ENCOUNTER-BASED ROUTING IN DISASTER RECOVERY NETWORKS

BY

SAMUEL C NELSON V

B.S., Bucknell University, 2006

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Adviser:

Robin Kravets

Abstract

The proliferation of wireless networking technology in a wide variety of devices allows for networks to exist in challenged environments where connectivity is intermittent. Networks capable of delivering data under such conditions are called delay or disruption tolerant networks (DTNs). A promenent example of a DTN is a disaster recovery network, in which emergency responders must be able to communicate even in complete infrastructure failure. Current work in routing protocols for these networks leverages epidemic-style algorithms that trade off injecting many copies of messages into the network for increased probability of message delivery. However, such techniques can cause a large amount of contention in the network, increase overall delays, and drain each mobile node's limited battery supply. We present a new DTN routing algorithm, called Encounter-Based Routing (EBR), which maximizes delivery ratios while minimizing overhead and delay. Furthermore, we present a means of securing EBR against black hole denial-of-service attacks. To properly evaluate EBR, we develop an event-driven, role-based mobility model for disaster recovery networks that highlights several high-level characteristics of this environment. We then evaluate EBR against many of the best current DTN protocols showing substantial improvements in common metrics as well as three composite metrics that more effectively describe the quality of DTN routing algorithms.

To my wonderful wife Rachel.

Acknowledgments

This thesis would not have been possible without the support of many people. I thank my family, particularly my wife Rachel, for their encouragement. I also extend a special thanks to Albert Harris, Mehedi Bakht, Riccardo Crepaldi, and our gracious adviser Robin Kravets, for their help and guidance on this project. Finally, I thank God for His numerous blessings on my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless networks have permeated society, enabling communication in many new environments. However, current solutions require connected networks. Even solutions aimed to support node mobility can not support communication if no end-to-end path exists at the time of the requested communication. In response, the research community has proposed new protocols that enable communication in such networks. These delay or disruption tolerant networks (DTNs) transport application data by creating a "store and forward" network where no infrastructure exists (*e.g.*, bus networks [8], wildlife tracking [21], Pocket Switched Networks [10]) or the infrastructure has been destroyed (*e.g.*, disaster recovery networks [24]). Although end-to-end connectivity may not be available between two nodes that wish to communicate, DTN routing protocols take advantage of temporal paths created in the network as nodes encounter their neighbors and exchange messages they have been asked to forward.

The goal of DTN communications is to deliver data in an effective and efficient manner for the applications using the network. Since DTNs are characterized by connectivity challenges, routing this data is often the most difficult challenge. Standard ad hoc routing techniques, such as AODV [26] and DSR [20] fail because they attempt to find a complete end-to-end path before transmitting any data. Instead, DTN routing protocols utilize a store-and-forward technique that tries to progress messages closer to the destination over time.

DTNs enable communication in a large class of environments. One environment where DTNs can be effective is a community-based bus network, where buses act as transport devices and can exchange data only when in close contact. Bus networks are characterized by a highly partitioned environment as well as somewhat predictable movement patterns, based on predetermined bus routes. Another environment where DTNs are useful is *disaster recovery networks*. This type of DTN is formed when communication infrastructure is suddenly unavailable, and communication must rely on wireless ad hoc networks that form mainly as a result of people and vehicles moving. The goal of these networks is to support communication for organizations like police and fire departments, as well as other first responders and civilians in the area. Given the characteristics of this environment, DTNs are well suited to handle communication. The challenge in creating a routing protocol for disaster recovery networks, and DTNs in general, is how to utilize specific characteristics of disaster networks, including mobility patterns exhibited by different roles, to make intelligent forwarding decisions.

The research presented in this thesis focuses on enabling communication in challenged environments. We first present our routing protocol, Encounter-based Routing (EBR), for DTNs that exploits the heterogeneity expected in DTNs, especially disaster recovery networks. Although the environments where DTNs are expected to be most effective are quite complex, little has been done to model these environments. Therefore, we next present our event and role based mobility model that captures the high-level mobility characteristics of disaster environments. Finally, we evaluate EBR using this new mobility model as well as other popular models. By exploiting a very simple piece of knowledge about mobility characteristics, EBR outperforms other DTN protocols in many environments.

1.1 DTN Routing Challenges

The challenge to providing effective communication in DTNs lies in finding solutions that provide a good delivery ratio despite the disconnected nature of these networks. Since there are no guarantees that a route will ever be available between two nodes, many current DTN routing protocols apply epidemic-style techniques [35], leveraging the fact that an increased number of copies of a particular message in the network should improve the probability that the message will reach its intended destination. However, such techniques come at a high price in terms of network resources, which can have adverse effects. First, flooding messages into the network can rapidly deplete buffer space on resource-limited mobile devices (*e.g.*, mobile phones). If the network or offered load is large enough, even nodes with relatively large disk space can suffer from space constraints. While storage limitations can limit successful delivery in a network, they can also be detrimental to applications wishing to store local data. Second, flooding messages into the network can rapidly deplete the available bandwidth, which in turn leads to heavy network congestion. Furthermore, a large amount of message passing has the potential to further increase end-to-end delay [8]. Finally, protocols that flood the network can quickly deplete the energy of the mobile devices. The goal for DTNs should be to provide high data delivery in a timely fashion, while not overwhelming the resources in the network.

A number of routing protocols have been proposed to enable data delivery in such challenging environments [3, 8, 14, 23, 27, 33, 34, 36, 38, 13, 12]. However, many of these protocols trade overhead and computational complexity for increased successful delivery. This overhead expresses itself as more traffic in the network creating more contention in areas of high node density and increased energy consumption for nodes exchanging messages. Furthermore, many DTN protocols make routing and forwarding decisions based on advertised, local contact information, allowing for denial-of-service attacks over the already intermittently connected network. All of these effects can decrease overall network performance.

One method to mitigate this overhead is to identify key properties in the network that allow for more intelligent forwarding and message replication decisions. The main contribution of our research capitalizes on these network properties to design a DTN routing protocol that uses local observations about a node's environment. Our protocol, Encounter-Based Routing (EBR), uses an encounter-based metric for optimization of message passing that maximizes message delivery ratio while minimizing overhead both in terms of extra traffic injected into the network and control overhead, as well as minimizing latency as a second order metric. The intuition behind EBR is that, in environments targeted by DTNs, such as disaster scenarios and certain vehicular networks, different classes of nodes naturally tend to have more node encounters than others. In addition to the routing component, we present a security extension to our protocol, found at the end of this thesis, that protects against denial-of-service attacks aimed at eliminating copies of messages in the system.

To fully evaluate EBR, we propose the use of three composite metrics, which clearly illustrate the interplay between fundamental metrics like message delivery ratio, goodput, and end-to-end delay. We then use these metrics to evaluate EBR and compare it to the major protocols developed for DTNs, showing improved performance and overhead. EBR achieves up to a 40% improvement in message delivery over the current state-of-the-art, as well as achieving up to a 145% increase in goodput.

1.2 Disaster Modeling Challenges

One major issue for developing DTN routing protocols is the lack of a realistic, parameterizable mobility model in which to evaluate the protocols over. To this end, we present a disaster mobility model that characterizes the high-level movement properties of different roles in a disaster scenario. Multiple parties, or organizations, make up the communicating nodes in a disaster recovery network. These organizations include police officers, ambulance, firefighters and other first responders, and civilians. The behavior of most of these organizations is driven by the need to respond to events and participate in those events based on the particular role of the organization. Since this type of behavior is very specific to emergency and disaster response scenarios, understanding communication patterns in such networks is critical to understanding how to improve the current state of emergency response. However, current mobility models for wireless networks do not capture the complexity of either the behavior of the different components of such networks or the specifics of the expected communication patterns. Realistic mobility and communication models can enable more effective evaluation via simulation and eventually lead to more effective solutions.

One of the biggest challenges faced by communication in disaster recovery networks comes from the high expectation of network partitions and dynamic object behavior. The high potential for object failure further complicates matters. Intuition says that, since objects react differently to different events, the variance in density of the communication graph will change and cause strain on current routing protocols. Due to this unique behavior, disaster recovery networks present challenging environments.

One of the fundamental aspects of simulating disaster recovery networks is realistically modeling the movement patterns of the mobile objects. Modeling mobility enables testing the effectiveness of current routing protocols as well as provides insight into how routing protocols can be improved. Disaster environments present unique challenges in that environmental events and roles directly affect a node's movement patterns. Intuitively, events act as stimuli for mobile nodes in the network, causing them to react in ways according to the predefined roles they take on. Many roles in disaster networks must react to multiple events by fleeing or approaching in a realistic fashion.

Many ad hoc network mobility models have been developed and analyzed [9, 4, 11, 19, 22, 25]. Models based on random movement are particularly popular and heavily studied [5, 6, 28, 16]. These models, while adequate to study environments for which they were designed, do not allow for groups of objects to react differently to environmental events. Therefore, a higher-level, more general mobility model is needed to incorporate the different roles objects play in disaster scenarios.

In this thesis, we present an event- & role-based mobility paradigm that effectively characterizes the movement patterns of objects in a disaster scenario. Different sets of these patterns are embedded into different object roles. By attaching actions to roles and not directly to objects, movement patterns are organized and objects can quickly shift from one pattern to another by assuming different roles. To the best of our knowledge, this is the first disaster mobility paradigm that is reactive, in a role-based fashion, to environmental events and their associated parameters.

The main contribution of this part of the thesis is the classification of a generic event- & role-based mobility paradigm that completely defines movement patterns given a series of environmental events for a set of characteristic roles. Additionally, we present a low-level physics-based gravitational mobility model that "plugs in" to our event-driven, role-based paradigm allowing objects to react to the presence of numerous disaster events based on the particular role of the node. This allows objects to flee from or approach multiple, unrelated events. To evaluate the effect of our comprehensive model on communication patterns in disaster recovery networks, we discuss a new set of relevant metrics that help characterize the changes in topology as disaster events unfold. Finally, we have developed a set of tools to realistically construct a mobility scenario and trace files of our disaster mobility model for the ns2 network simulator [1].

1.3 Map

The remainder of this thesis is organized as follows. Section 2 motivates EBR and presents a taxonomy of current DTN routing protocols. Section 3 presents encounter-based routing (EBR). Section 4 presents our event-driven, role-based mobility model for disaster recovery networks, and illustrates key property differences between our model and other mobility models. Section 5 displays an evaluation of EBR over multiple mobility models, including the disaster mobility model. Section 6 presents a security extension to EBR. Finally, Section 7 concludes.

Chapter 2

DTN Routing Protocol Taxonomy

It is challenging to route data through DTNs due to the inherent lack of end-to-end paths in the network. Although ad hoc routing protocols [20, 26] were designed to deal with the infrastructure-less nature of many wireless networks, they only operate correctly when there exists a path from the source to the destination at the time the messages need to be sent. These protocols fail in DTNs, since instantaneous end-to-end paths are not typically available. Instead of waiting for such paths to become available, current DTN routing protocols aggressively store and forward messages in hopes that the nodes will eventually encounter the messages' destinations. Such DTN routing protocols exploit the mobility of nodes [15], which leads to nodes encountering other nodes and so creating end-to-end paths that only exist in discrete segments of the time domain.

DTN routing protocols can be classified into two high-level approaches [3]: *forwarding-based* protocols and *replication-based* protocols. *Forwarding-based* protocols only keep one copy of a message in the network and attempt to forward that copy toward the destination at each encounter. In contrast, *replication-based* protocols insert multiple copies, or replicas, of a message into the network to increase the probability of message delivery. Essentially, replication-based protocols leverage a trade-off between resource usage (*e.g.*, node memory and bandwidth) and probability of message delivery. Although all replication-based protocols take advantage of this trade-off, these protocols can be further separated into two classes based on the number of replicas created: *quota-based* and *flooding-based*.

Flooding-based protocols attempt to send a replica of each message to as many nodes as possible, whereas quota-based protocols intentionally limit the number of replicas. Assume that m_t indicates the maximum number of unique messages (excluding replicas) that have been created prior to some time t. Then, an upper bound on the total number of messages (including replicas) in the network at time t is $m_t \cdot L$, where L is the maximum number of replicas for any given message. L can be a probabilistic or discrete variable. Given these definitions, a *quota-based* routing protocol can be defined as follows:

Classification	Previous Work
Forwarding	Jain *et al.* [18], DSR [20], AODV [26]
Flooding-based Replication	Epidemic, Prophet [23], MaxProp [8], RAPID [3]
Quota-based Replication	Spray and Wait [33], Spray and Focus [34]

Table 2.1: Taxonomy of DTN routing protocols

> A replication-based routing protocol is **quota-based** if and only if L is independent of the number of nodes in the network (assuming the characteristic of the network, such as storage, bandwidth, and mobility, allow for every node to have a replica of every message).

Conversely, any replication-based protocol where L is dependent on the number of nodes in the network is defined to be *flooding-based*.

These definitions allow us to classify routing protocols into three groups (see Table 2.1). Traditional Internet routing protocols (*e.g.*, IP [31]) and ad hoc routing protocols (*e.g.*, AODV [26], DSR [20]) are forwarding-based, since nodes along a route forward the message toward the destination without storing the message or creating extra replicas of the message. Forwarding-based approaches for DTNs have been proposed [17, 32], but are limited in their effectiveness due the instability or even non-existence of routes from any particular node to the destination. When a single copy of a message exists in the network, a single route break is sufficient to cause delivery to fail. One forwarding-based approach, proposed by Jain *et al.* [18], utilizes future knowledge about node mobility and specific node encounters to improve the protocol (*e.g.*, knowledge that a node will encounter a bus at noon that will have access to the Internet). However, the availability of such future knowledge constitutes a special class of DTNs and such approaches will not work in general environments. Given the unpredictability of many DTN environments, it is important avoid requiring such information for successful routing.

Epidemic routing is an obvious example of a flooding-based protocol, since the number of replicas in the network is directly dependent on the number of nodes in the network. One of the major flooding-based protocols for DTNs is MaxProp [8]. At its core, a MaxProp router prioritizes message replicas in decreasing order of estimated likelihood of delivery. Messages with higher priority are transmitted first, and messages with lower priority are dropped first. Furthermore, many useful, complimentary mechanisms, such as message acknowledgments, are incorporated. MaxProp is flooding-based, since, if resources and mobility allow, it is possible for every node in the network to have a replica of the same message. Other examples of flooding-based DTN protocols include Prophet [23], RAPID [3] and PREP [27]. Prophet attempts to use information about the likelihood of nodes encountering particular destinations to optimize the exchange of messages. RAPID orders messages through the use of utility functions, with the goal of intentionally maximizing specific metrics (*e.g.*, delay). PREP, a variant of Epidemic Routing, assigns priority to messages based on costs to destination, source and expiration time, and uses this priority to determine which messages should be deleted or transmitted when buffer or bandwidth is constrained respectively.

Recent work by Erramilli *et. al* recognizes similar problems with current DTN routing protocols and proposes techniques to utilize properties of nodes, such as contact rate, when making forwarding decisions [13, 12]. They are concerned with choosing the best node(s) to forward messages to based on utility values. This technique, however, can result in flooding-like behavior if many encountered nodes have high utility values. On the other hand, if many encountered nodes have low utility value, messages may never leave the source nodes.

The main problem with flooding-based protocols is their high demand on network resources, such as storage and bandwidth. This fact led to some work in developing quota-based protocols. Spray and Wait [33] is a quota-based protocol with a fixed upper bound on the number of replicas allowed in the network. Spray and Wait breaks routing into two phases: a *spray* phase, where message replicas are disseminated, and a *wait* phase, where nodes with single-copy messages wait until a direct encounter with the respective destinations. A follow-up protocol called Spray and Focus [34] calls for a similar spray phase, followed by a focus phase, where single copies can be forwarded to help maximize a utility function. The authors were able to show that the focus phase allowed for smaller message delays. While both Spray and Wait and Spray and Focus succeed in limiting some of the overhead of flooding-based protocols, their delivery ratios still suffer.

While quota-based protocols are much better stewards of network resources than their flooding-based counterparts, one possible criticism is their inability to successfully deliver a comparable amount of messages. In this thesis, we show that quota-based protocols are not always so handicapped by developing a quota-based protocol using an encounter-based routing metric that has extremely low routing overhead, while maintaining delivery ratios better than or comparable to current flooding-based protocols. The next section presents the design of EBR, our quota-based DTN routing protocol.

Chapter 3

Encounter-based Routing (EBR)

The primary goal of a DTN routing protocol is to obtain high message delivery ratio and good latency performance, while maintaining low overhead. However, current flooding-based protocols (*e.g.*, MaxProp [8], RAPID [3]) achieve high delivery ratios at the expense of excessive network resource usage, and current quota-based protocols ((*e.g.*, Spray And Wait [33], Spray and Focus [34]) that reduce this overhead are not able to achieve comparable delivery rates.

In response, we present Encounter-based Routing (EBR), which is a quota-based DTN routing protocol that achieves high delivery ratios comparable to flooding-based protocols, while maintaining low network overhead. This improvement in delivery ratio is accomplished by taking advantage of the following observed mobility property of certain networks: *the future rate of node encounters can be roughly predicted by past data*. This property is useful because nodes that experience a large number of encounters are more likely to successfully pass the message along to the final destination than those nodes who only infrequently encounter others. Many networks experience this phenomenon; examples include disaster recovery networks, where ambulances and police tend to be more mobile and bridge more cluster gaps than civilians, and vehicular-based networks, where certain vehicles take popular routes.

Since EBR is a quota-based routing protocol, it limits the number of replicas of any message in the system, minimizing network resource usage. Additionally, EBR bases routing decisions on a measure of a node's rate of encounters, showing preference to message exchanges with nodes that have high encounter rates. These routing decisions result in higher probability of message delivery, avoiding routes that may never result in delivery and so reducing the total number of message exchanges.

In EBR, information about a node's rate of encounter is a purely local metric and can be tracked using a small number of variables. Therefore, EBR is able to maintain very low state overhead, as compared to other protocols that can require up to $O(n)$ routing messages exchanged during *every* contact connection, and $O(n^2)$ routing state locally stored (*e.g.*, MaxProp [8], Prophet [23]).

A further strength of EBR is that its message replication rules are simple to understand and implement, as opposed to complex rules found in many protocols, minimizing the chance of bugs and reducing computational complexity (*e.g.*, the resources in terms of CPU cycles required to operate the protocol).

3.1 Algorithm

Every node running the EBR protocol is responsible for maintaining their past rate of encounter average, which is used to predict future encounter rates. When two nodes meet, the relative ratio of their respective rates of encounter determines the appropriate fraction of message replicas the nodes should exchange. The primary purpose of tracking the rate of encounter is to intelligently decide how many replicas of a message a node should transfer during a contact opportunity.

To track a node's rate of encounter, it maintains two pieces of local information: an encounter value (EV), and a current window counter (CWC). EV represents the node's past rate of encounters as an exponentially weighted moving average, while CWC is used to obtain information about the number of encounters in the current time interval. EV is periodically updated to account for the most recently CWC in which rate of encounter information was obtained. Updates to EV are computed as follows:

$$EV \leftarrow \alpha \cdot CWC + (1 - \alpha) \cdot EV.$$

This exponentially weighted moving average places an emphasis proportional to α on the most recent complete CWC. Updating CWC is straightforward: for every encounter, the CWC is incremented. When the current window update interval has expired, the encounter value is updated and the CWC is reset to zero. In our experiments, we found an α of 0.85 and update interval of around 30 seconds allow for reasonable results. These parameter choices are further elaborated upon in Section 5.

Since EV represents a prediction of the future rate of encounters for each node per time interval, the node with the highest EV represents a higher probability of successful message delivery. Therefore, when two nodes meet, they compare their EVs. The number of replicas of a message transferred during a contact opportunity is proportional to the ratio of the EVs of the nodes. For two nodes A and B, for every message M_i, node A sends

$$m_i \cdot \frac{EV_B}{EV_A + EV_B}$$

replicas of M_i, where m_i is the total number of M_i replicas stored at node A. For example, assume node A has 4 replicas of a message M_1 and 8 replicas of a message M_2. Furthermore, assume node A, with $EV_A = 5$, comes in contact with node B, with $EV_B = 15$. Node A sends $\frac{15}{5+15} = \frac{3}{4}$ of the replicas of each message. Therefore, node A transmits 3 replicas of message M_1 and 6 replicas of message M_2.

Algorithm 1 presents the basic form of the EBR protocol, where W_i represent the current window update interval parameter.

Algorithm 1 *EBRRouting*

```
if time ≥ nextUpdate then
  EV ← α · CWC + (1 − α) · EV
  CWC ← 0
  nextUpdate ← time + W_i
end if
if Contact C available then
  for All messages M_i in local buffer do
    m_i ← M_i.numOfReplicas
    m_send ← ⌊m_i · EV_c / (EV_c + EV)⌋
    Send m_send replicas of M_i to node C
  end for
end if
```

3.2 Generalizing EBR

In this section, we prove that EBR adheres to the definition of a quota-based protocol and show the relevant bounds, both for the simple version, where L, the maximum number of replicas of a message, is discrete, and for a more general version, allowing the use of probabilistic L values.

For discrete L values, it is easy to show that EBR is quota-based. Along with its data, every message contains a value indicating the maximum number of replicas into which this current message is allowed to be split. As an example, assume an application at node A creates a message with the maximum allowable replicas set to 10. Assume node A encounters node B and, based on the EBR protocol described in Section 3.1, wishes to transmit 8 replicas. Then, A creates a copy of the message for node B and assigns B's maximum allowable replicas to 8. Furthermore, A resets its maximum allowable replicas to 2. Continuing this procedure in a recursive fashion maintains the bound set by the initial message.

However, L values are not limited to a discrete maximum number of replicas. The discrete structure can easily be relaxed into a probabilistic structure, while maintaining meaningful (yet probabilistic) bounds. Probabilistic L values can allow for less sensitivity to exact network conditions. When using discrete L values, changes to the initial number of message replicas allows for a fundamental tradeoff between MDR, goodput, and average latency (see Section 5). Using probabilistic L values and increasing or decreasing variance and mean can allow applications to compromise and not require exact decisions about the number of allowable replicas.

While any distribution may be used in this probabilistic model, the Gaussian distribution allows for immediate, eloquent properties that help establish the bound on the number of messages in the network. In this case, the application specifies the mean and variance of the distribution, instead of a discrete number. Assume a node A wishes to split the message M into two replicas, M_A and M_B. Node A must follow the following EBR message splitting rule:

If $M \sim N(\mu, \sigma^2)$, then it can only be split into $M_A \sim N(\mu_A, \sigma_A^2)$ and $M_B \sim N(\mu_B, \sigma_B^2)$ such that $\mu = \mu_A + \mu_B$ and $\sigma^2 = \sigma_A^2 + \sigma_B^2$.

For example, a message with mean 10 and variance 5 may be split into two messages, one with mean 8 and variance 4, and one with mean 2 and variance 1. It may not, however, be split into a message of mean 8 and variance 4, and one with mean 7 and variance 1. As a further note, EBR maintains the ratio of mean to variance for all message splits.

This message splitting rule preserves the Gaussian distribution for the two newly created replicas. This is due to a result from statistics known as Cramer's Theorem:

- If $X + Y \sim N(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$,
  then $X \sim N(\mu_x, \sigma_x^2)$ and $Y \sim N(\mu_y, \sigma_y^2)$.

We now demonstrate that this general version of EBR is a quota-based replication protocol, and establish an upper bound, by proving the following theorem:

Theorem 3.2.1 *Let S be a schedule of future message creations. Let t be an arbitrary future time. Assume*

$M_1, M_2, ..., M_i \in S$ are all the messages created before time t. Assume each message M_i has a Gaussian random variable (for notational ease, we refer to this directly as the message M_i), with mean μ_i and variance σ_i^2, that represents the maximum number of replicas the current message is allowed to be split into.

The upper bound on the maximum number of message replicas in the system is:

$$U \sim N\left(\sum_{j=1}^{i} \mu_j, \sum_{j=1}^{i} \sigma_j^2\right).$$

Proof:

Let U be the sum of all message replicas in the system. Assuming messages never split, there will be i messages in the system, each with mean μ_i and variance σ_i^2. We utilize the following rule of linearity for Gaussian distributions (the converse of Cramer's Theorem):

- If $X \sim N(\mu_x, \sigma_x^2)$ and $Y \sim N(\mu_y, \sigma_y^2)$, then $X + Y \sim N(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$.

Therefore,

$$U = \sum_{j=1}^{i} M_i \sim N\left(\sum_{j=1}^{i} \mu_j, \sum_{j=1}^{i} \sigma_j^2\right).$$

Now assume a message, $M_j \sim N(\mu_j, \sigma_j^2)$ is split into $M_{j1} \sim N(\mu_{j1}, \sigma_{j1}^2)$ and $M_{j2} \sim N(\mu_{j2}, \sigma_{j2}^2)$ such that $\mu_j = \mu_{j1} + \mu_{j2}$ and $\sigma_j^2 = \sigma_{j1}^2 + \sigma_{j2}^2$ (the message splitting rule of EBR). Then by the same linearity rules, $M_j = M_{j1} + M_{j2}$, leaving U unchanged. *QED*

One minor issue to address is that the statistical rules and theorems each assume true Gaussian distributions. However, it does not make sense in our system for a message M to hold a negative value. The probability of this occurring can be made sufficiently small by forcing the application to choose sufficiently low variances for corresponding means (which can never be below zero).

Chapter 4

Disaster Network Mobility Model

In order to properly evaluate EBR, we present an event-driven, role-based mobility model for disaster recovery networks. First, we discuss the properties of objects in disaster recovery scenarios that affect their mobility patterns. Next, we present the details of our event- and role-based mobility paradigm. Following this, we present a gravity-based low-level mobility model that can be used with our high-level paradigm. Finally, we present a methodology used to study the properties of our model as well as simulation results that help characterize their properties.

4.1 Modeling Object Behavior

Modeling the movement behavior of mobile objects has been heavily studied. Due to its simplicity and effectiveness, the Random Walk [16] model is widely used to model such object behavior. In this model, a node randomly chooses a direction in $[0, 2\pi)$ along with a speed, and moves according to those choices for a set amount of time. After this time has expired, the node repeats the process. A recent model by Jardosh *et al.* [19] takes polygon-shaped objects into account by using Voronoi diagrams to build walks. While many of these models are adequate for their particular environments, all objects generally act in the same way and, therefore, do not give the flexibility required for modeling disaster scenarios. This is because real people and vehicles take on roles, allowing them to react to events in a distinct fashion.

Mobility in disaster recovery scenarios is fundamentally driven by environmental events. These events act as stimuli towards objects and directly cause them to change their movement patterns. While some current models, including [19], could be extended to react to external stimuli, truly capturing the complex interactions between more than one environmental event requires building new mobility models with event-driven actions as the primitive concept. Furthermore, while all objects react to relevant events, different classes of objects react in different ways. In other words, object behavior changes over time and is not uniform across all nodes. Objects also must react to multiple events in a realistic and smooth fashion. There is currently no adequate mobility model that takes into account these observations.

To illustrate this idea with an example, consider an apartment fire in a populated neighborhood. There are, intuitively, at least three different classes of behavior that objects can assume: (1) fleeing from the event, as is the case with civilians, (2) approaching the event with the intent of staying, as is the case with police and firefighters, and (3) oscillating from the event to a predetermined location, as is the case with ambulances. These high-level behavior classes, which we refer to as roles, help give general but clear mobility patterns, which are realistic and relatively easy to simulate.

Roles, however, need not be limited to specific movement patterns during a disaster scenario. It is easy to extend the concept of a role to cause an object to react differently during different stages of an event. For example, cleanup crews may want to react to disaster events by approaching them long after the event has occurred, whereas police may instead want to approach the event immediately.

In response to the need for unique behavior modeling in disaster scenarios, we have developed a high-level, role-based, event-driven mobility paradigm in which different roles take on different mobility patterns in reaction to specific events. Our paradigm is high level in the sense that an object may take on multiple mobility patterns over the course of some time period. For instance, a civilian may first be modeled by Random Walk. After some time, an event, such as a fire, may trigger the civilian to change its model to one of fleeing from the fire. By modeling mobility as a series of event-driven, role-based actions, we can properly select which specific mobility model to use for a given object and situation. While the specific rules for reaction should be based on observational studies, our paradigm is sufficiently general enough to allow future, accurate movement patterns to be used.

Our paradigm allows for different objects to react to events in unique ways by attaching a mobility pattern for each possible (event, role) combination. These lower level mobility patterns can be any of the previously developed mobility models, including Random Walk. To further capture the behavior of certain roles in a disaster scenario, we have developed a low level physics-based gravitational model that allows objects to flee or approach disaster events in an intuitively realistic fashion. It is important to note that our high level, event- & role-based paradigm is not tied to the gravitational model in any way, and can support any of the previously defined low level mobility models.

Our gravitational model captures the effect-distance relationship between events and objects, allowing events to act on objects via forces. Objects closer to events are affected more than objects farther away. Additionally, each event has an *event horizon*, defined as the maximum distance at which an event affects objects, which allows events to have a defined radius of effect. Finally, there is a *communication threshold*, which defines the time until emergency vehicles are notified of an event. Until this time, emergency vehicles outside the event horizon do not respond to the event; however, after the threshold time has passed, those emergency vehicles begin to converge on the event. It is possible that multiple events could take place in a single scenario. Additionally, these events may or may not be simultaneous. One of the major benefits of the gravitational model is that it easily captures the interactions between multiple events.

The classification of behavior into roles can also play a part in establishing realistic communication patterns. For instance, civilians are most likely in contact only with police and other civilians, police are in contact with all roles, firefighters are in contact with police and other firefighters, and ambulances are in contact with police. These communication patterns, along with the mobility, can simplistically model an entire disaster scenario. For this thesis, however, we concentrate solely on modeling mobility,

4.2 Disaster Mobility Paradigm

We now formally describe our high-level mobility paradigm that incorporates external, environmental events along with role-based reaction. By classifying objects with roles, which define reactions to events, our paradigm can realistically obtain a set of (role, event, action) triples that define the overall movement patterns of objects in disaster recovery scenarios. A single triple can be read as follows: "Role r reacts to event e by taking action a". Then, by instantiating the triples with the characteristics for different agents operating in a disaster scenario, a mobility pattern can be generated for the scene.

Three entities, and their relationships to one another, help define our high level mobility paradigm: objects, roles, and events. *Objects* are nodes in the system that provide movement and communication. Each object assumes a *role*, or set of roles, that indicate what movement pattern the object should assume in response to external stimuli. The specific areas of interest that provide external stimuli to objects are referred to as *events*. The event-based response a role dictates to an object is an *action*, which is generally a low-level mobility model, such as Random Walk or the gravitational model presented in Section 4.3.1.

We now elaborate on these three entities and their relationships to one another.

Objects are the critical components of the scenario, including people, buildings, and vehicles, and are defined by the following parameters:

- Location: The (x,y,z) location of the object.
- Role: The role, or set of roles, associated with the object.
- Velocity: The current velocity of the object (in vector form).

Roles dictate how objects react to events. We define four main categories of roles, although it is possible to define any number of them. First, the *repelling* category causes objects to be repelled from events. The low-level mobility models that support this role should allow objects to move away from events in a realistic and easy-to-use fashion. The most common use of this role is to model normal civilians in a disaster scenario. An attribute of this role is *curiosity*, which dictates how likely it is for an object to stop at the event horizon, simulating curious on-lookers. Second, the *attraction* role causes objects to converge on events. Low-level mobility models that support this role should cause objects to quickly approach an event or events. Common uses of this role are to model police and firefighters. Third, the *oscillating* role models objects that first approach an event and then, upon reaching the event, travel immediately to a predefined location. This movement pattern is then repeated continuously. Low-level mobility models that support this role should allow this action to be as realistic as possible. One use of this role is to model an emergency response system in which ambulances oscillate between the event and a hospital. Finally, the *immobile* role models any object that remains stationary for the duration of the simulation or until the object takes on a new role. This role can be supported by the lack of a low-level mobility model, since it does not perform any movement. This role is useful to model both naturally static objects, such as hospitals, as well as event-caused immobility.

We anticipate that the default action dictated by many roles is Random Walk, since it simply models motion when movement patterns are unknown or seem random. It is important to note, however, that any mobility pattern, such as one that accounts for navigating around buildings or objects, can be used.

Events act as the stimuli for mobility changes in the scenario. In a real disaster scenario, an object's proximity to an event is a major factor in how it reacts. Therefore, it is important to clearly mark distinct areas of an event. Our paradigm captures this type of behavior by defining the *Disaster Radius*, the *Event Horizon*, and the *Relevant Radius*. Reaction to an event is also dependent on time, which is modeled by the *Start Time* and the *Radio Contact Time*. The following defines the full set of parameters for an event.

- Location: The (x,y,z) location of the event.
- Start Time: Time when the event occurs.
- End Time: Time when the event ends.
- Radio Contact Time: Time when radio contact outside of the event horizon occurs.
- Disaster Radius: Area inside which all objects become immobile.
- Event Horizon: Area inside which all objects react to the event, even before radio contact occurs.
- Relevant Radius: Area inside which objects, based on their role, react to the event after radio contact occurs, assuming they are not already reacting. Some roles, such as the civilian role, may not react when inside this radius but outside the event horizon.
- Intensity: A numeric representation of the event's current intensity.

4.3 A Disaster Mobility Model

With the high-level disaster mobility paradigm formalized, we now describe a disaster mobility model that we believe intuitively models simple disaster scenarios. For implementation purposes, we have made some simplifying assumptions. First, we assume events are stationary, have a constant intensity, and, after their start, persist for the remainder of the simulation. Furthermore, objects assume a single initial role and do not change it for the duration of the simulation, unless changing to the immobile role. Finally, all roles except the Immobile role default to the Random Walk action. At the end of this section, we discuss how to capture events that are mobile and change intensity, as well as events that are not best represented by a single point.

4.3.1 Gravitational Model

Intuitively, many roles react to disaster events by either fleeing from or approaching them. To model these actions in the presence of multiple events, we use a physics-based gravitational model to define the "Flee" and "Approach" actions. Gravitation has been used to model group mobility dynamics, particularly in [29], but not event-based mobility. We designed this model based on the observation that objects, in general, either gravitate towards or away from disaster events.

Physics states that the gravitational force between two objects with masses m_1 and m_2 at a distance d from each other is:

$$F = \frac{G \cdot m_1 \cdot m_2}{d^2}$$

where G is the gravitational constant. The total resulting vector force on an object in the vicinity of multiple objects is calculated by the vector sum of all forces on the object. The resulting force directly affects the object's acceleration.

We borrow the concepts of gravity and force from physics to model the *flee* and *approach* actions. By letting m_2 be the "mass" of a given event, and assuming m_1 is negligible, the force on an object by that event can be described as $F = I/d^2$, where I, the intensity, encompasses G and m_2. Mobile objects can then be repelled or attracted to events (or multiple events) by assigning a particular intensity to every event.

To calculate the motion trajectories of objects as a result of multiple forces, we sway from physics slightly to allow for a more intuitively realistic movement pattern. Physics states that forces directly affect an object's acceleration. However, humans are more concerned with maintaining a particular velocity at a given time than maintaining a particular acceleration. We generally do not maintain a constant acceleration, but rather accelerate quickly to a desired velocity and then hold an acceleration of zero. Therefore, it is intuitively more correct to say that humans will adjust their *speed*, not their *acceleration*, according to how far they are from a disaster event (of course, they will adjust their acceleration to obtain that speed, but only for a short period of time). Therefore, in our gravitational model, forces act directly on velocity, not acceleration, to account for this phenomenon. The benefit of taking a gravitational-based approach to model mobility is that it allows the reaction of objects to be intuitive and easy to compute for multiple, unsynchronized, dynamic events.

4.3.2 Disaster Model

Let M be the set of (role, event, action) triples that define our mobility model. M is populated by adding triples to cover all components or the desired scenario. All mobile nodes initially start using the Random Walk Model to either walk or drive, with speeds appropriately bounded. Formally, there are a set of initial (role, event, action) triples for each role as follows:

$\{$*(r, No Event, Random Walk) :* $r \in R\}$,

where R is the set of all possible roles. The "No Event" event is simply the default event every role assumes when there are no relevant events in the network.

If a disaster event occurs, two areas are immediately formed. The first area is ground-zero, as defined by the disaster area parameter, within which objects are immediately immobilized. We model this by simply immobilizing all nodes within a set radius of the disaster event. We formally model this by the inclusion of the following triples into M:

$\{$*(r, DE1 - At Ground-Zero, Switch to Immobile) :* $r \in R\}$

The "DE1 - At Ground-Zero" event is a disaster event (with the label "DE1" representing disaster event #1) that has occurred when the object was within the disaster radius of the event. Once an object is immobile, it stays immobile for the remainder of the simulation. To accomplish this, the action "Switch to Immobile" instructs the object to immediately switch roles to the "Immobile" role. This role is defined in M as follows:

(Immobile, No Event, Stay Still)

It is important to note that this should be the only entry for the immobile role, since it should always default to staying still. Static objects, such as a hospital, are also assigned the immobile role.

The second area is defined by the *event horizon*. All objects within the event horizon react to the event by either gravitating towards it or fleeing from it, at a speed dependent on the object's proximity to the event. The inclusion of a set of triples into M formally models this phenomenon. For instance, the following triples define the area within the event horizon for a simple disaster scenario:

(Civilian, DE1 - In Event Horizon, Flee)

(Police, DE1 - In Event Horizon, Approach)

(Firefighter, DE1 - In Event Horizon, Approach)

(Ambulance, DE1 - In Event Horizon, Oscillate)

The event "DE1 - In Event Horizon" event refers to the situation that the object is within the event horizon radius of a disaster event. Notice that while all the events beginning with "DE1" are technically the same event, to incorporate proximity into the action taken by a role, we break the event into multiple areas (or regions), in which roles reacting to the same event may respond differently based on which area of the event they are in.

After the radio contact time of the event expires, the relevant radius of the event is formed. Roles with radio contact in this region, but outside the event horizon, should react to the event. Continuing from the previous example, the following tuples in M formally define this area:

(Police, DE1 - In Relevant Radius, Approach)

(Firefighter, DE1 - In Relevant Radius, Approach)

(Ambulance, DE1 - In Relevant Radius, Oscillate)

An object may have multiple applicable triples at any given instance. This would occur, for example, if a civilian were in the radii of two different events. Our gravitational model easily accommodates scenarios of this type.

It is possible to extend our model to account for events of different shapes and sizes, as well as mobile events. Currently, events provide forces from a central point within the event, and have different radii that allow for a circular (or spherical) shape. Elongated event shapes, such as floods, can be simulated by placing multiple events close to each other at varying intensities. Furthermore, there is nothing prohibiting the changing of intensity or location of an event, as forces can be quickly recomputed at every object's location based only on current information. The natural memoryless computation of forces allows for highly dynamic events.

4.4 Analyzing Mobility Models

The benefits of an effective mobility model come from its ability to capture and expose the characteristics of the network and the behavior of nodes in the network. This information can then be used by network designers to understand how to design protocols that are suitable for the specific scenarios. In this section, we first discuss the metrics necessary to describe network behavior in disaster scenarios. Although most evaluations focus on simple network characteristics, such as node density and path length, the unique behavior of nodes in a disaster scenario results in more interesting network conditions that require us to look at more complex parameters, such as average node density and partitioning. We then present a set of tools that we implemented to generate ns2 mobility scenario files. In the next section, we present our evaluation of a number of these metrics using our tools.

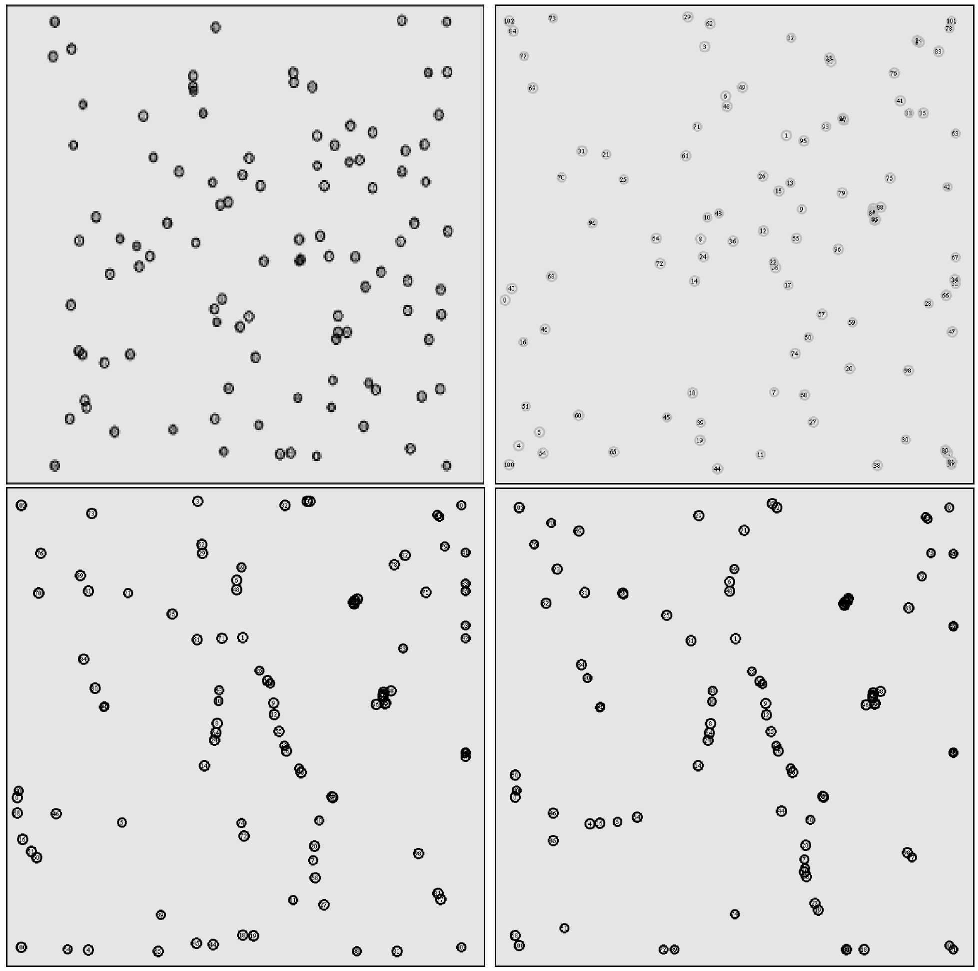


Figure 4.1: Network snapshots at time 0, 200, 400, and 1400 seconds

4.4.1 Metrics

When discussing mobility models, it is important to understand how a model affects different topological network metrics. Two standard metrics are average node density and average path length. Average node density, as defined by the average number of neighbors per node, can be used to help characterize the potential connectivity of a network, since a network with low density will likely be partitioned. Average path length, as defined by the average number of hops from source to destination, captures the distance between sources and destinations. However, due to the highly dynamic nature of networks under disaster scenarios, it is important to not only consider these metrics but also those that show how the structure of the network progresses over time. For an event-driven, role-based mobility model, the following network parameters highlight how the network is changing over time.

- Partitioning over Time: The average path length metric is meaningless when the graph is partitioned, which is likely in many disaster scenarios. Therefore, tracking whether or not the graph is partitioned is critical to understanding the flux in network topology.

- Clustering Coefficient over Time: The clustering coefficient of a particular node is the number of that node's neighbors that are connected to each other divided by the total possible links between them [37]. The higher the value of this metric, the more clustered the graph is.

- Average Node Density over Time: Although node density is an important metric, in an event-driven mobility model, it is important to capture how density changes in *reaction* to different events, which indicates churn.

- Maximum Node Density over Time: Maximum node density gives insight into the potential cluster sizes, which can provide insight into potential bottlenecks. Although it would be useful to track actual cluster sizes, maximum node density provides a much cheaper, though quite effective, heuristic.

- Variance of Node Density over Time: Since some parts of the network may be more stable than others, the variance in node density gives insight into the amount of variance in cluster sizes as a result of different events.

4.4.2 Tools

To evaluate the impact of our model on the metrics described above, we have implemented two tools for the ns2 simulator. The first tool is a parameters file generator that creates a properly formatted parameters file appropriately choosing random values when necessary. This tool prompts the user for the following input: size of the network (in terms of meters squared), number of civilians, number of ambulances, and number of police. Since both the police and firefighter roles are similar, we have chosen to omit firefighters and simply add more police to simulate firefighters. However, it is quite simple to include firefighters, or other responders, and give them appropriate behaviors. The output parameters file contains the following information:

- Grid size and simulation runtime

- Randomized coordinates for all objects and events, and coordinates for four hospitals located at the corners of the grid

- Minimum and maximum speeds for objects
- Percentage of curious civilians
- Random Walk parameters
- Randomized trigger times and radio contact times for four events
- Randomized intensities for events, which determine radii for event horizons and damage zones

The specific parameters generated for the experiments are detailed in Section 4.5.1. For any given input, this tool produces unique output since many parameters are randomly chosen. Usage for this tool is as follows:

Usage: paramGen > paramFile.

The second tool is our main event-driven simulator. This tool accepts as input the parameters file generated by the first tool and runs a complete simulation with knowledge of Random Walk and our physics-based gravitational model. It is important to note, however, that any mobility model can be plugged into the tool in place of Random Walk and/or the gravitational model. The output of this tool is an ns2-compatible mobility trace file that gives the current velocity and destination of every object at every second in the simulation. All of the mobility model logic is performed in this tool. For any given input, the output of this tool again produces unique output, since Random Walks performed by objects not reacting to events may differ from one simulation to the next. Usage for this tool is as follows:

Usage: disasterSim [-d] < paramFile > nsMobilityTrace.

The *-d* flag, when passed, displays each individual step of the simulation via "ASCII art" to the console. This is printed to standard error, so it is not written to the *nsMobilityTrace* file.

4.5 Evaluation

To analyze the difference in network topology changes generated by our disaster recovery mobility model, we generated numerous topologies and ran simulations with them using ns2. Using the same initial setups, in terms of node placement and numbers, we ran the simulations using the Random Walk model for comparison. In this section, we present results from 10 different sets of simulations, each with its own group of both deterministically and randomly chosen parameters. Each of the metrics presented in Section 4.4.1 are evaluated for each of the resulting sets of trace files. Sufficient numbers of experiments were run to minimize the 95% confidence interval.

For this thesis, we were only interested in the mobility patterns of objects in a disaster scenario and the topological affects the patterns have on the network graph. Therefore, we did not simulate communication between nodes. However, we did specify the communication range to be 150 meters to obtain information about links in the network.

4.5.1 Simulation Parameters

The simulation time runs from 0 to 1500 seconds with a grid size of 1000 m^2. The communication range of every node is 150 m to simulate an urban environment. There are 75 civilians, 10 ambulances, and 15 police each randomly located. A total of 90% of civilians, randomly chosen, are considered curious, meaning they stop at event horizons to look at the event. Furthermore, we have chosen a minimum and maximum speed of 1 m/s and 4 m/s respectively for civilians, and 17 m/s and 20 m/s for ambulances and police. All Random Walks are done for 30 seconds.

Four events are randomly placed on the grid. The first event occurs randomly between 100 and 200 seconds, the second between 125 and 225 seconds, the third between 150 and 250 seconds, and the fourth between 175 and 275 seconds. Radio contact for a specific event occurs randomly between 40 to 80 seconds after the event has occurred. The intensity of events are randomly chosen between 10000 m^3/s and 20000 m^3/s. The event horizon for each event is 2% of the intensity and the damage radius is 0.1% of the intensity.

We choose high intensity events to easily illuminate the differences between the topology of our model versus the topology of the Random Walk model. We also choose to include hospitals as stationary objects, since they will most likely participate in communication with other objects (particularly ambulances and police). Therefore, there are a total of 104 objects in the system, 4 being immobile from the beginning. Furthermore, we assumed that radio contact for all events reached ambulance and police regardless of where they were. All responders are in either CB or cellular radio contact at all times, meaning that the relevant radius for each event is set large enough to encompass the entire grid.

The model we present contains a large parameter space. This is due to not knowing how the different parameters affect communication and routing in disaster events. As future work, we plan to explore the effects of these parameters and simplify the mobility model based on those findings.

4.5.2 Snapshot of Topology Change

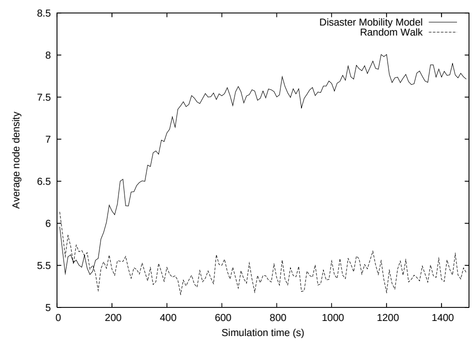


Figure 4.2: Average node density

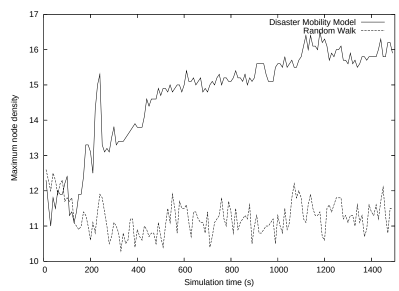


Figure 4.3: Maximum node density

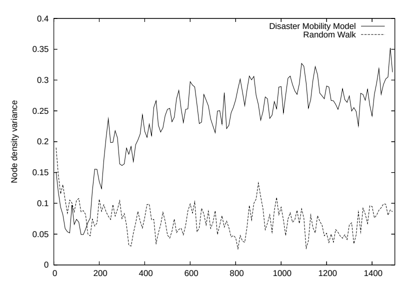


Figure 4.4: Variance of node density

Figure 5.1 shows a series of four snapshots during one simulation run of our disaster mobility model. The first box shows the state of the network at the start of the simulation. Object location at this point is random, except for the 4 hospitals located at each corner of the grid. The second box shows the state of the network 200 seconds into the simulation. At this time, some events have triggered but radio contact for many has not occurred. Only objects within the event horizon have reacted at this time. The third box shows the state of the network 400 seconds into the simulation. By this time, all events have been triggered and radio-contact has been made. All emergency response objects (police and ambulances) and civilians within the event horizon have reacted to the event. It is now possible to see some of the different roles active in the system, simply by visually observing their locations. The fourth box shows the state of the network 1400 seconds into the simulation. By this time, most metrics have come close to convergence and mobility is noticeable only by ambulances and civilians who have not approached the event horizon. Civilians who are not curious and have left the event horizon are mobile again.

A clear topological difference between the disaster mobility model and Random Walk is primarily due to the clustering of objects around the event horizon. This separates the graph into three primary areas: (1) areas inside event horizons, (2) areas at or near event horizons, (3) areas outside of event horizons. The first area is very sparse since all civilians able to move leave the scene. Almost all of the concentration is in the "damage radius", since emergency response workers immediately move towards that zone. The only interaction between that zone and the event horizon are the oscillators when they pass through the event horizon. The second area is very dense since all of the civilians inside the event horizon gravitate towards its edge and civilians that happen to stumble into the event horizon stay there. The third area contains objects who are performing random walks and have not been notified of the event. As the simulation continuously runs, the third area should slowly lose objects to the second area since they randomly hit the event horizon.

This series of snapshots clearly shows the location of events and the formation of crowds of people around the event horizon. It also illustrates the behavior of ambulances going to and from events and hospitals. We would expect very similar results in a real disaster scenario, further confirming our implementation.

4.5.3 Metric Evaluation

Figures 4.2 through 4.6 clearly show the event-driven response of the metrics around the time of the events. Between 0 and 100 seconds, the data sets are similar for all metrics, as expected. Between 100 and 355 seconds, during the triggering of events and radio contact time, a clear divergence between the disaster mobility model and Random Walk model is readily seen as the topology of the event-driven simulation starts to take form.

Figure 4.2 shows the average node density of the network as time progresses. The average node density in the disaster mobility model increases in response to events. This is due to the gathering of nodes around the event horizon, forcing them into a smaller area than before. An interest-

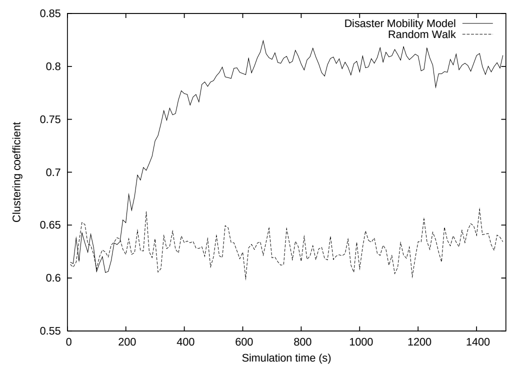


Figure 4.5: Clustering coefficient

ing observation is that the size and frequency of the oscillations in average node density become both smaller and less frequent in the disaster mobility model as time progresses. This is due to the topological convergence of the disaster mobility model that does not occur in Random Walk. The difference in average node density between the disaster mobility model and Random Walk is important because it gives overall information as to how many neighbors a node can expect to have at a given time and so provides hints about network connectivity.

Figure 4.3 shows the maximum node density between the two data sets as time progresses. The maximum node density quickly increases in response to the events. There are two highly-clustered areas for each event in the system, the area inside of the damage radius and the event horizon. The jump in maximum node density is due to the quick response by the police to the event, increasing the node density of nodes at the damage radius. After this, the maximum node density remains relatively constant from around 500 to 900 seconds, as the maximum density around the event horizon starts to catch up to the maximum density around the damage radius. At around 900 seconds, the maximum density starts to increase as the maximum density of the event horizon increases. The diminishing of high-frequency oscillations as time progresses is, again, due to the convergence of the disaster mobility model not found in Random Walk.

Figure 4.4 shows the variance of node density as the simulation progresses in time. The variance of node density clearly increases as events are triggered. This is because many nodes have either a fairly small node density (if they are being partitioned or close to partition in the graph), or a high density (if they are clustered at the event horizon or damage radius). It is interesting to note that the high-frequency oscillations do not seem to diminish as time progresses. This is likely due to both the Random Walk civilians and the ambulances oscillating between hospitals and events.

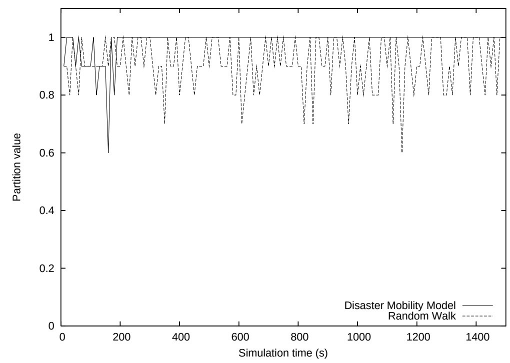


Figure 4.6: Network partitioning

Figure 4.5 shows the clustering coefficient in the network as the simulation progresses in time. The clustering coefficient of a node i is defined as:

$$C_i = \frac{2\,|\{e_{jk}\}|}{k_i(k_i - 1)} : v_j, v_k \in N_i, e_{jk} \in E,$$

where N_i are the neighbors of i, E is the set of edges in the graph, and k_i is the degree of node i [37] (this definition assumes an undirected graph). This gives a general indication of how well a node's neighbors know each other, which in turn gives insight into how clustered the network is. We define the clustering coefficient of a node with degree less than 2 to be 0. The average graph clustering coefficient increases sharply in response to the events. This is again due to clustering around the event horizon and damage radius for each event. As before, the diminishing of high-frequency oscillations is apparent.

At any given time, a graph is either partitioned or not. If it is, we say it has a "partition value" of 1. If not, it has a "partition value" of 0. For each data point, we have averaged the partition value of each of the 10 simulations. Figure 4.6 shows that the average partition value in the disaster mobility model increases as a result of the events. In fact, due to the high intensity of the events, after around 200 seconds the network is always partitioned in the disaster mobility model. This is because the nodes at the events are partitioned from the rest of the network, since the event horizons are generally out of their communications range. The disaster mobility model consistently has a partition value higher or equal to that of Random Walk, indicating a more fragile network.

These results show that our mobility model produces a topology much different then that of the popular Random Walk model. The vast difference between the topologies indicate that it is not sufficient to use Random Walk as a mobility model for disaster recovery networks.

Chapter 5

Evaluation

The primary goal of our evaluation is to show that EBR achieves a high message delivery ratio and good latency, while maintaining extremely low overhead. To demonstrate this, we first present the metrics used in our evaluation, followed by a brief description of the mobility models. Finally, we present a comprehensive evaluation of EBR in comparison to five other popular DTN routing protocols. To perform our evaluation, we use the Opportunistic Network Environment simulator (ONE) [2], which is a simulation environment designed specifically for disruption tolerant networks.

5.1 Metrics

Although traditional evaluation metrics provide a good understanding of the performance of a network, the evaluation of many current DTN routing protocols is hindered by the limited, and sometimes inappropriate, metrics used. Given the challenged environments and tolerant applications targeted by DTNs, it is important to consider meaningful ways to evaluate protocols against one another. In many cases, metrics heavily used in other environments, such as packet delivery ratio and end-to-end latency, have been trivially ported to DTN environments. However, such an approach can result in misleading evaluations of the protocols due to the inability of the metrics to account for the actual goals of the network. Therefore, we consider three traditional performance metrics and introduce three composite metrics that capture the big picture performance in DTNs.

Due to the disruptive nature of DTNs, applications benefit by creating stand alone data messages that are meaningful even in the presence of other packets failures [30]. Therefore, we discuss our evaluation metrics in terms of application messages, instead of lower layer packets. In comparison to such lower layer packets, messages in DTNs are generally larger with more variance in size.

Traditional performance metrics include message delivery ratio and latency, while resource usage can be captured by goodput. In a resource constrained network, effective use of available storage can be captured by the number of messages dropped due to buffer overflows. We evaluated this metric in all our scenarios; however, since it closely correlates to goodput, those results were omitted due to space constraints.

Message Delivery Ratio (MDR): MDR is defined as the total number of messages received by all destinations in the network divided by the total number of messages sent to all destinations in the network. This metric is the DTN equivalent of packet delivery ratio.

Average End-to-End Latency: The end-to-end latency of a message is defined as the difference in the time since the routing protocol on the message's source router sent the message to the time when the routing protocol on the message's destination router received the message. Messages that were not delivered are not included in this metric. Therefore, this metric is more meaningful if the MDR's of all protocols being compared are similar. This in an important metric since many messages lose relevance after being delayed for a long period of time.

Goodput: Goodput is defined as the number of messages delivered over the total number of messages transferred (including those transfers that did not result in a delivery). Intuitively, this metric corresponds to the amount of extra messages transmitted and stored in the system, thereby giving a view of resource usage and stewardship. Essentially, goodput indicates how "resource-friendly" the protocol is in terms of bandwidth and storage. Although some protocols evaluate a limited definition of goodput [33, 34], some recent protocols do not consider goodput at all [8, 3].

Although these three metrics provide a comprehensive view of the communication in DTNs, many protocols trade off effectiveness in one metric for effectiveness in another. Therefore, we evaluate all of the protocols using three composite metrics that capture the relationship between these individual metrics and provide greater insight into the individual metrics and trade-offs made by the protocols. For example, in our experiments, we found routing protocols that perform very well in terms of average end-to-end latency, but had poor delivery ratios. This is not uncommon, since messages that only travel one-hop generally have short end-to-end latency, and if only one-hop messages are delivered, the delivery ratio will be poor. Similarly, delivery ratios may be high, but at the unacceptable expense of network resource drain. Composite metrics are able to penalize protocols for performing poorly in individual primary metrics, giving a more complete picture of protocol performance.

Composite metrics are the product of their indicated primary metrics. To maintain the standard "higher is better" approach, the average delay is always inverted when used.

- MDR x Average Delay
- MDR x Goodput
- MDR x Average Delay x Goodput

These composite metrics illustrate the relative relationship between the primary metrics. The *MDR x Average Delay* metric takes MDR and penalizes it for having a poor end-to-end delay, allowing for a more complete picture. Similarly, the *MDR x Goodput* metric looks at MDR and penalizes it for having poor goodput, giving a view of the network stewardship along with traditional MDR. Finally, the *MDR x Average Delay x Goodput* metric looks at MDR and penalizes it both for poor average delay and poor goodput. It is important to note that the absolute value of composite metrics is more or less meaningless by itself, since the metrics are artificial in nature. Therefore, when comparing protocols using composite metrics, one should consider the protocols' relative performance to one another.

5.2 Experimental Setup

To perform our evaluation, we use the Opportunistic Network Environment simulator (ONE) [2], which is a simulation environment designed specifically for disruption tolerant networks. The ONE simulator does not take into account network interference. Therefore, we added one-hop interference code to defer transmissions until there are no neighbors transmitting. This is, of course, a very conservative interference model, and we expect that more realistic interference models will support our goal of reducing network resource usage.

We evaluate EBR against five other popular protocols: (1) basic epidemic [35], (2) Prophet [23], (3) Spray and Wait [33], (4) Spray and Focus [34], and (5) MaxProp [8]. Spray and Focus extends Spray and Wait by forwarding single-copy messages to optimize a specific utility function. To enable a comparison between EBR and Spray and Focus, we implemented Spray and Focus to use an EBR-style encounter value (EV) to optimize delivery ratios in the focus phase. This version of Spray and Focus shows that EBR is fundamentally different. When nodes running Spray and Focus are in the focus phase, they hand-off single-copy messages to nodes with a higher EV.

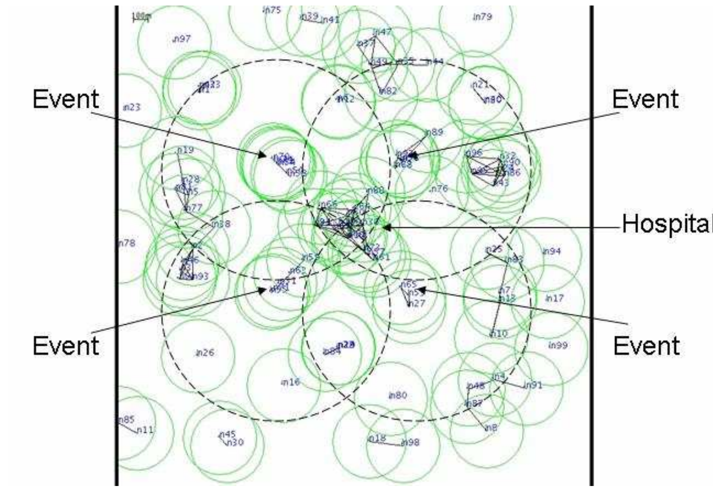


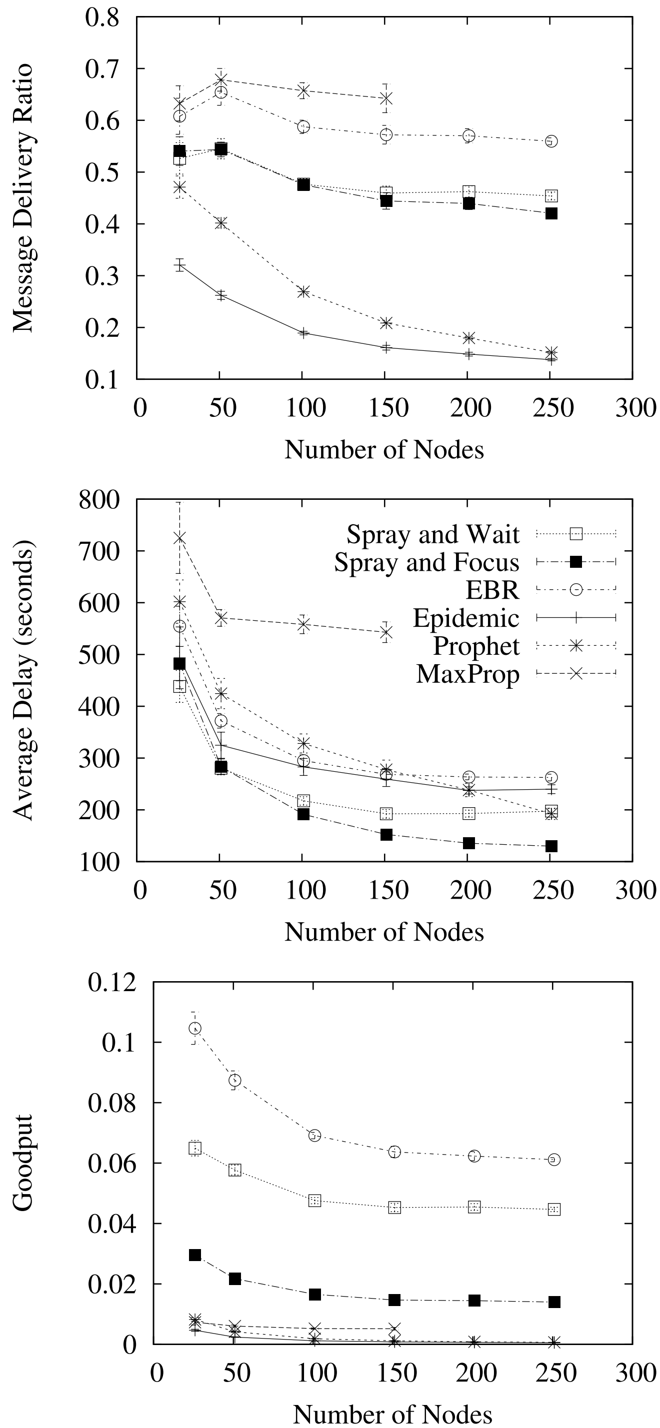
Figure 5.1: Disaster Model Snapshot

5.3 Mobility Models

Since DTNs can operate in many different environments, we use three different mobility models in our evaluation, specifically chosen to encompass a wide variety of DTN environments: an event-driven model simulating a disaster scenario [24], a map-driven model simulating a vehicular network, and a traditional random waypoint (RWP) model. We include the RWP model to demonstrate that protocols designed for random mobility may work well if the mobility is random, but their performance suffers when applied to models that capture more realistic mobility patterns.

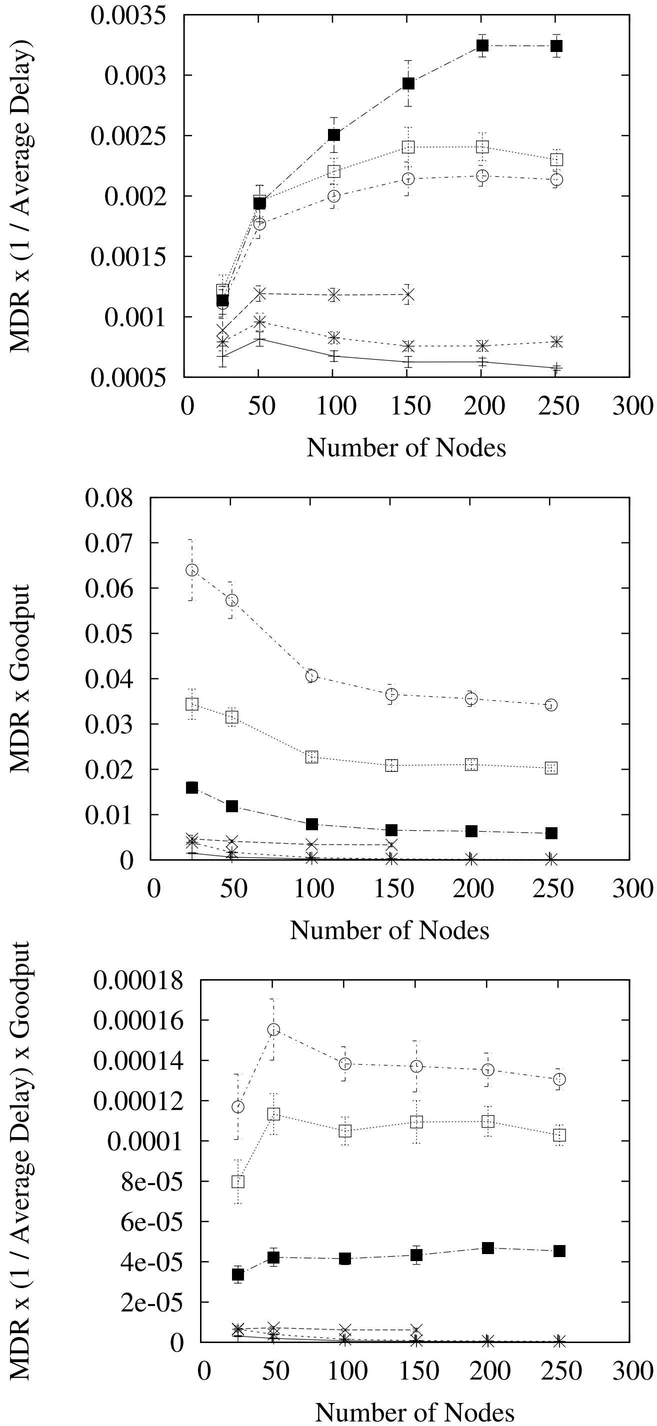
The role-based, event-driven disaster mobility model [24] captures distinct movement patterns of roles as they react to external events. For this model, we simulate four equally spaced disaster events and a hospital. 50% of the nodes are civilians that flee from the events, 25% are ambulances that oscillate to and from events and a centrally located hospital, and 25% are police personnel who at first gravitate towards an event, but then react by "patrolling" the area in a random walk fashion. An example snapshot of the model halfway through the simulation is shown in Figure 5.1, where the four events and the hospital can be seen by the clustering of nodes. Police and ambulances always travel between 17 and 20 m/s, unless stopped. Civilians always travel between 1 and 4 m/s, unless stopped. The dotted circles approximate the event horizons of the four events, were civilian clustering occurs. 90% of civilians are considered "curious", and stop at event horizons.

To evaluate the protocols in a realistic vehicular-based DTN, we utilize the map-driven mobility model implemented in the ONE simulator. This map-driven model limits node movement



(a) (b) (c)

Figure 5.2: Disaster: Varying number of nodes (a) MDR, (b) Average Delay, (c) Goodput



(a) (b) (c)

Figure 5.3: Disaster: Varying number of nodes (a) MDR x Average Delay (b) MDR x Goodput, (c) MDR x Goodput x Average Delay

to actual streets found on an imported map, an approximate 5 km x 3 km section of downtown Helsinki, Finland. The model can define certain nodes as Points of Interest (POI) so that they will have a configurable probability of being selected as the next destination. Approximately 15% of the nodes were configured to follow pre-defined routes (like tram lines) with speed between 7 and 10 m/s, the default for trams in the ONE simulator. The rest of the nodes were divided into four groups of nodes and four groups of POIs. Each node group was assigned different probabilities of picking the next node from a particular group of POIs to simulate the phenomenon that people often visit certain areas of a city more frequently than others based on their profession, age and other factors. The speed of these nodes varied between 2.7 and 13.9 m/s, the default for car simulation in the ONE simulator.

Finally, we simulate the routing protocols with a traditional random waypoint model. Nodes are initially placed uniformly at random in the simulation area. Nodes then chose a random location and travel in a straight line to that location. After arriving and pausing for some time, the procedure repeats. For our simulation, nodes are relatively slow moving, since the disaster scenario and vehicular models are relatively fast moving. Nodes move between 0.5 and 1.5 m/s, and stop for some time between 0 and 120 seconds.

For the disaster and random waypoint mobility models, the simulation area is 3 km by 3 km. For all simulations, the transmission range of each node is 250 m.

5.4 Performance Results

To demonstrate the effectiveness of EBR, we perform two groups of simulations on each of the three mobility models. To illustrate how each of the protocols reacts to changes in node density, we vary the number of nodes in the network starting at 26, followed by 51 to 251 in increments of 50, while keeping the area constant. The extra node represents a hospital in the middle of simulation area for the purpose of the disaster scenario mobility model. To illustrate how each protocol reacts to varying network loads, we vary the per-node offered load by adjusting the number of messages sent per minute per source from 1 (lower load), to 2 (medium load), to 4 (higher low). Following this comparative evaluation, we evaluate how EBR reacts to changes in two local parameters: the popularity counter weighting constant (α) and the number of initial replicas per message.

In all simulations we keep the area constant, the packet size constant at 25 KB, and the buffer space constant at 1 MB. Each simulation lasts for one simulated hour. Unless otherwise noted, each data point is the average of at least 10 runs, with 95% confidence intervals displayed. Due to the large amount of time required to simulate MaxProp, it was only evaluated fully for 26, 51, and 101 nodes, and is the average of four runs for 151 nodes, and is not evaluated for higher numbers of nodes. The slow performance of MaxProp is due to the frequent calculations for the path finding algorithms required. We believe, however, that the data obtained allows for a reasonable evaluation and comparison at lower number of nodes. MaxProp is omitted from the evaluation using the vehicular mobility model due to the large amount of time required to simulate it.

5.4.1 Comparative Results

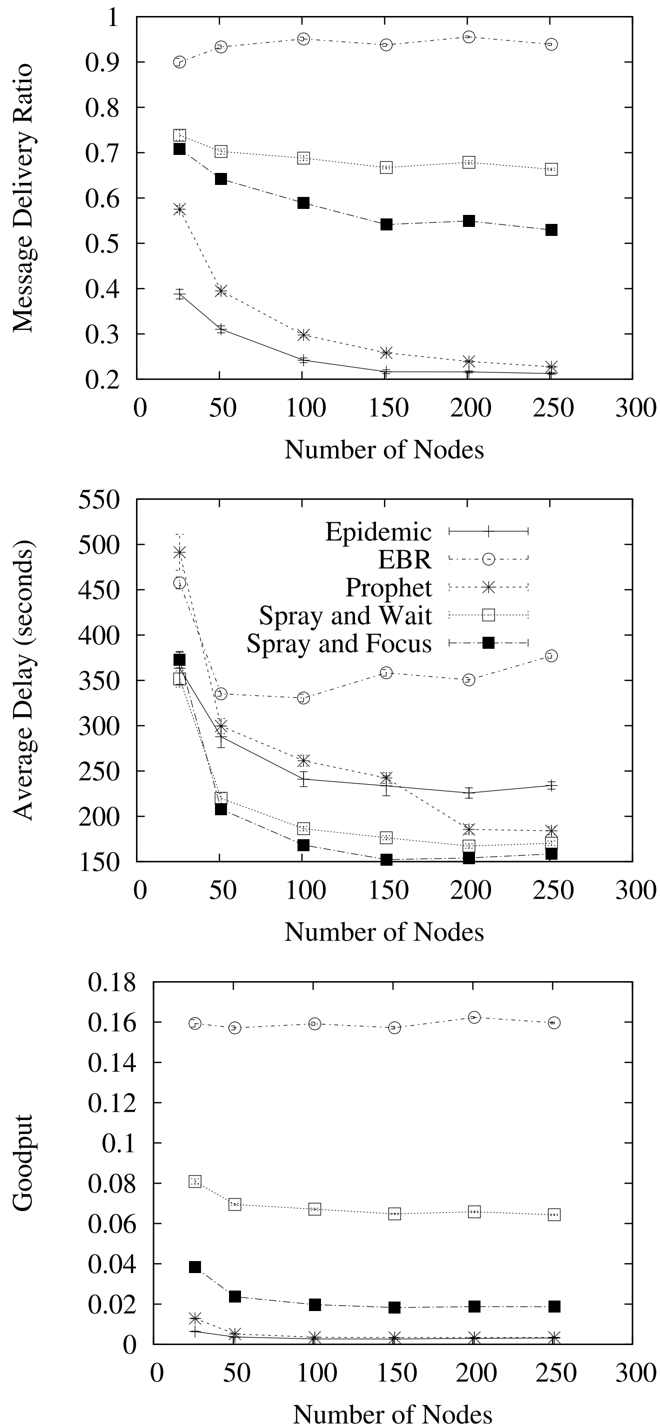
In the first group of simulations, we vary the number of nodes in the system to determine how this variable affects the protocols in relation to the three traditional metrics and the three composite metrics. As expected, in terms of MDR, MaxProp performs the best (see Figure 5.2(a)), due to its aggressive use of network resources. This strategy allows MaxProp to utilize as much storage and bandwidth as possible to maximize MDR. Closely following is EBR, which is never greater than 9 percentage points away from MaxProp. This is significant since EBR is much less demanding on network resources (as shown by the goodput metric in Figure 5.2(c)), yet can achieve a comparable MDR. Spray and Wait, which performs closest to EBR in terms of goodput (yet still significantly worse), performs noticeably worse in MDR. The reason EBR performs much better than Spray and Wait is due to the role-based characteristics of the disaster scenario mobility model. Both ambulances and police are highly active, more-so than civilians, and so EBR's assumption about predicting the rate of encounters using past data holds true. Furthermore, the goodput is significantly higher using EBR because if a large number of copies reach a high-encounter node, that node will not forward many of these copies to low-encounter nodes. This helps keep the network resource usages much lower than Spray and Wait. Note that both Prophet and Epidemic collapse as the number of nodes increases. In terms of latency, MaxProp performs worst, whereas Spray and Focus performs expectedly well (see Figure 5.2(b)).

The three composite metrics help put the total performance of the protocols into perspective. Spray and Focus performs the best when combining MDR with average delay (see Figure 5.3(a)), because the performance hit for MDR is not enough to offset its strong average latency performance. Notice, however, that EBR performs substantially better than MaxProp, due to better delay, and Prophet and Epidemic, due to better MDR. In terms of MDR with goodput, EBR performs at more than twice that of the second highest, Spray and Wait, and much greater than the other metrics (see Figure 5.3(b)). This, again, is due to EBR's efficient use of resources, which is complemented by the role-based mobility pattern of the disaster model. Finally, we evaluate how well each protocol performs when combining all three primary metrics (see Figure 5.3(c)). In this case, again due to high MDR and goodput, EBR performs substantially better than the second highest, Spray and Wait, which subsequently performs substantially better than the third highest, Spray and Focus.

Next, we present the results from the vehicular mobility model. Note that MaxProp is not included in this set of simulations due to the large amount of time necessary to simulate it on the ONE simulator. EBR performs extremely well in terms of MDR, compared to the other quota-based protocols, Spray and Wait and Spray and Focus (see Figure 5.4(a)). Two factors account for this. First, the mobility model fits perfectly into the assumptions of EBR, making EBR an appropriate protocol for this scenario. More specifically, past information on rate-of-encounters is a good estimator for future rate-of-encounters. Second, the network utilization seems to be correlated to MDR in this scenario, most likely due to constrained buffer space. EBR is, by far, the most resource friendly, as shown by the goodput metric (see Figure 5.4(c). While EBR seems to have unfavorable delay, this is, in part, due to a high MDR (see Figure 5.4(b)). Since delay is computed only over messages that have been delivered, it is deceptive to view delay alone since many protocols quickly deliver messages that take a small number of hops, and do not deliver most high-hop messages.

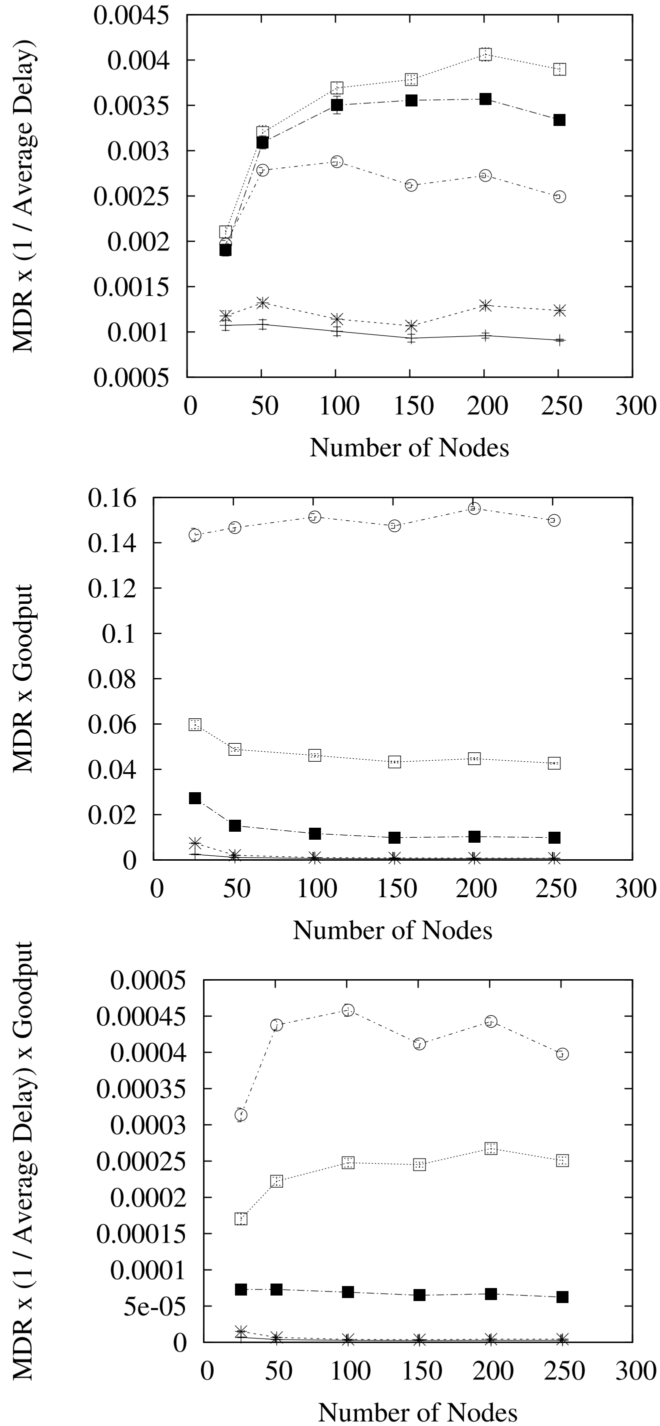
To obtain a more complete view of the protocols in light of a vehicular mobility model, the composite metrics are considered. Due to good delay results obtained by Spray and Wait and Spray and Focus, MDR with delay favors these protocols, with EBR following (see Figure 5.5(a)). Note however, that when goodput is combined with MDR, EBR performs exceptionally well (see Figure 5.5(b)). This is due to the vehicular network abiding by the assumptions of past rate of encounters predicting future rate of encounters, meaning that high rate of encounter nodes do not replicate often. Furthermore, EBR's high MDR and exceptionally high goodput allow it to perform the best when combining all metrics (see Figure 5.5(c)).

Finally, the random waypoint model is considered. Since EBR was designed to leverage heterogeneity in node mobility, it is not surprising that EBR does not perform as well for the random waypoint model. It is interesting to note, however, that the other metrics also drop, most likely due to the relatively slow speed at which the nodes travel. In terms of MDR (see Figure 5.6(a)),



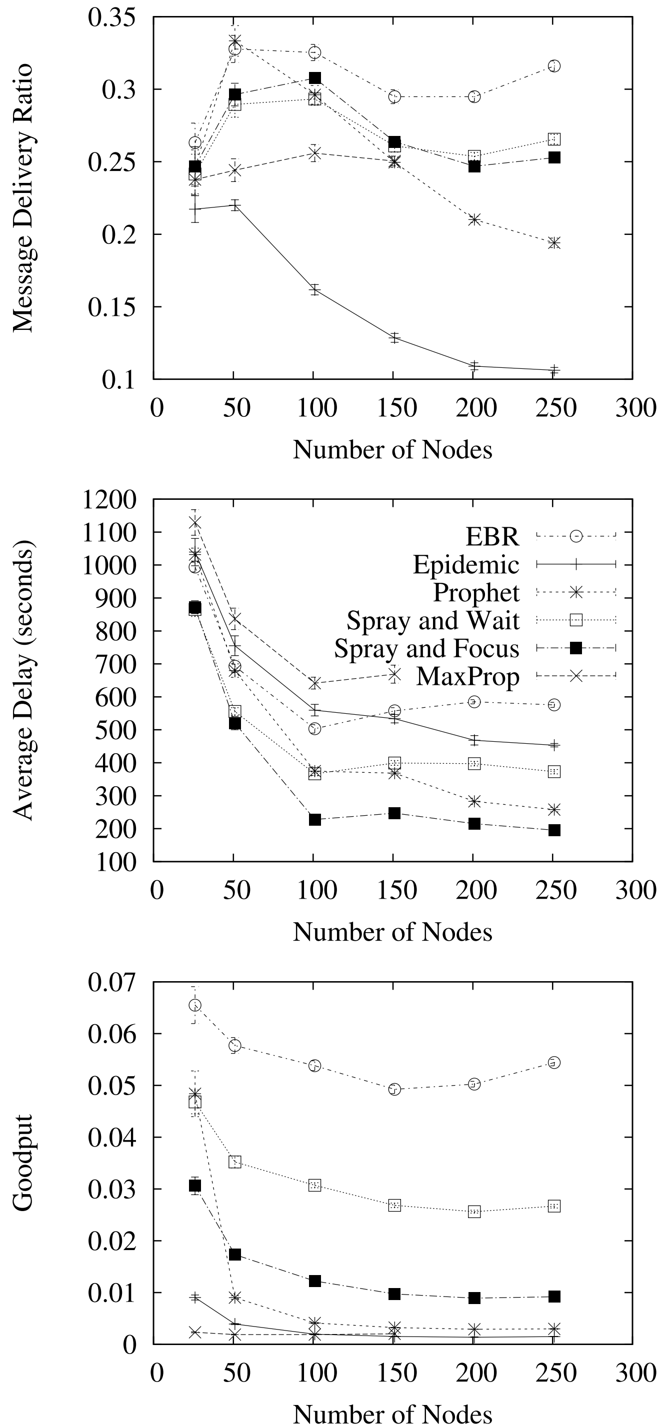
(a) (b) (c)

Figure 5.4: Vehicular: Varying number of nodes (a) MDR, (b) Average Delay, (c) Goodput



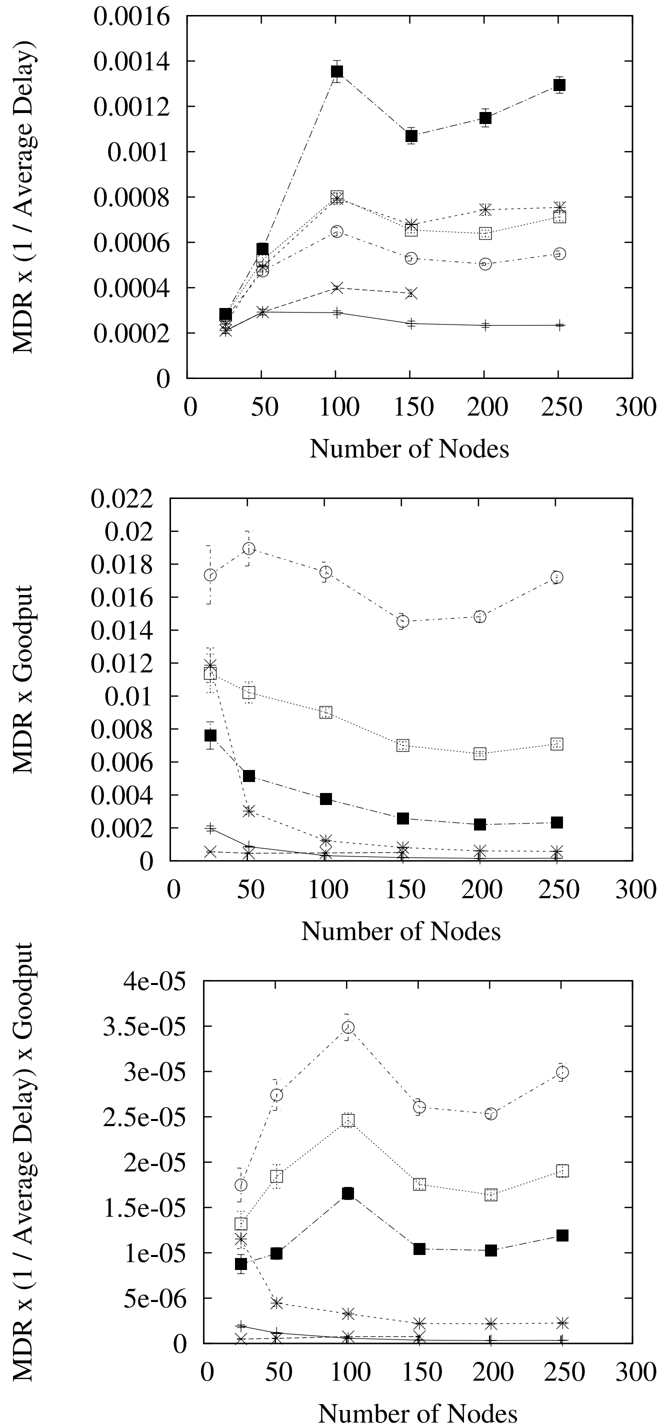
(a) (b) (c)

Figure 5.5: Vehicular: Varying number of nodes (a) MDR x Average Delay (b) MDR x Goodput, (c) MDR x Goodput x Average Delay



(a) (b) (c)

Figure 5.6: RWP: Varying number of nodes (a) MDR, (b) Average Delay, (c) Goodput



(a) (b) (c)

Figure 5.7: RWP: Varying number of nodes (a) MDR x Average Delay, (b) MDR x Goodput, (c) MDR x Goodput and Average Delay

the gap between EBR and Spray and Wait is closer than with the disaster scenario (notice the change in scale). However, as the number of nodes increases, the gap becomes larger. The sudden increase at 50 to 100 nodes is due to the density finally becoming adequate for good delivery. Past this point, there is a minor decrease in performance for EBR, Spray and Wait and Spray and Focus and a more dramatic decrease for Prophet and Epidemic. One interesting observation is that MaxProp performs consistently worse than EBR, Spray and Wait, and Spray and Focus, up to 151 nodes. We believe this is due to the relatively small amount of packets storable in a buffer (40), and if this buffer size were to increase, MaxProp would perform better. Prophet starts out strong, but quickly deteriorates as the number of nodes increase. In terms of latency, Spray and Focus again performs the best (see Figure 5.6(b)); however, EBR consistently performs better than MaxProp. As expected, the goodput metric strongly favors EBR (see Figure 5.6(c)). The gap between EBR and Spray and Wait is significant, but not quite as significant as with the disaster mobility model. This is due to EBR having to disseminate more copies of messages, as there is a more uniform rate-of-encounter for nodes.

The composite metrics help give a more complete view for the RWP scenario. Due to Spray and Focus's low average latency, it is able to overcome any MDR hindrance in the MDR with average latency metric (see Figure 5.7(a)). Also note that Prophet performs better than it did in the disaster scenario mobility model. When comparing MDR with goodput, and MDR with average delay and goodput (see Figure 5.7(b) and Figure 5.7(c)), EBR performs significantly better than all other metrics, seconded by Spray and Wait. These results indicate that, even in mobility models EBR is not specifically designed for, it maintains superiority in many important metrics.

In the second group of simulations, the offered load is varied from 1 to 2 to 4 messages per source per minute. This group of simulations illustrates how the routing protocols react to increased network load, while maintaining the same number of nodes. The number of nodes for all simulations in this class is 101. For this group of simulations, we present the results for the disaster mobility model and the random waypoint model.

For the disaster scenario, MaxProp and EBR perform expectedly well, with all protocols suffering as the offered load increases (see Figure 5.8(a)). The average latency, however, shows MaxProp performing much worse than the other metrics (see Figure 5.8(b)). Furthermore, as the offer load is increased from 1 to 4 messages per source per minute, EBR performs better than both Prophet and Epidemic. This is due to EBR's sharper drop in MDR as offer load increases. Spray and Focus and Spray and Wait perform the best, as expected.

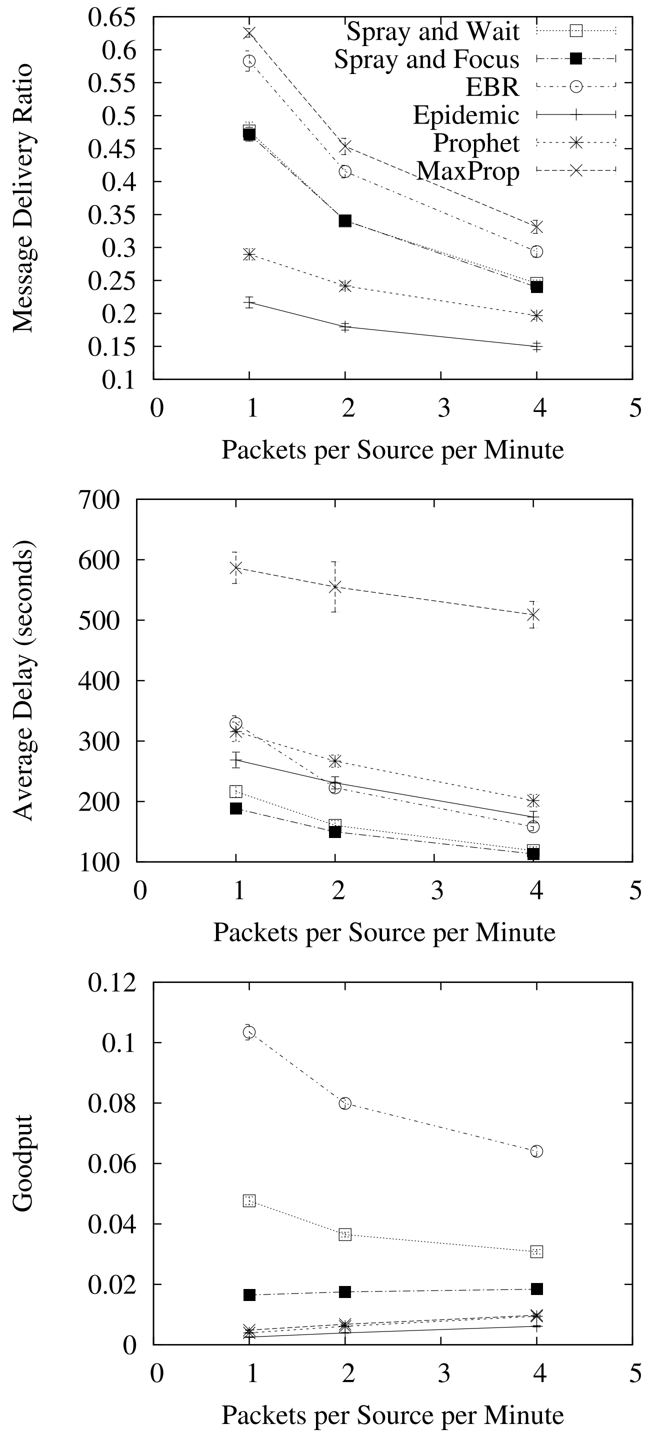
Figure 5.9 shows the results of the combination metrics for this class of simulations using the disaster scenario mobility model. In terms of MDR with average delay, there are two clear groups, with Spray and Wait, Spray and Focus, and EBR being in the top group, and the others being in the bottom group (see Figure 5.9(a)). Out of the top group, EBR is the only protocol that does not suffer as load increases, but instead closes the gap between itself and the top two protocols. In terms of MDR with goodput, EBR performs significantly better than the other protocols at all offered loads, with the gap tightening as the load increases (see Figure 5.9(b)). Finally, when combining all primary metrics, EBR performs significantly greater than the others, with Spray and Wait as second best (see Figure 5.9(c)).

When the offered load is varied using the RWP mobility model, the MaxProp data is averaged over three runs, with all other data averaged over ten runs. Due to the more uniform nature of per node rate of encounters, EBR does not perform as well as it does in the disaster scenario mobility model. However, in terms of MDR, it is still in the top tier, and performs higher than all others with lower offered loads (see Figure 5.10(a)). In terms of latency, as the offered load increases, the gaps between protocols tends to close (see Figure 5.10(b)). When combining all primary metrics, we notice that EBR performs at the highest level, primarily due to high overhead, and reasonable MDR and latency (see Figure 5.11(a)(b)(c)).

5.4.2 EBR Parameter Results

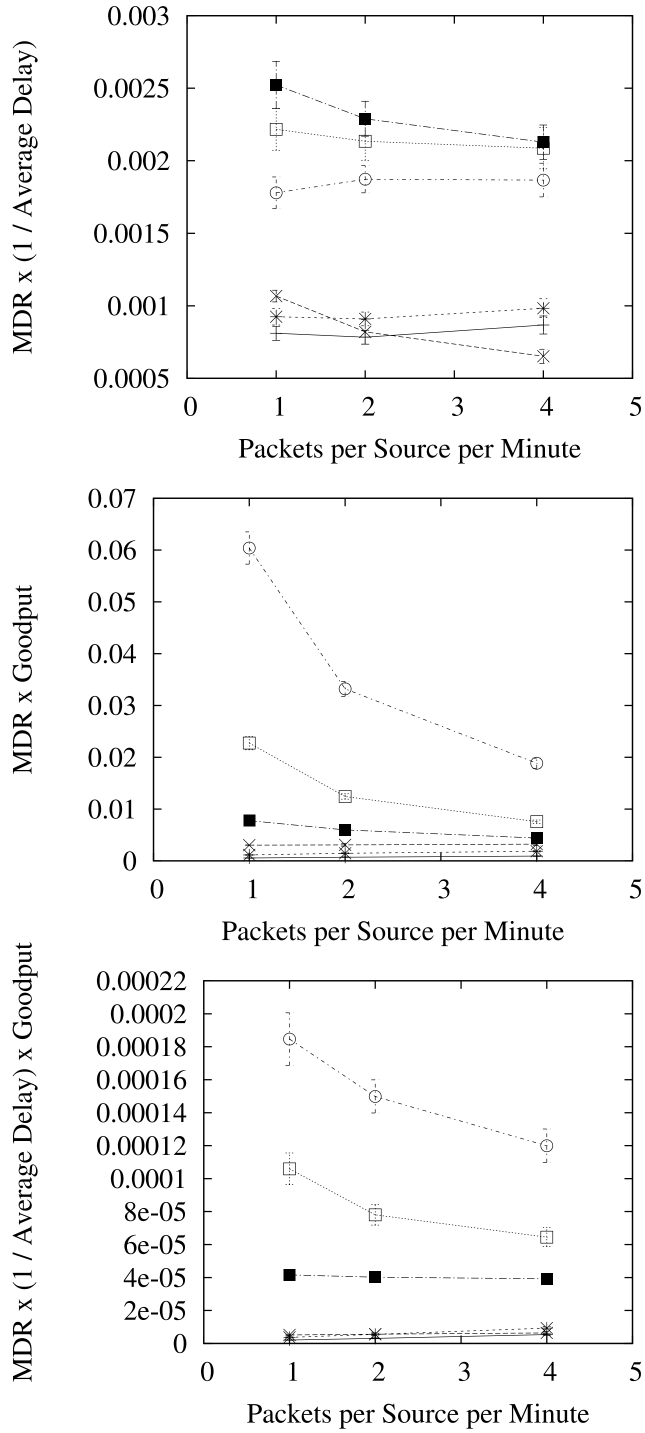
To determine how EBR reacts to changes in internal parameters, we evaluate EBR against itself using different parameter settings. Due to space constraints, we only present results for the disaster scenario mobility model and only vary the number of nodes in the system. To evaluate the impact of the weight of the current rate of encounter in the EV counter, we vary α from 0.5 to 0.85. Additionally, to capture the tradeoff between resource usage and delay, we vary the starting number of message copies between 5, 11, and 20. Therefore, a total of 6 lines are shown per graphs. Again due to space constraints, we only present the graphs for the primary metrics, not the composite metrics.

In terms of MDR, α does not make a substantial difference. However, the number of initial copies does. As the number of nodes grows larger, EBR using only 5 copies starts to perform best, with EBR using 11 copies within a few percentage points (see Figure 5.12(a)). However, in terms of average delay, EBR using 5 copies performs significantly worse than with both 11 and 20 copies (see Figure 5.12(b)). Again, changing the value of α has little effect. The goodput is significantly



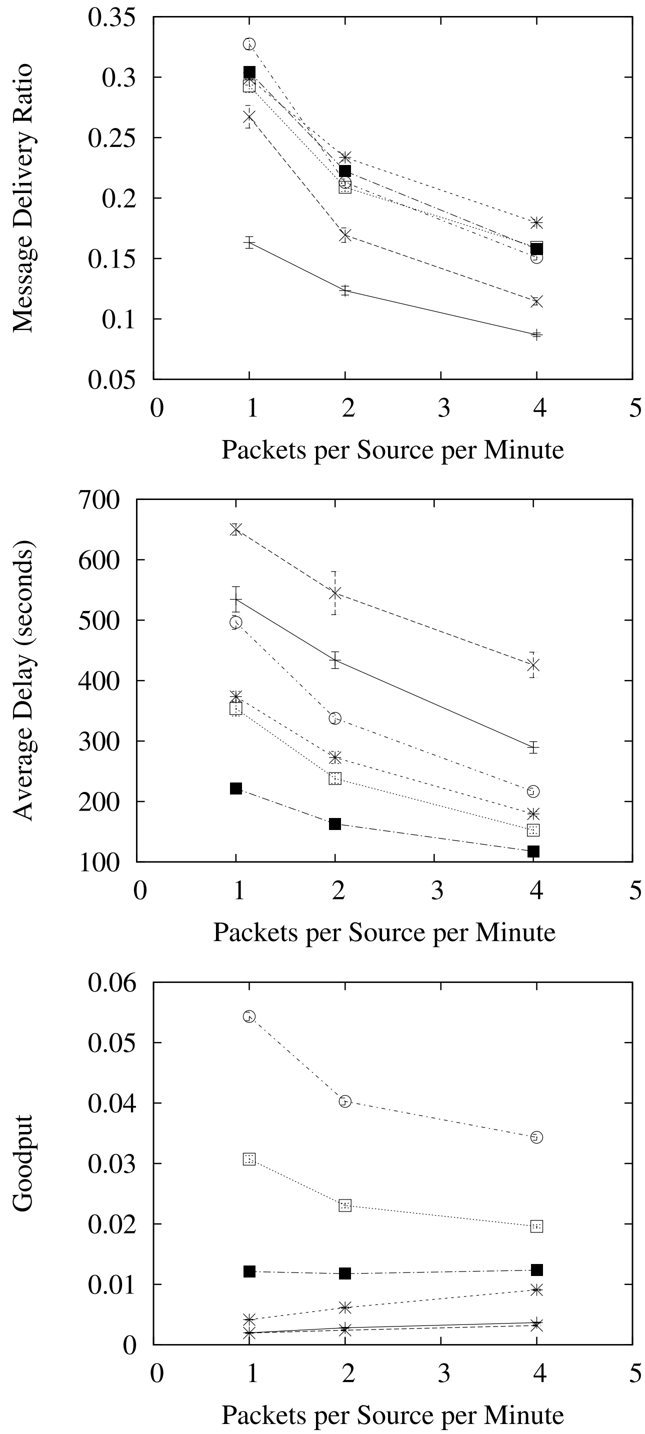
(a) (b) (c)

Figure 5.8: Disaster: Varying load (a) MDR, (b) Average Delay, (c) Goodput



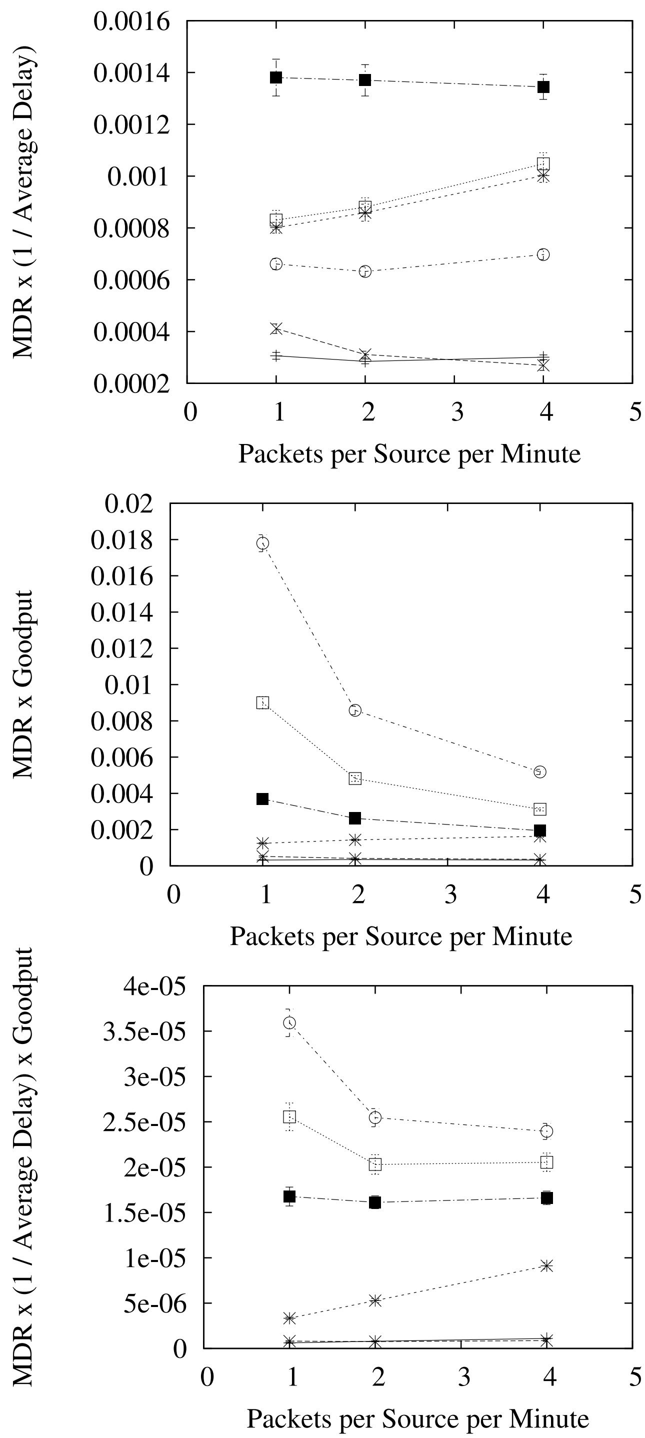
(a) (b) (c)

Figure 5.9: Disaster: Varying load (a) MDR x Average Delay, (b) MDR x Goodput, (c) MDR x Goodput x Average Delay



(a) (b) (c)

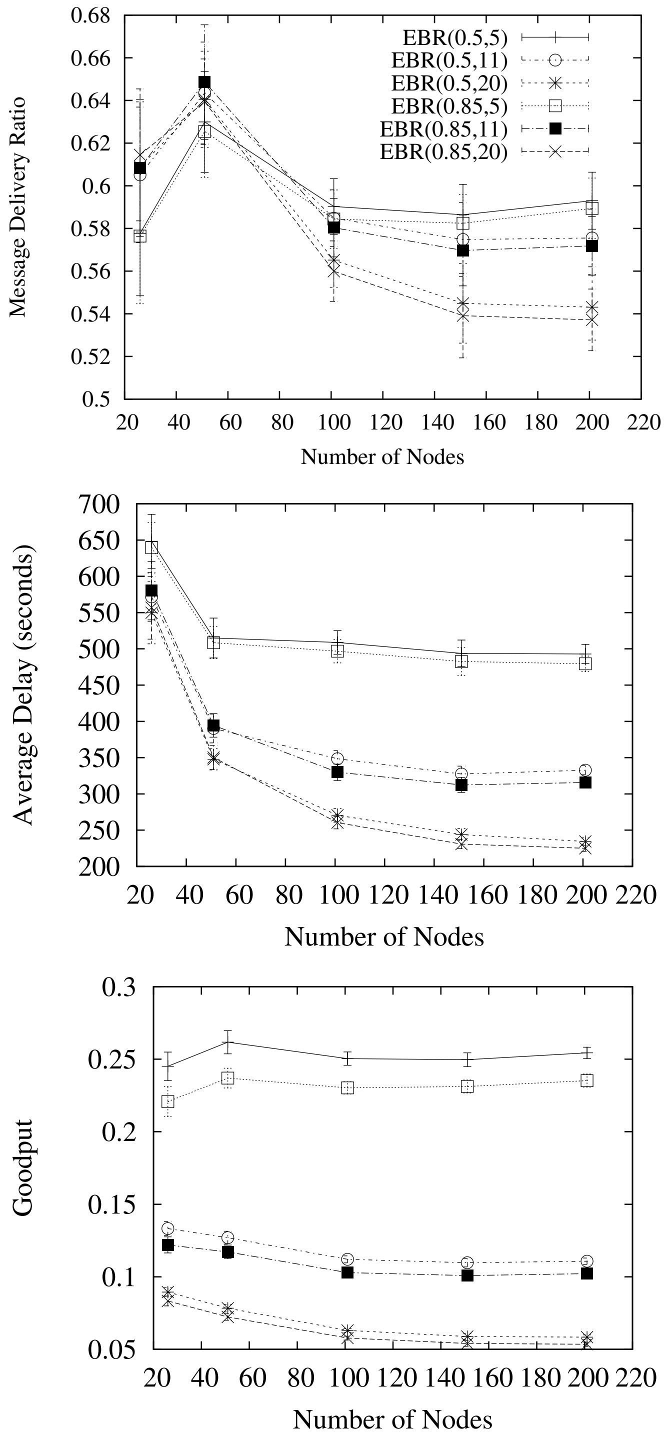
Figure 5.10: RWP: Varying load (a) MDR, (b) Average Delay, (c) Goodput



(a) (b) (c)

Figure 5.11: RWP: Varying load (a) MDR x Average Delay, (b) MDR x Goodput, (c) MDR x Goodput x Average Delay

greater when the number of copies is small, as expected (see Figure 5.12(c)). In total, when not considering latency, a small number of copies, such as 5, allows for good performance of EBR. However, when latency is considered, a bit of a trade off must be made. Therefore, we have chosen to compromise and recommend a value of 11 initial copies as default to EBR.



(a) (b) (c)

Figure 5.12: Disaster: Varying number of nodes (a) MDR, (b) Average Delay, (c) Goodput

Chapter 6

Securing EBR

The decision regarding how many replicas of a messages a node should transmit to a contact depends completely upon the ratio of both parties' encounter values. Therefore, a malicious node can convince a node following protocol to transmit virtually any percentage of replicas to it. One of the most worrisome results is the possibility of a denial-of-service (DoS) attack where malicious nodes act as "black holes". Malicious nodes performing this attack advertise an ultra-high encounter value, causing all contacts to send almost all replicas to them. The malicious nodes then simply delete these messages, attempting to stop, or at least slow, message delivery.

Work by Burgess *et. al* shows that two popular types of denial-of-service attacks, dropping all messages (which we refer to as black hole denial-of-service) and flooding the network with fake messages, result in similar network degradation [7]. This degradation does not cripple the network because malicious nodes suffer from the same level of intermittent connectivity as non-malicious nodes. In this thesis, we have chosen to consider the case of black hole DoS attacks. This is because EBR is a low-overhead quota-based protocol, and hence extra flooding is not as big a concern as black holes. In quota-based protocols, non-malicious nodes do not flood messages, real or fake, and should simply drop messages with a high number of copies, since they must be from malicious nodes.

To determine how vulnerable EBR is to black hole DoS attacks, we perform a series of simulations where a certain percentage of the nodes are malicious. Malicious nodes always advertise an exceptionally high encounter value, and immediately delete any message replicas obtained. Each data point is the average of 10 runs, and small 95% confidence intervals are shown. A vehicular mobility model is used, which is explained, along with simulation parameters, further in Section 5. The results of this experiment, shown in Figure 6.1, indicate that network performance can be hindered with a relatively small number of malicious nodes. This is concluded due to the sharp initial drop in message delivery ratio with only a small percentage of nodes being malicious. However, matching the work done by Burgess *et. al*, additional malicious nodes are not able to cripple

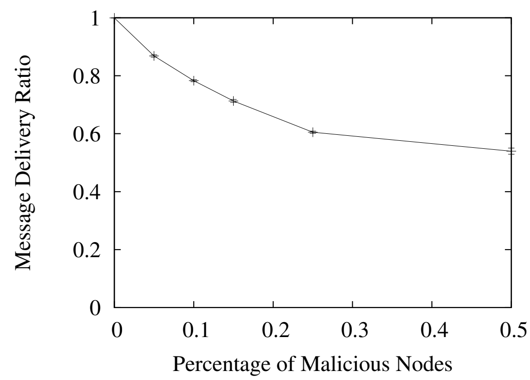


Figure 6.1: MDR in Attack Scenarios

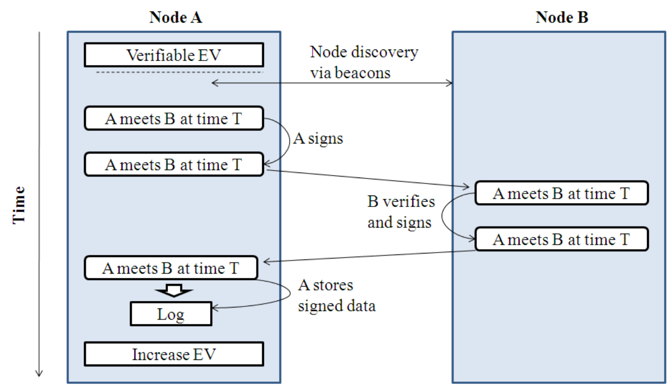


Figure 6.2: Timestamp Protocol

the network. These results indicate that it is necessary to provide an optional solution that prevents DoS attacks. Users not minding the decrease in performance may choose not to implement this solution. However, providing a solution is necessary for those users more concerned about maximizing network performance. The penalty for choosing the solution is that there must exist a means of digitally signing data as well as binding keys to identities, such as PKI.

The insight of the solution comes from the observation that an encounter value can *never* be altered unless an external event (e.g., coming in contact with another node) occurs. Therefore, proving that the encounter value was altered only during an external event assures other nodes that the node in question is not individually faking the value. Now, of course, nodes can still collude to artificially inflate their encounter values; this case will be considered shortly. Note that the goal is to prevent the artificial increase, not decrease, of encounter values.

The protocol works as follows. Assume node A comes in contact with node C, and node C wishes to send data to node A. The goal is for node A to offer acceptable evidence to node C that the encounter value is not forged. To give acceptable evidence for this, node A must keep a list of transactions in which all previously encountered nodes digitally sign a time stamped message stating that “node A met me at time T”. A graphical illustration of this is given in Figure 6.2. Node A can then offer all of these messages to node C, and allow node C to recompute node A’s encounter value from scratch. If the recomputed value is equal to the value provided by node A, then node C can confidently transmit replicas to node A.

It is possible, even probable, that inherently trustworthy nodes are present in the network. For instance, in disaster recovery networks, police and emergency responders can be considered highly trustworthy entities. These nodes can be utilized to sign, or *checkpoint*, actual encounter values. This checkpointing process allows a node to delete all previous transactions and simply start with the new, signed encounter value. Checkpointing nodes verify the encounter value in the same fashion as mentioned above and then provide a signed encounter value back to the node. Checkpointing nodes must be trusted by all nodes in the network since previous transaction data is deleted after a signed encounter value is obtained (e.g., a node is checkpointed by a checkpointing node).

It is possible for colluding nodes to artificially inflate each other’s encounter values by signing multiple “fake” meeting messages. This is a difficult problem, and we have not discovered a clear-cut solution. However, using statistical techniques, nodes diligent in looking for abnormal contact rates can mitigate the damage. If a node legitimately meets another node or group of nodes very frequently, it can lessen its chances of raising a false red flag by simply not storing some of the meetings, and not updating its encounter value for those meetings. A more thorough investigation of this is future work.

Chapter 7

Conclusions and Future Directions

The ability to efficiently and effectively support communications in disaster recovery networks will greatly help emergency responders. In this thesis, we present a disaster recovery mobility model and use that model to develop a resource-efficient routing protocol.

We first present a means of routing in DTNs, specifically geared toward the properties of disaster recovery networks. The ability to efficiently and effectively route data through intermittently connected networks is of critical importance to DTNs. Many current routing protocols utilize flooding-based techniques to obtain relatively high message delivery ratios. This, however, comes at the expense of overwhelming network resources, mainly bandwidth and storage. Resource outages then lead to reduced performance in clustered areas, due to congestion, as well as energy strain on the devices. Filling all available buffer space with message replicas can hinder an application's ability to store local data. Additionally, overloading the network channel hinders one-hop protocols that do not rely on routing. Unfortunately, protocols that allow for low network resource utilization generally are not able to obtain comparable delivery ratios. In this thesis, we show that basing routing decisions on the encounter rate of a node can increase the delivery ratio. As shown in Section 5, our Encounter-Based Routing protocol (EBR) provides comparable or better message delivery ratios than current flooding-based protocols, while maintaining extremely low resource utilization. Furthermore, we present a means of securing EBR against black hole denial-of-service attacks.

There are many interesting future directions for encounter-based routing. First, we plan on evaluating EBR using probabilistic splitting rules, as described in Section 3.2. More specifically, we plan to analyze the MDR, average latency, and goodput tradeoffs when the variance of the number of replicas is increased for all nodes, as well as when the variance is non-uniform for all nodes. Following this, we plan on exploring, both mathematically and experimentally, distributions other than Gaussian.

Another future direction is exploring the effects of using a second order derivative in terms of number of encounters. Currently, EBR only considers the current rate of encounters and averages this rate using an exponentially weighted average to account for both older and newer data. If EBR used a second order derivative, it would consider the *change* in rate of encounters over time and this trend could be used to distribute an appropriate number of message replicas.

Secondly, we present a generic event- & role-based mobility paradigm used to characterize movement patterns of objects in response to environmental events, and evaluate EBR and other DTN protocols over. We specifically concentrate on applying this to disaster recovery scenarios, where current mobility models fail to realistically represent objects. To accurately characterize the movement of objects in response to one or more disaster events, we have developed a gravity-based model in which events emit forces that attract or repel objects depending on the object's role. Using simplified laws of physics, it is straightforward to calculate the velocity vector of an object, even in the presents of multiple events.

Our disaster mobility model has been fully implemented in simulation form and was used to generate ns2 mobility trace files. The resulting topology of our disaster mobility model had a higher average node density, maximum node density, variance of node density, and clustering coefficient. This is due to the grouping of nodes at or near the event horizons and near the damage radius of events. Furthermore, the partitioned value was consistently higher with our disaster mobility model, indicating the network was partitioned more often.

Our event- & role-based disaster mobility paradigm realistically captures objects' responses to disaster events. Furthermore, our simulation results show that the topological characteristics of the network drastically differ from that of Random Walk. As future work, we plan to perform studies on actual disaster scenarios to develop a rich set of role-based rules and further refine our low-level gravitational model. Furthermore, we plan to use our disaster mobility model to understand the effects it has on routing protocols such as AODV [26] and DSR [20], as well as explore security concerns with a role-based system. This will most likely lead to the development of new DTN-style disaster routing protocols specifically tuned for disaster recovery networks.

References

[1] ns2 network simulator. http://www.isi.edu/nsnam/ns/.

[2] ONE simulator. http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[3] Aruna Balasubramanian, Brian Neil Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. In *Proc. ACM SIGCOMM*, August 2007.

[4] Christian Bettstetter. Smooth is better than sharp: A random mobility model for simulation of wireless networks. In *ACM MSWiM*, 2001.

[5] Christian Bettstetter, Hannes Hartenstein, and Xavier Perez-Costa. Stochastic properties of the random waypoint mobility model. In *ACM MSWiM*, 2002.

[6] Christian Bettstetter and Christian Wagner. The spatial node distribution of the random waypoint mobility model. In *German Workshop on Mobile Ad hoc Networks (WMAN)*, 2002.

[7] John Burgess, George Bissias, Mark D. Corner, and Brian Neil Levine. Surviving attacks on disruption-tolerant networks without authentication. In *MobiHoc 07*, 2007.

[8] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, April 2006.

[9] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. In *Wireless Communications & Mobile Computing (WCMC)*, 2002.

[10] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[11] Vanessa Davies. Evaluating mobility models within an ad hoc network. Master's thesis, Colorado School of Mines, 2000.

[12] Vijay Erramilli and Mark Crovella. Forwarding in opportunistic networks with resource constraints. In *Proceedings of the Fourth ACM Workshop on Challenged Networks (CHANTS 08)*, 2008.

[13] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. Delegation forwarding. In *MobiHoc*, 2008.

[14] Stephen Farrell, Vinny Cahill, Dermot Geraghty, Ivor Humphreys, and Paul McDonald. When tcp breaks: Delay- and disruption- tolerant networking. *IEEE Internet Computing*, 10(4):72–78, 2006.

[15] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE INFOCOM*, 2001.

[16] R.A. Guerin. Channel occupancy time distribution in a cellular radio system. *IEEE Transactions on Vehicular Technology*, 1987.

[17] Dan Henriksson, Tarek F. Abdelzaher, and Raghu K. Ganti. A caching-based approach to routing in delay-tolerant networks. In *ICCCN*, 2007.

[18] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proc. ACM SIGCOMM*, 2004.

[19] Amit Jardosh, Elizabeth Belding-Royer, Kevin Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *ACM MobiCom*, 2003.

[20] D. B. Johnson and D. A. Maltz. *Mobile Computing*, chapter Dynamic source routing in ad hoc wireless networks, pages 153–181. Kluwer Academic Publishers, February 1996.

[21] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, 2002.

[22] Ben Liang and Zygmunt J. Haas. Predictive distance-based mobility management for PCS networks. In *INFOCOM*, 1999.

[23] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *MobiHoc 03*, 2003.

[24] Samuel C. Nelson, Albert F. Harris, and Robin Kravets. Event-driven, role-based mobility in disaster recovery networks. In *CHANTS*, 2007.

[25] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, and Siamak S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *ACM MobiHoc*, 2000.

[26] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *The Second IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[27] Ram Ramanathan, Richard Hansen, Prithwish Basu, Regina Rosales-Hain, and Rajesh Krishnan. Prioritized epidemic routing for opportunistic networks. In *MobiOpp 07*, 2007.

[28] Giovanni Resta and Paolo Santi. An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks. In *ACM Workshop on Principles of Mobile Computing (POMC)*, 2002.

[29] Michele Rossi, Leonardo Badia, Nicola Bui, and Michele Zorzi. On group mobility patterns and their exploitation to logically aggregate terminals in wireless networks. In *IEEE VTC*, 2005.

[30] K. Scott and S. Burleigh. Bundle Protocol Specification (Internet draft). *IRTF*, July 2006.

[31] IEEE Computer Society. Internet protocol, rfc 791, September 1981.

[32] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *(IEEE SECON)*, 2004.

[33] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 2005.

[34] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007.

[35] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, apr 2000.

[36] Brenton D. Walker, Joel K. Glenn, and T. Charles Clancy. Analysis of simple counting protocols for delay-tolerant networks. In *CHANTS*, pages 19–26, New York, NY, USA, 2007. ACM.

[37] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[38] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37, 2006.