CS–2000–06

Epidemic Routing for Partially-Connected Ad Hoc Networks

Amin Vahdat and David Becker
Department of Computer Science
Duke University
Durham, NC 27708
(vahdat,becker)@cs.duke.edu

Department of Computer Science

Duke University

Durham, North Carolina 27708–0129

April 2000

Epidemic Routing for Partially-Connected Ad Hoc Networks

Amin Vahdat and David Becker
Department of Computer Science
Duke University
Durham, NC 27708
(vahdat,becker)@cs.duke.edu

April 2000

Abstract

Mobile ad hoc routing protocols allow nodes with wireless adaptors to communicate with one another without any pre-existing network infrastructure. Existing ad hoc routing protocols, while robust to rapidly changing network topology, assume the presence of a connected path from source to destination. Given power limitations, the advent of short-range wireless networks, and the wide physical conditions over which ad hoc networks must be deployed, in some scenarios it is likely that this assumption is invalid. In this work, we develop techniques to deliver messages in the case where there is *never* a connected path from source to destination or when a network partition exists at the time a message is originated. To this end, we introduce *Epidemic Routing*, where random pair-wise exchanges of messages among mobile hosts ensure eventual message delivery. The goals of Epidemic Routing are to: i) maximize message delivery rate, ii) minimize message latency, and iii) minimize the total resources consumed in message delivery. Through an implementation in the Monarch simulator, we show that Epidemic Routing achieves eventual delivery of 100% of messages with reasonable aggregate resource consumption in a number of interesting scenarios.

1 Introduction

The advent of inexpensive wireless networking solutions has enabled a broad range of exciting new applications. Wireless network adaptors in portable computing devices, such as cellular phones, personal digital assistants, and laptops, can enable ubiquitous access to global information resources. Challenges to achieving this vision include the need to have a wired base station in range of wireless hosts and the energy/expense of transmitting information across large distances. Ad hoc wireless networking addresses some of these challenges by allowing mobile hosts to communicate with one another with no pre-existing communication infrastructure. In ad hoc networks, arbitrary mobile hosts can be recruited to "fill the gap" by serving as intermediate routers between two hosts that may otherwise not be in direct transmission range of one another. Recent work investigates route discovery and maintenance [6, 16, 19, 21, 25, 26, 27], minimizing power consumption [2, 32], and maintaining QoS guarantees [23, 30, 33] in ad hoc networks.

The common assumption behind existing ad hoc routing techniques is that there is always a connected path from source to destination. However, the advent of short-range wireless communication environments (e.g., Bluetooth [15] and BlueSky [3]) and the wide physical range and circumstances over which such networks are deployed means that this assumption is not always valid in realistic scenarios. Unfortunately, with current ad hoc routing protocols, packets are not delivered if a network partition exists between the source and the destination when a message is originated. Certain applications, such as real-time, constant bit rate communication may require a connected path for meaningful communication. However, a number of other application classes benefit from the eventual and timely delivery of messages, especially in the case where frequent and numerous network partitions would prevent messages from ever being delivered end to end. We describe a few of these applications below:

- Mobile Sensor Networks: In this example, sensors with wireless connectivity are deployed over a geographic area [11, 17]. These sensors may be simple, e.g., used to detect motion, chemicals, temperature, or they may be more sophisticated, e.g., designed to record audio and video. Ideally, these sensors periodically transmit their findings to a base station, perhaps for analysis or permanent storage. These sensors may be small and have limited communication range, implying that they are not always able to establish a connected path (leveraging other sensors as routers) back to base stations. Such sensors may be mobile — for example, under their own power[1] or because they are suspended in air/water — implying that individual sensors may periodically come into contact with one another through node mobility.

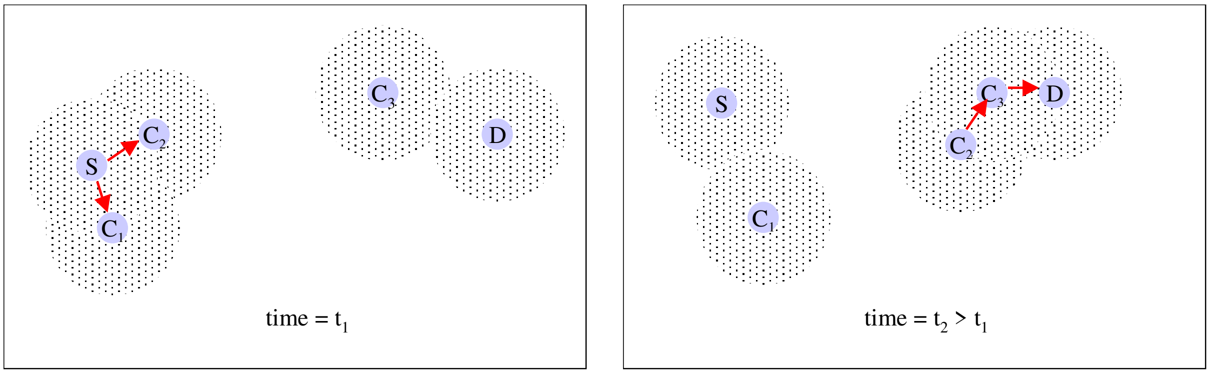
- Smart Dust: Related to the previous example, a recent proposal [20] describes challenges in networks comprised of Micro-electrical Mechanical Sensors (MEMS). Because of the power restrictions associated with their small size, these sensors might utilize optical connections for communication, requiring line of sight between each hop in a connected optical path from source to destination. Frequent physical obstructions may make the presence of such a "connected" line of sight path unlikely in some cases, though eventual pair-wise connectivity among MEMS is more likely if the MEMS are mobile.

- Disaster Recovery/Military Deployment: In this example, people, in addition to sensors, are deployed over an area with limited wireless coverage (i.e., few, if any, base stations). For disaster recovery, field agents wish to communicate their findings regarding, for example, environmental hazards or survivors to other field agents as well as to a command post. Again, battery concerns and the wide physical dispersement of individual agents make it unlikely that full wireless connectivity can be continuously maintained among all mobile hosts.

In the context of such applications, the goal of this work is to develop techniques for delivering application data with high probability even when there is *never* a fully connected path between source and destination. Thus, our work makes minimal assumptions about the connectivity of the underlying ad hoc network: i) the sender is never in range of any base stations, ii) the sender does not know where the receiver is currently located or the best "route" to follow, iii) the receiver may also be a roaming wireless host, and iv) pairs of hosts (not necessarily the sender and receiver) periodically and randomly come into communication range of one another through node mobility.

Our approach, called *Epidemic Routing* [9] is to distribute application messages to hosts, called *carriers*, within connected portions of ad hoc networks. In this way, messages are quickly distributed through connected portions of the network. Epidemic Routing then relies upon carriers coming into contact with another connected portion of the network through node mobility. At this point, the message spreads to an additional island of nodes. Through such transitive transmission of data, messages have a high probability of eventually reaching their destination. Figure 1 depicts Epidemic Routing at a high level, with mobile nodes represented as dark circles and their wireless communication range shown as a dotted circle extending from the source. In Figure 1(a), a source, S, wishes to send a message to a destination, D, but no connected path is available from S to D. S transmits its messages to its two neighbors, C_1 and C_2, within direct communication range. At some later time, as shown in Figure 1(b), C_2 comes into direct communication range with

[1] In one compelling example, sensors are carried by seals in the open ocean to increase the number of available deep ocean temperature readings in a region from 52 to 22000 [29].



(a) (b)

Figure 1: A source, S, wishes to transmit a message to a destination but no connected path is available in part (a). Carriers, C_1-C_3 are leveraged to transitively deliver the message to its destination at some later point in time as shown in (b).

another host, C_3, and transmits the message to it. C_3 is in direct range of D and finally sends the message to its destination.

We leverage a variant of the theory of epidemic algorithms [9] for our protocol. This theory states that given random exchange of data among replicas, all updates will be seen by all replicas in a bounded amount of time (i.e., the system will reach eventual consistency). The goal of Epidemic Routing is more modest: to deliver a message (update) with high probability to a *particular* host. In fact, other than the destination, we wish to minimize the set of other hosts that carries or transmits a particular message to bound aggregate system resources (i.e., memory, network bandwidth, or energy) consumed in message delivery. Of course, while not explored here, a simple extension to Epidemic Routing could support message broadcast/multicast in partially connected ad hoc networks.

The overall goal of Epidemic Routing is to maximize message delivery rate and minimize message delivery latency, while also minimizing the aggregate system resources consumed in message delivery. We accomplish this by placing an upper bound on message hop count and per-node buffer space (the amount of memory devoted to carrying other host's messages). By increasing bounds on these parameters, applications can increase the probability that a message will be successfully delivered in exchange for higher aggregate resource consumption. We evaluate the utility of Epidemic Routing and explore the design space of potential scenarios and system parameters through an implementation of our protocol in the Monarch [7] simulator. Monarch extends the popular ns simulator [24] with host mobility and an implementation of the IEEE 802.11 specification. We explore message delivery rate and resource consumption under a number of different scenarios. Our results show that Epidemic Routing is able to deliver nearly all messages in scenarios where existing ad hoc routing protocols fail to delivery any messages because of limited node connectivity. We also investigate our protocol's sensitivity to available resources. For one representative scenario, Epidemic Routing delivers 100% of messages assuming enough per-node buffering to store between 10-25% of the messages originated in the scenario. Thus, while Epidemic Routing can increase resource consumption, under some circumstances it may be the only viable technique for successfully delivering application data.

The rest of this paper is organized as follows. Section 2 describes our design goals and the Epidemic Routing Protocol. We present the simulation environment in which we implemented Epidemic Routing and the performance results of our protocol in Section 3. Related work and future research directions are described in Section 4. Section 5 summarizes our conclusions and describes future work.

2 System Architecture

2.1 Goals and Design Issues

The goals of Epidemic Routing are to: i) efficiently distribute messages through partially connected ad hoc networks in a probabilistic fashion, ii) minimize the amount of resources consumed in delivering any single message, and iii) maximize the percentage of messages that are eventually delivered to their destination.

Epidemic Routing raises a number of interesting issues for the underlying routing protocol:

- *Routing Under Uncertainty:* Message senders have inexact knowledge of the location of nodes throughout the system. Thus, a key issue is determining whether to transmit a message when a host comes into range of a potential carrier. For example, the system may account for the hosts that the target carrier has recently come into contact with and its current destination/velocity.

- *Resource Allocation:* Unlike standard routing, it is likely and perhaps even desirable to have multiple copies of a message in transit simultaneously. In general, the system must balance the conflicting goals of maximizing message delivery and minimizing resource consumption. For example, a single message should not consume buffer space at all the hosts in the Internet just to ensure its most timely delivery. On the other hand, copies of a message may be buffered at multiple hosts to maximize the likelihood that a particular message is eventually delivered.

- *Performance:* A given message exchange and routing protocol can be evaluated along a number of different axes. Performance metrics include the average latency in delivering messages, the average amount of system storage and communication bandwidth consumed in delivering a message, and the amount of energy consumed in transmitting the message to its destination. This last metric of energy consumption is particularly relevant to mobile hosts because a host must consider the energy consequences of becoming a carrier for a particular message. Since storing and transmitting messages consumes energy as well as traditional performance metrics such as CPU cycles, memory, and network bandwidth, it is important to balance the consumption of all system resources in transmitting messages to their final destination.

- *Reliability:* Given the probabilistic delivery of messages in our model, certain applications may desire acknowledgments of successful message delivery. For example, the originating host and all carriers can free up resources associated with a message upon learning of its successful reception at the intended host.

- *Security:* A message may traverse an arbitrary path of hosts before reaching its ultimate destination. Depending on the sensitivity of the information and the requirements of individual applications, receivers may require certain guarantees about the authenticity of a message. While well-known cryptographic techniques [31] can provide some such guarantees, it may also be beneficial to track the entire path that a message travels in reaching the receiver. In this way, receivers can learn if a message has been exposed (even in encrypted format) to untrusted hosts. Similarly, carriers can use the sensitivity information associated with a particular message to eliminate untrusted hosts from the list of potential carriers.

2.2 Epidemic Routing Protocol

Epidemic Routing supports the eventual delivery of messages to arbitrary destinations with minimal assumptions regarding the underlying topology and connectivity of the underlying network. In fact, only periodic pair-wise connectivity is required to ensure eventual message delivery. The Epidemic Routing protocol

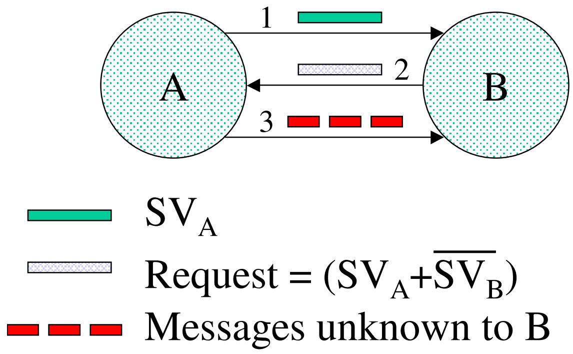


Figure 2: The Epidemic Routing protocol when two hosts, A and B, come into transmission range of one another.

works as follows. The protocol relies upon the transitive distribution of messages through ad hoc networks, with messages eventually reaching their destination. Each host maintains a buffer consisting of messages that it has originated as well as messages that it is buffering on behalf of other hosts. For efficiency, a hash table indexes this list of messages, keyed by a unique identifier associated with each message. Each host stores a bit vector, called the *summary vector* that indicates which entries in their local hash tables are set. While not explored here, a "Bloom filter" [4, 12] would substantially reduce the space overhead associated with the summary vector. When two hosts come into communication range of one another, the host with the smaller identifier initiates an *anti-entropy session* (this term is borrowed from the literature [35]) with the host with the larger identifier. To avoid redundant connections, each host maintains a cache of hosts that it has spoken with recently. Anti-entropy is not re-initiated with remote hosts that have been contacted within a configurable time period.

During anti-entropy, the two hosts exchange their summary vectors to determine which messages stored remotely have not been seen by the local host. In turn, each host then requests copies of messages that it has not yet seen. The receiving host maintains total autonomy in deciding whether it will accept a message. For example, it may determine that it is unwilling to carry messages larger than a given size or destined for certain hosts. While we do not experiment with such general policies, we do model a *maximum queue size* associated with each host, which determines the maximum number of messages a host is willing to carry on behalf of other hosts.

Figure 2 depicts the message exchange in the Epidemic Routing protocol. Host A comes into contact with Host B and initiates an anti-entropy session. In step one, A transmits it summary vector, SV_A to B. SV_A is a compact representation of all the messages being buffered at A. Next, B performs a logical AND operation between the negation of its summary vector, $\neg SV_B$, (the negation of B's summary vector, representing the messages that it needs) and SV_A. That is, B determines the set difference between the messages buffered at A and the messages buffered locally at B. It then transmits a vector requesting these messages from A. In step three, A transmits the requested messages to B. This process is repeated transitively when B comes into contact with a new neighbor. Given sufficient buffer space and time, these anti-entropy sessions guarantee eventual message delivery through such pair-wise message exchange.

Our design for Epidemic Routing associates a unique *message identifier*, a *hop count*, and an optional *ack request* with each message. The prototype described in Section 3 does not yet implement ack request. The message identifier is a unique 32-bit number. This identifier is a concatenation of the host's ID and a locally-generated message ID (16 bits each). Assigning ID's to mobile hosts is beyond the scope of this paper. However, if hosts in an ad hoc network are assigned the same subnet mask, the remaining bits of the IP address can be used as the identifier. In our implementation, the hosts in the ad hoc network are statically assigned ID's.

The hop count field determines the maximum number of epidemic exchanges that a particular message is subject to. While the hop count is similar to the TTL field in IP packets, messages with a hop count of one will only be delivered to their end destination. As discussed below, such packets are dropped subject to the requirements of locally available buffer space. Larger values for hop count will distribute a message through the network more quickly. This will typically reduce average delivery time, but will also increase total resource consumption in message delivery. Thus, high priority messages might be marked with a high hop count, while most messages can be marked with a value close to the expected number of hops for a given network configuration to minimize resource consumption.

Given that messages are delivered probabilistically in epidemic routing, certain applications may require acknowledgments of message delivery. The ack request field signals the destination of a message to provide an acknowledgment of message delivery. These acknowledgments are modeled as simple return messages from receiver back to the sender. Of course, the acknowledgment can also be piggybacked with any other message destined back to the sender after the message is successfully delivered. As future work, we intend to experiment with supplementing anti-entropy with the exchange of a "message delivered" vector. This vector can act as both both message acknowledgment and as a capability to free the buffer space associated with messages that have been previously delivered.

Each host sets a maximum buffer size that it is willing to allocate for epidemic message distribution. The buffer size limits the amount of memory and network resources consumed through Epidemic Routing. In general, hosts will drop older messages in favor of newer ones upon reaching their buffer's capacity. Of course, there is an inherent tradeoff between aggregate resource consumption and message delivery rate/latency. To ensure eventual delivery of all messages, the buffer size on at least a subset of nodes must be roughly equal to the expected number of messages in transit at any given time. Otherwise, it is possible for older messages to be flushed from all buffers before delivery. We explore the tradeoff between buffer size and message delivery in Section 3.2.

A number of management strategies are possible for the per-host message buffer. The simplest policy is first-in-first-out (FIFO). This policy is simple to implement and bounds the amount of time that a particular message is likely to remain "live" (i.e., resident in at least one buffer). Once enough new messages have been introduced into the system, older messages are likely to be flushed from most buffers. As long as the buffer size on all hosts is larger than the expected number of messages in transit at any given time, FIFO is a very reasonable policy. However, if available buffer size is limited relative to the number of messages, FIFO is sub-optimal with respect to fairness and quality of service (QoS). For example, a host's aggregate buffer utilization is directly proportional to the number of messages it sends, which may not be fair to other hosts. Further, FIFO does not provide any mechanisms for preferentially delivering or storing high priority messages. Fair Queuing algorithms [10], including Weighted Fair Queuing (WFQ), logically distribute available buffer space among competing hosts, providing differentiated QoS on a per-message granularity. For our experiments, we implement FIFO, but intend to investigate WFQ as future work.

3 System Evaluation

3.1 Implementation

We implemented Epidemic Routing using the Monarch [7] extensions to the ns-2 packet-level simulator. Monarch extends ns with radio propagation that models signal capture and collision. The simulator also models node mobility, allowing for experimentation with ad hoc routing protocols that must cope with frequently changing network topology. Finally, Monarch implements the IEEE 802.11 [34] Medium Access Control (MAC) protocol.

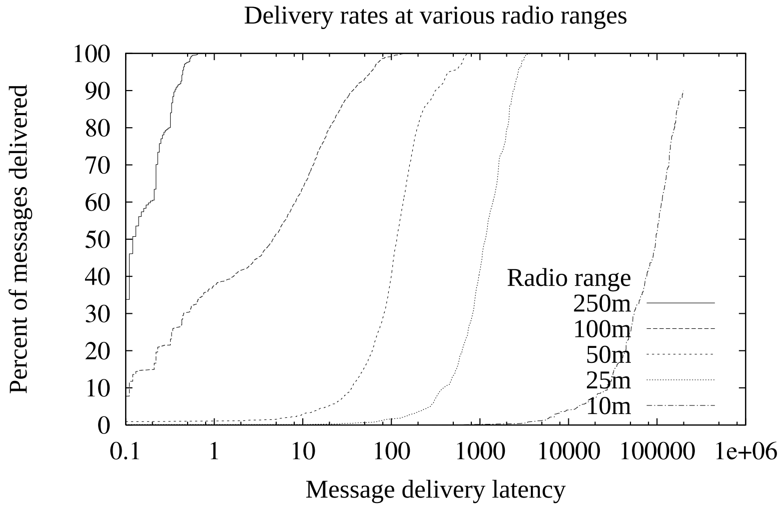


Figure 3: CDF for message delivery as a function of transmission range.

In our implementation, each simulated mobile node has an epidemic routing agent layered on top of the Internet MANET Encapsulation Protocol (IMEP) layer. The IMEP layer is responsible for notifying the epidemic agent when a new node comes into radio range, and when a neighboring node moves out of radio range. These neighbor change notifications are the hooks needed for the epidemic agent to initiate anti-entropy sessions. The epidemic agent itself consists of a buffer for messages the node is carrying, a summary vector for the buffer, and the code required to perform anti-entropy sessions. When IMEP notifies the agent of new neighboring nodes, the agent initiates anti-entropy exchange of messages as described in Section 2.2.

Unless otherwise noted, our simulations are run with the following parameters. We model 50 mobile nodes moving in a rectangular area 1500 m x 300 m in dimension. Each node picks a random spot in the rectangle and moves there with a speed uniformly distributed between 0-20 m/s (average speed of 10 m/s). Upon reaching this point, the node picks a new destination and repeats the process. These parameters are very similar to the default values used in a previous study of various ad hoc routing protocols [5]. Each message is 1 KB in length. We use the following default communication pattern. A subset of the 50 nodes are selected as message sources/sinks, with each of 45 nodes sending one message to 44 other nodes in the system, for a total of 1,980 messages. A new message is “initiated” (meaning ready for transmission) every second, with all messages initiated after 1,980 seconds. By default, each host allocates a 2,000-slot message buffer. While this effectively implies “infinite” buffer space as there are fewer than 2,000 initiated messages, we experiment with the effects of limiting buffer space below.

3.2 Baseline Results

In this section we will explore the characteristics of Epidemic Routing under a number of different scenarios. We first explore the robustness of Epidemic Routing to various radio transmission ranges, between 10-250 meters. Given the baseline configuration parameters described above, Figure 3 plots a cumulative distribution function (CDF) of message delivery latency for various transmission ranges. The percentage of messages delivered is on the y-axis and delivery latency is on a logarithmic x-axis. A key characteristic across all transmission ranges is that 100% of messages are eventually delivered (with the exception of the 10 m case as discussed below). This validates our hypothesis that epidemic algorithms will ensure eventual

Range	Delivery Rate (%)	Baseline Rate	Latency		Hops		Coverage Floor
			Avg (s)	Max (s)	Avg	Max	
250 m	100.0	98.2	0.2	1	2.4	8	10.91%
100 m	100.0	34.3	12.8	177	6.3	21	1.75%
50 m	100.0	0.9	153.0	760	3.7	14	0.44%
25 m	100.0	0.0	618.9	3758	3.3	9	0.11%
10 m	89.9	0.0	44829.7	198107	3.4	9	0.02%

Table 1: Characteristics of Epidemic Routing as a function of transmission range.

message delivery given pair-wise connectivity (which is provided by the continuous random movement of the hosts in this scenario) and sufficient buffering.

With a 250 meter transmission range, messages are delivered in 0.2 seconds on average. We include this case as a reference point because it is used in the literature [5] and because it is the nominal outdoor range for many 802.11 devices. However, Epidemic Routing is likely not entirely appropriate in this scenario because the 250 m range in conjunction with the node density (50 hosts) and coverage area (1500 m x 300 m) means that existing ad-hoc routing protocols would also deliver the same percentage of messages, while consuming fewer system resources by locating efficient routes. The interesting cases for Epidemic Routing are 25 and 50 meter transmission ranges. In these cases, existing protocols would be unable to deliver most messages because they would be unable to locate a connected path from source to destination. Epidemic Routing, on the other hand, is able to deliver all messages with average latencies of 618.9 and 111.6 seconds respectively.

Table 1 summarizes some of the key characteristics of the results depicted in Figure 3. The table depicts the percentage of messages eventually delivered under *Delivery Rate*. As a point of comparison, *Baseline Delivery* shows the percentage of messages delivered had we used the DSR routing protocol in our configuration, as described in [5]. This comparison is not entirely fair to DSR because the communication pattern in [5] is more intensive, involving constant bit rate communication and because DSR is not designed to operate in cases where connected paths are unavailable. We include these only to demonstrate that existing ad hoc protocols break down in the absence of sufficient wireless coverage, whereas Epidemic Routing is able to eventually deliver all messages given pair-wise connectivity. We show DSR delivery rates because comparable results were readily available and because it had among the highest delivery rates of the protocols studied in [5]. The *Latency* column shows average and maximum times in seconds to deliver messages, while the *Hops* column shows average and maximum number of hops that a message took in arriving at its destination. One interesting feature of the table is that the average number of hops increases to 6.3 for the 100 m range and drops back down to 3.7 for the 50 m range. In this case, nodes are on the verge of being fully connected as evidenced by the 34.3% of the packets that are successfully delivered using DSR (which requires full connectivity). Thus, Epidemic Routing transports many packets one hop at a time through the network with little intervening node mobility. At transmission ranges smaller than 100 m, Epidemic Routing relies upon node mobility to transport messages toward their destination, reducing the number of hops but increasing delivery latency.

Finally, *Coverage Floor* presents a lower bound on the percentage of the 1500 m x 300 m area covered by an individual nodes transmitter. This value is calculated by taking the area covered by a transmitter for a particular range and dividing by the total area (i.e., $\pi r^2/450,000\,m^2$ where r is the transmission range in this case). This value is then divided by 4 to arrive at a lower bound, the case where a node is situated in one of the four corners of the rectangle and only has a quarter of its transmission range available to it.

It is interesting to note that with a 10 m transmission (e.g., the nominal range for Bluetooth [15] devices), each node covers only .02% of the total area in the worst case and .07% if a node is at least 10 m from all

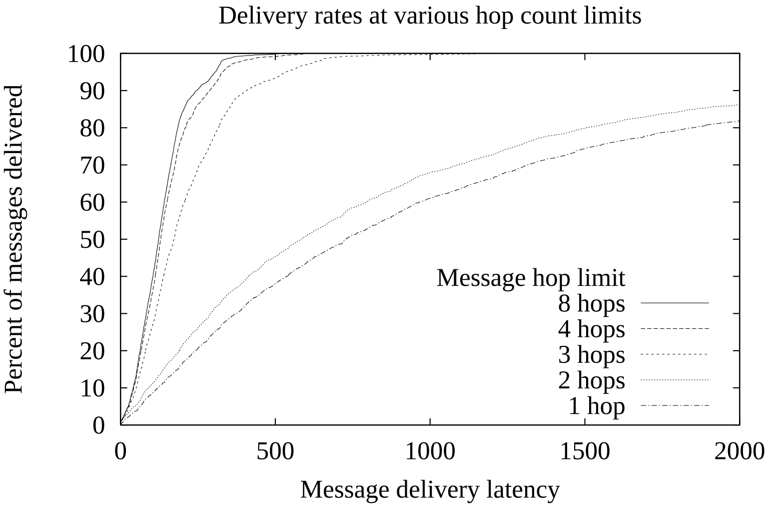


Figure 4: CDF for message delivery as a function of maximum number of hops in packet delivery for 50 m transmission range.

boundaries. Thus, across all 50 nodes, a maximum of 3.5% of the total area is covered at any given time. We included this as a stress case for Epidemic Routing. While it takes over 12 hours to deliver the average message in this scenario, we note that 89.9% of messages are delivered within the lifetime of the simulation (200,000 seconds). Given the shape of the curve and the properties of Epidemic Routing, it is likely that Epidemic Routing would achieve eventual delivery of 100% of messages with a 10 m transmission range given sufficient time. We believe that without the availability of any routing infrastructure, such long delays are inherent given the low coverage densities. Finally, it is important to emphasize that our approach is sensitive to node density and transmission coverage as a function of the total target area. For example, we ran the 10 m simulation in a 100 m x 500 m area (still large, i.e., for mobile sensors) with all other parameters set to their default values, and achieved 100% message delivery with a 9,610 second average delivery time.

3.3 Bounding Resource Consumption

As discussed earlier, there is a tradeoff between memory and network resources allocated to Epidemic Routing and maximizing the percentage of delivered messages. Intuitively, one way to reduce aggregate resource consumption is to limit the maximum number of hops a message can take, which also limits the average number of nodes exposed to a message. Figure 4 shows a CDF for message delivery rate for 50 m transmission range, with multiple curves representing the maximum number of hops that a particular message will take from source to destination (all other parameters are set to their default values). Recall that messages whose hop count reaches 1 will only be delivered to their destination (these messages are dropped subject to available buffer as described below). Figure 4 shows that reducing the hop count to 4 does not adversely affect message delivery rate or latency. Lowering the hop count to 3 still maintains 100% message delivery, though the average latency (not shown) increases by 33%. In general, while lower hop counts continue to deliver most messages, average latency climbs significantly.

Another way to limit total resource consumption is to bound the amount of buffer space available to Epidemic Routing. In order to guarantee eventual message delivery in the worst case, a subset of nodes must have buffer space equal to the maximum number of messages that are in flight at any given time. However, it is typically possible to achieve robust delivery rates with substantially less buffer space. In

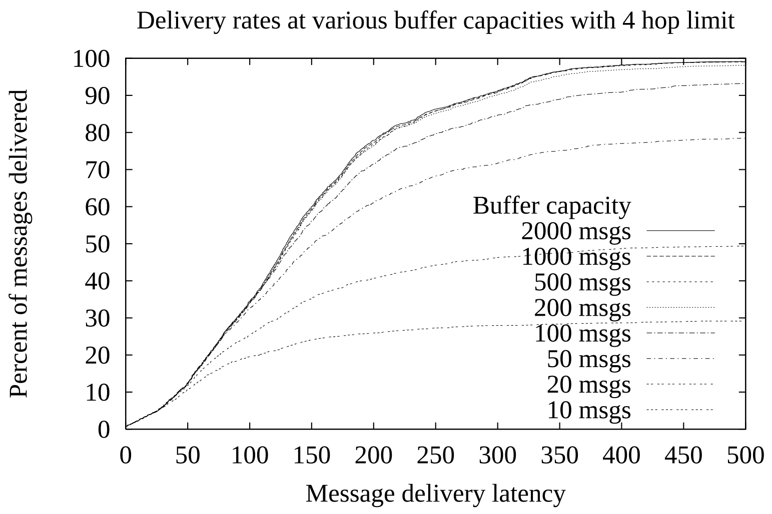


Figure 5: CDF for message delivery as a function of available buffer space for 50 m transmission range.

general, we expect that different nodes will have different buffer capacities (e.g., a small number of high-capacity nodes acting as a delivery "trunk"). For simplicity, we focus on equal buffer capacity across all nodes. Figure 5 explores this tradeoff for the case where each node has a 50 m transmission with a variable amount of available buffer space. For this experiment, we limit hop count to 4 since, as shown above, this value strikes a balance between resource consumption and message delivery (all other simulation parameters are the default values described above). Figure 5 is a CDF with the percentage of messages delivered (y-axis) in at most a given period of time (x-axis). Each curve on the graph represents the delivery rate for a given buffer size. A message buffer with 2,000 slots is effectively equivalent to infinite buffer space since slightly fewer than 2,000 messages are sent during the life of the simulation. As shown in Figure 5, infinite buffer size results in the fastest message delivery, with 100% of messages delivered in less than 700 seconds, and average message delivery in 147.3 seconds. However, buffer sizes of 1,000, 500, and 200 also show good performance, with only slight degradations in both average and maximum delivery times. Once the buffer size is reduced to 50 slots (2.5% of live messages), the total percentage of messages delivered is reduced to 79.7%. While required buffer size and delivery rate is application and scenario specific, a buffer size of between 5-25% of originated messages is sufficient to deliver a high percentage of messages with reasonable latency for our scenario.

Table 2 presents metrics of buffer consumption for the situation depicted in Figure 5. The first column, *Buffer Size*, varies per-node message buffers from 10-2,000. The second column, *Delivery Rate*, shows eventual delivery rate dropping from 100% to 29.3% for 10 per-node buffers. As noted above, delivery rate stays robust through a buffer size of 100 messages in this scenario. The third column shows average latency for delivered messages. The last five columns present a measure of the amount of memory resources consumed for the delivery of each message. The breakdown is split between two types of messages, *Dead* and *Live*, which is the number of each type of message at the end of the simulation. Dead messages are not present in the buffer of any node at the end of the simulation, while live messages that are present in at least one buffer at the end of the simulation. Note that a dead message does not imply that it was not delivered as copies of messages can continue to live in buffers long after message delivery (garbage collection is the subject of future work). For larger buffer capacities, most messages are live because sufficient capacity is present to hold a message in at least one of the 50 nodes. For example, at 1,000-message per node buffer

Buffer Size	Delivery Rate (%)	Latency Avg (s)	Buffer Utilization				
			Dead	Buffers	Lifetime (s)	Live	Buffers
2000	100.0	147.3	0	N/A	N/A	1980	44.6
1000	100.0	148.7	178	23.3	2721	1802	30.3
500	100.0	149.2	992	18.0	1664	988	25.4
200	99.6	152.0	1479	12.1	967	501	19.2
100	95.2	157.5	1708	8.4	691	272	16.9
50	79.7	148.2	1826	5.9	491	154	14.9
20	50.2	129.5	1897	3.9	298	83	11.0
10	29.3	98.9	1935	2.9	198	45	9.9

Table 2: Resource consumption characteristics of Epidemic Routing for 50 m transmission range, 4 hops, and variable buffer size.

capacity, 1,802 messages (out of 1,980 total) remain live in at least one host's buffer. For both types of messages, the *Buffers* column shows the average number of nodes that were buffering a particular message averaged across its entire lifetime. Finally, for dead messages, the *Lifetime* column depicts the amount of time the average message is stored in at least one host's buffer. Thus, for example, with 500-message buffers, 992 messages are eventually dropped from all hosts. Their average lifetime is 1,664 seconds and each message occupies an average of 18.0 buffers during this time. Similarly, there are 988 live messages (still occupying at least one buffer) at the end of the simulation, each of which occupies an average of 25.4 buffers during its lifetime.

To isolate the delivery behavior of a specific set of messages, nodes stop injecting new messages after a pre-determined amount of time. In steady state, if all nodes were to continuously inject new messages into the system (as would be the case for many real scenarios), we expect that all messages would eventually "die" (hopefully after delivery) as they get replaced in buffers by newer messages. Thus, in evaluating the tradeoff between resource consumption and message delivery, the resources consumed on behalf of dead messages are more interesting than those for live messages. It is likely that live messages occupy buffer space simply because they are not competing with any additional new messages. For dead messages, the buffer occupancy numbers multiplied by the average lifetime of the message measures the amount of memory resources required to achieve a given delivery rate and latency. In this way, we are able to capture the tradeoff between resource consumption and message delivery for a given scenario. For our runs, Table 2 shows that higher message delivery rates clearly require larger memory resources. We believe this methodology can be applied in a scenario-specific manner to allow system developers to pick the point in the delivery rate/resource consumption spectrum most appropriate for their application. As discussed earlier, message priorities and WFQ can also be leveraged to provide differentiated QoS on a per-message and per-host basis.

4 Related Work and Future Directions

Epidemic Routing builds upon the results of strong efforts in both ad hoc routing protocols and distributed consensus protocols. A large number of ad hoc routing protocols have been recently proposed [6, 16, 19, 21, 22, 25, 26, 27] possessing relative strengths and weaknesses under different circumstances [5, 8, 18]. However, our efforts are largely orthogonal to the details of the underlying ad hoc routing protocol. In this paper, we focused entirely on techniques for addressing the case where a connected path is not available from source to destination. In real scenarios, we expect that a hybrid approach that first attempts to use end-to-end ad hoc routing and falls back to Epidemic Routing if a path is not available will be most appropriate.

Further, it may be possible to exploit the expected number of hops from source to destination to adaptively switch from epidemic to ad hoc routing with the expectation that a message has reached a connected network subset that includes the destination. We plan to investigate such hybrid techniques as future work.

A number of proposals investigate multicast support in ad hoc routing protocols [13, 33]. Once again, these techniques are appropriate in the case where the network is connected. We observe that Epidemic Routing, by its very nature of widely distributing messages in partially connected networks, is appropriate for supporting multicast in partially connected networks. While strong real-time guarantees cannot be provided for timely delivery, eventually delivering messages to a group of receivers can provide benefits for many applications.

Epidemic algorithms [9] form the basis of our message distribution protocol. Epidemic algorithms were originally designed to provide eventual consistency for replicated databases without requiring any particular replica to be available at a given time. Given random pair-wise propagation of updates among pairs of replicas, all updates are eventually distributed to all replicas. Epidemic algorithms have since been used in a number of contexts, including group membership and weakly-connected (e.g., mobile) scenarios [14, 28, 35]. For our work, we introduce a variant of the general theory of epidemic algorithms by taking advantage of the semantics of our particular application domain. That is, rather than requiring all messages to be eventually seen by all replicas, we desire to have individual messages eventually seen by individual hosts. In fact, for Epidemic Routing it may be desirable to limit the distribution of messages to conserve host resources.

A number of efforts [1, 21] leverage the the global positioning system (GPS) to reduce the search space associated with ad hoc route discovery. We similarly intend to investigate the use of positioning information to aid in restricting resource utilization associated with Epidemic Routing. For example, during anti-entropy, nodes may exchange their current velocity vectors (speed and direction) to determine if a potential carrier is heading toward a region estimated to hold the receiver. In this way, the relative "appropriateness" of a carrier is evaluated to bound the degree to which a message is disseminated.

Query localization [6] uses the principle of spatial locality to reduce the portions of the network probed by reactive ad hoc routing protocols. Specifically, when a given route "breaks" (due to node mobility), new route requests are only propagated if they differ by at most (a configurable) k hops from the previous route. We intend to use a variation on this principle of spatial locality to improve the performance of Epidemic Routing. For example, under certain circumstances there may be locality to the movement patterns of mobile nodes. In this case, it would be worthwhile to exchange a list of the last n nodes encountered by a host during anti-entropy. This information can be utilized to once again identify appropriate carriers under the principal that if a particular host has been seen recently, it will be seen again in the near future.

5 Conclusions

In this paper, we develop techniques to allow message delivery in the case where a connected path from source to destination is never available in mobile ad hoc networks. While existing ad hoc routing protocols are robust to rapidly changing network topology, they are unable to deliver packets in the presence of a network partition between source and destination. For a number of compelling application classes, including mobile sensor networks and disaster recovery scenarios, nodes can be spread over wide geographical distances. Such wide dispersion makes it unlikely that a connected path can always be discovered, making it virtually impossible to perform message delivery using current ad hoc routing protocols. Thus, we introduce Epidemic Routing, where random pair-wise exchanges of messages among mobile hosts ensure eventual message delivery. The goals of Epidemic Routing are to maximize message delivery rate and to minimize message latency while also minimizing the total resources (e.g., memory and network bandwidth) consumed in message delivery. Through an implementation in the Monarch simulator, we show that Epidemic Routing delivers 100% of messages with reasonable aggregate resource consumption for scenarios where existing ad hoc routing protocols are unable to deliver any messages because no end-to-end routes are available.

References

[1] S. Basagni, I. Chlamtac, and V. R. Syrotiuk. Dynamic Source Routing for Ad Hoc Networks Using the Global Positioning System. In *Proceedings of the IEEE Wireless Communications and Networking Conference 1999 (WCNC'99)*, September 1999.

[2] Pravin Bhagwat, Chatschik Bisdikian, Ibrahim Korpeoglu, Arvind Krishna, and Mahmoud Naghshineh. System Design Issues for Low-Power, Low-Cost, Short Range Wireless Networking. In *IEEE International Conference on Personal Wireless Communications (ICPWC '99)*, 1999.

[3] Pravin Bhagwat, Chatschik Bisdikian, Ibrahim Korpeoglu, Mahmoud Naghshineh, and Satish Tripathi. BlueSky: A Cordless Dialup Networking Solution for Palmtop Computers. In *ACM Mobicomm 99*, August 1999.

[4] Burton Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communication of ACM*, 13(7):422–426, July 1970.

[5] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, October 1998.

[6] Robert Castaneda and Samir R. Das. Query Localization Techniques for On-Demand Routing Protocols in Ad Hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, August 1999.

[7] CMU Monarch Project. The CMU Monarch Project's wireless and mobility extensions to ns. ftp.monarch.cs.cmu.edu/pub/monarch/wireless-sim/ns-cmu.ps, August 1999.

[8] S. K. Das, R. Jayaram, N. Kakani, and S. K. Sen. Call Admission and Control for Quality-of-Service (QoS) Provisioning in Next Generation Wireless Networks. In *Fifth International Workshop on Mobile Multimedia Communication (MoMuc'98)*, October 1998.

[9] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic Algorithms for Replicated Database Maintenance. In *Proceedings of the Sixth Symposium on Principles of Distributed Computing*, pages 1–12, August 1987.

[10] Alan Demers, S. Keshav, and Scott Shenker. Analysis and Simulation of a Fair Queueing Algorithm. *Journal of Internetworking Research and Experience*, 1(1):3–26, September 1990.

[11] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, 1999.

[12] Li Fan, Pei Cao, Jussara Almeida, and Andrei Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *Proceedings of ACM SIGCOMM'98*, pages 254–265, 1998.

[13] Mario Gerla, Ching-Chuan Chiang, and Lixia Zhang. Tree Multicast Strategies in Mobile, Multihop Wireless Networks. *MONET*, 4(3):193–207, 1999.

[14] R. A. Golding. A Weak-Consistency Architecture for Distributed Information Services. *Computing Systems*, 5(4):379–405, Fall 1992.

[15] Jaap Haartsen, Mahmoud Naghshineh, Jon Inouye, Olaf J. Joeresson, and Warren Allen. Bluetooth: Vision, Goals, and Architecture. *ACM Mobile Computing and Communications Review*, 2(4):38–45, October 1998.

[16] Z. J. Haas and M. R. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In *Proceedings of ACM SIGCOMM'98 Conference*, pages 167–177, September 1998.

[17] Wendy Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 174–185, 1999.

[18] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 195–206, 1999.

[19] David B. Johnson and David A. Maltz. *Mobile Computing*, chapter Dynamic Source Routing in Ad Hoc Wireless Networks, pages 153–181. Kluwer Academic Publishers, 1996.

[20] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next Century Challenges: Mobile Networking for Smart Dust. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 271–278, 1999.

[21] Young-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–75, November 1998.

[22] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan. A Cluster-Based Approach for Routing in Ad Hoc Networks. In *USENIX Symposium on Location Independent and Mobile Computing*, April 1995.

[23] S.B. Lee, A. Gahng-Seop, X. Zhang, and A.T. Campbell. Supporting Quality of Service in Mobile Ad Hoc Networks. In *8th IFIP International Conference on High Performance Networking (Network 2000)*, May 2000.

[24] Steve McCanne, Sally Floyd, and Kevin Fall. ns - LBNL Network Simulator. See `http://www-nrg.ee.lbl.gov/ns/`, 1996.

[25] V.D. Park and M.S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. of IEEE INFOCOM '97*, May 1997.

[26] Charles Perkins and Elizabeth Royer. Ad Hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[27] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the SIGCOM '94 Conference on Communications Architecture, Protocols, and Applications*, pages 234–244, August 1994.

[28] Karin Petersen, Mike Spreitzer, Douglas Terry, Marvin Theimer, and Alan Demers. Flexible Update Propagation for Weakly Consistent Replication. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP-16)*, pages 288–301, October 1997.

[29] Otis Port. A Few Good Seals Haul Data from the Deep. *Business Week*, February 21, 2000.

[30] Ram Ramanathan and Martha Steenstrup. Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support. *MONET*, 3(1):101–119, 1998.

[31] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 2nd edition, 1995.

[32] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, 1998.

[33] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing. In *IEEE Wireless Communications and Networking Conference*, September 1999.

[34] IEEE Computer Society. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1997.

[35] Douglas B. Terry, Marvin M. Theimer, Karin Petersen, Alan J. Demers, Mike J. Spreitzer, and Carl H. Hauser. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pages 172–183, December 1995.