International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-3, January 2020

A Packet Size Based Intelligent Model for AODV in MANET

Pushpender Sarao

Abstract-Mobile ad-hoc networks are the networks that are established easily in a short span time period. But, at the same time, some challenges are also there for establishment and performance perspectives. Higher mobility of nodes creates a problem in secure and accurate data transmission. Packet size decision is to be taken for optimizing the data packets transmission from source to destination. In this paper, to decide the packet size, an ANFIS based intelligent model has been proposed. Model has been implemented at MAT Lab with generalized bell-shaped membership functions for four inputs. For this, throughput, normalized routing load, data packets received, average end to end delay metrics have been considered. Designed model has been verified with the help of network simulator-2. Proposed model will be applied for AODV routing protocol in mobile ad-hoc networks. Proposed model will enhance the performance of AODV routing protocol.

Keywords-AODV, throughput, ANFIS, average end to end delay, data packets received, training data set

I. INTRODUCTION

MANET (Mobile Ad-hoc Network): mobile ad-hoc networks are dynamic in nature, having neighbor discovery ability, flexible architecture (i.e. not fixed), peer to peer connectivity, no central communication control system, limited connectivity range, and no access points, diversity characteristic[15].

Disadvantage- some drawbacks of the MANET are: less secure, dynamic topology, poor performance in wired protocols.

AODV(Ad hoc on demand distance vector)- AODV[14] is on demand routing protocol in MANET which have very good performance when speed of nodes are high as compare to DSR and DSDV routing protocols. In the traditional AODV protocol only one metric 'hop count' is considered during selection of next hop. In route discovery phase-RREQ and RREP messages are used while RERR message is use for route maintenance phase. Reverse route details are stored in routing tables. Sequence numbers are utilized for timelines of the data packets. Routes are established only when required. AODV have a facility for loop free routes. Due to regular beaconing much bandwidth is consumed in AODV routing protocol. If network size is increased, as a result performance of AODV goes down.

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author
Pushpender Sarao*,Professor, Department of CSE, Institute of Technology and Management, Hyderabad (Telangana) India. Email: drpushpendersarao@gmail.com

In MANET, so many routing protocols are used for data transmission. To ensure better performance, network metrics play a great role. Packet size is one of them which directly effects on the performance of the routing protocols during routing process. In this domain, researchers proposed a lot of work in literature to enhance the performance of routing protocols [11][12][13]. Some analyzed the effects of packet size and delay on the performance of AODV routing protocol while other proposed some models and verified on network simulator environment. ANFIS and some other similar models like AROMA have been analyzed for their accuracy and desired outputs.

Vivek sharma et al. [1] proposed an ANFIS based AODV routing protocol for selecting an optimum route. Three input parameters (energy, hop count, and delay) have applied to produce one output as route selection. Three linguistic variables for input membership functions (triangular membership function) and five for output membership function (triangular membership function) were considered. Total 27 fuzzy rules were designed at ANFIS system for input and output parameters [7]. ANFIS model was implanted in MATLab platform. The proposed AAODV routing protocol was simulated on NS-2.35 network simulator for 250 seconds. Also an evaluation and comparison work was carried out with performance metrics (packet delivery ratio, average routing overhead load, average network throughput, and pause time.

Zahian Ismail et al.[2] analyzed the packet size effects on the performance of AODV in open as well as closed mobile ad-hoc networks. In this research work, implementation is carried out in homogeneous and heterogeneous MANETs. For simulation of AODV, OMNET++ was used for 3000s for different packet sizes (100-1000 bytes). Performance metrics like packet delivery and throughput were considered for evaluating the proposed work. Performance evaluation was carried out in three different communication network scenarios. It was concluded that AODV performs better in homogeneous mobile ad-hoc networks.

Deepika et al. discussed about the packet size and node density effects on the performance of AODV and DSR routing protocols in mobile ad-hoc networks [3]. Simulation work was carried out using the network simulator NS-2.32 for 500s. Performance metrics like end-to-end delay, packet delivery ratio, throughput, delay, and routing load were considered for both routing protocols (AODV and DSR). It was claimed that performance of AODV is excellent for overall experimental work as compared to DSR routing protocol.

Sesmita Acharya et al. suggested an ANFIS based NFOM (Neuro-Fuzzy based Optimized Model) in wireless sensor network [4]. A data aggregation scheme has been applied on ANFIS estimator.

Four input parameters (cluster threshold, grid size, and total number of clusters) have been used for getting one single output as 'node-status'. Trapezoidal membership functions have been used with three linguistic variables: low, medium, and high. The simulation work was carried out on MATLab 7.5 environment. Five different cases were considered to compare the proposed model with existing fault detection algorithms varying the number of nodes. The comparative analysis work was carried out with following performance metrics: first node dies (FND), network life time, energy cost, loss probability, fault detection accuracy (FDA), false alarm rate(FAR).

It was claimed that proposed scheme is best in terms of cost, accuracy, and fault tolerance as compared to existing algorithms like DFD, LEDFD, MV, and FTFK.

In [5], Wafa Benatou et al. proposed an intelligent handover scheme to forecast the suitable destination station. ANFIS system was used by considering three input parameters. Gaussian membership functions were used for input parameters: bandwidth, signal noise ratio and energy consumption. Total ten epochs were applied to train the training data sets. Total twenty seven fuzzy rules were designed using the three input parameters with 75 nodes. It was observed that proposed scheme generates better results in respect of handovers and energy consumption.

Rajnish et al.[6] compared ANFIS and ARIMA models in respect of network traffic timings at various network scenarios. Both models were evaluated for computer network traffic modelling. Three different network scenarios were considered in ANFIS. Four input parameters with two linguistic variables (for each membership function), 16 rules for first case, 55 nodes were considered to generate a ANFIS system. In second case, total 64 fuzzy rules with 161 nodes were applied while simulations work at ANFIS in MAT Lab environment. Both models were implemented in MAT Lab platform. It was concluded that ANFIS model performs well as compare to ARIMA (auto regressive integrated moving average) model during the evaluation of network traffic timings modelling.

Suman et al. proposed an algorithm PAIRS (Periodic Adaptive and Intelligent Route Selection) for selecting an appropriate route [8]. An ANFIS model was designed to implement the PAIRS algorithm. Three input parameters used in the proposed ANFIS based PAIRS algorithm: LC (Processing Capacity), E (Energy), PC (Processing Capacity). Total 27 fuzzy rules were designed for input and output parameters. It was concluded that proposed algorithm works well in route selection in respect of energy and cost.

In AODV routing protocol, packet size directly effects on the performance of the protocol. Here, we have to decide which size of the packet is most appropriate for optimizing the route and ensuring the higher throughput. In this work, we have proposed an intelligent ANFIS based model for AODV routing protocol, which is helpful to decide the best suitable packet size. Four input metrics (throughput, normalized routing load, data packets received, and average end to end delay) are taken to generate the one output parameter named as 'packet size'. With help of this model, as per our requirements (throughput, data packets received), we can select a right option for selecting a packet size.

This paper is partitioned into five sections. Section II described about the proposed intelligent ANFIS model for AODV routing protocol with detailed specifications. Simulation set up using NS-2.35 has been explained in section III. Section IV introduces about the results and discussion. This paper is concluded in section V.

II. PROPOSED PACKET SIZE BASED INTELLIGENT MODEL

During the packet size discovery phase, ANFIS possesses with four inputs throughput, normalized routing load, data packet received, and average end to end delay (see figure 1). The fuzzifier rule base of ANFIS contains 81 rules (see table 2) and its defuzzifier has one output. To achieve highly stable and balanced packet size, data training sets are required. The number of training data comprising of 81 data pairs while number of checking data comprising of 28 data pairs(see table 1). The first set is used to train the network. Data sets are derived from fuzzy 'if then rules'. The training data set is fed as an input to ANFIS tool under MATLAB environment. As a result, the optimal fuzzy membership function is obtained as an output which is used in NS2 environment to take the packet size decision. The training is done for error tolerance 1e-5, epochs 90 and hybrid learning method.

Two types of data sets are used for anfis model(figure 3): training data set and checking data set. First, for loading the data, select the training and checking option in 'load data' part of anfis designer. We loaded data from file/workspace. Next step, generate the fis. For generating the fis, grid partition option has been selected. To assign the different number of MFs (see table 4, 5, 6, figure 2) to each input, we have selected three linguistic variables for each input membership function (i. e. low, medium, high).

Generalized bell-shaped membership function has been used for all input parameters. For output membership function, we have selected as 'constant' in MF type space.

Now, in next step, for train the fis, hybrid method is used. Total 100 epochs are used for training the data sets.

Now, to execute the training for data sets, 100 epochs were decided to set the minimize values for error. After training the data set, we will get the training error values.

Note: For validating the model, checking and testing part have to be performed. ANFIS with 4 inputs and one output membership functions were used with anfis architecture (3 3 3 3). Sugeno model is used in this paper.

We plotted the structure overview of overall FIS model by using the structure option. In ANFIS model structure; there are four input parameters ((each having three linguistic variables). There are five layers in ANFIS structure. First and last layer represent the inputs and output respectively. Total eighty one rules were designed to generate the output (i.e. packet size).

Fuzzy rules are executed at layer 4 of the anfis model structure. Rules are normalized at layer 3 while layer 2 is responsible for generating the membership functions.

Now, in next step, for testing the ANFIS model (to plot the training data), select the 'training data' option and test the training data. After testing the training data, a plot will be generated in MAT Lab workspace, which shows index vs output (for training data with output).

For Checking data, we will select in respect of 'check data' option and test it, a plot for checking data will be generated and we will get plot for index vs output.

At this stage of implementation, training and checking part have been completed and model is validated successfully (see figure 4, figure 5). Now, we can see the output (packet size) with respect to input values (throughput, normalized routing load, data packets received, average end-to-end delay). Results for input/output values can be visualized using 3-D surface plot and rule viewer (figure 6, figure 7). In rule viewer, by sliding the vertical bar, we can set the desired values for inputs (for example desired throughput) and can get the output (i.e. packet size).

3-D surface plots (see figure 7) illustrates the desired throughput value and data packets received value with respect the packet size value.

In rule viewer(figure 6), if throughput (2948 kbps), normalized routing load (0.03776), data packets received (2646), and average end-to-end delay (380.3 ms) values are required, we have to set the packet size value as 2200 bytes.

Table 1: ANFIS specification

ANFIS Info:
Number of nodes: 193
No. of linear parameters: 81
No. of nonlinear parameters: 36
Total no. of parameters: 117
No. of training data pairs: 81
No. of checking data pairs: 28
No. of fuzzy rules: 81

To generate the FIS, we have selected the grid partition option.

Packet Size Metrics:

Adaptive Neural Fuzzy Inference System (ANFIS)

Layer1: Generate the Membership Grade

Layer2: Generate the Firing Rule.

Layer 4: Calculate rule Outputs based on the Consequent Parameters

Layer 5: is responsible for single output.

Table 2: Rule base for ANFIS AODV

>> showrule(fis)
ans =

1. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf1) (1)
2. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf2) (1)
3. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf3) (1)
4. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf4) (1)
5. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf5) (1)
6. If (Throughput is Low) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf6) (1)
7. If (Throughput is Low) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf7) (1)
8. If (Throughput is Low) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf8) (1)
9. If (Throughput is Low) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf9) (1)
10. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf10) (1)
11. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf11) (1)
12. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf12) (1)
13. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf13) (1)
14. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf14) (1)
15. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf15) (1)
16. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf16) (1)
17. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf17) (1)
18. If (Throughput is Low) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf18) (1)
19. If (Throughput is Low) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf19) (1)
20. If (Throughput is Low) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf20) (1)
21. If (Throughput is Low) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf21) (1)
22. If (Throughput is Low) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf22) (1)
23. If (Throughput is Low) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf23) (1)
24. If (Throughput is Low) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf24) (1)
25. If (Throughput is Low) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf25) (1)
26. If (Throughput is Low) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf26) (1)
27. If (Throughput is Low) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf27) (1)
28. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf28) (1)
29. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf29) (1)
30. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf30) (1)
31. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf31) (1)
32. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf32) (1)
33. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf33) (1)
34. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf34) (1)
35. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf35) (1)
36. If (Throughput is Medium) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf36) (1)
37. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf37) (1)
38. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf38) (1)
39. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf39) (1)
40. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf40) (1)
41. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf41) (1)
42. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf42) (1)
43. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf43) (1)
44. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf44) (1)
45. If (Throughput is Medium) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf45) (1)
46. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf46) (1)
47. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf47) (1)

48. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf48) (1)
49. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf49) (1)
50. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf50) (1)
51. If (Throughput is Medium) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf51) (1)
52. If (Throughput is Medium) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf52) (1)
53. If (Throughput is Medium) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf53) (1)
54. If (Throughput is Medium) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf54) (1)
55. If (Throughput is High) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf55) (1)
56. If (Throughput is High) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf56) (1)
57. If (Throughput is High) and (NRL is Low) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf57) (1)
58. If (Throughput is High) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf58) (1)
59. If (Throughput is High) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf59) (1)
60. If (Throughput is High) and (NRL is Low) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf60) (1)
61. If (Throughput is High) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf61) (1)
62. If (Throughput is High) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf62) (1)
63. If (Throughput is High) and (NRL is Low) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf63) (1)
64. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf64) (1)
65. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf65) (1)
66. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf66) (1)
67. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf67) (1)
68. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf68) (1)
69. If (Throughput is High) and (NRL is Medium) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf69) (1)
70. If (Throughput is High) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf70) (1)
71. If (Throughput is High) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf71) (1)
72. If (Throughput is High) and (NRL is Medium) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf72) (1)
73. If (Throughput is High) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Low) then (Packet_Size is out1mf73) (1)
74. If (Throughput is High) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf74) (1)
75. If (Throughput is High) and (NRL is High) and (Data_Packets is Low) and (A_E2E_Delay is High) then (Packet_Size is out1mf75) (1)
76. If (Throughput is High) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Low) then (Packet_Size is out1mf76) (1)
77. If (Throughput is High) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf77) (1)
78. If (Throughput is High) and (NRL is High) and (Data_Packets is Medium) and (A_E2E_Delay is High) then (Packet_Size is out1mf78) (1)
79. If (Throughput is High) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Low) then (Packet_Size is out1mf79) (1)
80. If (Throughput is High) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is Medium) then (Packet_Size is out1mf80) (1)
81. If (Throughput is High) and (NRL is High) and (Data_Packets is High) and (A_E2E_Delay is High) then (Packet_Size is out1mf81) (1)

Input Variables: Four Input Variables (Throughput, NRL, Data Packets Received, Average E2E Delay)
Output Variable: Packet Size

Table 3: Membership function 'Throughput' specifications

Membership Function	Throughput(kbps)		
RANGE	(304.8, 3315)		
Linguistic variables:	LOW	MEDIUM	HIGH
LOW	(752.8, 2, 3048)		
MEDIUM	(752.6, 2, 1810)		
HIGH	(752.6, 2, 3315)		

Table 4: Membership function 'NRL' specifications

Membership Function	NRL		
RANGE	(0.024, 0.166)		
Linguistic variables:	LOW	MEDIUM	HIGH
LOW	(.0475,2, .03113)		
MEDIUM	(.02044, 2, .006)		
HIGH	(.03237, 2, 0.1596)		

Table 5: Membership function 'Data Packet Received' specifications

Membership Function	DATA PACKET RECEIVED		
RANGE	(662,4629)		
Linguistic variables:	LOW	MEDIUM	HIGH
LOW	(991.7, 2, 662)		
MEDIUM	(991.8, 2, 2645)		
HIGH	(991.7, 2, 4627)		

Table 6: Membership function 'Average E2E Delay' specifications

Membership Function	AVERAGE END TO END DELAY		
RANGE	(125.2, 635.4)		
Linguistic variables:	LOW	MEDIUM	HIGH
LOW	(127.6, 2, 125.2)		
MEDIUM	(127.6, 2, 380.3)		
HIGH	(127.6, 2, 635.4)		

Membership Function	Packet size
RANGE	(256, 4500)

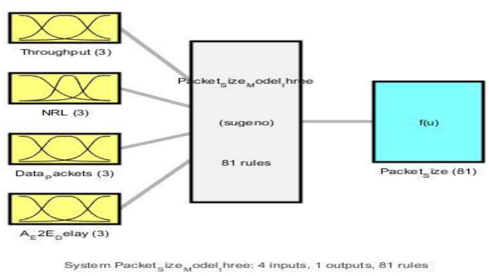

Figure 1: Sugeno system model with 4 inputs, one output, 81 rules

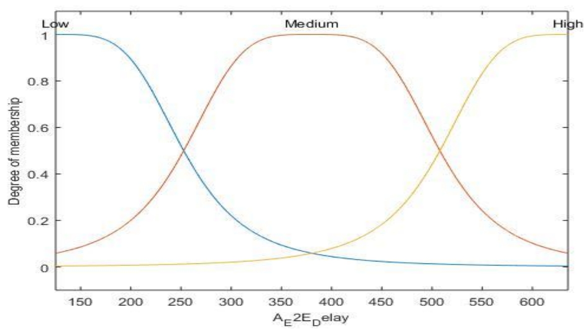

Figure 2(a): 'Average E2E Delay' Membership function

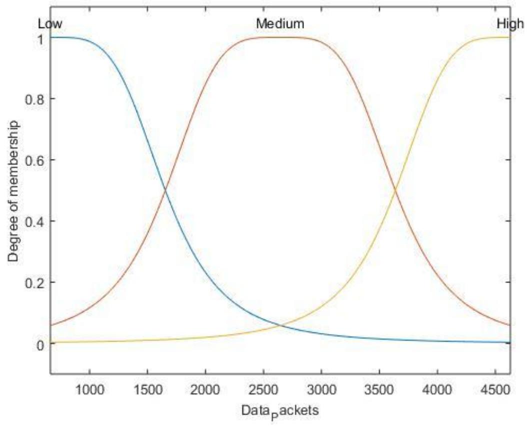

Figure 2(b): 'Data Packets Received' Membership function

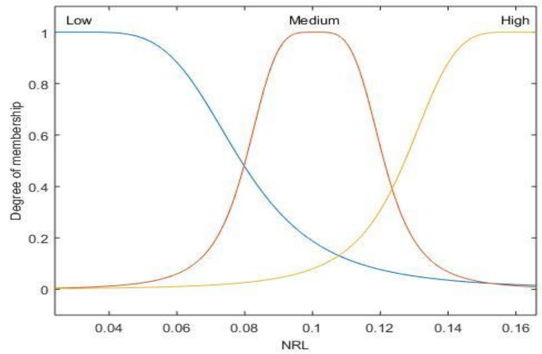

Figure 2(c): 'Normalized Routing Load' Membership function

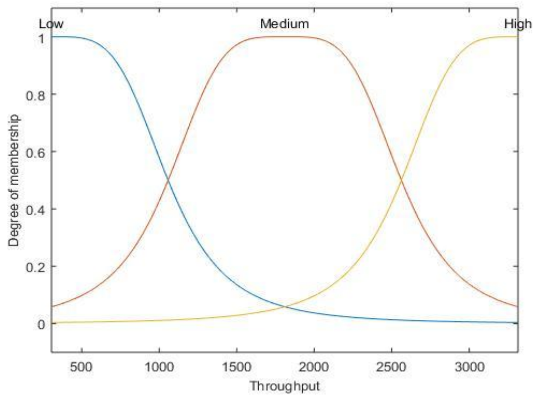

Figure 2(d): 'Throughput' Membership function

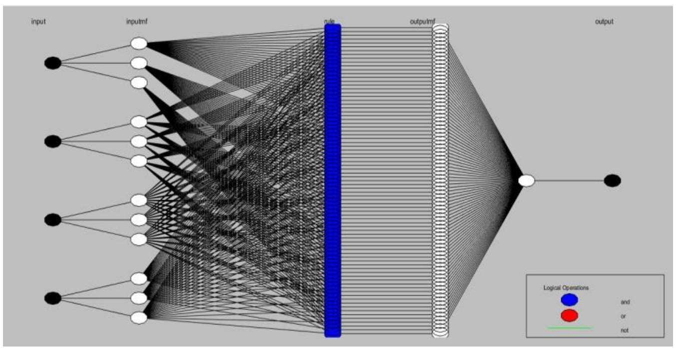

Figure 3: Five layered ANFIS Architecture with four inputs and one output

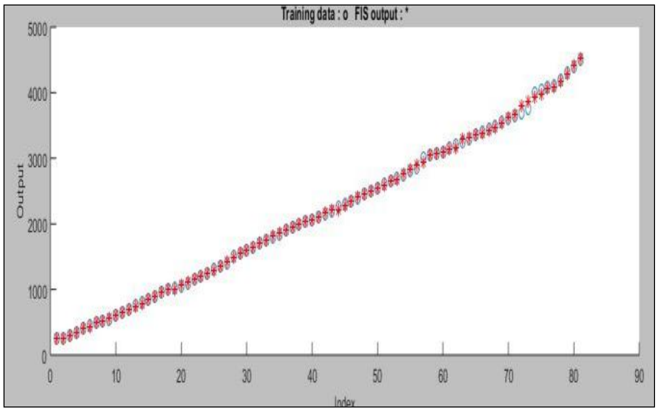

Figure 4: Training data and FIS output in ANFIS

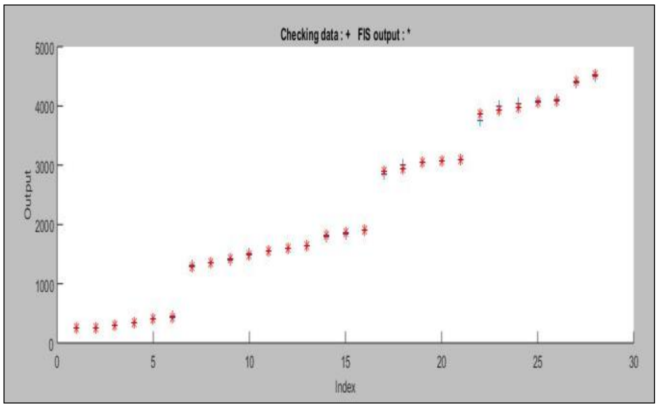

Figure 5: Checking data and FIS output on ANFIS system

Total fuzzy Rules=81
NFIS NAME=Packt_Size_Model_three
NFIS Type=Sugeno
Membership type: gbell membership function---generalized bell-shaped membership function

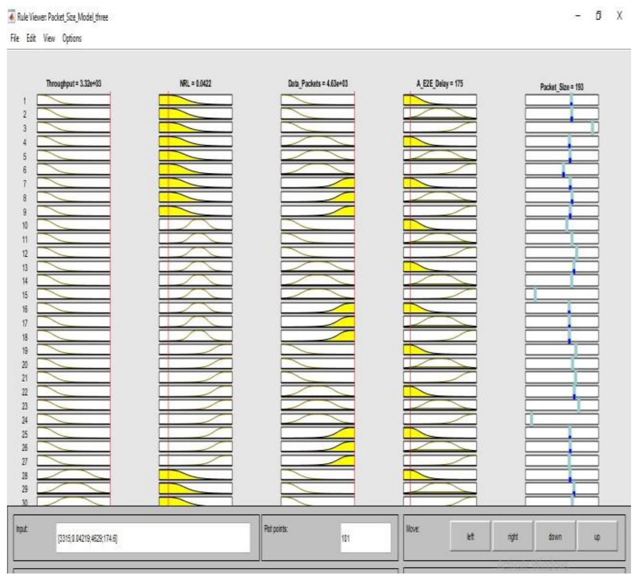

Figure 6: Rule viewer for output variable 'Packet Size' w.r.t. Input values

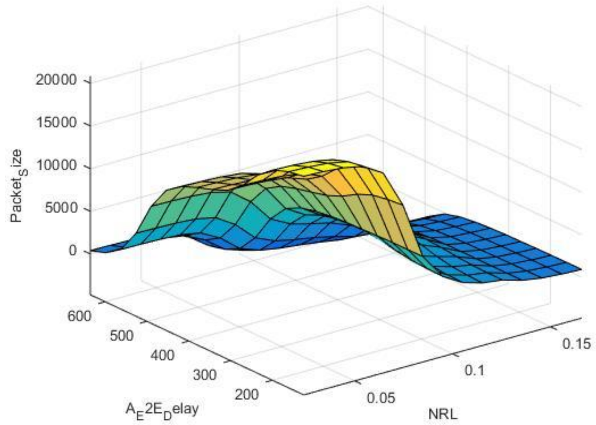

Figure 7(a): 3-D Surface plot packet size w.r.t. A E2E Delay and NRL

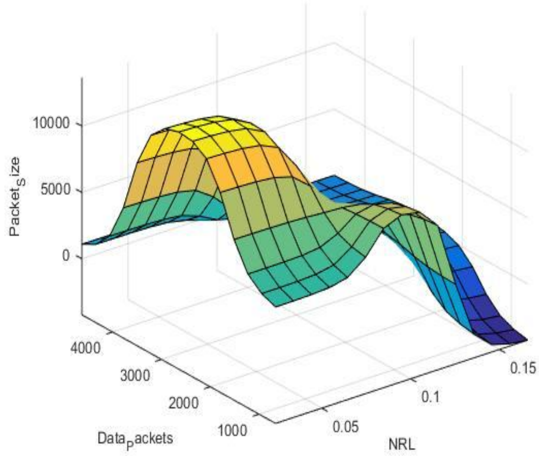

Figure 7(b): 3-D Surface plot packet size w.r.t. Data packet received and NRL

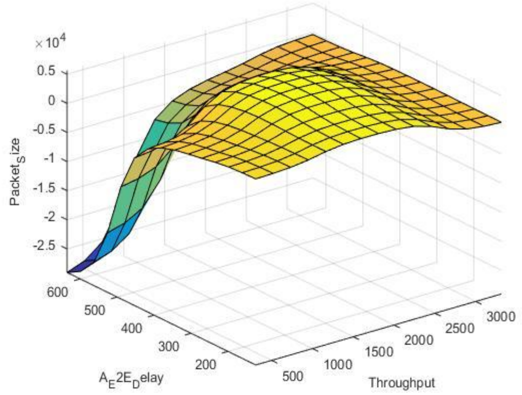

Figure 7(c): 3-D Surface plot packet size w.r.t. A E2E Delay and Throughput

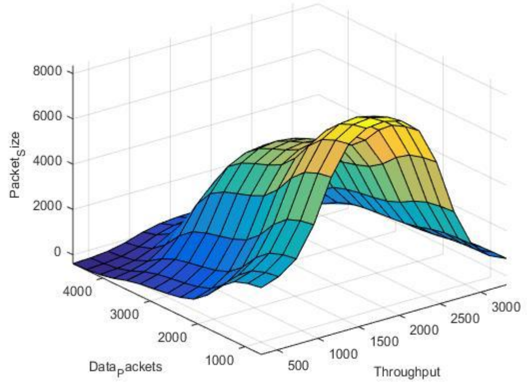

Figure 7(d): 3-D Surface plot packet size w.r.t. Data Packets received and Throughput

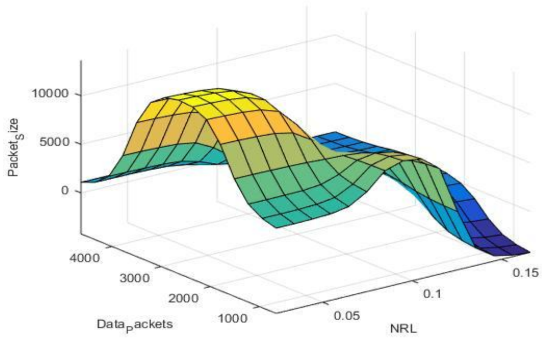

Figure 7(e): 3-D Surface plot packet size w.r.t. Data Packets and NRL

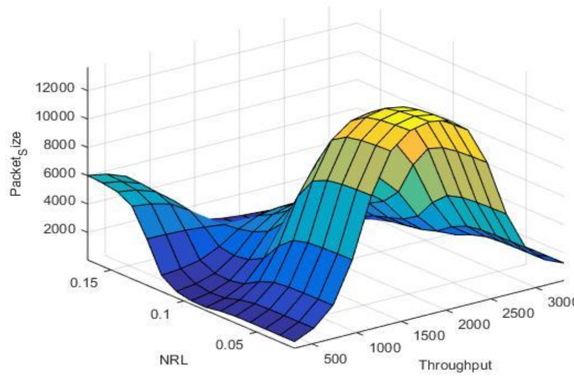

Figure 7(f): 3-D Surface plot packet size w.r.t. NRL and Throughput

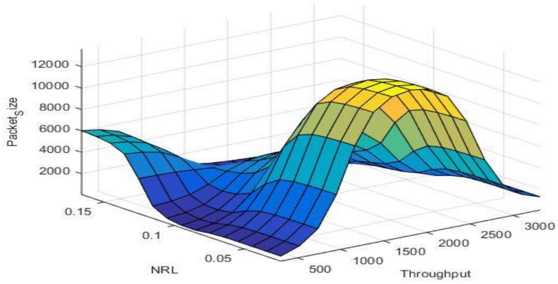

Figure 7(g): 3D Surface plot for output 'Packet Size' w.r.t Throughput and NRL

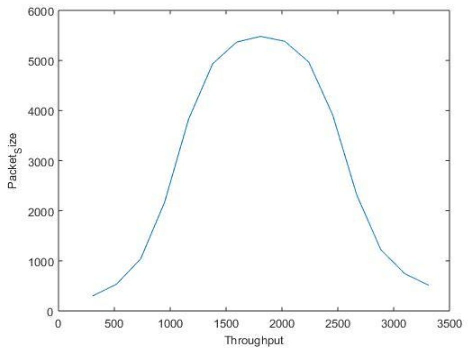

Figure 8(a): Throughput Vs Packet size

As depicted in figure 8(a), throughput is increased as the packet size is increased.

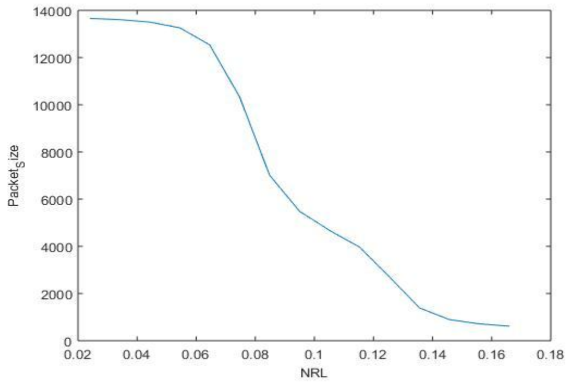

Figure 8(b): Normalized routing load Vs Packet Size

Figure 8(b) shows that normalized routing load is increased as the packet size is increased.

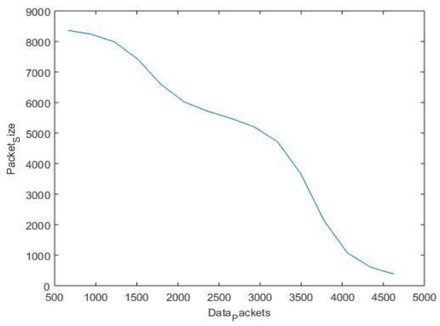

Figure 8(c): Data Packet received Vs Packet Size

As illustrated in figure 8(c), data packet received are less as the packet size is increased.

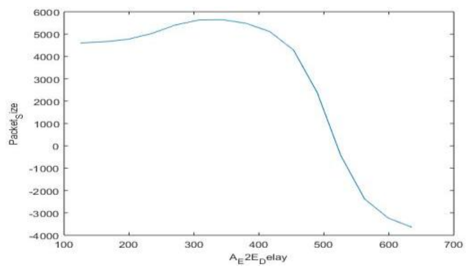

Figure 8(d): Average E2E Delay Vs Packet Size

Figure 8(a) shows that average end to end delay is higher at large size of packets. It means large size packets are responsible for higher end to end delay in AODV routing protocol in mobile ad-hoc networks.

A. ANFIS Results:

Table 7: output values with respect to input values in ANFIS rule viewer

Sr. No.	Throughput	NRL	Data Packets	Average E2E Delay	Packet Size
1.	305	.034	3460	193	517
2.	365	.024	4630	125	256
3.	399	0.053	2350	174	995
4.	450	0.082	1350	320	2060
5.	476	0.121	969	479	2990
6.	493	0.144	749	552	4040
7.	479	0.166	662	635	4520
8.	305	0.032	3400	190	516
9.	350	0.024	4500	125	256
10.	400	0.04	2000	165	1020
11.	500	0.12	950	450	2900
12.	600	0.166	700	635	4580
13.	3315	0.166	4629	635	41.7
14.	3315	0.042	4629	175	193
15.	3315	0.024	4629	175	195
16.	3315	0.024	4629	125	203

As illustrated in table 7, figure 6, figure 7, and in figure 8, from packet size 256-4040 bytes, throughput is increased from 305-493 kbps; except, for packet size 517 bytes, throughput is 305 kbps. Normalized routing load is increased at packet size 516 bytes to 4580 bytes. At packet size 4520-4580 bytes, normalized routing load is approximately same. Data packets received are less at higher values of packet size. As shown in table and figure, at packet size 516 bytes, total data packets received are 3400 while at packet size 4580 bytes, total data packet received are 700.

Average end to end delay is lowest at packet size 256 bytes. For packet size 4520-4580 bytes, average end to end delay is same (i.e. 635 ms). Overall, for all cases, average end to end delay is increased as packet size is increased. For packet size 193-195 bytes, average end to end delay is same (i. e. 175 ms). For packet size 203 bytes and 256 bytes, average end to end delay is same.

III. SIMULATION WORK

As we designed a neuro fuzzy model for deciding the packet size in respect of various performance metrics like throughput and end to end delay. Now, in this section we have to verify the proposed model at the platform of network simulator (NS-2.35). Network simulator is most likely platform which is used by almost researchers who are working in wireless network domain. NS-2[9] can be implanted on Linux as well as on windows platforms also. To evaluate and visualize the better results, NS-2 is an excellent simulator for beginners. To evaluate the model, we have considered some performance parameters: throughput, normalized routing load, data packets received, average end to end delay and packet size. Packet size is taken as: 256, 512, 1024, 2048, 3072, 4096, and 4500 bytes. Simulation work was carried out on the platform of NS-2.35 for 50 seconds with 5 nodes with 4 maximum connections only (see table 8). Pause time kept zero for all simulations. Traffic type was TCP while the protocol has been decided as AODV routing protocol. Script files like tcl, awk, and perl were written to implement the all simulation work.

Awk[10] and perl scripts were written for calculating the performance parameters (throughput, normalized routing load, data packets received, and average end to end delay). To create a node movement environment (traffic), we have applied setdest tool. Maximum speed for all moving nodes was decided as 40m/s. for all packet sizes, we calculated values for throughput (kbps), and normalized routing load (NRL), and data packets received, and average end to end delay (ms). All simulation was carried out on Ubuntu 16.5. For plot drawing, xgraph and origin tools were used. All results we got have been presented in a tabular form (see table no. 9).

To evaluate the proposed ANFIS based AODV routing protocol, the following performance based metrics were used for simulation model:

Table 8: Performance parameters

Parameters	Value
Simulator	NS-2.35
Simulation time	50s
Packet size	256,512,1024, 2048, 3072, 4096,4500
No. of Mobile Nodes	5
Routing protocol	AODV
Mac Type	802.11
Max Connection	4
Max. Speed	40 m/s
Radio Propagation Model	Two Ray Ground
Network Size	x=512, y=468
Queue Size	50
Antenna Model	Omni Antenna
Channel	Wireless Channel
Pause time	0s

```
setdest –V1 –n 5 –p 0 –M 40 –t 50 –x 100 –y 100 >node-30-size-100
```

IV. SIMULATION RESULTS AND DISCUSSION

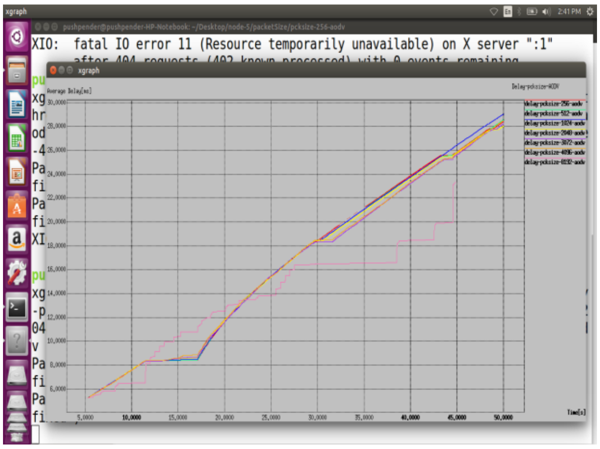

Figure 9: Average Delay with varying the packet sizes

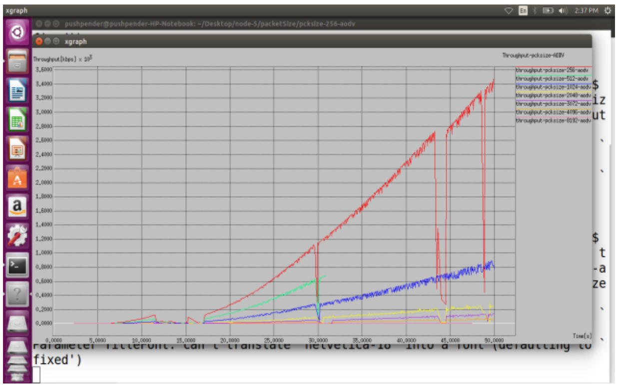

Figure 10: Throughput with varying the packet sizes

Table 9: Performance metric values w.r.t. Packet size

Packet Size	NRL	Average Throughput	Data Packet	Routing packet	PDR	Dropped Packets	Average E2E Delay
256	0.024	365.4	4629	112	98.171	0	125.187
512	0.034	304.83	3455	119	97.964	52	193.362
1024	0.053	398.87	2346	124	97.192	40	174.015
2048	0.082	450.37	1351	111	95.38	41	320.405
3072	0.121	476.09	969	116	94.099	43	478.712
4096	0.144	493.33	749	108	93.256	55	552
4500	0.166	473.59	662	110	92.55	53	635.407

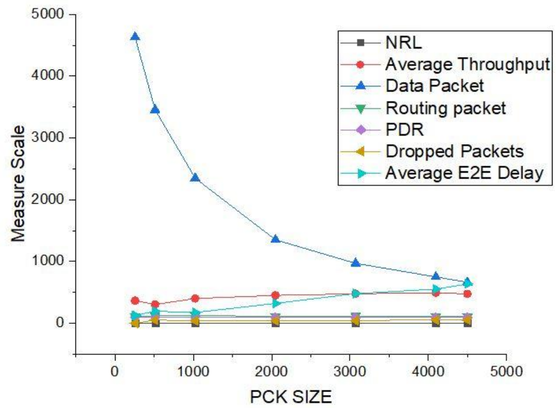

Figure 11: packet size w.r.t. performance metric values

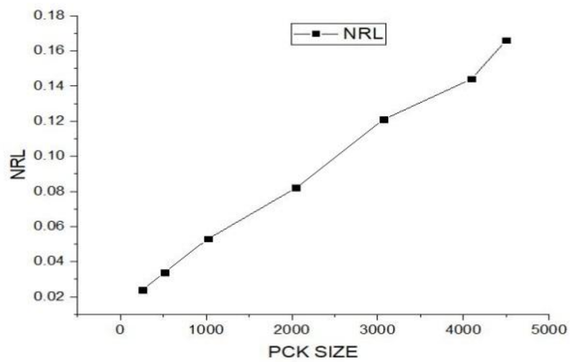

Figure 12: Packet size Vs NRL

A. Result analysis for neuro fuzzy model and NS-2 based outputs:

Table 10 (a): comparison of ANFIS results and NS-2 results

Throughput(ns-2)	Throughput(ANFIS)	Packet size(ANFIS)	Packet size(ns-2)
365.40	365	256	256
304.83	305	517	512
398.87	399	995	1024
450.37	450	2060	2048
476.09	476	2990	3072
493.33	493	4040	4096
473.59	476	2990	4500

Table 10 (b): comparison of ANFIS results and NS-2 results

Data Packet Received(ns-2)	Data Packet Received (ANFIS)	Packet size(ANFIS)	Packet size(ns-2)
4629	4630	256	256
3455	3460	517	512
2346	2350	995	1024
1351	1350	2060	2048
969	969	2990	3072
749	749	4040	4096
662	662	4520	4500

Table 10 (a): comparison of ANFIS results and NS-2 results

E2E Delay (ns-2)	E2E Delay (ANFIS)	Packet size(ANFIS)	Packet size(ns-2)
125.187	125	256	256
193.362	193	517	512
174.015	174	995	1024
320.405	320	2060	2048
478.712	479	2990	3072
552	552	4040	4096
635.407	635	4520	4500

As depicted in table 9, table 10(a), table 10(b), table 10(c), and figure 9, figure 10, figure 11, results are summarized as given below:

Throughput:

Mostly as packet size is increasing, throughput is also increasing in both cases (ANFIS, NS-2). Also there is no big difference for throughput results got from ANFIS and NS-2. For throughput 450 kbps, we have packet size 2060 bytes for ANFIS and 2048 for NS-2. For throughput 476 kbps, packet size is 2990 for ANFIS while it is 3072 bytes for NS-2 results. But in some cases, it total different situation. As for throughput 473.59 kbps (NS-2), packet size is 4500 bytes, while for throughput 476kbps, packet size is 4500 bytes.

Data packet received:

For data packet received in both cases (ANFIS, NS-2), packet size is mostly same. For data packet 4630, we need packet size 256 bytes. For data packet received 2350, we require 995 bytes packet size (in case of ANFIS) while it is 1024 bytes (in case of NS-2). For data packet received 749, 4040 bytes packet size is to settled(in case of ANFIS) while in case of NS-2, this value is 4096 bytes. Data packet received 662, we need 4520 bytes packet size (for ANFIS), but for NS-2, it is 4500 bytes.

Average End to End Delay:

E2E delay 125ms, we have to set the packet size 256 bytes (ANFIS and NS-2). Average end to end delay 512 ms, we need packet size 4040 bytes (ANFIS) while in case of NS-2, and this value is 4096 bytes. To minimize the average end to end delay, packet size 256 bytes is best in our experimental work.

It has been observed that as the packet size will be increased, average end to end delay will be increased. Packet size 1024 bytes (in case of AODV routing protocol) is best optimize packet size to minimize the average end to end delay.

V. CONCLUSION

Adaptive neuro fuzzy inference system is very strong fuzzy system that automatically converts input values into single output result. Here, five layers of the system contribute for implementing the accurate results. The proposed packet size based model for AODV routing protocol has been verified with network simulator-2 results. Mostly, in all cases, model is fit at all performance metrics. Packet size 1024 bytes is optimized packet size for achieving maximum throughput with minimum average end to end delay. As packet size is large, throughput will be very high, but average end to end delay will be increased. In future, more performance metrics will be considered for deciding the packet size in wireless ad-hoc networks. For getting more accurate results, more detailed data information for training data set and checking data set will be applied. In future work, we will extend this research work to implement the intelligent model for DSR and AOMDV routing protocols by considering more performance metrics.

REFERENCES

1. Vivek Sharma, Bashir Alam and M.N. Doja, “ANFIS Aided AODV Routing Protocolfor Mobile Ad Hoc Networks”, Journal of Computer Science 2017, 13 (10): 514.523 (DOI: 10.3844/jcssp.2017.514.523)
2. Zahian Ismail, Rosilah Hassan, “Effects of Packet Size on AODV Routing Protocol Implementation in Homogeneous and Heterogeneous MANET”, 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, IEEE Computer Society, pp. 351-356.
3. Deepika, Pankaj Kumar Verma, Vijay Kumar, “Impact of Packet Size and Number of Nodes on DSR and AODV Routing Protocols in MANET”, IJCAT International Journal of Computing and Technology, Volume 1, Issue 4, May 2014, pp. 66-72.
4. Sasmita Acharya, C.R. Tripathy, “An ANFIS estimator based data aggregation scheme for fault tolerant Wireless Sensor Networks”, Journal of King Saud University – Computer and Information Sciences (2018) 30, 334–348(hosting by Elsevier B.V. on behalf of King Saud University).
5. Wafa Benaatou1, Adnane Latif1, Vicent Pla, “Applying ANFIS Model in Decision-making of Vertical Handover between Macrocell and Femtocell Integrated Network”, Journal of Telecommunication, Electronic and Computer Engineering, Vol. 11 No. 1 January - March 2019, pp. 57-62.
6. Rajnish K Yadav and Manoj Balakrishnan, “Comparative evaluation of ARIMA and ANFIS for modeling of wireless network traffic time series”, EURASIP Journal on Wireless Communications and Networking (Springer), 2014:15, pp. 1-8.
7. Bashir Alam and M.N. Doja, “ANFIS -AODV Routing Protocol for Mobile Ad Hoc Networks”, Journal of Computer Science 2017, 13 (10): 514.523 DOI: 10.3844/jcssp.2017.514.523
8. Suman Sangwan, Parvinder Singh, P B Patel, “Periodic Route Selection for secure packet delivery”, ACM International Conference Proceeding Series, September, 2012, DOI - 10.1145/2381716.2381755
9. https://www.isi.edu/nsnam/ns/

10. https://www.nsnam.com/2013/03/awk-scripts-for-ns2-to-process-data.html
11. Pushpender Sarao, "A Table Driven Route Analysis in MANETS", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019
12. Pushpender Sarao, "F-EEAODV: Fuzzy Based Energy Efficient Reactive Routing Protocol in Wireless Ad-hoc Networks", Journal of Communications Vol. 13, No. 7, July 2018, pp. 350- 356.
13. Pushpender Sarao, "AODV-F: A New on Demand Enhanced Distance Vector Routing in Wireless Network", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-3, September 2019, pp. 5183- 5191.(DOI:10.35940/ijrte.C5806.098319)
14. Ad, Mobile & Perkins, Charles & Das, Samir. (2000). Ad hoc On-Demand Distance Vector (AODV) Routing.
15. K. Weniger, M. Zitterbart, "Mobile ad hoc networks - current approaches and future directions", IEEE Networks, Volume 18, Issue 4, July-Aug. 2004, pp. 6-11, DOI: 10.1109/MNET.2004.1316754

AUTHOR'S PROFILE

Dr. Pushpender Sarao is a Professor in CSE department at Hyderabad Institute of Technology and Management, Hyderabad, India.. He is BE (IT, honors), M.Tech, PhD in Computer Science and Engineering. He has worked as HOD and Dean Academics at Somany Institute of Technology & Management, Rewari, Haryana (2011-2016). In 2013, he was awarded with "Best Teacher Award-2013" by Somany Educational Society. He is author of five books in computer science, wireless networks. He is a member of IEEE, ACM, ICSES, RES, and IAENG, and life member of CSI, ISTE and IEI. He got published more than 67 research papers in national and international reputed journals. Also he shared his research experience in more than 24 national and international conferences. He attended several FDP and seminars/workshops in engineering institutions and state universities. He is member of editorial board for 19 national and international journals. His main research work focuses on routing protocols in wireless mesh networks, mobile ad-hoc network. He has 9 years of teaching experience and 10 years of Industrial Experience.