# Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks

Thrasyvoulos Spyropoulos
Department of Electrical Engineering, USC
spyropou@usc.edu

Konstantinos Psounis
Department of Electrical Engineering, USC
psounis@usc.edu

Cauligi S. Raghavendra
Department of Electrical Engineering, USC
raghu@usc.edu

## ABSTRACT

Intermittently connected mobile networks are sparse wireless networks where most of the time there does not exist a complete path from the source to the destination. These networks fall into the general category of Delay Tolerant Networks. There are many real networks that follow 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. Furthermore, proposed efforts to significantly reduce the overhead of flooding-based schemes have often be plagued by large delays. 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.

Using theory and simulations we show that Spray and Wait outperforms all existing schemes with respect to *both* average message delivery delay and number of transmissions per message delivered; its overall performance is close to the optimal scheme. Furthermore, it is highly scalable retaining good performance under a large range of scenarios, unlike other schemes. Finally, it is simple to implement and optimize to achieve given performance goals in practice.

## 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, 17], military networks [2], inter-planetary networks [7], nomadic communities networks [9] 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 [16], AODV [21], 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 [14]. However, there mobility is exploited in order to improve capacity, while here it is used to overcome the lack of end-to-end connectivity.

Despite a large number of existing proposals, there is no 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 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, (iii) it is highly scalable, exhibiting good and predictable performance for a large range of network sizes, node densities and connectivity levels; what is more, 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, and (iv) it can be easily tuned online to achieve given QoS requirements, even in unknown networks. 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 and summarize our contribution. Section 3 presents our proposed solution, Spray and Wait. Then, in Section 4 we show extensively how to optimize Spray and Wait in practical situations, and also examine its scalability. Simulation results are presented in Section 5, where the performance of all the strategies are compared with respect to message delivery delay and number of transmissions per message delivered. Finally, Section 6 concludes the paper.

## 2. RELATED WORK AND CONTRIBUTIONS

Although a significant amount of work and consensus exists on the general DTN architecture [1], there hasn't been a similar focus and agreement on DTN routing algorithms, especially when it comes to networks with "opportunistic" connectivity. This might be due to the large variety of applications and networks characteristics falling under the DTN umbrella.

A large number of routing protocols for wireless ad-hoc networks have been proposed in the past [6, 20], usually falling into the general categories of "proactive" or "reactive" protocols. However, both these types assume the existence of a complete path between the source and the destination. Lack of such a path, as in the case of intermittently connected networks, would prevent them from delivering any useful data. Reactive protocols would initiate a route request, that most times never reaches the destination, or even if it does, the path might brake again, before the route reply reaches back. Further, their performance would still be poor even if the network was only slightly disconnected. To see this, note that the expected throughput is approximately connected with the average path duration $PD$ and the time to repair a broken path $t_{repair}$ with the following relationship: $throughput = rate(1 - \frac{t_{repair}}{PD})$ [22]. Therefore, it is evident that even moderate node mobility would make path durations be much smaller than $t_{repair}$ if the network has some disconnections ($t_{repair}$ is at least 2 times the delay of the optimal algorithm). Proactive protocols, on the other hand, would simply declare lack of a path to the destination under intermittent connectivity, or result into a deluge of topology updates that would dominate the available bandwidth under high mobility. It is therefore clear that traditional ad-hoc routing protocols are not appropriate for the types of networks we're interested in here.

Another approach to deal with disconnections or "disruptions" [2] is to reinforce connectivity on demand, by bringing for example additional communication resources into the network when necessary (e.g. satellites, UAVs, etc.). Similarly, one could force a number of specialized nodes (e.g. robots) to follow a given trajectory between disconnected parts of the network in order to bridge the gap [28, 18]. Nevertheless, such approaches are orthogonal to our work; our aim is to study what can be done in the absence of such enforced mobility and connectivity.

A study of routing for DTN networks with *predictable* connectivity was performed in [15]. There, a number of algorithms with increasingly more knowledge about network characteristics like upcoming "contacts", queue sizes, etc. is compared with an optimal centralized solution of the problem, formulated as a linear programm. Although it is shown that even limited knowledge might be adequate to efficiently solve this problem, the algorithms proposed apply to the types of DTNs were connectivity is intermittent, but can be predicted (for example, due to planetary and satellite movement in IPN [7]). In our case instead, connectivity is rather *opportunistic* and subject to the statistics of the mobility model followed by nodes.

A number of routing proposals exist that are specifically targeted towards this new context of intermittently connected mobile networks. Many of them try to deal with application-specific problems, especially in the field of sensor networks. In [23], a number of mobile nodes performing independent random walks serve as *DataMules* that collect data from static sensors and deliver them to base stations. Each DatMule performs "Direct Transmission", that is, will not forward data to other DataMules, but only deliver it to its destination. 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 [5, 13].

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, 8, 11, 14, 17, 18, 19, 27]. *Epidemic routing* [27] 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). Although epidemic routing finds the same path as the optimal scheme under no contention [25], it is very wasteful of network resources. Furthermore, it creates a lot of contention for the limited buffer space and network capacity of typical wireless networks, resulting in many message drops and retransmissions. This can have a detrimental effect on performance, as has been noted earlier in [19, 26], and will also be shown in our simulation results.

One simple approach to reduce the overhead of flooding and improve its performance is to only forward a copy with some probability $p < 1$ [26]. A different, more sophisticated approach is that of *History-based* or *Utility-based Routing* [8, 17, 19]. Here, each node maintains a utility value for every other node in the network, based on a timer indicating the time elapsed since the two nodes last encountered each other. These utility values essentially carry indirect information about relative node locations, which get diffused through nodes' mobility. Therefore, a scheme can be designed, where nodes forward message copies only to nodes with a higher utility by at least some pre-specified threshold value $U_{th}$ for the message's destination. Such a scheme results in superior performance than flooding [17, 19], and makes better forwarding decisions than randomized routing [25]. This method has also been found to be quite efficient in the context of regular, connected, wireless networks [11]. Nevertheless, utility-based schemes, are still flooding-based in nature. What is worse, they are faced with an important dilemma when choosing the utility threshold. Too small a threshold and the scheme behaves like pure flooding. Too high a threshold and the delay increases significantly, as we shall see.

Single-copy schemes have also been extensively explored in [23, 25]. Such schemes generate and route only one copy per message (in contrast to flooding schemes that essentially send a copy to every node), in order to significantly reduce the number of transmissions. Although they might be useful in some situations, single-copy schemes do not present desirable solutions for applications that require high probabilities of delivery and low delays.

Finally, an optimal "oracle-based" algorithm has been described in [25]. This algorithm is aware of all future movement, 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. This algorithm is of course not implementable, but is quite useful to compare against proposed practical schemes.

Our scheme, Spray and Wait, manages to significantly reduce the transmission overhead 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. 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. These methods are complemented by an algorithm 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. Finally, we demonstrate that Spray and Wait is remarkably robust and scalable, exhibiting good *and* predictable performance under a very large range of connectivity levels, network sizes, and traffic loads, unlike other alternatives.

## 3. 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:

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 better way is the following.

Definition 3.2 (Binary Spray and Wait.). *The source of a message initially starts with $L$ copies; any node $A$ that has $n > 1$ message copies (source or relay), and encounters another node $B$ (with no copies), hands over to $B$ $\lfloor n/2 \rfloor$ and keeps $\lceil n/2 \rceil$ for itself; when it is left with only one copy, it switches to direct transmission.*

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

Theorem 3.1. *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 call a node "active" when it has more than one copies of a message. Let us further 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: assign the root a value of $L$; if the current node has a value $n > 1$ create a right child with a value of $1 - f(n)$ and a left one with a value of $f(n)$; continue until all leaf nodes have a value of 1.

A particular spraying corresponds then to a sequence of visiting all nodes of the tree. This sequence is random. Nevertheless, *on the average*, all tree nodes at the same level are visited in parallel. Further, as we saw earlier, the higher the number of active nodes, when $i$ copies are spread, the smaller the residual expected delay, let $ED(i)$. Since the total number of tree nodes is fixed $(2^{1+\log L} - 1)$ for any spraying function $f$, 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. □

As $L$ grows larger, the sophistication of the spraying heuristic has an increasing impact on the delivery delay of the spray and wait scheme, as can be seen in Figure 1. This figure also shows the delay of the Optimal scheme introduced in [25].

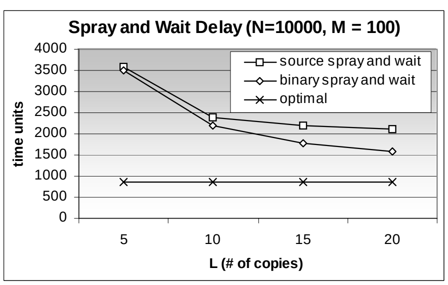


**Figure 1: Comparison of Source Spray and Wait, Binary Spray and Wait, and Optimal schemes ($N = 10000, M = 100$).**

# 4. OPTIMIZING SPRAY AND WAIT TO MEET PERFORMANCE CONSTRAINTS

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.

Before we do so though, we summarize in the following lemma a few of our own results from [25] regarding the expected delay of the Direct Transmission and Optimal schemes:

LEMMA 4.1. *Let $M$ nodes having a transmission range of $K$ perform independent random walks on a $\sqrt{N} \times \sqrt{N}$ torus. Then:*

1. *The delay of Direct Transmission is exponentially distributed with average*

$$ED_{dt} = 0.5N \left(0.34 \log N - \frac{2^{K+1} - K - 2}{2^K - 1}\right).$$

2. *The expected delay of the Optimal algorithm is*

$$ED_{opt} = \frac{H_{M-1}}{(M-1)} ED_{dt},$$

*where $H_n$ is the $n^{th}$ Harmonic Number, i.e, $H_n = \sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$.*

We have also computed a tight upper bound for the expected delay of Spray and Wait. Due to limitations of space we omit the proof and only state the result. The interested reader can find the proof in [24].

LEMMA 4.2. *The expected delay of Spray and Wait, when $L$ message copies are used, is upper-bounded by*

$$ED_{sw} \leq (H_{M-1} - H_{M-L})\, ED_{dt} + \frac{M-L}{M-1} \frac{ED_{dt}}{L}. \qquad (1)$$

*This bound is tight when $L \ll M$.*

## 4.1 Choosing L to Achieve a Required Expected Delay

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. Note that the issue of energy dissipation is also directly tied to the number of copies $L$ used by Spray and Wait, since Spray and Wait performs exactly $L$ transmissions. Let us assume 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 this is the best that any routing protocol can do.

LEMMA 4.3. *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 $M$.*

The above lemma states that the required number of copies only depends on the number of nodes, and is straightforward to prove from Eq.(1). Furthermore, since the upper bound of Eq.(1) is tight for small $L/M$ values, if the delay constraint $a$ is not too stringent, we can use one of the following methods to quickly get a good estimate for $L_{min}$: (i) solve the upper bound equation Eq.(1) for $L$, by letting $ED_{sw} = aED_{opt}$, and taking $\lceil L \rceil$, or (ii)

Table 1: minimum $L$ to achieve expected delay

| a | 1.5 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| exact | 21 | 13 | 8 | 6 | 5 | 4 | 3 | 3 | 3 | 2 |
| bound | N.A. | N.A. | 11 | 7 | 6 | 5 | 4 | 3 | 3 | 2 |
| taylor | N.A. | N.A. | 10 | 7 | 5 | 4 | 3 | 3 | 3 | 2 |

approximate the harmonic number $H_{M-L}$ in Eq.(1) with its Taylor Series terms up to second order, and solve the resulting third degree polynomial:

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

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

One could also iteratively calculate the exact number of copies needed using the system of recursive equations from [24]. However this method is quite more cumbersome. In Table 1 we compare exact results for $L_{min}$ to the ones calculated with the two approximate 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.

## 4.2 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. 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 intermeeting time statistics. Specifically, let us define $T_1$ as the time until a node (starting from the stationary distribution) encounters *any* other node. It is easy to see from Lemma 4.2 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 $ED_{dt}\left(\frac{1}{M-1} + \frac{1}{M-2}\right)$. Cancelling $ED_{dt}$ from these two equations we get the following estimate for $M$:

$$\hat{M} = \frac{2T_2 - 3T_1}{T_2 - 2T_1}. \tag{2}$$

Estimating $M$ by the procedure above presents some challenges in practice, because $T_1$ and $T_2$ are ensemble averages. Since hitting times are ergodic [4], a node can collect sample intermeeting times $T_{1,k}$ and $T_{2,k}$ and calculate time averages $\hat{T}_1$ and $\hat{T}_2$ instead. However, the following complication arises: when a random walk $i$ meets another random walk $j$, $i$ and $j$ become *coupled* [12]; in other words, the next intermeeting time of $i$ and $j$ is not anymore exponentially distributed with average $ED_{dt}$. In order to overcome this problem, each node keeps 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 [4]. Then, when node $i$ measures the next sample intermeeting time, it ignores all nodes that it's coupled with at the moment, denoted as $c_k$, and scales the collected sample $T_{1,k}$ by $\frac{M-c_k}{M-1}$. A similar procedure is followed for $\hat{T}_2$. Putting it alltogether, after $n$ samples have been collected:

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

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

Replacing $\hat{T}_1$ and $\hat{T}_2$ in Eq.(2) we get a current estimate of $M$. As can be seen by Eq.(2), 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 $\hat{M} = a\hat{M} + (1-a)\hat{M}_{new}$ $(0 < a < 1)$.

Although the exposition of our estimation method has been based on the random walk mobility model, it is important to note that *this method holds for any mobility model with approximately exponentially distributed meeting times.* The reason for this is that the expected meeting time, which differs from mobility model to mobility model, gets cancelled from the equations. What is important is that the expected time of meeting any of $i$ nodes is approximately $\frac{1}{i}$ the expected time of meeting one node.

Figure 2 shows how the online estimate $\hat{M}$, calculated with our proposed method, quickly converges to its actual value for a $200 \times 200$ torus with 200 nodes, for both the random walk and random waypoint models. (Note that even in this small scenario, our method's convergence is more than two times faster than ID-counting.) We have tested our estimator in different scenarios and have observed similar convergence, as well. In the future, we intend to examine how our method performs with other mobility models, too. As a general rule though, it is

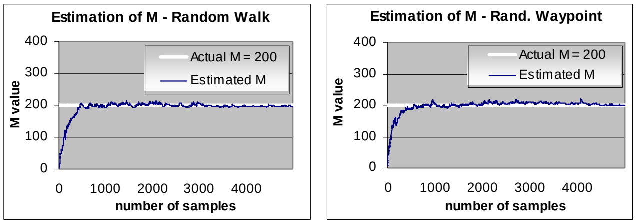


**Figure 2: Online estimator of number of nodes ($M$) — $200 \times 200$ grid, transmission range $= 0$, $a = 0.98$, mixing time $= 4000$.**

shown in [4] that the hitting time distribution of general random walks on graphs always has an exponential tail, and in many cases is approximately exponential itself. Consequently, we expect that if a given mobility model can be (even approximately) represented as a random walk on an appropriate graph, then our method should produce sufficiently accurate estimates.

As a final note, both our method and ID-counting could take advantage of indirect information learning, where nodes exchange known unique IDs or independently collected samples to speed up convergence.

## 4.3 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 $M$ 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 $L_{min}/M$ of the number of transmissions per message epidemic and other flooding-based schemes perform.

In Figure 3 we depict the behavior of $L_{min}/M$ as a function of $M$ for different values of $a$. It is important to note there 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 behinds 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; in that case, if $L/M$ is kept constant, the delay of Spray and Wait decreases roughly as $1/M$. On the other hand, the delay of the optimal scheme decreases more slowly as $\log(M)/M$, as can be seen by Eq.(1). The following Lemma gives a formal proof.

LEMMA 4.4. *Let $L/M$ be constant and let $L \ll M$. Let further $L_{min}(M)$ denote the minimum number of copies needed by Spray and Wait to achieve an expected delay*

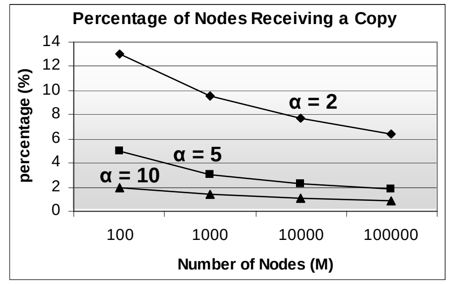


**Figure 3: Required percentage of nodes $L_{min}/M$ receiving a copy for spray and wait to achieve an expected delay of $aED_{opt}$**

*that is at most $aED_{opt}$, for some $a$. Then $\frac{L_{min}(M)}{M}$ is a decreasing function of $M$.*

PROOF. When $L \ll M$ we can use the upper bound of Eq.(1) to examine the behavior of Spray and Wait:

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

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

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

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 $ED_{sw} = \Theta(1) + \Theta(1)\frac{1}{M} = \Theta(\frac{1}{M})$.

On the other hand, for constant $N$ and $K$, $ED_{opt} = \Theta(\frac{\log(M)}{M})$ as can be easily seen from Eq.(1). Hence, $\frac{ED_{sw}}{ED_{opt}} = \Theta\left(\frac{1}{\log(M)}\right)$ (i.e. decreasing with $M$), if $L/M$ is kept constant. This implies that if we require $\frac{ED_{sw}}{ED_{opt}}$ to be kept constant for increasing $M$, then $L/M$ has to be decreasing. □

This behavior of $L_{min}/M$ (combined with the independence of $L_{min}$ 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.

# 5. SIMULATION RESULTS

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 contenting for the shared channel. The routing protocols we have implemented and simulated are the following: (1) Epidemic routing ("epidemic"), (2) Randomized flooding with $p = (0.02-0.1)$ ("random-flood"),

(3) Utility-based routing as described in [19] with a utility threshold $U_{th} = (0.005 - 0.2)$ ("util-flood"), (4) Optimal (binary) Spray and Wait with $L$ copies ("SW(L)"), (5) Seek and Focus single-copy routing ("seek-focus") [25], and (6) Oracle-based Optimal routing ("optimal"). (We will use the shorter names in the parentheses to refer to each routing scheme in simulation plots.)

In all scenarios considered, each message is assigned a TTL value between $4000 - 6000$ time units. We have tried to tune the parameters of each protocol in each scenario separately, in order to achieve a good tradeoff for the protocol in hand. Finally, we 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 4. (Specifically, such that its delay would be about $2\times$ that of the Oracle-based Optimal if the nodes were performing random walks.)

We first evaluate the effect of traffic load on the performance of all routing schemes (Scenario A). We then examine their performance as the level of connectivity changes (Scenario B).

## 5.1 Scenario A: Effect of Traffic Load

100 nodes move according to the random waypoint model [6] in a $500 \times 500$ grid with reflective barriers. 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 $[1, T_{max}]$ until time 10000. We vary $T_{max}$ 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 4 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 56000 to 144000), 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 $5 - 6$ times less transmissions than Randomized and Utility-based, 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.

## 5.2 Scenario B: Effect of Connectivity

In this scenario, the size of the network is $200 \times 200$ and we fix $T_{max}$ to 4000 (medium traffic load). We vary the number of nodes $M$ and the transmission range $K$ in order to evaluate the performance of all protocols in networks with a large range of connectivity characteristics, ranging from very sparse, highly disconnected networks, to *almost* connected networks.

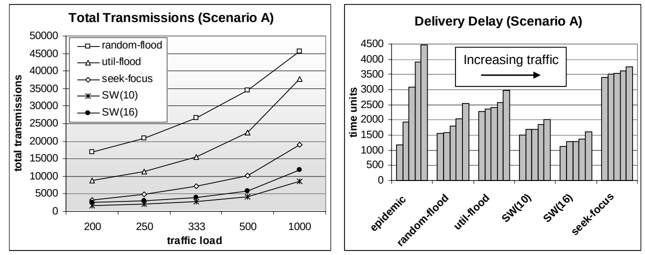


**Figure 4: Scenario A - performance comparison of all routing protocols under varying traffic loads.**

Before we proceed, it is necessary to define a meaningful connectivity metric. Although a number of different metrics have been proposed (for example [10]), no widespread agreement exists, especially if one needs to capture both disconnected and connected networks. We believe that a meaningful metric for the networks of interest is the expected *maximum cluster size* defined as the percentage of total nodes in the largest connected component (cluster). This indicates what percentage of nodes have already conglomerated into the connected part of the network, with "one" implying a regular connected network (with high probability). Figure 5 depicts the connectivity metric for the $200 \times 200$ torus, as a function of transmission range $K$ for two different values of $M$.

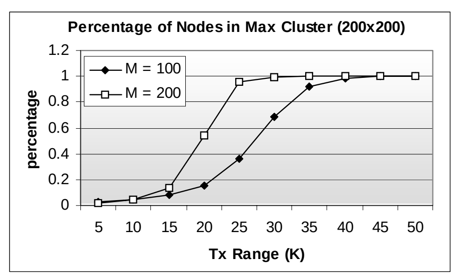


**Figure 5: Expected percentage of total nodes in largest connected component, as a function of $M$ and $K$ ($200 \times 200$ grid).**

We have picked a number of points from Figure 5 that span the entire connectivity range, and have evaluated the performance of all protocols under these scenarios. Figure 6 and Figure 7 depict the number of transmissions and the average delay, respectively. As can be seen there, Spray and Wait clearly outperforms all protocols, in terms of both transmissions and delay, for all levels of connectivity. Most importantly, it is extremely scalable and robust, compared to other multi-copy or even single-copy options. Epidemic routing and the rest of the schemes manage to achieve a delay that is comparable to Spray and Wait for very few connectivity values only, but perform quite poorly for the vast majority of scenarios. Furthermore, their performance seems to vary significantly for different connectivity levels (despite our effort to tune each protocol for the given scenario, whenever possible). Spray and Wait, on the other hand, exhibits

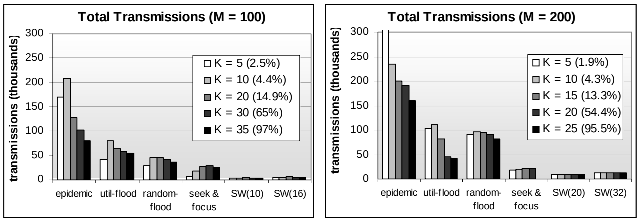


**Figure 6: Scenario B - Total transmission as a function of number of nodes $M$ and transmission range.**

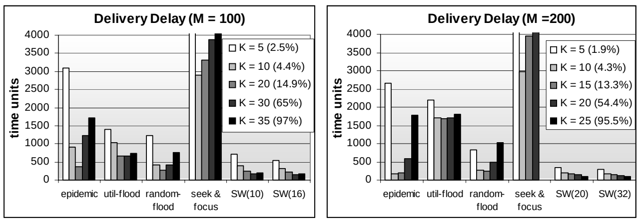


**Figure 7: Scenario B - Delivery delay as a function of number of nodes $M$ and transmission range.**

great stability. It performs a fixed number of transmissions across all scenarios, while achieving a delivery delay that decreases as the level of connectivity increases, as one would expect.

# 6. CONCLUSION

In this work we investigated the problem of efficient routing in intermittently connected mobile networks. We proposed a simple 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 or connectivity level increase.

# 7. REFERENCES


[1] Delay tolerant networking research group. *http://www.dtnrg.org*.

[2] Disruption tolerant networking. *http://www.darpa.mil/ato/solicit/DTN/*.

[3] Sensor networking with delay tolerance (sendt). *http://down.dsg.cs.tcd.ie/sendt/*.

[4] D. Aldous and J. Fill. Reversible markov chains and random walks on graphs. (monograph in preparation.). *http://stat-www.berkeley.edu/users/aldous/RWG/book.html*.

[5] A. Beaufour, M. Leopold, and P. Bonnet. Smart-tag based data dissemination. In *Proc. ACM Inter. Workshop on Wireless Sensor Nets and Applications(WSNA)*, 2002.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[7] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41:128–136, 2003.

[8] X. Chen and A. L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Proc. of Workshop on Principles of Mobile Computing, colocated with PODC'01*, Aug. 2001.

[9] A. Doria, M. Udn, and D. P. Pandey. Providing connectivity to the saami nomadic community. In *Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

[10] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proc. of Infocom*, 2002.

[11] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *Proc. ACM MobiHoc'03*, 2003.

[12] R. Durrett. *Probability: Theory and Examples*. Duxbury Press, second edition, 1995.

[13] N. Glance, D. Snowdon, and J.-L. Meunier. Pollen: using people as a communication medium. *Computer Networks*, 35(4):429–442, Mar. 2001.

[14] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, 2002.

[15] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. ACM/ Sigcomm*, Aug. 2004.

[16] D. B. Johnson, D. A. Maltz, and J. Broch. *DSR: the dynamic source routing protocol for multihop wireless ad hoc networks*, chapter 5. Addison-Wesley Longman Pub. Co., Inc., 2001.

[17] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *Proc. ASPLOS'02*, 2002.

[18] Q. Li and D. Rus. Communication in disconnected ad hoc networks using message relay. *J. Parallel Distrib. Comput.*, 63(1):75–86, 2003.

[19] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[20] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad-hoc networks. *IEEE Network*, 10(6):30–39, 2001.

[21] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad-hoc on-demand distance vector routing. IETF MANET DRAFT, 2002.

[22] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In *Proc. of MobiHoc*, 2003.

[23] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Elsevier Ad Hoc Networks Journal*, 1:215–233, Sept. 2003.

[24] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Multiple-copy routing in intermittently connected mobile networks. Technical Report CENG-2004-12, USC, 2004.

[25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Proc. of IEEE Secon'04*, 2004.

[26] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.*, 8(2/3):153–167, 2002.

[27] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, Apr. 2000.

[28] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. MobiHoc'04*, 2004.