An AODV Based QoS Routing Protocol for Delay Sensitive Applications in Mobile Ad Hoc Networks

Rakesh Kumar, Anil K. Sarje, Manoj Misra
Department of Electronics and Computer Engineering
Indian Institute of Technology Roorkee -247667 India
{rkmlcdec, sarjefec, manojfec}@iitr.ernet.in

ABSTRACT: In order to provide quality delivery to delay sensitive applications such as voice and video, it is extremely important that mobile Ad Hoc networks provide quality of service (QoS) support in terms of bandwidth and delay. Most existing routing protocols for mobile Ad Hoc networks (MANETs) are designed to search for the shortest path with minimum hop counts. However, the shortest routes do not always provide the best performance, especially when there are congested nodes along these routes. In this paper, we propose an on demand delay based quality of service routing protocol (AODV-D) to ensure that delay does not exceed a maximum value. First in this paper, we compare the performance of two prominent on demand routing protocols AODV and DSR in stressful situations. As the performance of AODV has been found better in our simulation than DSR, we decided to modify AODV routing protocol to provide quality-of-service. This protocol takes into consideration MAC layer channel contention information and number of packets in the interface queue in addition to minimum hops. The protocol is implemented and simulated using GlomoSim simulator. Performance comparisons of the proposed AODV-D protocol against AODV, and QoS-AODV is presented and shown that the proposed protocol outperforms existing ones.

Categories and Subject Descriptors
C.2.2: [Protocol architecture]; C.2.4: [Distributed Systems Distributed applications]

General Terms: Mobile networks, Adhoc networks, Routing protocols

Keywords: Mobile Ad Hoc Networks (MANETs), AODV, QoS, MAC, Performance Evaluation, GloMoSim

Received:

1. Introduction

In order to run delay sensitive applications such as voice and video, it is extremely important that mobile Ad Hoc Networks [6] provide Quality of Service (QoS) support in terms of bandwidth and delay. Most existing routing protocols for mobile Ad Hoc Networks are designed to search for the shortest path with minimum hop counts. However, the shortest route does not always provide the best performance, especially when there are congested nodes along the route. Delay aware routing protocols make path selection between source and destination based on the delay over the discovered links during route discovery. Conventional routing protocols such as AODV, DSR and OLSR [9, 10] are used to identify a path between the source and destination nodes. But these protocols do not take into consideration the node capabilities, queuing and contention delays of intermediate nodes, which play an important role in achieving QoS in modern day Ad Hoc networks. Moreover, such routing protocols use minimum hop count as the main metric for path selection. In this paper, we propose an on demand delay-oriented shortest path Quality of Service (QoS) routing protocol (AODV-D) to ensure that delay does not exceed a maximum value for each session between a pair of source and destination mobile nodes. Quality-of-Service (QoS) is a desirable feature for MANETs due to the growth of multimedia applications [23,30] and real time traffic. There are basic differences between the requirements for the transmission of bursty traffic and real time traffic such as voice and video. Bursty traffic is error sensitive, while real-time traffic is delay-sensitive. Thus real time sessions require bandwidth and delay guarantees. Traditional routing algorithms for Ad Hoc network can not meet this requirement. Even though some approaches have been proposed to provide assurance in wired networks [21,24], wire-based QoS models are not appropriate for Ad Hoc networks due to MANETs inherent characteristics. In this paper, we propose an efficient AODV based QoS routing protocol for providing end-to-end delay guarantee in mobile Ad Hoc networks with IEEE 802.11 [17] as the MAC protocol. The solution consists of tracing routes in a reactive way by taking into account the QoS requirements in terms of delay associated with each flow. The performance of the proposed algorithm is evaluated by simulation for different mobility and traffic patterns.

The remainder of the paper is organized as follows. Section 2 describes motivation behind development of the protocol. Related works are briefly presented in section 3. Brief description of AODV and DSR protocols, simulation environment to compare them under the scenarios we used, and results of simulation have been presented under section 4 and section 5 respectively. Further, in section 6 we present the details of our proposed AODV based delay sensitive QoS routing protocol (AODV-D) which include delay estimation, route discovery and route maintenance. In Section 7, the simulation environment, simulation results and analysis under various conditions of the proposed protocol are presented. Finally, section 8 concludes the paper and outlines future work.

2. Motivation

A key issue in MANET is that the routing protocols respond rapidly to topological changes in the network. Most of the Mobile Ad Hoc networks use IEEE 802.11 [17] as the underlying MAC protocol. Current standard considers the shortest path with minimum hop count for the route selection. Although this hop metric is easy to implement and is reliable in dynamic environments, the queueing delay and the contention delay at the intermediate nodes are not taken into account for route selection. Thus, a minimum hop path may sometimes incur a higher end-to-end delay than some alternate paths. Moreover, routing protocols based on minimum number of hops can not fairly distribute the routing load among mobile hosts [4,13].
In IEEE 802.11 DCF, each node contends with its neighbor nodes and also the neighbors of its neighbors in the medium contention procedure. Consider Figure 1 to understand the effect of contention delay and queuing delay [16]. In Figure 1 (a), node X has only one neighbor whereas in Figure 1 (b), node Y has two neighbors. If $N_i$ is the number of packets at the mobile node $p$ for transmission then, even though queuing delay of node X and node Y is identical, say four units each, node Y experiences higher contention delay than node X as it has more number of neighbor nodes.

Therefore packets passing through node Y experience higher node delay compared to node X as node Y has more number of active neighbors (2 numbers) as compared to node X (1 number). But if we do not consider the contention delay, then packets passing through node X and Y will experience identical node delay. Thus, MAC delay becomes much larger if the routing algorithm keeps routing other packets to pass through a heavily loaded node. Since the range of possible medium contention of a mobile node is wide, medium contention times can affect the end-to-end delay considerably. MAC layer contention information provides estimation of neighbor nodes activities whereas queue length gives a measure of traffic load at the mobile node itself. An unbalanced distribution of traffic may lead to higher packet dropping rate and faster battery power depletion on certain mobile nodes. If route selection criterion becomes least path delay with minimum hop count instead of simple minimum hop count, then it may find a route with less traffic load. If it can find alternate route(s) before congestion, then it can maintain the required QoS constraints throughout the session.

3. Related Works

Perkins et al [13] have extended the basic Ad Hoc on Demand Distance Vector (AODV) routing protocol [3] to provide quality of service (QoS) support in Ad Hoc mobile wireless networks. To provide QoS (bandwidth and delay guarantee), packet formats (route discovery) and routing table structure were modified in order to specify the service requirements which must be met by the nodes forwarding a RREQ or a RREP packet. Since NODE_TRAVERSAL_TIME at a mobile node is only the processing time for the packet, the major part of the delay at a node is contributed by packet queuing and contention delay at the MAC layer. Hence a packet may experience much more delay than this when the traffic load is high in the network. This gave a motivation to develop a delay sensitive protocol that not only accounts packet processing time at a node but also MAC contention and queuing delay. The end-to-end delay of a path is the summation of the node delay at each node plus the link delay at each link $(i,j)$ along the path. Node delay includes the protocol processing time, the queuing delay at node $i$ for link $(i,j)$ and MAC contention delay at node i. Link delay is the propagation delay on link $(i,j)$. In wireless network, the propagation delay is very small and almost equal for each hop on the path. So, queuing delay and MAC delay are considered as the two main factors that accumulated the node’s delay.

Sheu and Chen [11] proposed a table driven delay-oriented shortest path routing protocol (DOSPR). They analyzed the relationship between the MAC delay and the neighbor number in mobile ad hoc networks, and also provided an estimation method for MAC delay. The average transfer delay of a packet between a given source and destination includes MAC contention delay, buffer queuing delay and transmission delay. This protocol outperforms min-hop count routing. The total transfer delay from source to destination for each packet has been reduced significantly. But as it uses table driven approach to maintain access delay at each node, routing overhead will be higher and also pose scalability problem.

In [12,19], Sun and Hughes proposed an adaptive distributed QoS routing scheme based on the prediction of the local performance. They analyzed queuing delay by using two dimension finite-state Markov model. They gave the queuing delay distribution Pr $(D<d)$. The average queuing delay is defined to be the value $D$ for which the delay distribution is larger than 90%. Thus, the end-to-end delay of a path between two end points can be estimated by adding all the node delays and link delays along the path. An H hops path has average queuing delay given by [12]

$$D(p) = \sum_{i=1}^{H} \bar{D}_i$$

where $\bar{D}_i$ = average queuing delay at node $i$. Due to distributed structure of the routing scheme, this protocol is scalable and adaptive to its node’s mobility.

4. Performance Comparison of AODV and DSR Protocols (Stressful / Less Stressful Situations)


4.1 Dynamic Source Routing (DSR) Protocol

DSR [4] uses source routing and thus the sender has complete knowledge of hop-by-hop route to the destination. These routes are stored in a route cache. Every data packet carries the source route in its packet header. When a mobile node in the Ad Hoc network tries to send a data packet to a destination for which it does not already have the route in its route cache, it initiates a route discovery process.

4.2 Ad Hoc on Demand Distance Vector Routing (AODV) Protocol

The AODV [3] is a reactive routing protocol and shares features of both DSDV [9] and DSR [4] algorithm. AODV shares Dynamic Source Routing (DSR’s) on-demand characteristics in that it also discovers routes as and when needed by initiating a route discovery process. It maintains one entry per destination in its routing tables unlike in DSR, which maintains multiple route entries for each destination in its route cache. In the routing table, there is an entry for each active route. Whenever a source node wants to send data to a destination node, it broadcasts RREQ packet. RREQ packet is generated by either an intermediate node, which has a valid route to the destination or by the destination node. When an active link breaks, the upstream node of broken link broadcasts a route error (RERR) message to the source. The source node invalidates the listed routes and initiates a route discovery process.
5. Performance
In this section, we briefly describe the simulation environment and various parameters chosen to simulate the routing protocols AODV and DSR.

5.1 Simulation Environment and Scenarios
To compare the performance of AODV and DSR routing protocols, simulation experiments were performed using a parallel discrete event-driven simulator, GloMoSim (Global Mobile Information System Simulator) [5]. Table 1 describes the detailed setup used for our simulation. One of the interests of this paper is to test the ability of AODV and DSR routing protocols to react under stressful situation (under heavy load and high mobility). We experimented with 500 seconds of simulated time over a rectangular field. We also added 50 seconds at the beginning of the simulation to stabilize the mobility model. So the first data packet was initiated after 50 seconds only. Each simulation ran for 550 seconds of simulated time. We ran our simulations with movement patterns generated for six different pause times: 50, 70, 110, 170, 350, and 550 seconds.

<table>
<thead>
<tr>
<th>Routing Protocols</th>
<th>AODV, DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Layer</td>
<td>IEEE 802.11 DCF</td>
</tr>
<tr>
<td>Transmission bandwidth of each link</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Radio signal transmission range</td>
<td>250m</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Terrain size</td>
<td>1500m × 300m</td>
</tr>
<tr>
<td>Number of mobile nodes</td>
<td>50 and 100</td>
</tr>
<tr>
<td>Node placement</td>
<td>Uniform</td>
</tr>
<tr>
<td>Simulation time</td>
<td>550 sec (real simulation time = 500sec)</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint [Speed (0-20 m/sec), Pause Time (50,70,110,170,350,550 sec)]</td>
</tr>
<tr>
<td>Packet rate</td>
<td>4 and 6 packets/sec</td>
</tr>
<tr>
<td>Traffic type</td>
<td>Constant Bit Rate (CBR)</td>
</tr>
<tr>
<td>No. Of CBR source destination pairs (flows)</td>
<td>20 and 40</td>
</tr>
</tbody>
</table>

Table 1. Simulation Parameter Values

5.2 Performance Metrics
In comparing both the protocols, the following performance metrics were considered.

- **Packet delivery ratio**: The ratio of the number of data packets successfully delivered to the destinations to those generated by CBR sources.
- **End-to-end delay**: The average time from the beginning of a packet transmission at a source node until packet delivery to a destination. This includes delays caused by buffering of data packets during route discovery, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.
- **Routing overhead**: The number of control packets generated for routing by each routing protocol.

5.3 Simulation Results
The effect of node mobility and offered load were studied as per above simulation parameters. Five experimental runs were repeated for each configuration, with a different starting random seed value for each run.

5.3.1 Scenario I (Stressful Situation)
In the first set of simulation, the traffic was generated by 40 constant bit rate (CBR) sources spreading the traffic uniformly among all nodes to 40 destinations. There were 100 mobile nodes in the network. The packet rate was fixed at 4 packets/s and the size of each packet was 512 bytes. In this scenario, packet delivery rate of AODV is 30 % higher than DSR at high mobility whereas performance of DSR is about 14 % higher at low mobility than AODV (Figure 2). The delays of AODV are smaller than DSR by a factor of about 4 for high mobility, however at low mobility the DSR showed lower delay than AODV (Figure 3).

Routing overhead of DSR was found lower than AODV by a factor of 26 times at high mobility (Figure 4). We also examined the behavior of these two protocols with increase in number of CBR sessions (transfer rate 6 packets/second for each session) from 5 to 40, and observed the fraction of received packets at the destination. In both the protocols, the number of received packets starts decreasing as we increase the number of sessions. But AODV outperforms DSR when the number of sessions is more than 20 (Figure 5).

![Figure 2. Data packet delivery rate at varying mobility for 100 nodes](image)

![Figure 3. Packet delay at varying mobility for 100 nodes](image)
5.3.2 Scenario II (Less Stressful Situation)
In the second set of simulation, there were 50 mobile nodes in the network and 20 CBR sources were transmitting data to 20 destination nodes at the rate of 6 packets/second. From Figure 6, it can be observed that packet delivery rate of both AODV and DSR remain almost same with respect to mobility. However, the performance of AODV is slightly higher when the mobility is high. The delay experienced by AODV is lower than DSR at high mobility but at low mobility DSR outperforms with AODV (Figure 7).

DSR demonstrates significantly lower routing overhead by a factor of 5 times at high mobility (Figure 8) as compared to AODV. When we increase the number of CBR sessions (transfer rate 4 packets/second for each session) from 5 to 40, the number of
received packets starts decreasing as we increase the number of sessions. But AODV slightly outperforms to DSR when the number of sessions becomes more than 20 (Figure 9).

5.4 Observations
Packet delivery ratio and packet delay of AODV and DSR remains almost identical irrespective of the mobility in less stressful situation. However, the performance of AODV in terms of packet delivery is quite better as compared to DSR in the stressful situation at high mobility whereas DSR performs better than AODV at low mobility. In stressful situation, at high mobility DSR packet delay is higher as compared to AODV whereas DSR out performs AODV as pause time becomes large. The performance of DSR starts degrading due to cache staleness as well as high flow of traffic and large size of network at high mobility. The benefit of caching routes is completely lost in case of DSR at high mobility and larger number of flows in big networks. However, at low mobility, route information in route cache always remains up to date. Due to aggressive use of route caching and multiple routes per destination, DSR always shows a lower routing load than AODV. The chances of finding an alternate route from the route cache in DSR avoids frequent route discovery thereby reducing routing load significantly. In AODV, routing load is mostly due to route request packets while in case of DSR, it is mostly by route reply packets.

DSR outperforms AODV significantly in terms of routing overhead in low mobility. However, its performance deteriorates rapidly when the situation gets stressful. This is due to the aggressive usage of source routing cache. During a route discovery process, the source can learn several routes to its destination. This enables the source node to switch to some other cached routes in case of the current route breaks, which significantly reduced the possibility to restart a route discovery process. However, in stressful situations, it is more likely that all cached routes are already invalid, thus introducing unnecessary delay and extra traffic.

AODV sustains better in stressful situation. For one route discovery process, destination node only returns one route reply (RREP) packet. This requires that the source node should restart route discovery process to resume the transmission whenever current route fails. This is beneficial in high mobility situations, where movement of nodes can quickly invalidate current route and cached route entries.

Some authors [1,2,7,8] also carried out performance evaluation in different scenarios of AODV, DSR and other Ad Hoc routing protocols through simulation using ns2. Our observations regarding these two protocols were found close to them. AODV protocol outperforms DSR on performance metrics such as higher packet delivery ratio (PDR), lower end-to-end delay in high mobility situation. Moreover, many QoS routing protocols [13, 31, 32, 33, 34, 35, 36] designed are extension of AODV protocol, we therefore preferred to extend AODV routing protocol over DSR to propose a reactive QoS routing protocol for delay sensitive applications in mobile Ad Hoc networks which we call AODV-D. The protocol modifies and extends QoS-AODV [13] to discover a route with least traffic and maintain the required QoS delay constraint throughout the communication process. This algorithm selects routes with least traffic and follows alternate route method for route maintenance. The protocol estimates node delay dynamically and destination nodes monitor the healthiness of the paths by piggybacking delay information and thus selecting better route before congestion.

Earlier papers [3,13,16] consider only minimum number of hops as route selection metric. For route selection, it considers only those routes which have total path delay less than or equal to that specified in the route request. For calculating path delay, it takes into account MAC delay [11] at each mobile node along the path. For route maintenance, each mobile node in the path piggybacks the delay information on data packets, so that destination mobile node can initiate alternate route discovery in advance of congestion. In this section, we describe our proposed protocol, which includes calculation of forwarding delay (i.e. node delay) at each mobile node, initiation of route discovery and route maintenance processes.

6.1 Calculation of Node Delay
Figure 10 [11] depicts the state transition diagram of a mobile node \( MN_i \), which tries to transmit packets in IEEE 802.11 standard [17]. IEEE 802.11 works in two modes: DCF (Distributed Coordination Function) and PCF (Point Coordination Function). Data transmission in DCF [17] mode is depicted in Figure 11. The forwarding delay at a mobile node \( MN_i \), which includes MAC contention and transmission delay is calculated using equation (2) given in [11].

\[
P_{i\leftarrow MN_i}(t) = e^{-\lambda t} \text{, where } \lambda \text{ is the aggregate arrival rate (including neighbor nodes) at mobile node } MN_i.
\]

\[
D_{i\leftarrow MN_i} = P_{i\leftarrow MN_i}(t) \times \text{avg bt} + DA(i) + (1-P_{i\leftarrow MN_i}(t)) \times (SIFS + DB(i)) + (L/R).
\]

where

\[
P_{i\leftarrow MN_i}(t) \text{ is the probability that mobile node } MN_i \text{ detects no other mobile node transmitting data during time interval } t \text{ and is given by }
\]

\[
P_{i\leftarrow MN_i}(t) = e^{-\lambda t}, \text{ where } \lambda \text{ is the aggregate arrival rate (including neighbor nodes) at mobile node } MN_i.
\]

\[
DA(i) \text{ is expected delay encountered in the Attempt state and is given by }
\]

\[
DA(i) = P_{i\leftarrow sl}(slot) \times (RTS+2 \times SIFS + CTS) + (1-P_{i\leftarrow sl}(slot)) \times (RTS+2 \times SIFS + DB(i))
\]

DB(i) is expected delay encountered during backoff state and is given by

\[
DB(i) = \left[ \frac{1}{P_{i\leftarrow sl}(DIFS)} \times P_{i\leftarrow sl}(slot) \times (DIFS + DB(i)) \right] \times \left[ \frac{P_{i\leftarrow sl}(DIFS) \times P_{i\leftarrow sl}(slot)}{RTS+2 \times SIFS + DB(i)} \times (RTS+2 \times SIFS + DB(i)) \right]
\]

\[
CTS \right] + \left[ 1-P_{i\leftarrow sl}(DIFS) \times X \right] X
\]

avg bt is a random backoff time interval before transmission and is given by

\[
\text{avg bt} = \frac{1}{\text{avg bt}} \times \left( 1 - P_{i\leftarrow sl}(slot)^{n-1} \right) \times \text{X}
\]

L and R are packet length and data rate respectively, W is contention window size,

\[
X = RTS + 3 \times SIFS + CTS + L + ACK
\]

ACK is length of acknowledgement packet

In wireless links, the propagation delays are very small and almost equal for each hop along the path. So, here we assume that the propagation delay is negligible.

The AODV-D protocol has two phases:
(i) Route discovery phase
(ii) Route maintenance phase
Figure 10. State transition diagram of a mobile node (MN) in proposed protocol

Figure 11. Data transmission in IEEE 802.11 (DCF mode) using RTS and CTS

6.2 Route Discovery Phase

To provide end-to-end delay QoS, extensions are done in RREQ, RREP messages and the routing table structure of AODV protocol as shown in Figure 12.

RREQ contains two extra fields: acc_delay and max_delay. The max_delay extension specify a maximum bound on the acceptable time delay experienced on any acceptable path from the source to the destination. Accumulated Value extension field (acc_delay) enables the measurements to be accumulated for end-to-end delay. It provides information about the cumulative value that has been experienced by nodes along the path from the originating node to the node currently processing the RREQ. When a route is required but destination information is not available in the routing table, the source node floods the RREQ packet to discover a route. Every RREQ packet carries the source and destination addresses, a sequence number, hop count, delay parameter (max_delay) and Accumulated Value extension (acc_delay). Initially the value of acc_delay is zero. If the source node wants to discover a QoS route, it records the maximum acceptable delay in max_delay field otherwise it sets it to –1. A node, which receives a RREQ packet measures forwarding_delay as per equation (2) and records this value in its routing table. The forwarding_delay is MAC delay at a mobile node. This value is maintained at every node along the path being discovered in their respective routing table. If forwarding_delay > (max_delay – acc_delay), it immediately drops the RREQ packet, otherwise RREQ is broadcast to its neighbors after acc_delay field of RREQ is updated by adding in it forwarding_delay recorded in the route table. Also intermediate node records the current value of acc_delay in RREQ to the acc_delay field of that node route table. Route entries are created for every pair of source and destination, i.e., for each session of communication since each session may have different delay requirement. This process continues until the RREQ packet reaches the destination node. Since the destination node receives a set of RREQ packets from different paths, it waits for a small timeout to allow all RREQ packets to discover routes. After selecting an optimal route with the lowest value of acc_delay, the destination node unicasts a RREP packet along the reverse route towards the source. RREQs received after generation of RREP are also buffered and used for route maintenance phase. In AODV, RREP packet can be created either by destination node or an intermediate node with a fresh enough route to a destination [3]. Unfortunately, RREP packet can only be generated by destination node in AODV-D, because an intermediate node is not likely to have current enough information about whether the remaining nodes along the path to the destination can also satisfy the requested QoS. When an intermediate node receives a RREP packet, it updates the acc_delay of its route table entry with acc_delay value contained in RREP packet. Certain fields (see Figure 12 a) are added to each route table entry corresponding to each node requesting QoS. These fields are added to notify endpoint nodes in cases where QoS parameter value are agreed upon, but the desired service qualities can no longer be maintained.

The node also updates the fields of RREP packet in the same way as done for a RREQ packet as described above. Route discovery process of AODV-D is illustrated in Figure 13.

6.3 Route Maintenance Phase

AODV-D tries to maintain the QoS delay constraint throughout
the session by selecting alternate path(s). During data transmission, each mobile node appends the delay information to the data packets. Each packet header is time stamped when the mobile node receives a packet. Let $a_i$ and $b_i$ denote the arrival and departure time of the $i^{th}$ packet respectively at node $p$. After the $i^{th}$ packet’s successful transmission at a node $p$, the estimated average total node delay $q^p_i$, which includes contention, queuing and transmission delays at node $p$, is computed using equation (6) [16]:

$$q^p_i = (1 - \alpha) \times q^p_{i-1} + \alpha \times (b_i - a_i)$$

(6)

where $i > 1$, $0 \leq \alpha \leq 1$, and $a_i$, and $b_i$, are arrival and departure time stamps of previous packet $i-1$.

Each intermediate node adds its node delay computed as per equation (3.10) to the piggybacked acc_delay field of each data packet. Thus, destination node monitors the route capacity to serve the QoS requirements of the session. If total path delay (acc_delay) reaches the maximum limit the destination selects next better route from the buffered active routes, those routes which are not expired. If buffer does not contain any fresh routes then it generates route error packet (RERR) in advance of congestion. When a link breaks, then AODV-D tries to rebuild the broken link by doing local repair. The upstream node of a broken link sends a local repair request to find the node following the next node along the route to the destination. This request packet includes the session ID, with the time to live (TTL) value set to 3, which limits the broadcast area of the local repair request. To allow this mechanism, the node following the next node along the route is also recorded in each routing table. If a local repair mechanism fails, then a route error (RERR) packet is set to notify the corresponding source nodes that the link is broken. Those mobile nodes, which receive RERR packet, invalidate the associated route entry.

### 6.4 Pseudocode of AODV-D

The steps of the proposed algorithm (pseudocode) are as under.

#### Route Discovery

**Step 1:** If Source node $S$ has data packets to send and no route is known to the targeted destination

- Then initiate a RREQ with
  - $\text{acc\_delay} \leftarrow 0$; and
  - $\text{max\_delay} \leftarrow d$; // where, $d$ is an upper bound to delay.

**Step 2:** Each intermediate node after receiving RREQ packet, calculates its own node delay as per equation (1) and records in its routing table forwarding_delay field. Then compare its node delay value (i.e. current forwarding delay value) with the value of (max_delay – acc_delay) of RREQ packet.

**Step 3:** If $(\text{max\_delay} - \text{acc\_delay}) > \text{forwarding\_delay}$

```plaintext
{ 
  Then update acc\_delay field of RREQ as acc\_delay ← (acc\_delay + forwarding\_delay); 
  Record acc\_delay of RREQ in acc\_delay field of its routing table; 
  Then broadcast the RREQ to its neighbor nodes; 
}
```

Else

- Drop the RREQ packet;

**Step 4:** The RREQ packet is further broadcasted by intermediate node as per step 3 till destination node is reached.

**Step 5:** If destination node receives RREQ packet satisfying the QoS delay parameter

- Then buffer the route.

**Step 6:** If buffer time expires

- Then select a route with minimal delay and unicasts the RREP with acc\_delay $\leftarrow 0$ in the backward direction;

**Step 7:** If destination receives RERR packet with a RREPFAIL flag due to route repair fail, Then it selects a fresh route, next better route, from buffer and unicasts RREP to the source;

**Step 8:** Intermediate nodes updates acc\_delay field of RREP packet as

- $\text{acc\_delay} \leftarrow (\text{acc\_delay} + \text{forwarding\_delay})$;

**Step 9:** If RREP packet reaches to source node within RREP TIME, and

- If (acc\_delay < max\_delay)
  Then the source node buffers the route and start sending data;
- Else source node $S$ restart route discovery with new session Id;

**Step 10:** If $S$ receives a fresh RREP with same session Id

- Then divert data transmission through new route;

#### Route Maintenance

**Step 1:** Destination node $D$ monitors the healthiness of the ongoing route by examining piggybacked acc\_delay field value of each data packet received and comparing it with route delay constraints i.e. max\_delay;

**Step 2:** It pick up the next optimal route from buffered active routes and sends RREP to source node to use alternate route in the situation of congestion in advance;

**Step 3:** If a node receives link break

- Then upstream node of a broken link perform local route repair by sending a local route repair request with limited broadcast (TTL = 3) to find the node following the next node along the route to destination;

**Step 4:** If local repair successful

- Then update the route table;
- Else send RERR to source and invalidate the associated route entry.

### 7. Simulation Model and Performance Evaluation

To evaluate performance of our proposed AODV-D protocol, we used the GloMoSim simulator [5,20,25] and compared its performance with AODV and QoS-AODV protocols. The simulations were conducted on an Intel Pentium IV processor at 3.0 GHz, 512 MB of RAM running WINDOWS XP.

#### 7.1 Simulation Setup

The performance of AODV-D was compared with the best effort AODV [3] routing protocol and QoS-AODV [13] protocol using GloMoSim. The GloMoSim library is a scalable simulation environment for wireless network systems. It uses parallel discrete-event simulation provided by PARSEC [20,26], which is a C based parallel simulation language. We examined the
effect of node mobility and traffic load on the three protocols. Some simulation parameters common for these protocols are given in Table 2 [11].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot time</td>
<td>20µsec</td>
</tr>
<tr>
<td>SIFS</td>
<td>10µsec</td>
</tr>
<tr>
<td>DIFS (SIFS+2 × slot time=50µsec)</td>
<td></td>
</tr>
<tr>
<td>RTS</td>
<td>352µsec</td>
</tr>
<tr>
<td>CTS</td>
<td>304µsec</td>
</tr>
<tr>
<td>ACK</td>
<td>304µsec</td>
</tr>
<tr>
<td>MSDU (MAC Layer Data Unit)</td>
<td>1500 bits</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250m</td>
</tr>
<tr>
<td>Carrier sensing range</td>
<td>550m</td>
</tr>
<tr>
<td>Arrival Rate</td>
<td>8 packets/sec</td>
</tr>
<tr>
<td>CWmin</td>
<td>32 slots</td>
</tr>
<tr>
<td>CWmax</td>
<td>1024 slots</td>
</tr>
<tr>
<td>PHY scheme</td>
<td>DSSS</td>
</tr>
<tr>
<td>Number of mobile nodes</td>
<td>50</td>
</tr>
<tr>
<td>Terrain size</td>
<td>1500m × 300m</td>
</tr>
<tr>
<td>Simulation time</td>
<td>900 seconds</td>
</tr>
<tr>
<td>Transmission radius</td>
<td>100m</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
<tr>
<td>Traffic</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
</tbody>
</table>

Table 2. Simulation Parameters

7.2 Movement and Traffic Model
The random waypoint model [15] is used to model mobility of nodes. In this, mobile nodes speed is kept in between 0-20 m/second. We varied the traffic load and the degree of mobility in the simulation. We varied the traffic load by varying the number of sessions (i.e. active sources) to be 5, 10, 15, 20, 25 and 30. The pause time was kept at 400 second. We control the degree of mobility by varying the pause times as 0, 100, 200, 300, 400, 500, 600, 700, 800 and 900 and the number of session was 20. The number of mobile nodes in the simulation was 50. Other parameters are given in Table 2. We used Constant Bit Rate (CBR) source as the data source for each mobile node. Each source node transmitted packets at the rate of 8 packets/sec with a packet size of 512 bytes.

7.3 Performance Metrics
In order to investigate the performance of these protocols, we used following performance metrics:

- **Packet Delivery Ratio (PDR):** It is the ratio between the packets received at the destination and the packets generated by the sources.
- **Normalized Routing Overhead:** It corresponds to the ratio between the total control packets sent and forwarded, and the successfully received data packets. Each time a control packet is retransmitted, it is considered as a new control packet.

7.4 Results and Analysis
We present in this subsection the performance of the AODV, QoS-AODV and AODV-D protocols for the various metrics presented above.

7.4.1 Effect of Node Mobility
A simulation model consisting of 50 mobile nodes with 20 active sessions, each with 8 packets/second arrival rate and variable pause time was considered for simulation. To study the effect of mobility, pause time was varied from 0 to 900 seconds with steps of 100 seconds. This section describes the impact of mobility on the performance of AODV-D, QoS-AODV and AODV.

7.4.1.1 Average End-to-End Delay
With high mobility, AODV-D has lesser end-to-end delay compared to QoS-AODV but slightly higher delay than best-effort AODV (with pause time <= 200 ms). This is because of high interference and frequent link breaks. As mobility decreases (with pause time > 200 ms), average end-to-end delay of AODV-D also decreases and it starts outperforming with both AODV and QoS-AODV. After 700 ms pause time, average end-to-end delay remains almost constant. This is because destination nodes go for alternate paths if the congestion occurs during transmission. The average end-to-end delay with variable pause time of mobile nodes for these three protocols is depicted in Figure 14.

7.4.1.2 Normalized Routing Overhead
Routing overhead of AODV-D is similar to QoS-AODV but higher than AODV. This happens because AODV-D uses alternate route strategy in advance of congestion for route maintenance and delay information is piggybacked on data packets. There is a general principle that QoS routing generally incurs a greater overhead than best-effort routing due to the extra information being disseminated. The normalized routing overhead, with variable pause time of the mobile nodes, for these three protocols is depicted in Figure 15.
7.4.1.3 Packet Delivery Ratio
With high mobility, AODV-D has lesser packet delivery ratio compared to AODV, but same as QoS-AODV. This is because AODV-D estimates node delay dynamically. With high mobility, interference from neighboring nodes become high. As mobility decreases (pause time >= 200 seconds), AODV-D outperforms both AODV and QoS-AODV, because of its better route maintenance policy. Up to 600 ms pause time, packet delivery ratio is gradually increased, after 600 msec pause time it becomes almost constant. At low mobility, link breaks are few and routes are balanced. The packet delivery ratio with variable pause time of the mobile nodes for these three protocols is shown in Figure 16.

7.4.2 Effect of Traffic Load
In our model, different traffic rates were simulated using different number of sessions (sessions are varied from 5 to 30 in steps of 5). A simulation model consisting of 50 mobile nodes using random waypoint mobility with 400 sec pause time and with 8 packets/sec arrival rate is considered for simulation. This section describes the impact of traffic load on the performance of AODV-D, QoS-AODV and AODV.

7.4.2.1 Average End-to-End Delay
Average End-to-End delay of AODV-D is almost similar to AODV at low traffic. As traffic increases interference with neighbor nodes is increased resulting in frequent route changes. This causes the increase of average end-to-end delay at high traffic rate. Even though AODV-D estimates node delay dynamically, average end-to-end delay is almost similar to AODV and QoS-AODV. The average end-to-end delay with variable number of sessions for these three protocols is shown in Figure 17.

7.4.2.2 Normalized Routing Overhead
Normalized routing overhead of AODV-D is slightly higher than AODV as it uses local link repairs and congestion control to preserve the QoS requirements. At high traffic, as the load on the system is more, nodes become congested. To preserve QoS delay requirement, destination nodes send new RREP packets to source nodes changing to new routes. This causes the increase of routing overhead. The normalized routing overhead with variable number of sessions for these three protocols is shown in Figure 18.

7.4.2.3 Packet Delivery Ratio
Packet delivery of AODV-D is better than AODV and QoS-AODV at high traffic (number of active sessions >= 15). This shows the efficiency of the AODV-D. Average end-to-end delay

![Figure 15. Normalized routing overhead vs. pause time for 20 sessions at 8 packets/sec traffic rate](image1)

![Figure 16. Packet delivery ratio vs. pause time for 20 sessions at 8 packets/sec traffic rate](image2)

![Figure 17. Average end-to-end delay vs. traffic load with pause time = 400 second and 8 packets/sec traffic rate](image3)

![Figure 18. Normalized routing overhead vs. traffic load with pause time=400 second and 8 packets/sec traffic rate](image4)
and normalized routing overhead is better at high mobility in case of AODV-D than QoS-AODV. The packet delivery ratio is significantly better at high traffic rate as compared to AODV and QoS-AODV. AODV-D tries to maintain the routes with the required low delays. This makes the routes less loaded and resulting lesser packet drops as congestion decreases. Therefore packet delivery ratio increases. With little expense of routing overhead, AODV-D is able to deliver the packets at good percentage. The packet delivery ratio with variable number of sessions for these three protocols is shown in Figure 19. AODV-D is considerably performing well and it is able to meet the delay constraint efficiently up to 20 sessions and above 400 seconds pause time. The average end-to-end delay exceeds 400 ms when the number of sessions increases more than 20.

8. Conclusion and Future Scope

In this paper, first a performance comparison of two on demand routing protocols (AODV and DSR) was presented as a function of traffic load and mobility. As a result of our findings through simulation, it can be said that DSR performs poor in a stressful situations as compared with AODV at high mobility even though the routing overhead is higher in AODV as compared to DSR. DSR starts outperforming as mobility decreases. As the performance of AODV routing protocol has been found better in general than DSR, we decided to modify AODV routing protocol to propose a novel on demand AODV based QoS routing protocol which we call AODV-D for delay sensitive application in mobile Ad Hoc networks. For performance comparison, we considered various numbers of sessions with different packet rates and mobility model. Our proposed protocol discover routes based on path delay in addition to hop count instead of hop count only and also route maintenance is more efficient than existing standard. AODV-D estimates node delay dynamically instead of taking a constant value as in the existing QoS-AODV [13] standard. From these simulation studies, it is evident that our protocol outperforms AODV and QoS-AODV under average mobility and traffic load. These characteristics make the protocol suitable for reliable real time data and voice transmission applications in Ad Hoc networks under IEEE 802.11 DCF as MAC layer protocol. Our proposed protocol provides more accurate estimation of end-to-end delay. It has provision to avoid a congested path by keeping track of Accumulated delay value extension field (acc_delay). Earlier papers [3,13] for delay estimation only considered processing delay at each mobile node along the path. Hence it does not provide accurate estimation of end-to-end delay. In this paper, only QoS metric considered is end-to-end delay for a QoS flow, as finding a route subject to multiple metrics is inherently difficult and in many cases is considered to be an NP-complete problem [28]. It can be extended to bandwidth and other resource reservation schemes. In our protocol, we used IEEE 802.11 DCF as a MAC layer protocol. With right set of parameters, IEEE 802.11e EDCF [27] can be used at this place as a MAC layer protocol.

References


Authors Biographies

Rakesh Kumar received his B.E. degree in Computer Engineering from M.M.M. Engineering College Gorakhpur (U.P.), India in 1990 and his M.E. in Computer Engineering from S.G.S. Institute of Technology and Science, Indore, India in 1994. Since January 2005, he has been a PhD student in the department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee, India. He is a life member of CSI, ISTE and also a Fellow of IETE. His main research interests lie in Mobile Ad Hoc Routing, Quality of Service Provisioning, MANET-Internet Integration and Performance Evaluation.

Anil K. Sarje is Professor in the department of Electronics & Computer Engineering at Indian Institute of Technology Roorkee, India. He received his B.E., M.E. and PhD degrees from Indian Institute of Science, Bangalore in 1970, 1972 and 1976 respectively. He served as Lecturer at Birla Institute of Technology & Science, Pilani, for a short period before joining University of Roorkee (now Indian Institute of Technology Roorkee) in 1987. Dr. Sarje has supervised a large number of M.Tech. dissertations and guided several Ph.D. theses. He has published a large number of research papers in the International and National journals and conferences. He has also served as referee for many reputed Journals like IEE Proceedings, IEEE Transaction on Reliability, Electronics Letters, etc. He has been on a number of AICTE and DOEACC committees. He was a member of All India Board of Information Technology during years 2000-2003. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include Distributed Systems, Computer Networks, Real Time Systems and Network Security.

Manoj Misra is Professor in the department of Electronics & Computer Engineering at Indian Institute of Technology Roorkee, India. He received his B.Tech. degree in 1983 from H.B.T.I., Kanpur and M.Tech. from University of Roorkee in 1986. He did his Ph.D. from Newcastle upon Tyne, UK and joined Electronics & Computer Engineering Department, University of Roorkee (now Indian Institute of Technology Roorkee) in August 1998 as Assistant Professor. Before joining University of Roorkee, he worked in DCM, CMC Ltd., New Delhi, H.A.L. Kanpur and H.B.T.I. Kanpur. He has completed an AICTE funded project “A CORBA framework for distributed mobile applications”, as a co-Investigator with Dr. R. C. Joshi. Dr. Misra has supervised a large number of M.Tech. Dissertations and guided several Ph.D. Theses. He has published a large number of research papers in International and National journals and conferences. He is a member of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include Distributed Computing and Performance Evaluation.