Journal of Autonomous Intelligence (2023) Volume 6 Issue 3
doi: 10.32629/jai.v6i3.705

COMMUNICATION

An improved firefly algorithm for the rank generation to optimize the route discovery process in IoV

Sumit Kumar*, Jaspreet Singh

Department of CSE, Chandigarh University Gharuan, Punjab 140413, India

*** Corresponding author:** Sumit Kumar, kumarsumit.cse@gmail.com

ABSTRACT

Vehicular ad hoc networks (VANET) have been the attention gainer for the last couple of years due to increasing number of vehicles on the road. Incorporation of VANET with Internet of Things (IoT) has created large number possibilities in terms of power efficiency and secure transmission. The article focuses on the ad-hoc on-demand distance vector (AODV) protocol and its applications in route discovery in VANETs. In this work, the swarm intelligence (SI) inspired modified firefly algorithm has been employed for rank generation of the nodes. It is concluded that the use of IoT devices and advanced routing protocols with SI algorithms can lead to efficient and low-latency route discovery in VANETs using quality of service (QoS) parameters. The experimental analysis shown that the proposed technique has been outperformed the other existing technique in terms of QoS parameters and provides the optimal route discovery mechanism with high throughput and minimum latency.

Keywords: vehicular ad hoc network (VANET); Internet of Things (IoT)-based VANET routing; ad-hoc on-demand distance vector (AODV) protocol; quality of service (QoS); swarm intelligence (SI); firefly algorithm (FA); ant colony optimization (ACO); particle swarm optimization (PSO); bee algorithm (BI)

ARTICLE INFO

Received: 26 June 2023
Accepted: 14 July 2023
Available online: 28 August 2023

1. Introduction

Vehicular ad hoc network (VANET) is a rapidly growing field in wireless communication that focuses on the development of communication networks for vehicles. These networks are designed to provide communication services for vehicles so that the overall performance of the network increases and the via sharing the resources in traffic over the roads. One of the critical tasks in a VANET network is route discovery, which refers to the process of finding the optimal path between a source and a destination in the network. Internet of Things (IoT)-based VANET routing refers to the use of IoT technology in routing data packets in a VANET environment. In a VANET, vehicles act as nodes and communicate with each other to exchange information such as traffic conditions, road conditions, and other relevant data. IoT devices, such as sensors and cameras, can be deployed in vehicles to collect and transmit data to other vehicles or to a central network. Routing in a VANET is a challenging task due to the dynamic and unpredictable nature of vehicular movements. In order to address the issues discussed earlier, several algorithmic architectures that refers to different types of routing have been designed and developed. As for example, based on source routing DSR, and based on ad-hoc positions, ad-hoc on-demand distance vector (AODV) and destination-sequenced distance-vector (DSDV) have been proposed in different scenarios of proactive and reactive routing architectures[1–2]. In IoT-based VANET routing, the collected data from IoT devices is utilized to make routing decisions. For example, the data collected from sensors can be used to predict traffic conditions and choose the most efficient route. Similarly, the data collected from cameras can be used to identify roadblocks or accidents and avoid them while routing[3]. This research article focuses on route discovery in VANET using the AODV protocol and its advancement from the collected quality of service (QoS) parameters. AODV is a reactive routing protocol that is widely used in VANET networks. The main goal of AODV is to provide efficient, low-latency route discovery and maintenance in ad-hoc networks. Unlike proactive routing protocols, such as OLSR, AODV only discovers routes when they are needed. This makes AODV an ideal choice for VANETs, where network topology changes frequently. AODV is based on the distance-vector algorithm and uses routing tables to store information about the network. The protocol uses a source node to initiate a route discovery process when it wants to send a data packet to a destination node. The source node broadcasts a route request (RREQ) message to its neighbours, which then forward the RREQ to their neighbours, and so on. When the RREQ reaches the destination node or a node that has a valid route to the destination, the node broadcasts a route reply (RREP) message back to the source node[3]. AODV is designed to work efficiently in VANETs, where network topology changes frequently due to the movement of vehicles. When a node wants to send data to a destination node, it first checks its routing table to see if it has a valid route to the destination. If it does, it sends the data directly to the destination. If it does not have a valid route, it initiates a route discovery process as shown in **Figure 1** as follows.

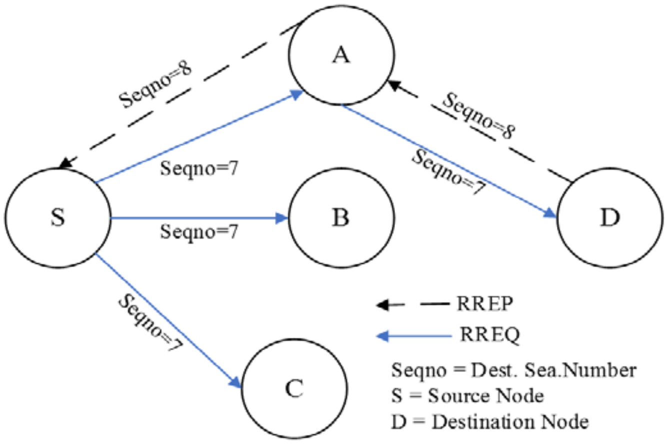

Figure 1. Source to terminal discovery process.

The source node broadcasts a RREQ message to its neighbours, which contain the source node's address, the destination node's address, and a unique identifier for the RREQ message. The RREQ message also includes the current sequence number for the source node, which is used to determine the most up-to-date route. When a node receives a RREQ message, it checks its routing table to see if it has a valid route to the destination. If it does, it sends a RREP message back to the source node with the route information. If it does not have a valid route, it forwards the RREQ to its neighbours. When the RREP message reaches the source node, the source node updates its routing table with the new route information and sends the data to the destination node. If the RREP message does not reach the source node within a specified time, the source node broadcasts a new RREQ message to try to find a valid route.

1.1. Contributions

Though a lot has been incorporated, the major contributions of the proposed work is as follows.

1) IoT based reactive routing protocol AODV has been introduced in order to obtain effective route discovery mechanism.

2) Design and development of improved FI for rank generation for the optimization of the route discovery mechanism based on reputation system for vehicles in Internet of Vehicle (IoV) environment.

3) Compare the proposed algorithm architecture with other state of art algorithms in terms of QoS parameters.

4) The results are tested and validated on MATLAB tool and proposed work showed superior results as discussed in section 4.

1.2. The AODV protocol

It uses the current knowledge of the nodes to distinguish between suitable and non-suitable path in VANET routing architecture. Once the route is discovered, the QoS parameters are evaluated. This paper illustrated an updated FI for the rank generation of the vehicle nodes that uses SI as ordinal behaviour analysis algorithm in the network. SI is a field of artificial intelligence that studies the collective behaviour of decentralized, self-organized systems, such as ant colonies and bee swarms. In recent years, SI has been applied to VANETs to optimize the route discovery process. SI algorithms, such as ant colony optimization (ACO), firefly algorithm (FA), particle swarm optimization (PSO) and bee algorithm (BA), are used to search for the optimal path by mimicking the behaviour of the corresponding natural systems. A literature survey of the use of SI in VANETs for route discovery optimization using QoS reveals several important findings. Firstly, ACO has been widely used in VANETs for route discovery optimization. ACO algorithms use a set of artificial ants that search for the optimal path by following the pheromones left by the other ants. The pheromones represent the quality of the path and are updated based on the QoS requirements of the applications[4]. Secondly, PSO and BA have also been applied to VANETs for route discovery optimization, but to a lesser extent compared to ACO. PSO algorithms use a set of particles that search for the optimal path by following the best particle in the swarm. BA algorithms use a set of bees that search for the optimal path by communicating with each other through a dance. Both PSO and BA have shown promising results for route discovery optimization in VANETs. Thirdly, many studies have focused on improving the performance of SI algorithms for route discovery optimization in VANETs by combining them with other techniques, such as genetic algorithms, artificial neural networks, and reinforcement learning. These studies have shown that the combination of SI and other techniques can improve the performance of route discovery optimization in terms of QoS metrics, such as latency, reliability and throughput[5].

The remaining paper is organised in the following manner. Section 2 represents the related work that is made in the context to the development of optimized routing algorithm with the usage of swarm intelligence and soft computation methods. Section 3 presents the proposed work algorithm that is divided into two phases. Section 4 provides the results of the implemented algorithm architecture as well the comparison to other state of art techniques that has implemented the algorithms in same scenario or sequence. The paper is concluded in section 5 and the future aspects of the current work is also provided in the same section.

2. Related work

The review of the technical papers in the field of VANETs focuses on the routing algorithms and optimization techniques used in these networks. The authors of these papers come from different backgrounds and employ a range of techniques and algorithms to address the challenges of VANET routing. Ali et al. focus on the use of machine learning techniques for secure vehicular communication in the IoV. The authors review the recent advances and applications of machine learning in this field, including the use of deep learning, reinforcement learning and evolutionary algorithms[1]. Magaia et al. propose a new routing scheme, “Group’n Route” (GnR), that leverages edge learning and social strength to provide efficient and effective communication in the IoV. This scheme uses clustering and efficient routing techniques to improve the performance of VANET communication[2]. Ding et al. present an improved version of the AODV routing protocol for VANETs. This protocol addresses the challenges of the standard AODV protocol and improves its performance in terms of reliability and efficiency[3]. Haerri et al. conduct a performance comparison of two popular routing protocols, AODV and OLSR, in urban VANET environments. The authors evaluate the performance of these protocols under realistic mobility patterns and find that OLSR outperforms AODV in terms of scalability and communication efficiency[4]. Ranjan Senapati and Mohan Khilar uses SI to optimize the performance parameters of the vehicular ad hoc network. The authors propose a swarm optimization algorithm to improve the network's performance in terms of communication efficiency, reliability and scalability[5]. Mouhcine et al. present a new routing strategy for VANETs that combines VANET with a distributed SI optimization. This strategy is designed to improve the routing performance of the network and provide effective communication in complex urban environments[6]. Sharma and Kaul propose a hybrid fuzzy multi-criteria decision making based multi-cluster head dolphin swarm optimized intrusion detection system (IDS) for VANETs. This IDS uses swarm intelligence and fuzzy logic to improve the security and reliability of VANET communication[7].

Joshua and Varadarajan present an optimization framework for routing protocols in VANETs using a multi-objective FA. This framework provides a systematic and efficient approach to routing optimization in VANETs and improves the performance of these networks[8]. Hamdi et al. propose a data dissemination scheme for VANETs that uses clustering and probabilistic forwarding based on an adaptive jumping multi-objective firefly optimization algorithm. This scheme improves the communication efficiency and reliability of the network[9]. Zehra et al. compare the performance of two optimization algorithms, the "artificial bees colony algorithm" (ABC) and the "FA" in the context of route selection processing time in VANETs. The authors find that the FA outperforms the artificial bees colony algorithm in terms of processing time and communication efficiency[10]. Sindhwani et al. provide a review of the soft computing techniques aware clustering-based routing protocols in VANETs[11]. Zandi et al. discussed the efficient route discovery and secure data transmission in VANET respectively. As discussed in **Table 1**, a comprehensive overview of the state-of-the-art techniques and algorithms used in this field and discuss the challenges and opportunities for further research[12,13].

Table 1. Summary of the related work.

Reference	Objective	Technique	Findings	Limitation
[1]	Secure IoV offloading, route discovery	ML and DRL	Reduced the latency and transmit data in a secure environment	Some suggested models are costly and complex in nature
[2]	Efficient data transmission over IoV, route discovery	Effective clustering data routing strategy GnR	Low computational time	In complex system bandwidth overhead, PDR, latency issue
[3]	Effective communication stability in VANET, route discovery	Improved AODV routing protocol	Higher link stability and PDR	In higher traffic scenario show high latency
[4]	Efficient route discovery	OLSR and AODV	Higher throughput, scalability	Data convergence issue
[5]	VANET communication and route discovery	SI, ACO, PSO	Minimize latency	Face issues with long distance data transmission with low PDR
[6]	IoV communication and optimal route discovery	ACO	High throughput, PDR	Bandwidth overhead
[7]	Data security and efficient route discovery in VANET	Fuzzy logic, SI based dolphin algorithm	Higher data security and intrusion detection rate, low latency	In overloaded network scenarios effects the performance of framework

Table 1. (*Continued*).

Reference	Objective	Technique	Findings	Limitation
[8]	VANET communication and route discovery	FA	Stable route link, low end-to-end delay	In dense network situations frequent network failure issue, higher latency and packet drop
[9]	Data transmission in VANET	Multi-objective FA	Higher PDR, throughput and low latency	In complex traffic scenarios the computational time increased
[10]	Route discovery and selection in VANET	ABC and FA	Low latency, improved network lifespan	Signal failure/drop issue in complex traffic scenarios
[11]	VANET communication and route discovery	Soft computing technique	Improved PDR, network lifetime	Higher cost of designed framework
[12]	Efficient route discovery	FA	Improved transmission time and network stability	Issue of packet drop during data transmission
[13]	Secure data transmission in VANET	ML based technique	Improved network security	In dense network scenario network lifetime effected

3. Proposed work

The proposed work analyze the energy efficient route discovery mechanism with help of reactive routing protocol[14]. The proposed work carried out an experimental analysis in which different parameters such as throughput, PDRr and latency has been evaluated in order to determine the efficiency of the discovered route. The proposed work is divided into two sections. The first section creates a route from source ned to terminal end by broadcasting the route requirements as illustrated in the introduction section. The detailed process is illustrated as follows.

AODV is a reactive routing protocol that only finds routes when they are needed and maintains them as long as they are required. In AODV, route discovery is initiated by a source node that wants to send data to a destination node. The route discovery process in AODV can be summarized using the following steps and equations:

1) Route request (RREQ) broadcast: A source node generates a RREQ message and broadcasts it to its neighbours. The RREQ message contains the source address (S), destination address (D), broadcast ID (B), and the current hop count (H). The broadcast ID is used to prevent loops in the route discovery process.

2) Route reply (RREP) generation: When a node receives a RREQ message, it checks if it has a valid route to the destination. If it does, it generates a RREP message and unicasts it back to the source node along the reverse path. The RREP message contains the source address (S), destination address (D), destination sequence number (D_{seq}) and the hop count (H).

3) Route maintenance: When a node receives a RREP message, it updates its routing table with the information contained in the RREP message. The node also marks the route as active and starts using it to forward data to the destination. If a node loses connectivity to the next hop on the route, it generates a RERR message and broadcasts it to its neighbours. The RERR message contains the address of the unreachable destination node. Once the route is discovered, the proposed work evaluates the following QoS parameters for the evaluation of the network.

3.1. Throughput

The throughput of a network refers to the amount of data that is successfully transmitted from the source to the destination in each time period. It can be represented mathematically as the ratio of the total number of bits transmitted to the total time required for the transmission, as follows:

$$Throughput\ (T) = (Data\ transmitted\ in\ bits)/(Time\ required\ for\ transmission\ in\ seconds) \quad (1)$$

3.2. Packet delivery ratio (PDR)

The throughput of a network refers to the amount of data that is successfully transmitted from the source to the destination in each time period. It can be represented mathematically as the ratio of the total number of bits transmitted to the total time required for the transmission.

3.3. Packet drop ratio (PDRr)

It signifies the packet not received by the destination.

$$PDRr = \frac{packet\ not\ delivered}{Total\ poackets} \tag{2}$$

3.4. Latency

Latency is a measure of the time it takes for a data packet to travel from the source to the destination. It is the difference between the time the packet is transmitted and the time it is received, as follows:

Latency = (Time of arrival of the packet at the destination) − (Time of transmission of the packet from the source).

Furthermore, the proposed work uses firefly algorithm for the rank generation of the nodes that will help the AODV discovery system for efficient routing. The overall work architecture can be presented by the following **Figure 2**.

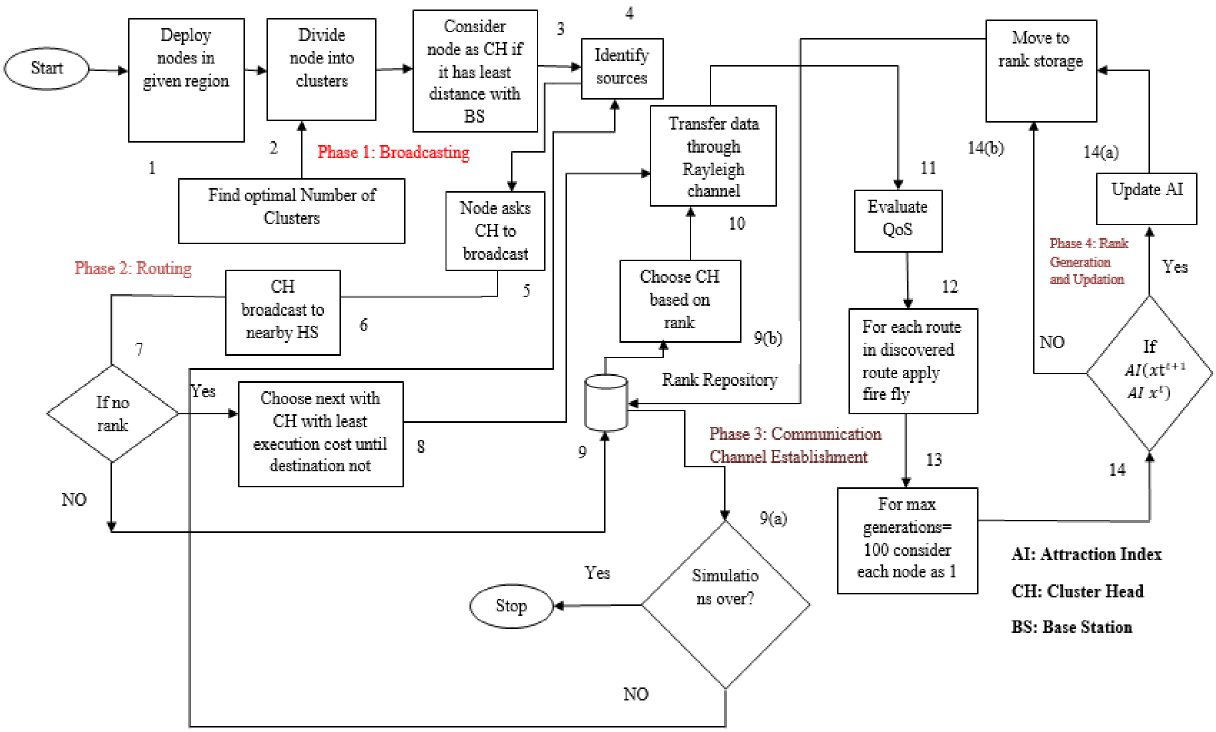

Figure 2. The proposed optimized route discovery mechanism.

Figure 2 represents the proposed methodology which is further discussed in stepwise that includes four phases as followed.

Phase 1: Broadcasting

1) Firstly, nodes are delayed in the specified region.
2) Classify the nodes into clusters and find the optimal cluster unit.
3) Depending upon the distance from the base station. The cluster nodes considered as required cluster head (CH).

4) Sources has been identified.
5) Assigned CH is broadcasted.

Phase 2: Routing

1) Broadcast the CH to neighbour CHs.
2) Analyse the CHs using define ranking system.
3) Choose the least cost CH in order to send data towards destination.
4) Finally transmit the data using the given channel.

Phase 3: Communication channel establishment

1) Evaluate the given QoS parameters.
2) Implement the proposed FA to the discovered route.

Phase 4: Rank generation and updating

1) Generate the rank against the nodes in the discovered route.
2) Based on criteria update the rank mentioned in **Figure 2**.
3) Stored the updated rank node and transmit the data through optimize route.

As shown in **Figure 2**, the network deploys nodes in a given region of deployment and the network is divided into multiple regions using Equation (1) as follows,

$$k = \log\left(\frac{n}{l} \times \frac{\sum_{i=1}^{n} d}{n}\right) \tag{3}$$

Firefly algorithm: The FA is a metaheuristic optimization algorithm inspired by the flashing behaviour of fireflies. It is used to solve optimization problems by mimicking the attraction mechanism between fireflies. The algorithm works by updating the position of fireflies based on the brightness of each firefly and the distance between them[15,16].

The brightness of a firefly represents its fitness value, and the attraction mechanism is modelled using the following equation:

$$Attraction\ intensity\ (\alpha) = \beta exp(-\gamma r\text{^}2) \tag{4}$$

where: α is the attraction intensity, β is the light absorption coefficient, γ is the light absorption rate, and r is the distance between two fireflies.

The movement of a firefly i is updated using the following equation:

$$X_i(t+1) = X_i(t) + \beta_0\, exp^{-\Upsilon r^2} \times (X_j(t) - X_i(t)) + \varepsilon \tag{5}$$

where: $X_i(t)$ is the position of firefly i at time t, $X_j(t)$ is the position of the brightest firefly at time t, β_0 is the initial light absorption coefficient, ε is a random term used to prevent premature convergence. The fireflies are iteratively updated until a stopping criterion is reached, such as a maximum number of iterations or a minimum change in the brightness of the fireflies. The final solution is the firefly with the highest brightness, which represents the optimal solution to the optimization problem. The FA works by mimicking the attraction mechanism between fireflies to optimize a given objective function[17,18]. The movement of fireflies is updated based on their brightness and the distance between them, and the algorithm terminates when a stopping criterion is reached. The final solution is the firefly with the highest brightness, which represents the optimal solution to the optimization problem. The proposed algorithm can be represented by the following Algorithm 1 architecture.

Algorithm 1 Optimize Swarm Intensity

INPUT:

1. current_firefly: current firefly being evaluated
   a. $\text{swarm}_{\text{intensity}}$: intensity of the other fireflies in the swarm
   b. alpha: randomization parameter
   c. beta: attractiveness parameter
   d. $\text{population}_{\text{ai}}$: population of fireflies in iteration i
   e. $\text{pairing}_{\text{swarms}}$: number of fireflies in the swarm
2. **OUTPUT:**
   a. $\text{population}_{\text{ai}}$: updated population of fireflies in iteration i+1
3. **PROCEDURE**:
4. For each firefly ai in the swarm, repeat steps 2 to 7:
5. Calculate the average of throughput and PDR for the current firefly and the ith firefly in the swarm:
6. x_ai = mean(current_firefly(1:2)) − mean(current_firefly(3:5));
7. x_aj = mean(swarm_intensity(ai, 1:2)) − mean(swarm_intensity(ai, 3:5));
8. If ai >$\text{population}_{\text{ai}}$, calculate the difference between the populations of the two previous iterations:
9. gaama = (population_ai(ai − 1) − population_ai(ai−2));
10. 4.Update the population of the current firefly using the formula:

$$\text{x}_{\text{ai}_{\text{nt}}} = \text{x}_{\text{ai}} + \text{beta} \times \exp\left(-\text{gaama} \times \left(\text{x}_{\text{ai}} - \text{x}_{\text{aj}}\right)^2\right) \times \left(\text{x}_{\text{ai}} - \text{x}_{\text{aj}}\right) + \text{alpha} \tag{6}$$

11. 5.Store the updated population in the population array:
12. $\text{population}_{\text{ai(ai)}} = \text{x}_{\text{ai}_{\text{nt}}}$;
13. 6.Repeat steps 2 to 5 for each firefly in the swarm.
14. 7.Return the updated $\text{population}_{\text{ai}}$ as the output.

The Algorithm 1 returns the SI intensity for every root. Each node is managed with a rank repository that initially holds a value 0. Every time a root shares gets an AI value; it is divided into every node that is associated in the root other than the source and destination node. All the attained ranks are sorted in the end. Highest rank holder vehicle will be preferred on the first chance for the selection prior to any other vehicle when the broadcast is made as shown in **Figure 2**. The vehicle ranking can be justified using the following Algorithm 2.

Algorithm 2 Return SI Intensity

Input: AI value for each root, number of nodes in each root
Output: SI intensity for each node
Step 1: Initialize a rank repository for each node with value 0
Step 2: For each root, do the following:
a. Share the AI value to every node associated with the root except the source and destination node.
b. Divide the AI value among the nodes as follows:
i. Calculate the average of the AI value for the node as:

$$vg = \frac{\sum_{i=1}^{n} AI_i}{n} \tag{7}$$

where n is the number of nodes in the root
ii. Update the rank of the node in the rank repository as:
rank of node i = avg/total AI value of all nodes in the root
c. Store the attained rank for each node in the rank repository.
Step 3: Sort the ranks in the rank repository in descending order.
Step 4: The node with the highest rank will be preferred for selection first when the broadcast is made.
Step 5: Return the SI intensity for each root.
Example:
Consider a scenario where a root has 4 nodes, with the AI value of [20, 30, 40, 50].
-The average of the AI value for each node will be (20 + 30 + 40 + 50)/4 = 35.
-The rank of node 1 will be 20/140 = 0.142857.
-The rank of node 2 will be 30/140 = 0.214285.
-The rank of node 3 will be 40/140 = 0.285714.
-The rank of node 4 will be 50/140 = 0.357143.
-The ranks of all nodes

All nodes in the root will be sorted in descending order, and the node with the highest rank, i.e., node 4, will be preferred first when the broadcast is made.

The proposed algorithm helps in selecting out the best possible nodes based on the rank of the node and performs better in terms of QoS analysis. The evaluation and the discussion is provided in the next section.

4. Results and discussion

The proposed work has been evaluated based on throughput, latency and packet drop ratio (PDRr). The following equations have been utilized to compute the QoS parameters.

$$Throughput = \frac{Rb}{T} \tag{8}$$

$$Latency = \sum_{i=1}^{T} Et \tag{9}$$

$$PDRr = \frac{Rbn}{Sb} \tag{10}$$

where Rb is the received bits, Rbn is the bits not received, sb is the sent bits, Et is the elapsed extra time, T is total time interval. To illustrate the parameters, the proposed algorithm and other set of algorithms were utilized under same simulation environment (**Table 2**).

Table 2. Simulation environment.

OS used	Window 10 64 bit
Processor	Intel core i3
RAM	8 GB
Number of vehicle nodes	250
Tool	MATLAB
Evaluation parameters	Throughput, packet drop rate, latency

The experimental analysis has been conducted by comparing the different parameters such as throughput, PDRr and latency. The proposed technique is compared with Joshua and Varadarajan[8] and Magaia et al.[2] using different QoS parameters in order to evaluate the efficacy of the proposed work.

Table 3 represents the throughput analysis in which the proposed technique is compared with the techniques implemented by Joshua and Varadarajan[8] and Magaia et al.[2]. The proposed technique shown significant improvement in terms of throughput as compared to other techniques. The average throughput (bits/sec) value shown by Joshua and Varadarajan[8] and Magaia et al.[2] and proposed are 5993.909091, 5932.181818 and 6449.545455. The proposed technique shown highest average throughput value, i.e., 6449.545455 as compared to other techniques.

Table 3. Proposed evaluation using throughput.

No of vehicle nodes	Throughput proposed (bits/sec)	Throughput Joshua and Varadarajan[8]	Throughput Magaia et al.[2]
50	6536	6459	5964
70	6378	5886	6090
90	6548	6409	6252
110	6488	5944	6292
130	6204	5895	6179
150	6325	6245	5678
170	6649	6242	5790
190	6307	5382	5631
210	6407	5995	5973
230	6554	5812	5693
250	6549	5664	5712

In **Figure 3**, the throughput analysis has been carried out against the number of vehicle nodes. Total 250 vehicle nodes have been used during the experimental analysis. The proposed technique outperformed the other techniques mentioned in Joshua and Varadarajan[8] and Magaia et al.[2]. The significant percentage improvement in throughput value has been shown using proposed technique, i.e., 7.61% over Joshua and Varadarajan[8] and 8.70% over Magaia et al.[2].

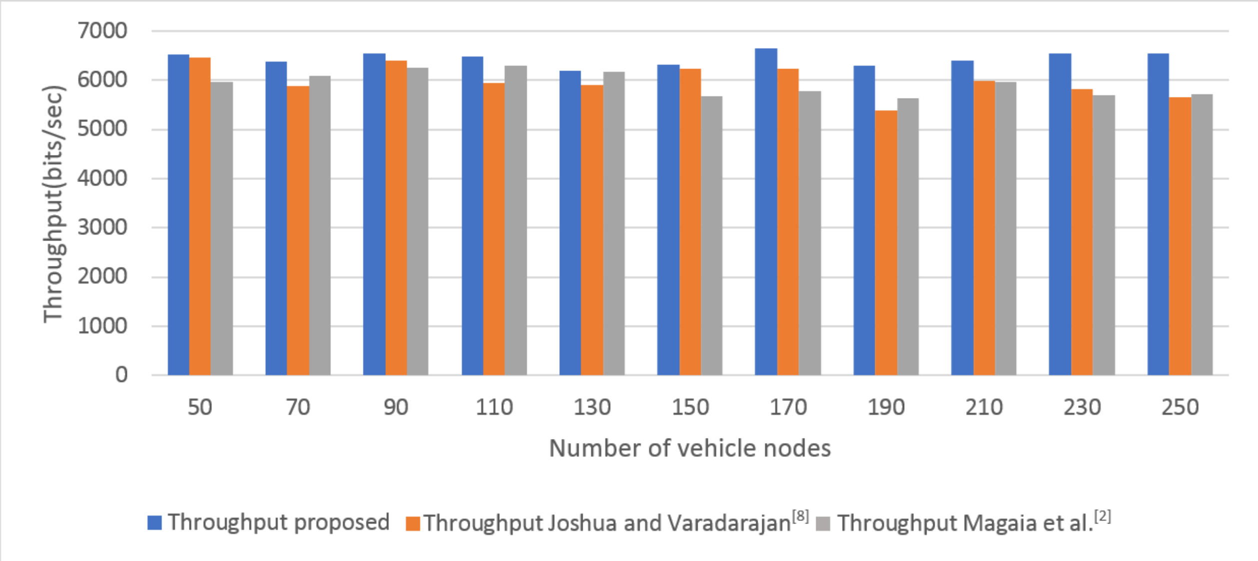

Figure 3. Throughput analysis.

Table 4 represents the PDRr analysis in which the proposed technique is compared with the techniques implemented by Joshua and Varadarajan[8] and Magaia et al.[2]. The proposed technique shown significant reduction in terms of PDRr as compared to other techniques. The average value shown by Joshua and Varadarajan[8] and Magaia et al.[2] and proposed are 0.043217153, 0.044725458 and 0.040603076. The proposed technique shown lowest average PDRr value, i.e., 0.040603076 as compared to other techniques.

Table 4. Performance evaluation using PDRr.

No of vehicle nodes	PDRr proposed	PDRr Joshua and Varadarajan[8]	PDRr Magaia et al.[2]
50	0.08483148	0.08858668	0.09358212
70	0.1153518	0.11830085	0.12784173
90	0.03032894	0.03184962	0.03474271
110	0.02566769	0.02779475	0.0258969
130	0.00484865	0.0049758	0.00546246
150	0.04021546	0.04616234	0.04374363
170	0.03	0.03148747	0.03376862
190	0.03	0.03435085	0.03258741
210	0.03	0.03311841	0.03414057
230	0.02538982	0.02748012	0.0290588
250	0.03	0.03128179	0.03115509

In **Figure 4**, the PDRr analysis has been carried out against the number of vehicle nodes. Total 250 vehicle nodes have been used during the experimental analysis. The proposed technique outperformed the other techniques mentioned in Joshua and Varadarajan[8] and Magaia et al.[2]. The significant percentage reduction in PDRr value has been shown using proposed technique, i.e., 5.62% from Joshua and Varadarajan[8] and 9.24% from Magaia et al.[2].

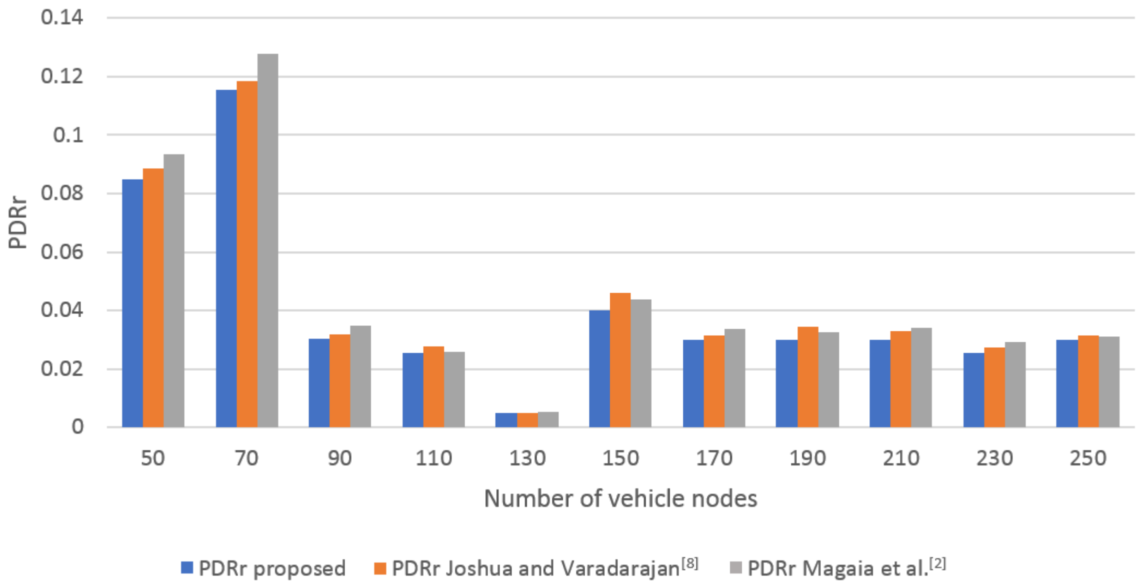

Figure 4. PDRr analysis.

Table 5 represents the latency analysis in which the proposed technique is compared with the techniques implemented by Joshua and Varadarajan[8] and Magaia et al.[2]. The proposed technique shown significant reduction in terms of latency as compared to other techniques. The average latency value shown by Joshua and Varadarajan[8] and Magaia et al.[2] and proposed are 9.519121787, 9.135702627 and 8.24726791. The proposed technique shown least average latency value, i.e., 8.24726791 as compared to other techniques.

Table 5. Proposed evaluation using latency.

No of vehicle nodes	Latency proposed (sec)	Latency Joshua and Varadarajan[8] (sec)	Latency Magaia et al.[2] (sec)
50	3.70393177	4.12635584	3.95394778
70	3.65679635	4.56120744	4.29461084
90	5.31897255	5.84689848	6.27058139
110	6.08171655	6.67726302	6.37139241
130	7.56521731	8.99776901	7.57429293
150	8.02148029	9.92948137	9.53179065
170	9.29024289	11.0330697	10.3939587
190	10.3570433	12.1118	11.9147096
210	11.1893689	11.6121076	12.2444725
230	11.7018713	13.1165083	13.2723848
250	13.8333058	16.6978789	14.6705873

In **Figure 5**, the latency analysis has been carried out against the number of vehicle nodes. Total 250 vehicle nodes have been used during the experimental analysis. The proposed technique outperformed the other techniques mentioned by Joshua and Varadarajan[8] and Magaia et al.[2]. The significant percentage reduction in latency value has been shown using proposed technique, i.e., 15.43% from Joshua and Varadarajan[8] and 10.75% from Magaia et al.[2].

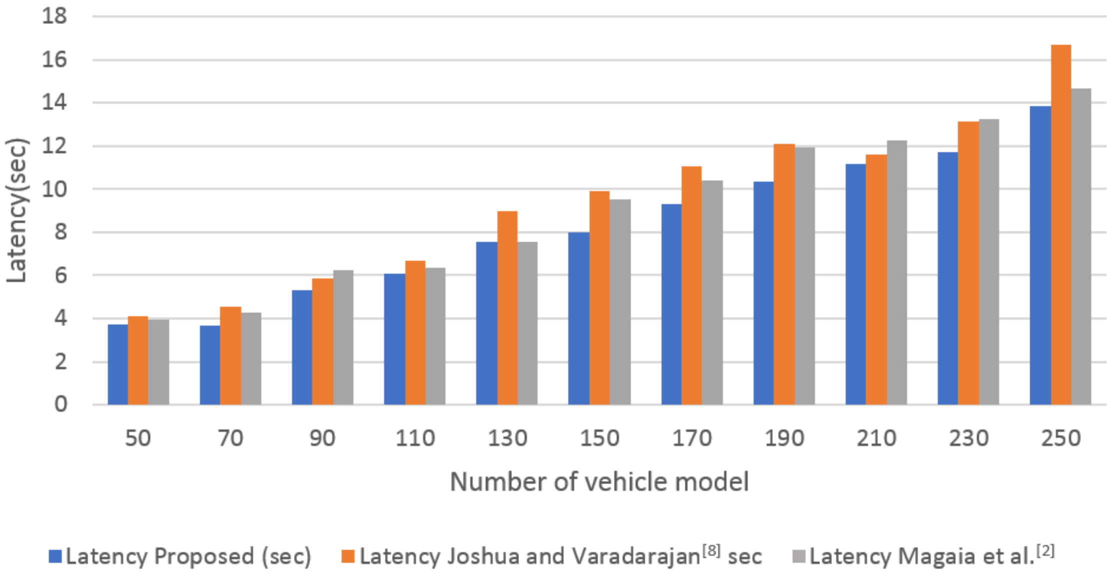

Figure 5. Latency analysis.

5. Conclusion

The establishment of IoT based communication infrastructure for automobiles is the primary objective of rapidly expanding wireless communication area termed as VANET. In order to improve the overall network performance and provide optimal communication capabilities to the moving vehicles an effective utilization of resources is required by finding the optimal route discovery in order to transmit the packet efficiently. The determining the best route between a source and a destination inside the network, is one of the crucial responsibilities in a VANET network. The proposed work provides the potential solution in order to determine the energy efficient route discovery. The experimental results showed the efficacy of the proposed work as compared to other techniques discussed by Joshua and Varadarajan[8] and Magaia et al.[2] in terms of QoS parameters such as throughput, PDRr and Latency. The proposed technique shown significant percentage improvement in throughput—7.61% over Joshua and Varadarajan[8] and 8.70% over Magaia et al.[2], PDRr—5.62% from Joshua and Varadarajan[8] and 9.24% from Magaia et al.[2], and latency—15.43% from Joshua and Varadarajan[8] and 10.75% from Magaia et al.[2]. In future, SI and deep learning strategies can be incorporated in order to improve the performance of the QoS parameters.

Conflict of interest

The authors declare no conflict of interest.

References

1. Ali ES, Hasan KM, Hassan R, et al. Machine learning technologies for secure vehicular communication in Internet of Vehicles: Recent advances and applications. *Security and Communication Networks* 2021; 2021: 8868355. doi: 10.1155/2021/8868355
2. Magaia N, Ferreira P, Pereira PR, et al. Group'n route: An edge learning-based clustering and efficient routing scheme leveraging social strength for the Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 2022; 23(10): 19589–19601. doi: 10.1109/TITS.2022.3171978
3. Ding B, Chen Z, Wang Y, Yu H. An improved AODV routing protocol for VANETs. In: Proceedings of 2011 International Conference on Wireless Communications and Signal Processing (WCSP); 9–11 November 2011; Nanjing, China. pp. 1–5.
4. Haerri J, Filali F, Bonnet C. Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns. In: Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop; 14–17 June 2006; Lipari, Italy.

5. Ranjan Senapati B, Mohan Khilar P. Optimization of performance parameter for vehicular ad-hoc network (VANET) using swarm intelligence. In: *Nature Inspired Computing for Data Science*. Springer, Cham; 2019. pp. 83–107.
6. Mouhcine E, Mansouri K, Mohamed Y. Solving traffic routing system using VANet strategy combined with a distributed swarm intelligence optimization. *Journal of Computer Science* 2018; 14(11): 1499–1511. doi: 10.3844/jcssp.2018.1499.1511
7. Sharma S, Kaul A. Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET. *Vehicular Communications* 2018; 12: 23–38. doi: 10.1016/j.vehcom.2017.12.003
8. Joshua CJ, Varadarajan V. An optimization framework for routing protocols in VANETs: A multi-objective firefly algorithm approach. *Wireless Networks* 2021; 27: 5567–5576. doi: 10.1007/s11276-019-02072-w
9. Hamdi MM, Audah L, Rashid SA. Data dissemination in VANETs using clustering and probabilistic forwarding based on adaptive jumping multi-objective firefly optimization. *IEEE Access* 2022; 10: 14624–14642. doi: 10.1109/ACCESS.2022.3147498
10. Zehra SS, Mustafa SMN, Qureshi R. Comparing artificial bees colony algorithm and firefly algorithm to achieve optimization in route selection processing time in VANETs. *Pakistan Journal of Engineering and Technology* 2021; 4(2): 159–164. doi: 10.51846/vol4iss2pp159-164
11. Sindhwani M, Sachdeva S, Arora K, et al. Soft computing techniques aware clustering-based routing protocols in vehicular ad hoc network: A review. *Applied Sciences* 2022; 12(15): 7922. doi: 10.3390/app12157922
12. Zandi M, Jahanshahi M, Hedayati AR. Routing optimization in vehicular social networks using firefly algorithm. *International Journal of Industrial Mathematics* 2021; 13(4): 361–369.
13. Hamdi MM, Yussen YA, Mustafa AS. Integrity and authentications for service security in vehicular ad hoc networks (VANETs): A review. In: Proceedings of 2021 3rd International Congress on Human-Computer Interaction Optimization and Robotic Applications (HORA); 11–13 June 2021; Ankara, Turkey. pp. 1–7.
14. Shrivastava PK, Vishwamitra LK. Comparative analysis of proactive and reactive routing protocols in VANET environment. *Measurement: Sensors* 2021; 16: 100051. doi: 10.1016/j.measen.2021.100051
15. Fister I, Yang XS, Fister D, Fister I. Firefly algorithm: A brief review of the expanding literature. In: *Studies in Computational Intelligence*. Springer, Cham; 2013. pp. 347–360.
16. Kalra M, Singh S. A review of metaheuristic scheduling techniques in cloud computing. *Egyptian Informatics Journal* 2015; 16(3): 275–295. doi: 10.1016/J.EIJ.2015.07.001
17. Bothra SK, Singhal S. Nature-inspired metaheuristic scheduling algorithms in cloud: A systematic review (Slovenian). *Scientific and Technical Journal of Information Technologies, Mechanics and Optics* 2021; 21(4): 463–472. doi: 10.17586/2226-1494-2021-21-4-463-472
18. Wang Z, Liu D, Jolfaei A. Resource allocation solution for sensor networks using improved chaotic firefly algorithm in IoT environment. *Computer Communications* 2020; 156: 91–100. doi: 10.1016/j.comcom.2020.03.039