Routing Strategies for Delay-Tolerant Networks

Evan P.C. Jones
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
ejones@uwaterloo.ca

Paul A.S. Ward
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
pasward@uwaterloo.ca

ABSTRACT

Delay-Tolerant Networks (DTNs) have the potential to interconnect devices in regions that current networking technology cannot reach. The idea is that an end-to-end connection may never be present. To make communication possible, intermediate nodes take custody of the data being transferred and forward it as the opportunity arises. Both links and nodes may be inherently unreliable and disconnections may be long-lived.

To realize the DTN vision, routes must be found over multiple unreliable, intermittently-connected hops. Many researchers have investigated this fundamental challenge, particularly over the past five years. This paper surveys the area of routing in delay-tolerant networks and presents a system for classifying the proposed routing strategies.

1. INTRODUCTION

Wired and wireless networks have enabled a wide range of devices to be interconnected over vast distances. For example, today it is possible to connect from a cell phone to millions of powerful servers around the world. As successful as these networks have been, they still cannot reach everywhere, and for some applications their cost is prohibitive. The reason for these limitations is that current networking technology relies on a set of fundamental assumptions that are not true in all environments. The first and most important assumption is that an end-to-end connection exists from the source to the destination, possibly via multiple intermediaries. This assumption can be easily violated due to mobility, power saving, or unreliable networks. For example, if a wireless device is out of range of the network (*e.g.* the nearest cell tower, 802.11 base station, *etc.*), it cannot use any application that requires network communication. Delay-tolerant networking (DTN) is an attempt to extend the reach of networks. It promises to enable communication between "challenged" networks, which includes deep space networks, sensor networks, mobile ad-hoc networks, and low-cost networks. The core idea is that these networks can be connected if protocols are designed to accommodate disconnection [1].

As an example of where these networks are useful, consider a classroom where each student has a laptop, but there is no network infrastructure. One would like the students to collaborate on projects using the wireless network cards in the laptops, and also to communicate with the Internet. Delay-tolerant networking can make this happen, as illustrated in Figure 1. The laptops communicate with each other to exchange data. If the destination laptop is not present, which may occur if the student has gone home, the network stores the messages until they return. To communicate with the Internet, the school could be serviced via a router attached to a bus travelling between the school and an Internet gateway. This device picks up requests from the school and delivers them to the gateway, and then provides the responses on its next trip. First Mile Solutions sells a system called DakNet that is based on this idea [2], while the Wizzy Digital Courier Project uses a simple one-hop delay-tolerant network to provide Internet access to rural South African schools [3].

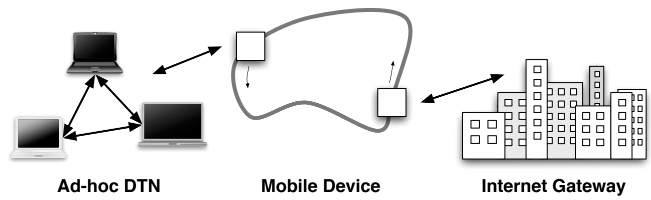

Figure 1: Laptops communicating with each other and the Internet via delay-tolerant networking

There are many other applications for delay-tolerant networks. In developing regions, applications range from education to health care to government services [4]. In developed nations, researchers have proposed augmenting low bandwidth Internet connections with a high-bandwidth delay-tolerant network built by sending physical media, such as DVDs, through the postal system [5]. This allows very large files to be quickly and cheaply exchanged while small files and control messages are exchanged over a low bandwidth link. Others have investigated using DTNs to provide Internet access to cars, by connecting temporarily to roadside wireless base stations [6]. DTNs could also be used to gather data from everything ranging from sensors in oceans [7], to satellites in space [8].

One of the fundamental problems that arises when designing networks that handle disconnection is routing. Before a network can be usable, it must be possible to get data from the source to the destination. This paper reviews the existing literature that attempts to solve the routing problem in DTNs. The research dates back to before the term "delay-tolerant" was widely used. The adjectives "intermittently-connected," "sparse," and "disconnected" are also used to describe networks without constant end-to-end connections. This paper presents two properties that can be used to classify delay-tolerant routing strategies: replication and knowledge. Replication describes how a routing strategy relies on multiple copies of each message, and knowledge describes how information about the network is used to make decisions.

Before discussing the details of DTN routing, we first discuss the important properties of DTNs and the metrics used to evaluate each technique. Next, we present a system for classifying each strategy based on the two properties. We divide the routing strategies into two broad families, flooding and forwarding, based on their use of replication and knowledge. We analyze each family, and show how prior research fits into the classification scheme. Finally, we present a summary of the current work in routing for delay-tolerant networks, and propose directions for future research.

2. NETWORK CHARACTERISTICS

In order to discuss the routing problem, we need a model that describes the network. A DTN is composed of computing systems participating in the network, called nodes. One-way links connect some nodes together. These links may go up and down over time, due to mobility, failures, or other events. When the link is up, the source node has an opportunity to send data to the other end. In the DTN literature this opportunity is called a contact [1]. More than one contact may be available between a given pair of nodes. For example, a node might have both a high-performance, expensive connection and a low-performance, cheap connection that are available simultaneously for communication with the same destination. The contact schedule is the set of times when the contact will be available. In graph theory, this model is a time-varying multigraph. The DTN architecture proposes to use this network by forwarding complete messages over each hop. These messages will be buffered at each intermediate node, potentially on non-volatile storage. This enables messages to wait until the next hop is available, which may be a long period of time.

As described by Jain *et al.*, the amount of time for a message to be transferred from one node to another can be divided into four components: waiting time, queuing time, transmission delay, and propagation delay [9]. The waiting time is the amount of time a message must wait between when it arrives at a node and when the contact to the next node becomes available. This depends on the contact schedule and the message arrival time. The queuing time is the time it takes to drain the queue of higher priority messages. This depends on the contact data rate and the competing traffic in the network. The transmission delay is the time it takes for all the bits of the message to be transmitted, which can be computed from the contact's data rate and the message length. The propagation delay is the time it takes a bit to propagate across the connection, which depends on the link technology.

2.1 Challenges

Delay-tolerant networks present many challenges that are not present in traditional networks. Many stem from the need to deal with disconnections, which directly impacts routing and forwarding. However, because these networks enable communication between a wide range of devices, there are secondary problems that routing strategies may need to be aware of, such as dealing with limited resources.

2.1.1 Contact Schedules

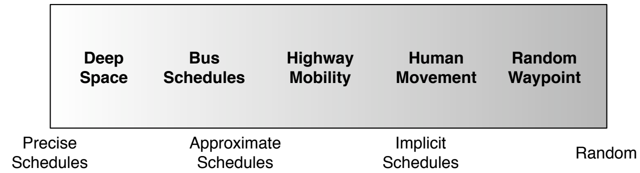

Figure 2: A spectrum of contact schedule predictability

Of the four inter-node delay components, the most significant is likely to be the waiting time, since it might range anywhere from seconds to days whereas the others are typically much shorter. Thus, one of the most important characteristics of a DTN is the contact schedule, which depends strongly on the application area under consideration. Contact schedules can be placed on an approximate spectrum based on how predictable they are, as shown in Figure 2. At one extreme we have contact schedules that are very precise. An example would be deep space networks, where disconnections are caused by movements of objects in space that can be calculated very accurately. One step less predictable would be scheduled networks with errors. For example, consider a DTN where the nodes are mounted on city buses. These buses have a schedule, but it is not precise. Due to traffic, equipment failures, or accidents, the actual arrival times can vary significantly, as anyone who regularly takes public transit can attest. Many human activities have implicit schedules, such as work. There is no guarantee when a person will be at work, but their schedule is fairly regular. Finally, at the other end of the spectrum are networks with completely random connectivity. These networks are widely studied in the ad-hoc networking community because the models are simple to work with. This spectrum is similar to the one presented in [10].

Some work on DTN routing has investigated networks with proactive mobility, where the nodes actively move in response to communication needs [10, 11, 12, 13]. In our model, this can be represented as contacts that can be selectively brought up or down. Networks with this type of mobility fall somewhere in the middle of the spectrum because some contacts are unpredictable, but the controlled contacts are predictable. The routing techniques discussed in this paper are equally applicable to networks with proactive mobility. However, these networks also require some type of cost/benefit optimization to make decisions about proactive movement, which is outside the scope of this survey.

2.1.2 Contact Capacity

A question that is closely related to the contact schedule is how much data that can be exchanged between two nodes. This depends on both the link technology and the duration of the contact. Even if the duration is precisely known, it may not be possible to predict the capacity due to fluctuations in the data rate. At a first glance, it might appear that this is a simple issue for routing strategies to deal with. A naive approach would be to ignore the contact capacity, except in cases where the message is simply too large to be sent across the contact without fragmentation. If the volume of traffic is very small compared to the capacity of contacts in the network, then this is a reasonable approach. However, if the volume of traffic increases due to a large number of users, or due to large messages being exchanged, the contact capacity becomes very important. In this situation, the best contact could become one that is "inefficient" according to other criteria, but has the largest contact volume and thus is best equipped to handle large traffic demands.

Although there are studies of real world contact duration and capacity [14, 15], few of the routing strategies surveyed attempt to use this information. One exception is the EDLQ and EDAQ schemes proposed by Jain *et al.*, which compute the delay caused by waiting for competing traffic, then route messages on the paths with the smallest delay [9].

2.1.3 Buffer Space

In order to cope with long disconnections, messages must be buffered for long periods of time. This means that intermediate routers require enough buffer space to store all the messages that are waiting for future communication opportunities. From one point of view, this means that intermediate routers require buffer space proportional to demand. An alternate point of view is that routing strategies might need to consider the available buffer space when making decisions. In the studies surveyed here, all nodes have an equal amount of buffer space and the strategies do not make decisions based on this resource.

2.1.4 Processing Power

One of the goals of delay-tolerant networking is to connect devices that are not served by traditional networks. These devices may be very small, and similarly have small processing capability, in terms of CPU and memory. These nodes will not be capable of running complex routing protocols. The strategies presented in this paper are not designed for extremely small sensors. However, research in routing for wireless sensor networks has extensively investigated this issue [16]. The routing strategies presented here could still be used on more powerful gateway nodes, in order to connect the sensor network to a general purpose delay-tolerant network.

2.1.5 Energy

Some nodes in delay-tolerant networks may have limited energy supplies either because they are mobile, or because they are in a location that cannot easily be connected to the power grid. Routing consumes energy by sending, receiving and storing messages, and by performing computation. Hence, routing strategies that send fewer bytes and perform less computation will be more energy efficient. Additionally, routing strategies can optimize power consumption by using energy-limited nodes sparingly. While researchers have investigated general techniques for saving power in delay-tolerant networks [17], none of the routing strategies surveyed has incorporated power-aware optimizations. Thus, we will not discuss this topic further.

2.2 Evaluation Criteria

In order to compare routing strategies, we must define some metrics for evaluating their performance. Since the exact numbers for the metrics depend on many factors, we will only discuss them in relative terms.

2.2.1 Delivery Ratio

In a delay-tolerant network, the most important network performance metric is the delivery ratio. However, in DTNs, a message is rarely actually "lost." Rather, the network was unable to deliver messages within an acceptable amount of time. Thus, we define the delivery ratio as the fraction of generated messages that are correctly delivered to the final destination within a given time period.

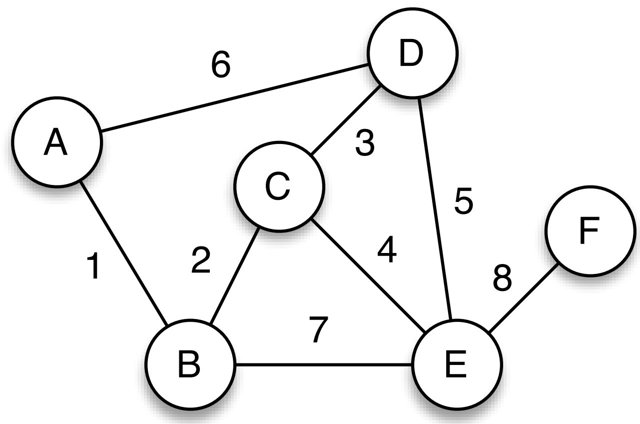

Figure 3: Example DTN scenario

2.2.2 Latency

A secondary metric is the latency, the time between when a message is generated and when it is received. This metric is important since many applications can benefit from a short delivery latency, even though they will tolerate long waits. Many applications also have some time window where the data is useful. For example, if a DTN is used to deliver e-mail to a mobile user, the messages must be delivered before the user moves out of the network.

2.2.3 Transmissions

Some routing strategies transmit more messages than others, either because they use multiple copies of each message, make different decisions about the next hop, or because of protocol overhead. The number of transmissions is a measure of the amount of contact capacity consumed by a protocol. It is also an approximate measure of the computational resources required, as there is some processing required for each message. Additionally, each transmission consumes energy, so it is also an approximate measure of power consumption.

2.3 Scenario

We will use the scenario shown in Figure 3 throughout this paper to illustrate the routifive nong strategies. There are six nodes labelled A through F. The contacts go up and down one at a time, in the sequence shown by the labels on each line. All links are bidirectional, and messages are always generated by node A.

3. STRATEGY PROPERTIES

This survey categorizes delay-tolerant routing strategies using two properties. The first property, replication, denotes how the strategy uses multiple copies of a message, and how it chooses to make those copies. The second property, knowledge, indicates how the strategy uses information about the state of the network in order to make routing decisions, and also how it obtains that information.

3.1 Replication

Delay-tolerant networks may rely on components that are unreliable or unpredictable. To compensate for this, many routing strategies make multiple copies of each message, in order to increase the chance that at least one copy will be delivered, or to reduce delivery latency. The intuition is that having more copies of the message increases the probability that one of them will find its way to the destination, and decreases the average time for one to be delivered. This is a clear trade-off between cost and performance. The cheapest approach is to have a single copy of the message. However, a single failure will result in the message being lost. The most reliable approach is to have each node carry a copy of the message. In this case, the message is lost only if all the nodes in carrying it are unable to deliver it. However, this consumes bandwidth and storage resources proportional to the number of nodes in the network.

A related issue is characterizing the best approach to making replicas. Jain *et al.* present a theoretical approach to determine which set of paths to use, provided that the path failure probabilities are known and independent [18]. Erasure coding and networking coding schemes have also been investigated to attempt to keep the benefit of multiple copies while reducing the resource costs [19, 20]. These techniques appear promising, and it should be possible to integrate them with the routing strategies presented here.

3.2 Knowledge

Some routing strategies require more information about the network than others. At one extreme, a node can make decisions with zero knowledge about the network, except which contacts are currently available. These strategies use static rules that are configured when the strategy is designed, and every node obeys the same rules. This leads to simple implementations that require minimal configuration and control messages, since all the rules are hard-coded ahead of time. The disadvantage is that the strategy cannot adapt to different networks or conditions, so it may not make optimal decisions. At the other end of the spectrum, a node might need to know the complete future schedule of every contact in the network. Provided that the information is accurate, this allows routing strategies to make very efficient use of network resources by forwarding a message along the best path. There is a range of values in between these two extremes. For example, approximate information about the future contact schedules might be available. Or, a strategy might require no information in advance, but instead will learn it automatically.

4. STRATEGY FAMILIES

We divide DTN routing strategies into two families based on which property a strategy uses in order to find the destination. Like all classification schemes, there are some cases that do not fall cleanly into either group, but we attempt to select the primary technique that a strategy uses. The families are flooding strategies, which rely primarily on replicating messages to enough nodes so the destination receives it, and forwarding strategies, which rely on knowledge about the network to select the best path to the destination. We

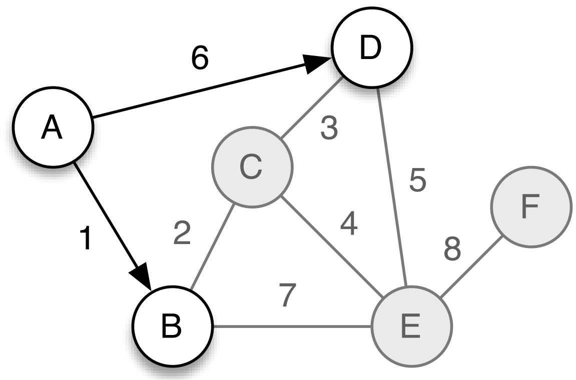

Figure 4: Direct Contact routing example

first describe each family in general, and then describe specific examples in each family.

4.1 Flooding Strategies

Strategies in the flooding family deliver multiple copies of each message to a set of nodes, called relays. The relays store the messages until they connect with the destination, at which point the message is delivered. The earliest work in the area of DTN routing fall into this family. Many of them date before the term “delay-tolerant” became popular. Traditionally, these strategies have been studied in the context of mobile ad-hoc networks, where random mobility has a good chance of bringing the source into contact with the destination. Message replication is then used to increase the probability that the message gets delivered. The basic protocols in this family do not need any information about the network, however more advanced schemes use some knowledge to improve performance.

4.1.1 Direct Contact

This strategy waits until the source comes into contact with the destination before forwarding the data. This is the degenerate case of the flooding family, where the set of relays contains only the destination. It can also be considered a degenerate case of the forwarding family, where it always selects the direct path between the source and the destination. However, since this strategy does not require any information about the network and only uses a single hop, we will consider it to be a flooding strategy. Due to its simplicity, it does not consume many resources, and it uses exactly one message transmission. However, it only works if the source contacts the destination. The Infostation architecture proposed using direct contact delivery between mobile nodes and fixed gateways as a technique for increasing wireless network throughput and decreasing cost [21]. Grossglauser and Tse showed that in their mobile ad-hoc network scenario, this strategy has a capacity that approaches zero as the number of nodes increases [22].

In the example scenario, node A can only deliver messages to nodes B and D, as shown in Figure 4. Additionally, it is faster for node A to deliver a message to node D via the path A-B-C-D, which it cannot do.

4.1.2 Two-Hop Relay

In this strategy, the source copies the message to the first n nodes that it contacts. The source and the relays hold the

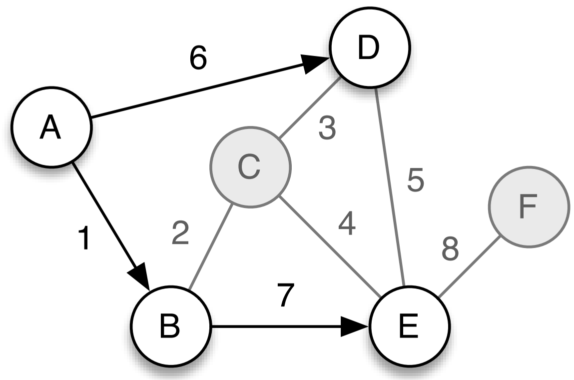

Figure 5: Two-hop routing example

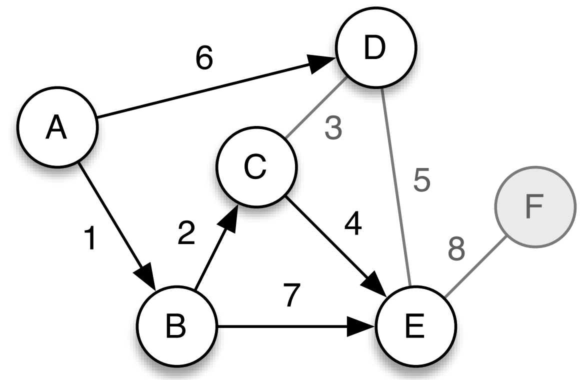

Figure 6: Tree-Based Flooding Example

message and deliver it to the destination. Since there are now $n+1$ copies of the message in the network, more bandwidth and storage are consumed. However, the resource consumption is limited and can be tuned by adjusting the number of copies. This strategy has a much better chance of delivering the message than the Direct Contact strategy. If we assume that each node contacts the destination with an independent probability p, then this strategy will deliver each message with probability $1-(1-p)^{(n+1)}$, which is approximately $(n+1)p$ if p is very small. Similarly, increasing the number of copies decreases the average latency, since the message is delivered as soon as any of the $n+1$ nodes contacts the destination. This strategy has the same fundamental limitation as Direct Contact: If the $n+1$ nodes never reach the destination, the message cannot be delivered. In scenarios where the mobility is random, this might be rare, but in networks with structured connectivity this could be very common.

In the example, if node A has a message for node E, it would send copies to both nodes B and D, as shown in Figure 5. When node B connects with node C, it would not send it the message, since node C is not the destination. Finally, the message would be delivered at time 7 when node B connects with node E. At the end, nodes A, B, D and E have all received the message. Node A can reach all other nodes via two-hop relay except node F, since it is a minimum of three hops away.

Grossglauser and Tse showed that this strategy can be used to increase the capacity of mobile ad-hoc networks under ideal conditions [22]. This approach has also been studied as a routing strategy for sensor networks [23], and for scenarios with proactive mobility [12]. It has been proposed as a fallback when ad-hoc routing cannot find a connected path [24].

4.1.3 Tree-Based Flooding

Tree-Based Flooding strategies extend two-hop relay by distributing the task of making copies to other nodes. When a message is copied to a relay, there an indication of how many copies the relay should make. This is called Tree-Based Flooding because the set of relays forms a tree of nodes rooted at the source. Two-hop relay can be viewed as Tree-Based Flooding with a depth of one.

There are many ways to decide how to make copies. A simple scheme is to allow each node to make unlimited copies, but to restrict the message to travel a maximum of n hops from the source [25]. This limits the depth of the tree, but places no limit on its breadth. A refinement is to also limit the node to make at most m copies [26]. This limits both the depth and the breadth of the tree, which limits the total number of copies to a maximum of $\sum_{i=0}^{n} m^n$. A more complex alternative is to limit the total number of copies to N. When a node makes a copy, it distributes the responsibility for making half of its current copies to the other node, and keeps half for itself [26, 27]. This scheme has been shown to be optimal if the inter-node contact probabilities are independent and identically distributed [27].

Tree-Based Flooding can deliver messages to destinations that are multiple hops away, unlike Direct Contact or Two-Hop Relay. However, tuning the parameters can be a challenge. If they are too conservative, many extra copies will be made. Conversely, if they are too aggressive, then the message may not propagate to the destination. Consider the example scenario if node A has a message for node E, and it can make a maximum of four additional copies. At time 1, it sends a copy to node B along with directions to make one copy ($\lfloor(4-1)/2\rfloor$), shown in Figure 6. Node A keeps two copies for itself ($\lceil(4-1)/2\rceil$). At time 2 when node B connects with node C, B's additional copy is delivered. At time 3, node C connects to node D. However, it cannot send D a copy because it has no copies to distribute. At time 4, C delivers the message to the E. At time 6, A sends D a copy, since it does not know that the message was already delivered. At this point, node A has one remaining copy that it will deliver if it contacts another node.

4.1.4 Epidemic Routing

Epidemic algorithms were originally proposed for synchronizing replicated databases [28]. Vahdat and Becker applied these algorithms to forwarding data in a DTN [25]. In effect, the queue of messages waiting to be delivered is the database that needs to be synchronized. Epidemic algorithms guarantee that provided a sufficient number of random exchanges of data, all nodes will eventually receive all messages. Thus, the destination node is guaranteed to have received the data. Epidemic Routing works as follows. When a message is sent, it is placed in the local buffer and tagged with a unique ID. When two nodes connect, they send each other the list of all the messages IDs they have in their buffers, called the summary vector. Using the summary vector, the nodes exchange

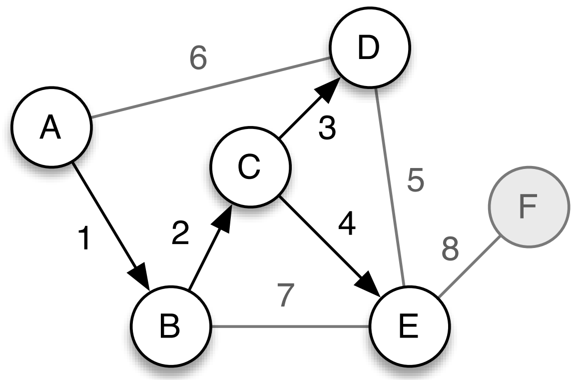

Figure 7: Epidemic Routing Example

the messages they do not have. When this operation completes, the nodes have the same messages in their buffers.

Epidemic Routing represents the extreme end of the flooding family because it tries to send each message over all paths in the network. This provides a large amount of redundancy since all nodes receive every message, making this strategy extremely robust to node and network failures. Additionally, since it tries every path, it delivers each message in the minimum amount of time if there are sufficient resources. In the example, the message will be delivered from A to E via the fastest path (A-B-C-E), as shown in Figure 7. All nodes will receive the message except node F because node E does not replicate messages that are destined for itself.

Epidemic Routing is relatively simple because it requires no knowledge about the network. For that reason, it has been proposed to use it as a fallback when no better method is available [29]. The disadvantage is that a huge amount of resources are consumed due to the large number of copies. This requires large amount of buffer space, bandwidth, and power.

Many papers have studied ways to make Epidemic Routing consume fewer resources [10, 14, 26, 30, 31, 32, 33, 34]. One of the problems is that the message continues to propagate through the network, even after it has been delivered. The original epidemic algorithms paper proposed "death certificates" to solve this problem [28]. The idea is that a new message is propagated informing nodes to delete the original message and to not request it again. Ideally, the death certificate will be much smaller than the original message, so overall the resource consumption is reduced. Researchers have explored various schemes for tuning how aggressively the death certificates are propagated. Small and Haas show that the more aggressive the death certificate propagation, the less storage is required at each node [26], while Harras and Almeroth show that the more aggressive strategies transmit more messages [34].

A critical resource in epidemic routing is the buffer. An intelligent buffer management scheme can improve the delivery ratio over the simple FIFO scheme [30]. The best buffer policy evaluated is to drop packets that are the least likely to be delivered based on previous history. If node A has met B frequently, and B has met C frequently, then A is likely to deliver messages to C through B. Similar metrics are used in a number of epidemic protocol variants [10, 14, 30, 31, 32]. This approach takes advantage of physical locality and the fact that movement is not completely random. While these protocols are more efficient than the original Epidemic routing protocol, they still transmit many copies of each message.

4.2 Forwarding Strategies

The strategies in this family take a more traditional approach to routing data in a DTN. They use network topology information to select the best path, and the message is then forwarded from node to node along this path. A path can be found using location-based routing, assigning metrics to nodes or by assigning metrics to links. Some of these approaches have been explored in wired and multi-hop wireless networks. However, the protocols designed for these environments will not function in delay-tolerant networks, since they assume that links are usually connected. By definition, the strategies in this family require some knowledge about the network. They typically send a single message along the best path, so they do not use replication.

4.2.1 Location-Based Routing

The forwarding approach that requires the least information about the network is to assign coordinates to each node. A distance function is used to estimate the cost of delivering messages from one node to another. The coordinates can have physical meaning, such as GPS coordinates, as has been studied for mobile ad-hoc networks [35]. Alternatively, the coordinates can have meaning in the network topology space, instead of physical space, which has been used to estimate network latency between arbitrary nodes on the Internet [36]. In general, a message is forwarded to a potential next hop if that node is closer in the coordinate space than the current custodian.

The advantage of location-based routing is that it requires very little information about the network, eliminating the need for routing tables and reducing the control overhead. In order to determine the best path, a node only needs to know its own coordinates, the coordinates of destination, and the coordinates of the potential next hops. Given these three pieces of information, a node can easily compute the distance function and determine where the message should be sent.

Location-based routing has two well known problems. The first problem is that even if the distance between two nodes is small, there is no guarantee that they will be able to communicate. In the case of physical coordinates, consider two wireless nodes on opposite sides of a wall that blocks all radio signals, as shown in Figure 8. Node A wants to send a message to D. It has two potential next hops: nodes B and C. Since node B is the closest to node D, it forwards the message to B. However, B cannot communicate with D due to the obstruction, whereas node C has a line of sight. The problem is that location does not necessarily correspond to network topology. This problem is somewhat alleviated by using virtual coordinates, since they are designed to closely represent the network topology. However, it is still possible that the message can fall into a local minimum and not reach the destination. In mobile ad-hoc networks, protocols have been explored which attempt to route around obstructions like this [37]. The second problem is that a node's coordinates can change. If a node moves, its physical coordinates change. If the network topology changes, a

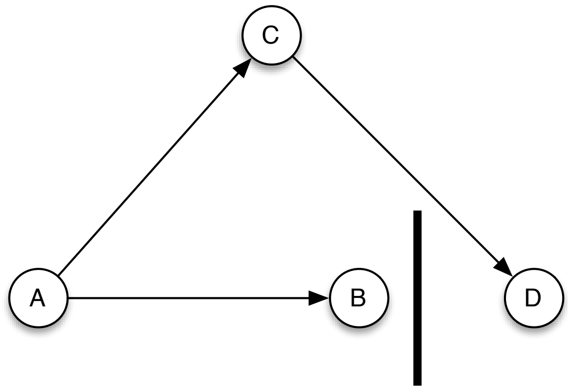

Figure 8: Location-based routing fails because of a local minimum

node's virtual coordinates change. This complicates routing because the source needs the coordinates of the destination node. These two problems mean that implementating location-based routing is not as simple as it appears.

Lebrun *et al.* proposed using the motion vector of mobile nodes to predict their future location. Their scheme passes messages to nodes that are moving closer to the destination [38], which results in a better delivery ratio than two-hop routing with less overhead than Epidemic routing. Leguay *et al.* presented a virtual coordinate routing strategy called mobility pattern spaces [39]. In their strategy, the node coordinates are composed of a set of probabilities, each representing the chance that a node will be found in a specific location. Various distance functions are then computed on this vector. Their results show that this approach reduces the amount of resources consumed when compared with Epidemic Routing, while still delivering a substantial fraction of the messages. Neither of these works address the local minima or changing coordinates problems. However, they do show that these techniques are applicable to DTNs.

4.2.2 Gradient Routing

An alternate approach is to assign a weight to each node that represents its suitability to deliver messages to a given destination. When the custodian of a message contacts another node that has a better metric for the message's destination, it passes the message to it. This approach is called gradient routing because the message follows a gradient of improving utility function values towards the destination. The idea was first applied to ad-hoc networks in 2000 [40]. This requires more network knowledge than location-based routing for two reasons. First, each node must store a metric for all potential destinations. Second, sufficient information must be propagated through the network to allow each node to compute its metric for all destinations. The metric could be based on many parameters, such as the time of last contact between the node and the destination, remaining battery energy, or mobility. An extremely simple utility function is to forward a packet with a certain probability. This approach does not seem practical, but it has been used as a baseline for comparison with more advanced techniques [41, 9].

Gradient Routing has been shown to decrease the delay when compared to direct contact [41]. Similar schemes have been proposed for routing data towards base stations in sensor networks. For example, the history-based protocol presented for ZebraNet is a gradient routing strategy [42]. A theoretical analysis of gradient routing can be found in [41], and a discussion about predicting utility function values can be found in [43].

One of the shortcomings of gradient routing is that it can initially take a long time for a good custodian to be found, since it may take some time for the utility function values to propagate, or because the metric values in the region around the initial custodian are all equally poor. One approach that has been shown to reduce the delivery latency is to initially use random forwarding until the utility value reaches a certain threshold [41]. This hybrid approach initially allows a message to actively explore the network until it finds a good carrier, and then it uses the standard utility routing to efficiently reach the destination. Burgess *et al.* use a similar technique to quickly propagate a new message in their epidemic routing variant [14].

4.2.3 Link Metrics

Routing strategies that use link metrics resemble traditional network routing protocols. They build a topology graph, assign weights to each link and finally run a shortest path algorithm to find the best paths. This requires the most network information as each node must have sufficient knowledge to run a routing algorithm. Link weights are assigned to try and provide optimal service to the endpoints, based on some performance metric: the highest bandwidth, lowest latency, and the highest delivery ratio. In delay-tolerant networks, the most important metric is the delivery ratio, since the network must be able to reliably deliver data. A secondary metric is the delivery latency. Thus, the challenge is to determine a system for assigning link metrics that maximizes the delivery ratio and minimizes the delivery latency. Some metrics may also attempt to minimize resource consumption, such as buffer space or power.

The first paper that proposed using link metrics for routing in delay-tolerant networks suggests that an appropriate metric is to minimize the end-to-end delivery latency [9]. The intuition is that this minimizes the amount of time that a message consumes buffer space, and thus it should also maximize the delivery ratio since there is more space available for other messages. Their work uses a metric that is the time it will take for a message to be sent over each link. Since this value may depend on the time a message arrives at a node, the authors present a time-varying version of Dijkstra's shortest path algorithm. Finding the path that delivers the message with the shortest delay has also been used for proactive mobile networks [11].

Jain *et al.* present a variety of different metrics for networks with precise schedules, each of which require different amounts of information. However, all of the metrics assume that the contact schedule is precise. The first metric, called Minimum Expected Delay (MED), is the metric that requires the least amount of information. It assumes that the queuing time is zero, and that the average of the sum of transmission time, propagation delay, and waiting time is known precisely. This value is the expected delay: the average amount of time it takes for a message to go from one node to another, assuming that all arrival times are equally likely. The next metric is Earliest Delivery (ED). The queuing delay is assumed to be zero, and the propagation and transmission delays are assumed to be known precisely. The path is selected that will get the message to the destination at the earliest time. This requires the complete contact schedule, whereas MED only requires the average value of the waiting time. The next metric is Earliest Delivery with Local Queuing (EDLQ), which uses the buffer occupancy at each node to add an estimate of the queuing delay to the ED metric. Finally, the Earliest Delivery with All Queues (EDAQ) uses the information about the traffic demands for all nodes in order to compute the exact queuing delay. This paper shows that the protocols with more information have higher delivery ratios and lower delay. However, for some scenarios the difference between them is small.

The techniques presented by Jain *et al.* assume that accurate information about the complete contact schedules is known in advance. This may be feasible for scenarios where the connectivity is extremely predictable and reliable. Unfortunately, in reality, schedules may be imprecise or completely unpredictable. To address this problem, Jones *et al.* presented a metric called the Minimum Estimated Expected Delay (MEED), where the weights are based purely on observed connectivity [44]. They compute a metric based on a sliding window of observed connectivity. The assumption is that the future connectivity will be similar to the past. To distribute the metrics throughout the network, they use an epidemic protocol to propagate link-state table updates. Their results show that in a scenario based on wireless LAN data [45], this technique approaches 95% of the delivery ratio of the ED metric.

An interesting issue that arises when using shortest path routing with link metrics is when to make the routing decisions. The traditional choice is to make the decision about the entire path at the source (source routing), or to make the decision about the next hop when the message arrives (per-hop routing). If the link metrics do not change while the message is in transit, the paths selected by these options will be the same. This is true for networks where the end-to-end delays are very small, or in networks where the contact schedules are precisely known. However, if the metrics are approximate and the messages take a long time to traverse the network, as is the case in delay-tolerant networks, the choice of when to make routing decisions may have a significant impact. Jones *et al.* argue that the best choice is to make decisions as late as possible, since that will allow messages to be forwarded using the most recent information [44]. To do this, they present what they call per-contact routing, where each node recomputes its routing table each time a contact becomes available, and then evaluates all the messages in its buffer to determine if they should be forwarded over the available links. This is computationally expensive, but always makes decisions with the most recent information. Their results show that these optimizations reduce the delivery latency. Handorean *et al.* also propose a very similar scheme, which they call a path update [29].

5. OPEN ISSUES

If we plot the approximate location of each of the routing strategies discussed here using the Replication and Knowledge properties, as shown in Figure 9, we can see that the edges along the two axes are very well explored, but the middle space is not. The middle represents strategies that take advantage of both Replication and Knowledge in order to improve the delivery ratio and decrease the delivery latency. In particular, the area represented by the lower left-hand corner, that uses a small amount of both Replication and Knowledge, is likely to yield routing strategies that can be applied in the real world, as they will not require precise schedules or substantial amounts of configuration, nor will they consume large amount of resources by flooding the network with duplicate messages. The variants of Epidemic Routing that take advantage of some learned topology information are a good first step in this direction.

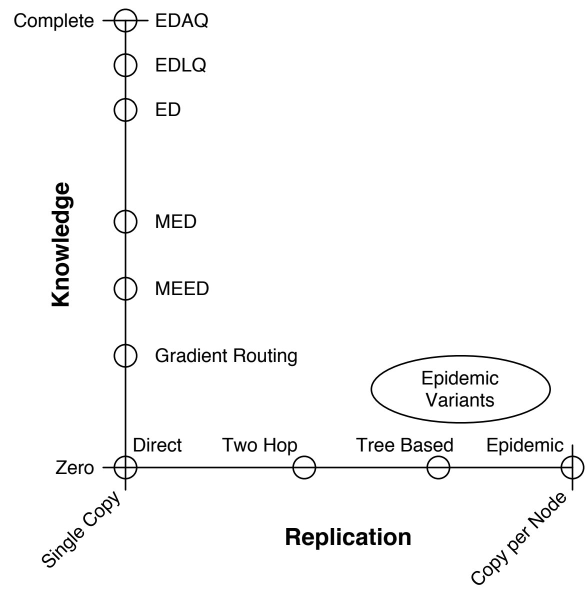

Figure 9: DTN Routing Strategy Properties

The papers surveyed here cover a large range of the mobility spectrum shown in Figure 2. Precise schedules are covered by the forwarding strategies, random networks are covered by epidemic and tree-based strategies, and the networks with implicit schedules are covered by some of the variants. However, one type of network in this spectrum has not been well examined: networks with imprecise schedules. These networks have fairly predictable contact schedules, which should be leveraged to improve performance, but it is not clear if the techniques pioneered for precise schedules can be used without modifications.

It is impossible to determine what approach is the right one when there are no real delay-tolerant networks. Perhaps the most important future work is to actually build DTNs and applications that use them, and then to see what routing problems occur in the real world. At the moment, there are a number of prototype DTNs that are being used to measure connectivity properties [14, 45, 15]. These projects are an extremely important first step, however these networks are not yet being used for any real applications. It is difficult to predict the requirements for DTN routing strategies and to evaluate their performance, without any information about what traffic patterns would be relevant. Building more experimental deployments and applications is critical for being able to determine where DTN routing strategies need improvement.

Finally, DTNs must be able to integrate multiple types of networks together. This means that techniques will be required that allow messages to be exchanged between DTNs that have different properties and possibly use different routing protocols. None of the work presented here addresses this issue. Additionally, there is an extremely important network that most DTNs will want to communicate with: The Internet. One proposal that enables communication between DTNs and the Internet is the Tetherless Communication Architecture [46]. Any widely adopted DTN routing protocol will need to address these issues.

6. CONCLUSIONS

Delay-tolerant networks are a promising new development in network research, that offer the hope of connecting people and devices that hitherto were either unable to communicate, or could do so only at great cost. In this paper we have surveyed existing techniques for routing in such networks. While we discovered a wide variety of methods that address the routing problem, we were able to classify them according to two key properties: replication and knowledge. Our survey and classification enabled us to make the following observations.

First, to achieve a high delivery ratio with low resource consumption, hybrid techniques that rely on both knowledge about the topology and replication will be required. This has been implicitly noted by several of the researchers in the field, though the challenge is to determine the correct balance between redundancy and resource consumption, and to find manageable solutions for using network topology information.

Second, in cases where message volume is low, simple epidemic routing works extremely well. This suggests that small experimental deployments could be rapidly developed based on epidemic routing, allowing researchers to have real network topology and traffic data, which could be used to design new routing strategies.

7. REFERENCES

[1] K. Fall, "A delay–tolerant network architecture for challenged internets," in *Proceedings of ACM SIGCOMM*, pp. 27–34, August 2003.

[2] A. A. Hasson, D. R. Fletcher, and D. A. Pentland, "A road to universal broadband connectivity," in *Proceedings of Development by Design (dyd02)*, December 2002.

[3] A. Rabagliati, "Wizzy digital courier – how it works," April 2004. `http://www.wizzy.org.za/article/articleprint/19/1/2/`.

[4] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall, "The case for technology in developing regions," *IEEE Computer*, vol. 38, pp. 25–38, May 2005.

[5] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy, "Turning the postal system into a generic digital communication mechanism," in *Proceedings of ACM SIGCOMM*, pp. 159–166, 2004.

[6] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.11b for "automobile" users," in *Proceedings of IEEE INFOCOM*, pp. 362–373, March 2004.

[7] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proceedings of ACM MobiHoc*, pp. 233–244, June 2003.

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

[9] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of ACM SIGCOMM*, vol. 34, pp. 145–158, ACM Press, October 2004.

[10] D. Marasigan and P. Rommel, "MV routing and capacity building in disruption tolerant networks," in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 398–408, March 2005.

[11] Q. Li and D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, pp. 44–55, August 2000.

[12] 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*, (New York, NY, USA), pp. 187–198, ACM Press, May 2004.

[13] 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*, vol. 2, pp. 1407–1418 vol. 2, April 2005.

[14] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networking," in *Proceedings of IEEE INFOCOM*, April 2006.

[15] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 244–251, August 2005.

[16] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[17] H. Jun, M. H. Ammar, and E. W. Zegura, "Power management in delay tolerant networks: A framework and knowledge-based mechanisms," in *Proceedings of IEEE SECON*, September 2005.

[18] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proceedings of ACM SIGCOMM*, pp. 109–120, October 2005.

[19] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 229–236, August 2005.

[20] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 284–291, August 2005.

[21] R. H. Frenkiel, B. R. Badrinath, J. Borres, and R. D. Yates, "The infostations challenge: balancing cost and ubiquity in delivering wireless data," *IEEE Personal Communications*, vol. 7, no. 2, pp. 66–71, 2000.

[22] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM*

Transactions on Networking, vol. 10, pp. 477–486, August 2002.

[23] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data mules: modeling a three-tier architecture for sparse sensor networks,” in *Proceedings of Sensor Network Protocols and Applications*, pp. 30–41, May 2003.

[24] D. Nain, N. Petigara, and H. Balakrishnan, “Integrated routing and storage for messaging applications in mobile ad hoc networks,” *Mobile Networks and Applications (MONET)*, vol. 9, pp. 595–604, December 2004.

[25] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” Tech. Rep. CS-2000-06, Duke University, July 2000.

[26] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN’05)*, pp. 260–267, August 2005.

[27] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN’05)*, pp. 252–259, August 2005.

[28] A. Demers, D. Greene, C. Houser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, “Epidemic algorithms for replicated database maintenance,” *SIGOPS Operating Systems Review*, vol. 22, pp. 8–32, January 1988.

[29] R. Handorean, C. Gill, and G.-C. Roman, “Accommodating transient connectivity in ad hoc and mobile settings,” *Lecture Notes in Computer Science*, vol. 3001, pp. 305–322, January 2004.

[30] J. A. Davis, A. H. Fagg, and B. N. Levine, “Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks,” in *Proceedings of Int’l Symp. on Wearable Computers (ISWC’01)*, pp. 141–148, October 2001.

[31] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *Lecture Notes in Computer Science*, vol. 3126, pp. 239–254, January 2004.

[32] K. Tan, Q. Zhang, and W. Zhu, “Shortest path routing in partially connected ad hoc networks,” in *Proceedings of Global Telecommunications Conference (GLOBECOM)*, vol. 2, pp. 1038–1042, December 2003.

[33] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, “Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks,” in *Proceedings of IFIP-TC6 Networking*, vol. 3462, pp. 1180–1192, May 2005.

[34] K. A. Harras and K. C. Almeroth, “Transport layer issues in delay tolerant mobile networks,” in *Proceedings of IFIP-TC6 Networking*, May 2006.

[35] M. Mauve, A. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[36] T. S. E. Ng and H. Zhang, “Predicting internet network distance with coordinates-based approaches,” in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 170–179 vol.1, June 2002.

[37] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Proceedings of ACM MobiCom*, pp. 243–254, August 2000.

[38] J. Lebrun, C.-N. Chuah, D. Ghosal, and M. Zhang, “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks,” in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, vol. 4, pp. 2289–2293, May 2005.

[39] J. Leguay, T. Friedman, and V. Conan, “Evaluating mobility pattern space routing for DTNs,” in *Proceedings of IEEE INFOCOM*, April 2006.

[40] R. D. Poor, “Gradient routing in ad hoc networks.” MIT Media Laboratory, *unpublished manuscript*, `http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf`, 2000.

[41] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Single-copy routing in intermittently connected mobile networks,” in *Proceedings of Sensor and Ad Hoc Communications and Networks (SECON)*, pp. 235–244, October 2004.

[42] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet,” in *Proceedings of ASPLOS-X*, pp. 96–107, October 2002.

[43] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive routing for intermittently connected mobile ad hoc networks,” in *Proceedings of IEEE WoWMoM*, pp. 183–189, June 2005.

[44] E. P. C. Jones, L. Li, and P. A. S. Ward, “Practical routing in delay-tolerant networks,” in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN’05)*, pp. 237–243, August 2005.

[45] Dartmouth College, *The Dartmouth Wireless Trace Archive*. `http://crawdad.cs.dartmouth.edu/`.

[46] A. Seth, P. Darragh, S. Liang, and S. Keshav, “An architecture for tetherless communication.” University of Waterloo, *unpublished manuscript*, `http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/05/tca.pdf`, June 2005.