PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

A cross-layer predictive routing protocol for mobile ad hoc networks

Marco Carvalho, Robert Winkler, Carlos Perez, Jesse Kovach, Steve Choy

Marco Carvalho, Robert Winkler, Carlos Perez, Jesse Kovach, Steve Choy, "A cross-layer predictive routing protocol for mobile ad hoc networks," Proc. SPIE 6981, Defense Transformation and Net-Centric Systems 2008, 69810O (3 April 2008); doi: 10.1117/12.777750

SPIE.

Event: SPIE Defense and Security Symposium, 2008, Orlando, Florida, United States

A Cross-layer Predictive Routing Protocol for Mobile Ad hoc Networks

Marco Carvalho[1], Robert Winkler[2], Carlos Perez[1], Jesse Kovach[2] and Steve Choy[2]

[1]Institute for Human and Machine Cognition, 40 S. Alcaniz St., Pensacola, FL, USA
[2]U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD, USA

ABSTRACT

In this paper we introduce a cross-layer mechanism to enhance the effectiveness of link-state routing protocols by providing a localized estimate of the validity of a given route. Accurate predictions of route staleness result in less packet loss by selecting alternative paths before primary path failures. We present a proof of concept implementation that utilizes shared location and velocity information provided from each node through a common cross-layer substrate. Simulations using random-walk mobility models are presented for the OLSR protocol. The results show that predictive route adaptation can significantly reduce the average packet loss, average delay and jitter.

Keywords: Cross-layer routing, OLSR, MANET, Predictive Routing, NS-2, POLSR

1. INTRODUCTION

Many MANET routing strategies suggested for tactical networks have, at some level, relied on link state-based routing protocols. Following the trend of protocols such as Fisheye OLSR, HSLS and MALSR, new variations of this class of protocols are likely to continue to play a prominent role in both the development and deployment of military and civilian tactical/edge networks.

A basic underlying assumption in most link state protocols is that each node can locally determine its one-hop neighbors and the quality of their associated links. Beaconing is commonly used for this purpose. In highly dynamic environments, however, estimates based on periodic beaconing are likely to suffer from both type I and type II errors. False positives result when the neighbor list includes nodes which have moved out of range since the last beacon. And false negatives result when nodes not detected by the last beacon have since come into range. Although most protocols provide contention and route recovery mechanisms to address these errors, there is always a performance cost associated with the recovery.

In this paper we introduce a cross-layer mechanism to enhance the effectiveness of link-state protocols by providing a localized estimate of the validity of a given route. The proposed strategy relies on predicting the rate of change in distance between nodes in order to build (based on theoretical propagation models) the likelihood of packet delivery success before deciding on the best route. Accurate predictions are shown to directly result in less packets dropped because alternative paths can be selected before the failure of primary paths. Although demonstrated in the context of routing protocols, this strategy can also be applied in overlay networks and peer-to-peer applications in dynamic environments.

Our local estimation of the accuracy of the topology is a computationally efficient algorithm based both on information contained in multiple layers in the communications stack (i.e., signal strength and fading) and/or explicit information provided by the platforms (e.g., position and velocity estimates). We present a proof of concept implementation that utilizes location and velocity information provided by each node through a common cross-layer substrate. Empirical tests are conducted in NS-2 simulations using random-walk mobility models.

Defense Transformation and Net-Centric Systems 2008, edited by Raja Suresh
Proc. of SPIE Vol. 6981, 69810O, (2008) · 0277-786X/08/$18 · doi: 10.1117/12.777750

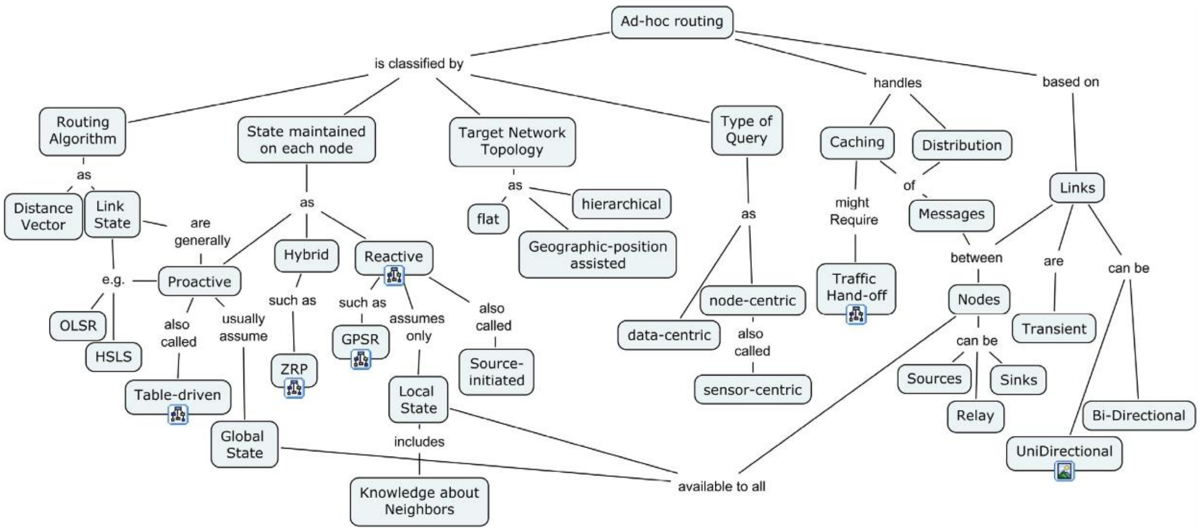

Figure 1. Classes of ad hoc routing protocols.

2. ROUTING IN MOBILE AD HOC NETWORKS

Routing is one of the most investigated networking services in mobile ad hoc networks. Currently there are a large number of routing protocols with different characteristics and capabilities. Out of the types described and classified in Figure 1, proactive link-state based routing protocols represent a class widely used in small to medium scale deployments.

OLSR[1] implements a link state protocol using optimized strategies for topology control, maintenance, and sharing. In its original form, OLSR maintains global topology information at each node which allows for routing decisions to be made locally beforehand (proactive routing). Proactive strategies, however, come with an overhead cost for the periodic monitoring, maintenance and sharing of node state and neighborhood information. Such overhead in proactive protocols like OLSR, generally constitute a limiting factor in the scalability of the protocol, constraining their application to small or medium scale networks. Variations of OLSR such as Fisheye OLSR[2] and hybrid approaches[3] have been proposed to improve scalability and reduce overhead. But any proactive strategy will always have to balance the accuracy of local routing information with the overhead induced by the protocol both for detection and maintenance.

In this work we propose a general enhancement for link state routing protocols. The goal is to allow such protocols to build, at low cost, an estimate of the validity of their neighborhood and routing information.

Recognizing the probability of success of using a specific route, allows the protocol to a) decide when to update and share its own neighborhood and state information, and b) to better make routing decisions proactively, by choosing secondary routes in lieu of primary routes that are "about to fail".

While we recognize the importance of using network stability metrics to, for example, adjust and direct the protocols beaconing, we reserve that for future work and focus in this paper on informing the routing decisions via short-term predictions of change in the local topology of a node.

3. RELATED WORK

A number of authors have previously proposed the use of predictive strategies for optimizing standard routing protocols. Baras and Mehta[4] have proposed the use of component based routing as adaptation mechanisms to changing network conditions and requirements. The central idea is to allow protocols to be "assembled" from elemental components (potentially at run-time) to accommodate environmental requirements and conditions.

Su and Veciana[5] proposed predictive routing protocols to enhance QoS for data streams. The protocol focuses primarily on proactive resource allocation based on load information shared by each node. Predictions about resource availability conditions are used to influence route calculation to comply with QoS requirements. Huang and Baras[6] also proposed the use of probabilistic routing for run-time adaptation to changes in topology and resource availability. That work, however, focuses on swarm intelligence strategies for probabilistic routing and uses predictive strategies only as part of its online learning algorithm.

Predictive location based routing protocols such as CBLR[7] also leverage the notion of predicting the future location of nodes in the networks. The focus of such protocols, however, is on building location estimates for other nodes in order to support geographical-based routing.

Other geographic position-assisted routing protocols such as GFG[8] and GPSR[9] also leverage location information from other nodes to make routing decisions. This class of protocols, however, generally use geographical location as a replacement for the addresses of a node, and often rely on greedy strategies for packet forwarding. Although not necessarily predictive by design, geographic position-assisted protocols do inherently provide position sharing capabilities that can be naturally extended by predictive routing strategies.

Position information can be generally obtained from GPS devices or other mechanisms such as dead reckoning and signal strength triangulation.[10] Although widely varying in terms of resolution and accuracy, the effective capability of each of these approaches is fundamentally the same. Variations of these classes of protocols that do not necessarily rely on geographical positions have also been proposed[11,12] where virtual coordinates are assigned to each node in the network based on a few reference points.

4. PREDICTIVE LINK STATE ROUTING PROTOCOLS

In this work, we argue that link state routing protocols can be generally enhanced with predictive mechanisms. In order to illustrate our claims, we propose an extension to the OLSR routing protocol that implements a predictive component to reduce packet loss and improve average delivery delay and jitter. In this paper, we call this POLSR (for Predictive OLSR).

OLSR is a link state routing protocol optimized for mobile ad hoc networks. As in most link state protocols, OLSR-enabled nodes rely on the periodic UDP broadcast of *hello* messages for neighborhood discovery and maintenance. Local changes in link state (both availability and quality) are propagated through topology updates to all nodes in the network, ideally allowing each node in the network to locally form a global view of the network topology. Controlled flooding of the topology updates (or other messages) is accomplished through the use of special flood relay nodes called multi-point relays (MPR). MPRs are used to relay flood messages from specific nodes, greatly minimizing the number of retransmissions without compromising the requirement for full network coverage.

The proposed POLSR algorithm extends the OLSR protocol by providing, at each node, a metric that indicates the probability of success of packet delivery to each of the nodes in its current routing table. In highly dynamic environments, instead of increasing the frequency of *hello* updates (and consequently the overhead) of the OLSR protocol, we propose to allow nodes to determine the probability of successful delivery when using a particular route. Such an approach allows nodes to optionally choose alternative routes in lieu of "optimal" OLSR routes that have a low probability of successful packet delivery.

4.1 The POLSR Protocol

The main idea behind the protocol is to allow nodes to predict the success of using a specific neighbor as the next hop (route) for a given target. The prediction algorithm involves two parts: a) the estimation of short-term changes in the physical location of neighbors, and b) the estimation of the probability of success of using each specific neighbor as the next hop for a route.

For the first part, position estimates are obtained from the physical coordinates and estimated velocity vectors periodically reported by each node. Node position and velocity estimates are distributed as four additional bytes added to the periodic *hello* messages used for neighborhood maintenance. From the periodic *hello* messages received from its neighbors, each node gathers the velocity vectors and the current positions of those nodes. Since *hello* messages are asynchronous, each message is associated with a time stamp upon receipt. The location reported by each node allows for the estimation of the relative distance between a node and each of its neighbors

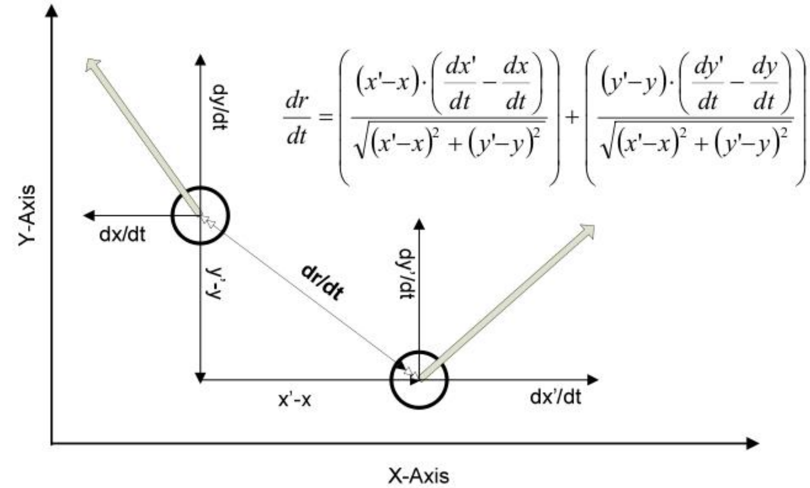

Figure 2. Distance-change rate estimation.

at the time the packet was received. The velocity vector reported with the location is used for estimating the temporal rate of change in relative distance.

As illustrated in Figure 2, the rate of change in distance is estimated under the assumption that nodes are moving in a linear trajectory with a constant speed. Such an assumption is reasonable if we consider that the rate change estimates are periodically corrected by subsequent *hello* messages. Drifting errors are small and, most importantly, bound to the maximum interval between *hello* messages and the maximum speed of a node. Within periodic *hello* messages, the projected location of each neighbor can be easily and efficiently estimated from the previously reported position and the rate of change of the relative distance between nodes. Because *hello* messages are asynchronous in the network, local connectivity (i.e. neighborhood information) inferred from such messages is dependent on the time elapsed since the last message from that node was received and the distance, speed and direction of each previously reported neighbor. Under the previously stated assumption that, within the *hello* interval, the speed and direction of the nodes are relatively constant, each node can estimate the current position of each of its neighbors based on the information shared in the previous *hello* message.

For the second part of the prediction algorithm, the fundamental task is to estimate the likelihood of reaching a node given its projected location and translational speed. For that, POLSR relies on theoretical RF propagation models to determine connectivity. In this work we use the shadowing propagation model. In particular, we utilize the implementation available as part of the NS-2 network simulator. The shadowing propagation model defines the signal strength at the receiver side through a log-normal random variable. To determine the probability of a transmitter reaching a certain distance, the received signal strength at the target distance is estimated and compared against a cutoff threshold.

If P_r is a random variable representing the signal strength at the receiver and $RxThresh$ is the reception threshold, the probability of receiving data at the receiver is the probability that the receiving power P_r is greater than $RxThresh$, that is $\mathrm{P}(P_r > RxThresh)$. Such probability is the same as 1- $\mathrm{P}(P_r \leq RxThresh)$, which can be calculated from the cumulative distribution function of P_r, which is log-normally distributed.

In our simulations, we have used the following random variable shown in equation 1 to represent the power at the receiver.

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d_0^2 L} \cdot \left(\frac{d}{d_0}\right)^{-\beta} \cdot 10^{X_{dB}/10}. \tag{1}$$

s In Equation 1, P_t is the transmission power, while G_t and G_r are the antenna gains for the transmitter and receiver respectively; L is the system loss, λ is the wavelength, d is the distance between nodes, d_0 is the close-in distance, β is the path loss exponent, and X_{dB} is a random normal variable with zero mean and standard deviation σ_{dB}.

From equation (1), the average and standard deviation estimates for the log of the power at the receiver ($log(P_r)$) are given by equations (2) and (3), respectively.

$$\mu = \log\left(\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d_0^2 L}\right) - \beta \log\left(\frac{d}{d_0}\right). \tag{2}$$

$$\sigma^2 = \left(\frac{\sigma_{dB}}{10}\right)^2. \tag{3}$$

5. EXPERIMENTAL EVALUATION

For the experimental evaluation of the predictive routing algorithm we have primarily relied on NS-2 simulations using the OLSR extension provided by Ros.[13] For our simulations, the standard OLSR *hello* messages were modified to include information about node location and velocity. For that, 4 additional bytes were added to the header of each *hello* message. Information about local position and velocity were assumed to be accurate and available at each node. For our tests this information was directly obtained from the mobility models.

We have also added an additional data structure to each node to maintain the last update information received from each of the nodes neighbors. The neighborhood table holds, for each neighbor, the time stamp of the last received *hello* message and the last reported location and velocity for that node. (Note that for efficiency purposes, some other derivable information is also maintained in that table in our implementation.)

When receiving a *hello* message from a neighbor, each node updates its neighborhood table by calculating the relative distance to the neighbor and the rate of change in distance as a function of time.

When updating routing information based on local topology, each node calculates the projected distance to every neighbor from their previously reported location and distance change rate estimates.

The projected distance is used with the shadowing propagation model to build an estimate of the likelihood of the node being reachable at that time, given the current transmission power. If the probability of the node being reachable is lower than a certain threshold, the neighbor is marked as non-preferred. In our implementation, we have chosen to represent this by setting the link to the node as non-symmetric, which will remove it from the MPR list for the node and disregard it as a potential next hop for routes. This strategy allows us to simply re-use the route update logic already implemented as part of the OLSR protocol.

In order to estimate delay and jitter, every message generated by the NS-2 agent was time-stamped. This modification is mainly for instrumentation purposes and provides no support to the predictive functionality. Packet delivery delay in our simulations was measured as the difference between the current simulation time (based on time reported by the simulators scheduler) and the timestamp of the packet.

The simulation environment consisted of a 30 node network randomly distributed (uniform distribution) in an area 1000 meters squared. The size of the area was chosen to minimize the probability of disconnected nodes or segmented clusters at the beginning of the simulation.

The nodes are randomly divided into three groups (again using a uniform distribution). We made no attempt to ensure that the groups were of the same size. Given a maximum speed for each simulation, we allocated the speed for each group as follows:

1. Group 1: All nodes moving at a constant speed equal to 10% of the maximum simulation speed,
2. Group 2: All nodes moving at a constant speed equal to 60% of the maximum simulation speed.
3. Group 3: All nodes moving at the maximum simulation speed.

Based on the geometric constrains of the simulation, we estimated a maximum allowable speed that would be slow enough to capture the transition of nodes between *hello* messages. Essentially, we estimated the communications range of each node and, given a fixed interval for periodic beaconing, we estimated the maximum traceable speed of a node. That value was estimated to be 90 meters per second, which we defined as our boundary test condition.

We also arbitrarily chose two other levels of maximum speed for testing: 10 and 30 meters per second. For each of these test conditions, the fixed speed for the nodes at each of the three groups was calculated as shown in table 1.

Following a random walk model, given a speed, nodes choose a random destination in the square area (uniform distribution) and start moving. Upon reaching the destination point, each node immediately chooses another destination (again through a uniform random distribution), and continues moving to that location.

Table 1. Simulation speeds for each group of nodes

Max. Speed	Group 1	Group 2	Group 3
10 m/s	1 m/s	6 m/s	10 m/s
10 m/s	3 m/s	18 m/s	30 m/s
10 m/s	9 m/s	54 m/s	90 m/s

Five different nodes are randomly selected as data sources for traffic. Each of these nodes chooses a target receiver (which will be the same for the life of the simulation) and generates 2 packets of 256 bytes per second which are transmitted with no acknowledgement of retransmission to the target node. We ensure, at each run, that data senders and data receivers are represented by different nodes in the simulation.

The *hello* interval for all nodes is 2 seconds and their transmission power is 366 mW, which enables a communications range of approximately 200 meters, on average.

The simulation runs for 5 minutes, which results in the generation of 600 messages (or data packets) from each of the senders. Logging and tracing is enabled at each receiver in order to track the number of messages successfully received in each of the streams. Messages are unreliable and not sequenced, so any message received from the sender (regardless of order) counts as a successful transmission.

For our simulations, the threshold chosen for deciding the validity of a route was 50%. That is, if the calculated probability of reaching a neighbor is less than 50%, the routes through that node are discarded in lieu of secondary alternatives (assuming they exist).

To visualize and illustrate the simulation traces we have used iNSpect.[14] The output trace of NS-2 was modified to also include probability estimations about the trustworthiness of each nodes routing table. That information was exposed to the visualizer to graphically illustrate the quality of each nodes routing table.

In Figure 3, we illustrate an example showing how the quality of the routing information quickly becomes stale and potentially wrong as a function of the dynamics of the network.

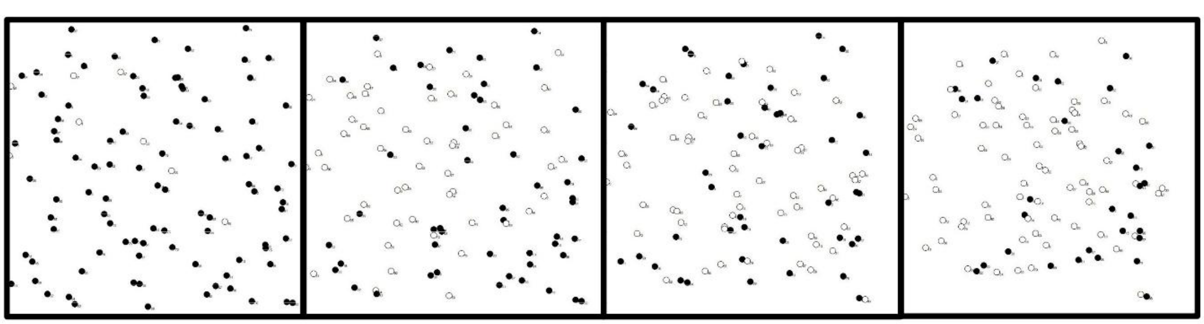

Figure 3. Mobility-induced routing table mismatches.

In the figure, the color of the nodes (from black to white) indicates how inaccurate the nodes table is in relation to the underlying true RT topology (obtained directly from the simulator).

One interesting phenomena observed in these tests is the cumulative effects of route mismatch. Although difficult to represent in the still images shown in Figure 3, the issue emerges from the fact that, as expected, the accuracy of routing information from a node will progressively degrade over time between periodic *hello* messages from its peers.

However, even when *hello* messages are received, the up-to-date information represents only the link to the sender, which is potentially in bad shape itself relying on an outdated routing table. The immediate consequence is that nodes featuring frequent *hello* broadcasts (short interval between messages) may be greatly favored as next hop options by its neighbors and yet, be extremely ineffective for message forwarding. Their own routing table accuracy is not a function of their *hello* interval, but instead, a function of the interval chosen by its peers.

In this paper we dont explore this specific characteristic any further, but we do submit that a small variation of the proposed predictive protocol may be effective in addressing this issue. We will explore this possibility in our future work.

6. EXPERIMENTAL RESULTS AND ANALYSIS

For this paper, we have primarily focused on the performance gains obtained with the proposed prediction strategy. Performance gains, in this context, will be measured in terms of average packet loss, average delay and average jitter measured for each of the test conditions.

The average packet loss in the experiments are intentionally high, varying from approximately 30% to 75% under different mobility and topology conditions.

The results shown in Figure 4 illustrate the significant improvement (dashed) in packet loss reduction under different levels of maximum speed. Recall that all nodes are mobile, however one third of the nodes moves at 10% of the maximum speed, while second third moves at 60% of the maximum speed and the remaining nodes (i.e. 1/3 of all nodes) moves at the maximum speed.

In the example shown in Figure 4, the top solid lines show the lower and upper limits of the confidence interval of the average packet loss for OLSR. The lower (dashed) lines show the same interval for POLSR, using the 50% threshold previously described.

As the figure shows, the improvement provided by the predictive component is statistically significant in every case, showing better improvements at lower mobility rates.

At low maximum speeds (10 m/s), the improvement of the predictive component in terms of packet loss is approximately 86%. If the maximum speed is increased to 30 m/s, the improvement in packet loss reduction is 65%. And at a maximum speed of 90 m/s the gain in packet loss reduction is 10.6%.

The decrease in relative improvement is explained by the fact that, at high mobility rates, the changes in topology may be so rapid that even the secondary routes are no longer available to the point that they cannot be used as a replacement for fast leaving neighbors. This is a limitation of our implementation. While POLSR can predict the fact that neighbors will become unavailable, in its current implementation it cant anticipate the arrival of new nodes about to be discovered.

A similar improvement is illustrated in terms of the reduction of average packet delay (see Figure 5). The improvements in delay are somewhat unexpected given that POLSR tends to use less optimum routes in lieu of primary (but possibly invalid) routes.

The observed reduction in average delay is essentially due to the conservative approach used by the protocol. Because route estimates are calculated based on 2 second neighborhood updates, the replacement of a previously preferred route generally lags in relation to the underlying changes in topology. That effect is minimized in POLSR because fast moving nodes, or nodes close to the communication range boundaries, are less desirable by default.

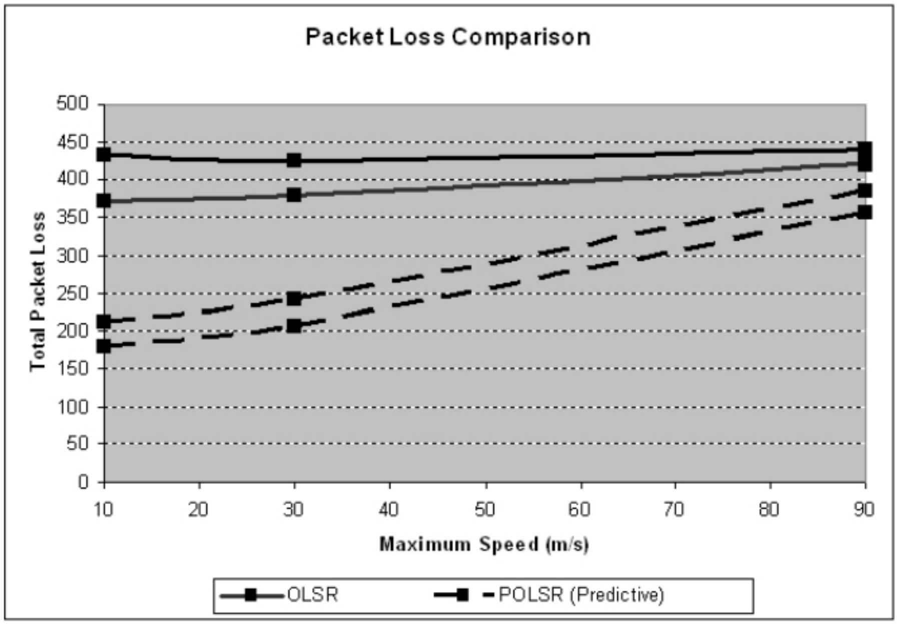

Figure 4. Confidence Intervals for the average packet loss in a) OLSR (solid) and b) predictive OLSR (dashed).

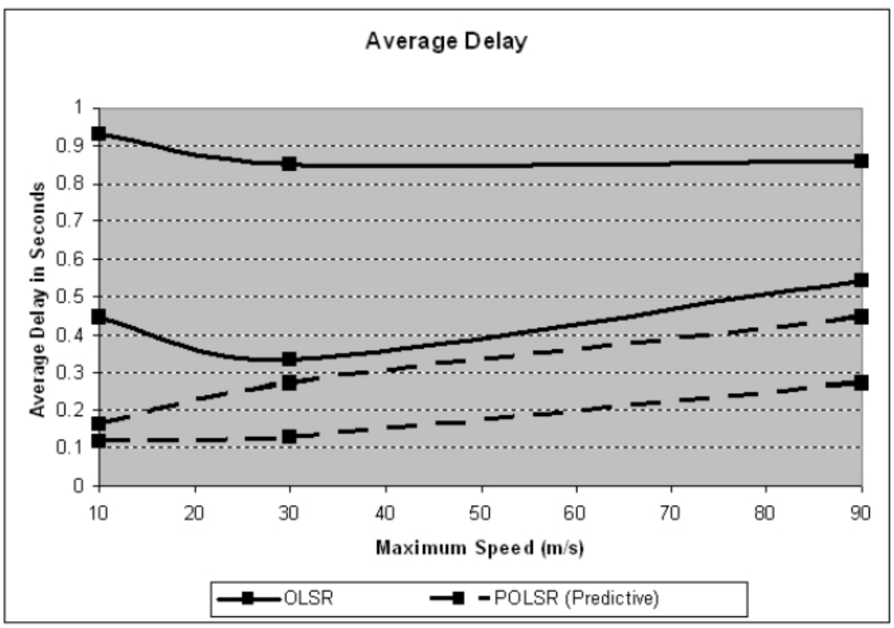

Figure 5. Confidence intervals for average delay in a) OLSR (solid) and b) predictive OLSR (dashed).

Similar improvement results are also observed in the reduction of jitter (Figure 6). The average jitter is significantly improved in POLSR again because of the conservative nature of the protocol, which tends to discard boundary options in lieu of more stable (and repeatable) alternatives.

The downside of choosing secondary routes in lieu of their primary alternatives is essentially cost. Primary routes are chosen because of a link cost estimate in OLSR that estimates (using Dijkstras single-source shortest path algorithm, typical of most link state-based protocols) the cost of one path versus another. The cost of a link in OLSR is estimated based on missed *hello* messages which is believed to affect the packet loss (and delay) of data traffic sent through that link. Our simulations, however, have shown that such metrics may fail to accurately reflect packet loss, delay, and jitter. From our observations, the stability of the route in terms of its persistence seems to be a better indicator of quality for these measures.

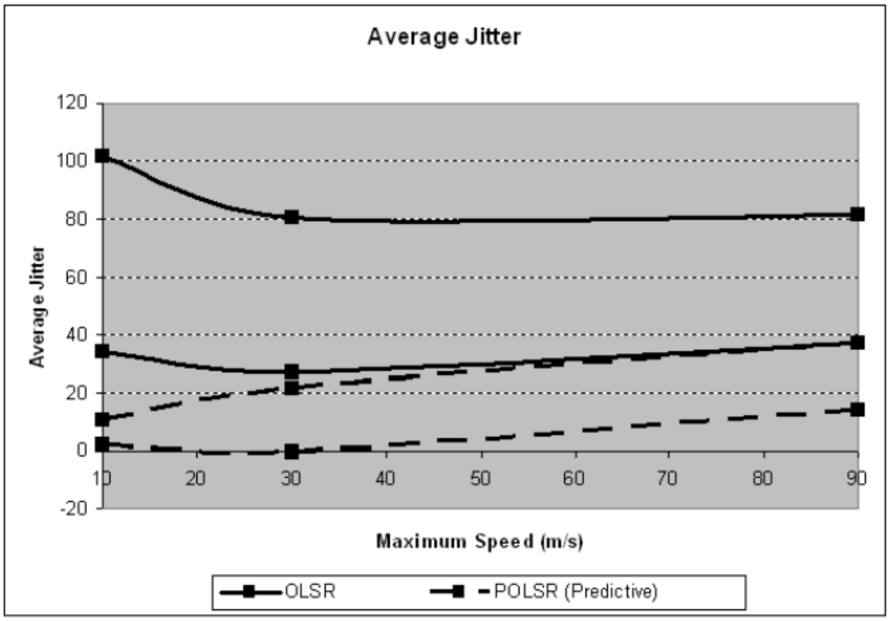

Figure 6. Confidence intervals for average jitter in a) OLSR (solid) and b) predictive OLSR (dashed).

7. CONCLUSIONS AND FUTURE WORK

We assume, in this work, that reachability is primarily a function of distance, which is a very simplistic assumption in highly contentious environments. However, the methodology proposed here is not directly dependent on this assumption. The underlying capability required for the operation of the protocols is the probabilistic projection of future system state in the short term (i.e. within the average *hello* interval).

In this work, we have demonstrated the applicability of this principle using a simple connectivity model defined by a well-known propagation model based on distance. In order to accommodate other environmental effects, short-term historical information about the link can be added to protocols, allowing for a more accurate estimation of future connectivity. This aspect is part of our future research.

Another weakness of the proposed algorithm is the assumption that position and velocity of a node are readily available. This requirement implies the availability of a local GPS device (or other external reference system). Although this assumption is often valid in many tactical networks, its requirement by the protocol is generally limiting. But recall that position and velocity is only locally shared by each node to build an estimate of the rate of change in distance. We argue that similar information can be obtained by monitoring short term trends on the quality of links to each node. Our cross-layer infrastructure,[15][16] provides the mechanism to locally collect and share such information, and implements a proof-of-concept version of POLSR. As part of our future work, we are also exploring mechanism to ensure that such information can be obtained at low cost (i.e. without requiring permanent communication between nodes). Once implemented, this approach will be evaluated and compared with GPS-enabled protocols.

8. ACKNOWLEDGEMENTS

This work is supported in part by the U.S. Army Research Laboratory under Cooperative Agreement W911NF-04-2-0013, and the Advanced Decision Architectures Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009.

REFERENCES

1. T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol." Retreived on October 18, 2003 from http://hipercom.inria.fr/olsr/draft-ietf-manet-olsr-11.txt, 2003.
2. M. Gerla, X. Hong, and G. Pei, "Fisheye state routing protocol (FSR) for ad hoc networks." Internet Draft (expired), http://www.watersprings.org/pub/id/draft-ietf-manet-fsr-03.txt, 2002.
3. V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 303–314, ACM, (New York, NY, USA), 2003.
4. J. Baras and H. Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc networks," in *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, March 3-5 2003.
5. X. Su and G. de Veciana, "Predictive routing to enhance QoS for stream-based flows sharing excess bandwidth," *Computer Networks* **42**(1), pp. 65–80, 2003.
6. H. Huang and J. S. Baras, "Component based routing: A new methodology for designing routing protocols for MANET," in *25th Army Science Conference*, (Orlando, FL), 2006.
7. R. A. Santos, R. M. Edwards, and N. L. Seed, "Using the cluster-based location routing (CBLR) algorithm for exchanging information on a motorway," in *The 4th IEEE International Conference on Mobile and Wireless Communications Networks (MWCN)*, pp. 212–216, (Stockholm, Sweden), 2002.
8. P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks* **7**(6), pp. 609–616, 2001.
9. B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, ACM, (New York, NY, USA), 2000.
10. S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 230–241, ACM, (New York, NY, USA), 2005.
11. D. S. J. De Couto and R. Morris, "Location proxies and intermediate node forwarding for practical geographic forwarding," Tech. Rep. MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.
12. A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 96–108, ACM, (New York, NY, USA), 2003.
13. F. J. Ros, "UM-OLSR." Available online at http://masimum.dif.um.es/um-olsr/html/, 2007. Accessed on May, 2007.
14. S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, "A visualization and analysis tool for NS-2 wireless simulations: iNSpect," in *MASCOTS '05: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 503–506, IEEE Computer Society, (Washington, DC, USA), 2005.
15. M. Carvalho, N. Suri, M. Arguegas, M. Rebeschini, and M. Breedy, "A cross-layer communications framework for tactical environments," in *MILCOM 2006*, 2006.
16. M. Carvalho, N. Suri, V. Shurbanov, and E. Lloyd, "A cross-layer network substrate for the battlefield," in *The 25th Army Science Conference*, (Orlando, FL), 2006.