

# A Reliability Layer for Ad-Hoc Wireless Sensor Network Routing

Edoardo Biagioni

Department of Information and Computer Sciences  
University of Hawaii at Mānoa  
esb@hawaii.edu

Shu Hui Chen

Department of Information and Computer Sciences  
University of Hawaii at Mānoa  
shuc@hawaii.edu

## A Reliability Layer for Ad-Hoc Wireless Sensor Network Routing

### ABSTRACT

*We have been studying communication in wireless ad-hoc sensor networks. One of the authors has designed the Multipath On-demand Routing (MOR) protocol [4], which makes use of all possible paths to a given destination. Each node in MOR has, wherever possible, a choice of next hops for a given destination. We have designed and implemented a MOR reliability layer to take advantage of this. The basic strategy is to use a different node on each retransmission, and to keep track of which transmissions are successful. The main benefit is that a packet is likely to be delivered even if a given neighbor is temporarily unavailable, thus improving the delivery ratio or decreasing the number of end-to-end retransmissions. A further benefit is that nonresponsive nodes are removed from the routing table after a number of consecutive failures.*

*In a sensor network transmission may be unreliable for a number of reasons, but particularly when congestion results in collisions. Without our reliability layer, collisions would cause packet loss and route loss. If the transport protocol is reliable, this results in end-to-end retransmission, which requires additional energy. We simulated transmission in congested conditions in different realistic sensor networks and compared MOR to available protocols. In our tests, the reliability layer helped MOR deliver data faster and using less energy than the other protocols.*

### I. INTRODUCTION AND BACKGROUND

The Pods project at the University of Hawaii at Mānoa has been developing technology to build networks of wireless sensors. To date, networks of Pods have been used to monitor natural environments and particularly the environment of endangered plant species [1]. Survivability for such a network is essential. Network nodes are

susceptible to radio interference both from within the network (congestion) and from outside. This interference causes packet loss which affects performance and, in turn, energy efficiency – the longer a node has to stay in active mode to transmit data, the quicker its energy is depleted, and the more times a node has to retransmit a given packet, the quicker its energy is depleted. In this paper we describe strategies incorporated into the MOR protocol to increase the reliability of data transmission, improving the robustness of MOR against interference.

These sensor networks are ad-hoc in that each node can be placed anywhere within the network and, without further configuration, is designed to forward the data from other nodes until this data reaches a base station. Our sensor networks also support more general communication between any pair of nodes, which can be useful for computation within the network, for example to decide whether data is useful and must be transmitted to a base station, or is redundant and need not be sent to the base station.

Such sensor networks have both similarities to mobile ad-hoc wireless networks (MANETs) and substantial differences. Like MANETs, such networks use established wireless protocols and are designed to carry conventional higher-level traffic such as TCP and UDP. Like MANETs, sensor networks may suffer from congestion, especially if data collection is periodic and all sensors are likely to send data at approximately the same time. Unlike MANETs, most nodes in a sensor network are typically stationary, so that loss of connectivity to a neighbor is due chiefly to factors such as RF interference, congestion, or temporary or permanent battery depletion – in other words, although some nodes may be mobile and mobility must be supported, most nodes are likely to move slowly if at all, and efficient support of mobility may not be a priority. Unlike MANETs, sensor networks must be designed to scale up to hundreds or thousands of nodes, with network diameter (maximum hop count between any pair of nodes) in the tens or hundreds of nodes. Also unlike MANETs, it is often the case that most nodes in a sensor network will

have a limited number of neighbors [2]. Finally, although nodes in a MANET may well have energy constraints [9], nodes in a sensor network may be left unattended for months and years. If nodes are powered by batteries, effective energy conservation determines the lifetime of the network, whereas if node batteries can be recharged, energy conservation determines the cost and visibility of the network. In either case, nodes in a sensor network must be extremely energy efficient, and may be designed to shut down periodically to save energy. Fast delivery of data is therefore one component of energy efficiency, since it allows the network to be put into low-power mode sooner while delivering a given amount of data.

#### A. The MOR Protocol

Shu Chen has designed the Multipath On-Demand Routing (MOR) protocol [4], an efficient protocol for routing and data delivery in a wireless ad-hoc network.

MOR in some ways resembles other wireless ad-hoc routing and data delivery protocols described in the next section, but has special characteristics that make it more energy efficient. The protocol is described fully by Chen [4], and will only be summarized here, with special focus on the characteristics that contribute to the protocol's energy efficiency.

Like other such protocols, MOR relies on flooding broadcasts, also known as network floods, to discover a route to a destination node when such a route is needed and no such route exists. Each node receiving such a broadcast forwards it to its neighbors. As is conventional in network floods, each node checks its own routing tables before forwarding the packet, only forwarding the packet if this is a new request (as determined by a per-sender sequence number) or a better route than has been seen before.

When the intended destination of the broadcast receives the route request, it responds, and the response is forwarded, using unicasts rather than broadcasts, along one of the reverse route back to the originator – along the *backtrace*.

At the end of this process, each node in the network has a route back to the originator, and some nodes in the network have routes to the intended destination.

Such network floods are relatively expensive in large networks, requiring each node to retransmit each such packet at least once. This has been studied in detail by Feeney [6]. If each node in a network of  $N$  nodes floods the entire network, this requires sending  $O(N^2)$  packets. The energy and time required are both  $O(N)$  per node. The overall energy is therefore  $O(N^2)$  for the entire network. The overall time is  $O(N)$  in the best case, a network where each node has at most a constant number  $d$  of neighbors, and  $O(N^2)$  in the worst-case, a network where

every node is in reach of every other node. Good energy efficiency and good performance are therefore obtained in part by minimizing the number of network floods required.

The first optimization introduced by MOR is therefore to have the base station do a network flood when the network first comes up, and periodically thereafter. This uses a single network flood to establish and maintain routes from every node in the network to the base station, in contrast to the number of network floods that might be required were every node to flood the network requesting a route to the base station.

The next optimization introduced by MOR is to allow data packets to build routes. Once a node has a route to the base station, for example, it can begin to transmit data. Each node forwarding this data also adds a route back to the sender (along the backtrace), and thus by the time the data has reached its destination, the destination has a route that can be used for replies or acknowledgments.

Another optimization used by MOR is to use unicast transmission whenever possible. Although network floods are most efficient when using broadcast, data transmission using broadcast requires each recipient in range to listen to the entire data packet and process it in software. Unicast transmission, in contrast, allows the wireless hardware to be shut off in every node other than the next hop, conserving energy. When MOR is run over 802.11, as in our tests, unicast transmission has the added benefit of utilizing collision avoidance (CA), which is not available for broadcast, and which provides better utilization of the available medium. While some other protocols use this optimization, many don't, leading to inefficiency.

A final optimization introduced by MOR is have each node store and use as many equal-cost routes to the destination as possible, hence the Multi-Path nature of MOR. This optimization will be covered in considerable detail below, in the section of this paper describing the MOR Reliability Layer. For now we simply point out that using multiple paths wherever possible helps distribute the load evenly, utilizing the available network bandwidth more efficiently and also balancing energy usage so that energy usage is as uniform as possible across nodes.

#### B. Related Protocols

Many other protocols have been developed for routing and data transport on wireless ad-hoc networks. These have generally been developed to work well on mobile networks, or MANETs, and usually on the scenarios originally published by Broch and others [3]. These scenarios have varying mobility of between 10 and 30 nodes, each with a radio range (communication radius) of approximately 250m in a rectangular space approximately 1500m by 300m. The maximum network diameter

recorded in different simulations was 8 hops, though a typical communication only required 2.6 hops [3]. This is very different from the sensor network scenarios described above, which feature low or no mobility, large network diameters, and typically (but not always) low numbers of neighbors. For example, the DSR (Dynamic Source Routing [8]) protocol is a source routing protocol which, as normally configured, can reach destinations that are at most 16 hops away. This is sufficient in the MANET scenario, but is inadequate for many sensor networks. In our comparisons, we have modified DSR to allow for larger network diameters, which however increases the per-packet protocol overhead.

Another protocol described by Broch and others [3] is AODV, the Ad-Hoc On-Demand Distance Vector protocol. On-demand route construction is common in wireless sensor networks, and is done by MOR as well as many other protocols – unlike a regular wireline routing protocol, routes are built and maintained only when a node has data to send to a specific destination.

DSR and AODV have been carefully studied in the literature and therefore we compare MOR to them in Section III.

Both DSR and AODV are single-path protocols, in which each node has at most one route to a given destination. Protocols have also been designed to support multiple routes to a given destination. One of the earlier of these is MDSR [11], which specifies that each intermediate node on the “main” route between a source and a destination should have a single alternate route to the destination which is disjoint from the main route.

Of particular interest among multipath protocols is the Ad-Hoc On-Demand Multipath Distance Vector protocol, or AOMDV [10]. This is a modification of AODV that allows multiple disjoint equal-cost paths to be used between any source and any destination.

Unlike AOMDV (or MDSR), the paths in MOR need not be disjoint, as shown in Figure 1.

While link-disjoint paths have many of the desirable properties of non-disjoint paths such as used by MOR, they have an important drawback when used in networks where individual nodes and links may be unreliable: the decision of which path to use must be made by the original sender, and this decision cannot be modified en route by nodes that may find that a given link is temporarily unavailable. Each node in a MOR network, in contrast, can make a local decision of where to forward each packet. This is further discussed in Section II.

Assume, in a network such as the one in Figure 1, that the upper right node suffers from interference from other senders, perhaps due to its proximity to a source of RF noise or to other parts of the network. Packets sent to this node would frequently be lost or, when using

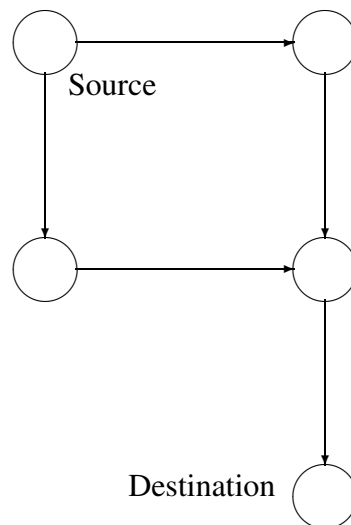


Fig. 1. While MOR would use all the paths shown, AOMDV must limit itself to one of the two ways of reaching the second node, since all paths must be link-disjoint.

IEEE 802.11 (in ad-hoc unicast mode) for transport, such packets might not even be sent if the node is participating in Collision Avoidance, having received a suitable Request to Send or Clear to Send from one of its neighbors. In such a situation it would be preferable for packets to be sent along the lower left path instead. A protocol such as AOMDV would build a single route, with a 50% probability of using the upper right node. This route would then become invalid when the node becomes unavailable. MOR, in contrast, would construct all possible routes, at no more cost than is required to build the single route, and use them interchangeably. When the upper right node becomes unavailable, MOR would automatically switch to only using the lower left route.

### C. The Need for Reliability

The above scenario is representative of the data losses that can occur in a multi-hop sensor network.

It may seem that such packet loss does little harm. After all, the protocol detects the route failure and builds the alternate route within a brief period of time. The higher-level protocol may adapt to the packet loss, either by retransmission, forward error correction, or graceful degradation of service.

Unfortunately, such adaptation usually leads to severe performance penalties. We look specifically at TCP, which would be used in a sensor network to reliably deliver data. If a packet is lost in TCP, once a timer expires TCP takes a number of actions (among many excellent references, we mention chapter 6 of Peterson and Davie [12]) of which retransmission is just one. Other actions include drastically reducing the window size, which substantially

reduces throughput, and doubling the length of the timer, so that should the retransmitted packet also be lost, the timeout will be twice as long (and the window will be further reduced).

These actions are appropriately designed to prevent TCP from continuously causing congestion, and in particular from causing a congestion collapse [7] in an Internet. Such a collapse would be caused by a number of nodes retransmitting at a fixed rate for long periods of time, injecting ever more data into an already congested Internet. However, these features also significantly slow down data transmission if the packet loss or packet error rate is significant.

While packet loss on an Internet is a symptom of congestion on a given route, a multipath protocol is designed to use multiple routes whenever possible. Even if one such route is congested, other routes may not be. Therefore, the loss of one or a few packets may fail to indicate the likelihood of losing additional packets. That is, unlike an Internet, a multipath routing network may experience congestion on just a subset of the possible routes between any two destinations, and reacting to this loss as if the entire set of routes was congested leads to much lower throughput for data.

To avoid, whenever possible, triggering TCP's congestion control mechanism, MOR incorporates a retransmission mechanism which is described in the next section.

## II. THE MOR RELIABILITY LAYER

The reliability layer in MOR is conceptually distinct from the main protocol as described above. Specifically, MOR could work without the reliability layer (though the performance may suffer), but the MOR reliability layer is specifically tailored to the needs of the MOR protocol. The principles we describe are more generally applicable to other protocols as well.

### A. Overview

The MOR reliability layer implements link-layer reliability. The goal of this reliability is to improve the chances of end-to-end packet delivery. TCP tries very hard to guarantee packet reliability, retransmitting packets end-to-end many times and failing only if retransmission fails over and over again. In contrast, the MOR reliability layer tries to retransmit a packet a limited number of times, and on a hop-by-hop basis rather than end-to-end. That is, the failure to transmit along a single link is detected, and the MOR node attempts to retransmit, if possible retransmitting to a different next-hop node. Such link-layer retransmission can use information, such as the link-layer acknowledgement delivered by protocols such as 802.11 and the link-layer (or network-layer) information about which next hops are valid for a given destination.

To implement this for MOR in both *ns-2* [5] and Linux, one of the authors (Chen) had to modify the underlying software to return the acknowledgement information to the MOR router. In our implementation MOR runs over 802.11, and acknowledgment information is available to the simulator and to the network device driver, but (without our modification) is not available to higher layers.

Transmitting to other nodes has the benefit of routing around any temporary congestion or node unavailability, providing much faster response than is available to the sender running TCP. In addition, the node that failed to acknowledge the packet can be placed on probation, and the probationary status can be used in future decisions about which route to take. Eventually, if a node fails to acknowledge a sufficient number of packets, the route through that node can be deleted.

The description of MOR in Section I-A pointed out that in many cases data packets can be used to perform routing functions, specifically, building return routes to the sender. In this section we see that data packets can be used to replace commonly used "Hello" packets, with routes only being deleted when data is actually being transmitted and the next hop node fails to acknowledge the data.

True "Hello" packets are particularly inconvenient in a sensor network where nodes go down periodically. Unless the transmission of such packets is synchronized with the node wake-up time, routes may time out unnecessarily, requiring additional overhead to rebuild routes that should never have expired in the first place.

### B. Definition of the MOR Reliability Layer

Formally, the MOR reliability layer is a link-layer retransmission protocol. This protocol is currently layered over 802.11 in ad-hoc mode, and is only used for unicast traffic.

When MOR wishes to forward a unicast data packet, it consults its routing table. All routes through a given node to a given destination have the same cost, and each has a distinct next hop. The node selects the least recently used of these routes<sup>1</sup> and forwards the packet to the corresponding next hop.

The reliability layer monitors the transmission for failure. The 802.11 layer itself will retransmit a unicast packet up to a fixed number of times (3 by default), and fail if the packet was not successfully acknowledged by the intended next hop. In case of such a failure, the reliability layer searches the MOR routing table for another route to the same destination. All such routes are tried before the packet is dropped.

If a next hop node fails to support data transmission three times in a row, all routes through that next hop node are deleted.

<sup>1</sup>Although algorithms other than LRU are possible.

receive packet from L, destination is D  
loop:

```

H = next hop in Least Recently
    Used (LRU) route to D
if there is no route to D,
    send a No Route to L, exit loop

/* found a route */
update LRU time for route to D via H

send packet to H
if packet successfully acknowledged,
    clear failure count for this route
    exit loop

/* packet failed */
increase failure count for this route
if failure count > 3,
    remove all routes through H
    repeat loop
    
```

Fig. 2. Pseudo-code for the reliability layer.

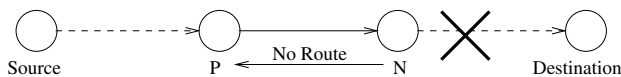


Fig. 3. Route Cancellation.

Figure 2 shows pseudo-code which summarizes the work of the reliability layer.

In MOR, a node that is unavailable for an extended period of time will result in the removal of that node from its neighbors' routing tables. This is a feature of the protocol, though it does mean that once the fault is over, the node will not be automatically integrated back into its neighbors' routing tables. Events that will cause the neighbor to again be listed include new broadcast route requests, or the node itself forwarding a packet.

### C. Active Route Management

When a node N deletes its last route to a given destination D, there is no point in upstream nodes continuing to use N as a next hop for data to D. To communicate this situation to the upstream nodes, a node N which is dropping a packet due to a lack of routes will send a No Route message to the previous hop P, the node that the packet was received from, as shown in Figure 3. P can then remove the route which has N as its next hop. If P also has alternative routes to D, it can send a message to N that a route to D is available using P as N's next hop.

### D. Benefits of the Reliability Layer and Active Route Management

The MOR reliability layer leverages the multipath nature of MOR to route around temporary congestion in ways that are not available to protocols that maintain multiple link-disjoint routes or that do not maintain multiple routes.

The benefits of active route management combine some of the benefits of connection-oriented communications with the benefits of connectionless communication. There are no connections in MOR, and a route expiration will eventually lead to that route being re-established if it is still needed, similar to the multicast request protocols in IP multicast. Active route cancellation and re-establishment on the other hand provides a mechanism to re-establish a route that has been invalidated using only local information, without requiring any (relatively energy intensive) network floods. If no route can be re-established from local information, eventually the No Route message will reach the original sender which can then use a network flood to re-establish a route, or to determine that the node is no longer reachable on this network.

The combination of these two mechanisms provides increased reliability at very low cost in energy. Packets are delivered when any route is available from the node currently holding the packet. Routes are generally preserved in the face of transient problems including localized congestion, routes are re-established quickly and cheaply in cases where that is possible, and network floods are only used as a last resort when the cheaper, more local mechanisms fail. Link-layer retransmission can be much faster than end-to-end retransmission (e.g. [13]), so the TCP congestion control mechanism is rarely invoked, and throughput is higher. When the entire network is congested, on the other hand, the reliability layer will fail to transmit packets and the TCP congestion control mechanism can take over, correctly reacting to congestion by lowering the overall data rate.

As pointed out by a reviewer of an earlier draft of this paper, routing tables in MOR may be larger than routing tables for protocols which only use a single path for each route, because multiple routes are kept to each destination. In general, with  $N$  nodes in the network, in the worst case each node would have to store at most  $O(N)$  additional routes, since each route can go through at most  $N$  more nodes than it would for a single-path routing protocol. In the worst case, each node might have up to  $O(N)$  next hops for each route, though for a network where each node has at most  $d$  neighbors, this cost is limited to  $O(d)$ . The overall increase in memory consumption per node compared to single path routing tables is  $O(N^2)$  for a network with unbounded degree, and  $O(d \times N)$  for a

network of degree  $d$ . In practice, these worst cases are rarely achieved with realistic networks. Even with many thousands of nodes, the number of routing entries tends to be small compared to the sizes of memories found in most embedded computers, and the trend, of course, is towards larger memories.

### III. SIMULATION RESULTS

The source code for DSR and AODV on the *ns-2* simulator is available and has been tested and analyzed by other researchers. The two protocols are also widely used as benchmarks in the published literature. It is for this reason that we decided to compare the performance of MOR to the performance of DSR and AODV. The only modification we made to these protocols is to increase the number of routes in DSR routing headers so that DSR can route on the relatively large-diameter networks that we use to simulate sensor networks.

In our application, one of the plant species we wish to study is found in an area that is almost 2km away from the nearest Internet connection. We used this scenario in designing our simulations. 100 sensor nodes are placed at random within a circular area: distance from the center and angle uniformly chosen at random, a distribution which gives slightly higher density near the center, which resembles many real-world deployments including our own. Relay nodes are likewise deployed at random within circles along a line leading back from the center of the circle to the base station, as shown in Figure 4. These constraints resemble an air drop of sensor nodes to cover the area of interest and connect it back to the base station.

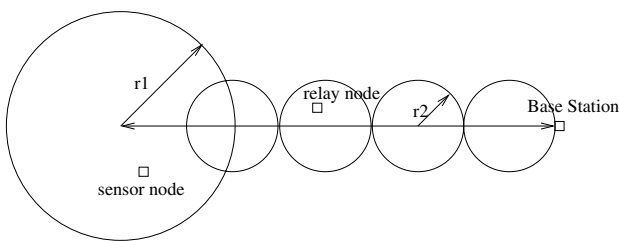


Fig. 4. Scenario design.

We have simulated using two basic scenarios:

- a high density scenario with the circular area having a radius of 400m and the line being 1200m from the center of the circle to the base station with 6 points around which the relay nodes are deployed, and
- a low density scenario with the circular area having a radius of 600m and the line being 1800m from the center of the circle to the base station, with the relay nodes distributed around 8 points along the line.

In the low-density scenario, it was necessary to scatter additional nodes at random over the observation area to

make the network connected – these additional nodes are relay nodes, which forward but do not generate any data. Nodes in the high density scenario have a mean of  $38 \pm 20$  neighbors and the network has a diameter of 7 or 8 hops, and in the low density scenario a mean of  $21 \pm 14$  neighbors and a diameter of 11 to 13 hops. 13 hops is still less than the DSR default maximum of 16 hops, but since protocols do not always find the shortest route in the presence of congestion<sup>2</sup>, the maximum route length was increased to 24 hops.

Twenty-five topologies were generated for each of the high density and low density scenarios, though one of the low density topologies remained disconnected even after the second pass, and could not be used.

To properly simulate a sensor network, we assumed that sensors produced data at a constant rate and used TCP to send this data to the base station. 90 of the 100 nodes in the observation area produce small amounts of data – 7000 bytes each – and the other 10 nodes are assumed to have cameras or other equipment producing data at high rates, and are assumed to transmit 444KB of data each. The total task requires sending 4.94MB of data to the base station<sup>3</sup>. As mentioned, nodes along the line connecting back to the base station were assumed to only function as relay nodes, and do not generate data, and the same is true for nodes placed in the second pass in the low-density scenarios.

Because TCP is used, a window size must be selected, and we used a large size of 200 packets or approximately 280,000 bytes – in our experiments, TCP never reached this maximum window, presumably because packet loss caused the congestion window to remain smaller than the selected window.

Energy usage is in arbitrary *energy units*, or *eu*. We assumed that a node consumes 0.03 eu/second when idle, 0.3 eu/second while receiving, and 0.6 eu/second while transmitting. These values are consistent with values measured on real-world wireless devices [9].

The overall results of our simulations are shown in Table I for the high density topologies, and Table II for the low density topologies. Numbers shown are the mean and the standard deviation of all measurements.

As can be seen, MOR outperforms the other protocols on every measure in both scenarios. It takes less time to

<sup>2</sup>MOR and AODV are better than DSR at finding optimal routes. MOR is especially good because each node uses the information in forwarded packets to improve its own routing tables back to the sender of the packet.

<sup>3</sup>Many authors have compared the performance of networks with nodes sending at constant bit rates (CBR traffic). For sensor networks, we believe it is more reasonable to evaluate protocols for the time to complete transmission of a fixed amount of data. TCP will retransmit if data is lost, and retransmission takes time (and energy and bandwidth), so the time to task completion is inversely related to the performance of the network.

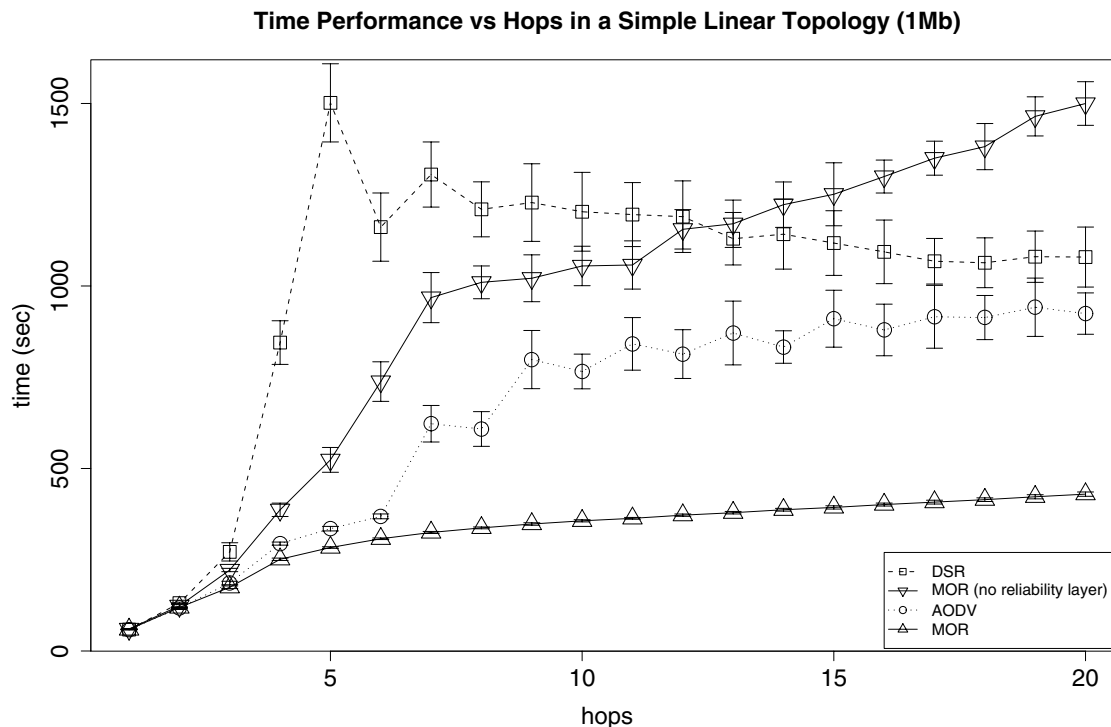


Fig. 5. Reliability Layer Contribution to Throughput.

TABLE I

PERFORMANCE OF MOR, DSR, AND AODV ON THE SIMULATION OF HIGH DENSITY TOPOLOGIES.

	MOR		DSR		AODV	
	mean	stdev	mean	stdev	mean	stdev
time (s)	401	33	2866	1081	856	80
energy	11128	890	23068	9785	14935	542
max energy	117	8	217	134	140	5

TABLE II

PERFORMANCE OF MOR, DSR, AND AODV ON THE SIMULATION OF LOW DENSITY TOPOLOGIES.

	MOR	DSR	AODV
time (s)	495 ± 41	4341 ± 1371	1420 ± 330
energy	11021 ± 801	28096 ± 5555	16849 ± 1894
max energy	123 ± 7	272 ± 49	169 ± 17

complete the task, meaning nodes can shut down sooner and conserve more energy. The energy expended by the network is less for MOR than for either of the other protocols, allowing a battery-powered network to stay up longer and an externally-powered network to require smaller, less expensive, or less visible sources of power.

Finally, the maximum energy expended by any given node is lower for MOR than for the other protocols.

More detailed studies [4] show that the difference in energy consumption is less, but still substantial, if the energy measurements only count the energy needed to transmit and receive packets, neglecting the idle energy which is necessarily higher for the protocols which require more time to transmit the same amount of data.

To analyze the effect of the reliability layer on the performance of MOR, we have run a simple test in which a number of nodes are placed in a straight line, each node within range of the node before it and the node after it. In this scenario, there is exactly one path from the sender, at one end of the line, and the receiver, at the opposite end. We have run this simulation for lines with 2 to 21 nodes, and graphed the length of time required to complete transmission of 4.94MB using TCP. We have done this for AODV and DSR, and also for MOR both with and without the reliability layer, so that when links fail, the corresponding routes are discarded immediately without probabon. The results are shown in Figure 5, where error bars show the standard deviation of the measurements.

As can be seen, the performance of DSR is generally the worst (the most time is needed to transfer the data), of MOR with the reliability layer is the best, with AODV

somewhere in between. Both DSR and AODV have much greater variability than MOR with the reliability layer<sup>4</sup>. The performance of MOR without the reliability layer, on the other hand, is comparable to the performance of DSR, becoming even worse beyond about 12 hops. In this very simple test with no possibility of using multiple paths, the reliability layer therefore is the crucial component which makes MOR significantly more efficient than AODV or DSR. The reliability layer's use of retransmission allows links to be kept alive when they are only temporarily unavailable or congested, avoiding both TCP retransmission and a new broadcast to establish routes. Even though in such a one-dimensional scenario a broadcast is not substantially more expensive than any other data transmission, first re-establishing the route and then retransmitting the packet end-to-end takes additional time, energy, and bandwidth compared to simply retransmitting a packet on the local link.

#### IV. CONCLUSIONS

We believe that the careful design of MOR described in this paper, and particularly the design of the reliability layer and active route management, contribute to the positive throughput and energy efficiency of MOR. By using local retransmission over locally available alternative paths to avoid end-to-end retransmission, the number of packet transmissions over each hop is reduced, leading to greater energy efficiency. Discarding a packet requires discarding all the energy that has been used so far to transmit that packet, and discarding a route requires discarding all the energy that has been used to build that route. MOR, through its reliability layer and active route management, takes reasonable steps to avoid discarding packets and to rebuild routes using local information, including both the routing tables of neighboring nodes and the reverse routes of data being forwarded. This minimizes the number of nodes that have to retransmit packets and the number of global network floods needed to establish routes.

The reliability layer leverages the multi-path nature of MOR to avoid dropping data packets, which leads to fewer TCP retransmission, higher throughput, and lower energy consumption. The reliability layer and active route management quickly eliminate routes that are experiencing congestion without necessarily discarding the data, and in most cases without having to rebuild routes from scratch. Using the reverse routes of data being forwarded effectively re-establishes routes once transient congestion is over, without the overhead of network floods. In addition to all these benefits, the multi-path nature of MOR

<sup>4</sup>We have not been able to explain this variability, beyond noting that the performance in this simple benchmark varies dramatically for different versions of AODV and DSR.

provides basic load balancing, avoiding unnecessarily depleting the energy of any one node.

We believe that these principles can be applied to other protocols, for both wireless and wired networks. At the very least our experience points out the benefit of computing and using multiple paths whenever possible. Even in a sensor network with many thousands of nodes, the amount of information exchanged is generally proportional to the number of neighbors of a node (as well as to the number of destinations, as is true for any routing table) rather than to the size of the entire network, and therefore remains manageable, leading to substantial benefit at little cost.

We have seen that these principles give improved throughput, performance, and energy efficiency and fairness, characteristics required for the long-term survivability of a wireless ad-hoc network.

#### REFERENCES

- [1] Edoardo Biagioni and Kent Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(3), 2002. Available from <http://www.ics.hawaii.edu/~esb/prof/pub/ijhpc02.html>.
- [2] Edoardo Biagioni and Galen Sasaki. Wireless sensor placement for reliable and efficient data collection. In *Hawaii International Conference on Systems Sciences*, Waikoloa, Hawaii, Jan 2003.
- [3] Josh Broch, David A. Maltz, David B. Johnson, and Yih-Chun Hu and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.
- [4] Shu Hui Chen. Multipath on-demand routing in sensor network topologies. Master's thesis, Department of Information and Computer Sciences, University of Hawaii at Mānoa, May 2003.
- [5] Kevin Fall and Kannan Varadhan. *The ns Manual*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2002. Available from <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [6] Laura Marie Feeney. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications*, 6(3):239–249, 2001.
- [7] Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88*. ACM, Aug 1998.
- [8] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [9] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh-Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.
- [10] Mahesh Marina and Samir Das. Ad hoc on-demand multipath distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3), 2002.
- [11] A. Nasipuri and S. Das. Demand multipath routing for mobile ad hoc networks. In *Proc. 8th Annual IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, 1999.
- [12] Larry L. Peterson and Bruce S. Davie. *Computer Networks – A Systems Approach*. Morgan Kaufmann, 2nd edition, 2000.
- [13] R. Yavatkar and N. Bhagwat. Improving end-to-end performance of tcp over mobile internetworks. In *Workshop on Mobile Computing Systems and Applications*, pages 146–152, Dec 1994.