SQ-AODV: A NOVEL ENERGY-AWARE STABILITY-BASED ROUTING PROTOCOL FOR ENHANCED QOS IN WIRELESS AD-HOC NETWORKS

Mallapur Veerayya
Centre for Artifitial Intellegence & Robotics
DRDO, Bangalore 560 093, India
Email: veerayya.mallapur@gmail.com

Vishal Sharma
Metanoia, Inc., 888 Villa Street, Suite 500
Mountain View, CA 94041-1259, USA
Email: v.sharma@ieee.org

Abhay Karandikar
Department of Electrical Engineering
IIT Bombay, Mumbai 400 076, India
Email: karandi@ee.iitb.ac.in

ABSTRACT

We propose a novel, energy-aware, routing protocol for Quality-of-Service (QoS) support in an infrastructure-less ad-hoc network. Our Stability-based, QoS-capable Ad-hoc On-demand Distance Vector (SQ-AODV) protocol is an enhancement of the well-known Ad-hoc On-demand Distance Vector (AODV) protocol. Our protocol utilizes a cross-layer approach, in which information about residual node energy is used for route selection and maintenance, and for quickly adapting to network conditions. The uniqueness of our scheme is that it uses only local information, requires no additional communication or co-operation between nodes, possesses a make-before-break capability that minimizes packet drops, and is compatible with the basic AODV data formats and operation, making it easy to adopt. We demonstrate, through extensive simulations in NS-2, that the increased route stability afforded by SQ-AODV leads to substantially better QoS performance. Our results show that under a variety of applicable network loads and network settings, our protocol achieves packet delivery ratio, on average, 10-15% higher than those of AODV and MDR (Min. Drain Rate) routing, and node expiration times 10-50% better than either AODV or MDR, with packet delay and control overhead comparable to that of AODV.

I. INTRODUCTION

With the growing use of networks (including ad-hoc networks) for real-time applications, such as voice, video, and real-time data, the need for QoS guarantees in terms of delay, bandwidth, and packet loss is increasingly important. This is particularly challenging for ad-hoc networks, where constraints on node energy, and mobility and the time-varying, shared wireless medium make QoS provisioning much more difficult. A key to enabling QoS guarantees in ad-hoc networks, therefore, is a dynamic routing protocol that can adapt quickly to network changes. The presentation of such a protocol is the goal of this paper.

A key to providing QoS guarantees in ad-hoc networks is to find a route to the desired destination, that can, with high probability, survive for the duration of the session. This ensures that communication once initiated will not be disturbed, and, as our subsequent results demonstrate, is a key criterion that impacts subsequent QoS (in this paper, by QoS we mean packet delivery ratio and packet delay) experienced by the flow. We propose a cross-layer approach to choose a stable route, which makes use of the current node energy and requires *virtually no overhead*. Our approach chooses a node as an intermediate router based on its current energy, its average energy drain-rate, and the specified session-duration (if known). We also propose a *"make-before-break"* mechanism for finding an alternate route for the session, when the energy drain rate of a node suggests that it will cease forwarding before the session is completed.

The remainder of the paper is organized as follows. In Section II we review some of the proposed stability-based and energy-aware routing protocols for ad-hoc wireless networks. We explain our proposed algorithm and its integration with AODV [1] in NS-2 [2] in Section III. In Section IV, we explain in detail our network model, and the specific scenarios simulated. In Section V, we compare the performance of our protocol with that of MDR, and with AODV via extensive simulations for a variety of network scenarios, and finally, we conclude in Section VI.

II. BACKGROUND AND RELATED WORK

Given the growing importance of QoS in wireless networks, over the last few years, a number of works have proposed ways to improve QoS in an ad-hoc

This research is supported in part by the TTSL-IIT Bombay Center for Excellence in Telecom.

wireless environment.

In [3], the authors proposed an extension to AODV to support QoS, assuming the availability of some stationary links in the network. The authors introduced the notion of *node stability*, based on a node's history, which incorporated both a node's mobility and its packet processing ratio. Only stable nodes were considered for routing. However, the authors did not consider the impact that unpredictable link failures would have on re-routing.

In [4] authors have proposed a stable, weight-based, on-demand routing protocol. The "weight" carried in the protocol messages used to select stable routes is based on three components: Route Expiration Time (RET), which is the predicted time of link breakage between two nodes due to mobility, Error Count (EC), which captures the number of link failures due to mobility, and Hop Count (HC). The authors have assumed that all nodes are synchronized via a Global Positioning System (GPS), so that two adjacent nodes may predict the RET. While the proposed scheme may combat against link breaks due to mobility, link breaks due to the draining node energy is a factor that also must be accounted for when computing weights for stable routing.

In [5], the authors have proposed a stable route selection scheme based on Link Expiration Time Threshold (LET_{th}). The Link Expiration Time (LET) is computed based on a prediction of neighbor mobility. LET computation needs to know the position of the neighbors, and hence requires periodic topology updates. However, the authors have not considered the impact that unpredictable link failures would have on re-routing.

In [6], the authors proposed a new metric, Energy-Drain-Rate, which is defined as the rate at which energy is consumed at a given node at time t. The corresponding cost function is defined as:

$$C_R = min\ T_r^i(t), \text{ where } T_r^i(t) = \frac{E_r^i(t)}{DR_r^i(t)},$$

where $DR_r^i(t)$ and $E_r^i(t)$ are the drain rate and the residual battery power respectively, of node i at time t along the path r. Thus the life-time of a path R is determined by the minimum $T_r^i(t)$ along that path. The Minimum Drain Rate (MDR) mechanism selects the route with maximum life-time. Each node monitors its energy consumption during a given past interval τ and maintains the drain rate value using an exponential weighted moving average. The proposed MDR algorithm attempts to select the best possible stable route for a given source and destination. The periodic route update used in MDR, however, soon becomes costly, as it increases control overhead and degrades performance at higher network loads.

From the proposals reviewed so far [3] – [6] it is clear that there is a need for a routing protocol that can provide stability to the routes selected for routing QoS-enabled applications, and also has mechanisms for fast re-routing to tackle unpredictable link breakages. Furthermore, for the scheme to be scalable, the stability should come at minimum or no overhead. In what follows, we propose modifications to the AODV protocol that, with high probability, provide routes that are stable for a session duration, and that also incorporate a fast *make-before-break* mechanism.

III. Stability-based QoS-capable Ad-hoc On-demand Distance Vector Routing

In this section, we discuss the features and operation of SQ-AODV an enhancement to the well-known AODV routing protocol. The two main features of SQ-AODV are that it:

- Provides stable routes by accounting for the residual life-time (calculated using the current Average-Energy-Drain-Rate (AEDR)) at intermediate nodes and the duration of the session (if known) at the route selection stage.
- Guards against link breakages that arise when the energy of a node(s) along a path is depleted, by performing a make-before-break re-route (where possible). This minimizes packet loss and session disruptions.

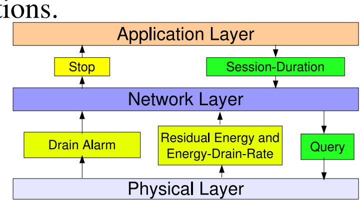

Fig. 1. Cross-Layer Architecture

The first feature ensures that SQ-AODV only routes sessions along routes that either have intermediate nodes with sufficient energy to last the length of the session or along routes that maximize the residual life-time of the bottleneck node, thus ensuring, with very high probability, that session disruption due to energy depletion at an intermediate node does not occur. It turns out that this increased stability leads to substantially better QoS in terms of packet delivery ratio (PDR) and packet delay (PD), even without explicitly accounting for bandwidth or delay requirements, as our subsequent results demonstrate. The second feature ensures that when a link break due to node energy depletion is imminent, SQ-AODV proactively re-routes sessions, without losing any packets. Once again, this provides near-zero packet loss and superior QoS performance.

The operation of SQ-AODV utilizes the cross-layer design depicted in Fig. 1, where energy information from the physical layer is used in admission control decisions at the network layer and to turn-off sessions at the application layer. We now explain these two features in more detail.

The first modification/feature is outlined in Algorithm 1, and helps in choosing an appropriate sequence of intermediate nodes for the requesting session.

The application layer of a source that wishes to communicate with a destination, generates data packets and transmits them to the network layer. At the network layer, the routing protocol responsible for finding a route to the desired destination initiates a route discovery procedure, if it does not already have a route for that destination. We assume here that, if the session-duration is known, the application layer directly provides that to the network layer, as shown in Fig. 1. If not, each intermediate node uses a heuristic and accepts a session only if it has at least **Threshold-1**[1] of residual life. The source broadcasts Route Request (RREQ) packets to its neighbors when it has no route to the desired destination.

Algorithm 1 : Selecting an intermediate **node** as **router**

An intermediate node N receives a **RREQ**;
if *Session-Duration* is specified in the **RREQ then**
    Check
    if *Current-Energy* $>$ (*Session-Duration* $\times$ *AEDR*) **then**
        Update **Bottleneck life-time** field of **RREQ**;
        ADMIT session & forward **RREQ** to the nbrs.
    else
        REJECT the session, and **DROP** the **RREQ**
    end if
else
    if *Current-Energy* $>$ **Threshold-1** **then**
        Update **Bottleneck life-time** field of **RREQ**;
        ADMIT session & forward **RREQ** to the nbrs.
    else
        REJECT the session, and **DROP** the **RREQ**
    end if
end if

When a RREQ packet reaches an intermediate node, Algorithm 1 queries the physical layer for the current residual energy, and checks whether the residual energy at the current AEDR is sufficient to last the duration of the flow. The session is only admitted if that is the case. If the session-duration is unknown, the algorithm admits the session only if the residual energy at the node is above **Threshold-1**. Before forwarding, the node updates the bottleneck life-time field of the RREQ packet.

The Energy-Drain-Rate (EDR) is computed as a difference between the energy En of the node at periodic intervals divided by the length of the interval. Thus,

$$EDR(t_2) = \frac{En(t_1)-En(t_2)}{t_2-t_1},$$

where $En(t_1)$ and $En(t_2)$ are energy levels of the node at times t_1 and t_2 respectively. This EDR is averaged using exponential averaging with $\alpha = 0.5$ to compute the AEDR as follows:

$$AEDR(t) = \alpha \times EDR(t) + (1\text{-}\alpha) \times AEDR(t\text{-}1).$$

Finally, when the RREQ packets reach the destination, it picks a route that maximizes the route life-time by selecting the one with maximum life-time of the bottleneck node.

Algorithm 2 : Route maintenance by **make-before-break** at node N

Periodically compute **EDR** & check *Current-Energy*;
if *Current-Energy* $<$ **Threshold-2** **then**
    Check
    if N == I **then**
        Send **RCR** to all sources using node N as router
    end if
    if N == D **then**
        Send a **Stop-Traffic** request to all sources that are communicating with this destination
    end if
end if

The second modification/feature helps the routing protocol to adapt quickly to imminent link breakage likely to occur when the energy of a node is fully drained. The algorithm for this is depicted in Algorithm 2. Since the physical layer keeps track of the AEDR, it sends an alarm to the network layer, shortly before it is about to drain completely i.e., when the current energy of the node is less than a **Threshold-2**[1]. The routing protocol adapts to this event, and its behavior depends on whether the node is an intermediate (**I**) or a destination (**D**) node.

If the node receiving the drain alarm from its physical layer is an **I** node, it sends a Route Change Request

[1]**Threshold-1** and **Threshold-2** are the residual energy of a node with which the node is alive for the next X and Y seconds respectively, in our implementation X = 5 and Y = 1

(RCR) packet to all source nodes using it as an intermediate hop towards their respective destinations. The source upon receiving the RCR packet, begins a new route discovery procedure for the session, and thus, with high probability, finds a new route before an actual link break occurs on the original route, leading to the *make-before-break* behavior. This reduces packet drops due to link breakage and the consequent delay incurred, and enables the routing protocol to quickly adapt to network changes, if an alternate path to the desired destination exists. If the node being drained is a **D** node, it sends a request to the source to stop all traffic transmission to itself. When the request reaches the source, the network layer sends a stop signal to the application, as shown in Fig. 1, preventing further transmission of data. This reduces the number of packet drops in the network and increases packet delivery ratio, and reduces resource usage by avoiding packet transmissions to unavailable destinations. If a source node itself is about to drain, it simply continues to transmit data until it cannot transmit anymore.

IV. Simulation Set-Up and Scenarios

We have conducted extensive simulations in NS-2.30 to compare the performance of SQ-AODV with that of MDR [6] and AODV [1], and have considered the following five parameters:

- **Packet Delivery Ratio (PDR)**: is the ratio of the number of packets successfully received by all destinations to the total number of packets injected into the network by all sources. The PDR is therefore a number between 0 and 1.
- **Normalized Control Overhead (COH):** is the ratio of number of routing packets transmitted (hop wise) by all the nodes to the total number of packets successfully received by all destinations in the network. The normalized control overhead is therefore a number greater than 0.
- **Average Packet Delay (PD):** is the sum of the times taken by the successful data packets to travel from their sources to destinations divided by the total number of successful packets. The average packet delay is measured in seconds.
- **Node Expiration Time (NET)**: is the time for which a node has been alive before it must halt transmission due to battery depletion. The node expiration time is plotted as number of nodes alive at a given time, for different points in time during the simulation.
- **Connection Expiration Time (CET)**: is the time for which a connection has been active before it must cease transmission due to the non-availability of a route between source and destination. This occurs when nodes along the path expire or become unreachable due to poor link conditions. The connection expiration time is expressed in seconds.

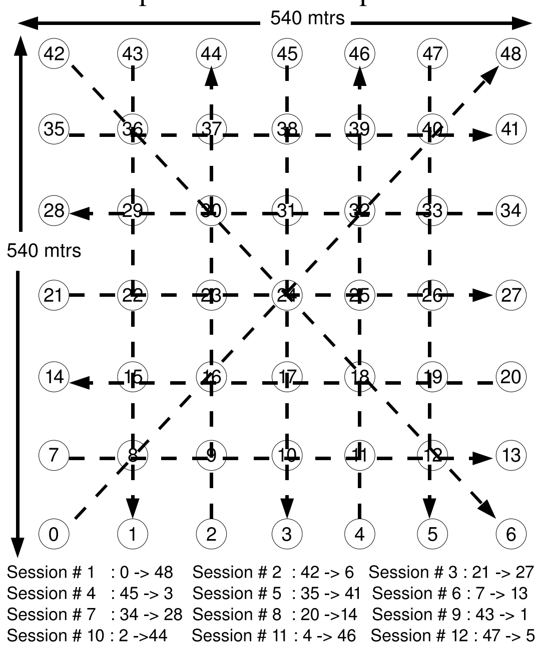

Fig. 2. Topology for Simulation with 12 Sessions

We consider the 49-node static topology (without node mobility) shown in Fig. 2 for our simulations. This is the same dense network scenario considered in [6]. The nodes are distributed uniformly in an area of size 540 m x 540 m, and are identical in their capability, but are initialized with different energies. As shown in Fig. 2, we consider 12 sessions in the network.

We have conducted two sets of simulations. The first set, **Set A**, is designed to evaluate the overall performance of SQ-AODV, MDR and AODV for CBR traffic sources, while the second set, **Set B**, is designed to evaluate the overall performance of these protocols for Poisson traffic sources and at *varying network loads*.

Note that our work is one of the first to examine the performance of ad-hoc routing protocols under *variable network load*. Practically all of the previous work that we are aware of, has focused on assessing protocol performance at only a given load, and varied other parameters. We note, however, that to understand whether the protocol gives consistent performance across a range of loads, it is also critical to assess how a routing protocol performs as the network load increases. This is because any realistic network will operate over a wide range of network loads, depending on its traffic profile, and it is imperative that the routing protocol give good performance at any load in this range.

Simulation Set A itself involves 2 experiments. In the first, all sessions begin transmission at the start of the simulation. In the second, the session start times are chosen randomly.

TABLE I

PARAMETER VALUES USED IN SIMULATION SET A AND B

Parameter	Value in Set A	Value in Set B
Packet size	512 Bytes	512 Bytes
Simulation time	800 seconds	Variable
Packets/Session	Variable	3000
Date traffic	CBR with 3 pkts/sec	Poisson with λ = 15 kbps - 65kbps
MAC Protocol	IEEE 802.11 DCF	IEEE 802.11 DCF
PCLP Data rate	1 Mbps	1 Mbps
Buffer length	50 Packets	50 Packets
Transmit power	0.2818 W	0.2818 W
Initial energy distribn.	25 J – 100 J	75 J – 300 J
Propagation model	Two-Ray Ground	Two-Ray Ground

Simulation Set B, on the other hand, is designed to test the overall performance of all the three protocols when the traffic arrives as per a Poisson process, for different network loads.

Every experiment was run 50 times (each initialized with a different seed), and the resulting parameters averaged over these 50 runs. The network parameters used in simulation Sets A and B are detailed in Table I.

V. SIMULATION RESULTS AND DISCUSSIONS

In this section, we present the results for subjecting our test network to CBR traffic, and to variable-rate Poisson traffic, respectively.

TABLE II

SIMULATION RESULTS OF SET A

Parameter	Set-A(1)			Set-A(2)		
	SQ-AODV	**MDR**	**AODV**	**SQ-AODV**	**MDR**	**AODV**
PDR	0.9760	0.8456	0.8681	0.9892	0.9201	0.8926
COH	0.7742	13.3207	1.0877	0.3402	4.1554	0.8256
PD (sec)	0.0618	0.2429	0.0543	0.0348	0.0508	0.0353

The results of the two experiments from simulation Set A are presented in Table II, while the plots of NET and CET are presented in Figs. 3 - 6. We see from Table II that the PDR for SQ-AODV in the two experiments is improved by 12.5% and 10.8%, respectively, relative to AODV. This is because choosing nodes with limited battery life, as happens in AODV, leads to (avoidable) disconnections of sessions. SQ-AODV, on the other hand, performs better because: (i) it chooses those intermediate nodes whose energy is sufficient to support the session for its entire duration or it chooses nodes to maximize the life-time of the route, and (ii) due to its make-before-break strategy, which re-routes a session proactively when link failure due to depletion of node energy is imminent. Thus, SQ-AODV successfully reduces packet drops in the network quite significantly.

Similarly, the PDR in MDR in the two cases is poorer by 15.4% and 7.5%, respectively as compared to SQ-AODV. This is because its periodic route update adds substantial routing overhead in the network. In fact, Table II shows that MDR overhead is approximately 17 times and 12 times worse than that of SQ-AODV, respectively, and almost 12 times and 5 times worse than that of AODV, respectively. This leads to its much lower PDR.

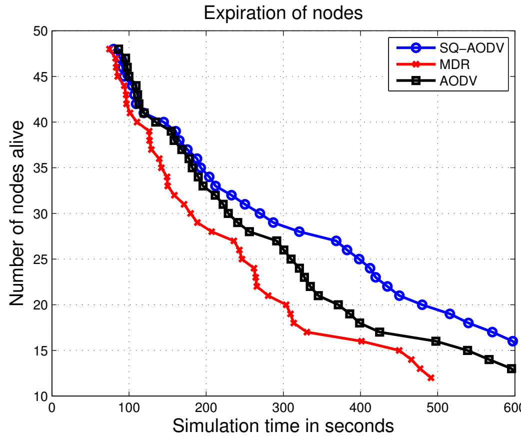

Fig. 3. NET: Sessions commence at start of simulation

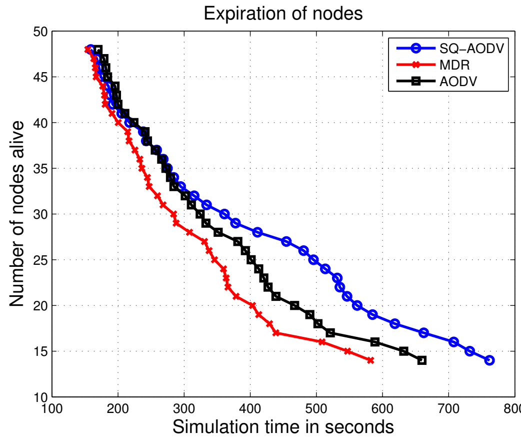

Fig. 4. NET: Random session start times

The packet delay for both SQ-AODV and AODV is comparable in both cases. We posit that this is because the delay in SQ-AODV is the result of two opposing factors. On the one hand, finding stable routes, where the life-time of the bottleneck node is maximized, may lead to longer (but more stable) routes, thus increasing delay. On the other hand, proactive route maintenance by way of make-before-break decreases delay, since no retransmissions need occur while an alternative route is located. These two factors have a compensatory effect, making the packet delay in SQ-AODV of the same order as that in AODV. MDR, by contrast, imposes a much higher load on the network due to its periodic route updates making the data packets wait longer, leading to a delay that is about 4 times and 1.5 times, respectively, the delay for AODV or SQ-AODV.

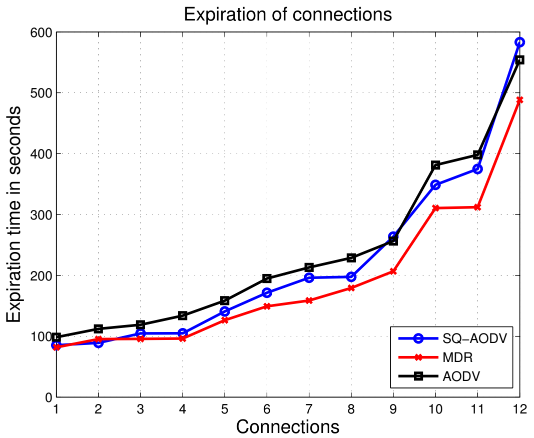

Fig. 5. CET: Sessions commence at start of simulation

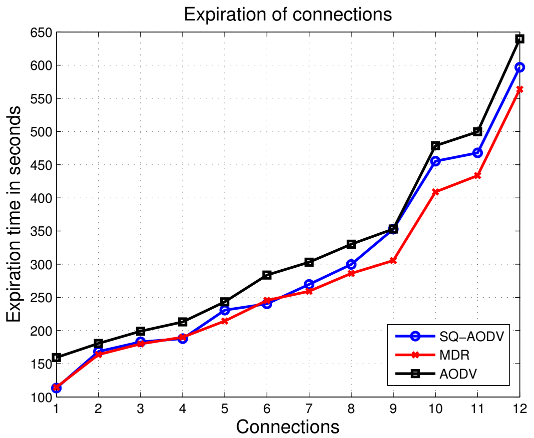

Fig. 6. CET: Random session start times

We see from Figs. 3 and 4 that in our network setting, running SQ-AODV improves the node expiration time by between 25 to 100 seconds over AODV, and by between 100 to 150 seconds over MDR. In other words, for a given number of nodes alive, this equates to SQ-AODV extending the node lifetime by between 10%-25% over AODV, and by between 25%-35% over MDR. Viewed another way, at a given simulation time, SQ-AODV typically has between 10%-25% more nodes alive than does AODV, and has between 20%-60% more nodes alive than does MDR. This is because SQ-AODV's proactive route maintenance is very economical of node energy. In addition, due to the proactive mechanism in SQ-AODV, a source stops transmitting traffic if a destination is about to drain, which saves resources by minimizing the transmission of packets that would not have been received by the destination in any case (due to its expiring). The nodes in MDR, on the other hand, expire faster than they do in either AODV or SQ-AODV by a significant margin, this is because the periodic route updates of MDR consume energy at a substantially faster rate, as our results demonstrate.

Figs. 5 and 6 illustrate that in our network setting, in terms of CET, AODV performs better by about 10-50 seconds over both SQ-AODV and MDR. (Note that the x-axis in these figures simply indicates the *number of connections* that have expired, and is *not* the connection identifier).

This equates to AODV connection expiration times being anywhere between 30%-7% better than those of SQ-AODV or MDR. This is because, in SQ-AODV: (i) a source on receiving an RCR from an intermediate node tries only a fixed (but configurable; in our case 3) times before it reaches the maximum number of retries and ends the session, and (ii) intermediate nodes reject a new session once its residual-energy is below Threshold-1. By contrast, AODV keeps retrying and so has a higher probability of finding a path, and keeping the session alive for longer. In the case of MDR, however, it is the COH packets that cause the node energy to drain faster, leading to the sessions expiring quicker than with AODV or SQ-AODV. We note that the slightly higher connection expiration times in AODV do come at the cost of lower PDR and lower node expiration times, which implies that even though the connections may be alive for a longer period in AODV, they do not successfully transmit as much data as SQ-AODV does.

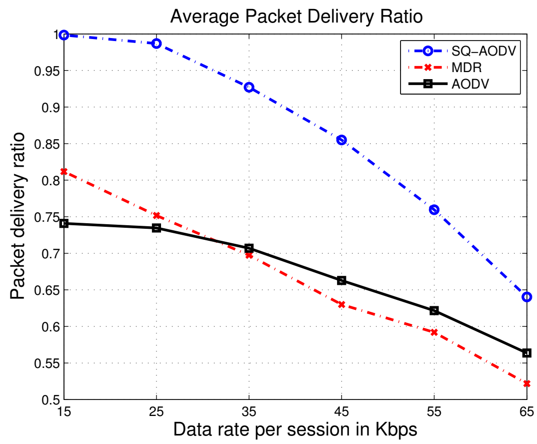

Fig. 7. Average Packet Delivery Ratio

The results for simulation Set B are presented in Figs. 7 - 9. We observe from Fig. 7, the PDR of SQ-AODV is substantially better than that for AODV or MDR. In fact, the PDR for SQ-AODV is improved by between 25%-13% over AODV and by between 22%-18% over MDR over the network loads considered. The key reason for this are the two properties of SQ-AODV discussed in Section III, which induce stable routes for the sessions and bolster PDR.

The PDR of MDR is better than that of AODV by 5%-10% at lower loads, but is reduced by equal amount at higher loads because of its extra overhead, which degrades MDR performance.

Fig. 8 shows that SQ-AODV has marginally higher normalized control overhead (between 1%-3% higher) than AODV. This is because, as explained in Section III, to support stable routing, SQ-AODV uses per-session (or per-flow) based routing (as opposed to simple destination-based routing used in AODV). For this, control packets of SQ-AODV carry extra flow-id information along with source and destination, and also packets need to travel all the way to destination to find a stable path, leading to marginally higher COH.

We see that MDR has the highest COH, almost 300% higher than either AODV or SQ-AODV at loads above 35 kbps, rising to over 1000% higher at lower loads. In particular, at lower loads the periodic COH of MDR becomes very high, because it takes substantial time for the sources to generate 3000 packets. In the meantime, the regular periodic updates of MDR continue accumulating significant COH.

Fig. 9 illustrates that, the delay experienced by packets in SQ-AODV is almost the same or marginally better than that in AODV, at all loads under consideration. The delay experienced in MDR is between 250-500 ms higher, or between 20%-50% higher than that in AODV and SQ-AODV because, at higher loads data packets have to wait longer due to periodic route updates.

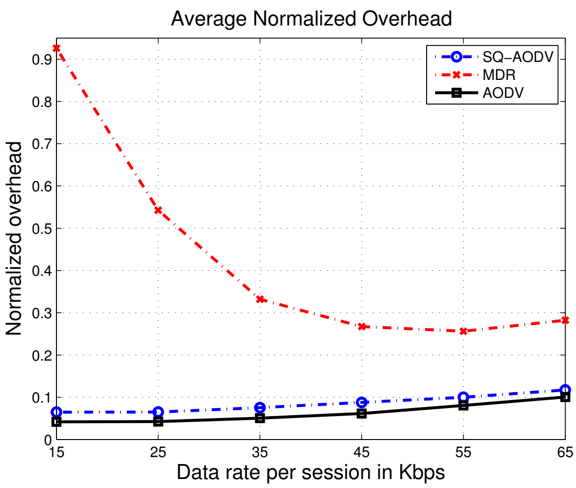

Fig. 8. Average Control Overhead

The advantage with SQ-AODV is that it is designed to provide stable routes and a fast re-routing capability to the nodes in ad-hoc networks at minimum overhead to the network. This helps in making effective use of network resources, as demonstated by our simulation results. A more detailed simulation scenarios and results can be found in [7]

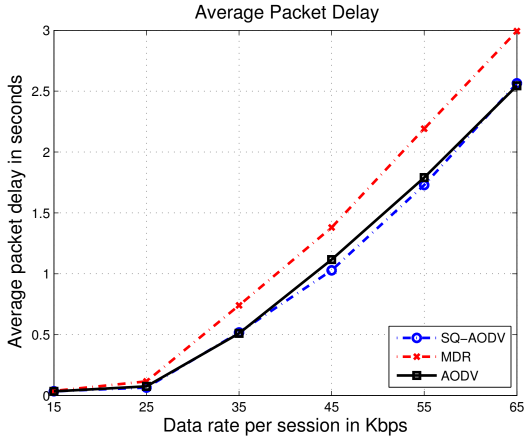

Fig. 9. Average Packet Delay

VI. Summary and Future Work

In this paper, we proposed a cross-layer based stable routing protocol for ad-hoc networks, which enhances support for QoS. The protocol uses only local information at a node without adding any significant overhead in the network. Indeed our scheme can be integrated with any on-demand protocol to improve the protocol's performance. Simulation results highlight the superiority of our protocol with respect to packet delivery ratio, packet delay, normalized control overhead, and node expiration time.

Several directions of future work are possible from here. The first is to combine this scheme explicitly with QoS routing, thereby incorporating bandwidth and delay constraints in the path selection process. Another is to consider the effects of mobility and fading in our stable routing protocol.

References

[1] C. Perkins and E. Royer, "Ad hoc on demand distance vector (AODV) routing," in *Proc. of IEEE WMCSA'99*, Feb. 1999, pp. 99–100.

[2] "Network Simulator, NS-2," http://www.isi.edu/nsnam/ns.

[3] J. Punde, N. Pissinou, and K. Makki, "On quality of service routing in ad-hoc networks," in *Proc. of LCN'03*, Oct. 2003, pp. 276–278.

[4] N. Wang and J. Chen, "A Stable On-Demand Routing Protocol for Mobile Ad Hoc Networks with Weight-Based Strategy," *IEEE PDCAT'06*, pp. 166–169, Dec. 2006.

[5] J. Lian., L. Li., and X. Zhu., "A Multi-Constraint QoS Routing Protocol with Route-Request Selection Based on Mobile Predictibg in MANET," *IEEE CISW'07*, pp. 342–345, Dec. 2007.

[6] D. Kim., J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, and P. Manzoni, "Routing mechanisms for mobile ad hoc networks based on the energy drain rate," *IEEE Transactions on Mobile Computing*, vol. 2, no. 2, pp. 161–173, 2003.

[7] V. Mallapur, "An Energy-Aware On-Demand Routing Protocol for Ad-Hoc Wireless Networks," Master's thesis, Department of Electrical Engineering, Indian Institute of Technology Bombay, India, Jul. 2008, http://arxiv.org/abs/0809.2322v1.