QOS IN GENETIC ALGORITHM DERIVED FUZZY BASED AD-HOC NETWORK

Mercy Sharon Devadas

Department of Electrical Engineering, IIT Madras
Chennai-600036. India.
mercy_devadas@yahoo.in

Abstract

Wireless networks are undergoing changes on many levels and are becoming more and more ubiquitous in recent years, ranging from mobile analog and digital cellular telephony to satellite broadcasting. Therefore, a high Quality of Service (QoS) in delivering voice, video and data in this context has emerged as one of the most important areas of challenge of the new century. QoS is especially important for the new generation of Internet applications such as VoIP, video-on-demand and other consumer services. Packet Delivery Ratio (PDR), number of nodes, latency (delay), mobility and error rate are elements of network performance that fall within the scope of QoS.

The allocation of network resources to different users looms as an area of major concern in communication networks of embedded systems. A limited amount of resources has to be shared among many different competing traffic flows in an efficient way in order to maximize the performance and the use of network resources. The current existing system adheres to the use of FIFO Scheduling algorithm.

In FIFO when the queue becomes full, congestion occurs and the new incoming packets are dropped. The queuing delay increases as congestion increases and such additional delay affects all queued packets. Consequently, real-time applications or, in general, applications with strict delay and jitter constraints, can suffer and experience bad performance. Hence a method to enhance the QoS of the system is required.

A genetic algorithm was hence written coupled with a fuzzy based priority scheduler to improve the QoS of an example wireless ad-hoc network. Appropriate simulations were performed on the example network using the simulation tool OPNET with respect to end to end delay and packet delivery ratio. The system responses were observed when AODV and DSR routing protocols were used on the system. The results were plotted and various graphs obtained.

This template will assist you in formatting your paper. Please, copy it on your computer and insert the text keeping the format indicated. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document. There is a limitation of 12 pages. [Arial, 10-point,normal, upper and lower case and alignment left]

Keywords - Ad-hoc, genetic algorithm, fuzzy based priority scheduler, Opnet, AODV, DSR, QoS.

1 INTRODUCTION

In the field of embedded systems, which encompasses a vast area of electronics, communication and software, wireless communication and networks occupy a major sector. An ad-hoc network is a local area network (LAN) that is built spontaneously as devices connect. In Latin, ad hoc literally means "for this," meaning "for this special purpose" and also, by extension, improvised or impromptu.

Ad hoc networks are of a decentralized nature. There is no static infrastructure for the network, such as a server or a base station [2]. Hence, in the absence of a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. They can be quickly deployed with minimal configuration. These properties make them ideal for use in case of emergencies and natural disasters. Since links can be connected or disconnected at any time, a functioning network must be able to cope with this dynamic restructuring, preferably in a way that is timely, efficient, reliable, robust and scalable. The network must allow any two nodes to communicate, by relaying the information via other nodes. A “path” is a series of links that connects two nodes. Various routing methods use one or two paths between any two nodes; flooding methods use all or most of the available paths [5]. Two routing algorithms and their effects on the proposed fuzzy based ad-hoc network are compared in this paper. The two protocols are DSR and AODV.

The Dynamic Source Routing protocol (DSR) is designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes and allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration.

This protocol, which operates entirely on-demand, is composed of the two main mechanisms of "Route Discovery" and "Route Maintenance", which work together to allow nodes to discover and maintain routes to arbitrary destinations in the ad hoc network.

The protocol allows multiple routes to any destination and allows each sender to select and control the routes used in routing its packets.

Consider a source node that has data packets to be sent to that destination but does not have a route to the destination. It initiates a RouteRequest packet that floods the network. Each node, upon receiving a RouteRequest packet, rebroadcasts the packet to its neighbors, if it has not forwarded it already, provided that the node is not the destination node and that the packet’s time to live (TTL) counter has not been exceeded. Each RouteRequest carries a sequence number generated by the source node and the path it has traversed. A node, upon receiving a RouteRequest packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RouteRequest. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same RouteRequest by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase. A destination node, after receiving the first RouteRequest packet, replies to the source node through the reverse path the RouteRequest packet had traversed. The DSR protocol , designed mainly for mobile ad hoc networks of up to about two hundred nodes, works well with even very high rates of mobility.

Ad hoc on demand Distance Vector or AODV is a routing protocol used specially for MANET or other wireless ad hoc networks. Consider a network where one particular node needs to transfer a message to some other individual node. The needy node broadcasts its need to communicate with the other node. As this message travels the network before forwarding it, each node notes down the particulars of which node it heard the message from. Thus various ‘paths’ are setup throughout the network..

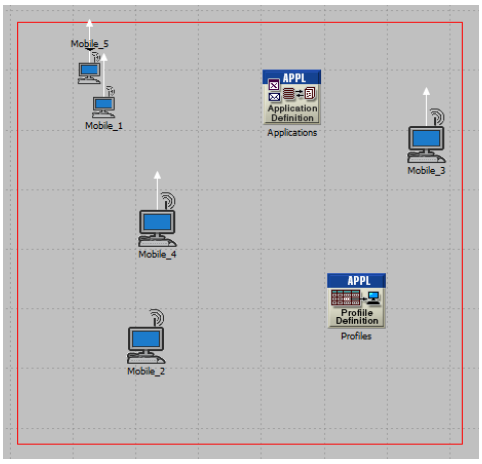


Fig.1. Network Layout

Once a node is reached that knows the location of the destination node it sends back a message by tracing the ‘path’ backwards till it reaches the first node. The first node then identifies which is the shortest multi-hop route to the destination node. The AODV Routing protocol derives its name because it uses an on-demand approach, that is, establishing a route only when it is required by a source node for transmitting data packets. When a link fails, a routing error is passed back to a transmitting node, and the process repeats. Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number. Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. If a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request.

The major difference therefore, between AODV and Dynamic Source Routing (DSR) stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission.

2 EXISTING SYSTEM

In the proposed networks, an example of which is shown in Figure1, the mobility of nodes and the error prone nature of the wireless medium pose many challenges like frequent route changes and packet loss. Such problems increase packet delay and decrease throughput. Also the absence of a base station and forwarding of packets across multiple broadcast regions makes it difficult to satisfy a flow's end to end QoS target [2].

Packet scheduling is the mechanism that selects a packet for transmission from the packets waiting in the transmission queue. It decides which packet from which queue and station are scheduled for transmission in a certain period of time. Packet scheduling controls bandwidth allocation to stations, classes, and applications.The existing system uses the FIFO method for scheduling of packets. The FIFO discipline is the basic first-in, first-out queuing method. It is very simple: the first packet in the queue is the first packet that is served. In FIFO, all packets are treated in the same way: they are placed in a single queue and are served in the same order they were placed. When the queue becomes full, congestion occurs and the new incoming packets are dropped. Since all packets are buffered in the same queue, FIFO does not allow handling packets of different classes in different ways. Therefore, it is not possible to provide different Classes of Service. The queuing delay increases as congestion increases and such additional delay affects all queued packets. Consequently, real-time applications or, in general, applications with strict delay and jitter constraints, can suffer and experience bad performance. Hence a method to enhance the QoS of the system is required. To improve the performance and maintain the QoS a scheduler can be used. C. Gomathy et al. [3] has been designing a fuzzy based priority scheduler to determine the priority of the packets. Kumar et al. [4] defined how to improve the end to end QoS target in MANET. Mary Bader et al. [1] has focused primarily on routing protocols.

3 PROPOSED SYSTEM

The proposed system uses a fuzzy based priority scheduler to set the priorities of the packets based on channel capacity and data rate. This fuzzy based scheduler enhances the QoS parameters and operates on a rule base given in [2].

3.1 A. Fuzzy Logic

Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic theory, where binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. The reasoning in fuzzy logic is similar to human reasoning. It allows for approximate values and inferences as well as incomplete or ambiguous data (fuzzy data) as opposed to only relying on crisp data (binary yes/no choices). Fuzzy logic is able to process incomplete data and provide approximate solutions to problems other methods find difficult to solve.

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership .FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

FL incorporates a simple, rule-based IF X AND Y THEN Z approach to solving a control problem rather than attempting to model a system mathematically. The FL model is empirically-based, relying on an operator's experience rather than their technical understanding of the system.

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them.

The general algorithm for a Fuzzy Logic application is:

1) Define the control objectives and criteria.
2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).
3) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number of fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for step inputs.
4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.
5) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.
6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

Given below is the block diagram for the fuzzy scheduler that is implemented. The scheduler is fed crisp inputs. These are then fuzzified. The role of the fuzzifier is to map the crisp input data values to fuzzy sets defined by their membership functions. The defuzzifier maps the output fuzzy sets to a crisp output value.

The fuzzy scheduler used assigns a priority index to each packet. The fuzzy scheduler uses 2 input variables and 1 output variable. The 2 input variables to be fuzzified are the data rate and channel capacity of the nodes to which the packet is associated with.

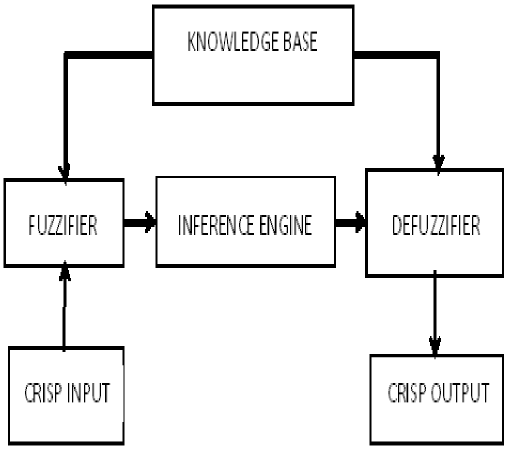


Fig. 2. Fuzzy Logic Block Diagram

The rule base used is given below,

Table 1. Existing rule base

DR \ CC	L	M	H
L	L	A	M
M	VL	M	H
H	A	A	VH

DR – Data Rate

CC – Channel Capacity

These rules are arrived at by a tedious method of trial and error. The proposed system makes use of a genetic algorithm to arrive at an optimal set of rules.

3.2 B. Genetic Algorithm

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. It involves processes such as initialization, selection, crossover, mutation and termination. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Here the initial rule base is encoded and a genetic algorithm is used to optimize the set of rules. The general form of a genetic algorithm is as follows:

1) Randomly generate an initial population M(0)
2) Compute and save the fitness u(m) for each individual m in the current population M(t)
3) Define selection probabilities p(m) for each individual m in M(t) so that p(m) is proportional to u(m)
4) Generate M(t+1) by probabilistically selecting individuals from M(t) to produce offspring via genetic operators
5) Repeat step 2 until satisfying solution is obtained.

By the use of a genetic algorithm the chromosome under goes first, initialization or a method of encoding. Then a set of chromosomes is selected. Selection used in our method was based on a fitness function. The selected chromosomes undergo crossover. Cross over is a process of taking more than one parent solutions and producing a child solution from them. In this case single point crossover was employed. The next step is mutation. Mutation is used to maintain diversity in each generation of chromosomes. A previously selected end condition is evaluated and depending on its fulfillment the process is looped. Using this method an optimized set of rules was obtained

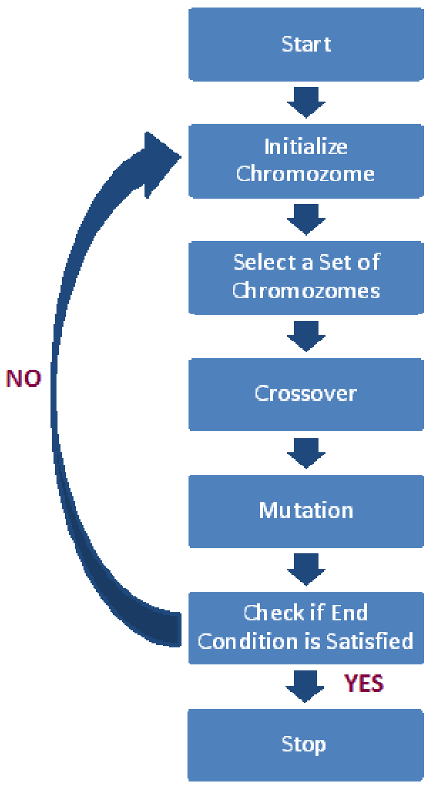


Fig. 3. Genetic Algorithm Flowchart

Table 2. Proposed rule base

DR / CC	Low	Medium	High

Low	Medium	Very High	Very High
Medium	Low	Average	Low
High	Very Low	Very Low	High

4 PERFORMANCE EVALUATION

4.1 A. Opnet

OPNET is a high level event based network level simulation tool. This simulation tool operates at "packet-level". OPNET can be used as a research tool or as a network design/analysis tool.NS-2 is widely used for network research in academia. NS-2 is also free. However, NS-2 is more difficult to learn and lacks a user interface. It requires the users to learn and use non-standard scripting interfaces such as tcl.Opnet is easier to use due to its Graphical User Interface (GUI) form.

B. Results and Simulations

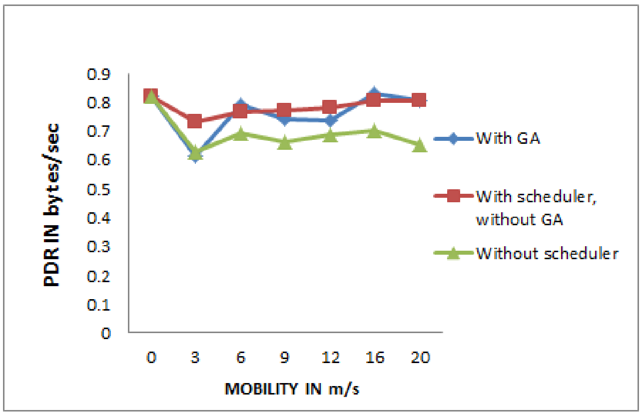
The code for the fuzzy logic priority scheduler with the original set of rules was written and coded in C++. The scheduler assigned priority to packets based on data rate and channel capacity. First a priority index is calculated. This is such that packets with the lowest priority index are assigned the highest priority. Each node is considered to have 3 queues . Packets are placed in theses queues based on their priority index. Once coded the code was interfaced with Opnet's Proto C language. Using these settings the Opnet simulations were run with and without the scheduler. The genetic algorithm was also coded in C++. The written code was compiled and checked. Then this code was entered in Opnet's Proto C language and the simulation was run. The results of these simulations were compared and plotted to obtain the following graphs. The parameters considered are throughput, packet delivery ratio, mobility and end to end delay. It is seen that there is more improvement in the system when the GA based rules are used in the fuzzy scheduler

Table 3. Simulation Parameters

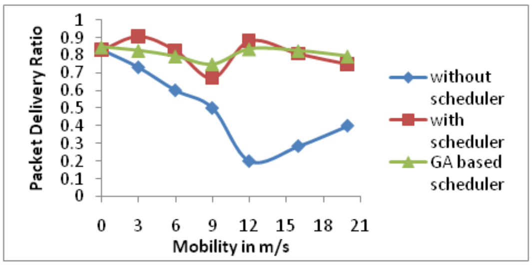
Number of nodes	5
Area	250*250 m
Simulation time	600 sec
Node placement	Random
Mobility model	Random waypoint
Speed	0-20 m/s
Propagation Model	Free space
Channel Bandwidth	5.5 Mbps

Traffic type	CBR
Data payload	1024 bytes/packet
Routing protocol	AODV / DSR
MAC protocol	IEEE 802.11

Throughput or network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps).The system throughput or aggregate throughput is the sum of the data rates that are delivered to all terminals in a network. When examining throughput, the term 'Maximum Throughput' is frequently used where end-user maximum throughput tests are discussed in detail. This number is closely related to the channel capacity. In this case, throughput can be defined as the overall percentage of data received over the data sent in a closed system for a specific period of time. This is a fundamental measure of the performance of a network, and therefore, an important factor to consider. The delay is the overall time taken from the moment the data is transmitted to the moment it is received by the designated destination. Delay affects applications in many ways. Applications that are delay-sensitive such as video streaming and voice cannot function properly when there is a long delay. Data Packet Delivery Ratio can be calculated as the ratio between the number of data packets that are sent by the source and the number of data packets that are received by the sink.

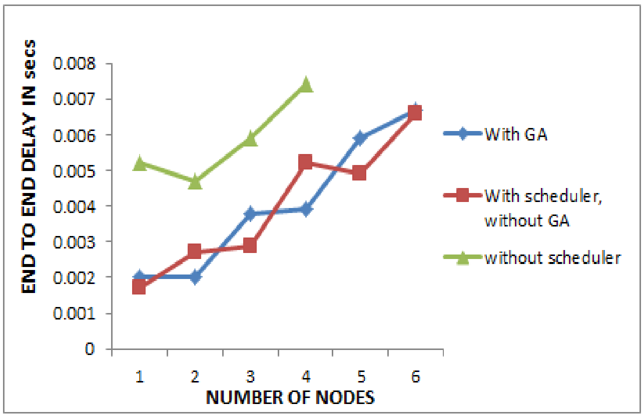


Graph 1. Packet Delivery Ratio Vs Mobility (with DSR)

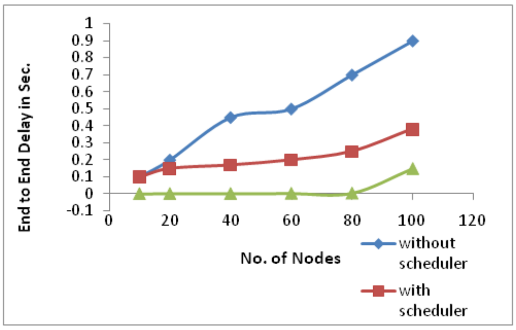


Graph 2. Packet Delivery Ratio Vs Mobility (with AODV)

Packet Delivery Ratio is the ratio of the number of data packets actually delivered to the destination to the number of data packets supposed to be received. This number presents effectiveness of the protocol.

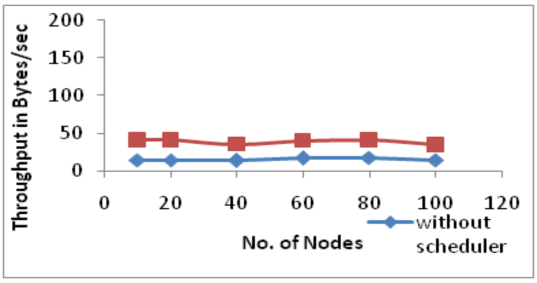


Graph 3. End to End Delay Vs No. of Nodes (with DSR)

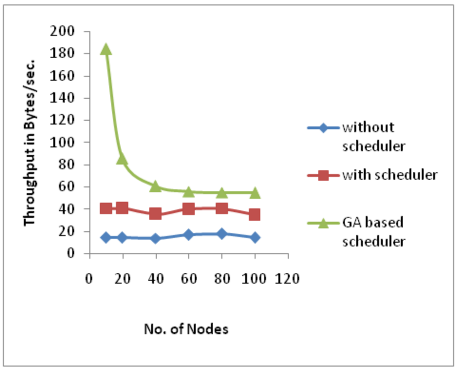


Graph 4. End to End Delay Vs No. of Nodes (with AODV)

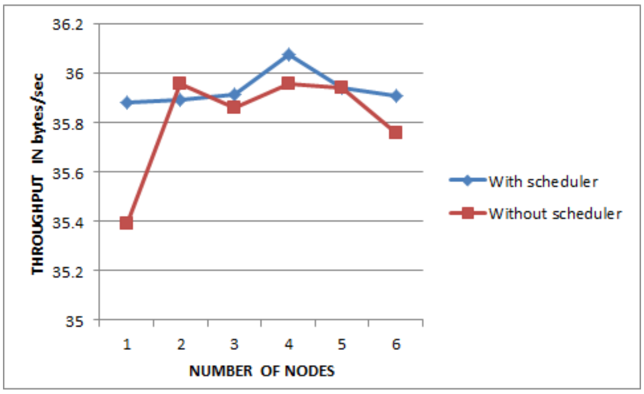
End to end delay indicates how long it took for a packet to travel from the source to the application layer of the destination.



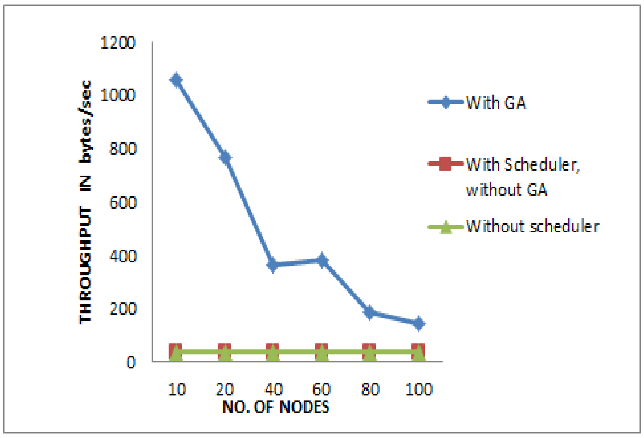
Graph 5. Throughput Vs No. of Nodes (with AODV, without GA)



Graph 6. Throughput Vs No. of Nodes (with AODV)



Graph 7.Throughput Vs Number of Nodes (with DSR, without GA)



Graph 8.Throughput Vs Number of Nodes (with DSR)

The graphical representations in graphs.7&8 show improvement in the throughput as the number of nodes increase in case of AODV. However, it is the opposite in case of DSR where the trend shows a drastic drop in throughout with increase in number of nodes. In the case of AODV, the graph shows an improvement of about 15 to 20 Bytes/sec when plotted against a steady increase in the number of nodes from 20 to 100. In the case of DSR, the graph shows drop from around 1050 bytes/sec to 150 bytes per second when plotted against a steady increase of nodes from 20 to 100. However one notices that the original starting throughput of DSR routing protocol is very high and is in itself a benefit when considering networks with small and limited number of nodes.

The graphs were plotted based on values obtained from numerous OPNET simulations.

5 CONCLUSION

In this research it is attempted to provide a Genetic Algorithm based fuzzy scheduler for wireless ad-hoc networks. The scheduler has been tested for AODV and DSR routing protocols. The QoS is compared in AODV and DSR with reference to PDR verses mobility and end to end delay verses number of nodes and throughput verses number of nodes. From the simulated results it is found that for AODV GA derived fuzzy based scheduler is highly suited for improving most of the QoS parameters. However, for DSR protocol GA derived fuzzy scheduler has almost the same advantage as that of the scheduler without GA. In the aspect of throughput verses number of nodes, the throughput improves tremendously 300 times 10 nodes and as the number of nodes increases the throughput level falls to four times for 100 nodes. Hence we can conclude that the GA based fuzzy scheduler is highly suited for AODV as compared to DSR protocol.

References

[1] Chun, B. G. and Baker, M. 2002. *Evaluation of packet scheduling algorithms in mobile ad hoc networks*. ACM SIGMOBILE Mobile Computing and Communications Review. 6 (3), (2002), 36-49.

[2] Faruque, M. N., Ahmad, S. N. and Kumar, M. 2011. *Performance of QoSin Wireless Ad hoc Network for AODV Protocol using Fuzzy Based Techniques*. International Journal of Electronics and Communication Technology. 2 (2) (2011).

[3] Gomathy, C. and Shanmugavel, S. 2005. *Supporting QoS in MANET by Fuzzy Priority Scheduler and Performance Analysis with Multicast Routing Protocols*. EURASIP Journal on Wireless Communication and Networking. 3 (2005), 426-436.

[4] Manoj, K., Sharma, S.C., Vijay, S. and Dixit, A. 2008. *Performances Analysis of Wireless Ad-hoc Network Using OPNET Simulator.* International Conference on Intelligent Systems and Networks. (ISN-08) ISTK Haryana. (2008), 267-270.

[5] Wu S.L. and Tseng Y.C. 2007. *Wireless Ad Hoc Networking*, Auerbach Publications. (2007). ISBN 978-0-8493-9254-2