwarding algorithms. However algorithms generally seem to find paths that are close to, though not exactly, optimal.

To understand the remarkably similar performance of the many different forwarding algorithms we examine *which* paths are being taken by the different algorithms. Some typical results are shown in Figure 11. Each plot in the figure corresponds to a particular message (σ, δ, t_1). The x axis measures the time since T_1 for this particular message, and the y axis shows the number of paths that reach the destination over time. Furthermore, for each algorithm, its symbol is superimposed on the bar corresponding to the path chosen by the algorithm.

The two cases in the figure show the typical path explosion phenomenon — the number of paths arriving at the destination grows approximately exponentially with time. Furthermore, the plots show that the paths used by all the forwarding algorithms reach the destination early in the path explosion process.

For example, the left hand plot shows that for this particular message, FRESH and Greedy are able to find paths arriving at time T_1 (i.e., optimal paths) while Greedy On-

line and Greedy Total find paths arriving in the next burst, 20 sec after T_1. Finally Dynamic Programming finds a path arriving in the third burst at 90 seconds after T_1. The situation is similar for the case of the message in the right hand plot.

This figure suggests that different algorithms may have similar (though non-optimal) performance because of the large number of paths that reach the destination *shortly after* T_1. When there are a large number of nearly-optimal paths, a forwarding algorithm may be able to find one of those paths relatively easily. That is, the existence of path explosion may be a factor that allows many algorithms to achieve somewhat similar performance.

If this hypothesis is correct, then the performance of forwarding algorithms should be strongly influenced by the way in which path explosion occurs. In Section 6.2 we found that path explosion occurs in different ways for different types of source-destination pairs, i.e., in-in, in-out, out-in, and out-out. Here we can put that insight to use.

If the path explosion effect is a major reason why different forwarding algorithms have similar performance, then algorithm performance should be fairly similar within pair types and quite different between pair types. For example, algorithm performance should be similar across all in-in pairs, but performance on in-in pairs should be quite different from that on in-out pairs.

To see whether this is the case, we separate our simulation results by pair type. The results are shown in Figure 12. The figure shows the average delay and success rate across the four pair types for each of the six forwarding algorithms.

The figure shows that success and delay depend primarily on the type of the source-destination pair as opposed to the type of algorithms used. All forwarding algorithms, with the exception of epidemic, again show similar performance.

However the figure also shows that, once messages are broken down by pair type, differences between forwarding algorithms start to emerge. In particular, we note a difference for algorithms making use of maximum information about contact patterns – i.e., Greedy Total and Dynamic Programming, which use both past and future information. These algorithms outperform others, but *only* in the case where one node is an 'out' node. That is, in the case where both nodes are 'in' nodes, then information about contact patterns is
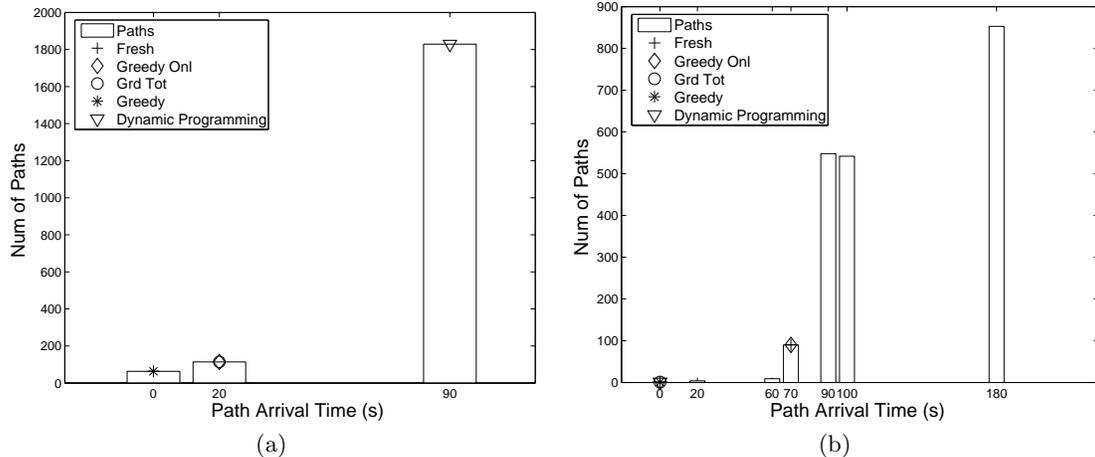
Figure 11: Paths taken by forwarding algorithms for two messages in Infocom '06 9-12 (a) Msg 88-82 (b) Msg 86-26

not particularly helpful. However, when one or both nodes are 'out' nodes, then maximum information about contact patterns is helpful.

Digging deeper, we see that Greedy Total performs particularly well when the source node is an 'out' node. Recalling the discussion in Section 6.2, when a source node has low rate λ_σ, then rapid path explosion does not occur until the message has been moved to a high rate node with $\lambda_i > \lambda_\sigma$. This is consistent with the strategy of Greedy Total. Note that Greedy Total is a destination-unaware strategy — it only seeks to move the message toward nodes with higher contact rates, i.e., those with $\lambda_i > \lambda_\sigma$.

These results suggest some heuristics for effective forwarding in settings like pocket switched networks. In particular, they suggest that a forwarding algorithm will be successful if it causes path explosion to take place as quickly as possible. This is a somewhat different principle for forwarding than has typically been used in past proposals. That is, rather than seeking to find a short path to the destination directly, a forwarding algorithm may instead work to cause path explosion to occur as quickly as possible. This suggests forwarding toward high rate nodes preferentially, regardless of their relationship to the destination.

The picture that emerges from the results in this paper gives a better view of the nature of node mobility, contact patterns, and forwarding performance in a conference type setting. In broad terms, the picture is as follows. Connection patterns between nodes lead to an exponential path explosion effect whose rate depends on the contact rates of nodes. Since contact rate varies considerably across nodes, path explosion occurs much faster among the higher rate nodes than the lower rate nodes. In this setting successful forwarding relies on moving the message toward high contact rate nodes so that path explosion can occur quickly. Once path explosion occurs among high contact rate nodes, contact between one of them and the destination leads to message delivery.

8. CONCLUSION

Forwarding in pocket switched networks is challenging because of the irregular and unpredictable contact patterns between nodes. In order to understand the nature of the forwarding problem, this paper has taken the novel approach of enumerating and characterizing the set of forwarding paths that are available in such settings. More specifically we study the setting where there is high node and contact density and movement is in a restricted physical space. Our main result is the observation that for most messages there is a path explosion effect, meaning that while the optimal path may be long, there are usually a large number of nearly-optimal paths to the destination. We support this result both analytically and empirically. We present a simple model of homogeneous inter-node contact and show that path explosion is to be expected in that case. And we show empirically that path explosion does occur in our data, although in a more complicated manner due to varying contact rates across nodes. Our empirical results focus attention on the difference between high contact rate ('in') nodes and low contact rate ('out') nodes.

These results help us understand why different forwarding algorithms show quite similar performance. As a result of path explosion, different forwarding algorithms often find paths to the destination of nearly-equal duration.

Our study leaves open a number of important questions. Our understanding of the inhomogeneous case is not as precise as that of the homogeneous case; more sophisticated analytic models would be useful in that regard. Furthermore, we have not considered the cost of forwarding, and so there may be good reasons to prefer one algorithm over another even if they show similar performance. It is possible that an understanding of the path explosion phenomenon may help in understanding the relative costs of different forwarding algorithms.
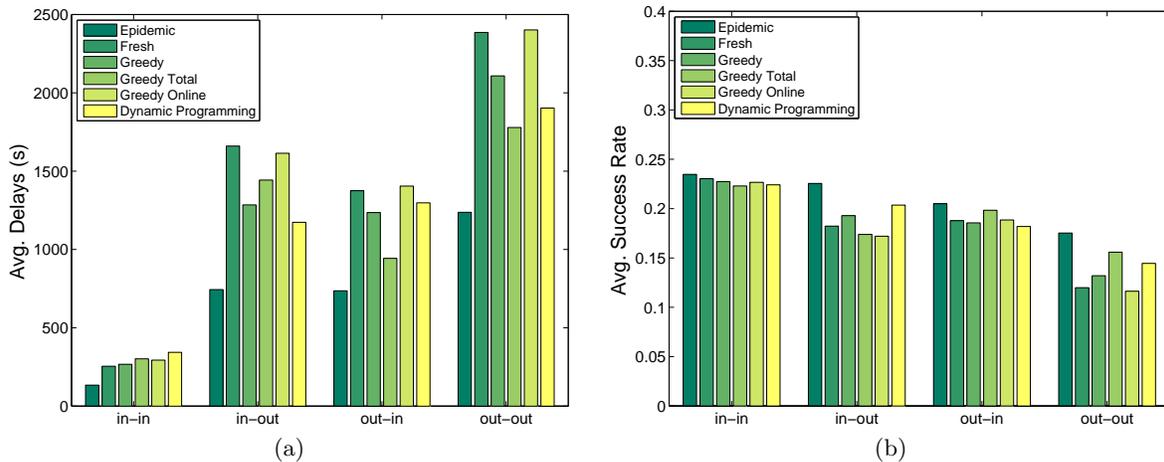
Figure 12: Performance of Forwarding Algorithms on Different Types of Source-Destination Pairs: (a) Average Delay (b) Success Rate.

9. REFERENCES

[1] BORREL, V., DE AMORIM, M. D., AND FDIDA, S. A preferential attachment gathering mobility model. *IEEE Communications Letters 9*, 10 (2005), 900–902.

[2] CAMP, T., BOLENG, J., AND DAVIES, V. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2*, 5 (2002), 483–502.

[3] CHAINTREAU, A., HUI, P., CROWCROFT, J., DIOT, C., GASS, R., AND SCOTT, J. Pocket Switched Networks: Real-world mobility and its consequences for Opportunistic Forwarding. Tech. Rep. UCAM-CL-TR-617, University of Cambridge, 2005.

[4] CHAINTREAU, A., HUI, P., SCOTT, J., GASS, R., CROWCROFT, J., AND DIOT, C. Impact of human mobility on the performance of opportunistic forwarding algorithms. In *Proceedings of IEEE INFOCOM* (2006).

[5] CONAN, V., LEGUAY, J., AND FRIEDMAN, T. The heterogeneity of inter-contact time distributions: its importance for routing in delay tolerant networks. *ArXiv Computer Science e-prints* (Sept. 2006).

[6] DUBOIS-FERRIERE, H., GROSSGLAUSER, M., AND VETTERLI, M. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *ACM MobiHoc '03* (New York, NY, USA, 2003), ACM Press, pp. 257–266.

[7] GROSSGLAUSER, M., AND TSE, D. N. C. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw. 10*, 4 (2002), 477–486.

[8] GROSSGLAUSER, M., AND VETTERLI, M. Locating mobile nodes with ease: learning efficient routes from encounter histories alone. *IEEE/ACM Trans. Netw. 14*, 3 (2006), 457–469.

[9] HUI, P., CHAINTREAU, A., SCOTT, J., GASS, R., CROWCROFT, J., AND DIOT, C. Pocket switched networks and human mobility in conference environments. In *WDTN '05* (New York, NY, USA, 2005), ACM Press, pp. 244–251.

[10] JAIN, S., FALL, K., AND PATRA, R. Routing in a delay tolerant network. In *ACM SIGCOMM '04* (New York, NY, USA, 2004), ACM Press, pp. 145–158.

[11] JONES, E. P. C., LI, L., AND WARD, P. A. S. Practical routing in delay-tolerant networks. In *WDTN '05* (New York, NY, USA, 2005), ACM Press, pp. 237–243.

[12] KURTZ, T. G. Limit theorems for sequence of jump markov processes approximating ordinary differential processes. *Journal of Applied Probability* (1971), 344–356.

[13] LINDGREN, A., DORIA, A., AND SCHELEN, O. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev. 7*, 3 (2003), 19–20.

[14] MERUGU, S., AMMAR, M., AND ZEGURA, E. Routing in space and time in networks with predictable mobility. Tech. Rep. GIT-CC-04-7, Georgia Institute of Technology, 2004.

[15] MUSOLESI, M., AND MASCOLO, C. A community based mobility model for ad hoc network research. In *REALMAN '06* (New York, NY, USA, 2006), ACM Press, pp. 31–38.

[16] NORDSTROM, E., DIOT, C., GASS, R., AND GUNNINGBERG, P. Experiences from measuring human mobility using bluetooth inquiring devices. In *MobiEval 2007: System Evaluation for Mobile Platforms* (New York, NY, USA, 2007), ACM Press.

[17] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. Single-copy routing in intermittently connected mobile networks. In *IEEE SECON '04* (San Jose, CA, USA, 2004), IEEE, pp. 145–158.

[18] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (New York, NY, USA, 2005), ACM Press, pp. 252–259.

[19] VAHDAT, A., AND BECKER, D. Epidemic routing for partially connected ad hoc networks. Tech. Rep.

CS-200006, Duke University, 2000.

[20] ZHANG, X., NEGLIA, G., KUROSE, J., AND TOWSLEY, D. Performance modeling of epidemic routing. In *Proceedings of IFIP Networking* (2006).