ARTICLE OPEN ACCESS

IMPROVING PERFORMANCE OF DSR WITH OPTIMIZED FUZZY RULES

Menaka[1*] and Ranganathan [2]
*[1]Dept. of Electronics and Communication Engineering, Dhanalakshmi College of Engineering, Chennai, TN, INDIA
[2]Dean, Dept. of Electrical and Instrumentation, Dr. Mahalingam College of Engineering & Technology, TN, INDIA*

ABSTRACT

In MANET environment the security of every individual network node is crucial due to the pervasive nature of MANETs. Securing the routing algorithms for ad hoc networks is an exceptionally difficult errand due its unmistakable qualities. The DSR Protocol is a source-routed on-demand routing protocol. In this study, Dynamic Source Routing (DSR) protocol is evaluated in untrustworthy environment and to improve performance of multi-objective Particle Swarm Optimization (PSO) and Quantum Annealing (QA) with fuzzy rule selection is proposed. Outcomes reveal that the suggested method performs better than the existing methods.

Published on: 10th– August-2016

KEY WORDS

MANET, Dynamic Source Routing (DSR) protocol and Multi-Objective Particle Swarm Optimization (MOPSO) with fuzzy rule selection.

***Corresponding author:** **Email:** menaka_2014@rediffmail.com

INTRODUCTION

MANET is the new developing innovation, which empowers users to communicate with no physical infrastructure regardless of their geographical location that is the reason it is sometimes alluded to as an infrastructure less network. MANETs are self-organizing groups of mobile nodes, which additionally work as router, joined by wireless links. MANETS don't have a centralized infrastructure, namely there is no fixed focal node to organize the task of routing.

Reactive DSR Protocol's operation is split into two stages; path discovery phase as well as path maintenance phase, these phases are activated on demand when a packet needs routing. Route discovery phase floods the network with RREQs if a suitable route is not accessible in the route [1]. DSR utilizes a source routing system to produce a complete route to the destination, this will then be stored briefly in nodes route cache. DSR addresses mobility issues using packet affirmations; failing to obtain an affirmation causes packets to be buffered and path error messages to be transmitted to every upstream node. Path error messages trigger path maintenance phase that expels wrong routes from the route cache and embraces another route discovery phase.

Fuzzy rule-based systems have been successfully applied to solve many classification problems. In many classification problems, fuzzy classification rules are de- rived from human experts as linguistic knowledge. Because it is not usually easy to derive fuzzy rules from human experts, many approaches have recently been proposed to generate fuzzy rules automatically from the training patterns of the considered classification problem [2]. In order to generate fuzzy rules from training patterns, fuzzy partitions in input spaces are typically regarded for determining premise part of fuzzy classification mode. Grid-type fuzzy partitions of input space and the scatter-type fuzzy partition of the input data have been often used to model fuzzy systems for training patterns. A heuristic method for generating fuzzy rules is applied to the grid-type fuzzy partitions, and a rule selection method, based on PSO algorithms, is then employed to select the relevant fuzzy rules [3] from the generated fuzzy rules. In the PSO-based approach, each individual in the population is considered to represent a fuzzy classification system. Then, a fitness function is implemented for guiding the search process to choose adequate fuzzy classification model as well as number of fuzzy rules.

Optimization issues which have more than one objective function are typical in every field or area of knowledge. In those problems, objectives to be optimized are generally in conflict with one other, which implies that there is

ISSN: 0976-3104

no one solution for these problems. Rather, aimed to find good "trade-off" solutions that represent the best possible compromises among the objectives.

Quantum annealing (QA) is a computational paradigm for searching for the minimal cost function (muti-variable function to be made minimum) through control of quantum fluctuation. Quantum annealing is utilized primarily for combinatorial optimization issues with discrete parameters. In this paper, the traditional DSR protocol is modified in the proposed method to improve Quality of Service (QoS) in untrustworthy environment. PSO and QA with fuzzy rules are propsoed for improved performance. Section 2 explains the related works, section 3 explains the methods used for the research, section 4 obtains the results and discussed on it and section 5 concludes the work.

LITERATURE SURVEY

Chen et al., [4] presented an Entropy-based Fuzzy controllers QoS Routing algorithm in MANET (EFQRM). The key thought of EFQRM algorithm is to build the new metric-entropy and fuzzy controllers with the assistance of entropy metric to decrease the number of route reconstruction in order to give QoS guarantee in the ad hoc network. The simulation results revealed that the suggested methodology and parameters gave an accurate and efficient technique for estimating and assessing the route stability in dynamic MANETs.

Geetha and Thangaraj [5] examined the effect of node mobility on trust establishment is considered and its utilization to propagate trust through a network. This work proposed an enhanced Associativity Based Routing (ABR) with Fuzzy based Trust (Fuzzy-ABR) routing protocol for MANET to enhance QoS and to mitigate network attacks.

Srivastava and Daniel [6] proposed a routing algorithm for the mobile ad hoc networks based on fuzzy logic to find an optimal route for transmitting data packets to the destination. This protocol helped each node in MANET to pick next efficient successor node on the premise of channel variables such as environmental noise as well as signal strength. The protocol enhanced the performance of a route by expanding network life time, diminishing link failure as well as choosing best node for transmitting data packets to next node.

Gupta et al., [7] proposed a routing algorithm based on Fuzzy Logic which is having low communication overhead and storage necessities. The proposed algorithm taken three information variables: signal force, mobility and delay. The supreme value of every parameter could take an extensive range at different points on the network.

Dahiya and Dureja [8] proposed a fuzzy based efficient routing protocol (FBERP) for substantial scale mobile ad-hoc networks that expected to minimize the packet loss rate. Every node in the network is portrayed by its communication parameters. The authors added to a fuzzy logic controller that consolidated these parameters, Packet Loss Rate, Communication Rate, Energy and Delay Parameters. The value acquired, demonstrated the need of a node and it is utilized as a part of route formation. The simulation demonstrated that the proposed protocol outperformed the standard AOMDV routing protocol in minimizing the packet loss.

METHODOLOGY

Fuzzy Modeling

Fuzzy Modelling (FM) [12] usually comes with two contradictory requirements to the obtained model: the interpretability, capability to express the behaviour of the real system in an understandable way, and the accuracy, capability to faithfully represent the real system. Since they are contradictory issues, more priority has generally been given to one of them (defined by the problem nature), leaving the other one in the background. Two FM approaches arise depending on the main objective to be considered:

- Linguistic FM, mainly developed by means of linguistic (or Mamdani) FRBSs, which is focused on the interpretability.
- Precise FM, mainly developed by means of Takagi-Sugeno FRBSs, which is focused on the accuracy.

Regardless of the approach, a common scheme has been considered to attain the desired balance between interpretability and accuracy. Firstly, the main objective (interpretability or accuracy) is tackled defining a specific model structure to be used, thus setting the FM approach.

Then, the modelling components (the model structure and/or the modelling process) are improved by means of different mechanisms to compensate for the initial difference between both requirements. Thus, accuracy improvements are proposed in linguistic FM at the cost of part of the interpretability whilst interpretability improvements are proposed in precise FM at the cost of part of the accuracy.

Actually, the interpretability-accuracy trade-off is a very important branch of research nowadays. Focusing on Linguistic FM with improved accuracy1 (still nearer of the interpretability) many examples can be find in the existing literature. This approach has been performed by learning/tuning the MFs by defining their parameters or shapes, their types (triangular, trapezoidal, etc.), or their context (defining the whole semantics), learning the granularity (number of linguistic terms) of the fuzzy partitions or extending the model structure by using linguistic modifiers, weights (importance factors for each rule), or hierarchical architectures (mixing rules with different granularities), among others. The main problem of these approaches is that although the system accuracy can be greatly improved (e.g., with a simple tuning of MFs), the original interpretability of the linguistic models is lost to some degree giving way to more complex systems or less interpretable rule structures. Additionally, although rule base reduction and input variable selection [13] processes improve the interpretability, they can also help to improve the accuracy when redundancy and inconsistency criteria are considered (but usually these improvements are not very significant).

Tuning of MFs usually needs an initial model with large number of rules to get an appropriate level of accuracy. Generally, to obtain a good number of initial rules, methods ensuring covering levels higher than needed are used. In this way, rules are obtained that being needed at first could be unnecessary once the tuning is applied or rules that could impede the tuning of the remaining ones in order to obtain the global optimum in terms of the accuracy (better configuration of rules to get the minimum error after tuning of the parameters). Thus, find the following types of rules respect to this global optimum in the complete set of rules: Bad Rules (erroneous or conflicting rules) that degrade the system performance (rules that are not included in the most accurate final solution); Redundant or Irrelevant Rules that do not significantly improve the system performance; Complementary Rules that complement some others slightly improving the system performance; and Important Rules that should not be removed to obtain a reasonable system performance. Obviously, this is a simplification of the problem by only considering in principle the most accurate solution in order to have an idea of the shape of the optimum Pareto. On the other hand, to determine those types of rules in advance is impossible since it directly depends on each concrete configuration of rules and still more on the optimal configuration of the MF parameters for each rule configuration. Therefore, this is impossible to establish any criteria that could be used in the search process. However, by taking into account the possible existence of these kinds of rules, different rule configurations and different tuning parameters, can estimate the following zones in the space of the objectives:

- Zone with Bad Rules, which contains solutions with bad rules. In this zone, the Pareto front does not exist given that removing these kinds of rules would improve the accuracy and these solutions would be dominated by others.
- Zone with Redundant or Irrelevant Rules, which is comprised of solutions with- out bad rules but still maintaining redundant or irrelevant rules. By deleting these kinds of rules the accuracy would be practically the same.
- Zone with Complementary Rules, comprised of solutions without any bad or redundant rule. By removing these rules the accuracy would be slightly decreased.
- Zone with Important Rules, which contains solutions only comprised of essential rules. By removing these kinds of rules the accuracy is really affected.

Proposed Particle Swarm Optimization (PSO) -Fuzzy Rules

Particle Swarm Optimization Algorithm (PSO) refers to a population-based optimization technique that finds the optimal solution using a population of particles [9, 10]. All swarms of PSO are solutions in solution space. PSO is fundamentally developed by simulating the flocking of birds. PSO is defined thus: All individual particles i have the following characteristics: Current positions in search space, x_{id} , a current velocity, p_{id} , and a personal best position in search space, p_{id} .

• The personal best position, p_{id} , relates to the position in search space wherein particle I represents smallest error as given by objective function f, presuming minimization task.

• The global best position denoted by represents the position yielding the lowest error amongst all the p_{gd} .

During the iteration all particles in the swarm are updated through equations (1 and 2):

Particles Velocity

$$v_{id} = w * v_{id} + c_1 * rand() * (\mathrm{p}_{id} - x_{id}) + c_2 * rand() * (\mathrm{p}_{gd} - x_{id}) \tag{1}$$

The current position of particles are updated for obtaining the subsequent position.

Particles Position

$$x(t+1) = x(t) + v(t+1) \tag{2}$$

wherein c_1 and c_2 refer to two positive constants, c_1 and c_2 are two arbitrary numbers within the range [0,l], and w is the inertia weight.

For applying the PSO scheme to solve multi-objective optimization issues, it is obvious that the original scheme has to be modified. Solution sets of problems with several objectives do not comprise of single solutions (as in global optimization) are

THE IIOAB JOURNAL ISSN: 0976-3104

explained. Instead, in multi-objective optimization, aim to find sets of different solutions (known as Pareto optimal set). Generally, when resolving multi-objective issue, there are three primary goals to attain [11]:

1. Make the number of elements found in the Pareto optimals set maximum
2. Make the distance of the Pareto front yielded by the suggested algorithm with respect to true (global) Pareto front minimum
3. Make the spread of solutions discovered maximum, so that it can have a distribution of vectors as smooth and uniform as possible.

First, the swarm is initialized. Next, sets of leaders are also initialized with non-dominated particles from swarms. As previously noted, sets of leaders are typically stored in external archives. Later, certain quality metrics are computed for all leaders for selecting one leader for all particles of swarms. With every generation, leaders are chosen and flight is performed. Almost all existing MOPSOs employ a kind of mutation operator after performing the flight. Later, particles are evaluated and corresponding pbest are updated. A new particle replaces its pbest particle usually when this particle is dominated or if both are incomparable (i.e., they are both non-dominated with respect to one another). After all particles are updated, set of leaders are updated as well. In the end, quality metric of the set of leaders are recomputed. The procedure is iterated for a specified (usually fixed) number of iterations.

Pseudo code of a general MOPSO algorithm

```
Begin
Initialize swarm
Initialize leaders in an external archive
Quality(leaders)
g = 0
While g < gmax
For each particle
Select leader
Update Position (Flight)
Mutation
Evaluation
Update pbest
EndFor
Update leaders in the external archive
Quality(leaders)
g++
EndWhile
Report results in the external archive
End
```

All required data regarding rule-base or memberships functions are to be definitely specified. They are used to accurately represent fuzzy logic.

PSO based Fuzzy Controller Design Method

The PSO algorithm [14] is a computation technique pro- posed by Kennedy and Eberhart. Its development was based on observations of the social behavior of animals such as bird flocking and fish schooling of the swarm theory. If a set p^g with N particles is called a population in the g-th generation and expressed by equation (3):

$$p^g = \{\mathrm{p}_1^g, \mathrm{p}_2^g, \ldots\ldots, \mathrm{p}_h^g, \ldots\ldots, \mathrm{p}_N^g\} \qquad (3)$$

The position vector and velocity vector of the h-th particle $(h \in \{1, 2, \ldots, N\})$ in the g-th generation $(\mathrm{g} \in \{1, 2, \ldots, \mathrm{G}\})$ are respectively denoted by equation (4 and 5):

$$p_h^g = (p_{(h,1)}^g, p_{(h,2)}^g, \ldots\ldots, p_{(h,\mathrm{j})}^g, \ldots, p_{(h,\mathrm{n})}^g) \qquad (4)$$

And

$$v_h^g = (\mathrm{v}_{(h,1)}^g, \mathrm{v}_{(h,2)}^g, \ldots\ldots, \mathrm{v}_{(h,\mathrm{j})}^g, \ldots, \mathrm{v}_{(h,\mathrm{n})}^g) \qquad (5)$$

where n is the number of searching parameters and $p_{(h,j)}^g$ denotes the position of the j-th parameter $(j \in \{1, 2, \ldots.n\})$ of the h-th particle in the g-th generation.

THE IIOAB JOURNAL ISSN: 0976-3104

The procedure of PSO algorithm can be described as follows:

Step 1: Initialize PSO by setting g=1, $F_1^{pbest} = F_2^{pbest} = ... = F_N^{pbest} = 0,$ the maximum number of generation (G), the number of particles (N), and four parameter values of $c_1, c_2, \omega_{\max}$ and $\omega_{\min}$.

Step 2: Generate the initial position vector $p_h^1 = (p_{(h,1)}^1, p_{(h,2)}^1, ..., p_{(h,j)}^1, ..., p_{(h,n)}^1)$ and the initial velocity vector $v_h^1 = (v_{(h,1)}^1, v_{(h,2)}^1, ..., v_{(h,j)}^1, ..., v_{(h,n)}^1)$ of N particles randomly byequation (6):

$$p_{(h,j)}^1 = p_j^{\min} + (p_j^{\max} - p_j^{\min})\,\text{rand}()$$

and

$$v_{(h,j)}^1 = \frac{(\text{v}_j^{\max} - v_j^{\min})}{20} rand() \tag{6}$$

where $p_j^{\max}$ and $p_j^{\min}$ are respectively the maximum value and minimum value of the j-th parameter. $\text{v}_j^{\max}$ and $v_j^{\min}$ are the maximum velocity and minimum velocity the j-th parameter. rand() is an uniformly distributed random number in [0,1].
Step 3: Calculate the fitness value of each particle in the g-th generation byequation (7):

$$F(p_h^g) = fit(p_h^g), h = 1, 2, ...N \tag{7}$$

Where $fit(.)$ is the fitness function.

Step 4: Determine F_h^{pbest} and p_h^{pbest} for each particle by equation (8):

$$F_h^{pbest} = \begin{cases} F_h^g, & \text{if } F_h^{pbest} \le F_h^g \\ F_h^{pbest}, otherwise \end{cases}, h \in \{1, 2, ..., N\}$$

and

$$p_h^{pbest} = \begin{cases} p_h^g, & \text{if } F_h^{pbest} \le F_h^g \\ p_h^{pbest}, otherwise \end{cases}, h \in \{1, 2, ..., N\} \tag{8}$$

Where p_h^{pbest} is the position vector of the h-th particle with the personal best fitness value F_h^{pbest} from the initial to the current generation.
Step 5: Identify an index q of the particle with the highest fitness by equation (9):

$$q = \arg \max_{h \in \{1,2,...,N\}} F_h^{pbest}$$

$$and \text{ determine } \text{F}^{Gbest} \text{ and } \text{p}^{Gbest} \text{ by}$$

$$\text{F}^{Gbest} = F_q^{Pbest} = \max_{h \in \{1,2,...,N\}} F_h^{Pbest}$$

$$and$$

$$\text{p}^{Gbest} = p_q^{Pbest} \tag{9}$$

Wherein p^{Gbest} refers to position vector of the particle with the global best fitness value F^{Gbest} from the beginning to the current generation.
Step 6: If g=G, then go to Step 12, Otherwise, go to Step 7.
Step 7: Update the velocity vector of each particle by equation (10):

$$v_h^{g+1} = \omega.v_h^g + c_1.rand1().(p^{Gbest} - p_h^g) + c_2.rand2().(p^{Gbest} - p_h^g) \tag{10}$$

Where v_h^g and is the current velocity vector of the h-th particle in the g-th generation. v_h^{g+1} is the next velocity vector of the h-th particle in the (g+1)-th generation. $rand1()$ and rand2() refer to two uniformly distributed arbitrary numbers in [0,1]. 1c and 2 c are constant values. ω is a weight value and defined by equation (11):

ISSN: 0976-3104

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{G}.g \qquad (11)$$

Where $\omega_{\max}$ and $\omega_{\min}$ are respectively a maximum value and a minimum value of ω.Step 8: Check the velocity constraint by equation (12):

$$v_{(h,j)}^{g+1} = \begin{cases} v_j^{\max}, & \text{if } \mathrm{v}_{(h,j)}^{g+1} > v_j^{\max} \\ \mathrm{v}_{(h,j)}^{g+1}, & \text{if } v_j^{\min} \le \mathrm{v}_{(h,j)}^{g+1} \le v_j^{\max} \\ v_j^{\min}, & \text{if } \mathrm{v}_{(h,j)}^{g+1} < v_j^{\min} \end{cases} \qquad (12)$$

H=1,2,….,N, j=1,2,…,n.
Step 9: Update position vectors of all particles by equation (13):

$$p_h^{g+1} = p_h^g + v_h^{g+1} \qquad (13)$$

Where p_h^g is the current position vector of the h-th particle in the g-th generation. v_h^{g+1} is the next position vector of the h-th particle in the (g+1)-th generation.
Step10: Bound the updated position vector of each particle in the searching range by equation (14):

$$p_{(h,j)}^{g+1} = \begin{cases} p_j^{\max}, & \text{if } \mathrm{p}_{(h,j)}^{g+1} > p_j^{\max} \\ p_{(h,j)}^{g+1}, & \text{if } \mathrm{p}_j^{\min} \le p_{(h,j)}^{g+1} \le p_j^{\max} \\ p_j^{\min}, & \text{if } \mathrm{p}_{(h,j)}^{g+1} < p_j^{\min} \end{cases} \qquad (14)$$

H=1,2,….,N, j=1,2,…,n.
Step11: Let g=g+1 and go to Step 3.

Step12: Determine the corresponding fuzzy controller based on the position of the particle p^{Gbest} with the best fitness value F^{Gbest}.

Quantum Annealing (QA)

Quantum mechanics works with wave-functions that can equally well sample wide regions of phase-space. Instead of thermal fluctuations, one exploits here the quantum fluctuations provided by a suitably introduced – and equally artificial – kinetic energy. Annealing is then performed by slowly reducing to zero the amount of quantum fluctuations introduced. Quantum fluctuations have, in many respects, an effect similar to that of thermal fluctuations – they cause, for instance, solid helium to melt even at the lowest temperatures – but they differ considerably in other respects. In particular, quantum systems can tunnel through classically impenetrable potential barriers between energy valleys, a process that might prove more effective than waiting for them to be overcome thermally.

Quantum PSO (QPSO) is that which drops velocity vector of the original PSO and subsequently changes updating scheme of particles' positions for making searches more simple as well as effective [15]. Pseudo code for proposed method is:

ISSN: 0976-3104

```
Begin
step1: initialization
          PSO
          Initialize a swarm by the proposed schedule initialization algorithm
          give initial value: $w_{max}, w_{min}, c_1, c_2, Gen$ and generation=0;
          set indicator m=0
         QA
         set initial temperature T, final temperature $T_0$
Step: 2 Iteration process
do{
Generate next swarm;
find new gbest and pbest;
update gbest of swarm and pbest of each particle;
generation++;
if(gbest is not improved)
m++;
end if
if (m==Gen)
{
while ($T>T_0$) do
{
$s' \leftarrow Produce\_Solution(s)$;
$Compute(s, s')$
if ($\Delta E \le 0$) or ($\exp(-\Delta E / T) > rand(0,1)$)
$s \leftarrow s', T \leftarrow BT$
end if
}
end while
m=0;
} endif
} while (termination condition is not satisfied)
Step 3: Output Optimization solution
End
```

RESULTS AND DISCUSSION

Simulations were carried out using OPNET in a rectangular network of size 4 Square Kilometer. The simulation parameters are given in **[Table- 1]**. Experiments are conducted with 25 nodes with random mobility. Throughput, end to end delay and percentage of malicious node detected are calculated and the simulation results are drawn in graphs and in tables. To imitate real-time situation, untrustworthy nodes are added to the network.

ISSN: 0976-3104

Table: 1. Simulation Setup

Total area of the network	4 sq.km
Number of nodes	25
Mobility of the nodes	constant speed of 10 kmph, 20 kmph, 30 kmph, 40 kmph, and 50 kmph
Data rate	2Mbps
Routing Protocol	DSR & Proposed
Node mobility	10, 20, 30, 40 and 50 Kmph
Transmission range of node	200 m
Data type	Constant Bit Rate

[Table-2, 3, 4] and [Figure- 1], [Figure- 2] , [Figure- 3] shows the throughput, end to end delay and percentage of malicious node detected respectively that are obtained from different experiments.

Table: 2. Throughput

Throughput x 100 %	FR-DSR	FR-DSR in untrustworthy environment	QAFR-DSR	QAFR-DSR-Untrustworthy	MOPSOFR-DSR	MOPSOFR-DSR in untrustworthy environment
No mobility	0.9715	0.9465	0.9784	0.9563	0.9853	0.9627
15 kmph	0.9372	0.9275	0.9442	0.9408	0.9553	0.9416
30 kmph	0.9246	0.9087	0.9329	0.9152	0.939	0.9216
45 kmph	0.9026	0.8726	0.9126	0.8849	0.915	0.8868
60 kmph	0.8547	0.7847	0.8674	0.7959	0.8713	0.7967
75 kmph	0.8109	0.7672	0.8177	0.7738	0.8229	0.7781

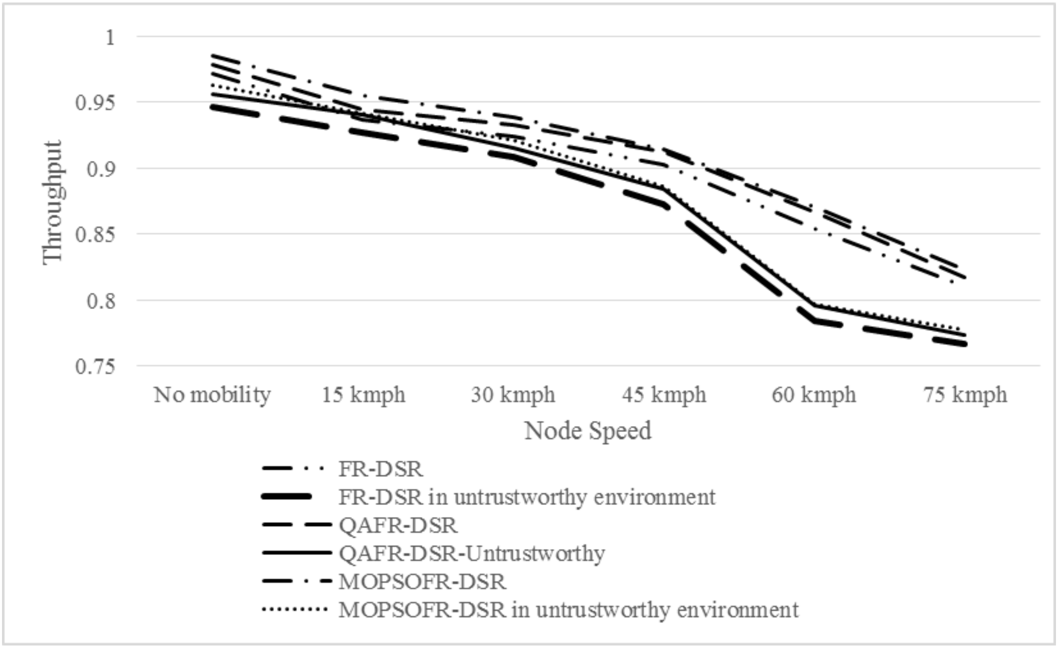

Fig: 1. Throughput

THE IIOAB JOURNAL
ISSN: 0976-3104

From **Table- 2** and **Figure- 1** it is observed that the throughput of MOPSOFR-DSR performs better by 1.41% than FR-DSR, by 4.02% than FR-DSR in untrustworthy environment, by 0.7% than QAFR-DSR, by 2.99% than QAFR-DSR untrustworthy, by 2.32% than MOPSOFR-DSR in untrustworthy environment for no mobility. The throughput gets decreased when the node speed increases. For node speed 75 kmph, the throughput of MOPSOFR-DSR performs better by 1.47% than FR-DSR, by 7% than FR-DSR in untrustworthy environment, by 0.63% than QAFR-DSR, by 6.15% than QAFR-DSR untrustworthy, by 5.59% than MOPSOFR-DSR in untrustworthy environment.

Table: 3. End to End Delay

End to End Delay ms	FR-DSR	FR-DSR in untrustworthy environment	QAFR-DSR	QAFR-DSR-Untrustworthy	MOPSOFR-DSR	MOPSOFR-DSR in untrustworthy environment
No mobility	0.00173	0.00194	0.00167	0.00189	0.001641	0.00189
15 kmph	0.00262	0.00287	0.00253	0.00279	0.002532	0.002702
30 kmph	0.00363	0.00376	0.00351	0.00368	0.003527	0.003681
45 kmph	0.00384	0.00421	0.00372	0.00406	0.003689	0.004114
60 kmph	0.00457	0.00468	0.00445	0.00451	0.004307	0.00445
75 kmph	0.00474	0.00482	0.00458	0.00464	0.004516	0.00462

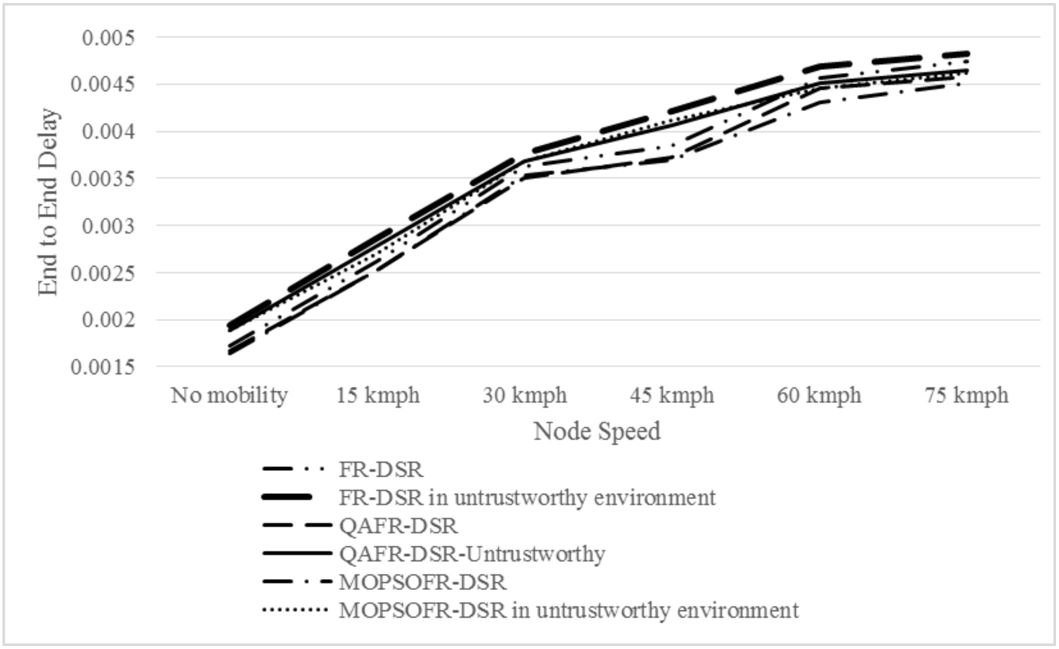

Fig: 2. End to End Delay

From **Table- 3** and **Figure- 2** it is observed that the End to End Delay of MOPSOFR-DSR performs better by 5.28% than FR-DSR, by 16.69% than FR-DSR in untrustworthy environment, by 1.75% than QAFR-DSR, by 14.0% than QAFR-DSR untrustworthy, by 14.1% than MOPSOFR-DSR in untrustworthy environment for no mobility. The end to end delay gets increased when the node speed increases. For node speed 75 kmph, the end to end delay of MOPSOFR-DSR performs better by 4.84% than FR-DSR, by 6.5% than FR-DSR in untrustworthy environment, by 1.41% than QAFR-DSR, by 2.71% than QAFR-DSR untrustworthy, by 2.28% than MOPSOFR-DSR in untrustworthy environment.

ISSN: 0976-3104

Table: 4.Percentage of Malicious Node Detected

Percentage of Malicious node detected	Initial state of DSR	Steady state of FR-DSR	Initial state of QAFR-DSR	Steady state of QAFR-DSR-Untrustworthy	MOPSOFR-DSR	MOPSOFR-DSR in untrustworthy environment
No mobility	0.4	0.9	0.47	0.94	0.5	0.9
15 kmph	0.4	0.8	0.45	0.94	0.4	0.9
30 kmph	0.3	0.8	0.34	0.93	0.4	0.8
45 kmph	0.3	0.7	0.35	0.83	0.3	0.7
60 kmph	0.2	0.6	0.22	0.69	0.3	0.7
75 kmph	0.1	0.4	0.12	0.45	0.2	0.5

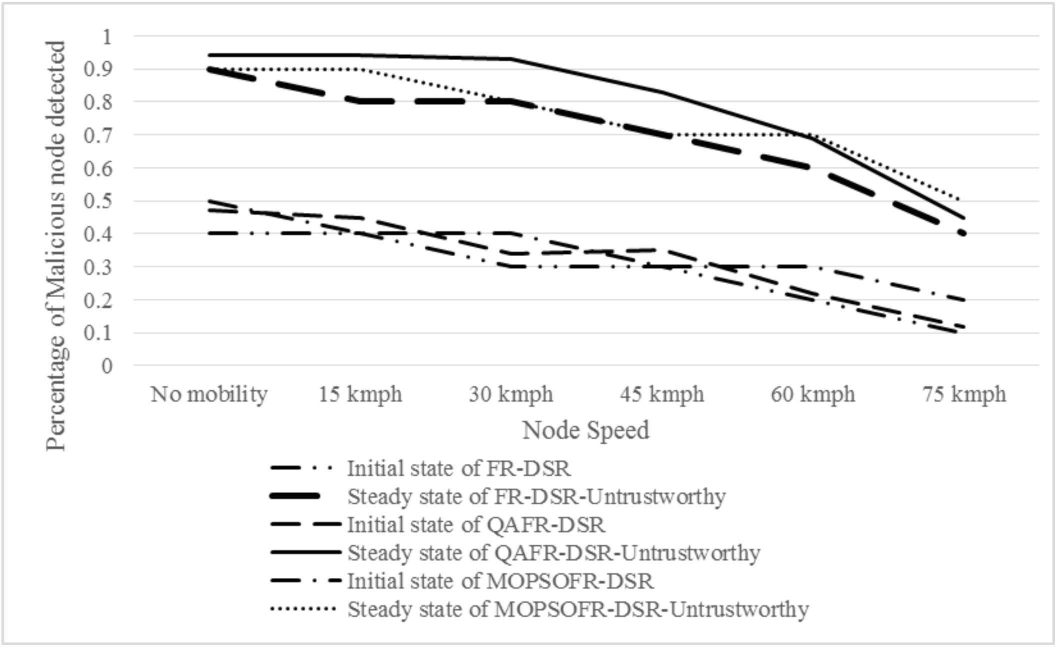

Fig: 3. Percentage of Malicious Node Detected

From **Table- 4** and **Figure- 3** it is observed that the Percentage of Malicious Node Detected of Initial state of MOPSOFR-DSR performs better by 22.22% than initial state of FR-DSR and by 6.19% than initial state of QAFR-DSR, the steady state of MOPSOFR-DSR performs equally to steady state of FR-DSR and by 4.35% than steady state of QAFR-DSR for no mobility. The Percentage of Malicious Node Detected gets decreased when the node speed increases. For node speed 75 kmph, the Percentage of Malicious Node Detected of MOPSOFR-DSR performs better by 6.66% than initial state of FR-DSR and by 50% than initial state of QAFR-DSR, the steady state of MOPSOFR-DSR performs better by 22.22% than steady state of FR-DSR and by 10.53% than steady state of QAFR-DSR.

CONCLUSION

All nodes are to maintain excellent reputation values for receiving network services. Only by forwarding other nodes' packets a node can maintain a high reputation value. It also affects the reality of the data transmission. In such untrustworthy environment, it is difficult to achieve high throughput. In this proposed model, the trust level of all the nodes are calculated and analyzed to identify the capability of nodes. This calculated trust values along with node mobility and inference from previous records are also used to calculate the nodes reputation value. Results show that the throughput of MOPSOFR-DSR performs better by 1.41% than FR-DSR, by 4.02% than FR-DSR in untrustworthy environment, by 0.7% than QAFR-DSR, by 2.99% than QAFR-DSR untrustworthy, by 2.32% than MOPSOFR-DSR in untrustworthy environment for no mobility. The throughput gets decreased when the node speed increases. For node speed 75 kmph, the throughput of MOPSOFR-DSR performs better by 1.47%

ISSN: 0976-3104

than FR-DSR, by 7% than FR-DSR in untrustworthy environment, by 0.63% than QAFR-DSR, by 6.15% than QAFR-DSR untrustworthy, by 5.59% than MOPSOFR-DSR in untrustworthy environment.

CONFLICT OF INTEREST

The authors declare no conflict of interests.

ACKNOWLEDGEMENT

None

FINANCIAL DISCLOSURE

The authors report no financial interests or potential conflicts of interest.

REFERENCES

[1] Johnson DB, Maltz DA. [1996] Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, T. Imielinski and H. Korth, Ed. *Kluwer Academic Publishers*, 5: 153-181.

[2] Ishibuchi H, Nozaki K, Yamamoto N and Tanaka, H. [1995] Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Trans Fuzzy Systems*, 3L: 260270

[3] Chen C. [2006] Design of PSO-based fuzzy classification systems. *Tamkang Journal of Science and Engineering*, 9(1): 63

[4] Chen H, Sun B, Zeng Y, He X. [2009] An entropy-based fuzzy controllers QoS routing algorithm in MANET. In Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on , *IEEE* 3: 235-239)..

[5] Geetha K., Thangaraj P. [2015] An Enhanced Associativity Based Routing with Fuzzy Based Trust to Mitigate Network Attacks.

[6] Srivastava S, Daniel AK. [2013] An Efficient Routing Protocol under Noisy Environment for Mobile Ad Hoc Networks using Fuzzy Logic. *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE*, 2(6).

[7] Gupta AK, Kumar R, Gupta NK. [2014] A trust based secure gateway selection and authentication scheme in MANET. In Contemporary Computing and Informatics (IC3I), 2014 International Conference on (pp. 1087-1093). *IEEE*.

[8] Dahiya R, Dureja A. [2014] Fuzzy Based Efficient Routing Protocol for Route Recovery In MANET. *International Journal Of Engineering And Computer Science* ISSN: 2319-7242 3(6).

[9] Kennedy J and Eberhart RC.[1995] Particle swarm optimization" , Proceeding of the 1995 IEEE International *Conference on Neural Networks (Perth, Australia), IEEE Service Centre, Piscataway*, NI, Iv: 1942-1948.

[10] Permana KE, Hashim SZM. [2010] Fuzzy membership function generation using particle swarm optimization. *Int. J Open Problems Compt* Math, 3(1), 27-41.

[11] Reyes-Sierra M, Coello CC. [2006] Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International journal of computational intelligence research, 2(3), 287-308.

[12] Alcalá R, Gacto MJ, Herrera F, Alcalá-Fdez J. [2007] A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(05): 539-557.

[13] HM Lee, CM Chen JM Chen and YL Jou.[2001] An efficient fuzzy classifier with feature selection based on fuzzy entropy, IEEE Transactions on Systems, Man, and Cybernetics — Part B: *Cybernetics* 31:3 :426–432.

[14] Wong CC, Wang HY, Li SA. [2008] PSO-based motion fuzzy controller design for mobile robots. *International Journal of fuzzy systems*, 10(1):24.

[15] Sun B, Gui C, Zhang Q, Chen H. [2009] Fuzzy controller based QoS routing algorithm with a multiclass scheme for MANET. *International Journal of Computers, Communications & Control*, 4(4): 427-438.