MV Routing and Capacity Building in Disruption Tolerant Networks

Brendan Burns   Oliver Brock   Brian Neil Levine
Department of Computer Science
University of Massachusetts Amherst

Abstract— **Disruption-Tolerant networks (DTNs) differ from other types of networks in that capacity is exclusively created by the movements of participants. This implies that understanding and influencing the participants' motions can have a significant impact on network performance. In this paper, we introduce the routing protocol *MV*, which learns structure in the movement patterns of network participants and uses it to enable informed message passing. We also propose the introduction of autonomous agents as additional participants in DTNs. These agents adapt their movements in response to variations in network capacity and demand. We use multi-objective control methods from robotics to generate motions capable of optimizing multiple network performance metrics simultaneously. We present experimental evidence that these strategies, individually and in conjunction, result in significant performance improvements in DTNs.**

I.. Introduction

Many routing protocols exist to support end-to-end messaging in ad hoc wireless networks [12], [17]. Such protocols assume an end-to-end connection through a contemporaneous set of links through intermediary peers. As a result, if a path between two peers in a network does not exist, communication is not possible, and the route creation process fails.

To adapt to situations where simultaneous links in the network are not practical or possible, a growing body of work is exploring techniques for moving network traffic over asynchronous paths. Such networks have varied names: highly-partitioned networks [6], [10], message ferrying [24], [25], delay-tolerant networks [8], and disruption-tolerant networks (DTNs) [7]. To enable end-to-end routing in a DTN (the term we choose for this paper), network participants (or peers) are relied upon to carry and deliver messages of others. Whenever two participants pass, they negotiate the exchange of messages. A message may be passed between a number of network participants before reaching its destination.

The performance metrics of disruption-tolerant networks depend on the number of network participants, their storage capacity, communication capabilities, and movement patterns. In this paper, we focus on the performance factor unique to DTNs, namely the movement patterns of participants. We examine how movements can be exploited or controlled to improve performance in DTNs. Movements of network participants are classified according to two independent properties: their inherent structure and their adaptiveness to the demand in the network (see Figure 1). In this context, structure refers to periodic patterns in peers' movements that can be exploited to estimate the probability of delivery for a specific message and peer.

This research was supported in part by National Science Foundation awards ANI-0133055 and EIA-0080199.

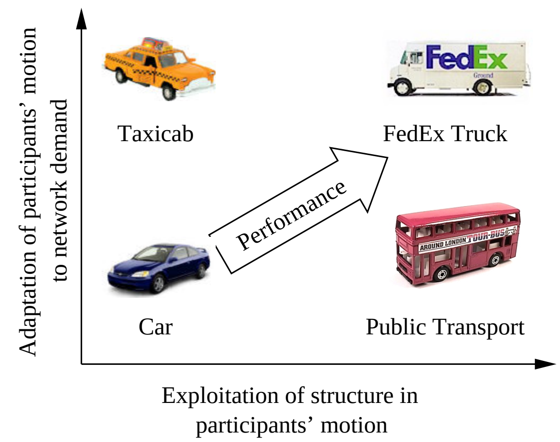


Fig. 1. Classification of routing methods for DTNs based on characteristics of participants' movement patterns.

For purposes of illustration, we relate this classification of participant's motion and associated routing protocols to everyday experience. The lower-left of the classification shown in Figure 1 corresponds to hitchhikers being picked up by randomly moving cars. In this scenario, cars move without periodicity and do not adapt to the route of the hitchhiker. The lower-right of the figure corresponds to public transport with fixed schedules. Independently operating taxicabs that pick up passengers in the street are represented in the top-left of the diagram. Finally, FedEx trucks are situated in the top-right corner. Here, packages are transported, rather than people. FedEx trucks travel on structured daily routes, but only stop for scheduled pickups and deliveries, i.e., their route is adjusted in response to demand.

From this description it is clear that performance metrics, such as bandwidth and latency, can be expected to improve toward the top-right of the diagram. To achieve these performance improvements, an increased amount of coordination among the network participants is required. Such coordination can exploit structure present in participant's motion patters to improve the efficiency of routing. A coordination of network participants

(a) Segway RMP (UMass Amherst)

(b) Helicopter (GeorgiaTech)

Fig. 2. Examples of autonomous agents that could be employed as network participants to improve performance in disruption-tolerant networks [4], [21].

that adapts the motion to the network's demands enables a better usage of the participants' capacity and thus can lead to increased bandwidth and reduced latency. The contribution of this paper lies in algorithms that take advantage of these insights to improve the performance of disruption-tolerant networks.

We present a routing protocol for forwarding of messages from a mobile source to a stationary destination that exploits movement structure by learning the motion patterns of the peers. It maintains information about *meetings* between participants and their *visits* to locations (hence the name *MV*) and uses this information for routing and buffer allocation. *MV* builds on our previous work [6], which kept track only of meetings between peers. We compare our work to a first-in-first-out strategy for managing the buffer allocated for carrying the messages of others. Our simulations show that *MV* performs significantly better, reaching 84% of maximum possible delivery rate, versus 64% for a first-in-first-out buffer management. This advantage is maintained as the offered load in the system and the number of peers increases.

To address the incongruence between the movement of network participants and traffic flow that may arise when movement pattern of participants do not match bandwidth requirements, we propose the introduction of autonomous agents as participants into the network. These agents can be ground-based or airborne mobile robots (see Figure 2). We propose methods for adapting the motion of such agents to bandwidth and latency requirements of a network. While the problem of choosing optimal motions for autonomous agents in this context is shown to be NP complete, we propose techniques from robotic control that are able to obtain high-quality approximations to the optimal solution. We found experimentally that the addition of agents can improve delivery rate by and latency metrics by up to 50%.

II.. Related Work

Since this work is a synthesis of ideas from networking and robotic control, it has related work in both areas.

A. Networking

DTN forwarding has been studied by a growing number of researchers. As we stated in the introduction, we can taxonomize previous work based on their assumptions about the inherent structure of the network and their adaptiveness to the demand in the network.

Ours is the only work that explores the adaptation of peer movements to meet communication needs of the network. We propose the use of robotic peers to improvement performance.

All other work can be distinguished by the degree to which peer movements are fixed and well-known. At one end of the spectrum, Zhao, et al. [24], [25] have proposed DTN networking based on ferries, which are peers that have completely predictable routes through the geographic area (e.g., a city bus or river ferry). Peers route message end-to-end by scheduling their movements to meet with the ferry.

Similar to Zhao, et al. is work from the IRTF Delay-Tolerant Network Research Group [8], where the focus again is on predictable, but non-contemporaneous routing.

At the other end of the spectrum is a series of papers that attempt to learn patterns in the movements of peers.

In 2000, Vahdat and Becker proposed a flooding algorithm called *epidemic routing* [22] that assumes infinite buffer. In 2001, we proposed routing algorithm for highly-partitioned networks by exploring a number of different strategies for deciding which messages to exchange when two network participants meet [6]. Our algorithm, called *Drop-Least Encountered*, had peers keep track of the other peers they meet regularly over time. Peers initialize their estimate of the likelihood of message delivery to a moving peer as 0. When a peer A meets another peer B, the former sets the likelihood of delivering messages to B as 1. Then A takes a portion of B's likelihood of delivering messages to the other peers in the system. These values degrade over time, such that they are reinforced only if A and B meet periodically. Versions of this same algorithm were subsequently proposed by others [15], [18], [9], with each paper showing a different analysis of the problem.

Also relevant to this paper is work by Jain et al [11], who showed that networks that have a large number of connection opportunities require less intelligent forwarding algorithms. As resources become scarce, more information about the network is required for better performance.

There are other challenges within the subject of DTNs. For example, an information retrieval service can be a vital service in a DTN used by disaster management workers. In our previous work, we proposed a method of dividing up a database such that any small random subset of peers can answer queries with high accuracy even though each peer carries only a small fraction of the full database [10]. In our method, no routing is required, yet it is robust despite the movement of peers, who may change groups at any time.

Finally, rumor routing [1] is a related approach to networking in sensor networks that avoids the costs of doing flood routing. In rumor routing each peer passes a message on to each of its neighbors with a probability p. In this way a message is probabilistically insured to travel from source to destination without an explicit route. Rumor routing is focused on networks with stationary peers, but the spread of messages through the network is very similar to that achieved by DTN routing.

B. Multi-Objective Control

The control method we propose for autonomous agents as participants in a DTN is derived multi-objective control in robotics [3], [13]. In general, the goal of multi-objective control is to coordinate a collection of controllers with individual goals to achieve a desired global behavior. Oftentimes it is easier to specify individual controllers that obtain local, atomic goals which are pieces of a larger behavior than it is to specify the complex, high-level behavior directly. The job of the multi-objective controller is to find a coordinated composition of these individual controllers such that the globally desired behavior (in our case the improvement of the DTN) is obtained.

Numerous algorithms for multi-objective control have been proposed, here we discuss those which are directly related to our proposed controller.

In the subsumption architecture [2] each individual controller is a finite state machine with inputs and outputs that may be connected to other controllers or real world sensors/actuators. These controllers are ordered into a layered hierarchy. Multi-objective control and coordination is achieved by having higher controllers modify the inputs or inhibit the operation of lower-level controllers.

The notion of nullspace from linear algebra [14] has also been used to construct multi-objective controllers [3], [13], [20], [21]. The nullspace of a linear mapping A consists of all vectors x such that $Ax = 0$. Here, the nullspace of a controller ϕ is considered to be the collection of control commands that, when performed *in addition* to the controller, do not affect the performance of ϕ. Using the nullspace, multi-objective control is obtained by arranging the controllers into a hierarchy, projecting the behaviors of a lower-level controller into the nullspace of a higher-level controller. At each level of the hierarchy the controller optimizes its actions within the nullspace of higher controllers. Since this optimization takes place in the nullspace of the higher controller, the choice of optimal action at a lower level is guaranteed not to affect the optimality of an action chosen earlier by the higher controller. This is in contrast to the subsumption approach, which achieves coordination through turning individual controllers on and off in a manner specified by the system designer.

Nullspace composition has been applied successfully in a variety of tasks [3], [13], including by Sweeney et al. [20], who used it to maintain network connectivity for distributed agents. In the work, agents maintain line of site (necessary for infrared communication) while pursuing the exploration of an unknown environment.

III.. *MV* FORWARDING ALGORITHM

We propose a new protocol for efficient message delivery in disruption-tolerant networks. Our protocol, called *MV*, learns the frequency of *meetings* between peers and of their *visits* to cells in a geographical grid. The past frequencies are used to rank each message in a peer's buffer according to the likelihood of delivering a message through a path of meetings and locations. Effectively, *MV* exploits the structure present in the peers' motion patterns to improve the efficiency of message delivery. Referring to the earlier transportation analogy, Figure 1, *MV* is situated in the bottom-right of the diagram — it attempts to learn the periodic connections between peers in the system but does not adapt to demand.

A. Assumptions

We make three major assumptions about the type of network that *MV* supports.

1) *Peers have an infinite buffer for the messages they originate, and only a buffer of constant size for the messages of others.* This is the most realistic assumption regarding buffers, as people add sufficient storage for their own needs, but generally limit how much they donate to others.
2) *When peers have an opportunity for transfer, they do so with a fully reliable, infinite bandwidth link layer.* We are trying to isolate and evaluate routing algorithms independent of the limits of the data link layer. Transfer opportunities are limited by meeting duration and bandwidth. However, in the experiments reported in Section V. we found that the meeting time of peers is sufficiently long to make this a realistic assumption given our buffer size parameters.
3) *Messages are delivered to stationary destinations located on a grid.* This last assumption is a design choice. Our previous work [6] considered only mobile-to-mobile deliveries. Here, we prefer mobile-to-stationary transfers here for the following reasons. First, the provision of naming and addressing in DTNs is difficult (i.e., DNS cannot be deployed) and geographic locations are easier to find with GPS. Second, destinations can be assumed to be always on. Third, destinations for all messages might be a public access point leading to the Internet, and those may be static locations. Finally, a mobile-IP-like solution can be employed based on what we propose. A message may be delivered to the geographic location a person is expected to visit often, and from there, a meeting-only strategy [6] could be used to locate the user.

B. *The* MV *Algorithm*

The *MV* algorithm for routing is operates as follows. When a peer A meets another peer B, they perform a message exchange through a number of steps. First, A gives to B a list of the messages A carries as well as their destinations. Each message is also annotated by A with A's likelihood of delivery according to the formula we derive below. A receives the same list from B and calculates the likelihood of delivering B's messages. A now sorts the unioned lists by the likelihood of delivery, removes its own messages, and also deletes all messages that

B has a higher likelihood of delivering. A then selects the top n messages remaining, and requests from B all the messages that are not already stored.

This is the same algorithm for exchange that we created for our previous work [6]. What has changed is the method for determining what messages are most likely to be delivered. Specifically, *MV* determines a probability, $P_n^k(i)$, that the current peer, k, can successfully deliver a message to a destination i within n transfers. Because it is more efficient, *MV* calculates an estimation of delivery likelihood assuming an infinite buffer at each peer and limits the number of hops that are required in practice; experimentally, we show that this assumption provides good performance.

C. Probability of Delivery

We first derive $P_0^k(i)$, the probability that passing a message to some peer k will result in the message being delivered with no more transfers (except the final delivery). In this case, the probability the message will be delivered is precisely the peer's probability of visiting the destination region. We assume that the probability of visiting a region in the future is strongly correlated with the peer's history of visiting a region.

Accordingly, for each peer k, we have a vector P_0^k with one entry for each region. Each entry i of $P_0^k(i)$ is based on the recorded movement of the peer during the last t *rounds*, where a round is a fixed length of time (e.g., 1 hour or 1 day, depending on the movement speed of a typical peer): $P_0^k(i) = t_i^k/t$, where t_i^k is the number of rounds peer k visited region cell i during the previous t rounds. (This average is likely too simple for many contexts and movement patters; clearly, it could be substituted for a more sophisticated statistic, including an exponentially weighted moving average or Markovian process.)

Second, we assume messages can be forwarded to at most one other peer before being delivered to the destination. Both the current peer k and the intermediate peer j have a copy of the message and either (or both) can delivery it.

Let $P_1^k(i)$ be the probability of successfully delivering a message to region i starting with peer k with the help of at most one intermediate peer. This is given by:

$$P_1^k(i) = 1 - \prod_{j=1}^{N}(1 - m_{jk}P_0^j(i)), \qquad (1)$$

where N is number of peers in the system and m_{jk} represents the probability of peer *k* and peer *j* visiting the same region simultaneously. As with the movement probability, we define meeting probability based on meetings during the last t rounds: $m_{jk} = t_{j,k}/t$ where $t_{j,k}$ is the number of times peers j and k are in the same region. Note, $m_{jj} = 1$. Thus, Eq. 1 represents the probability that neither peer k nor any other peer k visits the destination directly. Finally, we assume that messages can be forwarded to no more than n other peers:

$$P_n^k(i) = 1 - \prod_{j=1}^{N}(1 - m_{jk}P_{n-1}^j(i)) \qquad (2)$$

Unfortunately, Eq. 2 does not scale with the number of hops or peers in the system. To calculate the probability, the meeting maps of all other peers must be known. Fortunately, we found in our evaluations that $P_1^k(i)$ is a close enough approximation to $P_n^k(i)$ to serve.

IV.. AUTONOMOUS AGENTS IN DTNS

In this section, we describe how autonomous agents can be deployed in disruption-tolerant networks to increase network performance. This is accomplished by adapting the agents' motion to the demand in the network (Figure 1). We first show that determining optimal motions for agents is NP-hard, providing the justification for the approximation approach presented here. Then, we define methods to optimize particular network metrics. Subsequently, we define a multi-objective controller that coordinates the individual methods.

A. Complexity of Scheduling Agent Movement

The problem of determining optimal motions for agents in a DTN can be stated as a reduced form of the *dial-a-ride problem* [19], which consists of dispatching a vehicle to service a request for an item to be transfered from one location to another. That problem is a generalization of the traveling salesman problem [5], and is known to be NP-hard. A problem related to ours has been shown to be NP hard by Zhao et al. [25].

The reduction of some instance of the dial-a-ride problem to servicing a DTN is as follows. First, note that the graph representing the physical/geographical environment of a DTN is the same as in an instance of the dial-a-ride problem. We assume that at each peer in the graph there is a participant in the network, that each participant is far enough away from any other participant that no point-to-point communication is possible, and that each participant in the network is static.

Every request made to the dial-a-ride system for transport from a location A to a location B is exactly a message in the DTN sent from a peer statically located at location A to a peer statically located at location B. Since all of the participants in the network are static and incapable of communicating, the transport of the message from A to B must be accomplished by the agent. By optimizing the routing of messages by the agent we also obtain an optimal solution to the dial-a-ride problem. Since the dial-a-ride problem is NP-hard and reducible to the problem of routing agents to assist DTN routing, the routing of agents must be NP-hard as well.

B. Performance Metrics

Our aim in deploying autonomous agents in a DTN is to improve a variety of network performance metrics.

- *Bandwidth:* The total number of messages which are currently active in the network.
- *Unique Bandwidth:* The total number of *unique* messages which are currently active in the network (multiple copies of a message may exist in the network).

- *Message Latency:* The average amount of time it takes for a message to be delivered.
- *Peer Latency:* The average time since each peer was last visited by an agent. Since our perception of the network is maintained in a distributed manner, it is important that all of the participants in the network be visited intermittently. Further it insures that no peer starves, unable to send messages.

Each metric provides a specific optimization for the network. The total bandwidth metric measures use of the network; maximizing it ensures that every possible space available for the transport of a message is in use. The unique bandwidth metric measures the usefulness of the bandwidth usage, maximizing it ensures that messages not already in transit are more likely to be selected. Minimizing the message latency metric prioritizes peers which are sending or receiving messages. Minimizing the peer latency metric attempts to prevent starvation of peers whether they are sending or receiving messages.

C. Movement Controllers for Performance Metrics

For each metric, we present a method of determining motion to optimize this single metric. We refer to the algorithm that generates the agent's motion as a *controller*. The bandwidth controller directs the agent to act so as to maximize bandwidth. The latency controller acts to minimize latency, and so forth. The details of the individual controllers are as follows:

Total Bandwidth Controller ϕ_T: Traveling to any peer A will increase the bandwidth of the network by the number of messages which the agent can obtain from peer A. The peer chosen by this controller is the peer which has the largest number of unseen messages, amortized by travel time.

Unique Bandwidth Controller ϕ_U: The unique bandwidth controller chooses the peer that has the largest number of messages not present anywhere else in the network.

Delivery Latency Controller ϕ_D: The delivery latency controller chooses the peer whose average delivery time is the largest.

Peer Latency Controller ϕ_P: The peer latency controller chooses the location n_i least recently visited by an agent, unless there exists a set of locations such that $\text{LastVisited}(n_i) + \text{TravelTime}(n_i)$ is less than $\sum_{n \in N}(\text{LastVisited}(n) + \text{TravelTime}(n_i))$. This insures that traveling time to visit the least recently visited location does not actually cause an increase in the peer latency statistic.

D. Multi-Objective Control

Ideally, each metric could be optimized independently, but in practice the metrics are dependent. In traditional wired networks, a balance between various performance metrics is achieved through the specification of fixed network parameters by a network administrator. In the case of agent-augmented DTN routing, the network's performance must instead be optimized via the use of controllers. To optimize multiple metrics (or objectives) simultaneously, several of the controllers introduced in Section C have to be combined. The task of composing the controllers to achieve the mix of network performance desired by the network administrator can be accomplished with the use of multi-objective control algorithms [20], [23], [21].

1 Nullspace Composition

Informally, the *nullspace* of a particular controller ϕ_1 is defined as the set of actions that can be taken by the agent without affecting the performance of ϕ_1. For example, if n potential motions of an agent optimize the metric encoded by controller ϕ_1 equally well, these n motions represent the nullspace of ϕ_1. To optimize a second metric represented by controller ϕ_2, which we call the *subordinate* controller, one chooses among those n choices a motion that performs best with respect to the second metric. We say that ϕ_2 is optimized in the nullspace of ϕ_1. This permits an implicit ordering of metrics and corresponding controllers, in which the action taken by a subordinate controller never affects the performance of a superior one. The notion of nullspace together with an ordering provides a principled framework for the composition of the controllers introduced in Section C.

Motivated by the application of this framework to DTNs, we define the performance of controllers with respect to a threshold. This means, for example, that any motion of an agent achieving a specified minimum bandwidth requirement is said to perform equally well with respect to this metric. Such a definition of performance permits nullspaces of sufficient size to optimize multiple objectives.

2 Controller Ordering

An ordering of controllers captures the relative importance of the associated network metrics. The specification of such an ordering provides the network administrator with a simple way of specifying criteria for performance optimization. The ordering we use here is given by:

$$\phi_P \leftarrow \phi_D \leftarrow \phi_U \leftarrow \phi_T,$$

where $\phi_i \leftarrow \phi_j$ indicates that ϕ_j is more important than ϕ_i. Using multi-objective control, ϕ_i is optimized in the nullspace of ϕ_j.

The ordering was chosen based on the observation that the nullspace of equal bandwidths is significantly larger than the nullspace for unique bandwidth, and so forth down the ordering. This means that the ordering offers more flexibility for controllers with lower priority.

It is important to note that the above ordering is not the only appropriate one, future work may be warranted to explore the effects of different orderings either specified by administrators or learned automatically. Likewise, the choice of thresholds in the definition of nullspace is up to the end user. A network administrator can manipulate the performance of their network to suit its demands.

3 Subsumption

The subsumption approach to multi-objective control [2] of the DTN agents differs from the nullspace approach. We still use the four controllers described in Section B, prioritized as described in Section D.2. The difference is in how the controllers dominate one another. Whereas in the nullspace approach subordinate controllers optimized the motion of an agent in the nullspace of superior controllers, in the subsumption approach the controller with the highest priority exclusively determines the motion of the agent until its performance threshold is achieved. Only when all superior controllers achieve their performance metric, do subordinate controllers get a chance to optimize their performance. In the subsumption approach, the output of dominant controllers completely *subsumes* the output of the subordinate controllers.

E. Distributed Network State Maintenance

The controllers described in Section C use the state of the network to guide their behavior. In a real network, this global information is unavailable. Each peer in the network has perfect information about its own state, but must estimate the network's overall state. We accomplish this by constructing an approximate distributed model of the network through an additional exchange of information during each encounter between agents.

The approximate global model of the network maintained by each agent contains all of the information (bandwidth, latency, etc.) required by the agent's controllers. Each agent maintains information about every participant in the network. This information is tagged with the time at which the information was obtained. When two peers in the network meet, in addition to exchanging data traveling on the network, they exchange information about the state of the network. First they exchange the perfect information about their own state. Next, for every other peer in the network, they compare the timestamps of the data they are maintaining. The data from whichever peer has a more recent time stamp is propagated to the peer with older information.

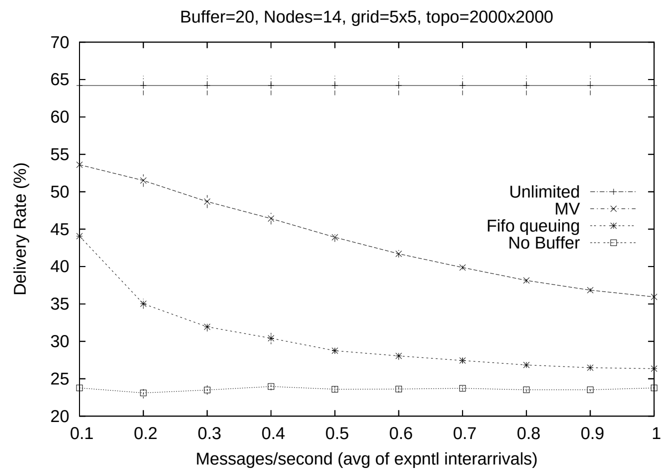


Fig. 3. Delivery rate versus offered load.

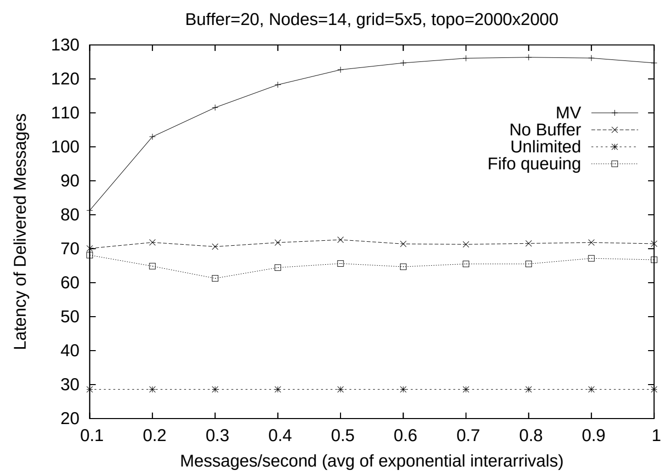


Fig. 4. Latency versus offered load.

V.. Experimental Evaluation

In Figure 1 we classified routing algorithms for DTN according to the degree to which they exploit structure in the motion of network participants and according to the degree to which the participants adapt their movements to the network demand. We now present experimental evidence that *MV* is capable of exploiting the structure in the motion of network participants to improve performance (Section B). When *MV* is employed in conjunction with autonomous agents, further performance improvements can be observed (Section C.2).

A. Methodology

To evaluate the proposed routing algorithm *MV* and the effect of introducing autonomous agents into a DTN, we ran a series of ns2 [16] simulations. We are interested in several metrics of the algorithm: message delivery rate, latency, and duplication of delivered messages at the destination. These metrics are measured over the offered load and the number of peers in the network.

The success of DTN forwarding algorithms is wholly tied to the movement pattern of peers. Traditionally, researchers have used the random waypoint model in lieu of empirical models. Such a movement model cannot be used for our evaluation of DTNs: if peers move randomly, then no peer is any better at delivering a message than any other. A successful routing algorithm exploits structured, distinguishable movements.

We believe movements of humans and vehicles (e.g., buses and planes) are structured. To generate movements which *MV* could exploit, peers move periodically between three geographic locations. Each peer has a home location and two remote locations. Peers move among the three points, with the home location being chosen 50% of the time, and the remote points visited 25% each. Peers move at a uniform speed of 30m/sec (similar to automobiles) in a world that is 2000m-by-2000m. Moving peer's radios reached 250m. As a result, most meetings are sufficiently long in duration to transmit all available data (empirically, we found that 95% of meetings were longer than 10 seconds) and we assumed all were sufficient. There are also 25 stationary *sinks* in a 5-by-5 grid. The sinks

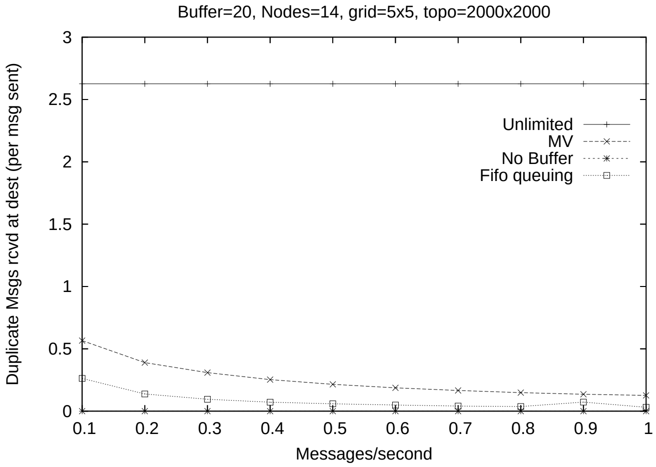


Fig. 5. Duplicates received at destinations versus offered load. (normalized by the number of messages sent).

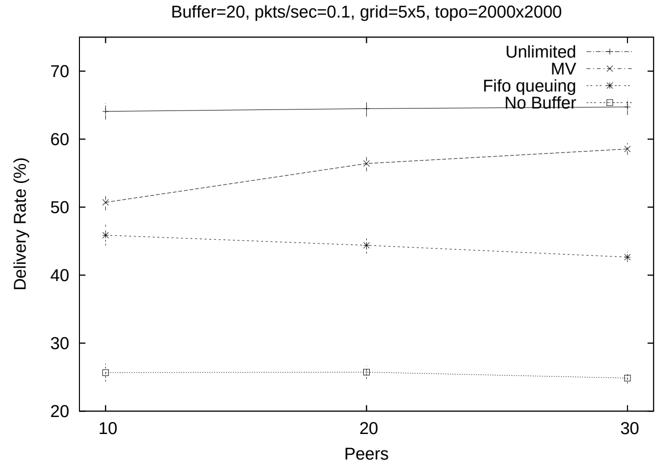


Fig. 6. Delivery rate versus number of peers.

have no storage, do not generate messages, existing solely to accept messages sent to them.

Peers generate messages with uniform inter-arrival times according to a specified mean, varied in the experiments. Buffers at peers are stated in terms of the number of messages they could store. We used a buffer size of 20 messages, each peer also had an unlimited buffer for storing messages it was sending. Each point on the graph represents 10 simulations with 10 different random seeds. Error bars show standard deviation, which is small in all cases. All simulations ran for 1000 seconds; *MV* is not given time to warm up.

B. MV : *Exploiting Movement Structure*

In this first set of experiments, we evaluate the performance gains introduced through exploitation of structure in the participants' movement by *MV* . We compare the performance of *MV* against three other algorithms. First, *no buffer*, where peers can only deliver messages by visiting the destination directly. This shows a lower bound on the connectivity of the network. As an upper bound on connectivity, we use *unlimited buffers* with flooding; if a route ever exists during the simulation, then the message is delivered. The final comparison is with a first-in-first-out (fifo) buffer control strategy: peers take previously unseen messages and when necessary push out the oldest messages in their buffer to make space. Previous work [6] found that delivering messages based on peer meeting probability alone (not considering peer location) is not significantly better than fifo. Therefore, this algorithm is not tested.

Figure 3 shows the effect of offered load on packet delivery rates for various algorithms. *MV* can deliver at 83% of the maximum achievable delivery rate; fifo can deliver 69%. As the offered load increased from an average of 0.1 messages/second to 0.8 messages/second, *MV* maintains a significant advantage. *MV* falls to 56% of the achievable delivery rate, while fifo falls farther to 40%.

Figure 4 compares the latency of delivered messages amongst

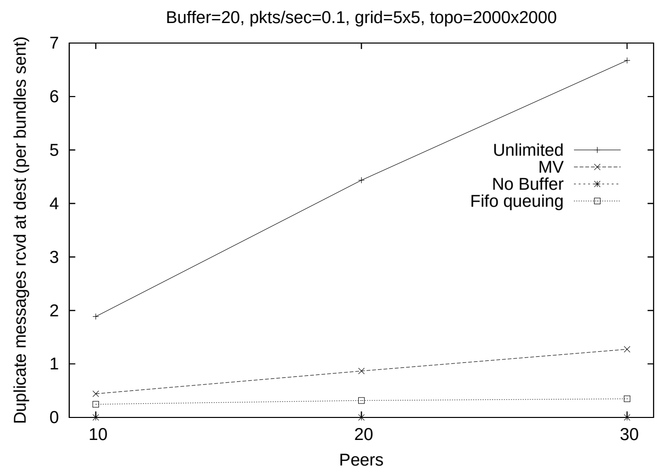


Fig. 7. Duplicates received at destinations versus number of peers (normalized by the number of messages sent).

the algorithms. That *MV* has higher latency is to be expected since it is delivering messages which other algorithms fail to deliver. The average distance traveled by a delivered message is lengthened. None of the other algorithms approach the latency the shortest possible path which is obtained by the unlimited buffer flooding algorithm.

In Figure 5, we see the cost of unlimited buffer: duplicate copies of messages are delivered at the destination. While *MV* delivers more duplicates on average than fifo, this is offset by the better delivery rate. This indicates that the buffers could be used more efficiently by a more sophisticated version of *MV*.

We evaluate the effect of the number of peers moving in the system on performance. As we add peers, each peer adds load by souring more messages, but provides message carrying capacity. Since this is a peer-to-peer system, it is important that the network improve in performance as peers are added. In fact, we see in Figure 6 that in terms of delivery rate this is the case for *MV*, but not fifo. More extensive simulations are needed before we can say conclusively that *MV* is stable, but the results are encouraging. Figure 7 shows that *MV* does not scale well

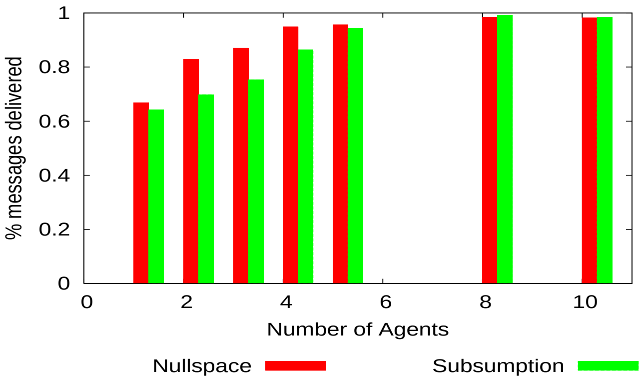


Fig. 8. Delivery as a function of number of agents, 0.1msg/sec.

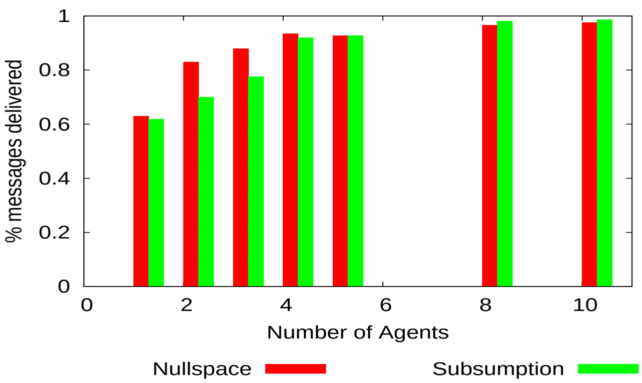


Fig. 9. Delivery as a function of number of agents, 0.2msg/sec.

in terms of the number of duplicates delivered as the number of peers in the system increases. This tradeoff appears worth the higher delivery rate. We can see from the figure that *MV* is using its buffer more efficiently than the flooding strategy.

C. Adapting the Movement of Agents to Network Demand

In this section, we provide experimental evidence that the introduction of autonomous agents improves network performance in DTNs.

1 Comparing Subsumption and Nullspace

To determine the appropriate multi-objective controller to use for autonomous agents in a DTN, we compared the performance of the two algorithms proposed in Section D by simulation. For each controller ten experiments were run with a moderate amount of random network traffic. The averaged results are shown in Figure 8.

The nullspace approach outperforms subsumption when resources (the number of agents) are limited. As the number of agents increases, resources for delivering messages become abundant, and both control algorithms converge on the same upper limit on accuracy. This indicates that the nullspace approach is using limited resources more effectively. When there are more than enough resources to provide effective message transport, the choice of control algorithm does not matter. However, when resources are limited (and thus their allocation more important) the nullspace approach balances the needs of the network while providing improvements in the most important metric, the percentage of messages delivered.

For all subsequent experiments we choose nullspace-based multi-objective control for the autonomous agents. We describe how the introduction of agents affects networks performance.

2 DTN Performance with Agents

We first explore the performance of agent-based DTN routing under increasing bandwidth loads. For this experiment, all of the traffic in the DTN is carried by the agents in the network. The number of agents and the level of traffic is varied. From these experiments we validate the agents ability to deliver network connectivity that could not otherwise exist. A graph of the delivery rate resulting from these variations is shown in Figure 10. The response to increased traffic seems to match that of earlier experiments (Section B). In these experiments, the agents has buffers of size 20, just like nodes.

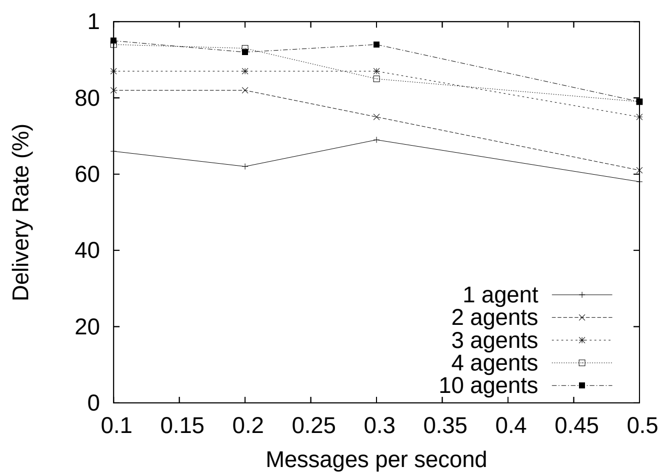


Fig. 10. Percent of messages delivered for different sending rates and number of agents.

To demonstrate that the nullspace multi-objective control provides improvement over less informed behavior, we performed an experiment in which two of the fourteen participants sent messages but neither moved nor passed messages. In addition, we added two controlled agents, which sent no messages but moved in service of the network and passed messages. In this way, both traffic and capacity of the network remained constant, only the movement of the peers changed. Figure 11 shows the resulting improvements in performance. We performed experiments with all peers running MV routing and with all peers running the fifo-queue algorithm. The results of the experiment shown in Figures 11-13 indicate that the addition of peers whose actions are controlled to enhance the performance of the network increases the delivery rate of the network significantly. Additionally, the choice of routing algorithm only becomes significant to the delivery rate at high loads. However, MV routing with agents has lower message latency compared

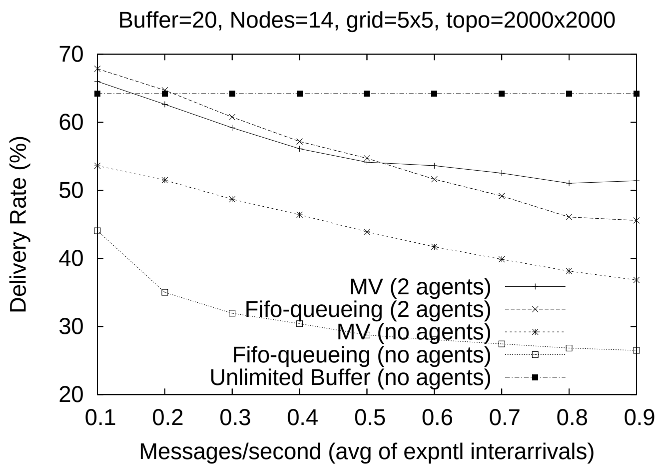


Fig. 11. Delivery rate versus offered load with agent assistance.

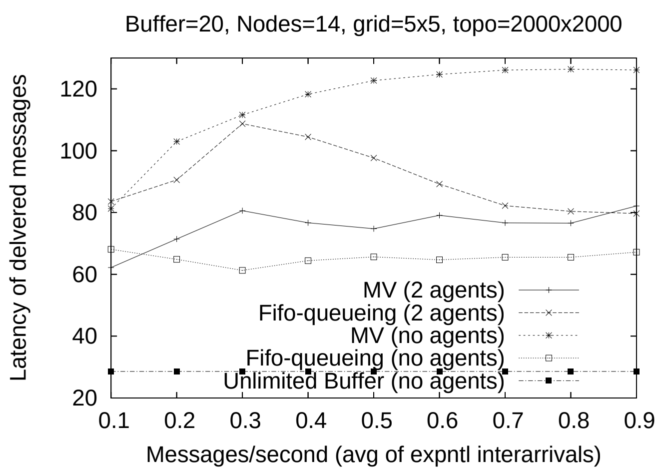


Fig. 12. Latency versus offered load with agent assistance.

to fifo-queue. Thus, MV routing is still the better choice, even with agents carrying a fraction of the network's load. As is to be expected, the addition of agents significantly increases the number of duplicate deliveries (especially at low packet rates) but the duplicate percentage decreases rapidly as the load increases, approaching similar levels as unassisted networks.

D. Evaluation of Distributed Network Statistics

The control strategies are dependent upon a high quality estimate of the state of the network. In a DTN the quality of the estimate is affected greatly by the disruptive nature of the network since peers find it hard to exchange information. To better understand this, we monitor the percentage error of each of the network statistics over time. Additionally we measure the accuracy of the estimate of where each agent is going. The graphs of these results are shown in Figure 14. The results shown are the average error over time during a run with moderate traffic (0.3 messages per second).

It can be seen that both bandwidth and latency error stabilize with low error as the experiment proceeds. Latency error is initially quite high due to pessimistic assumptions when no other information is available. The effect of this pessimistic assumption is also seen in the surge in error in last visited accuracy. The agent location error is relatively stable, although the difficulty in predicting agent location means that the resulting error is generally greater.

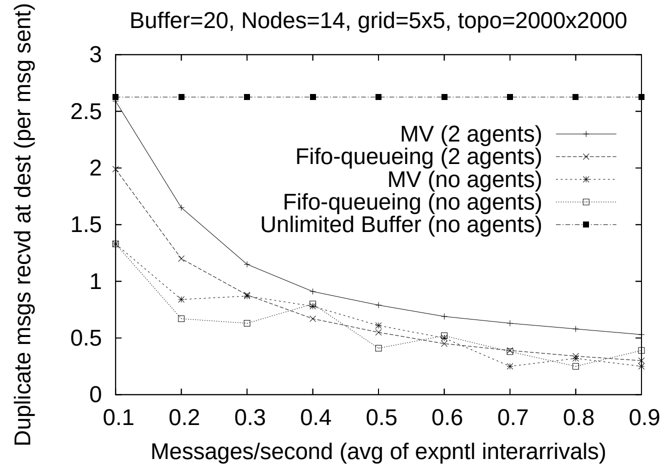


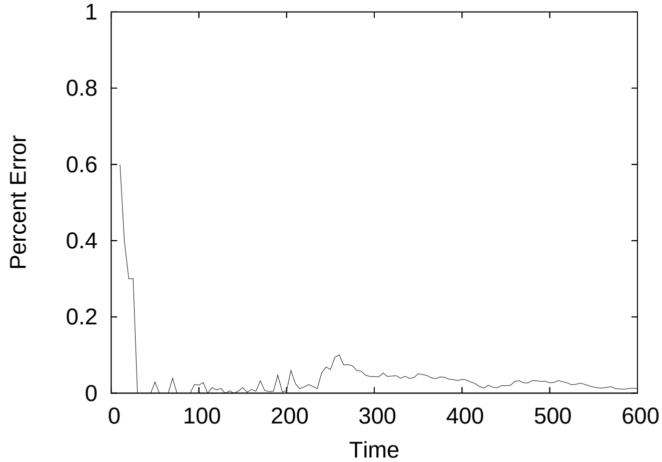
Fig. 13. Duplicates received at destinations versus offered load. (normalized by the number of messages sent) with agent assistance.

VI.. CONCLUSIONS

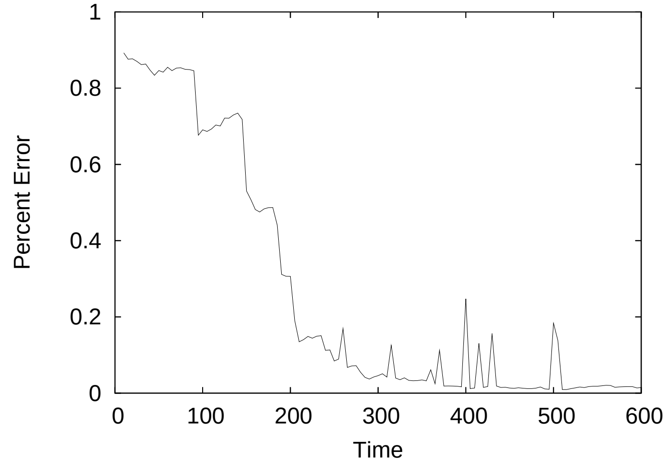
Disruption-tolerant networks require routing algorithms that are different from those designed for ad hoc networks. The capacity of a DTN is provided solely by the motion of its participants. For a routing algorithm to ensure performance under such conditions, it has to explicitly account for this motion in its strategy of forwarding messages. In this paper, we introduce a classification of routing algorithms for DTNs based on this observation. We differentiate routing protocols based on the degree to which they exploit structure in the movement patterns of network participants to improve performance metrics. Along a different dimension, we differentiate them based on the degree to which participants adapt their motions to network demand.

The exploitation of structure in the network participants' movement patters improves performance in DTNs. We introduce the routing protocol *MV*, which maintains a movement model of the network participants and uses this information to perform routing of messages on the network. It estimates the probability of a particular message being delivered by a given peer, and thus is capable of making informed routing decisions. We present experimental evidence that 1routing messages in DTNs using *MV* results in large performance improvements over other techniques in achieving delivery rates significantly closer to the true optimal rate. These improvements continue even as traffic on the network increases by an order of magnitude.

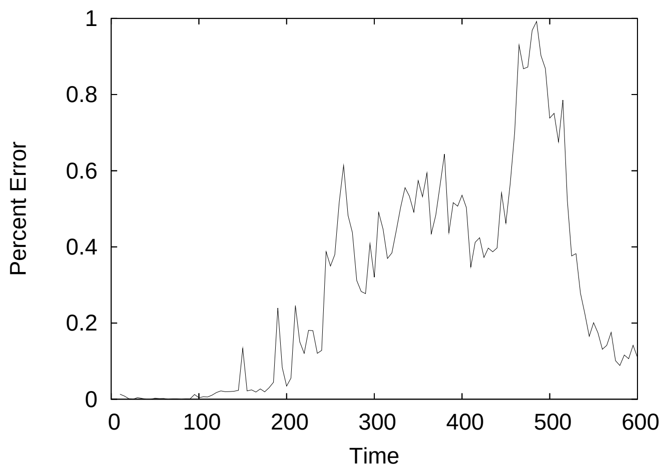
The adaptation of network participants' motion to network demand permits additional performance improvements for DTNs. For this purpose, we propose the introduction of



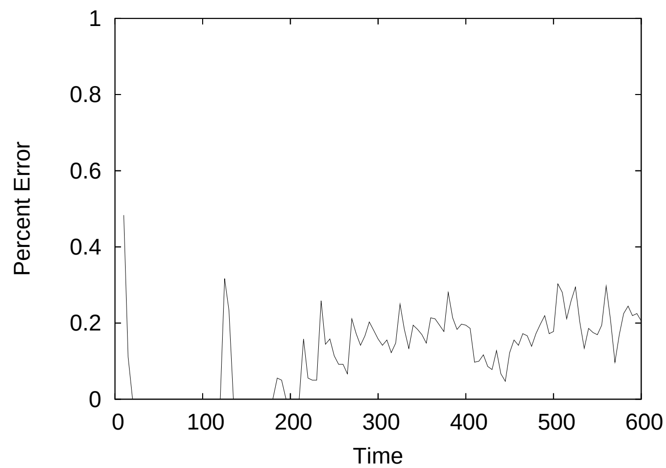
(a) Bandwidth Error



(b) Latency Error



(c) Last Visited Error



(d) Agent Location Error

Fig. 14. Accuracy of the distributed status information over time.

autonomous agents into the DTN. By adapting their motion, these agents are able to compensate for a mismatch between available capacity and demand. We propose multi-objective control algorithms from robotics to control the motion of autonomous agents in order to optimize network performance metrics. These algorithms permit for a simple prioritization among network metrics by network administrators. Experimental results demonstrate that multi-objective control methods are successful at improving network performance by adapting the movements of autonomous agents introduced into a DTN.

Employing a routing strategy based on *MV in conjunction* with multi-objective control for autonomous agents in a DTN is shown to have the most significant performance improvements. This indicates that it is desirable for routing protocols in DTN to exploit the structure present in movement patters of network participants to route messages as well as to change the movement patterns of participants in accordance with network demands.

ACKNOWLEDGMENTS

We are very grateful to Ping Hung Lee for programming many of the *MV* simulations presented in this paper.

REFERENCES

[1] D. Braginsky and D. Estrin. Rumor routing algorithms for sensor networks. In *First International Workshop on Sensor Networks and Applications*, pages 22–31, 2002.

[2] R. A. Brooks. A robust layered control system for a mobile robot. *International Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[3] J. Coelho and R. Grupen. A control basis for learning multi-fingered grasps. *Journal of Robotic Systems*, 14(7):545–577, 1997.

[4] Georgia Tech Aerial Robotics. Available at: http://controls.ae.gatech.edu/gtar/.
[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.
[6] J. Davis, A. Fagg, and B. Levine. Wearable computers and packet transport mechanisms in highly partitioned ad hoc networks. In *Proc. Intl. Symposium on Wearable Computers*, October 2001.
[7] BAA04-13: Disruption Tolerant Networking (DTN), May 2004. Technical POC: Preston Marshall, DARPA/ATO.
[8] Delay-tolerant networking research group. http://dtnrg.org.
[9] M. Grossglauser and M. Vetterli. Locating nodes with ease: Mobility diffusion of last encounters in ad hoc networks. In *IEEE Infocom*, 2003.
[10] K. M. Hanna, B. N. Levine, and R. Manmatha. Mobile distributed information retrieval for highly-partitioned networks. In *IEEE ICNP*, Nov 2003.
[11] S. Jain, K. Fall, and R. Patra. Routing in a delay-tolerant network. In *Proc. ACM Sigcomm*, August 2004.
[12] 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.
[13] O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics and Automation*, 3(1):43–53, 1987.
[14] D. C. Lay. *Linear Algebra and its Applications*. Addison Wesley, 2nd edition, 1997.
[15] A. Lindgren, A. Doria, and O. Schelén. Poster: Probabilistic routing in intermittently connected networks. In *Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, Annapolis, MD, June 2003.
[16] The Network Simulator, NS-2. http://www.isi.edu/nsnam/ns.
[17] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Compting Systems and Applications*, pages 90–100, February 1999.
[18] N. Sarafijanovic-Djukic and M. Grossglauser. Last encounter routing under random waypoint mobility. In *Networking 2004, The Third IFIP-TC6 Networking Conference*, May 2004.
[19] M. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 6(4):285–305, 1985.
[20] J. Sweeney, H. Li, R. A. Grupen, and K. Ramamritham. Scalability and schedulability in large, coordinated, distributed robot systems. In *Proceedings of the International Conferance on Robotic Applications (ICRA)*, 2003.
[21] B. Thibodeau, S. W. Hart, D. R. Karuppiah, J. D. Sweeney, and O. Brock. A cascaded filter approach to multi-objective control. In *Proceedings of the International Conference on Robotics and Automation*, pages 3877–3882, New Orleans, USA, April 2004.
[22] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, University of California San Diego, July 2000.
[23] Y. Yang, O. Brock, and R. Grupen. Exploiting redundancy to implement multi-objective behavior. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3385–3390, 2003.
[24] W. Zhao and M. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *IEEE Workshop on Futrure Trends in Distributed Computing Systems*, May 2003.
[25] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. ACM Mobihoc*, May 2004.