Evaluating Ad hoc Routing Protocols With Respect to Quality of Service

John Novatnack,
Lloyd Greenwald
and
Harpreet Arora

Technical Report DU-CS-04-05
Department of Computer Science
Drexel University
Philadelphia, PA 19104
October 2004

Evaluating Ad hoc Routing Protocols With Respect to Quality of Service

John Novatnack, Lloyd Greenwald and Harpreet Arora
Intelligent Time-Critical Systems Laboratory
Computer Science Department, Drexel University
3141 Chestnut Street
Philadelphia, PA, 19104
http://itcsl.cs.drexel.edu

***Abstract*— The ability of a Mobile Ad Hoc Network(MANET) to provide adequate quality of service (QoS) is limited by the ability of the underlying routing protocol to provide consistent behavior despite the inherent dynamics of a mobile computing environment. In this paper we study three MANET routing protocols, OLSR, DSR and AODV, with an emphasis on the effect they have on various QoS metrics. We describe and analyze how the protocols differ in the mechanisms they use to select paths, detect broken links, and buffer messages during periods of link outage. The effects of these differences are quantified in terms of packet delivery ratio, end-to-end hop count, end-to-end latency, and mechanism overhead. We show that the proactive protocol, OLSR, builds paths with consistently lower hop counts than the reactive protocols, AODV and DSR, a fact that leads to a reduction in end-to-end latency. The reduction in end-to-end latency assists a QoS model in meeting timing requirements and improves global network performance. We further show the impact of broken link detection latency on the packet delivery ratio. A routing protocol that can not quickly recover from link breakage caused by mobility renders a QoS model incapable of meeting delivery requirements. Finally, we analyze the effect of mobility on the distribution of end-to-end latencies. Traditionally, reactive protocols are criticized for buffering during the building of routes, however we also study buffering phenomenon caused by the proactive mechanisms of OLSR.**

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are wireless multi-hop networks characterized by a lack of centralization, dynamic topologies and unique interface characteristics. Routing protocols designed for MANETs must be able to operate on networks with low bandwidth and random mobility. In 1997, the Internet Engineering Task Force(IETF) [9] created a working group to deal with issues related to the complexity of building MANET routing protocols. The working group has since separated MANET routing protocols into two classes:

- Reactive (On-demand) Protocols
- Proactive Protocols

Reactive or on-demand protocols were designed to contend with the low bandwidth that exists on wireless mediums. They were aimed at decreasing the amount of control overhead by only initiating a request for a route when it is required. This allows the overhead of the routing protocol to scale to zero when no new routes are needed on the network; however, this mechanism creates a delay associated with the building of new routes. *Proactive* routing protocols periodically broadcast information that is sent across the network in a controlled flood. The information is used at each node to build a routing table. MANET proactive protocols must employ mechanisms to limit network wide flooding in order to reduce overhead. Recently, the IETF MANET group has recommended two reactive protocols (AODV [6] and DSR [2]) and two proactive protocols (OLSR [1] and TBRPF [12]) to be moved out of the research stage and toward the development of request for comments(RFCs).

Multimedia and military applications of MANET technology require explicit performance needs to be met. Multimedia applications are characterized by timing requirements that are necessary to provide seamless streaming of audio and video. Military applications involve critical battlefield information that must be guaranteed to arrive at its destination in a timely fashion. Building quality of service (QoS) frameworks for these applications require extensions to traditional QoS models. A MANET QoS framework must be able to find multiple hop paths with sufficient bandwidth and delay characteristics, despite network changes, low bandwidth links and shifting traffic patterns. Due to the dynamic nature of the environments where MANET technology is deployed a QoS framework operating without efficient performance from the other protocols in the network stack cannot meet service requirements. Specifically, the QoS model requires a routing protocol that is capable of providing consistent quality of performance in an environment with varying dynamics.

This paper describes how routing protocols can affect achievable QoS on MANETs. We look at how the routing protocols differ in achieving four metrics: (1) *packet delivery ratio* (2) *control packet overhead (packets and total bytes*, (3) *average hop count* and (4) *end-to-end latency*. These metrics have a direct impact on QoS requirements including: guaranteed delivery, guaranteed bandwidth, and guaranteed delay or latency.

In studying various routing protocols we identify and concentrate on three aspects of proactive and reactive protocols that can impact these quality of service metrics. The first focus is the **path selection mechanisms** employed by each proactive and reactive protocol. The second difference is the **mechanisms used to detect and repair broken links** (in coordination with the link layer). Routing protocols may detect broken links through mechanisms at the routing layer or

through *link layer feedback*. Link layer feedback is defined as notification, sent from the link layer to the routing layer, that a link to a neighbor has been broken. Finally, we focus on how **buffering** during link breakage affects guarantees made about transmission latency.

In this paper we analyze and compare two reactive and one proactive routing protocol: AODV with and without link layer feedback, DSR with link layer feedback, and OLSR with and without link layer feedback. We compare the performance of the protocols on random movement scenarios. Our experimental data was gathered using the NS-2 [11] network simulator with CMU wireless extensions. NS-2 uses an implementation of the IEEE 802.11 [7] standard as the link layer for wireless network simulations.

The rest of the paper is organized as follows: Section II outlines related work that has been done in the past comparing MANET routing protocols. In section III we present a summary of the three MANET routing protocols we have tested and discuss our unique work in analyzing the mechanisms of the protocols. In section IV we present a description of the simulation environment used in our experiments, and present our results concerning the effect of each mechanism on performance in section V. We finally conclude in section VI.

II. RELATED WORK

Early analysis of MANET routing protocols done by the Monarch team at CMU [3] compares the reactive protocols DSR and AODV to a proactive protocol DSDV[1]. However, due to its inefficiencies DSDV is no longer considered a viable option for a standardized MANET routing protocol.

Das et al. [4] compare reactive protocols AODV and DSR with the link-state protocol OSPF. Again, this paper presents a comparison between proactive and reactive protocols, but OSPF is not been modified to handle the mobility of a wireless environment. Although the paper reaches the conclusion that reactive protocols suffer from suboptimal paths and proactive protocols require a larger amount of bandwidth, it is not an accurate comparison of reactive protocols with proactive protocols that have been designed specifically for MANETs.

More recently, Das et al. [5] compare the reactive protocols AODV and DSR. This paper considers the differences between the mechanisms of AODV and DSR, but does not attempt to compare them with proactive protocols. The paper does not consider implementations of the protocols where link errors are detected using routing protocol mechanisms.

Laouti et al [8] study OLSR alone on a random scenario. The study does not attempt to compare the performance of OLSR with any reactive protocols. Jacquet et al. [13] explore the differences between the reactive and protocol mechanisms of DSR and OLSR. The numerical analysis of OLSR against DSR gives insight into the methods each protocol implements in distributing control messages, but does not give a thorough picture of how the differences affect overall performance.

[1]A fourth protocol, TORA, was also studied, but is not mentioned as it is still in a research stage.

III. SUMMARY AND ANALYSIS OF PROTOCOLS

We now present a summary and analysis of the mechanisms employed by AODV, DSR and OLSR. We study the mechanisms and provide timing estimates of the events that occur during the operation of the routing protocols. The Internet drafts of the three protocols ([2] [6] [1]) provide details on the full functionality of the protocols.

A. DYNAMIC SOURCE ROUTING PROTOCOL (DSR)

DSR is a reactive, source initiated routing protocol. It employs two phases, *Route Discovery* and *Route Maintenance*, to build and maintain active routes between sources and destinations. During the route discovery phase DSR discovers routes to previously unknown destinations in the network. When a traffic flow to an unknown destination is initiated, DSR uses expanding ring search to broadcast a route request. The header of the route request packet contains a record of the path the packet has followed. Before a node forwards the route request it appends it's address onto the path in the header. If and when the destination receives the route request it generates a route reply and sends it to the source along the reverse of the path stored in the header of the route request. Unique to DSR is the use of a route cache in each node to store routes to destinations. Each time a node receives or forwards a route request it updates it route cache with the path stored in the route request. DSR uses the route cache to store the entire path to a destination, not just the next hop. This allows the routing protocol to store and make use of redundant paths. Each data packet that is sent by a source includes the entire path in the header of the packet. This leads to an increase in the overhead of each data packet. Storing the entire path to a destination in the route cache also leads to a large memory requirement at each node. Despite this, the route cache allows nodes to store more information about the network. Knowledge about the network helps a node to find alternate paths to destinations using information stored in the route cache. DSR employs a number of optimizing mechanisms that make use of the route cache.[2]

During the route maintenance phase, the routing protocol repairs and maintains routes that were constructed during the route discovery phase. When an intermediate node attempts to forward a data packet to the next hop and becomes aware that the link is broken, it generates a route error packet and unicasts it back to the source. Each node that forwards the route error message removes the path from their route cache. After the source receives this packet it removes the path in its route cache and tries to find an alternate path to the destination, once again entering the route discovery phase. Although the DSR Internet Draft provides a means to detect broken links at the routing layer, DSR has not been studied in any previous work without link layer feedback. DSR's Internet Draft [2] outlines a mechanism named *passive acknowledgment* that may be used to detect broken links at the routing layer. In this paper we only consider DSR with link layer feedback. Implementation and testing of passive acknowledgment are a subject of future work.

[2]Refer to DSR's Internet Draft [2] for details

B. AD HOC ON-DEMAND DISTANCE VECTOR ROUTING PROTOCOL (AODV)

AODV is a reactive, source initiated, distance vector routing protocol. AODV's route discovery phase shares functionality with DSR. A source initiated route request is broadcast with expanding ring search across the network until it is received by the destination. The major difference between the two protocols is that AODV is a distance vector routing protocol that only stores the next hop information in its routing table. This allows for a smaller packet header size and routing table, but does not allow AODV to have access to beneficial information about the network. Similarly, the route maintenance phase of AODV operates like that of DSR. Detection of link breakage by an intermediate node in AODV causes a route error message to be generated and unicast back to the source.

ALLOWED_HELLO_LOSS	3
HELLO_INTERVAL	1 s
DELETE_PERIOD	4.5 s

TABLE I

AODV PARAMETERS USED IN OUR ANALYSIS

AODV allows for broken links to be detected using either link layer feedback or with mechanisms at the routing layer. The latter is accomplished through the use of periodic *HELLO* packets that are generated and broadcasted by each node in the network. Each node sends *HELLO* packets at a periodic *HELLO_INTERVAL*. If a node does not receive a *HELLO* packet from its neighbor in some *DELETE_PERIOD* amount of time, it assumes that the link to the neighbor is down and removes the associated table entry. From AODV's Internet Draft [6] we define the DELETE_PERIOD as:

$$DELETE\_PERIOD = 1.5 \cdot ALLOWED\_HELLO\_LOSS \cdot HELLO\_INTERVAL$$

Table I displays the parameters used in our analysis as well as the duration of our route delete period.

C. OPTIMIZED LINK STATE ROUTING PROTOCOL (OLSR)

OLSR is a proactive link state routing protocol. Each node using OLSR periodically broadcasts its routing table so that each node can have a complete view of the network. In doing so, it incurs a large control overhead. The biggest concern for a proactive protocol is to reduce the amount of periodic control overhead. OLSR addresses this concern by limiting the number of nodes that forward network-wide traffic. This is accomplished through the use of *multi point relays* (MPRs). A MPR is a node that is responsible for forwarding routing messages. Each node independently elects a group of MPRs from its one hop neighbors. MPRs are chosen by a node such that it may reach each two hop neighbor via at least one MPR. The nodes that have been selected as MPRs are responsible for forwarding the control traffic generated by that node. Figure 1 shows the MPR selection process. Node 2 first announces its presence to node 1. Node 1 then notifies node 0 of its new one hop neighbor. If node 0 previously did not have access to node 2 then node 0 chooses node 1 as a MPR. Then node 1 is responsible for forwarding control traffic generated by node 0. OLSR employs the following forwarding rule: control traffic received from a previous hop is forwarded only if that previous hop has selected the current node as a MPR. Through the use of MPRs OLSR is able to reduce the amount of control traffic in the network. It is shown in [14] [15] that MPRs reduce the amount of overhead without degrading network performance.

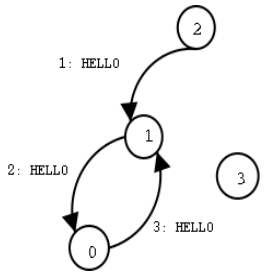


Fig. 1. OLSR Multi-Point Relay(MPR) Selection Process

The two primary control messages used by OLSR are the *HELLO* message and the *topology control* (TC) message. The *HELLO* message is broadcast to each one hop neighbor and includes: a list of one hop neighbors, a list of two hop neighbors, a list of nodes that it has selected as a MPR and a list of nodes that have selected it as a MPR. *HELLO* messages are never forwarded. *Topology Control* (TC) messages contain a list of all the nodes that have selected the sender as a MPR. They are forwarded across the network using the forwarding rule stated above.

HELLO_INTERVAL	1s $\pm$ 1s
TC_INTERVAL	4s $\pm$ 1s
NEIGHBOR_HOLD_TIME	6 s

TABLE II

OLSR PARAMETERS USED IN OUR ANALYSIS

By following the forwarding rule OLSR is able to reduce the control overhead, but it may increase delay in route table convergence. Consider the network shown in Figure 2. Let us assume that node 0 and node 1 are aware of each other and node 3 and node 4 are aware of each other. Also node 0 and node 1 are disjoint from node 3 and node 4. We now consider the case where node 2 moves between the four nodes creating a path from node 0 to node 4. Intuitively when node 3 generates a TC message, node 0 would receive it and become aware of the path to node 4 with a maximum delay of the TC interval (TC_INTERVAL). This does not happen because node 2 does not forward node 3's TC message until node 3 has chosen it as a MPR. Consider the following events that must occur before the route is created.

- **Event 1:** Node 1 broadcasts a *HELLO* message which is received by node 2. This message notifies node 2 that it

has a new one hop neighbor, node 1, and a new two hop neighbor, node 0
- **Event 2:** Node 2 sends a *HELLO* message notifying node 3 of node 0 and node 1. Since node 3 previously could not access node 1, it chooses node 2 as a MPR.
- **Event 3:** Node 3 sends a HELLO message notifying node 2 that it has chosen it as a MPR.
- **Event 4:** Node 3 generates a TC control message which is forwarded back to node 0. This TC message does not get forwarded back to node 0 until node 3 has alerted node 2 that it has been chosen as a MPR.

With the parameters used in our analysis this sequence of events has a maximum delay of 11 seconds. Long routing convergence times can be expected when the network is first initialized as well as when there is a sudden large scale change in the structure of the network. As the density of the network increases and the mobility decreases, the convergence time decreases.

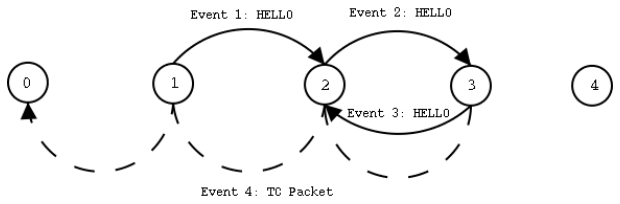


Fig. 2. OLSR Route Building Process

OLSR's Internet Draft [1] states its ability to be used with or without link layer feedback. However it is commonly studied without link layer feedback. In the absence of link layer feedback a node waits for a *NEIGHBOR_HOLD_TIME* for a *HELLO* packet. If this packet is not received from the neighbor during that time period then the link is considered down. Unlike the reactive protocols, OLSR does not notify the source immediately after detecting a broken link. The source becomes aware that the route is broken when the intermediate node broadcasts its next TC packet. The maximum delay of a source realizing a broken link is the sum of the delay in the intermediate node detecting the broken link and the delay of that node broadcasting a TC packet, or:

$$LINK\_BREAK\_DELAY = NEIGHBOR\_HOLD\_TIME + TC\_INTERVAL$$

With the parameters used in our analysis (Table II) the lower bound on the delay in the broadcast of information about a broken link is 6 seconds, assuming a TC packet is sent immediately after the neighbor times out. If a TC packet was sent out immediately before the neighbor timed out, the node would not broadcast the information about a broken link for an upper-bound of 11 seconds. If link layer notification is available then the delay in broadcasting the broken link is the amount of time until a TC packet is sent out, as the $NEIGHBOR\_HOLD\_TIME$ is negligible.

IV. SIMULATION ENVIRONMENT

The focus of this paper is comparing MANET routing protocols in environments with varying dynamics. We analyze the performance of the protocols with an emphasis on their ability to allow QoS models to provide application specific requirements. The empirical results presented in this paper were gathered using the NS-2 [11] network simulator. The simulation environment we study includes 50 nodes moving in a 1500 meter by 300 meter world. To simulate the movement of nodes the *random way-point model* [3] is used. Each node moves to a random location with a maximum speed of 10 meters per second, pauses for 10 seconds, and moves to a new random location. Twenty of the nodes were randomly selected as constant bit rate sources sending 256 byte packets. This model has become a standard when studying MANET routing protocols. The number of nodes and the dimensions of the world are chosen to ensure the impact of the mobility on multi-hop paths. We vary the rate at which each node sends packets and thus the load in the network. We quantify our results in the following metrics:

- Packet delivery ratio versus generated packets per second
- Average end to end hop count versus generated packets per second
- Average end to end latency versus generated packets per second
- Overhead (packets and bytes) versus generated packets per second

Packet delivery ratio is defined as the ratio of the total number of packets received by every destination to the total number of packets sent by each source. *Generated packets per second* is the total number of packets sent out by all sources per second. *Average end to end hop count* is defined as the ratio of the total hop counts of all packets successfully received by a destination to the total number of packets received. *Average end to end latency* is defined as the ratio of total time it takes all packets to reach the destinations to the total number of packets received. *Overhead* is defined as the total number of packets and total number of bytes generated by the routing protocol over the length of the simulation. Studying these specific metrics captures the diverse abilities of each routing protocol in handling the dynamics of the environment in a way that allows a stable platform for the QoS model to operate. We use the following notation to represent the protocols we study in this paper:

- OLSR without link layer feedback (OLSR-NL)
- OLSR with link layer feedback (OLSR-LL)
- AODV without link layer feedback (AODV-NL)
- AODV with link layer feedback (AODV-LL)
- DSR with link layer feedback (DSR-LL)

V. PROTOCOL EFFECTS ON QUALITY OF SERVICE

We study, analyze and compare three critical mechanisms of MANET routing protocols: (1) mechanisms used to build and maintain routes in the network, (2) mechanisms used to detect and advertise broken links and (3) buffering that occurs due to temporary route unavailability.

A. PATH SELECTION MECHANISMS

Both AODV and DSR have similar *Route Discovery* phases to discover new routes in the network. Routes that are found through the dissemination of *Route Request* packets represent the shortest hop count on the network. These routes are not removed from the routing table unless a *Route Error* is received or a *route timeout* expires. AODV and DSR restart their route timer each time a data packet is successfully sent using the path. AODV and DSR continue to use the original path when traffic is sent more frequently than the route timeout. The protocols lose the ability to take advantage of shorter routes that become available on the network. Paths that use needless hops lessen the ability to meet hop count requirements and degenerate overall network performance by wasting bandwidth. Wu et. all [16] study and give possible solutions to the problem of reactive protocols optimizing routes.

Through periodic transmission of *topology control* messages OLSR is able to recalculate its routing tables at constant time intervals. With OLSR, nodes have a more accurate view of the network and a more accurate view of the shortest path between a source and a destination. Figure 3(a) shows that OLSR without link layer feedback displays the lowest hop count of all the studied protocols. OLSR with link layer feedback does not perform as well as OLSR without link layer feedback. This is because the use of link layer feedback allows an intermediate node to immediately become aware of a broken link to select a redundant less optimal path. This leads to an increase in the packet delivery ratio of the protocol, seen in figure 3(b). The intermediate node that detects the broken link forwards the data traffic on a redundant path until the source realizes the path is broken through a *topology control* message. The redundant paths used by the intermediate nodes may not be the shortest paths on the network. The packets that are forwarded on the redundant suboptimal paths skew the average hop count to appear as poor as that of the reactive protocols.

When a node using OLSR with link layer feedback discovers a broken link, it removes the neighbor from its routing table and begins to send packets on alternate routes to the destination. The node does not send a reactive topology control message to notify the rest of the network that the link break has occurred. It instead relies on the periodic topology control message to notify the network of broken links. Figure 4 shows OLSR with link layer feedback and OLSR without link layer feedback have comparable overhead in the number of packets and bytes. AODV without link layer feedback sends more packets per second than OLSR without link layer feedback, but less bytes per second, however overall the protocols show a comparable overhead. Since AODV and OLSR are comparable in overhead and packet delivery ratio and OLSR without link layer feedback is superior in the quality of the paths that it builds, it is the preferable routing protocol in this scenario. OLSR builds paths with shorter hops than the reactive protocols, reducing the end to end transmission latency (Figure 6(a)).

The ability of OLSR to build paths with consistently lower hops allows a QoS framework to meet lower latency requirements. Additionally, a routing protocol that makes use of lower hop counts increases global network performance. When nodes in a wireless medium transmit data they traditionally broadcast the packet in a circular transmission range. During the transmission time the medium surrounding the sender cannot be used by any neighbors. Packets that are sent using paths with needless hops cause unnecessary collisions at the physical layer and degenerate overall network performance. For these reasons we feel the path selection mechanisms of OLSR support improved performance of a QoS framework.

B. LINK ERROR DETECTION

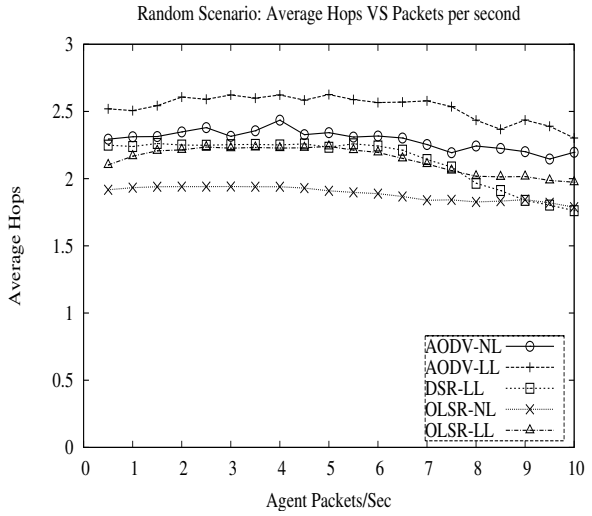
We define the *link error latency* as the time it takes for a node to discover that a link to a neighbor has been lost. Minimizing this latency is crucial when the neighbor is the next hop on a path that is actively being used. During the link error latency period, a node will not be aware that the link to a neighbor has been broken and continues to forward traffic on the broken link. Table 5 shows the link error latency of each protocol based on the mechanisms outlined in section III. When link layer feedback is available, nodes are notified in a negligible amount of time that the link to a neighbor has been lost. AODV without link layer feedback (Section III-B) has a link error latency of 4.5 seconds or the DELETE_PERIOD of a route. OLSR without link layer feedback has a link error latency of 6 seconds or the constant NEIGHBOR_HOLD_TIME. We did not consider DSR without link layer feedback in this study.

Protocol	**Link Layer Feedback**	**Link Error Latency**
OLSR	YES	0
OLSR	NO	6 s
AODV	YES	0
AODV	NO	4.5 s
DSR	YES	0
DSR	NO	N/A

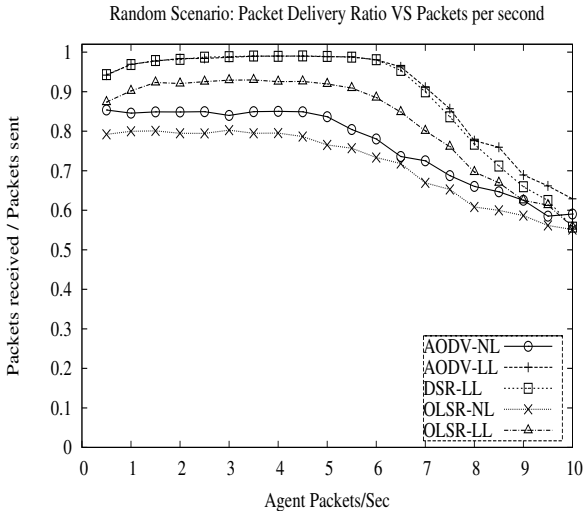
Fig. 5. Protocol Link Error Latencies

Figure 3(b) shows the packet delivery ratio of the five simulated protocols. OLSR without link layer feedback has a packet delivery ratio that is less than that of AODV without link layer feedback. Both protocols lack the ability to use link layer feedback, however OLSR without link layer feedback has a larger link error latency. OLSR with link layer feedback functions identically to OLSR without link layer feedback save for removal of the link error latency. Figure 3(b) shows the increase in OLSR's packet delivery ratio when it has the ability to process link layer feedback.

Figure 3(b) shows that if link layer feedback does not exist it is crucial that the mechanisms employed by the routing protocol are able to quickly detect the loss of a neighbor. This allows a protocol to quickly switch to an alternate path and deliver improved performance. The link error latencies of OLSR and AODV were calculated using the current Internet Draft's of each protocol. These numbers are constants that

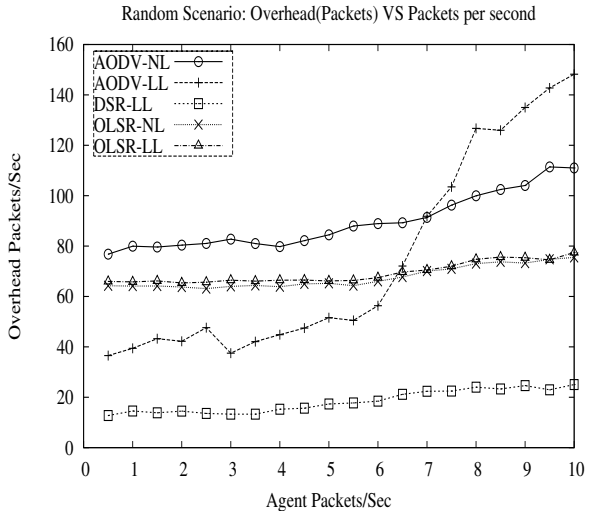


(a) Average hops vs packets per second

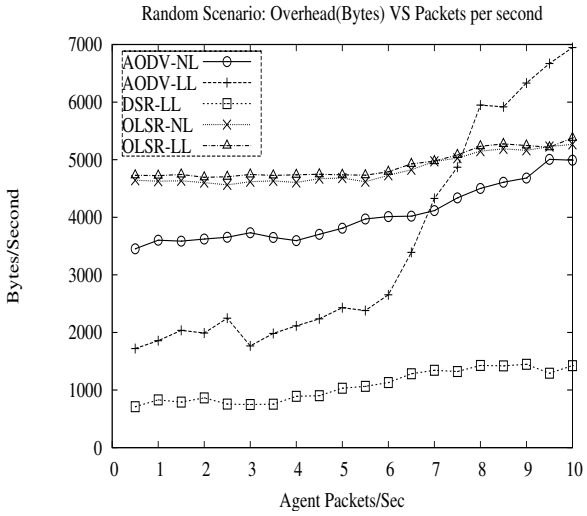


(b) Packet delivery ration vs packets per second

Fig. 3. Average hop count and packet delivery ratio



(a) Overhead (packets) vs packets per second



(b) Overhead (bytes) vs packets per second

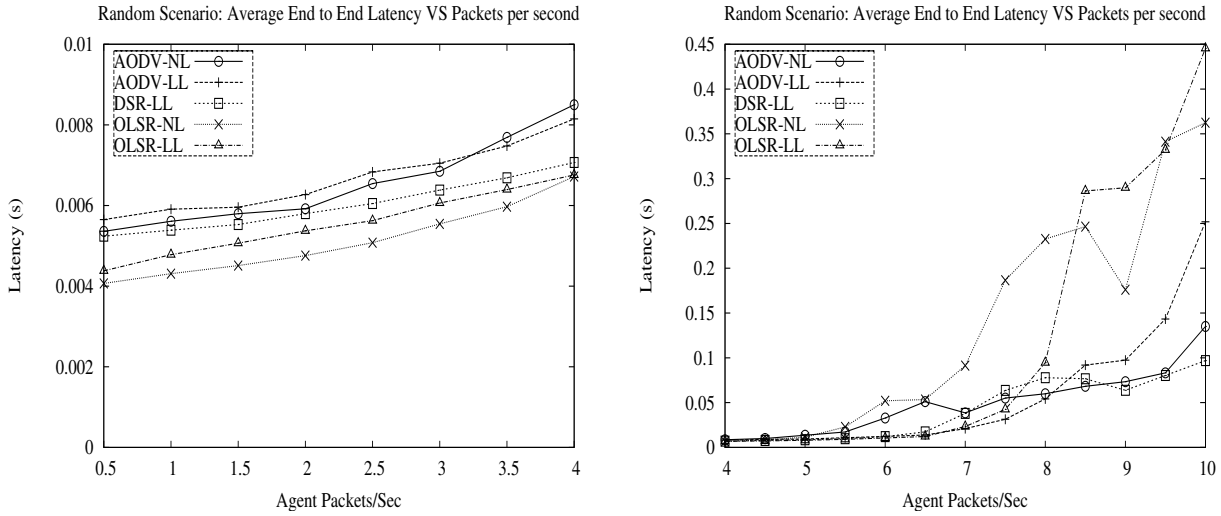
Fig. 4. Overhead, packets and bytes

are merely *suggested* by the authors of the protocols. The variability inherent in the link error latencies does not allow the declaration of a dominant protocol. However, the results imply the relationship between the link error latency and the ability of a QoS model to meet delivery requirements.

C. ROUTING LAYER BUFFERING MECHANISMS

During the route discovery phase, AODV and DSR temporarily buffer data packets while they search for the route to the destination. When the source receives a route reply packet it empties all the data packets within the buffer whose destinations correspond to the route just obtained. The effect on the end-to-end latency caused by buffering during the route discovery has been studied in the past [10] and is a common argument against reactive protocols.

In Section III-C we showed how the use of MPRs by OLSR can lead to delays in the construction of new routes. During these periods data packets intended for broken paths need to be buffered until the route has been reconstructed. In this study we add a routing layer buffer to OLSR that temporarily buffers data packets during periods when no route exists. The buffers employed in our implementations of AODV, DSR and OLSR have a timeout of 30 seconds and a maximum capacity of 64 packets. Without this buffer the packets will be dropped.



(a) Average Latency vs packets per second (1)

(b) Average Latency vs packets per second (2)

Fig. 6. Latency

Figure 6 shows the average end to end latency. Packets experiencing the top 5% delays have been removed from consideration to avoid skew effects. The graph is split in two separate portions to analyze two behaviors. Figure 6(a) shows the end to end latency with a traffic rate varying from 0.5 packets per second to 4 packets per second. The network experiences light load at these traffic rates. OLSR shows the lowest latency due to the lower hop count paths that are used. Figure 6(b) shows the end to end latency as the network becomes saturated. At this traffic load the reactive protocols deliver a more reliable latency. This is due to the reactive protocol's ability to search for new paths on-demand, while OLSR must wait for periodic control messages.

Protocol	**Max**	**Mean**	**95% Quantile**
AODV_NL	15.62	0.034	0.034
AODV_LL	5.82	0.021	0.035
OLSR_NL	13.52	0.047	0.013
OLSR_LL	12.11	0.019	0.013
DSR_LL	10.18	0.025	0.059

Fig. 7. Protocol latency results - 0.5 packets per second

Table 7 shows statistics about each protocol's performance with a traffic rate of 0.5 packets per second. The statistics were gathered without removing any latency times from the data set. OLSR with link layer feedback has a lower latency than the reactive protocols with link layer feedback. The protocols without link layer feedback show a higher maximum latency than all the simulated protocols with link layer feedback. OLSR has a 95% quantile value of 0.013, significantly less than that of the reactive protocols. A 95% quantile means that there is a 0.95 probability that a packet sent across the network has a latency less than or equal to 0.013. Even with this 95% quantile value, OLSR without link layer feedback displays the highest mean, implying a large number of packets are received with high latencies that skew the mean.

The impact on latencies that occurs during periods of route construction is not solely a problem with reactive protocols. The complexity of the forwarding rule of OLSR causes delays in the construction of routes which in turn causes packet delivery delays. Although reactive protocols suffer from high latencies during the route discovery phase, proactive protocols suffer from large latencies when substantial mobility exists on the network.

VI. CONCLUSION

In this paper we have evaluated three MANET routing protocols with an emphasis on how their behavior affects the ability to apply quality of service models. We have tested AODV with and without link layer feedback, OLSR with and without link layer feedback, and DSR with link layer feedback. This is the first paper to study OLSR with link layer feedback. We have analyzed the routing protocol mechanisms used by AODV and OLSR to detect broken links, and estimated the delay associated with them. We have defined and analyzed three areas of differentiation where the protocols affect unique quality of service guarantees. The proactive protocol uses consistently lower paths than the reactive protocols, a fact that leads to lower hop paths and lower end to end latencies. OLSR without link layer feedback has the highest *link error latency*, the delay of detecting a broken link, and thus has the lowest packet delivery ratio. This raises problems in the ability to make guarantees about delivery. The reactive protocols use buffering during *Route Discovery* and the proactive protocol buffers during periods of high mobility in the network. Buffering affects the distribution of latencies on the network, and can cause low priority packets that were generated some time ago to compete with higher priority packets being generated

at the present time. There is no perfect MANET protocol to apply quality of service, but understanding how each protocol affects quality of service is important to designing a reliable and robust QoS framework.

ACKNOWLEDGEMENT

This research is sponsored in part by a National Science Foundation (NSF) Instrumentation Award under grant CISE-9986105 and in part by a grant from Lockheed Martin Corporation and a grant from the Boeing Company.

REFERENCES

[1] Cedric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol. Internet-Draft Version 08, IETF, September 2003.

[2] J. Broch, David B. Johnson, and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet-Draft Version 08, IETF, February 2003.

[3] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[4] S. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile, October 1998.

[5] S. R. Das, C. E. Perkins, and E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications*, pages 3–12, March 2000.

[6] Santanu Das, Charles E. Perkins, and Elizabeth M. Royer. Ad hoc on demand distance vector (AODV) routing. Internet-Draft Version 13, IETF, February 2003.

[7] I. S. Department. Ieee 802.11 draft standard for wireless lan, mac and physical layer specifications. Technical report, IEEE, May 1997.

[8] A. Laouti, P. Mühlethaler, A. Najid, and E. Plakoo. Simulation results of the olsr routing protocol for wireless network. In *Med-Hoc-Net 2002*.

[9] J. Macker and M. Corson. Mobile ad hoc networking and the ietf, 1998.

[10] D. Maltz, J. Broch, J. Jetcheva, and D. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks, 1999.

[11] http://www.isi.edu/nsnam/ns.

[12] R. Ogier, M. Lewis, and F. Templin. Topology dissemination based on reverse-path forwarding. Internet-Draft Version 07, IETF, September 2003.

[13] A. Laouiti P. Jacquet. Analysis of mobile ad hoc protocols in random graph models. Research report rr-3835, INRIA.

[14] P.Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance analysis of olsr multipoint relay flooding in two ad hoc wireless network models. Technical Report Research Report 4260, INRIA, September 2001.

[15] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report Research Report RR-3898, INRIA, February 2000.

[16] Shu-Lin Wu, Tai-Ku Lin, Yu-Chee Tseng, and Jang-Ping Sheu. Route optimization on wireless mobile ad-hoc networks. In *In The 5th Mobile Computing Workshop*, pages 143–150, Taiwan, 1999.