

# UC Santa Cruz

## UC Santa Cruz Previously Published Works

### Title

Disruption-Tolerant Routing with Scoped Propagation of Control

### Permalink

<https://escholarship.org/uc/item/8z29w64g>

### Author

Garcia-Luna-Aceves, J.J.

### Publication Date

2007-06-24

Peer reviewed

# Disruption-Tolerant Routing with Scoped Propagation of Control Information

J. Boice, J.J. Garcia-Luna-Aceves and K. Obraczka  
Department of Computer Engineering  
University of California at Santa Cruz  
{boice,jj,katia}@cse.ucsc.edu

**Abstract**—We consider the problem of routing messages through a network with episodic connectivity without a priori knowledge of node schedules or locations. We present Steward Assisted Routing (StAR), an efficient loop-free routing framework that can operate in networks that are well-connected as well as in networks that exhibit intermittent connectivity; the proposed protocol uses *steward* nodes to deliver data to destinations that may be partitioned from the source. We also introduce a companion protocol, Scoped Contact and Interest Propagation (SCIP), which contains mechanisms with which dissemination of routing control information for destinations of interest is scoped to a well-defined region of the network. We evaluate our protocols through simulations with four distinct mobility scenarios, including two that are generated using data traces from real networks. Our results show that StAR achieves delivery rates comparable to Epidemic routing with far less signaling overhead. We also show that the addition of SCIP to StAR reduces route maintenance overhead significantly without impacting delivery rates.

## I. INTRODUCTION

The proliferation of battery powered mobile computers with short-range communication devices has resulted in new requirements for the design of network protocols. In particular, communication mechanisms must be developed for environments in which there is no defined network infrastructure. In these multi-hop ad hoc network (MANET) environments, often referred to as "networks without a network", nodes must cooperatively route and forward messages for one another. In addition, constraints on available bandwidth, processing, memory, and power in MANETs invite minimalist protocol design, as excess communication or processing can quickly deplete resources. As a consequence, the original mechanisms used for routing in the Internet have been deemed unsuitable for MANETs, and the notion of *on-demand* routing has emerged. This routing paradigm enables routes to be constructed and maintained only as needed, reducing the communication necessary to establish routes in the presence of frequent topology changes, and minimizing the number of routing entries each node must retain.

As mobile computing devices become more ubiquitous due to lower cost and smaller form factor, they motivate and enable a number of new applications. These emerging applications have challenged one assumption that was made when on-demand routing protocols were first designed; that there always exists a connected route between source and destination nodes, or that in the worst case, network disconnections are short-lived. Networks in which this is not guaranteed have been referred to in the literature as *delay-tolerant*, *intermittently-connected* or *highly-partitioned*. Examples of such networks are numerous. Consider, for example, a situation in which each communication device resides inside a moving vehicle. At some times, source and destination vehicles may be close to each other, and it may be possible to establish a connected route along which to forward messages. At other times, however, no route may be available, and messages must be stored by some vehicle who is likely to contact the destination in the future. Wildlife tracking is another situation in which a delay-tolerant protocol can be

useful. For example, each animal is tagged with a sensor pack (e.g., outfitted in a collar), consisting of sensors such as GPS, temperature, accelerometer, or others. Sensed data are locally stored until an opportunity arises to pass the data to another node such that it nears the information sink, which in turn will deliver data to the end user (in its original form or after some processing).

Clearly, intermittent network connectivity, where disconnections could be long-lived, changes the routing problem drastically. It is no longer sufficient to choose, given a current snapshot of the network topology, the best route based on some metric. Instead, past and/or future topologies must be considered to determine the best route over both space and time.

In this paper, we introduce a routing framework designed for use both in networks that are always connected and in networks that exhibit intermittent connectivity. We extend the notion of on-demand routing to its logical equivalent in disruptive networks, and, in doing so, make two notable contributions:

- We propose Steward Assisted Routing (StAR), which provides loop-free routes in both connected networks and in networks with intermittent connectivity. StAR assumes no a priori knowledge of node schedules or location and leverages past connectivity history to predict future routes.
- We introduce Scoped Contact and Interest Propagation (SCIP) as a technique for limiting the temporal and spatial scope of control information dissemination.

StAR uses *steward* nodes to deliver data to the destination on behalf of the source. A steward can be the destination itself (if there exists a direct route connecting source and destination), or another node deemed likely to have a path to the destination in the future. StAR ensures that forward paths are preserved toward destinations of current interest, while SCIP minimizes the amount of routing information disseminated throughout the network. SCIP ensures that only those nodes who may at some point be elected as a steward for a destination maintain related routing information. Just as on-demand routing limits routing table entries to those nodes on a direct path between source and destination, SCIP limits routing table entries to only those nodes who are likely to be on a path between source and destination at some point in the future. StAR does not require time synchronization among nodes, nor does it require location information (e.g., from GPS), since routes are established based purely on contact histories; information on recent connectivity.

It is possible that a MANET is sometimes connected while at other times, highly-partitioned. To our knowledge, no protocol has been explicitly designed for such networked environments where connectivity can vary over time, and thus can *sometimes* use traditional on-demand routing, while at other times must generate routes opportunistically. Existing routing protocols for ad hoc networks, such as AODV [21], DSR [15], and OLSR [6] do not have the

desirable quality of opportunistic forwarding necessary for networks with episodic connectivity.

In this paper, in addition to a detailed presentation of StAR, we discuss the impact of some tradeoffs in delay-tolerant networking, namely message replication and contact metrics. We discuss existing approaches to routing in delay-tolerant networks in Section II, and describe the proposed protocols in Section III. Our results in Section IV show that StAR provides levels of delivery comparable to Epidemic routing [27], an existing routing scheme designed for networks with disruptive connectivity, with far less overhead. We also show that we can reduce storage and signaling overhead even further through SCIP, particularly when there are a limited number of destinations in the network.

## II. RELATED WORK

Disruptive networks (also referred to as delay-tolerant, partitioned or disconnected networks) have recently received considerable attention from groups researching topics ranging from interplanetary research [2] to wearable computers [7] and wildlife tracking [16].

Since 2002, the DTNRG (Delay Tolerant Networking Research Group) [8] has been active in designing and implementing architectural standards and conducting research in the area of disruptive networks. The group has introduced the concept of *bundle* [9] layer that exists above the transport layer and groups messages into bundles that encompass entire sessions. They have also researched *custody* [10], an approach to reliability in disruptive networks and designed addressing and naming schemes [3] for such networks.

Beginning with Vahdat and Becker's *Epidemic routing* [27], in which all packets are forwarded to all neighbors, message redundancy has played a large role in delay-tolerant networking. Additional message replicas increase the likelihood of delivery and decrease end-to-end delay by selecting all or many possible paths to their destination. In networks with many nodes and/or data packets, however, the energy and transmission requirements of this scheme can be prohibitively expensive, and recent research [11], [19], [24], [25] has improved on Epidemic routing by seeking to control message flooding.

Other research has focused on nodes with predictable or controllable mobility patterns that aid or enable routing in partitioned networks. Shah et al.'s *Data Mules* [23] are mobile devices that provide connectivity to sparse sensor networks with scheduled trips to retrieve messages from data sources and deliver them to their intended destination. More recently, *Message Ferries* [28], [17], [29] and other specialized devices [18] have been proposed as nodes whose mobility can be controlled to provide connectivity to networks as needed.

Meruga et al. [20] and Jain et al. [14] take advantage of the periodicity inherent to some mobility patterns, and explore routing with perfect knowledge of future links. They propose variants of *space-time* routing tables, and employ, among other methods, a modified Dijkstra's algorithm to determine shortest paths over time in these structures. Although these tables rely on the availability of node contact schedules, the routes they form will be, without regard to conditions such as congestion and storage space, optimal.

Many metrics to approximate the distance in time and/or space to a destination have been proposed for use when node schedules are not available. Some approaches [26], [5] have relied on past mobility and topology knowledge to predict future behavior. The utility functions to determine these links differ, although they all rely on the premise that links that once existed are likely to exist again in the future. One notable protocol, *MaxProp* [1], showed better performance in a deployed network of buses than an oracle with

perfect schedule knowledge. The improvement was attributed to the load balancing implicitly caused by MaxProp's imperfect predictions of future routes.

*Pocket Switched* networking [13], [4] is another phrase for networks in which connected routes are not always guaranteed. The focus of this research has been to model the distribution of contact and inter-contact times in order to better design forwarding strategies. They have found that, in several traces of human mobility data, these distributions have followed an approximate power law. This is further evidence that mobility models commonly used in networking, such as the random waypoint model, are not representative of human-induced mobility. Their observations lead to some proposed mechanisms to extend opportunistic forwarding protocols for better performance in such environments.

In Table I we present a comparison among current approaches to routing with regard to some of the issues we address in this paper. We have included one example from each class of delay-tolerant protocols as they were described in this section. Our work fits in the category of those protocols that make use of routing metrics, as opposed to more global topology knowledge. Unlike these protocols, a primary design goal has been to limit the amount of control information disseminated, while satisfying delivery requirements.

## III. STEWARD ASSISTED ROUTING

StAR routes messages using a combination of global (network-wide) contact information and local (intra-partition) route maintenance. The topological location of active destinations in the network is propagated through periodic broadcasts, or *contact exchanges*, between neighbors. Contact exchanges and the way in which contact information is integrated into a node's routing table will be described in Section III-A.

Routes are established based on node routing tables, and allow packets to be forwarded at any time. The rest of this section presents StAR by describing its main protocol components, namely:

- Using the *Scoped Contact and Interest Propagation* protocol (SCIP), described in Section III-B, data sources initiate interest messages for destinations to whom they will send data. An interest in a destination indicates that nearby nodes should maintain and advertise a *contact value* for that destination, and propagate this value to other nodes who may be interested in its topological location.
- *Successor Maintenance* (refer to Section III-C) preserves a total ordering on the network with respect to each destination. Since each node has a contact value for destinations of interest, we define the successor relation to determine which of two neighbors is the successor with respect to that destination.
- Using this ordering, *Steward Maintenance* selects a node in each connected component as the steward for each destination. This is the only node in that partition who has no successors with respect to that destination, and has available space in its message buffer.
- Finally, in the *Data Forwarding* phase, messages intended for that destination are routed to the nearest steward until the destination itself is available to receive the data, or a new steward is nominated.

### A. Contact Values and Routing Tables

The routing table in StAR is destination-indexed, and is used in two contexts. The first context is network-wide information regarding the destination, i.e., when the node has last heard from that destination, and how far away it was. The second is local information, i.e.,

TABLE I  
COMPARISON OF ROUTING PROTOCOLS.

	AODV/DSR	OLSR	Epidemic	Space-time (Oracle)	MaxProp	StAR/SCIP
Route establishment	Route requests and replies	Proactive/MPR	None	Local computation	Local computation	Flooded interests
Topology information	Current state	Current state	None	Global	Past/current state	Past / current state
Enables routing with intermittent connectivity	No	No	Yes	Yes	Yes	Yes
Routing table size	Active routes	All nodes	None	All nodes	All nodes	Nearby destinations with interest
Message replication	Single copy	Single copy	One per node	Single copy	Multiple copies	One / two copies

whether the destination is currently in the same partition, and if not, who its local steward is.

Entries in routing tables are maintained through periodic contact exchanges, in which a node advertises entries in its own routing table. These broadcasts occur at a fixed interval if there are nearby nodes, and contain only those entries in the routing table that may have changed since the last broadcast to the same set of neighbors. The broadcast includes a unique sequence number indicating the broadcast from which the information came. Table II indicates fields of interest in the routing tables. This notation will be used in the remainder of the paper, and individual fields will be described in future sections.

TABLE II

NOTATION AND ROUTING TABLE FIELDS (FOR NODE  $a$ ) USED FOR STAR.

$d$	The destination of interest
$s_{ad}$	Sequence number known by $a$ for $d$
$h_{ad}$	Hops taken by $s_{ad}$ to reach $a$
$n_{ad}$	The address of $a$ 's local steward for $d$
$s_{sd}^a$	Sequence number at $n_{ad}$ for $d$ , as known by $a$
$h_{sd}^a$	Hops taken by $s_{sd}^a$ to reach $s$ , as known by $a$
$r_{ad}$	$a$ 's next hop towards $d$

A neighboring node hearing a broadcast incorporates the routing information into its own routing table. In particular, it calculates a contact value for each destination based on the most recent sequence number from the destination. This metric is described further in Section III-C.

### B. Scoped Contact and Interest Propagation

As described above, STAR's routing table is indexed by destinations in the network. In this section, we introduce Scoped Contact and Interest Propagation (SCIP), which provides two advantages:

- 1) Nodes only receive contact information for those destinations in whom someone has shown interest.
- 2) Nodes only retain contact information for those destinations for whom they may become a steward.

As a simple example we consider the behavior of SCIP in a network with a single data destination and multiple data sources; such a network could be formed with mobile sensors and a stationary sink. Without a notion of scoped contact information, every node would have an entry in their routing table for every other node in the network. With SCIP, however, each node maintains only a single entry (for the sink) in their table. Furthermore, the entry exists only at those nodes who are located between some source and the destination; all other nodes need not retain any contact information.

SCIP ensures the first fact, that nodes only receive contact information for valid destinations, with interest messages generated at each

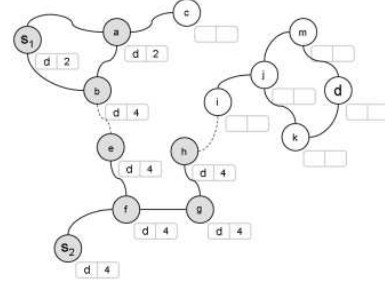


Fig. 1. Both  $S_1$  and  $S_2$  express interest in  $d$ , shown in  $\langle destination, duration, hops \rangle$  messages. These interests propagate across components until they reach their intended destination. The dotted links  $be$  and  $hi$  indicate links that are frequently unavailable.

data source. These interest messages define the destination, as well as the amount of time during which the source wants the location of the destination; after this amount of time, all nodes may drop that destination from their routing table. Because many nodes may show interest in the same destination, SCIP ensures that table entries will only be dropped when there are no remaining nodes with interest in that destination. Interest messages also indicate the distance in hops to the source of the interest, thus, the interest message consists of the tuple  $\langle destination-address, duration, hops \rangle$ . Once the message reaches its intended destination, that node is added to neighboring routing tables, and the location of that destination begins to propagate back towards the sources. Figure 1 illustrates a small network in which two sources have expressed interest in the same destination.

In order to determine the duration of an interest, these messages contain a shared timer, which indicate the remaining amount of time for which *somebody* has interest in that destination. For example, to express interest in a destination  $d$ , a source node  $a$  initiates an interest message for  $d$ , along with the amount of time for which  $a$  has interest in  $d$ . If  $a$  already knew of an interest in  $d$ , it will update the interest timer to be the maximum of the existing timer and  $a$ 's new interest timer. In a subsequent contact exchange between  $a$  and a neighboring node  $b$ ,  $b$  will learn of  $a$ 's interest in  $d$ , and add  $d$  to its own list of interests, along with the remaining amount of time in the interest timer. These timers do not require time synchronization between nodes, since they are simply durations relative to an arbitrary current time.

To further reduce the contact overhead disseminated throughout the network, each destination has knowledge of the number of hops  $h_d$  to the farthest node with interest in that destination. Using this distance, the destination establishes a horizon equal to  $h_d + \epsilon$  for the spread of

its contact value with a ttl (time-to-live) field in its contact exchange broadcasts equal to the maximum distance. The ttl is decremented until the contact update has spread as far as indicated. This prevents the overhead of far-away nodes retaining and advertising contact information for destinations for whom they will never route messages.

A reduction in routing table size results in smaller and/or fewer contact exchange messages and less computation to maintain routing entries. Furthermore, with fewer entries, it is more likely that tables will remain unchanged between contact intervals, and will not need to be transmitted in the next interval. It should be noted that, as described in the next section, the actual data contained in contact value updates in StAR is quite small, since we rely only on sequence numbers and hop counts. However, the mechanisms in SCIP can easily be generalized to other sources of data that could be much larger, for example, more detailed node schedules, geographic locations, node trajectories, or content information. The most prominent drawback to interest-based routing tables is the same faced by on-demand routing; initial delays in route establishment. In this case, an interest for a previously unrequested destination will need to propagate all the way to that destination before a route can be formed. This delay, however, can be quite small when averaged among all transmitted messages. We explore the impact of SCIP on StAR further in Section IV-A.

### C. Successor Maintenance

Contact values are used in the successor relation  $\succ$ , with which we maintain a total ordering over all nodes with respect to any destination. This relation is used to determine both the local steward for any destination and the best route to that steward.

Using the sequence numbers stored in routing tables, we can ensure an ordering among all nodes in the network with respect to each destination. The successor relation  $\succ$  provides a total ordering; for any two nodes  $a$  and  $b$ , and destination  $d$ ,  $\{a_d \succ b_d\}$  or  $\{b_d \succ a_d\}$ . If  $\{a_d \succ b_d\}$ , node  $a$  is a successor of node  $b$  with respect to destination  $d$ , and messages for  $d$  will be routed to  $a$  rather than  $b$ .

The  $\succ$  relation is defined with three components: 1)  $s_{ad}$ , the maximum sequence number for node  $d$  known to node  $a$ , 2)  $h_{ad}$ , the number of hops from node  $a$  to node  $d$  taken by the last sequence number update, and 3)  $n_a$ , a unique identifier for node  $a$ . Successors are determined as follows.

$$\{a_d \succ b_d\} \text{ iff } \\ (s_{ad} > s_{bd}) \text{ or } \\ (s_{ad} = s_{bd} \text{ and } h_{ad} < h_{bd}) \text{ or } \\ (s_{ad} = s_{bd} \text{ and } h_{ad} = h_{bd} \text{ and } n_a > n_b)$$

Thus, a node with a more recent sequence number for a destination is defined as closer to that destination by the  $\succ$  relation. Because it is likely that a sequence number update for a given destination will quickly spread to an entire partition, the number of hops to that destination is used to further refine the relation. In this manner, the node that was responsible for bringing a sequence number update into its current partition is the most likely local steward for that destination. If two nodes are equally far from a destination, with the same recorded sequence number, the node with the greater id is chosen as the successor.

In a contact exchange between node  $a$  and neighboring node  $b$ , for example, node  $a$  uses the advertised information from node  $b$  to compute  $s_{ad}$ , its sequence number for the destination  $d$ , and  $h_{ad}$ , the number of hops taken by this sequence number update. If  $b$  advertises a more recent sequence number for  $d$ ,  $a$  updates its routing table entry for  $d$ .

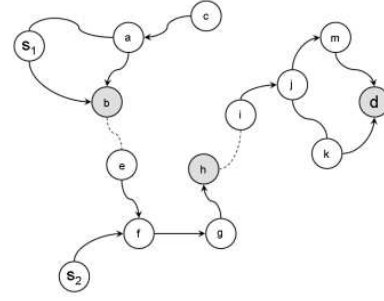


Fig. 2. StAR results in one tree per destination per partition, each rooted at the local steward. In this network, there are three partitions and one destination, resulting in three trees.

### D. Steward Maintenance

Because the successor relation provides a total ordering, we are guaranteed one node in each partition that has no successors with respect to some destination. This node is the local steward for that destination, and all messages in this connected component for that destination are routed to this steward. It is possible that this steward is the destination itself, since if the destination is directly reachable, it will always have the most recent sequence number and fewest hops to itself, and will therefore be its own steward.

Thus, route maintenance results in one tree per destination of interest in each partition, where each tree is rooted at the locally nominated steward for that destination. Note that it is possible (and quite likely) that a node can be the steward for more than one destination at any given time, and the tree for each destination will contain precisely the same nodes and links. An illustration of resulting trees is shown in Figure 2.

Initially, each node nominates itself as the local steward for each destination, and therefore does not route messages to any neighbor. As updates are received from neighbors that advertise better local stewards, the routing trees are formed. In a connected network, each tree will be rooted at the destination itself, and messages routed directly to the destination. Once routes are established, messages can be sent immediately, with no need to wait for opportunistic exchanges. Thus, StAR is well suited for use in both well-connected and highly disruptive networks.

Steward maintenance is performed using the global successor relation to determine the best local steward for each given destination. For each destination, node  $a$  knows  $n_{sd}^a$ , its local steward for  $d$ , and  $s_{sd}^a$ , the most recent sequence number for  $d$  known by the steward  $s$ , as known by  $a$ . Node  $a$  also maintains  $h_{sd}^a$ , the number of hops from the steward to the destination as known by  $a$ .

In an exchange between  $a$  and  $b$ , for example,  $a$  selects  $b$  for  $r_{ad}$ , its next hop to  $d$ , under one of two conditions. Either  $b$  has a steward who is a successor of  $a$ 's steward with respect to  $d$ , or  $a$  and  $b$  have the same steward  $s$ , and  $b$  is a successor of  $a$  with respect to  $s$ . In either of these cases,  $a$  updates its routing entry for that destination to contain the new steward address, and sets  $b$  as its next hop towards  $d$ , as shown in the following pseudocode.

```

for each  $d \in D_a$ 
  do if  $\{n_{sd}^b \succ n_{sd}^a\}$  or  $\{n_{sd}^b = n_{sd}^a\}$  and  $\{b_s \succ a_s\}$ 
    then
       $r_{ad} \leftarrow b$ 
       $n_{sd}^a \leftarrow n_{sd}^b$ 
       $s_{sd}^a \leftarrow s_{sd}^b$ 
       $h_{sd}^a \leftarrow h_{sd}^b + 1$ 
       $s_{as} \leftarrow s_{bs}$ 
       $h_{as} \leftarrow h_{bs} + 1$ 

```

---

### E. Local Loop Freedom

Local loop freedom is provided with each node's knowledge of its steward's sequence number and its distance to that steward. A loop can form only if some node  $a$  accepts a neighbor  $b$  as its next hop towards the local steward  $s$  despite the fact that  $a$  is already a part of  $b$ 's route to its steward. This update can occur only if one of two conditions is satisfied; either  $b$  is advertising a better steward for  $d$ , or  $b$  is advertising a better route to the same steward  $s$ .

If the neighbor  $b$  is advertising a different steward  $s'$  to  $a$ , this means that  $b$  cannot contain  $a$  on its route to the new steward, otherwise  $a$  would be using  $s'$  as its own steward. This contradicts the assumption that  $a$  is on the route from  $b$  to  $s$ , therefore we cannot form a loop in the case of a newly advertised steward.

If the neighbor  $b$  is advertising the same steward  $s$  to  $a$ , with an improved route to  $s$ ,  $a$  will only accept the update if  $b$  is a successor of  $a$  with respect to  $s$ . This cannot be the case, however, because  $s$ 's sequence number must have passed through  $a$  to reach  $b$  (because we have stated that  $a$  is on the route from  $b$  to  $s$ ), therefore  $a$ 's sequence number for  $s$  is at least as recent as  $b$ 's. In addition,  $b$  must be farther than  $a$  from  $s$  in hops to form a loop, so it is impossible that  $b$  is a successor to  $a$  with respect to  $s$ . Node  $a$  will not accept the offer, and no loops can be formed.

### F. Data Forwarding

Once routes are established, messages are forwarded along these paths until a receiving node has no successor for a message's destination; this node is a steward. The steward stores the message until it forms a route to another steward who is a successor for the destination, at which time the message is forwarded to the new steward. A single copy of each message is stored at any time.

For the experiments in this paper, we have implemented a drop-oldest policy for all buffers in StAR. We found that in many situations, the policy for buffer maintenance is data-driven, and highly dependant on what sort of data are most useful to the application. For example, in an animal-tracking scenario, is it more useful to obtain an entire day's worth of data for a single animal, or temporally correlated data for many animals? Since the buffer policy is likely to be selected based on criteria such as this, we have chosen to use the simplest method of drop-oldest for the time being.

## IV. PERFORMANCE EVALUATION

In this section, we present simulation results for StAR and SCIP in the four mobility scenarios described below. We first show that SCIP reduces the routing overhead of StAR significantly. The overhead reduction resulting from SCIP is a function of the number of sources and sinks present, as well as the network diameter. All subsequent simulations used StAR augmented with SCIP.

We also investigate the impact of two types of replication on the overhead-delay metric, and situations in which each can improve

performance. We show that the performance of StAR is comparable to other ad hoc routing protocols in a well-connected network, and that it retains high delivery rates with low overhead as network connectivity diminishes. Finally, we show that StAR can provide delivery rates near to that of Epidemic routing when buffer space is unlimited, and shows far better performance with reduced buffer size or increased amounts of data in the network.

The performance of routing protocols in disruptive networks is highly dependent on several factors including node mobility patterns, network density, and the locations of source and destination nodes. Thus, in some scenarios, it is quite possible that many messages could not be delivered even with an optimal routing protocol given the topologies over the life of the network. In addition, one protocol may perform significantly better than another with respect to delivery ratio. This makes a direct comparison of metrics such as packet delay and route length difficult, since these measurements are dependent on the messages that are successfully delivered. One protocol may deliver the 'difficult' messages, resulting in a higher delivery ratio, but this may also result in a higher per-packet average delay. Thus, delivery ratio must be taken into account when evaluating the measurements of delay and route distance.

We evaluate SCIP and StAR in the Qualnet simulator [22] with the four mobility scenarios described in this section. In all experiments, unless otherwise noted, the simulation includes 100 nodes, runs for a duration of 2000 seconds, and is run with 10 random seeds. There are 25 randomly selected CBR (constant bit rate) flows, each at a rate of 1 packets per second with a total of 1,000 packets sent for each flow. We compare results to Epidemic routing, in which data packets are transmitted to all nodes in the network, and what we call *Successor Forwarding*, in which messages are transmitted to all neighbors who are deemed closer to the destination by the successor relation.

*Random Waypoint*: In this scenario, each node selects a random point in the plane and moves towards it at a uniformly distributed speed between 5 and 10 meters per second. The node then pauses for 10 seconds before selecting and moving to a new destination. Random waypoint mobility is a somewhat anomalous situation for delay-tolerant networking; on one hand, it is the standard against which to evaluate a routing protocol. On the other, the behavior found in such mobility is the worst case for any protocol which seeks to discern trends in node mobility. It is included to illustrate the performance of StAR in a 'traditional', well-connected ad hoc network, and to evaluate different replication schemes in a random environment.

*Gridded*: In this scenario, the two dimensional plane is divided into a grid, in which each cell is of equal size and contains exactly one node. The nodes move within their cell according to the properties of random waypoint mobility. This scenario is of particular use in examining the behavior of protocols as the amount of overlap between radio signal ranges between nodes is increased; from a situation in which node connectivity is guaranteed at all times to larger sizes in which connectivity can be estimated analytically.

*Campus Laptop Trace*: In this scenario, we make use of data available from Dartmouth College's wireless network [12], which contains a continuous log of all wireless cards that are within range of every access point (AP) on campus. We selected the 100 most mobile nodes (based on number of APs accessed) from a one-month period (October 2004), and deemed two nodes who were connected to the same AP at the same time to be within communication range. In addition, we assume each AP to be out of range of all other APs, and we assume that once a node leaves an AP, it is 'turned off' until it resurfaces at another location. This scenario results in one fully-

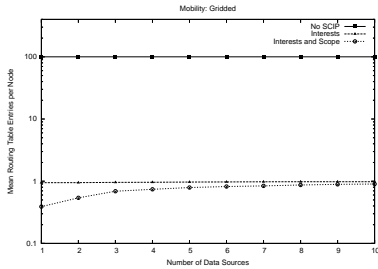


Fig. 3. Scoped interests result in far fewer routing table entries per node with any number of data sources and a single destination.

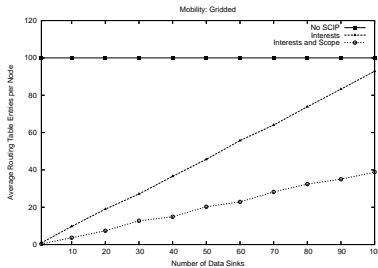


Fig. 4. SCIP provides smaller routing tables, particularly when there are a limited number of data destinations.

connected partition for each AP on campus at all times, which is a bit of a simplifying assumption. It is possible that two nodes who were connected to the same AP did not have direct communication, but they certainly could have had two-hop communication through the AP.

*Scheduled Bus Routes:* In this scenario, we use trace data gathered from the UMassDieselNet project [1]. They have equipped approximately 30 buses with computers and wireless cards that log the times at which two buses are near enough for communication. Each of our simulations represents one day’s worth of trace data, in which all buses act as either a source or destination for data.

#### A. SCIP Performance

We first evaluate the impact of scoping the dissemination of contact information. In all experiments below, the delivery ratio was not affected by the addition of SCIP, which ensures us that contact information was always disseminated to those nodes in need of it. We present the metric of mean routing table entries per node, which, given the mechanisms in StAR, is a good predictor of control overhead. This metric can be extrapolated to other types of contact information such as location, schedule, or content information which may be much larger in size. These experiments were run in the gridded mobility scenario with 100 nodes. In Figure 3, we illustrate the impact of SCIP with a single destination and a varied number of data sources. The top data set indicates performance without any notion of contact information scoping, in which each of the 100 nodes contains every other node in its routing table. Clearly, adding the notion of interests has a dramatic effect on the size of the routing tables, since each interest defines only the single destination. Disseminating contact information only as far as necessary reduces routing table size further, particularly when there are few sources. As more sources are added, the scoping has less effect, as updates are more likely to propagate to the entire network.

In Figure 4, we vary the number of data sinks in the network, while each sink has one related source. Adding interests clearly has the

largest impact when there are fewer destinations, since the interests define only that limited set of destinations. Note that the rate at which routing table size grows in the middle data set is approximately 1:1; as destinations are added, each node must enter that destination in its routing table. However, when information is scoped, the size of the table grows at a rate proportional to the scope of each contact dissemination; in this scenario, the scope was approximately 36% of the nodes.

The other mobility scenarios can also be expressed as a similar percentage of nodes that will receive contact information for a single destination in the average case, which is dependent on the time-based diameter of the network. While the gridded scenario naturally has large diameter, the other two scenarios we consider, the laptop trace and scheduled bus routes, exhibited on average 44% and 52% of nodes that received contact information with a single random source and destination.

#### B. Well-Connected Topologies

We next show the performance of StAR in a well-connected topology. In this experiment, the scenario size was 3600 x 500 meters with 100 nodes, providing full connectivity at nearly all times. Table III compares the performance of AODV, OLSR, and StAR in both static and random waypoint mobilities. With no mobility, all protocols successfully delivered all messages, and delay and route length were similar. StAR incurred higher delay due to slow interest propagation, however, averaged over all packets this delay was very small. When mobility was introduced, AODV and OLSR both resulted in orphaned packets whose routes were changed in transit. StAR combatted this by storing the messages at those intermediate nodes until a new route was established. Again, this led to higher delay, but since the delivery rates are not equivalent, the metric of average delay and route length over all delivered packets is less meaningful. These results indicate that StAR can successfully route packets in a well-connected network, or within a partition in a disruptive network.

TABLE III

STAR’S PERFORMANCE IS SIMILAR TO THAT OF AODV AND OLSR IN A WELL-CONNECTED TOPOLOGY.

Scenario	Deliv (%)	Delay (s)	Hops
AODV Static	100.00	7.92	4.20
OLSR Static	100.00	7.28	4.20
StAR Static	100.00	8.46	4.20
AODV Random	85.82	39.50	3.61
OLSR Random	76.28	41.74	3.63
StAR Random	99.35	44.21	3.84

#### C. Diminishing Connectivity

We next explore the behavior of StAR as the amount of connectivity in the network diminishes. In this experiment, we use the gridded mobility scenario with 49 nodes, and increase the dimensions of the gridded from 1200 meters x 1200 meters to 3000 meters x 3000 meters. The radio range is approximately 450 meters, which results in very low connectivity for the larger dimensions.

Figure 5 shows the delivery rates of AODV, Epidemic routing and StAR as connectivity is decreased. Clearly AODV cannot deliver messages once the size of the grid is too large, while both StAR and Epidemic routing retain delivery rates near to 100%. Epidemic routing transmits all messages to all nodes, resulting in an overhead of approximately 49 messages transmissions for all delivered packets. StAR, meanwhile, maintains extremely low delivery which is in fact

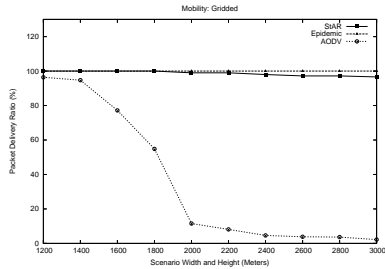


Fig. 5. As connectivity diminishes, Epidemic routing and StAR sustain high delivery rates, while AODV cannot establish routes.

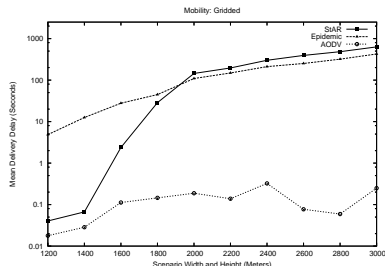


Fig. 6. The delay of StAR is near to that of AODV in a well-connected network, and near to that of Epidemic routing in a highly-partitioned network.

near to that of the route length in all scenario sizes. AODV's overhead rises steadily as its delivery rate drops. Finally, in Figure 6, we show the average delay for delivered messages. In well-connected situations, StAR's delay is close to that of AODV, as we showed in the previous section. Epidemic routing, however, incurs high delay due to the periodicity of its message exchanges.

#### D. Campus Laptop Trace

Figures 7 and 8 show the impact of density on protocol performance in the scenario generated from campus laptop trace data. In this experiment there were 20 randomly chosen flows, while the number of nodes in the network was varied. The additional connectivity clearly improved delivery rates in all protocols, however, it was at the expense of much additional overhead in Epidemic routing and successor forwarding. Since the scenario resulted in dense partitions around some APs (we suppose these were in large classrooms), many messages were lost due to transmission collisions. StAR, with its minimal replication scheme, avoided this issue. In fact, since delivery rates were so high, the overhead in StAR was approximately equal to the route length.

Figures 9 and 10 again illustrate performance in the laptop trace, this time with a varied amount of storage space available to each node. We measure storage space relative to the total number of messages injected into the network. In this situation, StAR performed very well with limited node buffers, again due to its minimal replication strategy. Similarly, the overhead drastically increased in other protocols as each node was able to store and forward most of the messages in the network.

#### E. Scheduled Bus Routes

Results for the scheduled bus mobility scenario were similar to those of the laptop trace scenario. Figures 11 and 12 show delivery rates and overhead of the routing protocols as available storage space increases. Similar to the previous mobility scenario, delivery rates become constant when buffer space reaches approximately 40% of all

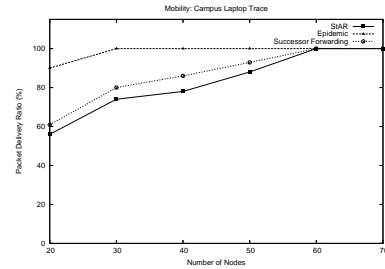


Fig. 7. A denser network results in higher delivery rates for all protocols.

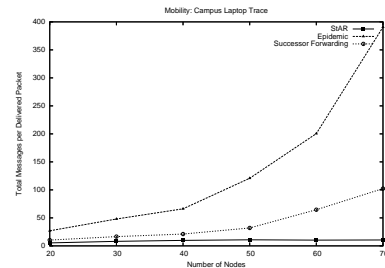


Fig. 8. The overhead of StAR scales well as the density of the network increases.

messages injected into the network. Again, Epidemic routing provides higher delivery rates when storage space is large, while StAR can effectively route messages when there is limited space. Overhead, as in prior scenarios, is lowest with StAR and SCIP. In this scenario, because delivery rates are lower than 100%, even in the best case, the number of messages transmitted per delivery is approximately twice the route length. Figure 13 illustrates a network in which the offered load increases from 1 packet per minute to 20 packets per minute, and storage space is assumed to be large enough to hold all messages. The delivery rate of all protocols suffers with increased traffic, and once traffic reaches a certain threshold (in this case, 10 packets per minute), StAR delivers more messages than the other protocols.

## V. DISCUSSION

We have presented mechanisms with which to limit the scope of contact information dissemination and route messages in a network without schedule or location knowledge. Contact-based forwarding is a solution for those situations where there is no information apart from prior contacts with which routing decisions can be made. Other sources of data, such as partial or full schedules, trajectory or GPS coordinates, or other localization techniques can clearly improve the routing decisions made by individual nodes. We view StAR as a protocol into which other sources of information can be plugged as

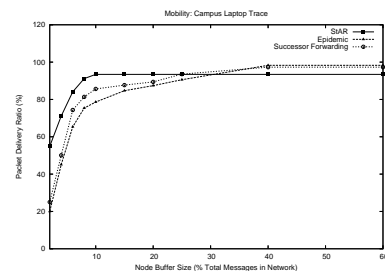


Fig. 9. StAR provides high delivery rates when storage space is limited.



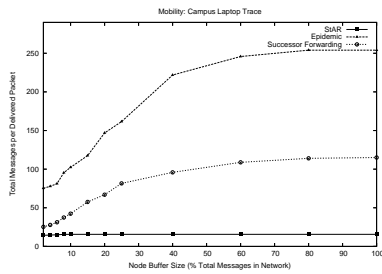


Fig. 10. The overhead of StAR does not increase with more available storage space.

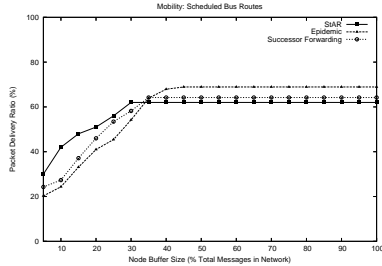


Fig. 11. StAR provides high delivery rates when storage space is limited.

they become available.

Some of the issues we have discussed such as the replication-delay tradeoff and the contact metric used for steward nomination have brought up the issue of modeling mobility scenarios in order to understand in what situations the tradeoffs can be made.

#### REFERENCES

- [1] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *Proceedings of IEEE Infocom*, 2006.
- [2] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, R. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: An approach to interplanetary research.
- [3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant network architecture, 2005.
- [4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proceedings of IEEE Infocom*, 2006.
- [5] X. Chen and A. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Workshop on Principles of Mobile Computing, colocated with PODC'01*, pages 21–27, 2001.
- [6] T. Clausen and P. Jacquet. Rfc 3626: Optimized link state routing protocol (olsr), 2003.
- [7] J. Davis, A. Fagg, and B. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *Proceedings of the*

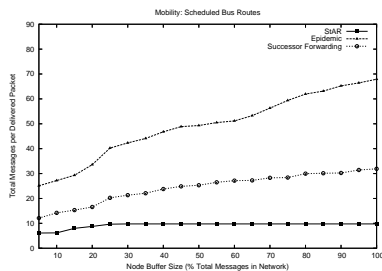


Fig. 12. Overhead remains fairly constant in StAR as available storage space increases.

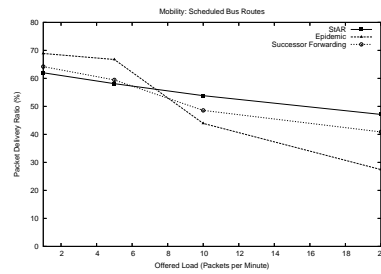


Fig. 13. Increasing the offered load results in diminished delivery rates in all protocols.

*International Symposium on Wearable Computing, Zurich, Switzerland, October 2001*, pages 141–148, 2001.

- [8] DTNRG. Delay tolerant networking research group.
- [9] K. Fall. A delay tolerant networking architecture for challenged internets. In *Proceedings of SIGCOMM*, 2003.
- [10] K. Fall, W. Hong, and S. Madden. Custody transfer for reliable delivery in delay tolerant networks.
- [11] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding schemes in sparse mobile networks. In *IFIP Networking*, 2005.
- [12] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2004.
- [13] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and the consequences of human mobility in conference environments. In *Proceedings of ACM Sigcomm (DTN Workshop)*, 2005.
- [14] S. Jain, K. Fall, and S. Patra. Routing in a delay tolerant network. In *Proceedings of ACM SIGCOMM*, 2004.
- [15] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [16] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of ASPLOS*, 2002.
- [17] H. Jun, W. Zhao, M. Ammar, C. Lee, and E. Zegura. Trading latency for energy in wireless ad hoc networks using message ferrying. In *Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05)*, 2005.
- [18] Q. Li and D. Rus. Communication in disconnected ad hoc networks using message relay. In *Journal of Parallel and Distributed Computing*, volume 63. Academic Press, Inc., 2003.
- [19] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, 2004.
- [20] S. Merugu, M. Ammar, and E. Zegura. Routing in space and time in networks with predictable mobility.
- [21] C. Perkins. Ad hoc on-demand distance vector routing, 1997.
- [22] Scalable-network technologies: Qualnet 3.6.
- [23] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks.
- [24] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *Proceedings of ACM SIGCOMM*, 2005.
- [25] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of SIGCOMM 2005*, 2005.
- [26] W. Su, S. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks, 2000.
- [27] A. Vahdat and D. Becker. Epidemic routing for partially connected ad-hoc networks, 2000.
- [28] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of ACM Mobihoc*, 2004.
- [29] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay tolerant network. In *Proceedings of IEEE Infocom*, 2005.