ABSTRACT
Intermittently connected mobile networks are wireless networks where most of the time there does not exist a complete path from the source to the destination. There are many real networks that fall into this paradigm, for example, wildlife tracking sensor networks, military networks, inter-planetary networks, etc. In this context, conventional routing schemes would fail.

To deal with such networks researchers have suggested to use flooding-based routing schemes. While flooding-based schemes have a high probability of delivery, they waste a lot of energy and suffer from severe contention which can significantly degrade their performance. With this in mind, we introduce a new routing scheme, called Spray and Wait, that “sprays” a number of copies into the network, and then “waits” till one of these nodes meets the destination. We compare this scheme to a number of different routing algorithms including an oracle-based optimal algorithm that minimizes average message delivery delay while using the lowest possible number of transmissions.

Theory and simulations show that Spray and Wait has a number of desirable properties: it is very simple to implement, it is highly scalable, it performs close to the optimal scheme, and quite surprisingly, it outperforms all existing schemes with respect to both average message delivery delay and number of transmissions per message delivered.

1. INTRODUCTION
Intermittently connected mobile networks (ICMN) are mobile wireless networks where most of the time there does not exist a complete path from a source to a destination, or such a path is highly unstable and may change or break soon after it has been discovered (or even while being discovered). This situation arises when the network is quite sparse, in which case it can be viewed as a set of disconnected, time-varying clusters of nodes. There are many real networks that fall into this paradigm. Examples include wildlife tracking and habitat monitoring sensor networks [3, 21], military networks [2], inter-planetary networks [9], nomadic communities networks [11] etc. Intermittently connected mobile networks belong to the general category of Delay Tolerant Networks [1], that is, networks were incurred delays can be very large and unpredictable.

Since in the ICMN model there may not exist an end-to-end path between a source and a destination, conventional mobile ad-hoc network routing schemes, such as DSR [20], AODV [24], etc., would fail. Specifically, reactive schemes will fail to discover a complete path, while proactive protocols will fail to converge, resulting in a deluge of topology update messages. However, this does not mean that packets can never be delivered in such networks. Over time, different links come up and down due to node mobility. If the sequence of connectivity graphs over a time interval are overlapped, then an end-to-end path might exist. This implies that a message could be sent over an existing link, get buffered at the next hop until the next link in the path comes up, and so on and so forth, until it reaches its destination.

This approach imposes a new model for routing. Routing consists of a sequence of independent, local forwarding decisions, based on current connectivity information and predictions of future connectivity information. In other words, node mobility needs to be exploited in order to deliver a message to its destination. This is reminiscent of the work in [16]. However, there mobility is exploited in order to improve capacity, while here it is used to overcome the lack of end-to-end connectivity.

Depending on the number of copies of a single message that may coexist in the network, one can define two major categories of hop-by-hop routing schemes. In single-copy schemes there’s only one custodian for each message. When the current custodian forwards the copy to an appropriate next hop, this becomes the message’s new custodian, and so on and so forth until
the message reaches its destination. On the other hand, _multiple-copy_ (or _multi-copy_) routing schemes may generate multiple copies of the same message which can be routed independently for increased robustness.

The majority of routing schemes proposed in the literature in the context of ICMNs are flooding-based, and, therefore, _multi-copy_ in nature [7, 15, 21, 23, 28]. Despite their increased robustness and low delay, flooding-based protocols consume a high amount of bandwidth and energy, as has been noted in [21, 23]. Further, under high load they suffer from severe contention, which yields a large number of drops and can significantly degrade their performance [27]. To alleviate for the large number of transmissions, researchers had suggested to only forward a copy of a packet to nodes that are “better” custodians according to some utility function [10, 12, 23, 26]. However, _utility-based_ flooding schemes have difficulty in choosing the right utility threshold. Too small a threshold and the scheme behaves like pure flooding. Too high a threshold and the delay increases significantly. When lower probabilities of delivery and larger delays are acceptable, single-copy schemes can be an efficient alternative to flooding-based schemes [26]. But in many applications this is not acceptable.

Summarizing, despite the large number of proposals, there is no existing routing scheme that both achieves low delivery delays and is energy-efficient (i.e. performs a small number of transmissions). With this in mind, in this paper we introduce a novel _multi-copy_ routing scheme called _Spray and Wait_. Spray and Wait bounds the total number of copies and transmissions per message without compromising performance. Using theory and simulations we show that: (i) under low load, Spray and Wait results in much fewer transmissions and comparable or smaller delays than flooding-based schemes, (ii) under high load, it yields significantly better delays and fewer transmissions than flooding-based schemes, and (iii) as the size of the network and the number of nodes increase, the number of transmissions per node that Spray and Wait requires in order to achieve the same performance decreases; this implies that Spray and Wait is highly scalable. We also show that Spray and Wait, using only a handful of copies per message, can achieve comparable delays to an oracle-based optimal scheme that minimizes delay while using the lowest possible number of transmissions.

In the next section we go over some existing related work. Section 3 describes a number of routing algorithms to tackle this problem, including our proposed solution, the Spray and Wait routing algorithm. Then, in Section 4 we highlight some useful known theoretical results and use them to analyze the performance of Spray and Wait. In section 5 we show how to specify the total number of copies per message that Spray and Wait should use to achieve desirable expected delays and network throughput. Simulation results are presented in Section 6, where the performance of all the strategies is compared with respect to message delivery delay and number of transmissions per message delivered. In the same section, simulation results for Spray and Wait are also compared to theoretical ones. Finally, Section 7 concludes the paper and gives some directions for future work.

## 2. RELATED WORK AND CONTRIBUTIONS

In the context of routing for intermittently connected mobile networks a number of efforts exist that mostly try to deal with application-specific problems, especially in the field of sensor networks. In [25], a number of mobile nodes performing independent random walks serve as _DataMules_ that carry data from static sensors to base stations in a sparse sensor network. The statistics of random walks are used to analyze the expected performance of the system. The idea of carrying data through disconnected parts using a virtual mobile backbone has also been used in [7, 15, 22].

In a number of other works, all nodes are assumed to be mobile and algorithms to transfer messages from any node to any other node are sought for [3, 10, 12, 16, 21, 22, 23, 28]. In [28] the concept of epidemic algorithms is applied to intermittently connected mobile networks. The proposed epidemic routing scheme is essentially flooding. Epidemic routing is used in [21] to reliably collect data from sensor nodes attached to zebras. Additionally, a simple method to take advantage of the history of past encounters is implemented in order to reduce the overhead of epidemic routing and improve its performance. The concept of history based or probabilistic routing is further elaborated in [23]. There it is shown that using the age of last encounter with a node, when making a forwarding decision, results in superior performance than flooding. Similar results have been found in the context of regular, connected, wireless networks in [12].

The authors in [10] generalize the concept of history-based/probabilistic routing to that of utility-based routing, where the utility of a node for a destination may be a function of the history of encounter, frequency of encounter, node speed, scheduled encounter, node resources, etc. Additionally, they propose a periodic request and reply method, similar to that used in DSR [20] and [12], to query potential next hops. This is a different model from the encounter-based approach of [21, 23, 28].

Despite the number of existing approaches, the majority of multi-copy schemes is based on epidemic-routing or some other form of controlled flooding [7, 15, 21, 23, 28]. These schemes are plagued by the shortcomings of flooding-based schemes, as mentioned in the introduction. Utility-based schemes [10, 12, 23] can only reduce the transmission overhead of epidemic routing with a significant penalty on delivery delay. Finally, single-copy schemes [25, 26] may not present desirable solutions for applications that require high probabilities of delivery and low delays. Our scheme, Spray and Wait, manages to significantly reduce the transmission over-
head of flooding-based schemes and have better performance with respect to delivery delay in most scenarios, which is particularly pronounced when contention for the wireless channel is high. Further, it does not require the use of any network information, not even that of past encounters.

Also, no theoretical analysis exists on the performance of flooding-based and utility-based schemes in intermittently connected mobile networks. To the best of our knowledge, the only prior analytical work is on direct transmission schemes [25] and oracle-based schemes [26]. In particular, the authors in [26] have used the theory of random walks on graphs [5] to generalize the direct transmission analysis of [25] and analyze the expected delay of an optimal “oracle-based” algorithm, that is assumed to have complete knowledge of all future movement. This delay is a lower bound on the delay achievable by any scheme, and equals that of flooding-based schemes under the assumption of no contention. In this paper we use a similar methodology to that in [26] to analyze and optimize the performance of Spray and Wait, and compare it to that of the oracle-based algorithm. Unlike the oracle-based algorithm, Spray and Wait can be easily implemented in practice. We also provide analytical methods to compute the number of copies per message that Spray and Wait requires to achieve a target average message delivery delay and a desirable network throughput. Finally, we present a method to estimate network parameters online, like the total number of nodes, in order to be able to optimize Spray and Wait when these are unknown.

3. ROUTING STRATEGIES

In this section we explore the solution space of the ICMN routing problem. We discuss in more detail a range of possible routing approaches, some existing, and other proposed by us, with a focus on multi-copy ones. We identify the strengths and shortcomings of each of these general techniques, and based on that we propose a set of design goals for an efficient multi-copy routing protocol. We then present Spray and Wait, a simple yet very efficient routing protocol that meets these goals and outperforms both single and multi-copy routing techniques.

Our problem consists of a number $M$ of nodes moving independently according to some mobility model. Additionally, we assume that the network is disconnected at most times, and that transmission is faster than node movement (a reasonable assumption with modern wireless devices). Finally, each node maintains a timer for every other node in the network, which records the time elapsed since the two nodes last encountered each other (i.e. came within transmission range). These timers are the only information available to a node regarding the network (i.e. no location info, etc.) and are similar to the age of last encounter in [12]. Note that Spray and Wait does not make use of these timers. We make this assumption here only to accommodate utility-based schemes proposed by others.

The function of every routing algorithm in this context is to decide under what circumstances a node currently holding a message copy will hand its copy over, or spawn and forward a duplicate copy to another node it encounters. We first present two algorithms whose performance, in terms of expected delivery delay, delimits the achievable performance by any implementable single or multi-copy algorithm.

3.1 Direct Transmission

The simplest possible routing scheme imaginable is the following: a node $A$ forwards a message to another node $B$ it encounters, only if $B$ is the message’s destination. This scheme is obviously a single-copy one. It has an unbounded delivery delay [16], but has the advantage of performing only a single transmission per message. It has been considered in some previous work [16, 25, 26], and its expected delivery delay is an upper bound on the expected delay of any routing scheme. It will therefore serve as our baseline.

3.2 An “Oracle-based” Optimal Algorithm

Direct transmission minimizes the number of transmissions, but has the highest expected delivery delay. The algorithm that minimizes the expected delivery delay must be aware of all future movement, and, thus, it is an “oracle-based” algorithm [26]. A die is thrown at the beginning of each scenario, whose outcome decides the full trajectory of all nodes. The algorithm then takes as input all these trajectories, and computes the optimal set of forwarding decisions (i.e. time and next hop), which delivers a message to its destination in the minimum amount of time.

The “oracle-based” algorithm cannot be implemented in reality. It provides an offline solution to an inherently online problem. However, its delay will serve as a lower bound on the performance of all routing strategies. Finally, notice that under the assumption of no contention (i.e. infinite buffer capacity and bandwidth), epidemic routing will find the same paths as the oracle-based scheme, and thus achieve minimum expected delay.

3.3 Flooding-based and Utility-based Algorithms

Epidemic routing [28] extends the concept of flooding in intermittently connected mobile networks. It is one of the first schemes proposed to enable message delivery in such networks. Each node maintains a list of all messages it carries, whose delivery is pending. Whenever it encounters another node, the two nodes exchange all messages that they don’t have in common. This way, all messages are eventually “spread” to all nodes, including their destination (in an “epidemic” manner). A Time-To-Live (TTL) field is often used, after which a message is discarded. This way message copies are prevented from occupying valuable buffer space, even after some copy has probably reached its destination.

Although epidemic routing finds the same paths as the
optimal scheme under no contention, it is very wasteful of network resources. Every message is received by every node in the network. Thus, transmissions grow fast with the number of nodes $M$ making it a very energy-consuming scheme. Furthermore, in most real wireless networks, wireless bandwidth and memory are scarce resources. In such networks, epidemic routing creates a lot of contention for the limited buffer space and network capacity, resulting in many message drops and retransmissions. This can have a detrimental effect on performance, as has been noted earlier in [23, 27], and will also be shown in our simulation results. As most flooding-based schemes, epidemic routing scales pretty badly as the traffic load or node density increases. Its performance in terms of total transmissions, delivery ratio and delay, is rather poor under medium to high traffic or in less sparse networks.

One simple approach to reduce the overhead of flooding and improve its performance is the following [27]: when a node encounters another node, it forwards a copy with probability $p \in (0, 1]$. We shall call this scheme Randomized Flooding. Obviously, for $p = 1$ this scheme becomes epidemic routing.

A different, more sophisticated approach to routing in intermittently connected networks is that of Utility-based or Probabilistic Routing. This scheme has been shown to improve the performance of epidemic routing in some scenarios [23]. In Utility-based routing each node maintains a utility function for every other node in the network, based on the last encounter timers described earlier. The respective timers essentially carry indirect information about relative node locations, which get diffused through nodes’ mobility. Consequently, the utility function reflects the probability of a node encountering another node in the near future. Here a node will forward a message copy to another node it encounters only if this other node has a higher utility for the message’s destination by at least some pre-specified value $U_{th}$. As was shown in [26], such utility-based forwarding that uses a monotonic function of the timer value as its utility function results in better forwarding decisions than randomized forwarding.

Nevertheless, Utility-based routing is in nature still flooding-based. Despite making better forwarding decisions than randomized and epidemic routing, it is faced with the following dilemma. In a large network, it is expected that the destination is going to be far from the source, when source-destination pairs are chosen randomly. In the neighborhood of the source, utility values for the destination will be low. If a large $U_{th}$ is used the total number of transmissions is significantly reduced, but the scheme suffers from a “slow-start” phase: the source has to wait a long time until it encounters a good next hop. On the other hand, if a very small $U_{th}$ is chosen, the scheme degenerates to epidemic routing.

Finally, in order to reduce the number of transmissions even more, one could envision a simple modification to Utility-based Routing, which we shall call Constrained Utility-based Routing. This scheme trades off delivery delay to further reduce the number of transmissions. Specifically, each node is allowed to only forward a message it receives to a fixed number of next hops. Setting this number to a very high value is equivalent to the original utility-based scheme, while setting it to one corresponds to single-copy utility-based routing, which suffers from large delays as any single-copy scheme.

3.4 Single-copy Routing - “Seek and Focus”

When transmission cost is an important factor in routing (e.g. energy-constrained or bandwidth-constrained wireless applications), single-copy routing algorithms can be an attractive option. Such schemes perform significantly fewer transmissions than flooding-based ones, but incur a high delay (and often robustness) penalty. We have included in our discussion and simulations the champion single-copy scheme from [26], called “Seek and Focus” routing, in order to obtain some insight on the relative performance between single and multi-copy routing. Seek and Focus initially looks around and quickly finds some node having a high utility for the message’s destination. In other words, it initially seeks for a good lead towards the destination. When it finds one, it switches to utility-based forwarding, in order to efficiently follow that lead (i.e. focuses its search).

3.5 “Spray and Wait” Routing

Based on the previous exposition, we can identify a number of desirable design goals for a routing protocol in intermittently connected mobile networks. Specifically, an efficient routing protocol in this context should:

- perform significantly fewer transmissions than epidemic and other flooding-based routing schemes, under all conditions.
- generate low contention, especially under high traffic loads.
- achieve a delivery delay that is better than existing single and multi-copy schemes, and close to the optimal.
- be highly scalable, that is, maintain the above performance behavior despite changes in network size or node density.
- be simple and require as little knowledge about the network as possible, in order to facilitate implementation.

To this end, we propose a novel routing scheme, called Spray and Wait that is simple yet efficient, and meets the above goals, as we will demonstrate in the next sections. Spray and Wait routing decouples the number of copies generated per message, and therefore the number of transmissions performed, from the network size. It consists of two phases:

\[ p \in (0, 1] \]
DEFINITION 3.1 (Spray and Wait). Spray and Wait routing consists of the following two phases:

- spray phase: for every message originating at a source node, \( L \) message copies are initially spread – forwarded by the source and possibly other nodes receiving a copy – to \( L \) distinct “relays”. (Details about different spraying methods will be given later.)

- wait phase: if the destination is not found in the spraying phase, each of the \( L \) nodes carrying a message copy performs direct transmission (i.e. will forward the message only to its destination).

Spray and Wait combines the speed of epidemic routing with the simplicity and thriftiness of direct transmission. It initially “jump-starts” spreading message copies in a manner similar to epidemic routing. When enough copies have been spread to guarantee that at least one of them will find the destination quickly (with high probability), it stops and let’s each node carrying a copy perform direct transmission. In other words, Spray and Wait could be viewed as a tradeoff between single and multi-copy schemes. Surprisingly, as we shall shortly see, its performance is better with respect to both number of transmissions and delay than all other practical single and multi-copy schemes, in most scenarios considered.

The above definition of Spray and Wait leaves open the issue of how the \( L \) copies are to be spread initially. A number of different “spraying” heuristics can be envisioned. For example, the simplest way is to have the source node forward all \( L \) copies to the first \( L \) distinct nodes it encounters. A slightly better way would be to have the source forward a few of the \( L \) copies to a few relays, and then have the relays further spread the remaining copies. This way many relays look in parallel for more nodes to receive new copies, making the spraying process faster. As \( L \) grows larger, the sophistication of the spraying heuristic has an increasing impact on the delivery delay of the spray and wait scheme. We shall deal with the issue of optimal spraying in Section 4.

4. PERFORMANCE ANALYSIS

In this section we will analyze the performance of Spray and Wait. Throughout this section we will be making the following assumptions:

- A.i \( M \) nodes perform independent random walks on a \( \sqrt{N} \times \sqrt{N} \) 2D torus (finite lattice). This model has been used in [25, 26] and we choose it here for its analytical tractability.

- A.ii Each node can transmit up to \( K \geq 0 \) grid squares away, where \( K/\sqrt{N} \) is much smaller than the value required for connectivity [18]. We use Manhattan distance \( d_{ab} = |a_x - b_x| + |a_y - b_y| \) to measure proximity between two positions \( a \) and \( b \) (or between two nodes).

- A.iii There is no contention in the network. In other words, buffer space is infinite, and any communicating pair of nodes does not interfere with any other pair communicating at the same time.

In most theorems and proofs, we use the same notation as in [5]. However, we define here a couple of quantities that we will use repeatedly:

i. \( E_i T_j \) denotes the expected hitting time until a walk starting at position \( i \) first arrives at position \( j \) (\( j \) can be replaced by a subset of states \( A \)); on a symmetric graph (like a 2D torus), this quantity only depends on the distance \( d_{ij} \) between \( i \) and \( j \) and we denote it as \( ET(d) \).

ii. \( E_x T_j \) denotes the expected hitting time until a walk starting from the stationary distribution reaches \( j \); on a symmetric graph, this quantity is independent of \( j \), and we denote it as \( ET \).

iii. \( EM_{ij} \) denotes the expected time until two independent random walks, starting at positions \( i \) and \( j \), respectively, first meet each other.

iv. \( EM \) denotes the expected time until two independent random walks, starting from the stationary distribution, first meet each other. On a symmetric graph, \( EM \) is just half the respective hitting time \( ET \) [5].

For completeness of presentation, we first highlight some useful results and the associated methodology from [26], which are used to prove several properties of Spray and Wait. These results concern the expected delay of direct transmission and oracle-based schemes, which as we explained earlier establish upper and lower bounds, respectively, on the performance of any routing scheme.

4.1 Direct Transmission – An Upper Bound on Delay

We first state here some known results on a \( \sqrt{N} \times \sqrt{N} \) torus, regarding the expected hitting time of a single random walk and the meeting times of independent random walks [5, 4]. When \( K = 0 \) these results give the expected delivery delay of the direct transmission scheme \( ED_{\theta(K=0)} \), and thus an upper bound on the expected delay of any routing scheme.

**Lemma 4.1.** Let independent random walks be performed on a \( \sqrt{N} \times \sqrt{N} \) torus. Then:

i. \( ET = cN \log N \), where \( c = 0.34 \). (This result is valid as \( N \to \infty \). However, according to [4, 25] this result becomes quite accurate for \( N > 25 \).)
The hitting time probability distribution function can be approximated by an exponential function:
\[ P(T > t) = \exp \left( -\frac{t}{cN \log N} \right). \]

iii. \( ED_{dt(K=0)} = EM = \frac{1}{2} ET \).

**Proof.** See [5]. \( \square \)

Let us now assume that every node has a transmission range \( K \geq 1 \). This means that any node that comes within distance \( d \leq K \) from a node \( i \), can communicate with it. Lemma 4.2 extends the expected hitting and meeting time results of Lemma 4.1 to the general case where \( K \geq 1 \).

**Lemma 4.2.** Let us pick uniformly a position \( j \) on an \( \sqrt{N} \times \sqrt{N} \) torus. Let further \( T(K) \) denote the hitting time until a random walk, starting from the stationary distribution, comes within range \( K \geq 1 \) of some position \( j \), and let \( M(K) \) denote the meeting time of two such walks. Then, for large \( N \),
\[ i. \quad ET(K) = N \left( c \log N - \frac{2K+1 - K - 2}{2K - 1} \right), \quad (1) \]
\[ ii. \quad ED_{dt} = EM(K) = \frac{1}{2} ET(K). \quad (2) \]

**Proof.** (sketch of proof, see [26] for details) Let \( A(K) \) denote the subset of all positions \( a \), such that \( |x_a - x_j| + |y_a - y_j| = K \). This is the border of the area that is within range of \( j \). Consequently, the expected time until a random walk (starting from the uniform distribution) comes within range of \( j \) equals the expected hitting time \( E_x T_A \) at set \( A(K) \). Now, it is evident that \( E_x T_A = E_x T_j = E_{\pi(A)} T_j \). Lemma 4.1 implies that \( E_x T_j = ET = cN \log N \). Also, using some combinatorics and the theory of random walks on graphs, one can calculate \( E_{\pi(A)} T_j \) as \( \left( \frac{2K+1 - K - 2}{2K - 1} \right) N \), which gives us Eq.(1). The proof of ii. is a straightforward extension of the proof for Lemma 4.1. \( \square \)

### 4.2 “Oracle-based” Optimal Algorithm – A Lower Bound on Delay

The following coloring problem analog is used to analyze the expected performance of the optimal algorithm: The randomly chosen source node is colored red and all other nodes (including the randomly chosen destination) are colored blue; Whenever a red node encounters a blue node, the latter is colored red too.

It is evident that the expected time until the destination node is colored red is equal to the expected message delivery time of the optimal algorithm. It is also evident by this coloring analog why epidemic routing will find the same paths (and therefore have the same delivery delay) as the oracle scheme, when buffer capacity and bandwidth are unlimited.

The following theorem calculates the expected time until the destination node is colored red. It uses the meeting time of direct transmission in order to calculate the expected time until a new node is colored red, and then estimates the total number of coloring steps necessary.

**Theorem 4.1.** When every node has a transmission range of \( K \geq 0 \), the expected message delivery time of the optimal algorithm is given by
\[ ED_{opt} = \frac{NH_{M-1}}{(M-1)} ED_{dt}, \quad (3) \]
where \( H_n \) is the \( n^{th} \) Harmonic Number, i.e., \( H_n = \sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n) \).

**Proof.** (sketch of proof, see [26] for details) Let us assume that at some time instant we have \( m \) red nodes and \( M-m \) blue nodes, and let us pick a red node \( i \) and a blue node \( j \). According to Lemma 4.2, the meeting time \( M_{ij} \) of nodes \( i \) and \( j \) is exponentially distributed with average \( ED_{dt} \). Let \( M_i \) denote the meeting time of \( i \) with any of the \( M-m \) blue nodes, that is, \( M_i = \min_j (M_{ij}) \). Finally, let \( M^{(m)} \) denote the time until any of the red nodes meets any of the blue ones, when there are \( m \) total red nodes. Then, \( M^{(m)} = \min_i (M_i) = \min_{ij} (M_{ij}) \). However, all \( M_{ij} \) are IID exponential random variables with average \( ED_{dt} \). Thus, \( M^{(m)} \) is also an exponential random variable with average \( \frac{ED_{dt}}{m(M-m)} \). Since we start with 1 red node, the expected time until all nodes are colored red is given by
\[ ED_{dt} \sum_{m=1}^{M-1} \frac{1}{m(M-m)} \].

Finally, the destination may be colored red in any of the \( M-1 \) total coloring steps with equal probability. Consequently, averaging over the number of coloring steps needed until the destination is colored red we get Eq.(3). \( \square \)

### 4.3 Performance of Spray and Wait Routing

In the Spray and Wait routing algorithm there are \( L \) copies that need to be spread initially to \( L \) different relays. Irrespective of how the \( L \) copies are sprayed, after all copies are distributed, each of the \( L \) relays will independently look for the destination to directly deliver the message, if the latter has not been found yet. In other words, the delay of the wait phase is independent of the spraying method. We compute it in the following Lemma.

**Lemma 4.3.** Let \( EW \) denote the expected duration of the “wait” phase, if needed. Then, for large \( N \), \( EW \) is the same for every Spray and Wait algorithm, regardless of the way the \( L \) copies are spread, and given by
\[ EW = \frac{ED_{dt}}{L}. \quad (4) \]
Proof. The time until one of the relays finds the destination is the minimum of $L$ independent and exponentially distributed random variables, with average $ED_{dt}$. Thus, the expected value of this minimum is equal to $ED_{dt}/L$. □

4.3.1 Source Spray and Wait

Unlike the expected duration of the wait phase, the duration of the spray phase largely depends on the way the $L$ copies are spread. The simplest way is to have the source distribute $L - 1$ copies to the first $L - 1$ nodes it encounters, and keep one for itself. We shall call this the Source Spray and Wait routing algorithm, hereafter. We derive the exact delay of the Source Spray and Wait in Theorem 4.2. However, the exact delay result is not in closed form, making it difficult to theoretically compare the performance of Source Spray and Wait to that of the optimal scheme. For this reason, in the following lemma we first derive a tight upper bound that is in closed form. Its proof uses a similar methodology to that of Theorem 4.1.

Lemma 4.4. The expected delay of Source Spray and Wait routing is upper bounded by

$$ (H_{M-1} - H_{M-L}) ED_{dt} + \frac{M - L}{M - 1} EW, $$

where again $H_n$ is the $n^{th}$ Harmonic number.

Proof. Assume that, at some time instant, $i$ of the $L$ copies have already been spread, that is, there are $i$ nodes, including the source node, carrying a message copy (recall that in the coloring analog this means that $i$ nodes are red already). Since only the source can forward another copy, the expected time until another message copy is distributed is equal to the time until the source meets one of the $M - i$ remaining nodes, that is, $ED_{dt}/M - i$. (Note that the difference here with the optimal algorithm is that only the source node can color a node other than the destination red, and that only up to $L$ nodes other than the destination can be colored red in total.) Hence, the expected time until $L - 1$ different relays are encountered equals

$$ \sum_{i=1}^{L-1} \frac{ED_{dt}}{M - i} = \left(H_{M-1} - H_{M-L}\right) ED_{dt}. $$

This is the time until $L$ message copies, including that of the source, are spread among the $M$ nodes.

Now, let $P_i$ denote the probability that the destination is the $i^{th}$ node to receive a message copy, and let $t_i$ denote the expected time until the $i^{th}$ copy is given by the source. Then, the expected delivery time is given by $\sum_{i=1}^{L} P_i t_i$. Clearly,

$$ \sum_{i=1}^{L} P_i t_i \leq \sum_{i=1}^{L-1} P_i t_{i+1} + P_L t_L. $$

Also, $t_{L-1} = \sum_{i=1}^{L-1} \frac{ED_{dt}}{M - i}$, $t_L = t_{L-1} + EW$, and the probability $P_i$ that the destination was not among the $L - 1$ initial relays is equal to $\frac{M - i}{M - L}$. This is the probability that a “wait” phase will be necessary. Substituting $t_{L-1}$, $t_L$, and $P_L$ in the above equation, and noting that $\sum_{i=1}^{L} P_i = 1$, we get that the expected delay is bounded by

$$ \sum_{i=1}^{L-1} \frac{ED_{dt}}{M - i} + \frac{M - L}{M - 1} EW. $$

□

This bound is tight for a small $L/M$ ratio, but becomes pessimistic as this ratio grows larger. This is because the bound basically includes the full time until all copies are spread, regardless of whether the destination is found in some previous step. The following theorem provides an exact solution for the expected delivery time of the Source Spray and Wait algorithm. It defines a system or recursive equation that calculates the (expected) remaining time after $i$ copies have been spread, in terms of the time until the $i + 1^{th}$ is distributed, plus the remaining time thereafter.

Theorem 4.2. Let $ED_{sw(src)}(L)$ denote the expected delay of the Source Spray and Wait algorithm, when $L$ copies are spread per message. Let further $ED(i)$ denote the expected remaining delay for the same scheme, after $i$ message copies have been spread. Notice that $ED_{sw(src)}(L) \equiv ED(1)$. Then, $ED_{sw(src)}(L)$ can be calculated by the following system of recursive equations:

$$ ED(i) = \begin{align*}
ED_{di}(i + 1), & \quad i \in [1, L - 1], \\
ED(L) & = EW = \frac{ED_{dt}}{L}.
\end{align*} $$

Proof. This proof uses again as building blocks the results of Lemma 4.1, the independence of node movements, and the fact that the exponential distribution is memoryless. When a message is generated at a source node, we can calculate the expected time until any of the remaining $M - i$ nodes is encountered as $ED_{di}/M - i$. If the node encountered is the destination, an event occurring with probability $\frac{1}{M - i}$, the algorithm stops. Otherwise, with probability $\frac{M - i}{M - 1}$, the source hands over a second copy to the node encountered, and an expected time $ED(2)$ remains until the destination is found. This gives the first equation in the system.

Let us now look at the time instant when the $i^{th}$ copy $(i < L)$ has just been forwarded to a node other than the destination. At that time, there are $i - 1$ relays that will only forward their copy to the destination, and the source that can give a copy either to the destination or
another node among the $M - i$ remaining, depending on who is encountered first. Therefore, the time until any of the nodes with a message copy encounters any of the nodes without one is equal to $\frac{ED_\mu}{n(M-1)}$. If that node is the destination (with probability $\frac{1}{M-1}$) the message gets delivered. Otherwise (with probability $\frac{M-1}{M-1}$) the algorithm continues, performing one of the following: a) with probability $\frac{1}{n}$ it is the source that encountered this other node, and therefore hands it over the $i+1$th copy; an expected time $ED(i+1)$ remains till delivery; b) with probability $\frac{n-1}{M-1}$ it was one of the other nodes carrying a message copy that encountered a new node. Since these relays only forward their message copy to its destination, nothing happens, and the remaining time is still $ED(i)$.

Finally, as we saw in Lemma 4.3 the remaining time after $L$ copies are spread in the network is given by Eq.(4).

### 4.3.2 Optimal Spray and Wait

Let us call a node that currently carries a message copy (either source or relay) as “active”, if it can spawn and forward an additional copy to a node it encounters, during the spraying phase. Let us further define a specific spraying method/heuristic as “opportunistic”, if an active node during the spray phase will always forward a copy to a new node it encounters (i.e. a node without a copy). It is easy to see that, when node movements are IID, opportunistic schemes are faster than non-opportunistic ones. The Source Spray and Wait algorithm is an opportunistic algorithm, where only the source is active, during the spraying phase. It is easy to see that the Source Spray and Wait routing has the largest expected delivery delay among all opportunistic spray and wait schemes. Hence, note that Eq.(5) is also an upper bound on the delay of any opportunistic spray and wait scheme.

We can now look into the issue of optimal spraying. From our previous discussion, it is evident that an optimal spraying method is one at every step of which there are as many “active” relays as possible. Let us define the following spray and wait algorithm.

**Definition 4.1 (Binary Spray and Wait).** In Binary Spray and Wait routing the $L$ message copies are spread according to the following rules: the source of a message starts with $L$ copies; Any node is considered active if it owns more than one copy, and switches to direct transmission if it has exactly one copy; An active node $A$ that has $n > 1$ message copies, and encounters another node $B$ (with no copies), hands over to $B$ $\lceil n/2 \rceil$ and keeps $\lfloor n/2 \rfloor$ for itself, if $n$ is even, or $\lfloor n/2 \rfloor + 1$, if $n$ is odd.

The following theorem states that the Binary Spray and Wait scheme is optimal, when node movement is IID.

**Theorem 4.3.** When all nodes move in an IID manner, Binary Spray and Wait routing is optimal, that is, has the minimum expected delay among all spray and wait routing algorithms.

**Proof.** Let us define a spraying algorithm in terms of a function $f : N \rightarrow N$ as follows: when an active node with $n$ copies encounters another node, it hands over to it $f(n)$ copies, and keeps the remaining $1 - f(n)$. Any spraying algorithm (i.e. any $f$) can be represented by the following binary tree with the source as its root: starting from the root, assign the root node a value of $L$, and create a right child with a value of $1 - f(L)$ and a left one with a value of $f(L)$. Repeat the procedure with the children nodes, their children, etc., until a node has a value of 1 making it a leaf node.

Let us initially color the root node red, and all other nodes blue. An active node corresponds to a red non-leaf node in the tree. Whenever an active node encounters a new node, its children are painted red, and the node itself is painted back blue. It is evident that the spraying process stops when all leaf nodes are painted red. Since all nodes move independently, the exact sequence of nodes becoming red is random. Nevertheless, on the average, all tree nodes at the same level are colored red in parallel. Now, as we saw earlier, the higher the number of active nodes $X_i$ when $i$ copies are spread, the smaller $ED(i)$. Also, the total number of tree nodes is fixed $(2^{1+\log L} - 1)$ for any spraying function $f$. Therefore, it is easy to see that the tree structure that has the maximum number of nodes at every level, also has the maximum number of active nodes (on the average) at every step. This tree is the balanced tree, and corresponds to the Binary Spray and Wait routing scheme.

It is possible to derive the exact delay for Binary Spray and Wait, but it involves meticulous calculations that do not offer any interesting insight. Therefore, Theorem 4.4 derives a tight lower bound for the Binary Spray and Wait scheme, using a similar system of recursive equations as Theorem 4.2. This is also a lower bound for any spray and wait algorithm.

**Theorem 4.4.** Let $ED_{\text{sw} (\text{opt})}(L)$ denote the expected delay of the Binary Spray and Wait algorithm, when $L$ copies are spread per message. Let further $ED(i)$ denote the expected remaining delay of the same scheme, after $i$ message copies have been spread. Then, $ED(1) \leq ED_{\text{sw} (\text{opt})}(L)$, where $ED(1)$ can be calculated by the following system of recursive equations:
5. OPTIMIZING SPRAY AND WAIT TO MEET PERFORMANCE CONSTRAINTS

We have so far calculated the expected delay of spray and wait routing as a function of the number of copies used. By definition, most ICMN networks are expected to operate in stressed environments and by nature be delay tolerant. Nevertheless, in many situations the network designer or the application itself might still impose certain performance requirements on the protocols (e.g. maximum delay, maximum energy consumption, minimum throughput, etc.). For example, a message sent over an ICMN of handhelds in a campus environment, notifying a number of peers about an upcoming meeting, would obviously be of no use if it arrives after the meeting time. It is of special interest therefore to examine how Spray and Wait can be tuned to achieve the desired performance in a specific scenario.

In this section we analyze how to choose L (i.e. the number of copies used) in order for Spray and Wait to achieve a specific expected delay, and per node throughput. Note that the issue of energy dissipation is directly tied to the number of copies L used by Spray and Wait, since Spray and Wait performs exactly L transmissions. We also examine how to estimate important network parameters in practice, like the number of nodes M, in order to be able to choose the right L in unknown network environments. Finally, after having optimized Spray and Wait, we evaluate its scalability as the size of the network increases.

5.1 Choosing L to Achieve a Required Expected Delay

Let us assume first that there is a specific delivery delay constraint to be met. This might be for example a maximum expected delay dictated by the application, as in the case of the meeting notification message. It is reasonable to assume that this delay constraint is expressed as a factor a times the optimal delay $ED_{opt}$ ($a > 1$), since as we showed this the best that any routing protocol can do.

Lemma 5.1. The minimum number of copies $L_{min}$ needed for Spray and Wait to achieve an expected delay at most $aED_{opt}$ is independent of the size of the network N and transmission range K, and only depends on a and the number of nodes M.

Proof. The delay requirement can be expressed as $ED(1) \leq aED_{opt} = \frac{H_{M-1}ED_{dt}}{M-1}$. If we replace this in Theorem 4.2 or Theorem 4.4, it is easy to see that $ED_{dt}$ cancels from all equations and the remaining terms are functions of M, a, and L only. The same holds for the upper bound of Eq.(5).

The required number of copies $L_{min}(M)$ for (optimal) Spray and Wait to achieve a desired expected delay can be calculated in any of the following three ways: (i) solve the system of equations of Theorem 4.4 for increasing $L$, until $ED_{sw(opt)} < aED_{opt}$, or (ii) solve the upper bound equation Eq.(5) for $L$, by letting $ED_{sw} = aED_{opt}$, and taking $[L]$, or (iii) approximate the harmonic number $H_{M-L}$ in Eq.(5) with its Taylor Series terms up to second order, and solve the resulting third degree polynomial:

$$(H_{M-1}^{3}-1.2)L^{3} + (H_{M-1}^{2} - \frac{\pi^{2}}{6})L^{2} + \left(a + \frac{2M-1}{M(M-1)} \right) L = \frac{M}{M-1}.$$ 

where $H_{n}^{r} = \sum_{i=1}^{n} \frac{1}{i^{r}}$ is the $n^{th}$ Harmonic number of order r.

Method (i) is obviously the most accurate one. However, it is also the most cumbersome. Since the upper bound of Eq.(5) is tight for small $L/M$ values, if the delay constraint $a$ is not too tight, we can use method (ii) or (iii) to quickly get a good estimate for $L_{min}$.

In Table 1 we compare results for $L_{min}$, as calculated with each of these three methods for different values of $a$. We assume the number of nodes M equals 100. 'N.A.' stand for 'Non Available' and means that such a low delay value is never achievable by the bound. As can be seen in this table the L found through the approximation is quite accurate when the delay constraint is not too stringent.
Table 1: minimum \( L \) to achieve expected delay

<table>
<thead>
<tr>
<th>( a )</th>
<th>1.5</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>recursion</td>
<td>21</td>
<td>13</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>bound</td>
<td>N.A.</td>
<td>N.A.</td>
<td>11</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>taylor</td>
<td>N.A.</td>
<td>N.A.</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

5.2 Choosing \( L \) to Control Network Throughput

It is also interesting to examine how to choose \( L \) in order to control how much traffic per node can be carried over by the network, when source-destination pairs are randomly chosen. The capacity of regular (connected) wireless networks has been thoroughly studied (see, for example, [16, 17]). However, to the best of our knowledge, there exists no capacity analysis for ICMNs, i.e., networks that are not connected with probability one at all times. Due to lack of space, we will not present here a rigorous capacity analysis for such networks. Instead, we will give a couple of qualitative arguments along the lines of [16], in order to show how to choose \( L \) to achieve a specific asymptotic capacity behavior, when Spray and Wait is used.

The basic argument of the proof in [16] is the following: At every time slot each node generates a new message that it transmits to a neighboring node, which then becomes a ‘relay’ for this message. The relay then will only forward this message to its destination. If the network is connected with probability one, then it is guaranteed that there will always exist a relay node within range to receive a newly generated message. Also, at equilibrium (after some time has passed) each relay node will have gathered enough messages to guarantee that, with high probability, a destination for a message in its queue will be within range, at every time unit. Consequently, one message per node will be delivered per time slot implying a capacity of \( \Theta(1) \) per node per time unit. Note that a two-phase schedule is needed to ensure that transmissions by message sources do not collide with transmissions by relays. This however does not change the above asymptotic result.

In the ICMN case the above argument only brakes in that a source node (relay) does not have a relay (destination) within range at every time unit. However, we know that on average a node sees some other node every \( E D_{\text{av}}(M-1) \) time units. Consequently, each node will be able to forward at most \( M-1 \) messages per unit time. Assuming the transmission range \( K \) is fixed, the maximum per node throughput in an ICMN is then \( \Theta \left( \frac{M}{N \log(N)} \right) \) (see Lemma 4.2). Clearly, \( M < N \) for the network to be disconnected. Therefore, the resulting capacity is always smaller than \( \Theta(1) \), the capacity achieved by the single-relay scheme in [16] for connected networks.

It is easy to see that a throughput–delay tradeoff exists here, as in the case of connected networks [14]. If \( L \) copies are used per message in Spray and Wait, only a percentage \( \frac{1}{L} \) of the total network capacity will be available for each end-to-end message delivered, resulting in a per node and per time unit sustainable throughput equal to \( \Theta \left( \frac{M}{L N \log(N)} \right) \) (i.e. decreasing with \( L \)). We saw also that epidemic routing has the smallest expected delay under no contention. However, given a large enough TTL it uses \( \Theta(M) \) copies per message, and thus has the smallest throughput \( \Theta \left( \frac{1}{N \log(N)} \right) \).

Remark: As is already evident from the previous discussion the goals of delay and energy minimization, and throughput maximization are conflicting goals. It would therefore be interesting to see how one could jointly optimize Spray and Wait for all three goals. However, coming up with appropriate joint optimization functions is more of an application-specific task, and goes beyond the scope of this paper.

5.3 Estimating \( L \) when Network Parameters are Unknown

Throughout the previous analysis we’ve assumed that network parameters, like the total number of nodes \( M \) and size of the network \( N \), are known. This assumption might be valid in some networks operated by a single authority. Nevertheless, in many envisioned ICMN applications, either or both \( M \) and \( N \) might be unknown. For example, a user that uses his PDA to exchange text messages over a low cost ICMN network formed by similar users, may not know the number of other such users out there or there geographical spread, at that specific time.

Furthermore, even in situations where the initial number of nodes deployed is known, this number might change throughout the lifetime of the network, due to nodes dying out or new nodes being deployed independently. In order to make Spray and Wait equally efficient in such scenarios as well, we would like to produce and maintain good estimates of necessary network parameters, and adapt \( L \) accordingly. Specifically, the analysis of the previous section indicated that only an estimate of the number of nodes \( M \) is required, in most situations.

This problem is difficult in general. A straightforward way to estimate \( M \) would be to count unique IDs of nodes encountered already. However, this method requires a large database of node IDs to be maintained in large networks, and a lookup operation to be performed every time any node is encountered. Furthermore, although this method converges eventually, its speed depends on network size and could take a very long time in large disconnected networks. However, if we assume that nodes perform independent random walks, we can produce an estimate of \( M \) by taking advantage of inter-meeting time statistics. Specifically, let us define \( T_1 \) as the time until a node (starting from the stationary distribution) encounters any other node. We know from
Section 4 that \( T_1 \) is exponentially distributed with average \( T_1 = ED_{dt}/(M - 1) \). Furthermore, if we similarly define \( T_2 \) as the time until two different nodes are encountered, then the expected value of \( T_2 \) equals \( ED_{dt} \left( \frac{1}{M-1} + \frac{1}{M-2} \right) \). Cancellation \( ED_{dt} \) from these two equations we get the following estimate for \( M \):

\[
\hat{M} = \frac{2T_2 - 3T_1}{T_2 - 2T_1}.
\]

Estimating \( M \) by the procedure above presents some challenges in practice. \( T_1 \) and \( T_2 \) are ensemble averages. We can estimate their value online using time averages, since hitting times are ergodic \([5]\), yet the following complication arises: when a random walk \( i \) meets another random walk \( j \), \( i \) and \( j \) become coupled \([13]\); in other words, the next intermeeting time of \( i \) and \( j \) is not anymore exponentially distributed with average \( ED_{dt} \), and \( i \) is more probable to meet \( j \) again much sooner than any other node. In order to overcome this problem, each node needs to keep a record of all nodes with which it is coupled. Every time a new node is encountered, it is stamped as “coupled” for an amount of time equal to the mixing or relaxation time for that graph, which is the expected time until a walk starting from a given position arrives to its stationary distribution. Mixing times for random walks on various graphs can be found in \([5]\).

When node \( i \) measures the next sample intermeeting time, it ignores all nodes that it’s coupled with at the moment. Let’s assume now that when the \( n \)th sample intermeeting time \( T_{1,n} \) is measured by node \( i \) there are \( c_n \) other nodes coupled with \( i \). Then sample \( T_{1,n} \) needs to be scaled by \( \frac{n-1}{n} \) since \( T_{1,n} \) is essentially the intermeeting time between \( i \) and any of the uncoupled \( M - c_n \) nodes. We can similarly calculate a time average of \( T_2 \) by scaling the first \( (ED_{dt} \frac{1}{M}) \) and second \( (ED_{dt} \frac{1}{M-2}) \) meeting times separately. Putting this alltogether, if \( n \) sample meeting times have been measured so far, we can calculate time averages \( \hat{T}_1 \) and \( \hat{T}_2 \) of \( T_1 \) and \( T_2 \) as follows\(^2\):

\[
\hat{T}_1 = \frac{1}{n} \sum_{k=1}^{n} \left( \frac{M - c_k}{M - 1} \right) T_{1,k}
\]

\[
\hat{T}_2 = \frac{1}{n} \sum_{k=1}^{n} \left[ \left( \frac{M - c_{k-1}}{M - 1} \right) T_{1,k-1} + \left( \frac{M - c_k}{M - 2} \right) T_{1,k} \right]
\]

Replacing \( \hat{T}_1 \) and \( \hat{T}_2 \) in Eq.(7) we get a current estimate of \( M \). As can be seen by Eq.(7), the estimator for \( M \) is sensitive to small deviations of \( T_1 \) and \( T_2 \) from their actual values. Therefore it is useful for a node to also maintain a running average of \( M \). Specifically, the running estimate \( \hat{M} \) is updated with every new estimate \( \hat{M}_{new} \) as \( \hat{M} = a\hat{M} + (1 - a)\hat{M}_{new} \) \((0 < a < 1)\). Figure 1 shows how the online estimate \( M \), calculated with our proposed method, quickly converges to its actual value for a \( 200 \times 200 \) torus with 200 nodes. We have also tested our estimator in different scenarios and have observed similar convergence.

In contrast to the ID-counting method described earlier, our estimate mainly depends on the number of intermeeting time samples gathered, which do not need to come from distinct (far away) nodes. As a result of this, even in the small scenario above, our method is more than two times faster than ID-counting. Additionally, we should note that both methods could take advantage of indirect information learning, where a node exchanges known unique IDs or independently collected time samples with every node it encounters, in order to speed up convergence. In this case, our meeting-time based method is going to be even faster than ID-counting in large networks.

As a final note, the procedure above assumes that nodes perform random walks. It is interesting to investigate how accurate Eq.(7) might be under different mobility models. While the principle that a larger \( T_1 \) implies a smaller \( M \) is universal, the exact relationship between them clearly depends on the mobility model. In that case, a combination of ID-counting and heuristic estimates could prove useful. Further investigation of the problem of estimating network parameters is beyond the scope of this work.

### 5.4 Scalability of Spray and Wait

Having shown how to find the minimum number of copies \( L_{min} \) to achieve a delay at most \( a \) times the optimal, it would be interesting, from a scalability point of view, to see how the percentage \( L_{min}/M \) of nodes that need to receive a copy behaves as a function of \( a \) and \( M \). The reason for this is the following: If we assume a large enough TTL value is used and no retransmissions occur, flooding-based schemes will eventually give a copy to every node and therefore perform at least \( M \)

\[ \text{Figure 1: Online estimator of number of nodes (} M \text{) — 200} \times 200 \text{ grid, transmission range = 0, } a = 0.98, \text{ mixing time = 4000.} \]
transmissions. Increased contention and the resulting retransmissions will obviously increase this value significantly, as we shall see. Even utility-based schemes with reasonable utility thresholds will perform $\Theta(M)$ transmissions. On the other hand, Spray and Wait performs $L$ transmissions, and produces very little contention compared to flooding-based schemes. Consequently, the number of transmissions that Spray and Wait performs per message is at most a fraction $L_{\text{min}}/M$ of the number of transmissions per message epidemic and other flooding-based schemes perform.

In Figure 2 we depict the behavior of $L_{\text{min}}/M$ as a function of $M$ for different values of $\alpha$. It is important to note here that, as the number of nodes in the network increases, the percentage of nodes that need to become relays in Spray and Wait to achieve the same relative performance is actually decreasing. The intuition behind this interesting result is the following: when $L \ll M$ the delay of Spray and Wait is dominated by the delay of the wait phase. If $L/M$ is kept constant, it can be seen by Eq.(4) that the delay of Spray and Wait decreases as $1/M$. On the other hand, the delay of the optimal scheme decreases only as $\log(M)/M$, that is, more slowly, as can be seen by Eq.(3). The following Lemma gives a formal proof.

**Lemma 5.2.** Let $L/M$ be constant and let $L \ll M$. Let further $L_{\text{min}}(M)$ denote the minimum number of copies needed by Spray and Wait to achieve an expected delay that is at most $aED_{\text{opt}}$, for some $a$. Then $L_{\text{min}}(M)/M$ is a decreasing function of $M$.

**Proof.** When $L \ll M$ we can use the upper bound of Eq.(5) to examine the behavior of Spray and Wait:

$$ED_{\text{sw}} \leq ED_{\text{dt}}(H_{M-1} - H_{M-L}) + \left(\frac{M - L}{M - 1}\right) ED_{\text{dt}}. L.$$

Since $H_n = \Theta(\log(n)), H_{M-1} - H_{M-L} = \Theta(\log(M-1)/L)$. Also, let $L = cM$, where $c$ is a constant ($c \ll 1$). Replacing $L$ in the previous equation gives us

$$\Theta(\log(1 - 1/M) - \log(1 - c)) ED_{\text{dt}} + \left(\frac{M}{M - 1}\right) \left(\frac{1 - c}{c}\right) \frac{ED_{\text{dt}}}{M}.$$

Now, for large $M$, $\frac{M-1}{M} \simeq 1$. Therefore, keeping the size of the grid $N$ and transmission range $K$ constant we get that $ED_{\text{sw}} = \Theta(1) + \Theta(1 / M) = \Theta(1)$. On the other hand, for constant $N$ and $K$, $ED_{\text{opt}} = \Theta(\log(M)/M)$ as can be easily seen from Eq.(3). Hence, $ED_{\text{sw}}/ED_{\text{opt}} = \Theta\left(\frac{1}{\log(M)}\right)$ (i.e. decreasing with $M$), if $L/M$ is kept constant. This implies that if we require $ED_{\text{sw}}/ED_{\text{opt}}$ to be kept constant for increasing $M$, then $L/M$ has to be decreasing.

This behavior of $L_{\text{min}}/M$ (combined with the independence of $L_{\text{min}}$ from $N$ and $K$) implies that Spray and Wait is extremely scalable. While most of the other multi-copy schemes perform a rapidly increasing number of transmissions as the node density increases, Spray and Wait actually decreases the total transmissions per node as the number of nodes $M$ increases. Thus, its performance advantage over these schemes becomes even more pronounced in large networks.

### 6. SIMULATION RESULTS

#### 6.1 Simulation Environment and Routing Protocols

We have used a custom discrete event-driven simulator to evaluate and compare the performance of different routing protocols under a variety of mobility models and under contention. A slotted collision detection MAC protocol has been implemented in order to arbitrate between nodes contending for the shared channel. The routing protocols we have implemented and simulated are the following: (We will use the shorter names in the parentheses to refer to each routing scheme in simulation plots.)

1. Direct transmission (“direct”),
2. Epidemic routing (“epidemic”),
3. Randomized flooding (“random-flood”),
4. Utility-based routing (“util-flood”),
5. Constraint utility-based routing (“c-utility”),
6. Source Spray and Wait routing (“SW(src)”),
7. Optimal (binary) Spray and Wait routing (“SW”),
8. Seek and Focus single-copy routing (“seek-focus”),

In our simulations we have used as a utility function the one described in [23]. This utility function is maintained as follows: whenever node $X$ encounters node $Y$, $U_X^{(\text{new})}(Y) = U_X(Y) + (1 - U_X(Y))0.75$; else, at every time unit $U_X^{(\text{new})}(Y) = 0.99U_X(Y)$. 

![Figure 2: Required percentage of nodes $L_{\text{min}}/M$ receiving a copy for spray and wait to achieve an expected delay of $aED_{\text{opt}}$](image-url)
6.2 Random Walk - Simulation vs. Analysis

In this section we compare our analytical results for Spray and Wait against simulations. We assume that all nodes perform independent random walks on a 2D torus, and that only a single message needs to be routed from a uniformly chosen source to a uniformly chosen destination. In other words, there is almost no contention for network resources. Recall that in this case the expected delay of Epidemic routing is very close to the optimal delay.

In Figure 3, we compare the expected delay of Source Spray and Wait and Optimal Spray and Wait to that of Direct transmission and Oracle-based Optimal (using both simulation and analytical results). We fix \( N \) to 2500, \( M \) to 30, and \( K \) to 0 (i.e. only nodes in the same position can communicate), and depict delay as a function of the number of copies spread \( L \). As one can see from this figure, simulation and analytical plots for the Spray and Wait schemes present a very close match. Additionally, it is evident that the upper bound from Eq.(5) is tight for low \( L/M \) ratios. (Note that simulation and theoretical results for the delay of Direct transmission and Oracle-based match as well, as was also shown in [26].)

Now, in Epidemic routing all nodes receive a message. Hence, at least \( M \) transmissions are performed. On the other hand, Spray and Wait routing performs only \( L \) transmissions to deliver a message. Therefore, one can see from Figure 3 that Spray and Wait is able to reduce the number of transmissions of Epidemic routing by at least \( 80 \%-90\% \) (i.e. using \( L/M = 0.1 \- 0.2 \)), while still achieving a delay only \( 1.5 \- 2 \) times that of the optimal scheme.

Finally, in Figure 4 we compare the performance of both versions of Spray and Wait and Oracle-based Optimal, in order to better see the effect of the spraying method in the overall performance of Spray and Wait. We choose a larger scenario with \( N = 10000 \) and \( M = 100 \). As can be seen by this figure the use of an efficient spraying method becomes increasingly important when larger number of copies are used. The small discrepancy between simulation and analysis for Optimal Spray and Wait is because the analysis corresponds to the lower bound derived in Theorem 4.4. Note that, in this scenario, the expected delay of Direct transmission is equal to 15652.

6.3 Random Waypoint - Simulation Results

In this section we assume that nodes move independently according to the Random Waypoint model [8]. We first evaluate the effect of traffic load on the performance of all routing schemes (Scenario A). We then examine the scalability of various protocols as the number of nodes increases (Scenario B).

In all scenarios considered, each message is assigned a TTL value of 5000 time units. The utility threshold \( U_{th} \) (0.005 - 0.07) and forwarding probability \( p \) (0.02-0.1) were tuned in each scenario separately, in order to achieve a good tradeoff for the protocol in hand. Additionally, in the Constrained utility-based scheme each node may forward at most 4 copies of the same message. Finally, we usually depict two plots for Spray and Wait for two different \( L \) values, in order to gain better insight into the transmissions-delay tradeoffs involved. We choose these values according to the theory of Section 5. (Specifically, such that its delay would be about \( 2 \times \) that of the Oracle-based Optimal if the nodes were performing random walks.) Note that, in this section, Spray and Wait refers to Optimal Spray and Wait, and that \( SW(L) \) in the simulation plots denotes that \( L \) message copies are used.

Scenario A: 100 nodes move according to the random waypoint model in a 500 x 500 grid with reflective barri-
ers. The transmission range $K$ of each node is equal to 10. Finally, each node is generating a new message for a randomly selected destination with an inter-arrival time distribution uniform in $[1, T_{\text{max}}]$ until time 10000. We vary $T_{\text{max}}$ from 10000 to 2000 creating average traffic loads (total number of messages generated throughout the simulation) from 200 (low traffic) to 1000 (high traffic).

Figure 5 depicts the performance of all routing algorithms, in terms of total number of transmissions and average delivery delay. Epidemic routing performed significantly more transmissions than other schemes (from 560000 to 1440000), and at least an order of magnitude more than Spray and Wait. Therefore, we do not include it in the transmission plots, in order to better compare the remaining schemes. As is evident by these plots Spray and Wait outperforms all single and multi-copy protocols discussed and achieves its design goals set in Section 3. Specifically: (i) under low traffic its delay is similar to Epidemic routing and is $1.4 - 2.2$ times faster than all other multi-copy protocols; it performs an order of magnitude less transmissions than Epidemic routing, and $3 - 4$ times less transmissions than Constrained-utility, which is the second best in terms of transmissions among multi-copy schemes, and (ii) under high traffic it retains the same advantage in terms of total transmissions, and outperforms all other protocols, in terms of delay, by a factor of $1.8 - 3.3$.

As a final note, the delivery ratio of almost all schemes in this scenario was above 90\% for all traffic loads, except that of Seek and Focus which was about 70\%, and that of Epidemic routing which plummeted to less than 50\% for very high traffic, due to severe contention.

**Scenario B:** In this scenario, we fix $T_{\text{max}}$ to 4000 (medium to high traffic load) and vary the number of nodes $M$ from 100 to 300. The size of the network is again $500 \times 500$ and transmission range $K = 10$. $M = 100$ corresponds to a relatively sparse network (about 12\% of the grid space is within range of any of the 100 nodes), while a higher $M$ obviously implies a denser and more connected network.

Figure 6 depicts the number of transmissions and the average delay of different routing schemes for Scenario B. Spray and Wait clearly outperforms all protocols, in this scenario too, in terms of both transmissions and delay. Most importantly, it is extremely scalable, compared to other multi-copy options. Specifically, for $M = 100$ Spray and Wait performs $3 - 6.9$ times fewer transmissions and is faster by a factor of $1.6 - 3.6$, compared to different schemes. (The number of transmissions in epidemic routing were extremely high, ranging from 120000 to more than 400000.) Further, when $M$ grows to 300 it outperforms all other schemes by an order of magnitude in terms of transmissions and its delay is still $1.8 - 4.6$ times smaller. This is because, as $M$ grows, flooding-based and utility-based schemes are more heavily plagued by contention and retransmissions. On the other hand, Spray and Wait performs only a few extra transmissions corresponding to the few extra copies used, so as to sustain the same relative performance compared to the optimal. As a final note, we also run a number of simulations for other values of $N$ and $K$, and similar performance trends were observed.

**7. CONCLUSION**

In this work we investigate the problem of multi-copy routing in intermittently connected mobile networks. We propose a simple multi-copy scheme, called Spray and Wait, that manages to overcome the shortcomings of epidemic routing and other flooding-based schemes, and avoids the performance dilemma inherent in utility-based schemes. Using theory and simulations we show that Spray and Wait, despite its simplicity, outperforms all existing schemes with respect to number of transmissions and delivery delays, achieves comparable delays to an optimal scheme, and is very scalable as the size of the network increases.

In future work we intend to look in detail into schemes that “spray” a number of copies quickly, and then use utility-based or other efficient single-copy schemes to route each copy independently. Finally, we plan to extend our analysis to cover contention for the wireless channel, and more realistic mobility models, that exhibit correlation in space and time [6, 19].
8. REFERENCES


