Only the Short Die Old: Route Optimization in MANETs by Dynamic Subconnection Shrinking

Zeki Bilgin
Department of Computer Science
The Graduate Center, CUNY
New York, 10016
zbilgin@gc.cuny.edu

Bilal Khan
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
bkhan@jjay.cuny.edu

Ala Al-Fuqaha
Department of Computer Science
Western Michigan University
alfuqaha@cs.wmich.edu

ABSTRACT

In reactive routing protocols, active routes for multihop connections retain their topological structure in spite of node movement over time. Unfortunately, node movements may make the connection route sub-optimal in terms of hop length, thereby resulting in unnecessarily high end-to-end delays, energy consumption and channel contention. In AODV, for example, a connection route is recomputed only if one of its constituent links suffers catastrophic failure, at which point global route discovery attempts repair, and after which the topological structure of the connection again returns to near-optimality.

In this paper, we propose an extension to AODV that performs periodic subconnection shrinking of the topological substructure within each connection. We show that this not only reduces the average end-to-end connection length, but also increases the mean time between catastrophic link failures of the connection's constituent links, thereby reducing the number of repair-related global route discoveries experienced. The control traffic needed to operate our scheme can be amortized against the reduction in repair-related global route discovery traffic. Through ns2 simulations, we show that our dynamic subconnection shrinking scheme manifests connections that, on average, have (i) shorter hop length, (ii) higher packet delivery fraction; moreover, this extension operates using less control traffic than standard AODV. We demonstrate that these conclusions continue to hold scalably over a wide range of operating regimes.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: -Networks Protocols–*Routing Protocols*

General Terms

Algorithms, Design, Performance

IWCMC '10, June 28 - July 2, 2010, Caen, France

Keywords

MANET, Ad hoc, mobility, route optimization

1. INTRODUCTION

Routing issues have been well-studied for wireless networking during 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 factors which developers must consider is the behavior of the proposed algorithm in the presence of node movement, since mobility can yield in dynamic changes to network topology.

Connection *suboptimality* arises in reactive routing protocols because the topological structure of connection routes between source-destination pairs is determined at the very outset, during the connection establishment phase. At the very beginning of a connection, the number of hops that the route takes tends to be close to the number of hops on the min-hop path (e.g. one that would be calculated by an omniscient instance of Dijkstra's algorithm). As time passes, nodes move, but as long as no link breaks, the connection retains its topological structure, and so in time can become significantly longer (in terms of hops) when compared to the length of the min-hop path between source and destination at that instant of time. An example in which route suboptimality occurs (over time) is depicted in Figure 1. In this

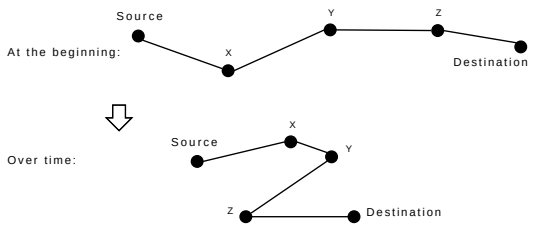

Figure 1: Formation of a non-optimal route

paper, we propose a route optimization scheme which eliminates the unnecessary hops in an active route "on the fly" while the connection is active. As we shall see, the scheme amortizes the additional control traffic for this optimization against the *data traffic on the connection itself.* Thus, connections with low traffic are less frequently optimized than connections than those that carry 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.

2. RELATED WORK

Several researchers have attempted to address the problem of dynamic path optimization in MANETS. Indeed, a recent paper by Jia et al. [3] compares several existing path compression techniques, while presenting a model of the path compression problem.

One of the seminal in this field is the work of Wu et al. [4] 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. Unfortunately, the ideas presented in this work are somewhat superficial, and have not been tested through experimental works. Similar to the work of Wu et al, there are other studies [2, 1, 5] which exploit the promiscuous mode of the wireless cards to search for possible short-cuts on a route. These authors' strategies are frequently criticized because of their use of promiscuous mode, which may consume excessive power and thereby degrades the performance of the wireless cards.

In contrast, Zapata [6] presents a shortcut detection mechanism for active routes that does not rely on the promiscuous mode of the devices. In his proposed work, a special packet (Shortcut Request–SREQ) is broadcasted with TTL of 0 at each node along the route starting from the source node to the destination node. The SREQ packet includes IP addresses of the source, the destination, and the next hop along to route, in addition to corresponding distance in hops to both endpoint of the path. Since all 1-hop neighbors can receive a broadcasted SREQ packet, upon receiving such a packet, they check if there is a short-cut between the sender of SREQ packet and themselves by comparing the hop count metric for the same route if defined in their routing table. If so, the corresponding node sends back a SREP packet to inform sender of SREQ node about the possible shortcut. Upon receiving a SREP packet, a node modifies its routing table such that the next hop for the corresponding destination is specified as the sender of SREP packet. Regrettably, the author's scheme has not been tested in extensive experiments, as is evidenced by the following oversight in the protocol's design: A node may have a valid routing table entry for the destination and yet might no longer lie on the connection itself. If such a node responds to an SREQ by sending an SREP, it will cause the formation of an invalid route. Our own work in this paper uses a similar path optimization mechanism, but our scheme uses a protocol which is designed to sidestep the aforementioned pitfalls.

3. ASSUMPTIONS

To make the proposed scheme adaptable to other reactive routing protocols, we are careful to not make strong AODV-specific assumptions. The following assumptions were made in developing the proposed mechanism:

A1. *Bidirectionality:* We assumed that the links in the network are bidirectional.

A2. *Distance (hop count) to the destination:* We assumed that the hop count to reach destination is known at each node on the connection.

A3*. *Distance (hop count) to the source:* (Optional) Although it is not required for the operation of our scheme, some reduction in control traffic overhead can be obtained if we assume that the hop count from the source is also available at each node along the route, which can be obtained from the IP headers of data packets traveling along the connection, if we assume that all data packets are originated with a specific TTL (e.g. 255).

4. ROUTE OPTIMIZATION

We call our proposed route optimization scheme Multihop Shrinking. The objective of the scheme is to shorten unnecessarily long connections by eliminating inessential hops as illustrated in Figure 2. In order to do this, the mechanism periodically checks if there is any direct short-cut between non-adjacent pairs of the nodes on the connection. More precisely, the mechanism checks (repeatedly) whether there is any upstream node within the direct transmission range of the node of interest, and it does this for each node along the connection. If there is such a shortcut between two distant nodes on the same route, and the channel quality of the shortcut is "good enough", then the Multihop Shrink mechanism modifies the connection topology so that two endpoint nodes of the short-cut connect to each other directly, thereby eliminating the inessential intermediary *node(s)* between them (see Figure 2. Such an operation has many potential advantages, including (i) reducing the end-to-end delay incurred by packets by decreasing the number of hops on the path, (ii) increasing spatial reuse and network capacity by eliminating unnecessary transmissions, and as we shall see, (iii) providing energy savings by reducing the number of transmitting nodes.

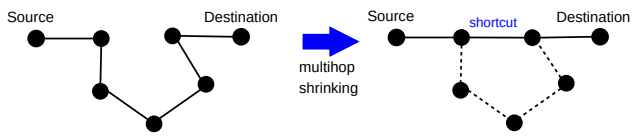

Figure 2: The main idea of *Multihop Shrink* mechanism.

The Multihop Shrinking mechanism is activated only *after* the routing protocol has established a route between source and destination. Once the connection has been established, and for as long as the connection is active, the source node periodically sends a special Shrink packet towards the destination node, which triggers the shrinking operation at each node it traverses.

The periodic sending of the Shrink packet is to be viewed as stochastically determined by the flow of data on the connection, rather than as a message with a fixed temporal frequency. More precisely, the source node of the connection initiates the shrinking process when a certain fixed amount of data traffic has been sent on the connection. In our experiments, we consider connections that carry CBR traffic that is conveyed in fixed size packets and thus, the shrinking process is initiated every time a specified number ($\boldsymbol{p}$) of data packets have been sent. We can avoid keeping a data packet counter at each node for each destination by implementing

the counters probabilistically: each source node may initiate the Shrink packet with probability $1/p$ whenever it sends a data packet. In what follows, however, for simplicity of exposition, we will not consider this space optimization. A natural question that arises concerns the effect of the choice of p on the performance of the optimization scheme. This is just one of the questions we will investigate in the subsequent sections, where we will consider $p = 4$, 8, 16, 32. In general we will denote the Shrinking mechanism when $p = \alpha$ as ***Shrink(α)***.

As the Shrink packet travels downstream, nodes that see the special packet attempt to discover a shortcut to *upstream* nodes on the connection. A further optimization is evident now, since such a shortcut can never be found by the source (the first node), or the second node. Thus, in scenarios where the assumption A3* can be made (see Section 3), the shrinking operation is initiated by the *third node* within the connection, rather than at the source. In settings where assumption A3* cannot be made, identifying the third node may not be feasible: the Multihop Shrink scheme still operates correctly, but is slightly more wasteful in terms of control traffic overhead. In short, the ***Multihop Shrink*** mechanism achieves its goal of optimizing connection topology by replacing inefficient multihop subconnections with a direct 1-hop connection thereby eliminating unnecessary relay node(s). The protocol that achieves this will now be described, with the aid Figure 3.

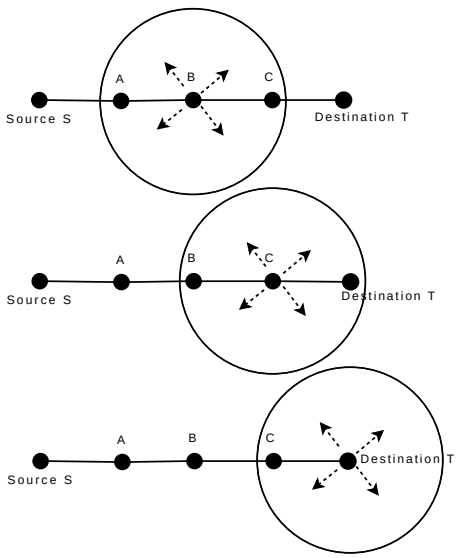

Figure 3: Illustration of how multihop shrinking works

Initiation of the process: Suppose that source node S has established a connection to destination node T, and the route between them is constructed by a reactive routing protocol such as AODV. Assume furthermore that node S is sending data packets at constant rate of F packets/second. In our proposal, the shrinking mechanism is initiated by the *third node B*, which can recognize itself as the third node by looking at the corresponding field in the IP header of data packets received, as per assumption A4*. Periodically, upon receiving a certain number of data packets (4, 8, 16, and 32, respectively), the *third node B* initiates the procedure by making a special Shrink packet, and *broadcasting* it with TTL=1 (which means the packet can go at most 1-hop away from the originator of it). A Shrink packet contains the following fields:

- **sender**: The IP address of the sender of the Shrink packet (i.e. node B in the example),
- **next-hop**: The IP address of the next hop (i.e. node C in the example),
- **final-destination**: The IP address of the final destination (i.e. node T).
- **hops-to-destination**: The number of hops to the final destination (available in the routing table)

From the vantage point of other nodes: A node which receives a Shrink packet performs actions depending on its relative placement on the connection with respect to sender of the Shrink packet. A node, upon receiving the Shrink packet, determines its relationship to the Shrink packet, selected from the following set of five mutually exclusive classes:

- *The next hop:* A node identifies itself to be the *next hop* by noting that its own IP is the one specified in the next-hop field of Shrink packet it received. In that case, it modifies the received Shrink packet by updating the related fields using the information in its routing table, and then broadcasts the updated Shrink packet.
- *Further downstream hops:* A node recognizes itself to be in this class if its routing table indicates that the hop count to the destination is smaller than the value in the hops-to-destination field of the received Shrink packet. Nodes in this category discard the received Shrink packet.
- *The previous hop:* A node identifies itself to be the *previous hop* when it realizes that in its routing table, the sender of the Shrink packet is listed as the next hop to the final-destination. This node just discards the received Shrink packet.
- *Further upstream nodes:* A node recognizes itself to be in this class if its routing table indicates that the hop count to the destination is greater than the value in the hops-to-destination field of the received Shrink packet. If this is the case, then it can be concluded that there may be a short cut available between this node and the sender of Shrink 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 to the routing table, the quality of the prospective new link is checked by looking at the signal strength at which the Shrink packet was received. 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 final-destination is replaced with the address of the sender the Shrink packet.
- *Irrelevant nodes:* When a node receives Shrink packet, but there is no next hop in its routing table for the final-destination specified in the Shrink packet, the the node classifies itself as irrelevant to the Shrink packet, which is simply ignored.

Every node receiving a Shrink packet falls into precisely one of the above five classes, and behaves as specified. In practice, there can be many data flows (i.e. many connections) between different source/destination pairs, all active simultaneously. Such a situation does not alter the operation of proposed mechanism, since each node responds to different Shrink messages arising from different transient connections independently. No further state needs to be maintained at each node than what is mandated in AODVs routing table format and a 4 bit counter for each destination address in the routing table[1].

5. 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+Multihop Shrinking are compared for the following network size, mobility models, and traffic/connection pattern as follows:

Networks: Networks of 50 nodes were deployed uniformly at random in a 700 m x 700 m rectangular field.

Traffic Patterns: 20 traffic connections between different source and destination pairs are initiated among 50 nodes. Traffic sources generate constant bit rate (CBR) traffic consisting of packets of size 512 bytes, at a rate of 4 packets per second.

Mobility Model: To investigate the performance of the proposed mechanism under different mobility levels, we modified random waypoint (RWP) mobility model as follows: First we set the pause time to zero (nodes move without stopping between subsequent movements). We then generated a movement plan with maximum speed of 5 m/s. To obtain relatively higher mobility scenario, we multiplied the velocity of each node by $\beta > 1$. The advantage of this modified RWP is that as β is increased, exactly the same topological changes of (link level) networks arise, though this changes evolve at a faster rate. We considered $\beta = 1, 2, 3, 4, 5$. Bigger values of β signify higher mobility scenarios. Notice that the simulation duration has to be changed inversely proportional with the β in this model. For example, for maximum speed of 5 m/s the simulation duration is adjusted 1200 seconds, while it is 600 seconds for 10 m/s maximum velocity scenario, and so on.

Performance metrics: The following three metrics are evaluated:

- *Normalized path length*– The length of the paths (i.e. the number of hops) is normalized by the number of hops on a min-hop path that would be calculated by an omniscient centralized instance of Dijkstra's algorithm. Values closer to 1 are exhibited by schemes which are better at maintaining path length optimality.

- *Packet delivery fraction*– The percentage of successfully delivered data packets to destination nodes, out of the number of data packets sent by source nodes.

- *Normalized Routing load*–We calculate the number of routing-related control packets transmitted for each data packet delivered at the destinations.

[1]As noted earlier, this minor space requirement can be removed by considering a scheme in which Shrink packets are generated probabilistically with probability $1/p$ every time a data packet is sent from the source node.

6. RESULTS

Normalized Path Length (NPL): In this set of experiments, we measured the optimality of the paths for standard $AODV$ and $AODV + Shrink(\alpha)$ under different mobility levels. This experiment may be the most important because it tests the extent to which the schemes meet the primary objective, namely to dynamically optimize length (i.e.hop count) of a connection. The results are normalized with respect to the optimal path lengths that would be calculated by Dijkstra's algorithm. We vary mobility settings from maximum velocity of 5m/s to 25m/s.

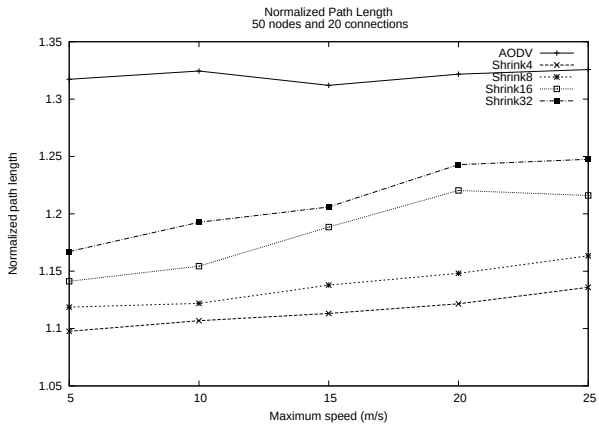

Figure 4: Path length normalized by the optimum length(Dijkstra) for the 50-node with 20 traffic connections

Figure 4 depicts the extent to which our scheme maintains path optimality at different mobility levels, as quantified by mean normalized average path length. The most significant result which can be inferred from this figure is that *the proposed mechanism works.* According to the figure, the paths produced by standard $AODV$ are about 32% longer than optimal one, regardless of the mobility level. When Multihop Shrinking is in effect, however, path lengths are much typically within 9% of optimal. Another observation that can be deduced from the figure is that more running shrinking operations more frequently (e.g. Shrink4 with respect to Shrink32) yields better results in terms of path optimality. This is because it is less likely that we miss a possible shortcut when shrinking operations are done more frequently. It can be also seen from the figure that normalized path lengths increase slightly as mobility level increases (except for pure $AODV$). This is because the decisions made by the shrinking operation yield benefits for ever shorter time intervals as mobility increases; this is not a case for pure $AODV$.

Packet Delivery Fraction (PDF): Figure 5 shows PDF for pure $AODV$ plus different shrinking mechanisms. It can be inferred from the figure that the proposed mechanism improves the packet delivery fraction regardless of its version α. For instance, while PDF is 81% for pure $AODV$ at 5m/s case, it rises to up to 88% when multihop shrinking is applied. Moreover, the more frequent shrinking (e.g. Shrink-4) contributes more towards better packet delivery,

than the less frequent version of it (e.g. Shrink-32). The increase in PDF is mostly because of the decrease in contention due to both shorter routes and lower control traffic overhead (as shown in the following subsection).

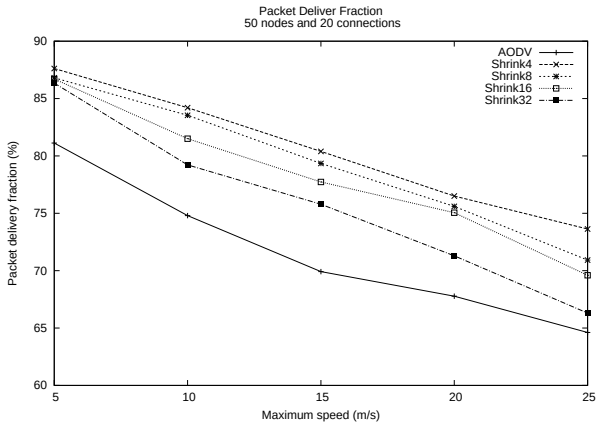

Figure 5: Packet delivery fraction for the 50-node with 20 traffic connections

Normalized Routing Load (NRL): Figure 6 indicates normalized routing load of both pure $AODV$ and $AODV + Shrinking$ mechanisms at different mobility levels. The first observation inferred from the figure is that although the proposed mechanism incurs additional control traffic over $AODV$, the NRL of $AODV + Shrinking$ is lower than pure $AODV$'s NRL in almost all cases regardless of the shrinking period α. This is because of the fact that experiencing a link failure on shorter routes is less likely that it is on longer routes. Since the proposed scheme reduces the path length, it results in longer connection lifetime and fewer route discovery attempts–the latter being expensive in terms of control traffic incurred. Another exciting conclusion deduced from the figure is that there is an optimal frequency for Multihop Shrinking, as indicated by the observation that the lines cross each other at different mobility levels. For example, Shrink-32 has the lowest NRL for 5m/s maximum velocity, while Shrink-4 has the lowest NRL in the case of 25m/s maximum velocity.

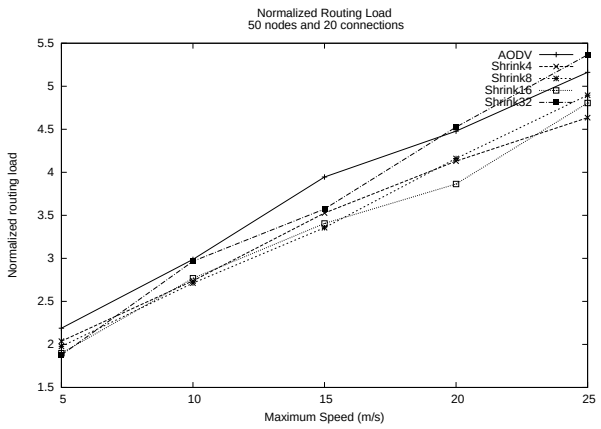

Figure 6: Routing load per delivered packet for the 50-node with 20 traffic connections

7. CONCLUSION

In this paper, we described a scheme called **Multihop Shrink** as an extension to AODV. The scheme seeks to counteract the inefficiencies in connection topology that arise due to node mobility. In contrast with AODV, however, our scheme works proactively; it does not wait for catastrophic link failure to occur, nor does it rely on global route recovery to serendipitously rectify connection topology inefficiencies. Instead, the proposed Multihop Shrink scheme performs periodic shrinking of the topological substructure within each connection.

Through extensive ns2 simulations, we showed that the proposed scheme reduces the average end-to-end connection length (compared to AODV). The proposed scheme reduces the number of repair-related global route discoveries triggered by a long lived connection. Indeed, the simulations indicate that the control traffic needed to operate our Multihop Shrink scheme can be amortized against the corresponding reduction in repair-related global route discovery traffic. Thus, the Multihop Shrink modification to AODV achieves intended objectives (i.e. NPL and PDF) with lower control traffic compared to even standard AODV in most settings. We demonstrated that our conclusions continue to hold scalably over a wide range of mobility scenarios.

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.

8. REFERENCES

[1] V. C. Giruka, M. Singhal, and S. P. Yarravarapu. A path compression technique for on-demand ad-hoc routing protocols. In *In Mobile Ad-hoc and Sensor Systems (MASS) 2004*, 2004.

[2] C. Gui and P. Mohapatra. Short: self-healing and optimizing routing techniques for mobile ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 279–290, New York, NY, USA, 2003. ACM.

[3] X. Jia, L. Qian-mu, Z. Hong, and L. Feng-yu. Model and analysis of path compression for mobile ad hoc networks. *Computers and Electrical Engineering*, In Press, Corrected Proof:–, 2009.

[4] 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.

[5] Y.-S. Yen, H.-C. Chang, R.-S. Chang, and H.-C. Chao. Routing with adaptive path and limited flooding for mobile ad hoc networks. *Computers and Electrical Engineering*, 36(2):280 – 290, 2010. Wireless ad hoc, Sensor and Mesh Networks.

[6] M. G. Zapata. Shortcut detection and route repair in ad hoc networks. *Pervasive Computing and Communications Workshops, IEEE International Conference on*, 0:237–242, 2005.