EURASIP Journal on Wireless Communications and Networking 2005:3, 426–436

Supporting QoS in MANET by a Fuzzy Priority Scheduler and Performance Analysis with Multicast Routing Protocols

C. Gomathy

Telematics Lab, Department of Electronics and Communication Engineering, Anna University, Chennai-600 025, India
Email: cgomathy@yahoo.co.uk

S. Shanmugavel

Department of Electronics and Communication Engineering, Anna University, Chennai-600 025, India
Email: ssvel@annauniv.edu

Received 5 November 2004; Revised 9 March 2005; Recommended for Publication by George Karagiannidis

Mobile ad hoc network is an autonomous system of mobile nodes characterized by wireless links. The major challenge in ad hoc networks lies in adapting multicast communication to environments, where mobility is unlimited and failures are frequent. Such problems increase the delays and decrease the throughput. To meet these challenges, to provide QoS, and hence to improve the performance, a scheduler can be used. In this paper we design a fuzzy-based priority scheduler to determine the priority of the packets. The performance of the scheduler is studied with the multicast routing protocols. The scheduler is evaluated in terms of the quantitative metrics such as packet delivery ratio and average end-to-end delay and the results are found to be encouraging.

Keywords and phrases: mobile ad hoc networks, scheduling algorithms, multicast routing protocols, fuzzy logic.

1. INTRODUCTION

Ad hoc network is a collection of wireless nodes, which form a temporary network without relying on the existing network infrastructure or centralized administration. Ad hoc networks form a multihop network, where the communication is over the wireless channel, hopping over several mobile nodes.

In recent years, a number of unicast routing protocols have been proposed. Multicasting routing and packets forwarding in ad hoc networks is a fairly unexplored area. In today's network, data transmission between multiple senders and receivers is becoming increasingly important. There are many applications which send from a single source to multiple destinations or from multiple senders to multiple receivers. Multicasting reduces the communication costs, link bandwidth consumption, sender and router processing, and delivery delay. In addition, it also provides a simple and robust communication mechanism when the receiver's individual addresses are unknown or changeable. It also can improve the utilization of the wireless link, when sending multiple copies of messages and exploit the inherent broadcast property of wireless transmission. Hence, multicasting plays an important role in ad hoc networks.

Many multicast protocols have been proposed for ad hoc networks [1, 2, 3, 4, 5, 6, 7]. The ad hoc multicast routing protocol (AMRoute) [1] is a shared tree protocol, which allows dynamic core migration based on group membership and network configuration. The protocol utilizing increasing id-numbers, (AMRIS), builds a shared tree to deliver multicast data [7]. A multicast extension of ad hoc on-demand distance vector (MAODV) routing protocol has also been proposed [4]. It is unique in using a destination sequence number for each multicast entry. The sequence number is generated by multicast group head to prevent loops and to discard state routes. The on-demand multicast routing protocol (ODMRP), is an ad hoc multicast protocol based on multicast mesh [5]. ODMRP uses soft states. So, learning a group is automatically handled by timeouts. It relies on frequent network-wide flooding when the number of source nodes is large and this may lead to scalability problem. In ODMRP, the control packet overhead becomes more prominent when the multicast group is small in comparison with the entire network. The core-assisted mesh protocol (CAMP) supports multicasting by creating a shared mesh structure [2]. All nodes in network maintain a set of tables with membership

Table 1: Comparison of protocols.

Protocol	Multicast topology	Loop free	Dependence on unicast protocol	Periodic message	Control packet flooding
AMRoute	Hybrid	No	Yes	Yes	Yes
AMRIS	Tree	Yes	No	Yes	Yes
MAODV	Tree	Yes	Yes	Yes	Yes
LAM	Tree	Yes	Yes	No	No
ODMRP	Mesh	Yes	No	Yes	Yes
CAMP	Mesh	Yes	Yes	Yes	No
MCEDAR	Hybrid	Yes	Yes	Yes	Yes
NTPMR	Mesh	Yes	No	No	Yes

and routing information. It classifies nodes in the network as duplex or simplex numbers. It relies on underlying unicast routing protocol, which guarantees correct distances to all destinations within finite time. A new on-demand multicast protocol called node transition probability-based multicast routing (NTPMR) is proposed in [3]. It uses a mesh infrastructure instead of a tree. It minimizes the frequency of control message broadcasts. The reduction of channel overhead makes NTPMR more attractive in mobile wireless networks. A comparison of different multicast protocols is shown in Table 1.

With routes being decided by these multicasting protocols, the transmission of packets is to be performed. For this, a scheduler is used. A scheduler should schedule the packets to reach the destination quickly, which are at the verge of expiry. Scheduling discipline manages the queue of requests awaiting service. Without a scheduler, packets will be processed in FIFO manner and hence there are more chances that more packets may be dropped and hence the network may not meet the QoS target [8, 9, 11]. Typical metrics for providing QoS include delay, loss rate, jitter, bandwidth and so forth. In the proposed scheduler, end-to-end delay and packet delivery ratio are considered to analyse the performance of the network, thus providing QoS.

Ad hoc networks have several features, including possible frequent transmission of control packets due to mobility, the multihop forwarding of packets, and the multiple roles of nodes as routers, sources, and sinks of data, that may produce unique queuing dynamics. The choice of scheduling algorithm to determine which queued packet to process next will have a significant effect on the overall end-to-end performance when traffic load is high. For this, various scheduling algorithms were studied. To experiment and evaluate the scheduler, three multicast protocols, namely, ODMRP, CAMP, and NTPMR, are considered. The protocols are so chosen because they all use mesh configuration but different mechanisms as shown in Table 1.

In this paper, a fuzzy-based priority scheduler is designed and implemented. It schedules the data packets based on its priority index. The priority index is attached to the header of the data packets. Its value is based on the queue length of the node, data rate of the source (which is normalized with respect to channel capacity), and expiry time of the packet. This scheduler favors data packets as compared to control packets. It aims to improve the average throughput by quickly delivering packets with greater remaining hops or distance. The fuzzy-based scheduling algorithm is coded in C language. The C code is linked with GloMoSim [10] and tested. It is found from the results that the proposed fuzzy scheduler improves the packet delivery ratio and decreases the end-to-end delay.

The rest of the paper is organized as follows. Section 2 deals with details of the various scheduling algorithms. Section 3 gives the details of the fuzzy scheduler. Section 4 describes the simulation environment, methodology, and performance metrics used. The simulation results are also presented in Section 4. Finally Section 5 details the conclusions of the paper.

2. SCHEDULING ALGORITHMS

Ad hoc networks have several features that may produce unique queuing dynamics. The choice of scheduling algorithm has a significant effect on the overall end-to-end performance when traffic load is high. This motivated us to evaluate the existing scheduling algorithms and propose a new fuzzy-based scheduler. The effects of setting priorities to control and data traffic are studied. The study is performed with the three multicast protocols as described in the previous section.

There are several scheduling policies for different network scenarios. Different routing protocols use different methods of scheduling. The drop-tail policy is used as a queue management algorithm in all scheduling algorithms for buffer management. For the scheduling algorithms that give high priority to control packets, different drop policies are used for data and control packets when the buffer is full. When the incoming packet is a data packet, the data packet is dropped. When the incoming packet is a control packet, the last enqueued data packet is dropped. If queued packets are control packets, the incoming control packet is dropped. Except for the no-priority scheduling algorithm, all the other scheduling algorithms give higher priority to control packets than to data packets. The differences in the algorithms are in assigning priority between data packets. In no-priority scheduling, both control and data packets are served in FIFO order. In the priority scheduling, control and data packets are maintained in separate queues in FIFO order and high priority is assigned to control packets. Currently, only this scheme is used in mobile ad hoc networks [11].

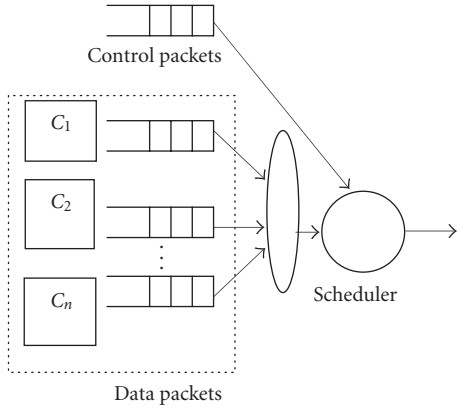

Figure 1: Priority scheduler for data packets.

When looking onto the effect of setting priorities to data packets and considering the suitability of the different types of scheduling algorithms for MANET, several scheduling schemes were studied in literature. In order to consider the effect of setting priorities to data packets, these schedulers give high priority to control packets [11]. Their differences are in assigning priorities among data queues. Figure 1 shows the priority scheduler for data packets. Weighted-hop and weighted-distance scheduling methods use the distance metrics. Weighted-hop scheduling gives higher weight to data packets that have fewer remaining hops to traverse. If the packet has fewer remaining hops, then it has to reach the destination quickly. The data packets can be stored in round-robin fashion. The remaining hops to traverse can be obtained from packet headers. Weighted-distance scheduling gives higher weight to data packets which have shorter geographic distances. The remaining distance is the distance between a chosen next hop and a destination. Round-robin scheduling maintains per-flow queues. The flow can be identified by a source and destination pair. Here each flow queue is allowed to send one packet at a time in a round-robin fashion. In the greedy scheduling scheme, each node sends its own data packets before forwarding those of other nodes [8]. The data packets of other nodes are serviced in FIFO order.

Two other schedulers are the earliest deadline first (EDF) and the virtual clock (VC) [9]. In EDF, a packet arriving at time t and having delay bound d has a deadline $t + d$. The packets will be scheduled based on this deadline. In VC, a packet with size L of a flow, with rate r, has a priority index L/r plus the maximum of current time t and priority index of the flow's previous packet. In these scheduling algorithms, the parameters used to find the priority of data packets are remaining hops to traverse, distance, per-flow queues, greediness of nodes, delay bound, and flow rate.

With the thorough study of ad hoc networks, and the above-mentioned scheduling algorithms, it is found that a number of metrics can be combined into a single decision so as to find the crisp value of the priority of packets. Our solution to determine the priority index of the packets utilizes the fuzzy logic concept [12, 13]. The three input variables, namely, expiry time of packet, queue length of the node, and data rate of the source, are considered and the application of fuzzy logic to combine these variables and hence find the priority index of the packet is found to be suitable. This led to the design of a fuzzy-based priority scheduler.

3. THE FUZZY SCHEDULER

3.1. Fuzzy logic

Fuzzy logic implements human experiences and preferences via membership functions and fuzzy rules. The application of fuzzy logic to problems of traffic control in networks is more attractive. Since it is difficult for a network to acquire complete statistics of the input traffic, it has to make a decision based on incomplete information. Hence the decision process is full of uncertainty. It is advantageous to use the fuzzy logic in the target system because it is flexible and capable of operating with imprecise data.

Basically the fuzzy system consists of four blocks, namely, fuzzifier, defuzzifier, inference engine, and fuzzy knowledge base. The following section explains the working of the general fuzzy system.

Fuzzification of inputs and outputs

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). A fuzzy set A in the universe of discourse U is a set of ordered pairs $\{(x_1, \mu_A(x_1)), (x_2, \mu_A(x_2)), \ldots, (x_n, \mu_A(x_n))\}$, where $\mu_A : U \to [0, 1]$ is the membership function of the fuzzy set A and $\mu_A(x_i)$ indicates the membership degree of x_i in the fuzzy set A.

Fuzzy inference process

If a fuzzy system has n inputs and a single output, its fuzzy rules R_j can be of the following general format.

(R_j) If X_1 is A_{1j}, X_2 is A_{2j}, X_3 is $A_{3j}, \ldots$, and X_m is A_{mj}, then Y is B_j. The variables $X_i\{i = 1, 2, 3, \ldots, n\}$ appearing in the antecedent part of the fuzzy rules R_j are called the input linguistic variables, the variable Y in the consequent part of the fuzzy rules R_j is called the output linguistic variable. The fuzzy sets A_{ij} are called the input fuzzy sets of the input linguistic variable X_i and the fuzzy sets B_j are called the output fuzzy sets of the output linguistic variable Y of the fuzzy rules R_j.

Implication method

Before applying the implication method, the rule's weight must be taken care of. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Once proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weighs appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function

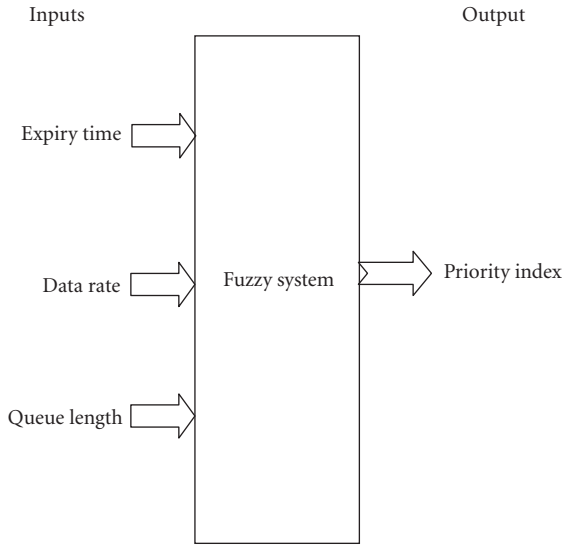

Figure 2: Fuzzy priority scheduler.

associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a membership function, implemented for each rule.

Aggregation of all outputs

Since decisions are based on the testing of all of the rules, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation occurs only once, for each output variable, just prior to the final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

Defuzzification

As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. The most popular defuzzification method is the Centroid calculation, which returns the center of area under the curve. By Centroid method of defuzzification, the crisp output η is calculated using the formula

$$\eta = \frac{1}{\Sigma \mu_{x1\cdots xn}^{\text{output}}(y)} \Sigma y \mu_{x1\cdots xn}^{\text{output}}(y), \tag{1}$$

where y is the center point of each of the output membership function in the output fuzzy set B_j and $\mu_{x1\cdots xn}^{\text{output}}(y)$ is the strength of the output membership function [7].

3.2. Fuzzy scheduler

The proposed fuzzy scheduler, with three inputs, namely, expiry time (E), data rate (D), and queue length (Q), and one output, priority index, is shown in the Figure 2. Here, the process is considered as multiple input and single output (MISO) system.

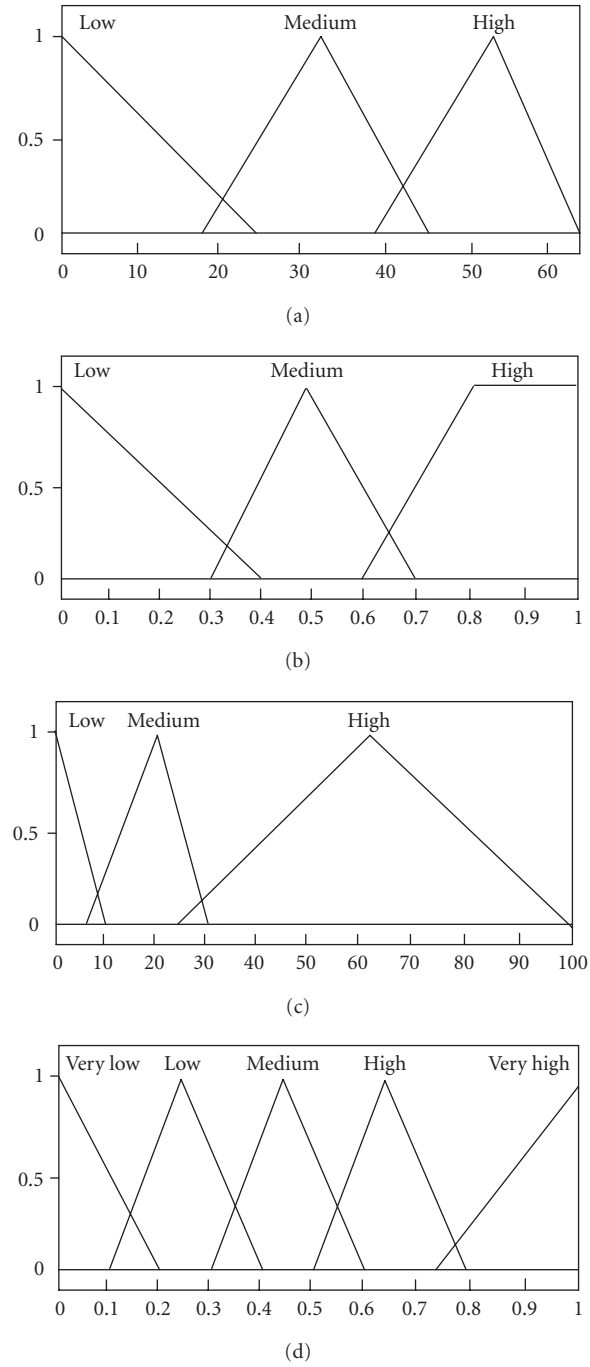

Figure 3: Membership functions: (a) expiry time; (b) normalized data rate; (c) queue length; and (d) priority index.

The linguistic terms associated with the input variables are low (L), medium (M), and high (H). Triangular membership functions are used for representing these variables except for the high data rate where a trapezoidal function is used. The membership functions and rule bases of the scheduler are shown in Figure 3. The bases of functions are chosen

Table 2: Fuzzy rule base.

D \ Q	L	M	H
Expiry time (low)			
L	L	L	VL
M	VL	VL	VL
H	L	VL	VL
Expiry time (medium)			
L	M	M	L
M	M	M	L
H	M	M	M
Expiry time (high)			
L	VH	VH	H
M	H	M	M
H	H	H	M

so that they result in optimal value of performance measures. For the output variable, priority index, five linguistic variables are used. Only triangular functions are used for the output [12, 13].

The rules are defined with due care and are shown in Table 2. To illustrate one rule, the first rule can be interpreted as follows:"If expiry time is low, data rate is low, and queue length is low, then priority index is low." Since in this rule, data rate and queue length are low and packets are associated with low delay, the priority index is set to be low. The ninth rule is interpreted as "If expiry time is low, data rate is high, and queue length is high, then priority index is very low." In this rule, even though the expiry time remains same, since the data rate and queue length are high, priority index is set to be very low. Similarly, the other rules are framed. The priority index, if very low, indicates that the packets are associated with the highest priority and will be scheduled immediately. If the index is very high, then packets are with the lowest priority and will be scheduled only after high priority packets are scheduled. The surface viewer for the fuzzy scheduler is shown in Figures 4a and 4b.

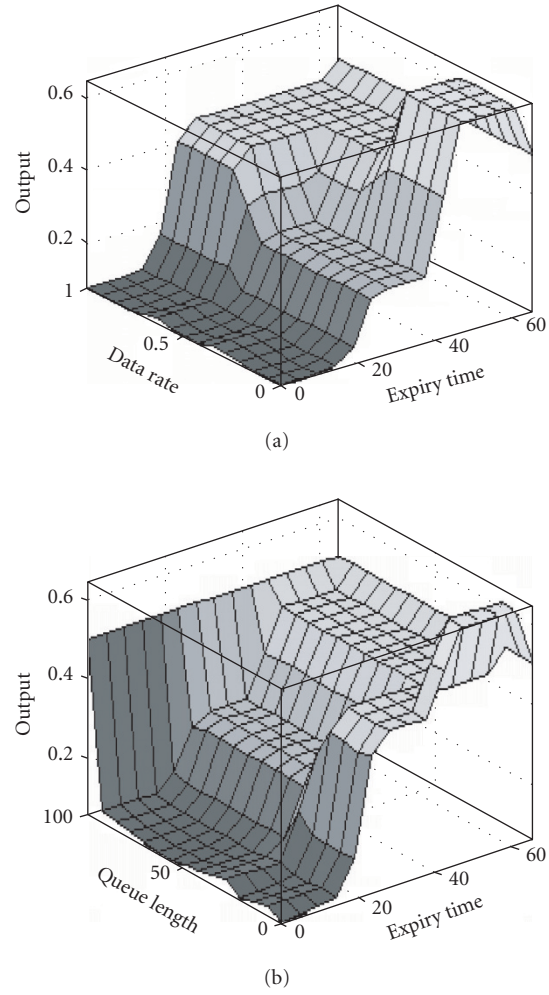

Figure 4: (a) Surface viewer for the fuzzy scheduler in case of constant queue length. (b) Surface viewer for the fuzzy scheduler in case of constant data rate.

4. PERFORMANCE EVALUATION

The fuzzy scheduler is tested using the public domain simulator, GloMoSim [2, 14]. The algorithm is evaluated in terms of packet delivery ratio and end-to-end delay and the results are presented in this section.

4.1. Simulation environment and methodology

The simulation for evaluating the fuzzy scheduler was implemented within the GloMoSim Library. The simulation package GloMoSim [10] is used to analyze and evaluate the performance of the proposed fuzzy scheduler. The GloMoSim (GLObal MObile information system SIMulator) provides a scalable simulation environment for wireless network systems. It is designed using the parallel discrete-event simulation capability provided by PARSEC (PARallel Simulation Environment for Complex Systems) [15]. It is a C-based simulation language developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete event simulation model.

In the simulation, a network of mobile nodes placed randomly within a 1000×1000 meter area is modeled. Radio propagation range for each node was 250 meters [3, 5, 16] and a channel capacity of 2 Mbps is chosen. There were no network partitions throughout the simulation. Table 3 indicates the simulation environment for analyzing the performance of the scheduler.

Table 4 lists the simulation parameters, which are used as default values unless otherwise specified. Multiple runs with different seed values were conducted for each scenario and collected data was averaged over those runs. A traffic generator was developed to simulate CBR sources and FTP items. The size of the data payload is 512 bytes. Data sessions with randomly selected sources and destinations were simulated. Each source transmits data packets at a minimum rate of 4 packets/s and a maximum rate of 10 packets/s. The traffic load is varied by changing the number of data sessions and the effect is examined on the scheduler with different routing protocols.

Table 3: Simulation environment.

Processor	450 MHz, PIV
Hard disk	40 GB
RAM	128 SDRAM
Operating system	Windows 2000

Table 4: Simulation parameters.

Frequency of operation	2.4 GHz
Number of nodes	30
Node placement	Random, uniform
Mobility model	Random way point
Node pause time	0–10 s
Propagation model	Free space
Received power threshold	−81 dBm
Transmitted power	7.89 dBm
Transport layer	UDP, TCP
Network-layer routing protocols	NTPMR, ODMRP, CAMP
MAC	IEEE 802.11

Table 5: Comparison of FPS with other schedulers.

Pause time (s)	Average throughput (packets/s)					
	NPS	PS	WHS	RR	GS	FPS
0	1.7	1.8	1.8	1.75	1.75	1.9
50	1.8	1.9	1.95	1.85	1.85	1.95
100	1.85	1.95	2.0	1.9	1.95	2.1
300	2.0	2.1	2.2	2.1	2.1	2.25
600	2.2	2.3	2.3	2.3	2.3	2.4
900	2.7	2.85	2.9	2.8	2.75	2.95
Pause time (s)	Delay (s)					
	NPS	PS	WHS	RR	GS	FPS
0	3.75	2.25	2.25	2.25	2.25	2.15
50	3.18	2.05	2.1	2.1	2.1	1.9
100	2.9	1.8	1.7	2.05	1.75	1.85
300	3.7	2.3	2.2	2.5	2.25	2.15
600	3.8	3.2	2.5	3.1	3.15	2.35
900	3.2	2.9	2.3	3.0	3.0	2.1

4.2. Performance metrics

The following metrics are used to evaluate the effect of the modified fuzzy scheduler.

(i) *Packet delivery ratio*. Packet delivery ratio is the ratio of the number of data packets actually delivered to the destinations to the number of data packets supposed to be received. This number presents the effectiveness of the protocol.

(ii) *Average end-to-end delay*. This indicates the end-to-end delay experienced by packets from source to destination. This includes the route discovery time, the queuing delay at node, the retransmission delay at the MAC layer, and the propagation and transfer time in the wireless channel.

4.3. Performance evaluation using GloMoSim

The simulation for evaluating the proposed fuzzy scheduler is implemented using GloMoSim Library. First the task of identification of input variables used in the fuzzy logic C code is performed. Then the calculated priority index is used for scheduling the packet. By this way of scheduling, the packets, which are about to expire, or the packets in highly congested queues are given first priority for sending. As a result of this, the number of packets delivered to the client node and the end-to-end delay of the packet transmission improve.

The inputs to the fuzzy system are identified by a complete search of the GloMoSim environment. The input expiry time is the variable TTL, which is present in the network layer of the simulator. TTL stands for time to live and is set a default value of 64 seconds. If the packet suffers excessive delays and undergoes multihop, its TTL falls to zero. As a result of this, the packet is dropped. If this variable is used as an input to the scheduler for finding the priority index, a packet with a very low TTL value is given the highest priority. Hence due to this, the dropping of packets experiencing multihops gets reduced.

The next input to the scheduler is the data rate of transmission and it is normalized with respect to the channel bandwidth. The third input to the scheduler is the queue length of the node in which the packet is present. If the packet is present in a highly crowded node, it suffers excessive delays and gets lost. So, such a packet is given a higher priority and hence it gets saved.

The priority index is calculated with the inputs obtained from the network layer. This is then added to the header associated with the packet. Hence whenever the packet reaches a node, its priority index is calculated and it is attached with it. The buffer is shared by multiple queues when the scheduler maintains multiple queues [17]. Here we consider that each node has three queues. Each queue in the node is sorted based on the priority index and the packet with the lowest priority index (i.e., packet with the highest priority) is scheduled next when the node gets the opportunity to send. By this method of scheduling, the overall performance increases.

4.4. Comparison of FPS with other scheduling algorithms

The scheduling algorithms such as no-priority scheduling (NPS), priority scheduling (PS), weighted-hop scheduling (WHS), round-robin scheduling (RR), greedy scheduling (GS), and fuzzy-based priority scheduling (FPS) are compared under various mobility conditions, with DSR (dynamic source routing) as the underlying unicast protocol. The results are shown in Table 5. Amongst the first five algorithms, the WHS algorithm performs better under high mobility conditions [11]. For the same scenario, when the FPS is evaluated, it provides high throughput compared to all other scheduling algorithms. This is due to the fact that now the queue length, data sending rate, as well as the packets expiry time are taken into account for the crisp calculation of priority index.

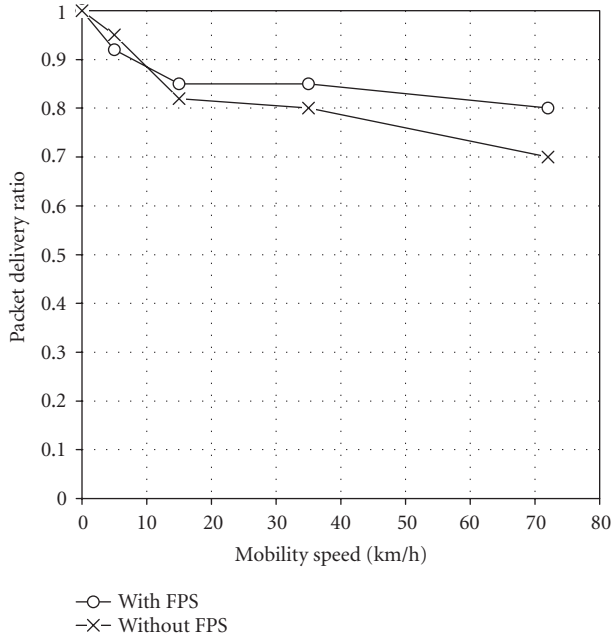

Figure 5: PDR versus mobility for NTPMR.

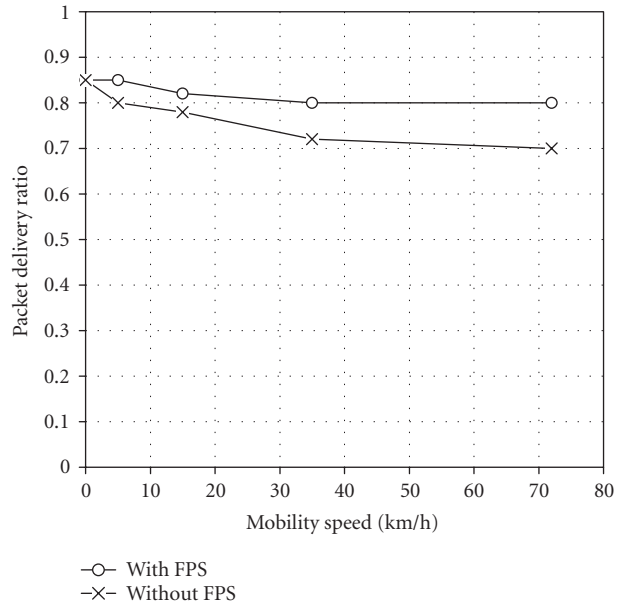

Figure 6: PDR versus mobility for ODMRP.

Moreover as also seen from the delay characteristics, FPS reduces delay by 8% compared to WHS under low mobility conditions. With moderate mobility, the reduction in delay is still significant with FPS. Under high mobility conditions, the reduction in delay is negligible. As seen from the simulation results, with high mobility, most of the packets in the queue are control packets. So setting priorities in data traffic does not change much the servicing order of packets in the queue. Greedy and round-robin scheduling show little difference in performance compared to FPS. In case of greedy scheduling, looking at the performance of individual flows, some flows are severely penalized, although the overall performance does not change. In case of round-robin scheduling, the small difference in performance is due to source type being CBR. With a bursty source, the effect of RR will be higher. Hence, these results prove that FPS performs better compared to all other scheduling algorithms.

Variations in mobility

In this simulation, each node is moved constantly with a predefined speed. Moving directions of each node were selected randomly and when nodes reached the simulation terrain boundary, they bounced back and continued to move. The node movement speed was varied from 0 km/h to 72 km/h. In the mobility experiment, twenty nodes are multicast members and five sources transmit packets. It is evident from the results that NTPMR provides higher packet delivery ratio as compared to ODMRP and CAMP [18]. NTPMR enables packets to travel distant destinations since a packet is sent to different neighbors during repeated encounters with a node, resulting in high packet delivery ratio. Lack of periodic updates and updates only under conditions of packet drops leads to decrease in PDR

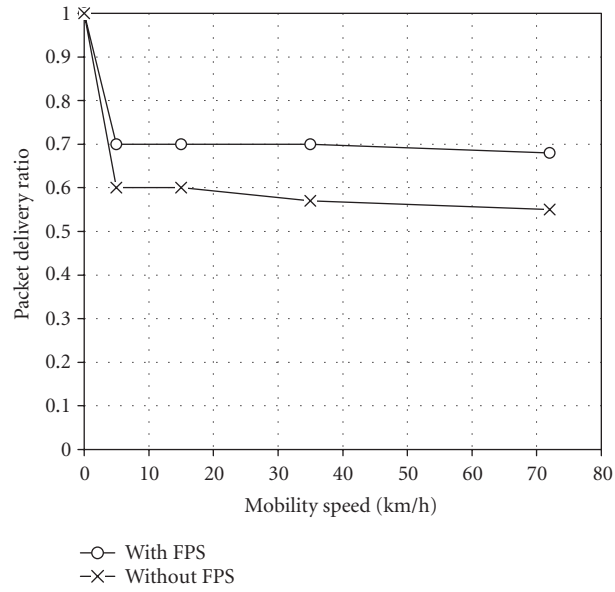

Figure 7: PDR versus mobility for CAMP.

at high mobility. In ODMRP, control packets are transmitted periodically, which results in collisions and congestion. This causes low PDR even at low mobility rates. In CAMP, due to fewer redundant paths, they are prone to link breaks.

It is now proposed to include the fuzzy scheduler for these three protocols and test whether there is any improvement in packet delivery ratio. As seen from the Figures 5, 6, and 7, the packet delivery ratio (PDR) increases by 7%–12% for all protocols.

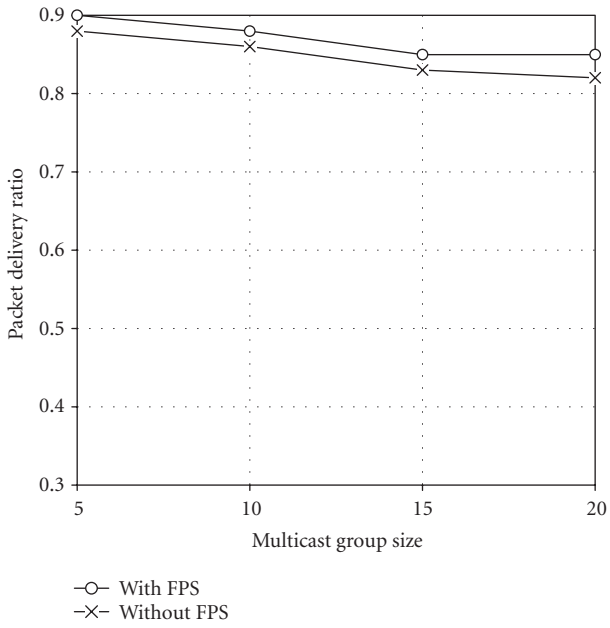

Figure 8: PDR versus group size for NTPMR.

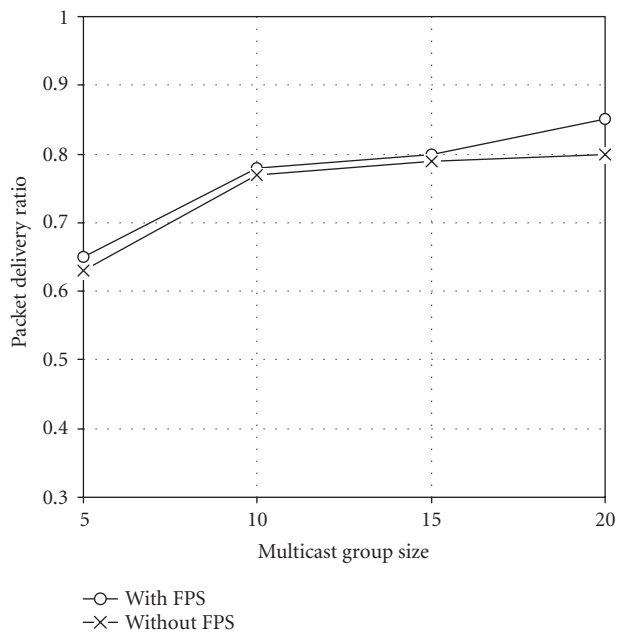

Figure 9: PDR versus group size for ODMRP.

This is due to the fact that the crisp calculation of priority index leads to scheduling of packets in an orderly way. Hence even at higher mobility speeds of nodes, the packets are able to reach the destination and thus improving the PDR. Hence it is verified that even at high mobility speeds, the multicast routing protocols could be used.

Multicast group size

The number of multicast members was varied to investigate the scalability of the protocol. The number of senders is fixed at five, the mobility speed at 1 m/s, network traffic rate at 10 packets/s and the multicast group size is varied from 5 to 20 members. The routing effectiveness of the three protocols as a function of multicast group size is compared [3].

For NTPMR, the packet delivery ratio is found to remain constant with the increase in group size. Here the routing of packets does not depend on any forwarding group. CAMP performs better as the number of groups increases. Since the mesh becomes more massive with the growth of members, more redundant routes are formed. In ODMRP, as the number of receivers increases, the number of forwarding group nodes increases; this in turn increases the connectivity.

With these results, the fuzzy scheduler is inserted in-between the MAC layer and the routing agent. The simulation is run and the results are presented in Figures 8, 9, and 10. As seen from the figures, the NTPMR shows an increased performance of 3%. This is again due to the fact that, as the data packet scheduler is added, the packets at the verge of expiry are scheduled immediately, which in turn increases the PDR. For ODMRP, the PDR characteristics with FPS are closer to those without FPS. Again in CAMP, the PDR improves by 5% due to the proper selection of the priority index.

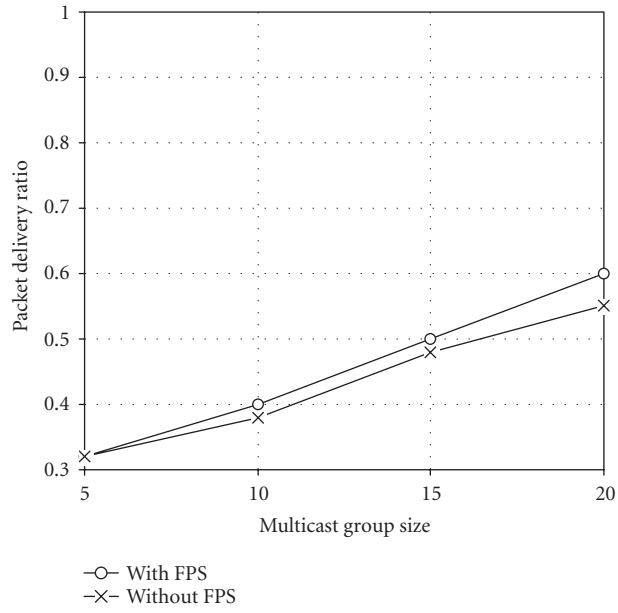

Figure 10: PDR versus group size for CAMP.

Delay performance

The average delay of no-priority and fuzzy-based priority scheduling algorithms are now studied. The use of priority for data packets has a greater impact on delay reduction as mobility increases as shown in the figures. As the nodes move without a pause and when the priority is given to control packets, the delay distribution shifts left as seen from the cumulative distribution function (CDF) [11]. This is because giving high priority to control packets helps notify the source of the route discovery or route error quickly. With low

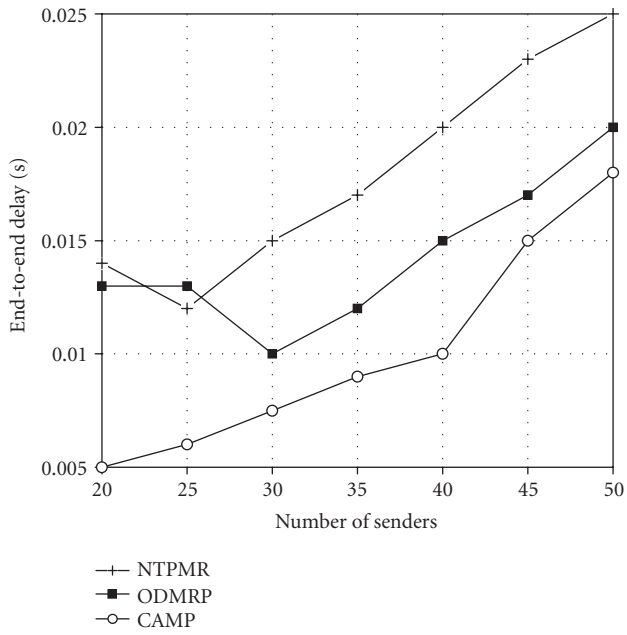

Figure 11: Delay versus senders for all protocols.

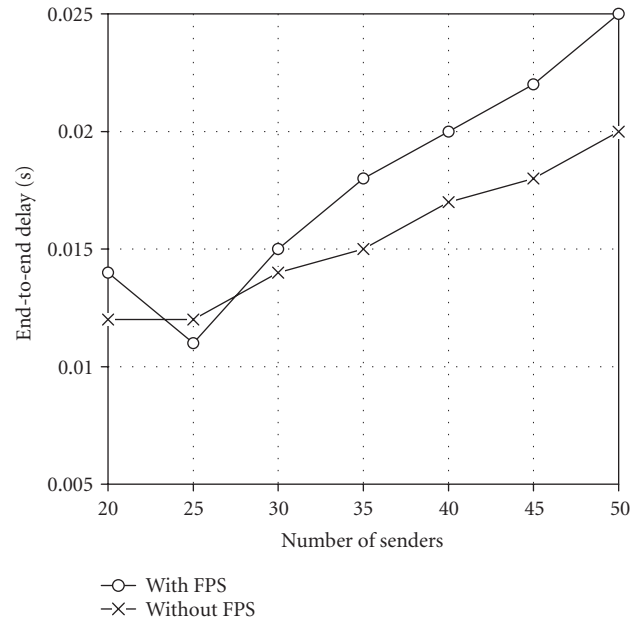

Figure 12: Delay versus senders for NTPMR.

mobility, the CDFs of the no-priority and priority schedulings are almost the same. So under low mobility, since most of the packets in queue are data packets, giving high priority to control packets only improves delay slightly and does not improve the packet delivery ratio.

We now evaluate the effects of setting priority to data packets, varying the number of senders. The delay curve for all three protocols is shown in Figure 11.

After inclusion of FPS, the delay performance is again evaluated and plotted. From Figure 12, it is clear that when the number of senders is lesser than 25, NTPMR shows a reduction in delay by about 20 milliseconds. With low number of senders, setting priorities among data packets has a greater impact. Now the reduction in delay is more significant. For senders up to 30, the performance is better. But as the number increases above 30, it shows a poor performance due to increase in the number of collisions. ODMRP and CAMP show consistent reduction in delay for increase in the number of senders, as seen from Figures 13 and 14. This is due to the maintenance of redundant paths at high number of senders and scheduling of data packets based on priority index set by FPS.

Variations in mobility

In this simulation, the same mobility conditions are employed. The node movement speed or mobility of nodes is varied from 0 to 18 m/s. The routing protocols are chosen to be NTPMR and ODMRP. As the protocols are run with and without the fuzzy scheduler, the end-to-end delay is measured and plotted in graphs as shown in Figures 15 and 16.

As seen from Figure 15, the inclusion of scheduler for NTPMR definitely reduces the end-to-end delay whereas it increases the delay as far as ODMRP is concerned. In NTPMR, the increased delay was the main constraint, which is overcome by the inclusion of the novel fuzzy scheduler. The scheduler, in context of delay performance, is not very superior for ODMRP as seen from Figure 16 and proper modification could be done in rule bases and membership functions so as to meet with the specifications of the routing protocol.

5. CONCLUSION

In this paper, we have analyzed the performance of the novel fuzzy-based priority scheduler for data traffic and evaluated the effect of inclusion of this scheduler with different underlying multicast routing protocols, like NTPMR, CAMP, and ODMRP, run over IEEE 802.11 as the MAC protocol. Queuing dynamics with different degrees of mobility and routing protocols show that the composition of packets in the queue determines the effect of giving priority to control packets or setting priorities among data packets, for the average delay. During low mobility, the average delay is dominated by network congestion due to data traffic. During high mobility, it is dominated by route changes.

We have addressed a fuzzy-based priority scheduler for data packets, which improves the quality-of-service parameters in mobile ad hoc networks. The fuzzy scheduler attaches a priority index to each packet in the queue of the node. Unlike the normal sorting procedure for scheduling packet, a crisp priority index is calculated based on the inputs such as queue length, data rate, and expiry time of packets, which are derived from the network. The membership functions and rule bases of the fuzzy scheduler are carefully designed. The coding is done in C language and the output is verified using Matlab fuzzy logic toolbox with FIS editor. Then the inputs are identified in the library of GloMoSim and the fuzzy scheduler is attached.

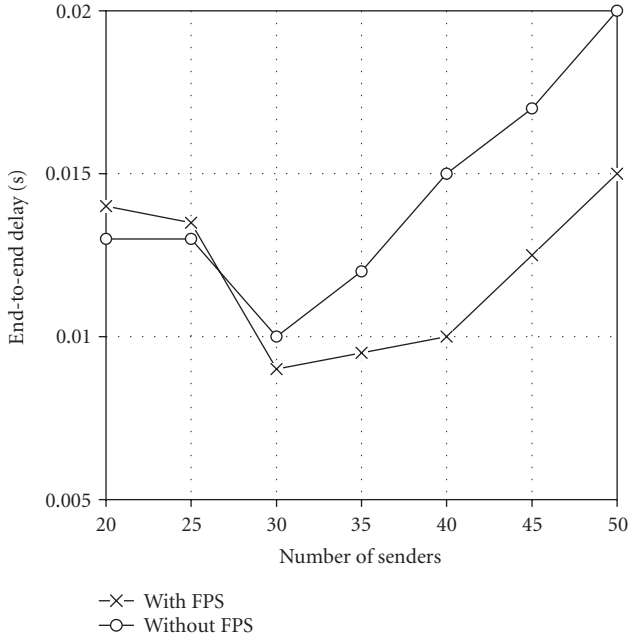

Figure 13: Delay versus senders for ODMRP.

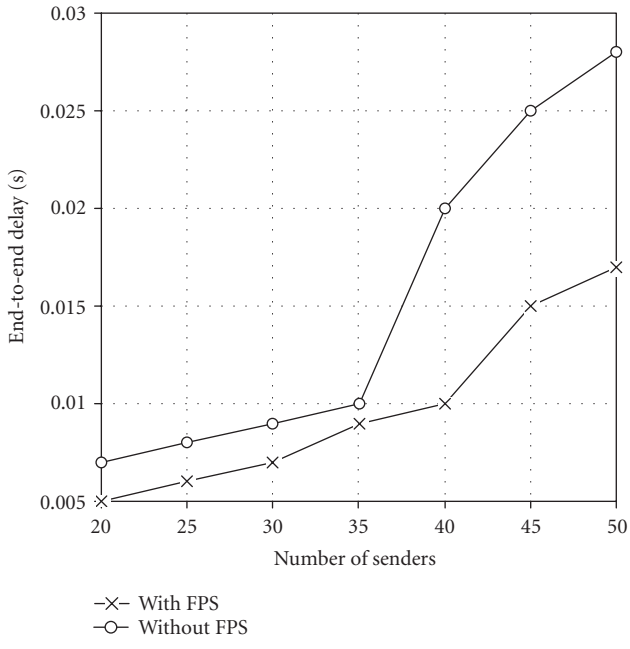

Figure 14: Delay versus senders for CAMP.

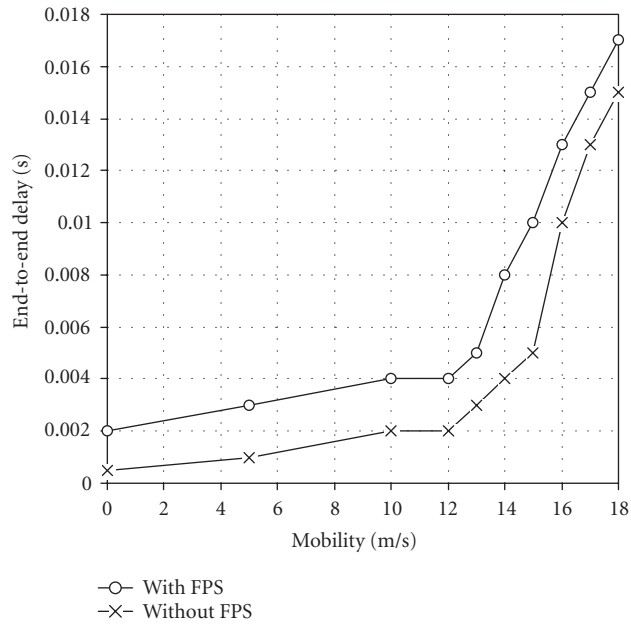

Figure 15: Delay versus mobility for NTPMR.

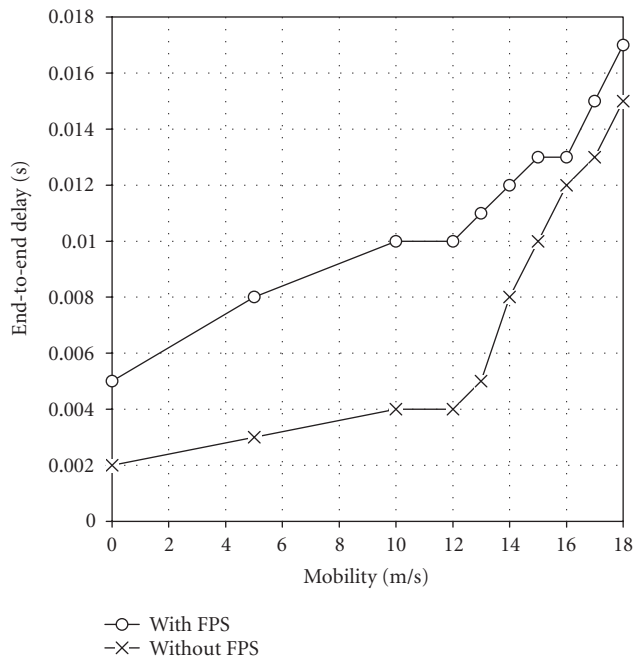

Figure 16: Delay versus mobility for ODMRP.

In this paper, the performance of the fuzzy scheduler is studied for mobile ad hoc networks using GloMoSim simulator and results are presented. It is found from the results that priority scheduling helps in effective routing of packets without much loss and with less delay. In a real network environment, where timely reception of each packet plays a crucial role, priority scheduling helps in effective transmission of packets. Based on the studies, we conclude that the proposed fuzzy-based scheduling algorithm performs better compared with the network performance without scheduler. The results are verified for the multicast routing protocols, such as NTPMR, CAMP, and ODMRP, and they are found to be encouraging.

REFERENCES

[1] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AM-Route: Ad hoc Multicast Routing Protocol," Internet draft, draft-talpade-manet-amroute.00.txt, August 1998.

[2] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The core-assisted mesh protocol," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1380–1394, 1999, Special Issue on Ad hoc Networks.

[3] S. Radha and S. Shanmugavel, "Multicasting in ad hoc networks using NTP protocol," in *Proc. 15th International Conference on Computer and Communication (ICCC '02)*, pp. 144–160, Bandra, Mumbai, India, August 2002.

[4] E. Royer and C. E. Perkins, "Multicast ad hoc on demand distance vector routing (MAODV)," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 207–218, Seattle, Wash, USA, August 1999.

[5] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '99)*, vol. 3, pp. 1298–1302, New Orleans, La, USA, September 1999.

[6] C. W. Wu, Y. C. Tay, and C. K. Toh, "Ad hoc multicast routing protocols utilizing increasing id numbers (AMRIS) functional specifications," Internet draft, draft-ietf-manet-spec-00.txt, November 1998.

[7] http://www.computing.surrey.ac.uk/courses/cs364/FuzzyLogicFuzzySystems_3.ppt.

[8] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *Proc. 6th Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '00)*, pp. 76–86, Boston, Mass, USA, August 2000.

[9] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed priority scheduling and medium access in ad hoc networks," *Wireless Networks*, vol. 8, no. 5, pp. 455–466, 2002.

[10] GloMoSim User manual, http://pcl.cs.ucla.edu/projects/glomosim.

[11] B.-G. Chun and M. Baker, "Evaluation of packet scheduling algorithms in mobile ad hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 36–49, 2002.

[12] C. Gomathy and S. Shanmugavel, "Performance evaluation of a novel Fuzzy based Priority Scheduler for mobile Ad Hoc networks and its effect on MAC protocols," in *Proc. 12th International Conference on Advanced Computing & Communication (ADCOM '04)*, Ahmedabad, Gujarat, India, December 2004, at IEEE Gujarat section.

[13] C. Gomathy and S. Shanmugavel, "Implementation of modified Fuzzy Priority Scheduler for MANET and performance analysis with mixed traffic," in *Proc. 11th National Conference on Communication (NCC '05)*, Indian Institute of Technology, Kharagpur, India, January 2005.

[14] C. Gomathy and S. Shanmugavel, "Fuzzy based Priority Scheduler for mobile ad hoc networks," in *Proc. 3rd International Conference on Networking (ICN '04)*, Gosier, Guadeloupe, French Caribbean, February–March 2004.

[15] R. Bagrodia, R. Meyer, M. Takai, et al., "Parsec: a parallel simulation environment for complex systems," *IEEE Computer*, vol. 31, no. 10, pp. 77–85, 1998.

[16] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," in *Proc. 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 565–574, Tel Aviv, Israel, March 2000.

[17] C. Gomathy and S. Shanmugavel, "Effect of Packet Scheduling and Evaluation of fuzzy based Priority scheduler on Ad Hoc network Unicast communication," in *Proc. IEEE International Conference on Signal Processing and Communications (SPCOM '04)*, Indian Institute of Science, Bangalore, India, December 2004.

[18] C. Gomathy and S. Shanmugavel, "An efficient fuzzy based priority scheduler for mobile ad hoc networks and performance analysis for various mobility models," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 2, pp. 1087–1092, Atlanta, Ga, USA, March 2004.

C. Gomathy acquired a B.E. (with honors) degree in electronics and communication engineering from Government College of Engineering, Tirunelveli, in the year 1986, and an M.S. degree in electronics and control engineering from Birla Institute of Technology and Science, Pilani, in 1992. She also obtained an M.S. (by research) degree from Anna University in 2001. She is currently pursuing the Ph.D. in the Department of Electronics and Communication Engineering, College of Engineering, Anna University, Chennai, India. She has published over 18 research papers in national and international conferences and journals. Her areas of interest include mobile ad hoc networks, high-speed networks, and digital communication.

S. Shanmugavel graduated from Madras Institute of Technology in electronics and communication engineering in 1978. He obtained his Ph.D. degree in the area of coded communication and spread-spectrum techniques from the Indian Institute of Technology (IIT), Kharagpur, in 1989. He joined the faculty of the Department of Electronics and Communication Engineering at IIT, Kharagpur, as a Lecturer in 1987 and became an Assistant Professor in 1991. Presently, he is a Professor in the Department of Electronics and Communication Engineering , College of Engineering, Anna University, Chennai, India. He has published more than 68 research papers in national and international conferences and 15 research papers in journals. He has been awarded the IETE-CDIL Award in September 2000 for his research paper. His areas of interest include mobile ad hoc networks, ATM networks, and CDMA engineering.