

Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks

Delphine Nain *, Noshirwan Petigara, and Hari Balakrishnan

Massachusetts Institute of Technology

Cambridge, MA 02139

{delfin, nosh}@alum.mit.edu, hari@mit.edu

Abstract

This paper is motivated by the observation that traditional ad hoc routing protocols are not an adequate solution for *messaging applications* (e.g., e-mail) in mobile ad hoc networks. Routing in ad hoc mobile networks is challenging mainly because of node mobility—the more rapid the rate of movement, the greater the fraction of bad routes and undelivered messages. For applications that can tolerate delays beyond conventional forwarding delays, we advocate a *relay-based* approach to be used in conjunction with traditional ad hoc routing protocols. This approach takes *advantage* of node mobility to disseminate messages to mobile nodes. The result is the Mobile Relay Protocol (MRP), which integrates message routing and storage in the network; the basic idea is that if a route to a destination is unavailable, a node performs a controlled local broadcast (a relay) to its immediate neighbors. In a network with sufficient mobility—precisely the situation when conventional routes are likely to be non-existent or broken—it is quite likely that one of the relay nodes to which the packet has been relayed will encounter a node that has a valid, short (conventional) route to the eventual destination, thereby increasing the likelihood that the message will be successfully delivered. Our simulation results under a variety of node movement models demonstrate that this idea can work well for applications that prefer reliability over latency.

keywords: ad-hoc mobile networks, ad-hoc routing, messaging applications, mobility models.

* The corresponding author, Delphine Nain, is now with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30032-0280, USA, Fax: 404-894-0673.

Noshirwan Petigara is now with Kubi Software Inc., 55 Old Bedford Rd, Lincoln, MA 01773, USA. Hari Balakrishnan is with the Department of EECS at the Massachusetts Institute of Technology, Laboratory for Computer Science, 200 Technology Square, Cambridge, MA 02139, USA.

1 Introduction

Mobile ad hoc networks are useful in a number of situations where a deployed infrastructure using cellular base stations is inconvenient, expensive, or impossible. Examples of this include disaster relief or emergency response, autonomous teams of robots in a remote area, sensor networks, and military networks. In these situations, nodes in an ad hoc network move from time to time, often across a large area, while also cooperatively routing packets for other nodes.

Routing in ad hoc networks is challenging mainly because of node mobility. If nodes were mostly static, then traditional routing approaches from wired networks could be brought to bear on the problem, with occasional mobility being handled as an “uncommon” case in the same way that link failures in traditional wired networks are handled. Current approaches to mobile routing, broadly classifiable into *proactive* and *reactive (on-demand)* protocols, work toward providing a connected routing topology in the face of node mobility. Proactive protocols maintain routes via a background process (e.g., using distance-vector updates) [8], while reactive protocols discover routes on-demand when they need to communicate with another node more than one hop away [8, 7]. The performance of both types of protocols, measured in terms of the number of packets successfully delivered to the total number of packets sent (the *packet delivery ratio*), degrades as the rate of node movement increases [1].

We make the observation that not all applications in mobile ad hoc networks require the same “best-effort” semantics offered by an IP-based ad hoc network, where a packet not delivered within a small amount of time (on the order of a small number of round-trip times) is simply discarded. Messaging applications, such as e-mail, are an important class of applications in these networks; for these applications, the eventual delivery of a message is more important than any other metric of performance such as delivery latency. This paper addresses the problem of designing a message routing protocol to support such applications in an ad hoc network.

Our approach to this problem takes *advantage* of node mobility, rather than treat movement as a problem that needs to be tackled and overcome. To improve the delivery ratio of packets in the network, we use *relay nodes*; under various circumstances, when a node decides that it does not have a good route to the destination and is unlikely to obtain one, it distributes the message to one or more of its immediate neighbors, which store the message for a while. In a network with sufficient mobility—precisely the situation when conventional routes are likely to be non-existent or broken—the hope is that one of the relay nodes to which the packet has been distributed will encounter a node that has a valid route to the eventual destination, thereby increasing the likelihood that the message will be successfully delivered to the destination.

Of course, this increased reliability in terms of the packet delivery ratio comes at the expense of increased latency of message

delivery, but this is a good trade-off for messaging applications. The challenge, however, is to ensure that such relaying can be done without using up a large amount of network bandwidth or node storage. We give the general design principles of the *Message Relay Protocol*, *MRP*, as well as our particular implementation of the MRP protocol when used in conjunction with DSDV protocol (the MRP-enhanced DSDV protocol). We show using simulations that an MRP-enhanced DSDV protocol increases the average delivery ratio by 31% compared to DSDV, while using less than 1% of extra bandwidth and a relatively small amount of extra node storage space (40 packets). We evaluate MRP under a variety of movement models, node densities and message transmission rates and show that the MRP protocol scales well, demonstrating that its overall performance is not affected by node density, node movement, or message transmission rates. Our results indicate that MRP can be a viable substrate on which messaging and email-like applications in ad hoc networks can be layered. In addition, an attractive property of MRP is that it achieves these goals without assuming any knowledge of node position or any knowledge of node movement vectors.

2 Related Work

Simulations of proactive (e.g., DSDV [8]), reactive (e.g., DSR [3], TORA [7]), and hybrid (e.g., AODV [9]) ad hoc routing protocols demonstrate that the packet delivery ratio degrades when mobility is high because of route instability and frequent network partitions [1]. However, the methods proposed to overcome these problems fall under the category of timeout-based retransmissions solutions, which we argue are not the best substrate over which to build messaging applications in ad hoc networks. For example, DSR has a mechanism called “route maintenance,” by which the sender is notified that a route is no longer valid. Upon receiving such a message, it can then retransmit the packet with a cached or newly discovered route. Similarly, when DSDV detects a broken route, it stores the packet until the next route update at which time it tries to resend the packet. In contrast to these approaches, we find that the packet delivery ratio can be greatly improved by using the idea of *relaying*.

Li and Rus present routing algorithms using the relay nodes [5]. However, their assumptions are significantly different than ours. First, they assume that there is knowledge of either movement trajectories and/or node positions. Our algorithm does not assume knowledge of these. Second, their definition of a relay node is based on the the concept of mobile agents[4]. These nodes can move under their own control depending on where the message has to be delivered. In contrast, our mobile nodes cannot be controlled to move along particular paths; rather, they move randomly through the network independent of the requirements

of the relaying protocol. Zhao and Ammar also presented a proactive routing protocol for highly-partitioned networks, called Message Ferrying (MF) [10]. With MF, nodes move proactively to guarantee the delivery of messages without the knowledge of their location and the movement of all nodes in the network. These proactive algorithms work within their more controlled framework to guarantee delivery of packets by modifying the trajectories of relay nodes. We relax these constraints to show how the concept of relay nodes can increase delivery ratio even without actively constraining node movement along particular trajectories. We envision our algorithm to be useful in environments where there is no knowledge of positions and trajectories, and one where cannot easily predict or control where nodes move.

Grossglauser and Tse have explored a theoretical framework [2] where n nodes with infinite buffer move independently around the network and every node gets close to any other node for $1/n$ seconds per time slot. Within this framework, a node s gives a message addressed to node t to another randomly chosen node one hop away in the network called a “receiver.” When the receiver happens to be one hop away from the destination node t , then it gives it the message. Hence a message will only make two hops and no message will be transmitted more than twice. This ensures that the bandwidth used is minimal and that the long-term throughput per sender-receiver pair can be kept constant ($O(1)$) even as the number of nodes per unit area increases. Their other result is that a message is guaranteed to be delivered, even if its delivery time is averaged over many time slots. This result sets a theoretical bound, since it assumes a complete mixing of the trajectories so that every node can get close to another one. But the idea behind the theory is important: there is an opportunity to trade-off delay for bandwidth. This motivates our work since we believe that there is an opportunity to trade-off delay for message delivery ratio in mobile networks by using intermediary relay nodes. We also add temporary storage to the primitive operations allowed, and produce a practical protocol based on the idea of integrating storage with routing.

3 The Mobile Relay Protocol

The Mobile Relay protocol (MRP) is designed to be layered on top of an existing ad hoc routing protocol and is responsible for the forwarding, storage and delivery of relay packets. We first describe the general MRP protocol and how it integrates with different types of ad hoc routing protocols. Then, we describe in more detail our particular implementation of MRP when used in conjunction with DSDV [8].

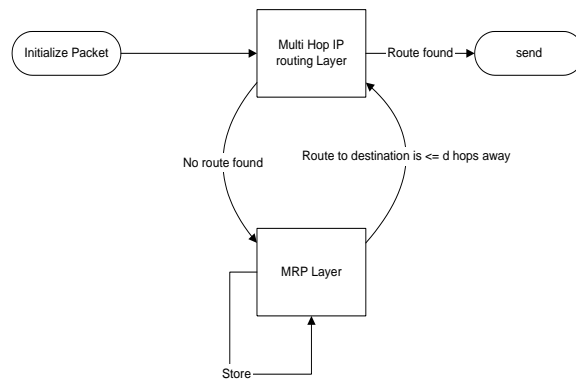


Figure 1: Integration of the MRP Protocol with a traditional ad hoc IP Routing Protocol

3.1 Overview

Figure 1 shows the forwarding of a packet by an MRP-enhanced ad hoc routing protocol. If the traditional routing step fails to find a route for the packet, the packet is handed over to the MRP layer. The MRP layer broadcasts that packet locally to the node’s immediate neighbors, who become *relay* nodes for that packet¹. Each relay node may store that packet until it discovers an IP route to the packet’s destination according to the IP routing protocol being used. When a relay node finds a route less than d IP hops away for a relay packet it has in storage, it passes it back to the traditional routing layer that forward it to its destination. d is a protocol parameter ≥ 1 ; we find in our simulations that $d = 1$ provides good performance, implying that the transition from MRP back to using an IP route can be done “late,” when one of the relay nodes comes within radio range of the intended destination.

Using MRP is optional. Whether MRP is used depends on information in the transport protocol and IP headers, which allows applications or the network operator to choose if packets can afford to be relayed if reliability matters (e.g., a messaging application packet), or whether they should not be relayed because delivery order and latency is more important than reliability (e.g., for interactive traffic, audio, etc.).

Figures 2 and 3 summarize the interaction between the MRP layer (bold boxes) and the traditional ad hoc IP routing layer (regular boxes). The rest of this section discuss how a packet transitions from traditional IP forwarding to MRP relaying, how packet relaying works, and how the protocol reverts to IP forwarding for final delivery.

¹Our assumption is that the network is not sparse, so every node is at least connected to one other node. If this is not true, then the node broadcasting could ask for a receipt from nodes who received the packet. If it gets no receipt, then it can perform another local broadcast at fixed intervals until it gets at least a certain number of receipts.

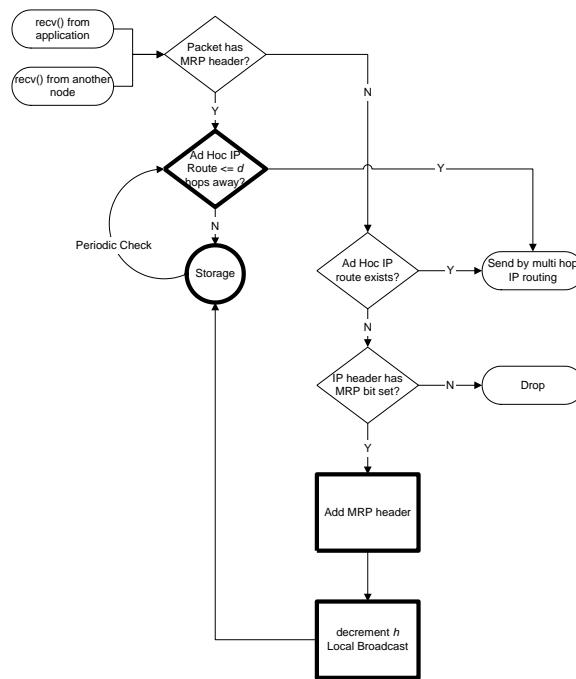


Figure 2: MRP Protocol: Forwarding

3.2 IP forwarding → MRP Relaying

With traditional IP forwarding, the network layer at a node can receive three types of packets:

1. A packet sent by an application/transport layer running on the node that needs to be forwarded to its destination.
2. A packet sent by another node that doesn't match the IP address of the receiving node, which needs to be forwarded to its destination.
3. Other protocol-specific packets, such as route updates or route discovery packets. These packets are not relayed by MRP, since they typically require low-latency delivery.

When the routing layer receives a packet of the first two types, it checks if it has an MRP header. If it does not already have an MRP header, then the node needs to check if a route exists, or can be found, or if the packet needs to be relayed via MRP.

If the packet does not already have an MRP header, the IP ad hoc routing layer uses a protocol-specific mechanism to discover a route for the packet. For example, this might be a standard forwarding table lookup (in proactive protocols) or a route discovery mechanism (in reactive protocols). If this mechanism succeeds, the packet is forwarded on as usual. If it fails, then the packet is handed over to the MRP layer.

At this stage, the packet has no MRP header. When the MRP layer receives such a packet, it first checks whether the packet may be relayed using MRP. There are several ways of doing this: one way might use a layer-4 classification with rules that decide whether MRP is suitable or not; another way might be to have the originating application or node set a bit in the IP header for the packet. If the packet is deemed unsuitable for MRP, then the packet simply loops back to the IP forwarding layer, which handles it like a regular packet for which no route is known (most protocols drop the packet and some store it for later attempts).

The interesting case for MRP is if the packet is deemed suitable for relaying using MRP. If so, an MRP header is added to the packet. The MRP header incorporates the IP address of the original sender and the final destination from the IP layer, and adds two other fields, h and d .

1. h specifies a running count of the number of remaining allowed relay hops, and decrements every time another relay operation is done. If h becomes 0, no further relays are allowed. h is initialized to a value that depends on the expected movement range of nodes in the network. Section 3.3 describes the details of how this is done.
2. d specifies how close to the eventual destination, in IP hops, might a relay node arrive before reverting from MRP to IP forwarding to have the packet delivered. For proactive routing protocols, $d \geq 1$ makes sense, since a background process will cause routes to destinations to be updated as nodes move. For reactive protocols, all we can reliably count on is $d = 1$ —when a relay node is within one hop of the destination, it will learn of this and can revert to IP forwarding to deliver the packet.

After the addition of the MRP header, the packet is locally broadcast by the node to its immediate neighbors. All nodes that receive this packet store it and go to the relaying step, described next. We emphasize that this is the *only* time in MRP that a local broadcast is done for any packet—a packet that already has an MRP header is never rebroadcast during subsequent relaying steps.

3.3 Relaying

If a packet received at the routing layer already has an MRP header, the packet is handed over to the MRP layer directly (Figure 2). This situation happens either when a node receives an MRP packet to relay from another node or an MRP packet to forward from the MRP layer directly.

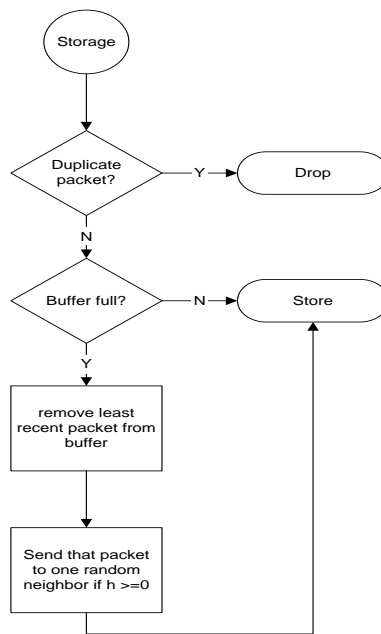


Figure 3: MRP Protocol: Storage

The MRP layer first checks with the routing layer to see if a route of less than d hops exists to forward the packet. If so, it forwards the packet and the relay MRP packet is delivered. If no valid route exists for the packet, it enters the storage phase (Figure 3), which consists of the following steps:

1. If the packet is already stored in the node's buffer, then the older version of the packet is discarded.
2. Otherwise, the storage buffer is checked. If it is not full, then the packet is stored and the h parameter in the MRP header of the packet is decremented by 1.
3. If the buffer is full, then the least recent packet is removed from the buffer and the packet is relayed to a *single* random neighbor as long as the h parameter in the MRP header of the packet is greater than 0.

In this design, a packet is eliminated from a node only when the node runs out of buffer space for packets it is currently relaying. This means that the only time a packet is eliminated from the system is if all the nodes currently relaying it run out of space. It is conceivable that a mechanism where the original sender specifies a time-to-live field in seconds, with relaying nodes eliminating the packet when this time elapses, might prove useful. Our mechanism is simpler and demand-driven, and we think that the semantics of most messaging applications are probably not worth the additional complexity of maintaining real time.

3.4 MRP Relaying → IP Forwarding

When a packet is stored, the MRP layer needs a mechanism to routinely check if a route has been found for any packets in the relay buffer. The details of this mechanism are specific to the routing protocol being used.

3.5 MRP-enhanced DSDV

In this section, we describe how MRP can enhance DSDV. Our goal is not to create a better DSDV protocol, but rather to show that MRP can help improve DSDV's packet delivery ratio for latency-tolerant messaging applications during times of high mobility by taking advantage of node movement. Our mechanisms and evaluation in the rest of this paper is for MRP/DSDV.

In DSDV, each node maintains a routing table that contains the next hop for each known (reachable) node on the network. Nodes synchronize their routing table by broadcasting periodic routing table update messages.

3.5.1 MRP Relaying with DSDV

The address resolution protocol (ARP) resolves an IP address to a hardware (MAC) counterpart and caches these bindings. For uncached bindings, most ARP implementations store the packet for which they are currently seeking ARP resolution. In ns-2 (our simulation platform), and indeed in many real systems, the ARP buffer only stores one packet, so consecutive packets for the same destination will cause all packets but one to be dropped until the ARP resolves the MAC address. To solve this problem, we paced the rate at which a relay node sends its relay packets so that there is no collision in the ARP cache. This proved to be insufficient in some cases, so we added an ARP callback where the link layer could tell the MRP layer which packets were dropped. The MRP layer then handled the packet as described before.

3.5.2 MRP → IP Forwarding in MRP/DSDV

Since a mechanism for frequent route updates is part of the DSDV protocol, it is simple for the MRP layer to check if it has a relay message stored for any node whose route has been updated and is less than or equal to d hops.

When the DSDV layer discovers a valid route to a destination it did not have previously in its routing table, it notifies the MRP layer. If the MRP layer has any relay packets stored for that destination, and the destination is d hop away, the MRP layer sends the packet back to the DSDV layer that is then responsible for forwarding it.

4 Evaluation

The overall goal of our experiments is to measure the ability of MRP-enhanced DSDV to deliver packets under a variety of movement models and compare its performance to DSDV.

We evaluate protocol performance using two metrics:

- *Packet delivery ratio*: The ratio between the number of packets originated by the application layer CBR sources and the number of packets received by the CBR sink at the final destination.
- *Packet latency*: The time taken from when the packet leaves the application layer CBR source until it reaches the CBR sink at the final destination ².

We evaluate scalability using two parameters:

- *Node Density*: We vary the number of nodes for the given simulation area.
- *Routing Overhead*: We measure the overhead created by the MRP protocol compare it to the overhead of DSDV. The overhead is calculated by measuring the total number of local broadcasts and the number duplicate relay packets recieved at the destination.

4.1 Simulation Environment

Our simulations were run on ns version 2.14b [6], which provides support for simulating both the physical aspects and protocol layers for multi-hop wireless networks. It provides implementations of DSDV and DSR, derived from the CMU Monarch code. For the MAC and physical layer of our implementation, we used the ns implementation of the 802.11b and the built-in radio model that has a radio range of 250 meters.

Our first set of simulations evaluates performance when 50 nodes are subject to different movement models. With our second set of simulations, we measure performance with varying node density (25 to 100 nodes) and use the random movement model as the underlying movement pattern of the nodes ³.

²MRP can produce duplicate packets, which may or may not reach their destination. We register a packet as “received” when the first of these duplicates arrive at their final destination and calculate its latency as the time between when the first packet was sent at the original source and the arrival of this packet at the destination. We ignore all subsequent packets that arrive at the destination, although we keep track of how many such packets arrive to measure the overhead of MRP.

³In this work, we did not experiment with very sparse networks, since we wanted to use similar topology and network density situation than in [1]. Simulation on sparse network would however be a very interesting direction for future work.

There are 150 scenario files generated per movement model. We ran both MRP enhanced DSDV and DSDV on each of these files, thereby imposing identical scenarios (node movement, node density, messaging rate) on the two protocols. Section 5 describes and explains our results in detail.

4.2 Movement Models and Topology

We investigated the performance of MRP using three different movement models, to understand how sensitive it is to these differences.

1. *Random model.* This model is the same as in some previous work on ad hoc networks [1, 3]. In this model, each node chooses a random destination point within the allowed area to move to. Once the destination is selected, the node moves toward that point with a speed distributed uniformly between 0 and a speed of 1 meter/second. Once the node reaches its destination, it pauses there for a specified *pause time*. After this time, it chooses a new destination and moves there in memoryless fashion.
2. *Soccer player model.* This model is somewhat facetiously named after the way children play soccer; here, from the current point, a node picks a random point to move to according to a probability distribution that has a decreasing probability of longer distances being picked. Specifically, the probability density of a node moving a distance r from the current distance falls off as $1/r^2$, suitably normalized so that the node does not leave the rectangular field. The speed of movement is distributed uniformly between 0 and a speed of 1 meter/second. The model does not bias the direction in which the node moves; all directions are equally likely. Each subsequent move depends only on the current location. We generated movement patterns for this model in the same way as for the random model.
3. *Homing pigeon model.* This model differs from the previous one in a subtle but important detail. Each node has a randomly chosen “home” location in the field, which never changes during the simulation. The speed of movement is distributed uniformly between 0 and a speed of 1 meter/second. Each time a node decides to move (governed by the pause time and speed of movement), it picks a random point that has a probability of being chosen that falls off as $1/r^2$, but from the home location. This models a node that ends up always moving around a home region, but also has the property that if it in fact moved far away at some time, will likely come back toward the home location (since the distribution is chosen from the home location) rather than choose another point from the current position.

In all these models, a lower pause time implies a higher degree of mobility. We generated movement patterns for 10 different pause times: 0 (i.e., continuous motion), 30, 60, 120, 300, 600, 800, 850, 895, and 900 seconds. The nodes were constrained to move inside a rectangular area of 1500x300m for 900 seconds. With a radio range of 250 meters, this area allowed for paths with a varying number of hops. In addition, during times of mobility, some network partitions occurred and some nodes were unreachable for varying durations of time.

We emphasize that these are not necessarily models of any particular real situation, but rather a way by which to obtain an understanding of the factors in any movement model that impact performance. They help us evaluate the sensitivity of the protocol to different movement patterns.

4.3 Workload

We chose to use constant bit rate (CBR) traffic sources instead of a feedback-based TCP workload in order to evaluate raw routing protocol performance without having the sources backoff in the face of loss. For the first and second set of simulations (testing performance under different movement models and node densities), we generated traffic by selecting 20 random nodes and having them send messages of 3 packets each at a rate of 4 packets per second. Each message transfer started at time uniformly distributed between 0 and 200 seconds.

4.4 MRP Parameters

We found that the value of $d = 1$ (the number of permitted hops from a relay node and the destination) is reasonable for DSDV (and better than larger values). DSDV (like other ad hoc protocols) tends to have a significant number of stale routes in its tables during high mobility [1]. By delivering a relay message only if the destination is one hop away, a relay node has a much greater chance to deliver the message before the destination node moves outside of the relay node's "reachable" zone.

The choice of an initial value for h (number of permitted relays) depends on the probability of routes breaking as well as the parameter d . If $d > 1$, then a relay node is allowed to send a stored packet to a destination even if it is more than one hop away. But if the route breaks between the relay node and the destination node (because the destination node is highly mobile for example), then one of the intermediate nodes along the route will not be able to reach the destination node anymore. In that case, it is preferable that this intermediate node act as a relay node for the packet, rather than drop it because it cannot find its destination. So a value of $h > 1$ would be preferable.

If $d = 1$, as in our experiments due to the underlying DSDV protocol, a value of $h = 1$ is sufficient since there will be no

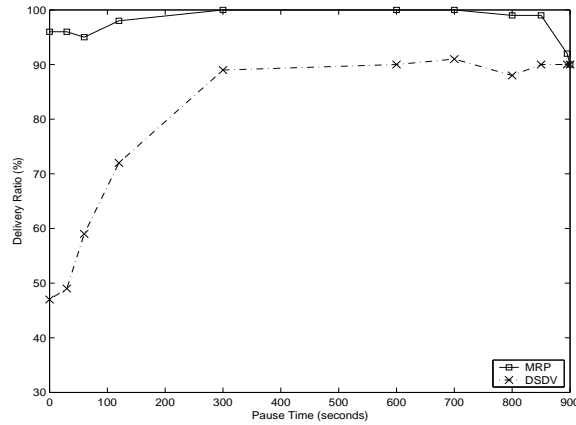


Figure 4: Packet delivery ratio v/s pause time for MRP-enhanced DSDV (MRP) and DSDV on the *Random Movement Model* for 50 nodes.

intermediate node between the relay node and the destination node, therefore no opportunity for multiple relays.

In the future, if we chose another underlying protocol that is more robust during high mobility and can detect broken routes, then it would be interesting to vary the h and d parameters to find their optimal values based on different scenarios. For example in sparse networks a high value of permitted relay steps h might be necessary to keep the delivery ratio high.

5 Simulation Results

In this section we present our results for two sets of simulations previously described. The first set evaluates the performance of MRP under different movement models (Sections 5.1 and 5.2). The second set of simulations evaluates MRP performance under different node densities (Section 5.3.1). In addition we measure the overhead of MRP in Section 5.3.2 to better understand how it scales.

5.1 Delivery ratio

5.1.1 *Random Movement Model*

Figure 4 shows the fraction of messages each protocol was able to deliver, as a function of node mobility rate (pause time) when nodes follow a random movement model. The MRP-enhanced version of DSDV significantly improves the delivery ratio of messages.

DSDV performs badly when there is high mobility in the network. When nodes continually move, DSDV is able to deliver only fewer than half the packets that are sent. This trend continues until a pause time of 60 seconds, after which point the performance of DSDV starts improving. The performance of DSDV levels off at around 90% at a pause time of 300 seconds,

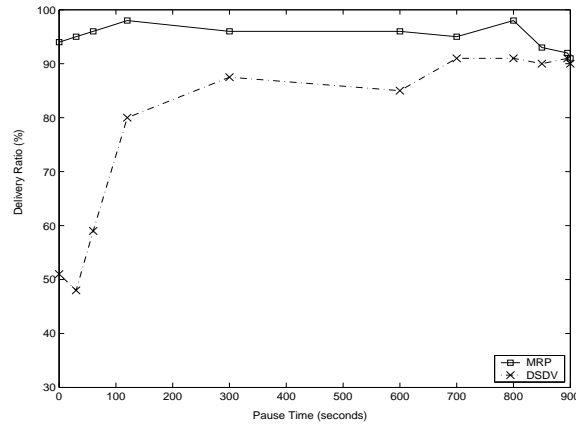


Figure 5: Packet delivery ratio v/s pause time for MRP-enhanced DSDV (MRP) and DSDV on the *Soccer Player Model* for 50 nodes. before dipping slightly in the completely static case. The behavior observed in our experiments is similar to the one observed in [1] where it is also reported that DSDV fails to converge below pause time 300.

For pause times that are between 60 and 300 seconds, we see that although the routing tables are still not able to converge, there is an increasing correlation between the DSDV routing tables and the state of the network. Starting from pause times of about 300 seconds, the routing protocol converges and the delivery ratio is around 90%. After this point, the lost packets are due to two factors. First, some packets simply cannot be delivered. This is due to transient network partitions between a source and destination. Second, routing overhead causes MAC layer collisions and some packets are dropped.

The MRP-enhanced version of DSDV is able to deliver over 94% of the packets in all cases, except the completely static case. The reason MRP-enhanced DSDV is able to perform so well under a wide range on network conditions is because it delivers packets to the destination when the underlying DSDV protocol is unable to do so.

In the case of complete mobility, MRP delivers around 94% of the sent packets. This quickly rises to 100% for less-mobile configurations. In a highly mobile network, each node comes into contact with a large number of nodes by moving to disparate areas of the network. If they encounter the destination, they deliver the message at this point. In a completely static network, the performance of MRP dips to just under 90%, where it matches the performance of DSDV as expected.

5.1.2 Soccer Player Movement Model

Figure 5 shows the fraction of messages each protocol was able to deliver when nodes follow the soccer player movement model. Nodes following this movement model have a strong tendency to move in small increments. The MRP-enhanced version of DSDV performs substantially better than DSDV when nodes move.

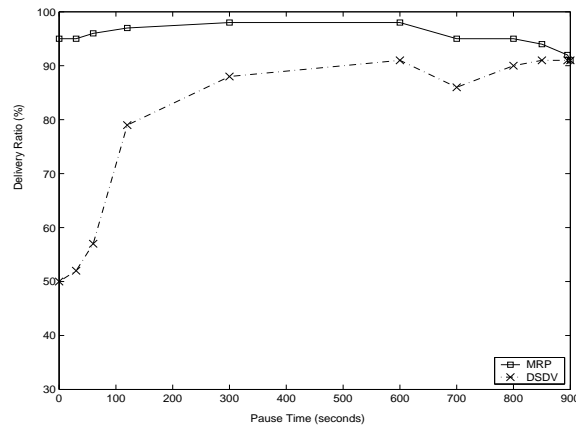


Figure 6: Packet delivery ratio v/s pause time for MRP-enhanced DSDV (MRP) and DSDV on the *Homing Pigeon Model* for 50 nodes. More interestingly, the performance of MRP on this model is only slightly lower than on the random movement model when pause times are high. This can be explained by the fact that when the pause times are high, the nodes move only once or twice during a simulation. As nodes move mainly in small increments, there is a high probability that a nodes will move relatively slightly during a simulation. Consequently, the overall topology of the network is similar to that of a static network, in which case the performance of MRP is close to the performance of the underlying DSDV. However, overall it is still a noticeable and significant improvement.

5.1.3 Homing Pigeon Movement Model

Figure 6 shows the fraction of messages each protocol was able to deliver when nodes follow the pigeon movement model. In the homing pigeon model, nodes have a strong tendency to move small distances around a random point of origin assigned to each node at the start of each simulation. Again, MRP-enhanced version of DSDV performs substantially better than DSDV except for the completely static case. The performance of MRP on this movement model is slightly lower than on the random movement model, but slightly higher than on the soccer player. Although unintuitive, it appears that nodes in this movement model have longer ranges of movements on this model, than on the soccer player model. This is explained by the fact that a node's next position is calculated from its origin regardless of its current position. Consequently, the total distance traversed by any node is higher than on the soccer player model, and each node is likely to encounter a greater number of nodes as it moves. The performance of MRP-enhanced DSDV on these movement models shows that MRP is resilient in the face of varied movement models. In each of the models, MRP is able to deliver over 95% of the packets in most cases. An interesting finding is that even when movement patters are highly localized as in the homing pigeon model, MRP does as well as in the more

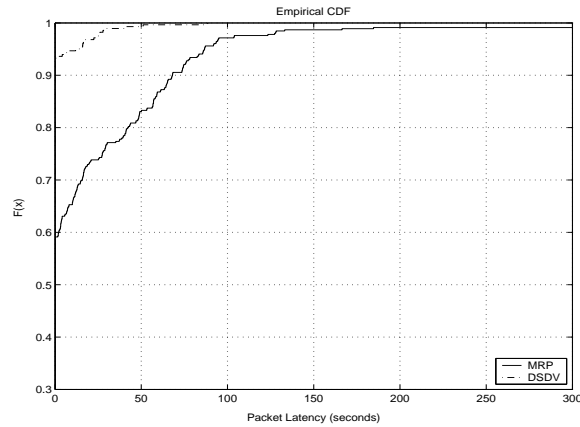


Figure 7: CDF of packet latency for DSDV and MRP-enhanced DSDV for the *Random Movement* model for a pause time of 60 seconds and 50 nodes.

unconstrained mobility case. An analytic explanation of this effect is an interesting direction for future work.

5.2 Latency

Although MRP is able to deliver a substantially higher percentage of packets than vanilla DSDV, this higher delivery ratio comes at the expense of increased latency. This is because the relay nodes carry the packets to their destination by movement rather than passing it along to the next hop in the route by radio transmission. However, we will show that in most cases the packets are delivered within acceptable limits for messaging applications.

5.2.1 *Random Movement Model*

Figure 7 shows the cumulative distribution function (CDF) of packet latencies for DSDV and MRP/DSDV for all the packets in the simulations at a pause time of 60 seconds. The shape of the CDF for 60 seconds presented above is similar to the shape of the distributions for other pause times. The only difference between these curves is that the distributions for the different pause times are centered around different latencies.

As expected, DSDV has a much tighter CDF for packet latency: over 90% of the packets are delivered within two seconds. It is important to note that this distribution is only based on the number of packets that are delivered. Undelivered packets are not included in this distribution.

For the MRP/DSDV, we see that around 60% of the packets are delivered within two seconds. These are the packets that were delivered by the underlying DSDV layer. The rest of the packets, mostly relay packets, are spread over a much wider range. By 50 seconds, 90% of the packets have been delivered, while by 350 seconds, almost all the packets have been delivered. As is

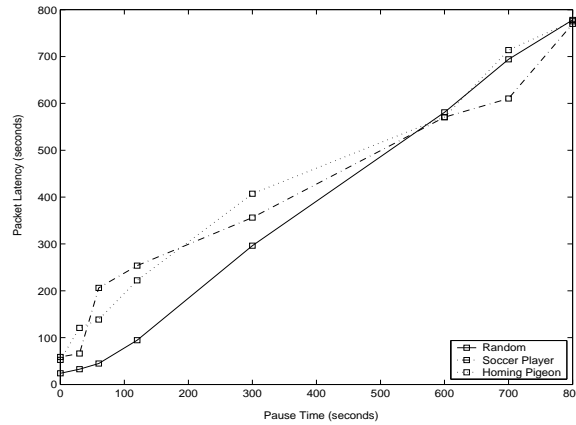


Figure 8: Mean for relay packet delivery latency on the 3 movement models for 50 nodes.

evident from this distribution, the MRP is able to deliver packets at a higher delivery ratio, though it trades this capability for increased latency of packet delivery. Still, we see that over 90% of the packets are delivered in under one minute, and all are delivered in under six minutes. For many messaging applications, where eventual delivery is more important than low latency, these delays are acceptable.

The mean latency for *relay* packet delivery times over a range of pause times is shown in Figure 8. For the random movement model, the mean delivery time increases linearly with the pause time. This is because at higher pause times the trajectories of the nodes progress slower than at low pause times.

5.2.2 Soccer Player Movement Model

The mean latency for packet delivery for the soccer player model is shown in Figure 8. At small pause times, the latencies are comparable to the random movement model. However, at intermediate pause times, the latencies are much higher. At low pause times, nodes move to many destinations during a simulation and consequently can deliver packets quickly. In contrast, at intermediate pause times when nodes move more infrequently, the tendency toward small movements dominates and nodes take longer to encounter their destinations. Hence, the latency sharply rises during these times.

5.2.3 Homing Pigeon Movement Model

The mean latency for packet delivery for the homing pigeon model (Figure 8) is very similar to the soccer player movement model. However, the delivery times for this model are slightly better. This is because the total per-node movement in this model is longer than in the soccer player model (but small a ratio compared to the random model). This leads to a similar shaped curve, with slightly better delivery models

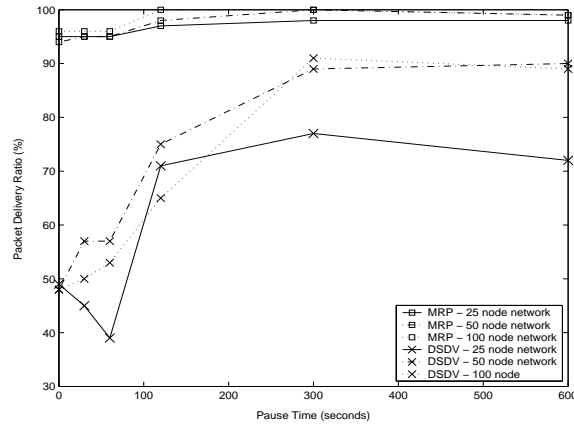


Figure 9: Comparison of delivery ratios vs pause time for networks of different densities on the *Random Movement* model.

An analysis of the mean latencies for packet delivery on the different movement models indicates that the movement patterns do not have a substantial effect on the delivery latency. Movement patterns that have less inherent mobility will have slightly higher delivery latencies for relay packets. However, delivery latencies are all within reasonable bounds for messaging applications.

5.3 Scalability

5.3.1 Node Density

For an ad hoc protocol to be robust in the face of a variety of scenarios, it is important that it not be adversely affected by an increasing density of nodes in the network. To evaluate the performance of MRP under varying density, we performed the experiments of the previous sections with different number of nodes. We ran simulations on networks of 25 nodes to 100 nodes, representing a medium to a high density of nodes.

Figure 9 shows that the delivery ratio is virtually unaffected by node density. This demonstrates that MRP works to deliver packets even if the number of relay nodes that a packet is handed to is small. This is in stark contrast to the performance of DSDV, also shown in Figure 9, whose performance spans a wide range depending on the density of the nodes. In particular for a density of 25 nodes, the packet delivery ratio considerably drops at around 60s pause time, which is also reported in [1]. We conjecture that for this pause time, the nodes might have enough time to exchange route information with their neighbors and try to send them packets. However the neighbor nodes still move fast enough to invalidate the routing tables, so a high ratio of packets are dropped. In contrast, for pause times under 60 s., nodes are so mobile that they do not attempt to send packets, so on average a smaller ratio of packets are dropped. For pause times over 60 s., nodes are less mobile so a higher ratio of sent packets reach their destination.

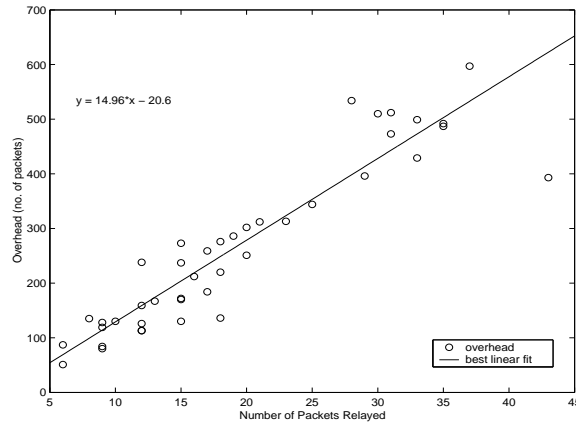


Figure 10: MRP overhead. Each point is the number of broadcasts plus the number of duplicates received over an entire simulation, plotted as a function of the number of relays performed.

It is significant that MRP is able to adaptively adjust to the shortcomings of DSDV to consistently deliver an extremely high ratio of packets over all pause times and densities.

These results are also significant as they indicate that further fine-tuning of the protocol is possible to reduce overhead. In the current implementation, the node density dictates the number of nodes that receive a relay packet during a local broadcast. Instead, one could allow nodes to probabilistically accept or reject a packet that is received to relay. Consequently, fewer nodes would relay a packet, lowering the overhead, but keeping the performance characteristics as shown above.

5.3.2 MRP Overhead

In this section we examine the overhead for the MRP protocol to determine its impact on the overall bandwidth consumed by routing overhead when MRP is paired with DSDV. Routing overhead in MRP emanates from two sources. The first is at the time of a local broadcast, which is done to distribute relay packets. As this is only one transmission, it can be taken as to be equal to the overhead of a single packet. Overhead is also caused by duplicate packet deliveries at the receiver. Figure 10 shows the MRP overhead for varying numbers of relay packets. This data was extracted from the dataset gathered from running the simulation on the random movement model. As can be seen, the plot can be approximated by a linear fit with a slope of approximately 15. The variation is the result of varying numbers of nodes in the vicinity of the node making the local broadcast. While one might be led to believe that linear relationship does not bode well for scalability, when examined within the context of the overhead of the underlying protocol, it is reasonable. In their paper, Maltz et al document the routing overhead for DSDV to be 44,000 packets on a similar network for a simulation of the same length. Consequently, when paired with DSDV as the underlying protocol, MRP constitutes less than 1% of the total routing overhead. As MRP overhead packets constitute such a

tiny portion of the total routing overhead, the bandwidth consumed by overhead can be approximated to be only dependent on the underlying protocol. This property of MRP is crucial to its success; if MRP overhead is high, it would saturate the network and reduce the number of packets that are delivered.

6 Conclusion

We motivated and presented an opportunistic relay-based approach to message routing in ad hoc networks, integrating routing and storage to improve message delivery ratios. The problem with simply running reliable TCP connections is that it is oblivious to node movement; route failures cause new connections from the messaging applications to be initiated, but these connections will end up breaking over and over as nodes in the network move.

MRP takes *advantage* of node mobility to disseminate messages to mobile nodes, rather than view mobility as a problem to be solved. Our simulation results under a variety of node movement models demonstrate that this idea can work well for applications that prefer reliability over latency, without adding any significant overhead.

A few opportunities to improve MRP present themselves. We can reduce the number of duplicate packets sent to the final destination by allowing nodes to promiscuously listen to the network traffic and restrain from delivering a relay packet if it is being delivered by another node. An alternative is for nodes to store and forward packets with a probability that is a decreasing function of the number of neighbors per node.

Another interesting feature could be to guarantee reliable message delivery. The MRP layer of the relay node could store a copy of each packet it sends, until it gets a receipt from the destination node acknowledging the delivery of that packet. If the relay node does not get a receipt, it considers the packet undelivered and continues to try to deliver it.

Increasing the message rate would impose a higher load on each relay node, and the buffer size would need to be scaled to handle that load. We determined in preliminary experiments that the required buffer size to keep a high delivery ratio only slowly grows with the message rate. In the future, it would be interesting to investigate further how to pick an appropriate buffer size based on the parameters of the system such as active nodes (number of nodes sending a message per second), message size, and node mobility.

We found that even when the probability of moving “far” from its current location is “small,” and in the homing model, $h = 1$ worked well. In general, however, this will not work; if each node moves only in a fraction of the total area, then multiple relaying hops will be needed. Determining automated ways of doing this is a topic for future work.

The performance of MRP over other protocols like DSR is also an interesting direction for future work.

Finally, an interesting direction for future work would be to run simulations on even sparser networks with a smaller node density.

References

- [1] BROCH, J., MALTZ, D. A., JOHNSON, D., CHU, Y., AND JETCHEVA, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. 4th ACM MobiCom* (Oct. 1998).
- [2] GROSSGLAUSER, M., AND TSE, D. Mobility Increases the Capacity of Ad-Hoc Wireless Networks. In *Proc. IEEE INFOCOM* (Apr. 2001).
- [3] JOHNSON, D., AND MALTZ, D. Dynamic Source Routing in Ad-Hoc Wireless Networks. In *In T. Imielinski and H. Korth, editors, Mobile Computing* (1996), Kluwer Academic Publishers, pp. 153–181.
- [4] KOTZ, D., GRAY, R., NAG, S., RUS, D., CHAWLA, S., AND CYBENO, G. Agent Tcl: Targetting the Needs of Mobile Computers. *IEEE Internet Computing* (July/August 1997).
- [5] LI, Q., AND RUS, D. Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks. In *Proc. 6th ACM MobiCom* (Aug. 2000).
- [6] ns-2 (network simulator and nam network interface). <http://www.isi.edu/nsnam/ns/>.
- [7] PARK, V. D., AND CORSON, M. S. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. IEEE INFOCOM* (Apr. 1997).
- [8] PERKINS, C., AND BHAGWAT, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. ACM SIGCOMM* (Oct. 1994), pp. 234–244.
- [9] PERKINS, C. E., AND ROYER, E. M. Ad Hoc On Demand Distance Vector (AODV) Algorithm. In *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)* (Feb. 1999).
- [10] ZHAO, W., AND AMMAR, M. Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks. In *Proc. of IEEE Workshop on Future Trends in Distributed Computing Systems* (May 2003).