This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

A Dynamic Route Optimization Mechanism for AODV in MANETs

Zeki Bilgin
Computer Science Department
Graduate Center, City University of New York (CUNY)
Email: zbilgin@gc.cuny.edu

Bilal Khan
Department of Mathematics & Computer Science
John Jay College, CUNY
Email: bkhan@jjay.cuny.edu

***Abstract*—In reactive routing protocols, active routes for connections retain their topological structure even when node movements over time make the route sub-optimal (in terms of path length). In AODV, for example, a connection route is recomputed only when one of its constituent links suffers catastrophic failure– at which point repair is initiated, often requiring global route discovery and introducing significant control traffic overhead. In this paper, we propose extensions to AODV which perform continuous route optimization. The objective of our proposed extension is to ensure connection path lengths are topologically efficient in spite of node mobility. Our experiments indicate that in many typical operational regimes, this objective is achieved successfully by our proposed scheme. The proposed mechanism has been implemented and tested in ns2 as an extension to AODV.**

I. Introduction

Issues of *routing* in wireless networking have been well-studied in the last two decades. Many routing protocols have been proposed in accordance with different network structures, mobility scenarios and types of applications. While developing a routing algorithm for MANETs, one of the most challenging peculiarities which developers must consider is the behavior of the proposed algorithm in the presence of node movement. Mobility has a potential to result in dynamic changes to network topology, making the task of routing algorithms more difficult. Two major challenges introduced by mobility, are the occurrence of *disconnection* and *suboptimality* in connection routes. More precisely, if the network topology is dynamically changing due to node movement, at least one of the following two events eventually occurs: (i) the connection breaks because one of its constituent links becomes disconnected, or (ii) the route becomes sub-optimal in terms of path length (i.e. number of hops) due to changes in the network topology. Reactive routing protocols generally take care of the former event by initiating local recovery or restarting the route discovery process. The latter event is not often considered, and accordingly is the focus of this paper. An example in which route suboptimalty occurs (over time) is depicted in Figure 1.

In reactive routing protocols, connection routes between source-destination pairs are constructed on-demand at the very outset, when their necessity becomes known due to a request by the source node. At the very beginning of a connection, the number of hops that the route takes tends to be very close to the number of hops on a min-hop path (e.g. one that would be calculated by Dijkstra's algorithm). As time passes, nodes move, yet as long as no link breaks, the connection retains its topological structure, and in doing so may become require significantly more hops than the instantaneous min-hop path between source and destination.

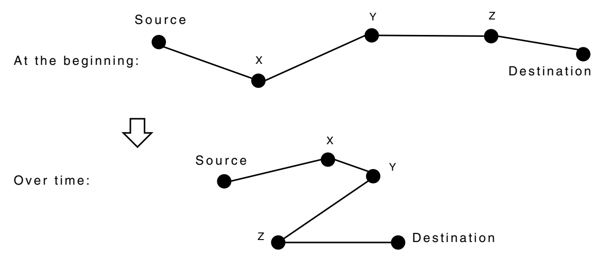

Fig. 1. Formation of a non-optimal route

In this paper, we propose a dynamic route optimization mechanism which eliminates the unnecessary hops in an active route "on the fly". The mechanism amortizes the additional control traffic for this optimization against the data traffic on the connection itself. Thus, connections with low traffic are less optimized than connections that exhibit high traffic volumes. The proposed scheme attempts to maintain near-optimality for connections throughout their lifetime. We give a concrete instantiation of the mechanism in the context of AODV, though the proposed method can be readily adapted to other reactive routing protocols.

II. Related Work

One of the earliest studies in this field is the work of Wu et al. [1] which considers several route optimization schemes for the routing protocols DSR, SSA, AODV, and ZRP. The authors exploit the promiscuous receive mode of wireless devices to collect fresh routing information. This strategy is criticized because the use of promiscuous mode may consume excessive power and thereby degrade the performance of the wireless cards. This study aims to optimize active routes, and requires some modifications to packet headers and routing tables of the original routing protocols.

In another study Park and Voorst [2] present an algorithm, called "Anticipated route maintenance" which predicts whether a link between two nodes will be broken within a predefined time interval, using their locations and velocities as determined

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

using GPS. Their algorithm consist of two phases: Expand and Shrink. The Expand routine prevents the route from being broken by inserting bridge nodes into a weak link. The Shrink routine eliminates unnecessary hops and shortens the path, thereby preventing it from being unnecessarily long. In order for the shrink phase to work the nodes must exchange routing table information. The implementation and performance studies were published subsequently in [3].

In contrast Park and Voorst's GPS-based approach, Qin et al. [4] propose a link breakage prediction algorithm based on the change in the signal strength of consecutive received packets. When a node estimates that there is a adjacent link which is likely to be broken soon, then it initiates a "Broken route message" and requests the source node to find an alternative path. In summary, Qin's mechanism tries to find an alternative path before the path become disconnected. It does not however try to eliminate the necessity of global route discovery processes when links break, but it decreases the expected number of dropped packets during the fail-over, by anticipating the link's imminent failure. While the algorithm is developed specifically for DSR, it is claimed to be easily adaptable into other routing protocols.

Following Park and Voorst's GPS-based solution, a recent 2008 paper by Sjaugi et al. [5] appeals to location information for nodes (as provided by GPS) in order to detect unsafe links–that is, links whose geometric length exceeds a certain threshold distance. The location information is updated by piggybacking it into the headers of packets. When a link is found unsafe, local (1-hop) broadcasting is performed in order to find a bridge node which can serve as an intermediate relay node between endpoints of the unsafe link. This is a path expanding routine, similar to what was proposed in Park and Voorst. However, Sjaugi et al. do not comment about shrinking operations, and so, their proposed mechanism may cause paths to become arbitrarily (and unnecessarily) long.

The related work above is characteristic of research efforts in recent years. Most of the focus concerns the problem of route recovery on prior to link failure, rather than establishing a continuous route optimization process. The latter is our main consideration here. The problem of dynamic route optimization is inherently problematic because a viable proposal cannot allow the energy required to determine *whether* routes are suboptimal to exceed the benefit of switching to more efficient routes. Put simply, continuous route optimization is difficult because one does not know when it would be beneficial to optimize, and merely checking whether it is (or not) costly. Our approach here is to *amortize the control traffic required for route optimization against the actual traffic on the route being optimized*. Our method does not rely on location information from GPS (like Park and Voorst, Sjaugi et. al, and others). Nor do we make use of any temporal extrapolations of signal strength (like the work of Qin et al., and others). Our proposal does not require any changes to the original routing table formats of AODV, and the header structure of existing AODV packets remains unaltered.

III. Background

The main function of a routing algorithm is to find an initial path between a source and a destination, and then to maintain data forwarding between the two nodes. Although there have been many reactive routing protocols proposed in the literature, for concreteness, we will cast our proposal as an extension to AODV in this paper[1].

In AODV, when a node (source) requires a connection to another node (destination), a global route discovery operation is initiated by the source, resulting in a flooding of ROUTE_REQUEST messages in the network. When (at least) one of these messages is received by the intended destination or by a node which has a fresh enough route to the destination, a ROUTE_REPLY message is sent back to the originator of the route discovery process. As the ROUTE_REPLY packet travels back towards to the originator of the request, each node along the route inserts next hop information into its routing table for the destination requested. Once the source node receives the ROUTE_REPLY message, it can start to forward data towards the destination along the route established.

IV. Route Optimization

We propose a route optimization mechanism we call ***Shrink***. The shrinking mechanism becomes active after a route between a source-destination pair has been constructed by the routing protocol. The objective of the shrinking mechanism is to shorten unnecessarily long paths by eliminating inessential hops. Such an operation has many potential advantages, including (i) it reduces the end-to-end delay incurred by packets by decreasing the number of hops on the path, (ii) it increases spatial reuse and network capacity, and provides energy savings by removing unnecessary transmissions, and in fact (iii) it makes the connection more resilient to breakages due to node mobility.

The **Shrink** mechanism is initiated periodically by the source node of each connection, as long as the connection is active and has data being sent on it. Rather than considering the period to be a fixed time interval, we view it as stochastically determined by the data rate. Concretely, the source node initiates the shrinking mechanism for a connection with some fixed probability **p** every time a data packet is to be sent.

The natural question that arises is what value of p should be used? This is one of the questions we will consider in the subsequent sections, where we will consider $p = 1/4$, $1/8$, $1/16$, $1/32$. In general we will denote the Shrinking mechanism with $p = 1/\alpha$ as ***Shrink-***α.

At the heart of the ***Shrink*** mechanism is the goal of reducing a 2-hop connection to 1-hop connection by eliminating an unnecessary relay node. The protocol will now be described in the context of Figure 2.

[1]However, to make the proposed mechanism compatible with other reactive routing protocols, we are careful to not make prohibitively strong AODV-specific assumptions. We will exploit next hop information for a specified destination at each node; this is already available in AODV, and indeed in almost all the reactive routing protocols.

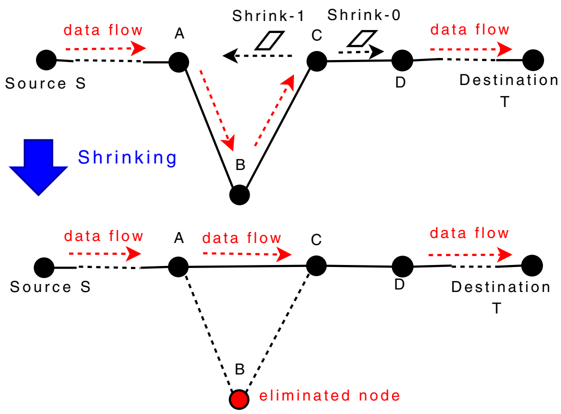

Fig. 2. The ***Shrink*** mechanism.

From the vantage point of the source node: Suppose that source node S has established a connection to destination node T, and the route between them is constructed by the routing protocol (e.g. AODV in this case). Assume furthermore that node S is sending data packets at constant bit rate of F packets/second. In our proposal, the shrinking mechanism is initiated by the source node S, as follows: Whenever node S has a data packet to send, with probability p it generates (an additional) special packet of type Shrink-0[2]. A Shrink-0 packet contains the following fields:

- The IP address of the previous node in the connection. Since S is the first node in the connection, there is no previous hop, and so in this case, the Shrink-0 packet generated by S has a special sentinel value in this field.
- The IP address of the sender (i.e. node S in this case),
- The IP address of the final destination (i.e. node T).
- The time to live (TTL) value (which is set to 1 by S).

Node S makes the Shrink-0 packet and sends it to the next hop, along on the same route as the data packet.

From the vantage point of intermediate nodes: When an intermediate node (e.g. node C) receives a Shrink-0 packet from the previous hop (e.g. node B), it produces two Shrink packets, one with flag 0 and one with flag 1. The packets include the IP addresses of the previous node, the current node, and the final destination. The Shrink-0 message is sent to the next hop (in the example, node D) along the path towards the destination. The Shrink-1, on the other hand, is sent to the upstream node two hops away (i.e. node A) with TTL of 1.

Elimination of the unnecessary hops: In this stage, two subsequent links on the route are replaced by just a single one if the all necessary conditions are satisfied: When a node receives Shrink-1 packet from another node, then it can be concluded that there may be a short cut available between this node and the sender of Shrink-1 message. However, the quality of this new hypothesized link may not be good enough to warrant changing the routing table. Therefore, before doing any update in the routing table, the quality of the prospective new link is checked by looking at the received signal strength at the receiver side (i.e. this is done by node A in Figure 2). If the received signal strength is greater than a predefined threshold level, then the node updates its routing table in such a way that the next hop for the corresponding final destination (specified in the Shrink-1 packet) is replaced with the address of the originator of the Shrink-1 packet. The process is illustrated in Figure 2.

In the previous paragraphs, we have explicated the proposed mechanism in terms of data flowing along a single connection, in order to make the exposition simpler. In practice, however, there can be many data flows (i.e. many connections) between different source and destination pairs, simultaneously. Such a situation does not alter the operation of the proposed mechanism, since the shrinking of each connection occurs independently of all others.

V. Experimental Setup

The proposed route optimization mechanism is implemented as an extension to the standard implementation of AODV in ns-2.33. The performance of the original AODV and AODV+Shrinking are compared for different network sizes, mobility models, and traffic/connection patterns. We followed those papers [6] [7] while setting up the experiments as follows:

Networks: Networks of 50 nodes were deployed uniformly at random in a 1500 m x 300 m rectangular field. Networks of 100 nodes were deployed in 2200 m x 600 m rectangular field. Note that the node densities in the two configurations were identical.

Mobility Model: The random waypoint mobility model was adapted in accordance with the deployment area. In this model, the nodes are randomly situated at the beginning of the simulation within the deployment area. Each node then starts to move towards a randomly chosen location with a uniformly random chosen speed between 0 and 20 m/s. When a node reaches its destination, there is a pause time before it starts a new movement. The pause time is used to adjust level of mobility. When pause time is set to 0, the nodes move continuously without stopping, which provides a maximally mobile setting. On the other hand, when pause time is set to the duration of the entire simulation, the nodes remain fixed; this provides us with a stationary network. Intermediate values of pause time correspond to intermediate levels of mobility.

Traffic Patterns: 10, 20, and 30 traffic connections between different source and destination pairs are initiated for 50 and 100 nodes cases. Traffic sources are constant bit rate (CBR) with 512 bytes packet size, and data rate of 4 packets/s.

Performance metrics: The following three metrics are evaluated:

- *Normalized path length*– This metric represents path optimality. The length of the paths (i.e. the number of hops) is compared by the number of hops on a min-hop path that would be calculated by Dijkstra's algorithm, which is provided by ns2.
- *Normalized average route lifetime*– The time interval between two consecutive link breakage on a connection is

[2]Here the 0 represents the status of a flag in the packet header.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

taken as a lifetime benchmark. The average value of these periods for each version is compared with the average value of connection lifetimes in pure AODV.

- *Normalized routing load*– Firstly, the number of routing packets transmitted per data packet delivered at the destinations is calculated. Then, these values for each version are normalized with the values in pure AODV.

VI. RESULTS

Although we obtained many results under different experimental parameters, we included only six of them due to space constraint.

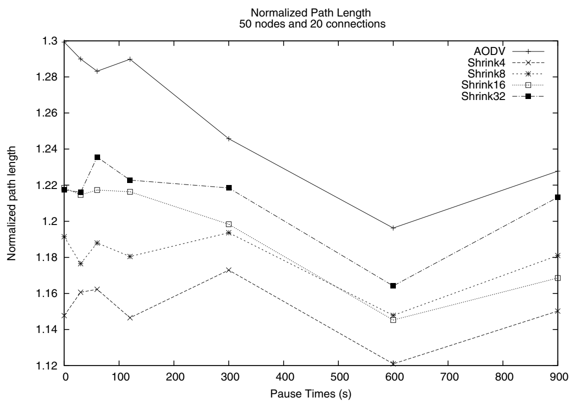

Fig. 3. Path length normalized by the optimum length(Dijkstra) for the 50-node with 20 traffic connections

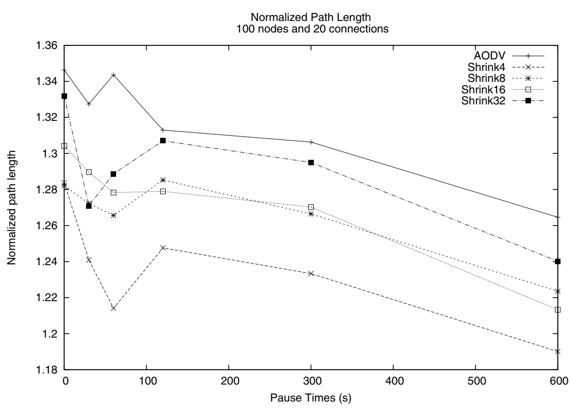

Fig. 4. Path length normalized by the optimum length(Dijkstra) for the 100-node with 20 traffic connections

Normalized Path Length: In this set of experiments, we measured the optimality of the paths, which is the primary objective of the proposed shrinking mechanism, comparative the pure AODV. Figure 3 and 4 represent the path optimality at different mobility levels in the context of normalized average path length of the connection. According to the figures, the average path lengths in the existence of shrinking mechanism (for all versions) are shorter than they are in plain AODV regardless of the network size. For example, in Figure 3, path length at maximum mobility level (i.e. pause time is 0) is 30% longer than the optimal one for pure AODV, as it is within the range of 15%-22% for the corresponding version of Shrink mechanism. This results show that the proposed mechanism works. Another observation deducted from the figures is that the more frequent shrinking operation (e.g. Shrink4 with respect to Shrink32) performed, the better results obtained. This is because of the fact that it is less likely to miss a possible shortcut at more frequent shrinking operations. Regarding the effect of mobility, it can be seen from the figures that normalized path lengths get worse as mobility level increases. This is because of the fact that occurrence of the sub-optimality at higher mobility happens more frequently. One interesting outcome is that even if there is no mobility (i.e. at pause time 900 seconds or 600 seconds in 50-node or 100-node networks respectively), the normalized path lengths are still greater than the min-hop path. This is because of the fact that AODV may not always provide the shortest path even if there is no mobility.

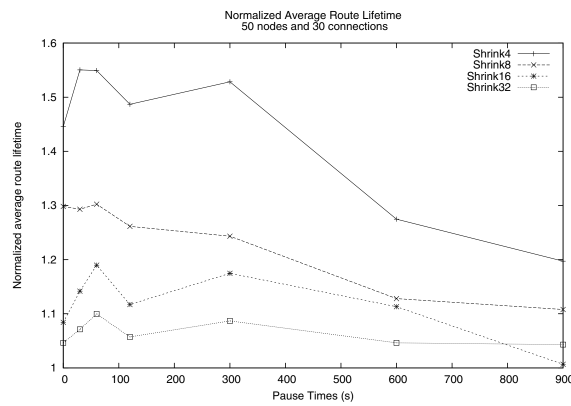

Fig. 5. Average route lifetime normalized by pure AODV for the 50-node with 30 traffic connections

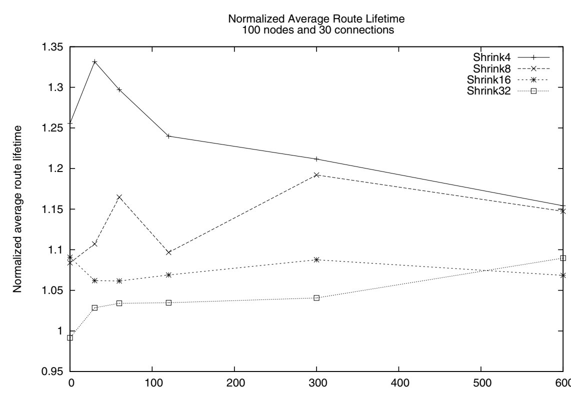

Fig. 6. Average route lifetime normalized by pure AODV for the 100-node with 30 traffic connections

Normalized Average Route Lifetime: Figure 5 and 6 show normalized average route lifetime at different mobility levels. According to the figures, average lifetime of the routes in the existence of shrinking mechanisms is always longer than it is in pure AODV. This is a sign of the fact that shrinking operation makes routes more consistent against node movements. It can also be explained by the fact that the probability of experiencing disconnection on shorter paths is smaller than it is on longer paths. Another interpretation of the figures is that more frequent shrinking operation increases connection lifetime more (notice that Shrink4 performs shrinking operation four times frequent than Shrink32). As mobility level decreases, the average route lifetime in each version converges to each other.

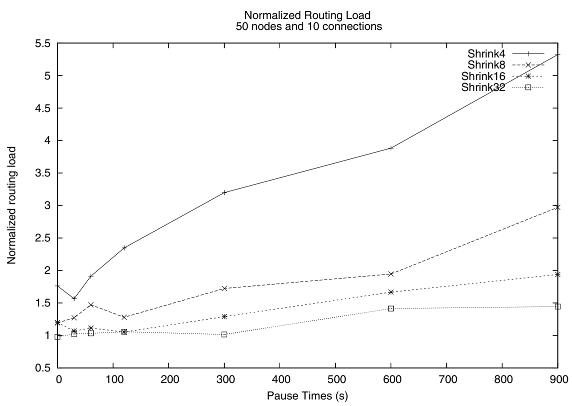

Fig. 7. Routing load normalized by pure AODV for the 50-node with 10 traffic connections

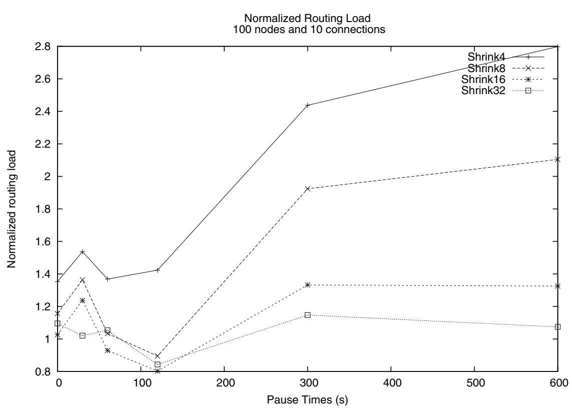

Fig. 8. Routing load normalized by pure AODV for the 100-node with 10 traffic connections

Normalized Routing Load: In this set of experiments, we examined the traffic load incurred by the proposed mechanism. Figure 7 and 8 indicate normalized routing load of Shrinking mechanisms at different mobility levels with respect to plain AODV. For example, in Figure 7, NRLs of Shrink-16 and Shrink-32, when pause time is 300 seconds, are about 1.25 and 1 respectively, which means the corresponding schemes increases NRL 25% and 0%. It is an expected result that more frequent shrinking operation produces more control traffic as seen in the corresponding figures. On the other hand, according to the figures, normalized routing load at high mobility is lower than it is at low mobility. This is because of the fact that when there is no mobility, all control traffic caused by the shrinking operation are wasted. Another observation from the figures is that the normalized routing load in 100-node experiments is lower than it is in 50-node experiment. This is because of the fact that the shrinking operation reduces the number of route discovery attempts (which is an expensive operation in terms of control traffic overhead especially in large networks) performed in the network by causing increase in connection lifetime, and thus decreases the amount of control traffic produced. As a final observation, the shrinking operation sometimes reduces the NRL below 1 at high mobility, which means it may even beat pure AODV in terms of NRL.

VII. Conclusion and Future Work

We propose and evaluate an extension to AODV which makes use of continuous route optimization through a *shrinking* process. We show that the proposed extension performs well with respect to routing overhead incurred, while serving to minimize path stretch relative to optimality. The experimental results show that the proposed route optimization mechanism works well.

As a future work, we are planning to improve the proposed scheme in terms of both control traffic incurred and resulting gain in path optimality. Indeed, the proposed scheme detects the shortcuts, if any, only between the nodes which are located 2-hop away to each other. Therefore, as a next step, we would like to develop a route optimization scheme which is able to detect shortcuts between any pair of nodes on a connection. Finally, such a route optimization scheme can lead researchers to develop efficient local recovery protocols without worrying about the path optimality, which has been addressed as a problem in local recovery operations.

Acknowledgments

The first author thanks the Turkish National Police for supporting his graduate studies at City University of New York, of which this research is a part.

References

[1] S. Wu, S. Ni, Y. Tseng, and J. Sheu, “Route Maintenance in a Wireless Mobile Ad Hoc Network.” In 33rd Hawaii International Conference on System Sciences, Maui, 2000.

[2] S. Park and B. Voorst, “Anticipated Route Maintenance (ARM) in Location-Aided Mobile Ad Hoc Networks.” Globecom, San Antonio, TX, USA, 2001.

[3] M. Al-Shurman, S.-M. Yoo, and S. Park, “A Performance Simulation for Route Maintenance in Wireless Ad Hoc Networks,” in *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*. New York, NY, USA: ACM, 2004, pp. 25–30.

[4] L. Qin and T. Kunz, “Pro-active Route Maintenance in DSR,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 79–89, 2002.

[5] M. F. Sjaugi, M. Othman, and M. F. A. Rasid, “A New Distance Based Route Maintenance Strategy for Dynamic Source Routing Protocol,” *Journal of Computer Science*, vol. 4, no. 3, pp. 172–180, 2008.

[6] C. Perkins, C. E. P. Nokia, E. M. Royer, S. R. Das, and M. K. Marina, “Performance comparison of two on-demand routing protocols for ad hoc networks,” 2001.

[7] J. Broch, D. A. Maltz, D. B. Johnson, Y. chun Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” 1998, pp. 85–97.