Using Connection Expansion to Reduce Control Traffic in MANETs

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

We consider the problem of control traffic overhead in MANETs with long-lived connections, operating under a reactive routing protocol (e.g. AODV). In such settings, control traffic overhead origins can be traced principally to connection link failures, which trigger expensive global route discoveries. In this paper, we introduce a route maintenance scheme developed with the objective of reducing global route discoveries in such settings. The proposed scheme decrease the expected number of route discovery attempts by taking pre-emptive action to counteract impending link disconnections due to *node movement*. The proposed scheme was implemented as an extension of AODV in ns2, and compared with the standard AODV under different network regimes. Through the analysis of data derived from extensive simulations, we demonstrate that the proposed scheme significantly decreases overall control traffic while maintaining comparable packet delivery rates, at the cost of only very minor degradation in path optimality.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Performance

Keywords

MANET, mobility, route maintenance, link expansion

1. INTRODUCTION

The problem of routing has been the subject to extensive research efforts, both in static settings, where nodes are fixed, and dynamic settings, in which they are mobile. In static networks, the problem of routing is relatively easy, because the problem needs to be solved infrequently. Once a route is established for a pair of source and destination nodes, it is likely to be valid a long time of period because link failures and link formation is rare in static networks. However, this is not the case for dynamic networks. In these settings, routing is a more challenging problem since in the presence of mobility, node movement induces frequent network topology changes, causing both the *formation* of new links and the *disconnection* of existing ones.

Disconnection of links generally entails extra work for routing protocols, since existing connection paths must be repaired. More precisely, when a link on an active connection is broken, the routing protocol generally reinitiates a local or global route discovery operation, which usually requires flooding–an expensive operation in terms of control traffic incurred. Strangely, the dual event–i.e., node mobility causing the creation of new links resulting in the *connection* of previously disconnected nodes–presents a resource for the routing protocol, that is, for the most part, ignored. In this work, we use this resource. Specifically, we develop a route maintenance scheme that reduces overall control traffic overhead by pre-emptively and surgically replacing the weak links within a connection on which failure is immanent.

2. RELATED WORK

One of the earliest studies in this field is the work of Park and Voorst [3], who presented an algorithm called "Anticipated route maintenance" to predict whether a link between two nodes will break within a predefined time interval. The authors use the node locations and velocities, as determined via GPS, to derive their likelihood estimates. Park and Voorst's 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. The implementation of the scheme and performance studies of their scheme were published subsequently in [1]. Unfortunately, their scheme requires GPS, and its implementation requires nodes along a connection to exchange routing table information.

In contrast to 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 an incident link which is likely to be broken soon, it initiates a "Broken route message" and requests the source node to find

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

an alternate path. The authors proposed mechanism tries to find an alternate path *before* the path become disconnected. They do not attempt to eliminate the frequency of global route discovery, but rather seek to minimize the expected number of dropped packets during the fail-over sequence by anticipating the link's imminent failure.

Following Park and Voorst's GPS-based solution, a recent paper by Sjaugi et al. [6] 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.

3. ROUTE MAINTENANCE SCHEME

We call our proposed route maintenance scheme *Expansion*. Our objective is *to decrease the overall control traffic overhead of a dynamic network in MANET environment by reducing the expected number of expensive global route discovery attempts during the connection's lifetime.*

We assume that nodes are able to estimate link quality based on the signal strength of received packets. Based on this, we classify links into two groups: (i) *Weak link:* If the received signal strength is lower than a predefined threshold level[1], then the link the data packet arrived on is labeled as a *weak link* and regarded as likely to break soon; (ii) Otherwise, the link is regarded as strong.

The Expansion scheme detects weak links by measuring the received signal strength of incoming data packets, and then performs actions in order to eliminate these weak links, on-the-fly, before failures occur that would require expensive global route recovery. These actions entail the following two fundamental operations:

- **Expand operation:** A *weak* link is replaced with two strong links by inserting an extra relay node between two endpoints of the weak link (see Figure 1).

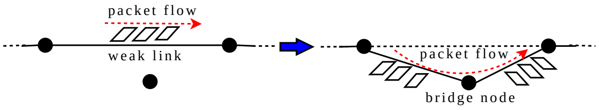

Figure 1: Expanding operation

- **Shrink operation:** A subpath which includes the *weak* link is eliminated by directly connecting either (i) the downstream endpoint of the weak link to an upstream node, or (ii) the upstream endpoint of the weak link and the final destination, as illustrated in Figure 2. Such shortcuts may come into existence because of topological changes in network due to node movements.

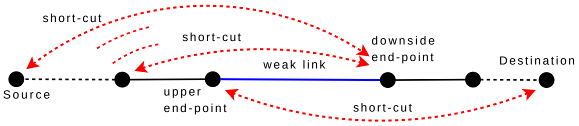

Figure 2: Shrink operation

Both of the above operations can result in eliminating weak links without causing *any* interruption on the connection. When a weak link is detected, both operations are attempted simultaneously. The Shrink (resp. Expand) operation may not always succeed because of unavailability of appropriate shortcuts (resp. bridge nodes). If both succeed, the side-effects of the Shrink operation supersede the side effects of the Expand operation by rendering the expanded segment irrelevant to the connection flow. Notice that the expand operation performs its task by lengthening the path, while the Shrink operation fulfills its mission by shortening the path, and thus over time, they complement each other.

3.1 Implementation

The implementation is explained by reference to Figure 3.

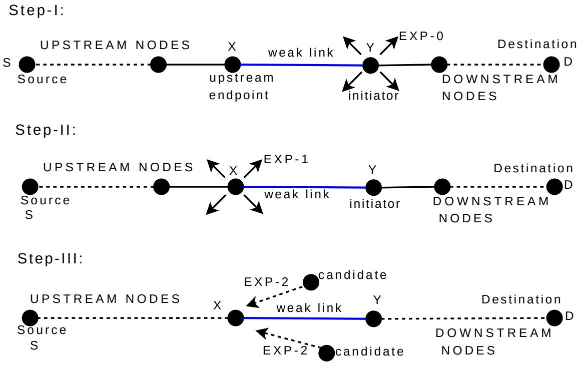

Figure 3: Steps I, II and III of the Expansion scheme.

The protocol is built using EXP messages with the following fields:
-source-node: The IP address of the source node.
-destination-node: The IP address of the final destination.
-initiator: The IP address of the node initiating Expansion.
-previous-hop: The IP address of the node before the initiator on the route (can be obtained from ARP).
-hops-to-destination: The number of hops to the final destination (available in the routing table).
-flag: 0, 1, 2, or 3, depending on whether the message is an EXP-0, EXP-1, EXP-2, or EXP-3.

Step Ia. Initiation of the process: When a node (i.e. Y) detects, on the basis of the signal strength of a received data packet, that it is incident to a weak link (e.g. X–Y), the node is declared an ***initiator***. An initiator node starts the Expansion process by preparing a special EXP-0 packet (i.e.

---

[1] We took this to be the strength of a received signal at 90% of the common transmission radius. For example, if the transmission radius is 250m, then threshold level corresponds 225m, when we consider the two-ray ground reflection model [5] as implemented in ns2.

initiator=Y, previous-hop=X, final destination=D, source node=S, flag=0) and broadcasts it with TTL of 1.

Step Ib. Applying shortcut from the upstream nodes: After EXP-0 is broadcasted by the initiator node, any upstream node on the connection that receives it "over a strong link"[2], modifies its routing table so the initiator becomes the next hop for the destination declared in the received EXP-0 packet. The weak link is thus eliminated by shortening the path. If there are shortcuts between many upstream nodes and the initiator node, the shortcut established by the upstream node closest to the source node determines the flow of packets. Note that if both steps Ib and step II are successful, then the shortcut established in step Ib simply nullifies the side effects of step II in terms of the route of packet flow.

Step II. Applying shortcut to the destination: After the EXP-0 packet is broadcast by the initiator node, the upper endpoint of the weak link (i.e. node X) receives it and prepares another packet, identical to the received EXP-0 except that the previous-hop field is cleared, and flag=1. This EXP-1 is broadcast by X with a TTL of 1, as illustrated in Step-II of Figure 3. If the final destination (i.e. node D) receives the EXP-1 packet over a strong link, it sends a special unicast EXP-3 packet[3] to the originator of EXP-1 (e.g. node X) in order to inform it about the possible shortcut between them. Upon receiving the EXP-3 packet, the upper endpoint of the weak link (e.g. node X) updates its routing table so that the next hop is directly the destination node D. In this manner, the weak link is eliminated, and the path is be shrunk. One might wonder why only the destination node is taken as a potential candidate, rather than the more general strategy of finding downstream shortcuts. This is because there may be nodes with a valid route for to the destination (in their routing table) which are no longer located on the route itself (because of shortcut operations executed in earlier occurrences of steps Ib/II). Such nodes may mislead the proposed scheme if they are taken as candidate downstream shortcuts. The analogous problem does not manifest in searching for upstream shortcuts.

Step III. Bridge node insertion:

Regardless of whether the previous steps find a shortcut to eliminate the weak link, the proposed scheme performs a path expansion process to circumvent immanent link breakage. In this process, an extra relay node is inserted into the route. It is obvious that such a bridge node should be selected among the nodes located in the vicinity of the two endpoints of the weak link. However, the problem is that there is no information available regarding location of the nodes. Therefore, we developed a path expansion process by benefiting from EXP-0 and EXP-1 packets, which are broadcasted in earlier steps by the downstream and upstream endpoints of the weak link respectively. The *expand* process works as follows:

If a node receives EXP-0 and EXP-1 packets (with matching initiator, source and destination fields), both with high signal strength, it concludes that it is a candidate bridge node which lies in the intersection of transmission radii of weak link 's endpoints. Candidate bridge nodes send a special unicast packet, EXP-2, to upstream endpoint of the weak link (i.e. node X) in order to advertise itself as a candidate (see Step-III in Figure 3).

Bridge quality: Because there may be many candidate bridge nodes, the upper endpoint of the weak link may receive several EXP-2 packets. To facilitate the choosing from among the candidates, the EXP-2 message is augmented to carry an assessment of the bridge node's candidacy. In our implementation, each candidate bridge node records the received signal strength of the EXP-0 and EXP-1 messages, and puts the lower of these two values in the EXP-2 message. Upon receiving the first EXP-2 packet, the upstream endpoint of the weak link "executes" the following procedure: It inserts the bridge node by modifying its own routing table entry for D to be the sender of the EXP-2 packet. Upon receiving later EXP-2 packets, it compares the channel quality specified for the current bridge node, against the one advertised in the later EXP-2 packet. The later EXP-2 is deemed "executable" only if the bridge being advertised in it is of higher quality than the old one. Bridge node selection converges once all candidates have send their EXP-2 packets.

3.2 Updating Distance Information in Routing Tables

In AODV, there is a field in routing tables of the nodes which maintains hop-count to destination. The values kept in this field must be strictly increasing along a route from destination to source in order to prevent loop formation during the route discovery processes [2]. However, when a bridge node is inserted into an active route as in the proposed expand process, this monotonicity may be disturbed because the field's values in upstream nodes from the bridge node are no longer correct; they need to be increased by 1. Such an disorder may also cause malfunctioning of the proposed *Shrink* scheme, since nodes use the hop-count to destination to determine their topological relationship to the initiator. To fix this problem, all upstream nodes from bridge node to source node need to be informed about the new correct hop-count to destination. The challenge is that nodes on the route are generally not aware of their upstream nodes, rather they only know the next (downstream) hop towards a specific destination. To address this issue, we developed the following mechanism:

When the upstream endpoint of a weak link receives an executable EXP-2 packet (meaning the advertised bridge node will be inserted) it prepares a special packet, called EXP-4, and broadcasts it with TTL of 1 (initiator=X, final destination=D, source node=S, flag=4). When a node receives EXP-4, it infers that distance updating is to take place. To do this, the node checks its routing table to determine if the sender of the EXP-4 packet is the next hop for the specified destination in the packet. If it is, then it updates the hop-count to destination field in its routing table, and then broadcasts a similar EXP-4 packet in which its address is in the initiator field. Thus, the update process progresses towards source node using EXP-4 messages, and stops when it reaches the source node.

4. EXPERIMENTAL SETUP

The proposed scheme is implemented as an extension to the standard implementation of AODV in ns-2.33. The performance of the original AODV and AODV+Expansion are compared for the following network size, mobility models, and traffic/connection pattern as follows:

Networks: Several network sizes are investigated, comprised of between 50 and 100 nodes. Node density is kept

---

[2] i.e. with a signal strength exceeding the weak link threshold, as defined in footnote 1

[3] Only the flag field is relevant in EXP-3 packets.

fixed at 50 nodes per 700 m x 700 m. Initial placement of nodes is uniformly random.

Traffic Patterns: The traffic connections are initiated between randomly chosen source and destination pairs chosen uniformly from the nodes. The number of connections varied between 1 and 50, depending on the type of experiment. 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). Then we generated a movement plan with maximum speed of 5 m/s. To obtain higher mobility scenarios, 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 $\beta = 1, 2, 3, 4, 5$, with larger values of β signifying higher mobility scenarios. Note that under this mobility model, the simulation duration changes inversely with β. For example, when maximum speed is 5 m/s the simulation duration is 600 seconds, while it is 300 seconds when the maximum speed is 10 m/s. Our performance metrics are robust to this fact.

Performance metrics: The following three metrics are evaluated:
Normalized Routing load–the number of routing-related control packets transmitted per data packet delivered to its destination.
Packet delivery fraction– The percentage of successfully delivered data packets to destination nodes, out of the number of data packets sent by source nodes.
Path Optimality–For each data packet delivered to its destination, we note both the number of hops that packet traveled and the length of the optimal source-destination path at the time of packet delivery. Then we calculate the discrepancy, both as ratio and as a difference.

Trials: To increase our confidence in conclusions drawn from the analysis of simulation data, we repeated the experiments 10 times with different mobility plans generated as above; we averaged the outcomes, and showed the standard deviations of our performance metrics.

5. RESULTS

Normalized Routing Load (NRL): We investigated NRL of both pure *AODV* and *AODV + Expansion* under different mobility levels, network sizes, and traffic loads. Figure 4 indicates the relationship between NRL and different mobility levels. It is clear from the figure that the proposed scheme reduces NRL of the network at all velocity levels, and that the gains are more pronounced at higher mobility levels. This is because the proposed mechanism decreases the number of global route discovery attempts performed in the network.

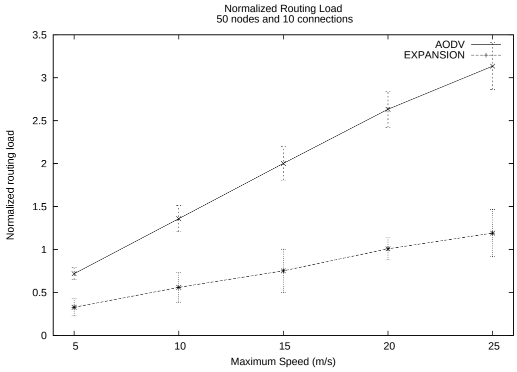

Figure 4: NRL vs. Mobility

In the second set of NRL experiments, we consider the effect of network sizes. For these experiments, we normalized AODV's NRL with respect to the Expansion scheme's NRL. As seen in Figure 5, the relative advantage of the Expansion scheme is significant. While its advantage decreases for larger networks, but yet remains always above 1, indicating that the proposed scheme is always better than pure AODV in terms of NRL, regardless of the network size. The relative advantage gained by the Expansion scheme decreases for larger networks because updating hop-count information becomes expensive where connection length is high.

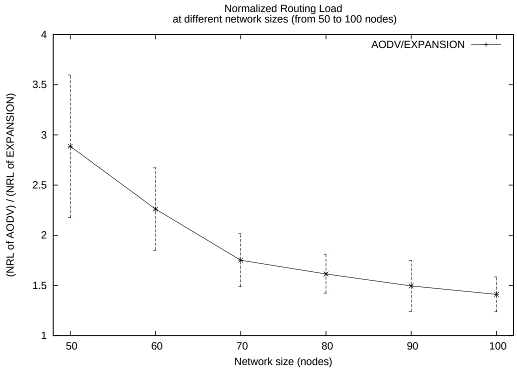

Figure 5: NRL vs. Network size

In the third set of NRL experiments, we consider the effect of data traffic load. For these experiments, we normalized the AODV's NRL with respect to Expansion scheme's NRL. As seen in Figure 6, the relative advantage of the Expansion scheme is significant. While its advantage decreases in settings with higher data traffic load, but yet remains always above 1, indicating that the proposed scheme is always better than pure AODV in terms of NRL, regardless of the traffic load. This effect is explicable because there is more contention when data traffic load is high, and this phenomenon interferes with the operation of the Expansion scheme itself.

Packet Delivery Fraction (PDF): Figure 7 shows PDF for pure $AODV$ and $AODV+Expansion$. The graph shows that the proposed mechanism improves the PDF at all velocity levels. For instance, while PDF is about 97% for pure $AODV$ at 15m/s case, it rises to up to 98% when the Expansion scheme is applied. This increase in PDF comes from the fact that the proposed mechanism decreases packet losses which may occur due to connection failures, by preventing anticipated link breakages before they occur.

We see from the preceding analysis that the Expansion scheme is capable of significantly reducing normalized traffic

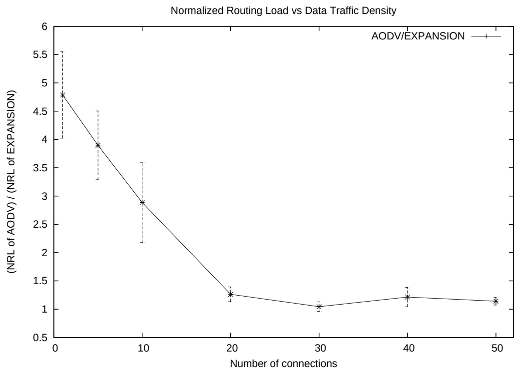

Figure 6: NRL vs. Traffic load

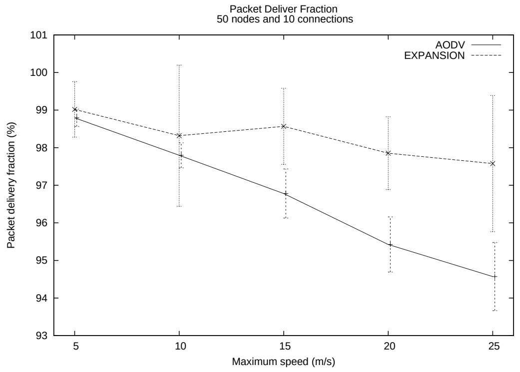

Figure 7: Packet Delivery Fraction vs. Mobility

overhead, without compromising on packet delivery fraction. The question is, at what price is this achieved? Clearly, since the scheme relies on path expansion, the scheme will be worse than pure AODV when it comes to path optimality.

Path Optimality: In this set of experiments, we calculated for each data packet delivered to its destination, the number of hops that the packet traveled in excess of the length of the optimal source-destination path (at the time of packet delivery). The histogram below shows the relative frequencies of the discrepancy from optimal, for both $AODV$ and $AODV + Expansion$ schemes. The experimental distribution was determined by averaging the histograms of 10 experiments, each having 10 random connections among 50 nodes moving at a maximum velocity of 5 m/s.

As expected, both AODV and AODV+Expansion make data packets travel on paths that are longer than optimal. The histogram makes apparent that $AODV + Expansion$ tends to make more packets travel on longer paths than pure $AODV$, yet the increase is quite modest. Interpreting the distribution in Figure 8 cumulatively, one can verify that $AODV$ delivers 92% of the data packets on paths whose length is at most 1 more than optimal. In contrast, $AODV+ Expansion$ delivers 92% of the data packets on paths whose length is at most 3 more than optimal, and only 68% of the data packets on paths whose length is at most 1 more than optimal. The results are to be expected, since the Expansion scheme causes packets to travel on paths even further from optimal than those selected by AODV–this is because the *Expand* operation favors forming longer paths over incurring global route discovery when connection links fail. The *Shrink* operation balances the Expand operation to some extent, however, as is apparent from the fact that the distributions in Figure 8 are quite close.

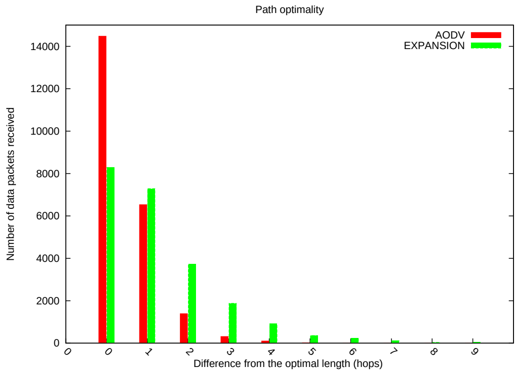

Figure 8: Distribution of path length discrepancy from optimal.

6. CONCLUSION

In this paper, we described a scheme called *Expansion* as an extension to AODV. The scheme tries to prevent link breaks on active connections by taking preemptive actions against them. The advantages of the proposed scheme are that it decreases overall control traffic overhead of the network while maintaining packet delivery fraction. The scheme pays for these improvement through a degradation in path optimality, which is seen to be quite modest. These conclusions were validated by extensive simulations using ns-2.

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.

7. REFERENCES

[1] 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*, pages 25–30, New York, NY, USA, 2004. ACM.

[2] C.Perkins, E.B-Royer, and S.Das. Ad hoc on-demand distance vector (aodv) routing. Internet Engineering Task Force, 2003.

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

[4] L. Qin and T. Kunz. Pro-active Route Maintenance in DSR. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):79–89, 2002.

[5] T. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[6] 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*, 4(3):172–180, 2008.