QoS Constrained Adaptive Routing Protocol For Mobile Adhoc Networks

Vinod Kone and Sukumar Nandi
Indian Institute of Technology Guwahati, Guwahati, India 781039
Email: {kvinod,sukumar}@iitg.ernet.in

Abstract— **The ease of deployment and the infrastructure less nature of Mobile Ad hoc Networks (MANETs) make them highly desirable for the present day multi media communications. Traditional routing protocols may not suffice for real time communications which require QoS support from the network. Though there has been considerable research in this area, most of the routing protocols take into account a single QoS constraint. In this paper, we propose an adaptive distributed routing protocol for MANETs which provides a QoS aware path from a given source to destination. The novelty of the protocol lies in taking multiple QoS constraints into consideration. Admission control and bandwidth reservation schemes are incorporated in the protocol to satisfy the QoS constraints of the applications. Relevant information is stored at the intermediate nodes to achieve the required QoS. The protocol has been simulated on top of IEEE 802.11 MAC, in ns-2 network simulator, to show it's correctness and efficiency.**

I. INTRODUCTION

MANETs are becoming the crucial medium of present day communication owing to their self configuring, easily deployable and infrastructure less nature. These networks are particularly suitable for emergency situations like warfare, floods and other disasters where infrastructure networks are impossible to operate. The rising popularity of MANETs for real-time multimedia applications like video and audio conferencing make QoS support for MANETs an unavoidable task. In addition, considering the bandwidth constraints and low memory space, providing QoS in MANETs is an interesting design objective.

Traditional Quality of Service(QoS) protocols developed for wired networks cannot be easily adapted to wireless scenarios due to the dynamic topology and error-prone nature of the wireless links. This is especially true for MANETs where every node moves arbitrarily causing the multi-hop network topology to change randomly and at unpredictable times.

In the literature, the research on QoS support in MANETs can be categorized into QoS models, QoS adaptation, QoS signaling, QoS routing and QoS MAC protocols. In this paper, we will be dealing with QoS routing in particular though signaling functions are also incorporated. The existing QoS models can be classified into Integrated services (IntServ [1]) and Differentiated services (DiffServ [2]) based on their operation. IntServ classifies the network traffic into different flows on the basis of source and destination pairs. RSVP [3] is the control protocol inbuilt in IntServ to setup and control QoS reservations. The major drawback of IntServ is the scalability problem due to the per-flow granularity provided by the service. DiffServ, on the other hand, partitions the traffic into different service classes with different QoS parameters. Though this tackles the scalability problem associated with integrated services, it has several drawbacks. One of the limitations of DiffServ is that traffic conditioning is done at boundary nodes [4] whereas in a MANET there are no boundary and interior nodes. Considering the problems of both the approaches, it is desirable to have a hybrid approach of integrated and differentiated service.

A survey of QoS aware routing protocols for MANETs show that most of them take into consideration one or at most two metrics. The problem becomes complex when the number of independent metrics to be satisfied by the network is increased. It has already been proved in the literature that the problem of finding a path with two independent constraints is a NP-complete problem [5]. Therefore a heuristic approach is needed for QoS routing protocols dealing with multiple independent constraints.

In this paper, an adaptive QoS routing protocol for MANET has been proposed. This includes a scheme for reserving bandwidth at intermediate nodes along the path for different flows. The QoS metrics taken into consideration are bandwidth, end-to-end delay and cost of the path. The protocol finds a bandwidth and delay constrained least cost path. A bandwidth estimation technique is used to find the available bandwidth of any node in the network. The protocol has been simulated in ns-2 network simulator and it's performance is compared with another QoS routing protocol.

The rest of the paper is organized as follows: Section II deals with the related work on the QoS routing in MANETs, the problem at hand is formulated in section III and our protocol, AQR (Adaptive Quality of Service Routing) is explained in section IV. Section V contains the simulations and results.

II. RELATED WORK

This section discusses some of the existing network layer solutions that support QoS provisioning out of many existing solutions like CEDAR [6], INSIGNIA[7], On-demand QoS routing protocol [8], [9], [11] and [12].

In [6] a core extraction algorithm establishes a core network dynamically and incrementally propagates the link state of high bandwidth links to core nodes. An on demand route computation is performed by the core nodes of the network using only local state. Any adhoc routing algorithm like [10] can be used in the core graph of CEDAR. Besides, CEDAR

9th International Conference on Information Technology (ICIT'06)
0-7695-2635-7/06 $20.00 © 2006 IEEE

has it's own route computation algorithm which consists of three phases: i) Discovering and establishing a core path to the destination ii) Searching for a stable QoS path and iii) Dynamic re-computation of routes in case of topology changes or route breaks. The major drawback of CEDAR is that it is not scalable for large scale networks. The routing algorithm only works well for a network consisting of ten to hundred nodes.

INSIGNIA [7] is an in-band signaling system that supports fast reservation, restoration and adaptation algorithms specifically designed to deliver adaptive service. An IP portion of the IP data packet, called INSIGNIA option, is used to carry the signaling control information. Per-flow granularity is supported by the INSIGNIA in order to provide end-to-end QoS guarantees. However, since the flow state information should be kept in the mobile hosts, the scalability problem may hinder it's deployment in large scale networks and also as resources are reserved only after the actual data transmission begins, it may be not be suitable for real time applications having stringent QoS requirements.

In [11], the nodes of the network are partitioned into centrally located nodes and non-centrally located nodes. When a new connection request comes, the network graph is preprocessed by a prioritized admission control algorithm on the centrally located nodes. Those central links not satisfying the bandwidth requirements are deleted by the admission control algorithm and the pre-processed graph is utilized as input for a conventional QoS routing algorithm. A similar scheme is also proposed to achieve load balancing and avoid bandwidth fragmentation.

In [12], the authors concentrate on solving the problem of efficient routing caused by nodes moving with a relatively high velocity. The solution, a cooperative three-tier framework, consists of three algorithms, Suitable Size Clustering Algorithm (SSCA), Dynamic Self-Adaptive Route Update Algorithm (DSRU) and Soft QoS Assurance Routing Algorithm(SQAR) which have their respective functions. The function of SSCA is topology management; DSRU is proposed in order to control the routing overhead; and SQAR is responsible for selecting and maintaining QoS constrained paths.

III. PROBLEM FORMULATION

A MANET can be represented as an undirected graph $G(V,E)$ where V is the set of nodes(vertices) and E is the set of links(edges) connecting the nodes. Note that the links can be broken at any time owing to the mobility of the nodes. Any link $e = (u,v) \in E$ has an associated cost $C(e)$, delay $D(e)$ and bandwidth $B(e)$. A path between two nodes u and v is given as $P(u,v) = (u, e(u,x), x, e(x,y), y, ..., e(z,v), v)$. It can be emphasized that a path between any two nodes is a set consisting of all possible paths between them. Formally, $P(u,v) = \{P_0, P_1, ..., P_n\}$ where each P_i is a candidate path between u and v. Cost of a path P_i is defined as

$Cost(P_i) = \sum_{e \in P_i} C(e)$.

Similarly, the end-to-end delay along the path P_i is

$Delay(P_i) = \sum_{e \in P_i} D(e)$

We further define, a bottle neck bandwidth for a path P_i is

$BotBand(P_i) = min\{B(e_0), B(e_1), ..., B(e_n)\}$ where $e_0, e_1, ..., e_n$ are the edges making up the path.

Now, we define the problem at hand as follows: **"To find a least cost path from source s to destination d satisfying both bandwidth constraint(Δb) and delay constraint (Δd)."** We call this path as QoS Aware Path (QAP). Supposing that $i \in \{0, 1, \ldots, n\}$,$P(s,d) = \{P_0, P_1, ...P_n\}$ for given source s and destination d, we can define the problem mathematically as follows:

$$\begin{aligned} QAP(s,d) &= P_i \qquad (1) \\ with\ Cost(P_i) &= min\{Cost(P_0), Cost(P_1)...Cost(P_n)\} \\ \forall i\ BotBand(P_i) &\geq \Delta b \\ \forall i\ Delay(P_i) &\leq \Delta d \end{aligned}$$

More generally, the objective of AQR is to find a path with the least cost among all the candidate paths from source to destination which have a bottleneck bandwidth of atleast the requested bandwidth and the total end to end delay experienced by a packet is atmost the requested delay.

IV. ADAPTIVE QOS ROUTING PROTOCOL

In Adaptive Quality of Service Routing (AQR), QoS is achieved by coupling on-demand routing with resource reservation and maintenance. But in reservation based approaches like RSVP the overhead of connection establishment and maintenance generally outweighs the initial cost of establishment. This has an adverse effect on MANETs because of the scarcity of resources and dynamic topology. Hence care has to be taken to keep the end-to-end signaling at minimum.

A. Design Issues

The challenges addressed in AQR are as follows:

1) To provide a QoS path as requested by the application.
2) Local route repair mechanism is included in the protocol which takes care of down links.
3) QoS recovery is made possible without any external signaling. The inherent nature of the protocol reports the source of any QoS violations.
4) Bandwidth reservation is done on the fly at the nodes, along the path, which provides admission control.

B. Neighborhood Maintenance

Since global state of the network is not maintained, it is very crucial to have neighborhood information at every node. Each node maintains a Neighbor List. The Neighbor List is periodically updated by HELLO packets. Every node broadcasts HELLO packets (contains Seq. No., Source ID, Bandwidth Consumed) to all it's neighbors A node receiving the HELLO packets from it's neighbors, stores the information (i.e. Next Hop ID, bandwidth Reserved and delay) in it's Neighbor List and drops the packet. If a node doesn't receive HELLO packets from one of it's neighbors for HELLO_INTERVAL period, it assumes that the neighbor has moved out of it's range and thus purges the corresponding entry in the Neighbor

List. In an unsynchronized network, the delay to the one hop neighbors is calculated as Round Trip Time (RTTi)/2. RTT can be easily calculated by time stamping the HELLO packets. The receivers of HELLO packets will unicast their replies back to the source node. By calculating the difference between the time stamp of the reply packet and the current local time a node can estimate the RTT. Similarly, the cost of a logical link between two nodes can be exchanged using HELLO packets.

C. Route Request Phase

AQR's route establishment is done on-demand using selective flooding. When a source needs to send data packets and the route to the destination is not present in it's route table, the source broadcasts a Route Request (RREQ) packet to all it's one hop neighbors. The RREQ packet contains Sequence number, Source ID, Destination ID, QoS parameters, Delay_till_now and Cost_till_now. The QoS parameters are Minimum Bandwidth B_{min} and Maximum Delay T_{max} requested by the application. Upon receiving a RREQ packet an admission control decision is made at the node as described in Section IV-E. If the packet passes the admission control decision, the receiving node adds an entry into it's routing table and broadcasts it to all it's neighbors. This process continues either till the destination is reached or the delay experienced by the packet exceeds the limit T_{max}. If the delay constraint is violated the RREQ packet is simply dropped. If the destination receives the RREQ packet, it unicasts a Route Reply (RREP) packet back to the source. The RREP contains Sequence number, Source ID and Destination ID.

Two new fields, Delay_till_now and Cost_till_now, are also included in the RREQ packet. The information about the delay that the RREQ packet has experienced so far and the cost of the path so far traveled are stored in these fields respectively. These fields are initialized to zero at the source node. Whenever an intermediate node receives a RREQ packet, it updates these fields based on the link information which is stored in it's Neighbor list.

D. Forwarding RREQ packets

Each node implementing AQR maintains a Metric Table (having Sequence number, Source ID, Destination ID, QoS parameters and wait with delay_till_now and Cost_till_now) which is the crucial data structure of the protocol. We note that a node can receive the same RREQ packets through different paths due to the broadcast. Instead of dropping the duplicate RREQ packets, the intermediate nodes cache the RREQs in the Metric Table. Whenever an intermediate node wants to forward a route request packet it consults the Metric table and selects the packet with the least Cost_till_now parameter. This packet will be forwarded to all the downstream neighbors. If a destination node receives a RREQ packet it simply sends a ROUTE REPLY (RREP) acknowledgment packet back to the source. This is facilitated by the routing tables which are updated during the ROUTE REQUEST/REPLY phase. If an intermediate node receives a RREP before timeout, it will forward the packet upstream towards the source and also purges the RREQ packets cached in the Metric Table. Otherwise, it will select the next packet in the Metric table with the lowest cost and forwards again toward the destination. Since an intermediate node always broadcasts the RREQ packet with the least Cost_till_now parameter it can be intuitively seen that the path selected will be the least cost path that obeys the delay and bandwidth constraints.

E. Admission Control

Admission control decision at each node ensures that the requested minimum bandwidth B_{min} and the maximum end-to-end delay T_{max} constraints are satisfied.

1) Bandwidth Estimation: The difficulty of calculating bandwidth of a wireless channel arises from the fact that it is shared by multiple nodes unlike in the wired scenario. A simple method of estimating the total available bandwidth of the wireless channel of a node is by calculating the total bandwidth consumed by all the nodes in the interference region and deducting it from the raw data rate of the node. This follows from the fact that a node can't use the channel if other interfering nodes are transmitting or receiving at the same time. The bandwidth estimation technique proposed in [13] is used for this purpose and it is briefly explained here for clarity.

The following symbols are defined before estimating the bandwidth of a wireless channel:

- B: Data rate of the node
- B_{avail}: Bandwidth available at the node
- B_{int}: Total interfering traffic at the node
- B_{resv}: Bandwidth reserved at the node by different flows
- B_{cons}: Bandwidth consumed by the requesting flow

To estimate the available bandwidth at any node, the total traffic in it's interference region needs to be calculated. There are three kinds of traffic that contribute to B_{int}.

1) Self Traffic $B_{self}(I)$, bandwidth consumed by the traffic transmitted or received by node I.
2) Neighborhood traffic $B_{neigh}(I)$, total traffic between $I's$ neighbors,$N(I)$.
3) Boundary traffic $B_{boundary}(I)$, total traffic between $I's$ neighbors and nodes that are outside $I's$ range, whose connection crosses the boundary of node $I's$ accessible range.

Therefore, the total interfering traffic at node I can be calculated as:

$$B_{int}(I) = B_{self}(I) + B_{neigh}(I) + B_{boundary}(I) \quad (2)$$

This can be better understood from the Fig. 1. Node I has four neighbors (A, B, C and D) and node E is outside of I's range. There is self-traffic between I and node A which equals T_{IA}, neighborhood traffic between nodes B and C which equals T_{BC} and boundary traffic between nodes D and E which is T_{DE}. Therefore the total interfering traffic (B_{int}) is the sum of all the three kinds of traffic. However, since in some cases neighborhood traffic and boundary traffic can transmit simultaneously the total interfering traffic is a

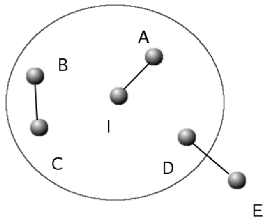

Fig. 1. Total Interfering Traffic

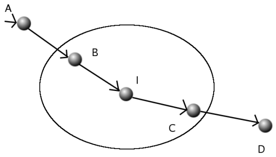

Fig. 2. Additional Traffic By Requesting Flow

conservative estimate. It has been shown in [13] that, $B_{int}(I)$ can be estimated as follows

$B_{int}(I) = \sum_{J \in N(I)} B_{self}(J)$

The available bandwidth of the node I can be estimated as

$B_{avail}(I) = B - B_{int}(I)$

The bandwidth consumed at a node is different from the requested minimum bandwidth(B_{min}) of the flow. This is due to the extra traffic that is brought by the flow j, in case it is accepted. To understand the situation, look at the Fig. 2. Let $B_I(j)$ be the bandwidth to be reserved at node I for flow j.

$$B_I(j) = \begin{cases} B_{min} & \text{if A or D} \\ 2B_{min} & else \end{cases} \qquad (3)$$

Since the intermediate nodes need to receive and forward the flow j, the reservation required at the node is double the requested bandwidth.

However, $B_I(j)$ doesn't include the additional boundary traffic, brought by the flow, to node I ; e.g., T_{AB} and T_{CD} as shown in the Fig. 2. Due to traffic aggregation, both the new self-traffic and new boundary traffic introduced by the requesting flow is included in computing the total bandwidth consumed by the flow j at node I, $B_{cons}(I, j)$.

$B_{cons}(I, j) = T_{AB} + T_{BI} + T_{IC} + T_{CD} = B_B(j) + B_C(j)$.

In general, we call B and C as uplink and downlink nodes respectively. Therefore

$B_{cons}(I, j) = B_{uplink(I)}(j) + B_{downlink(I)}(j)$

Now, the total bandwidth reserved at node I for all flows can be calculated as:

$$B_{cons} = \sum_{\forall j} B_{cons}(I, j) \qquad (4)$$

2) Bandwidth Reservation: Every node maintains a Flow Table which stores information about the flows that are accepted to travel through this node (i.e. Source ID, Destination ID, bandwidth Reserved and T_{max}). In particular, Flow Table contains bandwidth reserved for various flows at the current node. As discussed above, every node calculates the total reserved bandwidth by simply adding up the bandwidth of all flows present in it's Flow Table. This information is disseminated to all the neighbors through HELLO packets. A node receiving HELLO packets stores the total reserved bandwidth of it's neighbors in it's Neighbor List. It simply calculates the available bandwidth by the formulas given earlier.

Whenever, a new flow j requests for a certain minimum bandwidth B_{min}, the node I compares $B_{cons}(I, j)$ and $B_{avail}(I)$. If the flow can be accepted ($B_{cons}(I, j) \leq B_{avail}(I)$), an entry is made into the flow table otherwise the RREQ packet is dropped.

F. Adaptive Route Recovery

Local route repair and QoS violation recovery are crucial in a MANET environment because of the mobility of the nodes. As the nodes keep moving away from their neighbors the chances of route breaks become high and hence the need to re-route the packets. The effectiveness of any routing protocol lies in detecting these violations and re-routing the packets seamlessly.

1) Local Route Repair: AQR incorporates a simple and effective local route repair mechanism to take care of route break. As explained earlier, every node maintains a Neighbor List where neighborhood information is stored and updated periodically with the help of HELLO packets. When a node stops receiving HELLO packets from one of it's neighbors it assumes that the route to the node is down and sets a Route Down flag in it's route table. The node sends a Route Error (RERR) packet (having Type of error, Destination ID and Destination Sequence number) back to the source, which then reroute the data packets along a new path.

2) QoS violation Detection and Recovery: QoS violations can be detected by AQR either during the control phase or during the data transmission phase. In the control phase, Delay_till_now parameter of the RREQ packets and the admission control procedure described earlier are used to detect the QoS violations. During the route discovery, if an intermediate node finds that Delay_till_now parameter of the RREQ packet has exceeded the T_{max} (stored as a field of RREQ) requested by the application it discards the RREQ packet. Similarly, if bandwidth and cost constraints are violated by the RREQ packets they are simply dropped. During the data transmission phase, the destination stores the T_{max} information in it's Flow Table and hence can detect QoS violation if the time delay of the data packets exceeds T_{max}. To recover from this, destination sends a new QoS Error (RQERR) packet (having type of error, Source ID and Destination ID) back to the source, which then reroutes the data packets along a new path.

V. SIMULATIONS AND RESULTS

AQR routing protocol is implemented on top of the IEEE 802.11 MAC implementation of ns-2. An omni directional antenna with Two-Ray ground propagation model is used for transmission and receiving. The Table I gives the information that is common for all of our simulations. Random way point mobility is used in our simulations. The other attributes of

TABLE I

NETWORK PARAMETERS IN NS-2

Channel	Wireless Channel
Propagation Model	Two Ray Ground
Network Interface	Wireless Phy
MAC Layer	IEEE 802.11
Link Layer	LL
Antenna	Omni Antenna

our simulations viz., number nodes, mobility, topology are changed from scenario to scenario.

The following metrics have been used to compare and analyze the performance of AQR.

1) *Jitter*: This is the time difference between the successive data packets received at the destination node.
2) *Normalized Control Overhead*: This is the ratio of total number of control packets sent into the network and the total number of data packets delivered to the destinations. This metric gives a measure of overhead induced in the network by the protocol.
3) *Traffic Admission Ratio*: This is the ratio of the packets actually sent by the sources and the packets generated by the sources. This measures the admission control policy of the protocol.
4) *End-to-End Delay*: This gives the time delay that a data packet has encountered from the time it was sent by a source to the time it was received at the destination.
5) *Packet Delivery Ratio*: This is the ratio of the number of data packets sent into the network and the number of packets successfully received at the destination. This measures the reliable data delivery nature of the protocol.

A. Scenario 1

In this scenario, jitter performance of AQR with AODV is compared. A small network of 3 mobile nodes is used for this case. The nodes move inside a field of $670m \times 670m$ with a flat grid topology. All the flows request a minimum bandwidth of 400 Kb/s and a time delay of 0.1s. Node 0 is a CBR source sending packets of size 512 bytes to node 2, whereas node 1 is a TCP source sending ftp packets to destination node 2. The simulation time for this scenario is set to 400s.

From simulation result we observed that AQR provides a better jitter performance compared to AODV though there hasn't been any jitter constraint in the QoS parameters. Since AQR sets up a QoS aware path with bandwidth control, the packets tend to travel in a path with less congestion and hence the jitter is very low. On the other hand, AODV doesn't provide any admission control mechanism and hence jitter reaches as high as 0.05 secs.

B. Scenario 2

AQR has been simulated on a large network of 50 nodes to test it's performance. The scenario in which the nodes move is a $1000m \times 500m$ flat-grid. The protocol has been simulated for various maximum speeds of nodes ranging from 1 m/s to 10 m/s and a pause time of 0 s. The simulation time is set to 500 s.

TABLE II

SIMULATION PARAMETERS

Number of Nodes	50
Minimum Bandwidth	40 Kb/s
Maximum Delay	0.1 s
Topology	1000m x 500m
Pause Time	0 s
Traffic	CBR
Packet Size	512 bytes
Rate	10 packets/s
Simulation Time	500 s

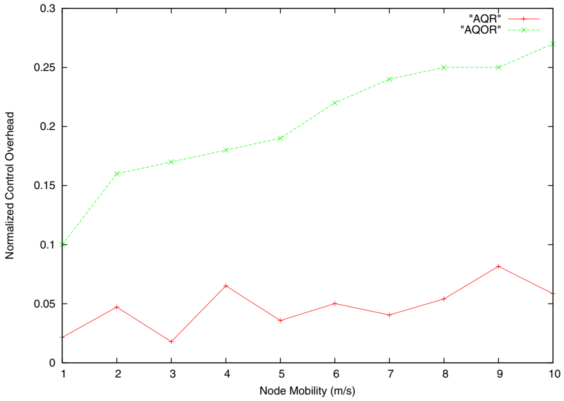

Fig. 3. Graph of Normalized Control Overhead vs Mobility

The connection pattern consists of 10 flows between source and destination pairs randomly chosen from the whole network. The traffic type is CBR and each source sends packets of 512 bytes at a rate of 10 packets per second. All the flows have the same QoS requirement of minimum bandwidth (B_{min}) 40 Kb/s and a maximum delay (T_{max}) of 0.1s. The Table II gives the information about the scenario. AQR's performance is compared with another QoS aware routing protocol, AQOR[13].

From the Fig. 3 we notice that the control overhead of AQR is very low compared to AQOR. Whereas AQOR's overhead was 0.3, AQR's was below 0.1, a reduction of 70%. We also note that our protocol adapts well to the mobility unlike AQOR which has a steep increase in the control overhead. The graceful increase in control overhead can be explained by the nature of AQR, which sends a new RREQ packet cached at the intermediate nodes in case the previous one has violated QoS constraints. We can reason that, with increase in mobility there will be more route breaks and hence intermediate nodes are not likely to receive RREPs from the receiver. But the intermediate nodes time out for RREPs and send the next best least cost packet cached in the Metric Table and hence the protocol adapts to the changes in mobility.

Fig. 4 plots the traffic admission ratio characteristics of AQR with respect to changes in the mobility. The admitted traffic percentage is around 95% for the various mobility scenarios. This shows that a QoS path is returned to the

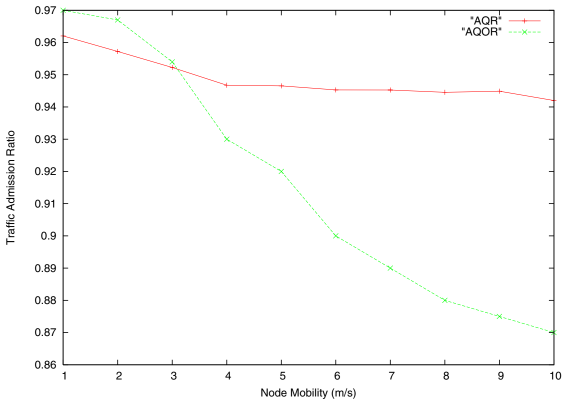

Fig. 4. Graph of Traffic Admission Ratio vs Mobility

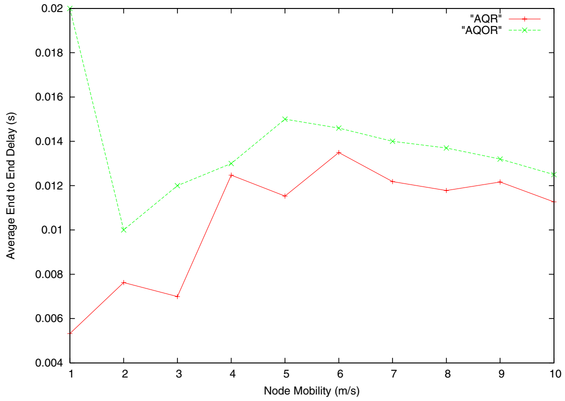

Fig. 5. Graph of Avg. End to End Delay vs Mobility

sources successfully as long as there is a possibility for it. Since the RREQ packets are broadcasted at the intermediate nodes repeatedly till a RREP is received, multiple possible paths will be checked for satisfying the QoS constraints. It can also be observed that, the degradation of the admission ratio of AQOR is very steep whereas AQR adapts well to the mobility.

Figs. 5 and 6 show the end to end delay and the packet delivery ratio characteristics of AQR with respect to the mobility of the nodes in the network . From Fig. 5 we notice that the average end to end delay is well within the bound of the required 0.1 secs. As the mobility of the nodes increase the congestion in the network increases (as can be seen from Fig. 3) which in turn results in an increase in packet collisions. Due to congestion, the QoS path may lead through a path of more number of nodes to satisfy the bandwidth constraint. Thus the end to end delay increases with increase in mobility.

From Fig. 6 we can notice that, AQR promises a high packet delivery ratio to the admitted flows. The percentage of packets successfully delivered to the destinations ranges from 96% to 98% similar to AQOR. This shows that the protocol adapts efficiently to the highly dynamic environment of MANETs.

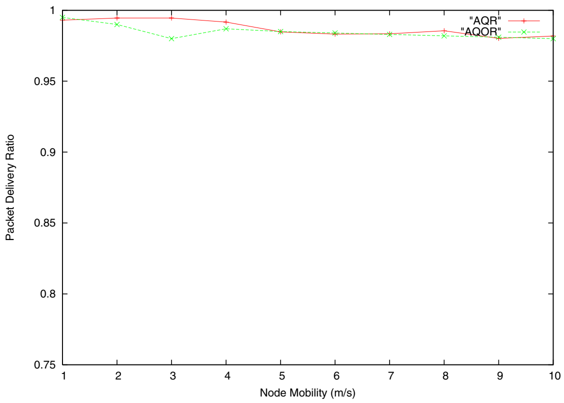

Fig. 6. Graph of Packet Delivery Ratio vs Mobility

VI. CONCLUSION

In this paper we have proposed AQR, an on-demand routing adaptive protocol with takes into account the QoS parameters of bandwidth, delay and cost. Specifically, AQR finds a bandwidth and delay constrained least cost path from source to destination. Bandwidth estimation and reservation is also included in the protocol, thus enabling admission control. AQR has QoS violation detection and recovery mechanisms which enables quick re-routing of the packets along new paths. The protocol has been implemented in ns-2 network simulator and it's performance is analyzed for small, as well as large scale networks. Our results show that AQR indeed achieves the QoS performance claimed and also adapts well to different topologies and mobility conditions.

REFERENCES

[1] D. R.Braden and S.Shenker, "Integrated services in the internet architecture an overview," IETF RFC1633, June 1994.
[2] S.Blake, "An architecture for differentiated services," IETF RFC2475, December 1998.
[3] L. R.Braden and S.Jamin, "Resource reservation protocol (rsvp)-version 1 functional specification," IETF RFC2205, September 1997.
[4] K. Wu and J. Harms, "Qos support in mobile adhoc networks," Crossing boundaries-An Interdisciplinary journal, 2001.
[5] Z. Wang and J. Crowcroft, "Bandwidth-delay based routing algorithms," in *IEEE GLOBECOM*, 1995.
[6] R. P.Sinha and V.Bharghavan, "Cedar: Core extraction distributed ad hoc routing," in *IEEE INFOCOM'99*, 1999.
[7] S. L. G.S. Ahn, A.T. Campbell and X. Zhang, "Insignia: An ip-based quality of service framework for mobile ad hoc networks," in *Journal of Parallel and Distributed Computing*, 2000.
[8] R.Lin and J.S.Liu, "Qos routing in ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, August 1999.
[9] S. Chen and K. Nahrstedt, "Distributed qos routing in ad hoc networks," *IEEE journal on selected areas in communications*, vol. 17, no. 8, August 1999.
[10] C.E.Perkins and E.M.Royer, "Ad hoc on-demand distance vector routing," in *IEEE WMCSA'99*, New Orleans, 1999, pp. 90–100.
[11] M. K. A. Bo Rong and K. Elhakeem, "Achieve load balancing and avoid bandwidth fragmentation in manet qos routing," in *IEEE CCECE/CCGEI*, 2005.
[12] Y. Z. Xin Jin, Hongbo Wang and B. Meng, "A three-tier framework supporting soft qos routing in manet," in *WONS 2004 LNCS*, 2004.
[13] Q. Xue and A. Ganz, "Ad hoc qos on-demand routing (aqor) in mobile ad hoc networks," in *Elsevier, Journal of Parallel and Distributed Computing*, 2003.