A New Fog-Based Routing Strategy (FBRS) for Vehicular Ad-Hoc Networks

Khaled S. El Gayyar (✉ khaledsamy00@gmail.com)

Delta University for Science and Technology <https://orcid.org/0000-0003-0263-5926>

Ahmed I. Saleh

Mansoura University

Labib M. Labib

Mansoura University

Research Article

Keywords: VANET, Routing, Fog Computing, Cloud, Dijkstra, FBRS

Posted Date: February 9th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-200216/v2>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Peer-to-Peer Networking and Applications on October 9th, 2021. See the published version at <https://doi.org/10.1007/s12083-021-01197-0>.

A New Fog-Based Routing Strategy (FBRs) for Vehicular Ad-Hoc Networks

Khaled S. El Gayyar^{a,1*}, Ahmed I. Saleh^b and Labib M. Labib^b

^a *Communication Engineering Department, Faculty of Engineering, Delta University, P.O. Box: +11152, Gamasa, Egypt*

^b *Faculty of Engineering, Computers Engineering and Systems Department, Mansoura University, P.O. Box: +35516, Mansoura, Egypt*

Abstract. Vehicular ad-hoc network (VANET) plays a significant role in future intelligent transportation systems. The main objective of vehicular ad hoc networks (VANETs) is to improve driver safety and traffic efficiency. Many researchers proposed different schemes to improve communication efficiency. It is quite challenging where vehicles' speed, Direction, and density of neighbors on the move are not consistent. Although several routing protocols have been introduced to manage data exchange among vehicles in VANETS, they still suffer from many drawbacks such as lost packets or time penalties. This paper introduced a new Fog Based Routing Strategy, which constructs a reliable system of adaptive, stable, and efficient routing networks. FBRs consists of two main phases: System Setup Phase (SSP) and System Operation Phase (SOP). SSP creates a cluster network, collects its nodes' data, mining routes between them, and ranking paths using Dijkstra's algorithm into a simplified table. Although, SOP generates a reliable route between the request of any two nodes for a communication channel and maintains the route against any simultaneous crashes. Recent VANET routing protocols have been compared against FBRs. Experimental results have proven the outperforming of the proposed algorithm against recent routing protocols in terms of packet delivery ratio and routing overhead.

Keywords: VANET, Routing, Fog Computing, Cloud, Dijkstra, FBRs

1. Introduction

Vehicular ad hoc networks (VANETs) are innovative technology in transportation systems and smart cities [1-3]. It is a critical component of intelligent transportation systems to improve vehicular traffic systems' efficiency and safety [4]. VANETs implementation can be executed effectively by exchanging data between vehicles using multiple hops. Even so, considering the essential characteristics of VANET networks, as its limited data delivery delay, infrequent connectivity, and highly dynamic network topology. The most susceptible challenge is remaining the participated node in the network for just a limited time. Therefore, to prevent data loss, a stable routing technique should be implemented to fulfill VANET

applications and handle intermittent connectivity and dynamic topology changes.

VANETs are considered an exceptional case of Mobile Ad hoc Networks (MANETs) except that mobile nodes are vehicles [5]. The distinctive characteristics of VANET are the high mobility speed, but it is restricted by following organized routes and traffic laws. Furthermore, in VANET, vehicle communication channels are not only generated with other vehicles (V2V) but also with Base Stations (BS) and Road-Side Units (RSUs). The main target of VANET is to deliver specific data efficiently with the minimum delay possible in time. It supports many critical services such as traffic propagation alerts, sudden accidents, unusual traffic movements, route planning and control services, V2V communications, and cloud services [6]. Many

*Corresponding author.

*Khaled S. El Gayyar. E-mail: KhaledSamy00@Gmail.com.

Ahmed I. Saleh. E-mail: AISaleh@Yahoo.com.

Labib M. Labib. E-mail: LabibMLabib@Gmail.com.

Tel.: +20 109 502 7037.

Tel.: +20 111 409 0654.

Tel.: +20 115 689 4550.

route planning and control services are already implemented in some countries [7, 8]. Yet, many VANET systems' challenges are still struggling in both deployment and management [9, 10]. With the current transportation systems, more challenges remain unsolvable such as geographical issues, unstable traffic flow, multiplying numbers of vehicles in the network, very high mobility traffic, communication channel crashes, and lack of security, which affect the quality of service (QoS) [11]. Although, self-organized distributed network styles such as MANET can be applied to "fill the gaps" in these scenarios. However, the lack of a dedicated resource and communication management mechanism makes some services unenforceable. There is a need to suggest new network structures to deal with these issues.

A large amount of data between vehicles can be shared and distributed by Users. The delay in transferring data and information from vehicles to the remote cloud server and back to the vehicles after storage and processing is a critical area of concern in cloud computing. With the massive increase in the number of connected vehicles and their ever-increasing transportation, the demand for applications that support low latency and uninterrupted services is increasing day by day.

Fog computing is a model that spreads cloud computing and services to the edge of a network [12-14]. It supplies an intermediate haze layer between Cloud and mobile devices. Unlike the Cloud, which is more central, Fog computing aims for services with widespread distribution. With a local feature, low latency, connection, and context awareness in Fog computing can be raised. Like Cloud, Fog provides data computing, storage, and application services to end-users [15]. Fog computing as a new architecture is the right candidate for VANETs to meet requirements, such as rapid reaction on the base device, reduced cloud load, real-time data flow analysis using the Cloud, and better quality of service. Any device equipped with computing, storage, and network connectivity can be a haze node, including portable vehicles and road infrastructure [16]. In VANETs, Fog nodes closest to the network's edge absorb data from different devices, and developers decide how to use them.

Therefore, this paper proposes a routing algorithm called Fog-Based Routing Strategy (FBRS) for Vehicular Ad-Hoc Network. FBRS consists of two main phases: System Setup Phase (SSP) and System Operation Phase (SOP). SSP creates a cluster network, collects its nodes' data, mining routes between them, and ranking paths using Dijkstra's algorithm [17-19] into a simplified table. Fog server collects clusters data and

their Route Tables (RT) to be ready for communication requests. Although, SOP generates a reliable route between the request of any two nodes for a communication channel and maintains the path against any simultaneous crashes. Using Fog computing servers to collect, compare, and analyze data has improved routing speed and efficiency.

The main contributions of the proposed model are as follows:

- The proposed model is a hybrid proactive and reactive routing approach for VANETs that gains both advantages as it works as a proactive routing system (table-driven) at SSP. Also, a reactive routing approach at SOP.
- Using Cluster Formation Protocol (CFP) in SSP, which not only can identify each node in the cluster, collect their data and inner relations, but it also can identify neighboring nodes from other clusters. It also identifies orphan nodes with no cluster to belong to and creates a subcluster for them. These cases were mentioned barely in a few papers. CFP is capable of handling these exceptional cases.
- In mining the data and the inner relations, we used Dijkstra's algorithm to optimize the routing table in a much simpler and faster form. It handles the proactive approach's primary disadvantage, which consumes time to build and mine all network routes. It depends on handshaking time, direction, average speed, and acceleration to calculate the route's reliability factor, not only the distance between nodes.
- Furthermore, Depending on Fog computing and Fog servers to handle data computing and transferring which aims for services with widespread distribution. Unlike the Cloud, which is more central, Fog is an intelligent gateway that offloads clouds enabling more efficient data storage, processing, and analysis.
- Moreover, the delay of the communication channel requesting is reduced mainly by using Real-time Rerouting Maintenance (R^2M), which maintains crashed paths and finds a new one without calling back to the sender to search for an alternative route.

This paper's organization is as follows: Section 2 shows the related works, explains Fog computing's essential characteristics as a new computing paradigm, and illustrates a quick survey of VANET routing protocols. Section 3 depicts the proposed Fog-Based Routing Strategy (FBRS). Section 4 presents the experimental results, and section 5 presents the

conclusion of the observed experimental results. Finally, section 6 is the references.

2. Related works

2.1. Fog computing

Fog computing is a highly virtualized platform that provides computing, storage, and networking services between end devices and traditional Cloud Computing Data Centers. In Fig. 1, a description of the ideal information and computing architecture that provides future IoT applications and illustrates Fog computing's role. However, Fog Computing implies several characteristics described below, which plays a significant role in VANET information exchange as an extension of the Cloud.

Fig. 1 Network Routing Layers.

- Geographical distribution. In sharp contrast to the most central Cloud, the applications and services targeted by the Fog demand widely distributed deployments. For example, the Fog will play an active role in providing a high-quality flow of moving vehicles through agents and access points placed along highways and tracks.
- Extensive environmental monitoring and smart grid sensor networks are other examples of inherently distributed systems, which require distributed computing and storage resources.
- As a result of the wide geographical distribution, a large number of nodes, as evidenced by the sensor networks in general and the smart grid in particular.
- Mobility support. Many Fog applications are necessary to communicate directly with mobile devices, thus supporting mobility techniques
- Real-time interactions. Critical Fog applications include real-time interactions rather than batch processing, which support online analytic and interplay with the Cloud. Fog plays a significant role

in the ingestion and processing of the data close to the source.

The publication of Connected Vehicles offers a rich scenario of communication and interactions [20]: Vehicle to vehicle (V2V), vehicle to infrastructure (V2I), and infrastructure to infrastructure (I2I). Fog has many features that make it the ideal platform for providing productive, existing services in the field of information, entertainment, safety, and traffic support and analysis: geographical distribution (across cities and along roads), site mobility and awareness, low latency, heterogeneity, and support for real-time interaction.

The publication of Connected Vehicles offers a rich scenario of communication and interactions [20]: Vehicle to vehicle (V2V), vehicle to infrastructure (V2I), and infrastructure to infrastructure (I2I). Fog has many features that make it the ideal platform for providing productive, existing services in the field of information, entertainment, safety, and traffic support and analysis: geographical distribution (across cities and along roads), site mobility and awareness, low latency, heterogeneity, and support for real-time interaction.

2.2. VANET routing protocols

MANETs and VANET are similar in their ad hoc nature, unpredictable communication links, and low bandwidth. Thus, many studies have modified the routing protocols of MANETs to adopt VANETs [21]. VANETs is a unique type of MANETs and offers various features, such as frequent network disconnection, dynamic topological changes, narrow delay limits, no problems with power limitation, geo-connectivity, and specific mobility modeling after a road topology. Path stability and efficiency are the main design metrics for VANET routing protocols.

2.2.1. Topology-based routing protocols

These routing protocols use the critical information existing in the network to forward the packets. It can also be divided into proactive (table-driven) and interactive (on-demand) orientation. Even proactive routing has a distinct advantage: routing information such as post-error forwarding is corrected as packets gradually approach the destination. As such, interactive routing opens a path only when necessary for the node to communicate with another node. It only maintains methods that can reduce the burden on the network. Reactive paths usually include a path detection stage where query packets are flooded in the network.

2.2.2. Geographic-based routing protocols

It is also called position-based routing protocols that depend on forwarding decision-making made by a source node on the destination node's geographic position and the next one-hop node position. It is accomplished by storing the destination's position in the packet's header by the source. Hence, all nodes' positions in the network must be obtained periodically using beacon messages to prevent route crashes. In geographic routing protocols, all nodes in the same network are assumed to know their location using an onboard Navigation System and a Global Position System (GPS) unit. Although these protocols do not maintain crashed paths or exchange information about route status, they provide ready routes immediately, which decreases the delay time and more suitable for highly dynamic environments like VANETs.

3. The proposed Fog-Based Routing Strategy (FBRs)

This paper introduced a new Fog Based Routing Strategy (FBRs) for VANETS. It constructs a reliable system of adaptive, stable, and efficient routing network using static infrastructure - Road Side Units (RSUs) - in crowd areas. It is powered by Fog-Computing technology to build non-overlapped grouped clusters, each cluster divided into levels as well as on fast roads supervised by the central cloud server.

FBRs consists of two main phases, as described in Fig. 2: System Setup Phase (SSP) and System Operation Phase (SOP). Each phase is divided into two or three steps.

Fig. 2 Proposed Fog-Based Routing Strategy (FBRs).

In SSP, steps are the following:

- (i) Network Setup: using Cluster Formation Protocol (CFP) as it defines cluster nodes, their level, relations with each other and gathers routes and data between all nodes back to Cluster Head (CH). These data are set in Data Table (DT).

- (ii) Mining and Ranking: Mining using DT creates an Undirected Cyclic Graph (UCG) of routes. Each node's data parameters define each relation's reliability in the cluster network to build Relation Reliability (RR) Table and send RR Table to all nodes in this cluster. Then ranking using Dijkstra's algorithm which has not often been used in VANET routing algorithms.

- (iii) Route Table Formulation & Delivery, each node builds its own Route Table (RT) – depend on the reliability factor of routes - to all neighbor nodes in the same cluster.

In SOP, steps are the following:

- (i) Global Routing Operation (GRO): any node that needs to establish a connection must run through two operations. First, in Route Discovery Operation (RDO), it searches for the receiver as if it is in the same cluster, neighbored cluster (same Fog server), or in a faraway cluster (same cloud server). Second, Reliable Route Selection (RRS): creates a single reliable route between sender and receiver nodes. The second and final step,
- (ii) Real-time Route Maintenance (R^2M). If there was any change in path while transmitting the message or any crashes in route connection, it initiates the Dynamic Rerouting Operation (DRO). DRO maintains the crashed paths and finds a new one without calling back to the sender to search for an alternative route.

3.1. System Setup Phase

System Setup Phase is divided into three primary operations: Network Setup, Mining and Ranking, and Route Table Formulation and Delivery.

3.1.1. Network Setup

In this subsection, an illustrated example describes how to create a cluster network and collect its nodes' data. CFP is a periodical Cluster formation and routes discovery, in which the primary function of the protocol is to: divide the cluster into levels as in Fig. 3, discover routes between CH and all other nodes (vehicles), define new nodes into the cluster, initialize nodes level. At the start of constructing the cluster, a maximum number of cluster levels (L_M) must be initiated. L_M value depends on the size of the cluster and the number of nodes that hold. Then Data Gathering Protocol (DGP) gathers specific information about every node (node level, handshaking time, direction, speed, and acceleration) in Collected Data Flags (CDF) and its relation with neighbored nodes in the

Fig. 3 Cluster divided into level example.

same cluster. CFP is a proactive topology-based routing protocol. Therefore, the network service areas are split up into nonoverlapping clusters.

So, nodes of a cluster can be concerned as (i) Cluster Head (CH), (ii) Normal Node (NN), (iii) Gateway Node (GN), and (iv) Sub Cluster Node (SCH). CH can be considered as the central processing unit of the cluster. It initiates and broadcasts a Route Discovery Message (RDM) which its flags are described in Fig. 4. RDMs identify all current active nodes in the cluster's range and gather their information and routes along to CH. It may be a road-side unit (RSU) as a traffic light, a bus stop, or a standalone RSU tower.

Fig. 4 route discovery message Flags.

In CFP, the CH initiates Route Discovery Message (RDM). It is assuming that Route Discovery Message Level (M_L) = 0 and message direction = F where F: Forward and B: Backward. Then, the CH starts to broadcast RDM every λ of time to collect cluster nodes' Identity Code (N_{ID}), their data, and relations (routes) with other nodes in the cluster. Nodes receive RDM. There are two possibilities that if the node does not have a N_{ID} (new node) or has its own N_{ID} (old node). If it is a new or unidentified N_{ID} (has N_{ID} from a different cluster), RDM force it to generate a new N_{ID} and copy it into the RDM sequence flag, set Node

Level (N_L) equal to M_L , then increment M_L by one, and rebroadcast. RDM sequence flag holds a list/sequence of node addresses accumulated as the RDM propagates to be used as source routing. On the other hand, if the receiver node has its own N_{ID} of the same cluster, it has one of three cases:

(i) If $N_L > M_L$: Check if $M_L \leq L_M$ (maximum number of cluster levels). If so, then RDM reached a normal node in the cluster. It copies its N_{ID} to the RDM route sequence, increment M_L by one, and rebroadcast again. If not ($M_L > L_M$), then check if the receiver node has no N_{ID} . If so, then it means that RDM reached a node with no cluster to hold. In that case, RDM generates a Sub-Cluster Head ID (SCH_{ID}) for it and sets its N_L to zero, and increment M_L by one. RDM broadcasts again to find any more nodes that do not have a CH either. If any node has been found, it joins to a sub-cluster of the neighboring static cluster. It will also be a gateway to the main cluster. Moreover, any discovered nodes in the sub-cluster will be treated as normal nodes and follow the same routing protocol. If not (the receiver node has N_{ID} from a different cluster), it copies its N_{ID} to the RDM route sequence as a gateway to this cluster and broadcast back to CH.

(ii) If $N_L < M_L$: The direction will be set to B (Backward), gather data in CDF, then broadcast message backward as the sequence to CH to store this route.

(iii) If $N_L = M_L$: Route flag set to be inner relation then treated as the second case.

In DGP, CH collects all RDMs from its cluster nodes. Collected data and routes help the CH to build Data Table (DT). Each relation and data collected in the received RDM is listed in the DT. GNs data is listed as well in DT as gateway's relation with other neighbored clusters to be ready for any communication requests with neighbored clusters. CFP and DGP algorithms are illustrated in Flowchart 1, Cluster Formation and Data Gathering, and Pseudocode 1 Cluster Formation and Data Gathering Pseudocode in detail.

Pseudocode 1 Cluster Formation and Data Gathering.

Input:

- Ad hoc network with n nodes
- Cluster size = L_M
- τ : The initial cluster formation time.

Trigger:

- At the start of each timestamp.

Output:

- Dividing the network into clusters, in which each cluster consists of cluster-head, nodes, and sub-clusters.
- Discover all nodes shared the same cluster and each available route between them.

– Collect nodes' data to the cluster-head server.

Steps:

```

1  Set  $M_L = 0$  & Direction = F
2  If ( $\tau$  is not finished), then
3      For each CH  $\in$  NSA Do
4          CH generates RDM and broadcast
5          For each  $N_r \in$  CH Do
6              If  $N_L \leq M_L$  Then
7                  If  $N_r$  is New Then
8                      Generate  $N_{ID}$ 
9                      Copy Sequence in RDM
10                      $N_L = M_L ++$ 
11                 Else
12                     If  $N_r \in$  Different CH
13                         Set  $N_r$  as GN to this neighbor CH
14                     Add Neighbor  $CH_{ID}$  to  $N_{ID}$  in gateway
15                 flag
16                 Set Direction = B
17             End If
18         Else If N, E Any Cluster then
19             Generate  $SCH_{ID}$ 
20             Set node as GN to this SCH
21             Set  $N_L = 0$ 
22         Else
23             If  $N_L > M_L$  then
24                 Copy Sequence
25             Else
26                 Set Direction = B
27                 Copy Sequence
28             End If
29             End If
30              $M_L = M_L ++$ 
31         Next
32     If Direction = B, then
33         For each node in sequence, Do
34             Gather Node Data
35             Broadcast RDM back to CH as Se-
36             quence
37             If  $N_r \neq NNS$  Then
38                 If  $N_L \neq 0$  Then
39                     Discard RDM
40                     Break loop
41                 End If
42             Else
43                 Break loop
44             End If
45              $M_L = M_L - 1$ 
46         Next
47     End If

```

Illustrative Example: In this subsection, a simple illustrative example showing how CH mines its cluster members' paths will be presented in Fig. 5. Through the model, the following assumptions are considered.

Fig. 5 Cluster nodes divided into levels.

The example is applied to a VANET service area consist of one cluster divided into levels with a maximum of 4 levels and 18 NN members. Each level has some members. Level (0) = {CH}, Level (1) = {A, B, C, D}, Level (2) = {E, F, G, H, I}, Level (3) = {J, K, L, M, N, O, P, Q}. Each node must represent its data (Level, CHID, NID, Direction, Handshaking time τ , Average speed ΔS , and Average acceleration ΔACC). All collected data are filled into DT as followed in Table 1.

Table 1 Collected relations and data of the Illustrative Example.

Relation		τ (sec.)	D (deg.)	ΔS (km/h)	ΔACC (km/h ²)
CH	A	3.55	51	71	19
CH	B	1.87	34	4	23
CH	C	0.39	134	43	24
CH	D	2.89	23	75	10
A	B	4.00	108	68	12
A	E	0.29	110	19	19
B	F	3.74	115	8	11
B	G	1.05	107	18	4
C	D	0.89	130	0	18
C	G	4.01	94	47	5
C	H	2.55	68	41	19
D	I	2.24	49	27	25
E	K	3.62	82	46	11
G	L	1.22	41	10	1

G	M	2.68	45	60	25
H	N	3.22	32	23	23
I	N	5.92	113	55	13
I	O	2.22	110	6	3
I	P	3.18	111	43	6
K	J	3.53	56	19	4
O	P	12	7	115	2.8
P	Q	4.31	35	0	10

- **Definition 1. Handshaking time (τ)**

The handshaking time factor is defined as the summation of time taken to send a communication request message and reply with acknowledgment between each two direct contact nodes in a route. It can be expressed by Eq. (1).

$$\tau = \sum_{n=1}^{R_L-1} \tau_n \quad (1)$$

Which: R_L (Route Length) is defined as the number of intermediate nodes (e.g., the number of hops) between source and destination nodes. It should be as low as possible to decrease the loss of time in handshaking and then rebroadcast inner nodes. Furthermore, τ_n is expressed by Eq. (2).

$$\tau_n = \text{message received time} - \text{Message set time} \quad (2)$$

- Message set time: when a node receives RDM with lower M_L , it sets the node's internal clock of this M_{ID} to zero and wait for its return.
- Message received time: the time it takes for the same RDM to return with acknowledging that it has been received successfully. An example is described in Fig. 6.

Fig. 6 Example of Handshaking time between three nodes A, B, and C.

- **Definition 2. Direction (D)**

The direction factor is defined as the relative Direction in degrees between two direct related nodes - using an inner compass - in the route from the source node along to the destination node.

- **Definition 3. Average Route Speed (ΔS)**

Average Route Speed is defined as the average of total relative speeds between each two direct contact nodes in a route. It can be expressed by Eq. (4).

$$\Delta S = \sum_{n=1}^{R_L-1} |S_n - S_{n-1}| / (R_L - 1) \quad \text{km/h} \quad (4)$$

- **Definition 4: Average Route Acceleration (ΔA_{CC})**

Average Route Acceleration is defined as the average of total relative accelerations between each two direct contact nodes in a route. It can be expressed by Eq. (5).

$$\Delta A_{CC} = \sum_{n=1}^{R_L-1} |A_n - A_{n-1}| / (R_L - 1) \quad \text{km/h}^2 \quad (5)$$

3.1.2. Mining and Ranking Operation (MRO)

MRO is used to mine and rank routes in the cluster and ready for any communication requests to minimize route analysis time and provide the most reliable route. Two main steps accomplish this operation: Path Mining and Route Ranking.

- **Path Mining Algorithm (PMA)**

In PM, CH collects all routes between nodes, creates UCG formation, and designs Cluster Head Route Table (CHRT), described in Flowchart 2. PMA consists of four main steps:

- Convert collected routes to UCG.
- Rank main and alternative routes between nodes and others with fuzzy rules depend on four factors discussed in detail below.
- Broadcast Ranked UCG Route Table to all nodes in the cluster.
- Create node RT (NRT) using Dijkstra's Algorithm and choose the optimal routes between source and destination nodes.

PM's main aim is to design UCG by filling NRT and CHRT with the routes between each node and other, including CH as if any node requests a connection between any other node. Initially, to convert a cluster formation as in Fig. 7 into a route map UCG as in Fig. 7, the following steps must be considered:

- Split graph into levels (zero to $L_M - 1$) and arrange nodes according to their levels, as shown in Fig. 7 (a).
- Read each route relationship in RT and create a relation configuration in UCG until it completes as in Fig. 7 (b).

After PM and UCG are designed, The RT has been filled with all possible routes. However, routes reliability factors are not computed yet. Generally speaking, a path (e.g., route) can be broken into a set (and possibly one) of links. For example, the route ($X \rightarrow Y \rightarrow Z$) can be divided into two links, " $X \rightarrow Y$ " and " $Y \rightarrow Z$ ". Setting the reliability factor (RF) of a path can be

Flowchart 1 Path Mining Algorithm.

Fig. 7 Converting Cluster formation (a) into UCG formation (b) according to nodes' levels and relations.

done by predicting all links that can set up that path. Route factors must be computed by Route Ranking Operation (RR) to evaluate each route's RF.

Based on UCG design, ranking is taking place to acknowledge each relation in RT. Which is defined with four factors: Handshaking time (τ), Direction (D), Speed (S), and Acceleration (A_{cc}). The main target of the VANET network is to exchange information with each other as fast as possible. So, time is a fatal factor to judge a relation, and so, the direction. Speed and acceleration are helpful factors but not as weight as Time and Direction.

Hence each factor has a different weight in fuzzy rules. Accordingly, there is a need for combining those factors (e.g., τ , D, S, A_{cc}) to get the RF of any route. Although techniques can be considered to achieve such an aim, we claim that the most suitable one is fuzzy inference. The cause is that it perfectly makes decisions using estimated values under partial or indefinite input data [22, 23]. So, decision making in recent VANET routing protocols is taken by applying

Fuzzy logic. However, in the proposed FBRS, the fuzzy inference is used for estimating the route RF. Generally, there are three steps to apply the fuzzy inference system, namely: (i) Fuzzification, (ii) Fuzzy rule Induction, and (iii) Defuzzification.

• Fuzzification

In FBRS, four fuzzy parameters are considered, namely, τ , D, ΔS , and ΔA_{cc} . At the fuzzification step, the inputs' range values are converted into membership levels (High, Medium, Low) for the employed fuzzy sets. For every fuzzy set, a membership function is used to present the input range's resemblance grade to the corresponding fuzzy set.

The result is a value that varies from "zero" (as no membership) to "one" (as full membership). Table 2 demonstrating each α and β considered fuzzy set values. Furthermore, in Fig. 8, The employed membership functions are demonstrated for the regarded fuzzy sets.

Table 2 Considered α and β values for each parameter.

Assigned value				Parameter
$\alpha\tau$	2	$\beta\tau$	4	seconds
αD	45	βD	90	degrees
$\alpha_1\Delta S$	15	$\beta_1\Delta S$	30	km/h
$\alpha_2\Delta S$	45	$\beta_2\Delta S$	60	
$\alpha_1\Delta A_{CC}$	5	$\beta_1\Delta A_{CC}$	10	km/h ²
$\alpha_2\Delta A_{CC}$	15	$\beta_2\Delta A_{CC}$	20

Fig. 8 Fuzzy membership input set. (a) Handshaking Time. (b) Direction. (c) Average speed. (d) Average Acceleration.

- *Fuzzy rule Induction:*

After input fuzzification, the fuzzy rule base is employed. In FBRS, 36 different roles will be considered, which are depicted in Table 3. (assuming ‘H’ for “High”, ‘M’ for “Medium”, and ‘L’ for “Low”). For example, in Table 3, the fifth role indicates that: if τ is Low (L), D is Low (L), ΔS is Medium (M), and ΔA_{CC} is Low (L), so the Output would be High (H).

Table 3 Fuzzy Roles.

Rule ID	τ	D	ΔS	ΔA_{CC}	Output
1	L	L	L	L	H
2	H	L	L	L	M
3	L	H	L	L	M
4	H	H	L	L	L
5	L	L	M	L	H
6	H	L	M	L	M
7	L	H	M	L	M
8	H	H	M	L	L
9	L	L	H	L	H
10	H	L	H	L	M
11	L	H	H	L	M
12	H	H	H	L	L
13	L	L	L	M	H
14	H	L	L	M	M
15	L	H	L	M	M
16	H	H	L	M	L
17	L	L	M	M	H
18	H	L	M	M	M
19	L	H	M	M	M
20	H	H	M	M	L
21	L	L	H	M	H
22	H	L	H	M	M
23	L	H	H	M	M
24	H	H	H	M	L
25	L	L	L	H	H
26	H	L	L	H	M
27	L	H	L	H	M
28	H	H	L	H	L
29	L	L	M	H	H
30	H	L	M	H	M
31	L	H	M	H	M
32	H	H	M	H	L
33	L	L	H	H	H
34	H	L	H	H	M
35	L	H	H	H	M
36	H	H	H	H	L

On the other hand, the max-min average composition operation is considered in FBRS as it was more suitable and efficient for the collected data nature. Therefore, let be assumed $\mu(i,j)$ is the input fuzzy rule value, μ_K is the minimum fuzzy rule value of each row, μ_M is the output fuzzy rule value, M is the number of output rules, and N is the number of parameters.

As a result, the max-min compositional interface formula can be described as equation (6,7).

$$\mu_K[i] = \text{Min}(\mu_{(i,j)}, \mu_{(i,j+1)}) \quad (6)$$

$$\mu_M = \text{Max}(\mu_K) \quad (7)$$

where $i \in \{1, 2, 3, \dots, M\}$ and $j \in \{1, 2, 3, \dots, N-1\}$

- *Defuzzification:*

As the output is a fuzzy set of the inference operation, many real-life applications are usually required to be crisp values. Accordingly, defuzzification of the output of the fuzzy rules must be achieved by mapping from a fuzzy space into a non-fuzzy action space. Generally, defuzzification techniques have many different methods, and the most commonly used methods are: mean of maxima, max-criterion, and center-of-gravity (COG). As COG is the most popular defuzzification technique, it is the used technique in this paper. Fig. 9 demonstrates the output membership function that achieved the defuzzification operation.

Fig. 9 Fuzzy output membership set.

Let be assumed the relations' data collected from the previous example is described in Table 4. The methodology of computing the RF of any accessible route will be explained in the following subsection. Assuming that there are two available paths between the two nodes, S and D, as demonstrated in Fig. 10.

Fig. 10 Example on computing route reliability.

By assigning RF for each path, the paths can be ranked. The RF will be calculated for the two paths with the same procedure. Therefore, the three steps that must be applied are: (step 1) Find the fuzzy set membership values for each input, (step 2) Calculate the fuzzy outputs for each role, and finally (step 3) Defuzzification of the output RF value as followed.

Table 4 Assumed data table of the relations in the example of Fig. 6.

Path No.	Relation		τ	D	ΔS	ΔA_{CC}
1	S	A	3.40	30	48	8
2	A	B	3.55	51	71	19
3	B	D	1.87	34	4	23
4	S	C	0.39	134	43	24
5	C	E	2.89	23	75	10
6	E	D	3.55	51	71	19

- *Step 1:*

Find the fuzzy set membership values. Let be considered the first relation S-A data ($\tau = 3.4$ seconds, $D = 30$ degrees, $\Delta S = 48$ km/h, $\Delta A_{CC} = 7.5$ km/h²). The S-A relation data was used, and the fuzzy membership set values were as described in fig. 11.

Fig. 11 Fuzzy input set values of S-A relation example. (a) Handshaking Time (b) Direction (c) Average speed (d) Average Acceleration.

– Step 2:

Calculate the fuzzy outputs for each role as in Table 5. Hence, μ_{High} of the direction = 0 and average speed = 0, and μ_{Low} of the average acceleration = 0, so some rule of table 3 will not be used which are:

- If μ_{High} of the direction = 0 then 3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23, 24, 27, 28, 31, 32, 35, 36 will not be considered.
- If μ_{High} of the average speed = 0 then 9, 10, 11, 12, 21, 22, 23, 24, 33, 34, 35, 36 will not be considered.
- If μ_{Low} of the average acceleration = 0 then 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 will not be considered.
- the remaining rules are 13, 14, 17, 18, 25, 26, 29, 30 (8 rules).

So that the output will be as in Fig. 12:

High membership = MAX (0.3, 0.2, 0.3, 0.2) = 0.3

Medium membership = MAX (0.4, 0.2, 0.6, 0.2) = 0.6

Table 5 The fuzzy output membership.

Role ID	τ	D	ΔS	ΔAcc	Role Output	Output membership
1	L	L	M	L	H	Min = 0.3
2	H	L	M	L	M	Min = 0.4
3	L	L	H	L	H	Min = 0.2
4	H	L	H	L	M	Min = 0.2
5	L	L	M	M	H	Min = 0.3
6	H	L	M	M	M	Min = 0.6
7	L	L	H	M	H	Min = 0.2
8	H	L	H	M	M	Min = 0.2

Fig. 12 The output membership.

– Step 3:

Defuzzification using the Center of Gravity (COG) method as in equation (8):

$$COG = \frac{\sum \mu_x * x}{\sum \mu_x} \quad (8)$$

$$\sum \mu_x * x = (0.0 * (0.0 + 0.1 + 0.2) + 0.5 * 0.3 + 0.6 * (0.4 + 0.5 + 0.6) + 0.5 * 0.7 + 0.3 * (0.8 + 0.9 + 1.0)) = 2.21$$

$$\sum \mu_x = \sum (0.0 \rightarrow 1.0) = 5.5$$

$$COG = \frac{\sum \mu_x * x}{\sum \mu_x} = \frac{2.21}{5.5} = 0.402 = 40.2\%$$

Hence, Path 1 RF = 0.402. Therefore, other paths' RF will be calculated as the same.

Broadcasting:

Now, UCG and RT are designed with routes and relations between each other nodes in the cluster. So, CH begins to broadcast and share UCG and RT with all nodes in the cluster. Nodes receive this data and start to process this data to design their NRT.

Dijkstra's routing protocol:

One of the most used methods to solve the shortest path problem is the Dijkstra algorithm. The Dijkstra algorithm can be easily implemented in a crisp number to model arc lengths [24]. However, it is used to determine the most reliable route based on the four parameters. It can get each relation's RF using fuzzification and defuzzification techniques. In [25], collects the output data in a large routing board table, which contains all routes between nodes of the cluster just in the cluster head. Any node that requests a communication must request a route from the cluster head.

If there were any crashes in the obtained route, the source node must return to the cluster head to request an alternative route. This because the size of the output routing board table must remain at the cluster head to analyze. However, using Dijkstra's algorithm optimizes the output routing table in a much smaller and simpler table to analyze. Also, it minimizes the mining and ranking delay to generate RT, making it more suitable to handle high dynamic networks such as VANETs rather than other protocols. This protocol is based on finding the most reliable route with a more straightforward table than discussed papers previously.

Hence every car mines its own Dijkstra's NRT, all routes to all nodes in the same cluster are ready to make a connection anytime if requested. In the original Dijkstra's algorithm, its target to find the shortest path between any two nodes in a network. However, in a high mobility network like VANET, many parameters would be concerned rather than the distance between source and destination. The combined parameters into the route RF are more suitable and efficient for the VANET system. When a node in the cluster receives CHRT from CH, ROT generates its Optimized Route Table (ORT).

It finds the most reliable path between the receiver node and each other node in the same cluster. Although, it provides the route sequence and average reliability percentage of the path. The Reliability factor Optimization Technique (ROT) processes are introduced in detail in Pseudocode 2.

Input:

- CHRT message containing cluster members, relations, and their RFs

Trigger:

- After each CHRT message is received

Output:

- Optimized Route Table (ORT) has communication routes between the receiver node and all other clusters, including CH.

Steps:

```

1  Set Unchecked [ ] = all unduplicated nodes in CHRT
2  Set Related [ ] ;
   //define list Related with empty value
3  Function ROT (source, CHRT):
4      For each relation r in CHRT:
5          Related [r] = RN;
6          RR [r] = 0;
7          NN[r] = Undefined;
8      End for;
9      RR [source] = 1;
10     RL = 0;
11     While (Undefined in not empty):
12         SN = Related [0];
13         Remove SN from Unchecked & Related list;
14         For each RN of SN in Unchecked:
15             P = (RR(SN,RN)+(RR[SN]*RL))/(RL+1);
16             If (P > RR[RN]) then:
17                 RR[RN] = P;
18                 NN[RN] = SN;
19             End if;
20         End for;
21         RL = RL + 1;
22     End while;
23     Return (RR, NN);
24 End function;

```

For the illustrative example, let us have the same example of Table 4. It has collected relations and data in Table 5, which minded each relation's RF using fuzzification (Min-Max) and defuzzification (COG) techniques. The calculated RFs are filled into CHRT, as in Table 6, to be ready for the next step. The ROT is based on Dijkstra's Algorithm to find the highest path reliability.

Assuming that a source node E request a communication channel to destination node P in the illustrated example in Fig. 5. ROT algorithm starts to generate

ORT of node E. The result table is demonstrated in table 7. The path's RF can be calculated as the RF average of each relation from the N_s to N_D . As in ORT, the path ($E \rightarrow P$) will be as followed:

$$(P \leftarrow O \leftarrow I \leftarrow D \leftarrow CH \leftarrow B \leftarrow A \leftarrow E)$$

The route RF

$$\begin{aligned}
 &= \text{Average (E-A, A-B, B-CH, CH-D, D-I, I-O)} \\
 &= (0.213 + 0.614 + 0.081 + 0.206 + 0.186 + 0.308) / 6 \\
 &= 0.268 = 26.8 \%
 \end{aligned}$$

Table 6 Cluster Head Route Table.

Relation		Reliability	Relation		Reliability
CH	B	0.081	I	D	0.186
A	E	0.213	J	K	0.373
B	A	0.614	K	E	0.570
C	CH	0.227	L	G	0.119
D	CH	0.206	M	G	0.184
E	--	1.000	N	I	0.426
F	B	0.572	O	I	0.308
G	B	0.227	P	O	0.415
H	C	0.375	Q	P	0.267

Table 7 ORT of node E.

Relation		Reliability	Relation		Reliability
CH	A	0.319	D	I	0.186
CH	B	0.081	E	K	0.570
CH	C	0.227	G	L	0.119
CH	D	0.206	G	M	0.184
A	B	0.614	H	N	0.210
A	E	0.213	I	N	0.426
B	F	0.572	I	O	0.308
B	G	0.227	I	P	0.482
C	D	0.267	K	J	0.373
C	G	0.540	O	P	0.415
C	H	0.375	P	Q	0.267

3.2. System Operation Phase (SOP)

The main idea of the VANET network is to exchange information between nodes and each other and the government servers like traffic, Ambulance, Fire Extinguisher, Police Stations. So, vehicles can act as communication hubs or intermediaries to share local resources instead of sending redundant bulk data to cloud servers [26, 27]. Fog computing can refine the data obtained from social networks and vehicular networks to predict the possible interdependencies. It includes both local decisions making and geo-distribution characteristics to bring less delay [28]. This subsection describes how to start a Private Communication Channel (PC²) between any two nodes (source

node N_S and destination node N_D) in the service area. It also maintains any simultaneous crushes in the communication channel without returning to the source node to request an alternative route. The SOP is divided into two primary operations: Global Routing Operation (GRO) and Real-time Routing Maintenance (R^2M).

3.2.1. Global Routing Operation

GRO's primary goal is to establish a PC^2 between N_S and N_D with minimum delay and the most reliable network connections. To accomplish that, the GRO task is to find N_D in the service area and generate a reliable route between N_S and N_D .

Let be assumed that N_S requests for a PC^2 with N_D . First, GRO searches for N_D in the same cluster of N_S using N_S NRT to generate a route with the highest RF. This route is called Selected Communication Sequence (SCS). Communications between N_S and N_D have four cases: (i) N_S and N_D are in the same cluster, (ii) N_S and N_D are not in the same cluster but in the same Fog server, (iii) N_S and N_D are in different Fog servers, and (iv) N_D is not found any servers.

As follow in the flowchart (3), each case will be described in details:

- Case (i): N_S and N_D are in the same cluster:

In this case, N_S finds a relation with N_D in its NRT. N_S initializes a Communication Request Message (CRM) to communicate with N_D and generates Route Sequence (RS) - path routing address list between N_S and N_D using NRT of N_S . Then it adds RS to CRM and sends it as in RS along to N_D . In case N_D agrees to communicate, the information appears and otherwise. If the data package is larger than the size of CRM, it returns with acknowledgment to N_S to roll over the remaining data in a sequence of packages. In any case, the CRM must include handshakes and returns the acknowledgment message with a reply (agreed or refused) as it follows the same sequence back to the source.

- Case (ii): N_S and N_D are in the same Fog Server (FS):

In case N_S cannot find a relation with N_D in its NRT. It generates a CRM and sends it to its CH, which forwards it to its Fog server. If N_D is found in the Fog service area, CRM returns with confirmation, its CH_{ID} , the Gateway-to-Gateway route (G2G) to N_D cluster. Then N_S adds RS to the first gateway to CRM, G2G, and last gateway to N_D .

- Case (iii): N_S and N_D are in different FSs:

In case that N_D is not in the same cluster or any neighbored clusters (Fog service area), the CRM generated is forward to the central cloud server to search for N_D . If N_D is found, CRM returns to the FS with confirmation, N_D CH_{ID} and FS_{ID} , and FS to FS route sequence. Then the FS returns the CRM to N_S CH again as described in case (ii).

- Case (iv) N_D is not found on any servers:

If N_D is not found on any servers, CRM is forward back to N_S as in case (iii) but with access denied as wrong N_D ID, or N_D is not available at the moment.

3.2.2. Realtime Route Maintenance (R^2M):

One of VANET routing networks' significant problems is crashed links due to its high mobility network nature [29]. Most VANET routing protocols depend on CH to find a new route to ND. It requires returning every crashed message to CH, which causes more traffic on Cluster network and CH, more delay time to find a route back to CH, and a higher chance of network changes.

So, R^2M is critical for improving network efficiency and reliability in delivering data successfully. The R^2M algorithm principle is to automatically find an alternative path during the connection without returning to CH requesting a new path. It starts with a crashed CRM that is received by the Last Successful Node (LSN). The R^2M algorithm removes the crashed node from the NRT of LSN to send any messages to the crashed node until the next RT is received. Then, it generates a Request an Alternative Route (RAR) message at LSN and broadcast it to all Direct Relation Node(s) (DRNs). Every node that received a RAR message should return it with route sequence to N_D if found and its RF. LSN collects all RAR messages and selects RS with the Highest Reliability Factor (HRF). So, LSN replaces the crashed RS of CRM with the active one and resend CRM as the new RS. The R^2M algorithm is described in detail in Flowchart 4.

Flowchart 4 The R²M Algorithm.

4. Experimental Results:

The proposed routing strategy FBRS is compared with four approaches, which are: SRS [30], OEB [31], R²P [25], and 5G-FSDN [32]. The simulation was performed using the Network Simulator NS2 (version 2.35) [33], the vehicular mobility simulator SUMO (version 1.4.0) [34], and OpenStreetMap API [35]. The vehicles' starting positions are distributed randomly on the road maps. NS-2 implements 802.11 physical and MAC models for V2V communication with a maximum transmission range of each vehicle is set to 500m. Table 8 demonstrates the simulation environment parameters used to set up for NS2.

The following Figures (from Fig. 13 to Fig. 18) illustrate the comparison of the proposed algorithms in terms of packet delivery ratio, routing overhead, normalized routing load, average end-to-end delay, and throughput of the system described in detail below. The reported results are the average of 10 independent runs with the same configuration and circumstances.

Table 8 NS2 parameters.

Parameter	Value
Platform	Linux Ubuntu 18.04 (64 Bit)
NS-Version	NS 2.35
Traffic Type	CBR
Radio Propagation	Two-Ray Ground
Antenna Type	Omni-Antenna
Simulation time	120 seconds
Pause Time	30 seconds
Average speed	10, 20, 30, 40, 50, 60 km/h
Number of nodes	200 nodes
Network size	2000 x 2000 m ²

4.1. Packet delivery ratio (PDR)

The packet delivery ratio for SRS, OEB, R²P, 5G-FSDN, and FBRS was measured against the pause time and nodes' mobility speed. Fig. 13 demonstrates compared algorithms against different pause times of 20 sources with an average moving speed of 30 km/h inside the city [36]. As illustrated in Fig. 13, the PDR of all tested algorithms is increased along with pause time. FBRS's PDR achieved its maximum value at a pause time of 30 seconds. After this pause time, other algorithms start to variant. Some algorithms are increasing in PDR as in SRS, and some others are decreasing as in FBRS.

Fig. 13 Packet delivery ratio of the compared algorithms against pause time.

Moreover, in 5G-FSDN, it increased till 20 seconds, decreased and stabilized at 35, then increased again after 45 seconds. Therefore, 30 seconds pause time was chosen as it achieves the highest average PDR and most stabilization with compared algorithms. Although PDR was measured against different mobility speeds between 10 and 60 km/h, various sources vary between 10 and 60. Fig. 14 illustrates the road map with 10 to 60 sources of 200 nodes request a communication channel.

Fig. 14 Packet delivery ratio against different average speeds and number of sources.
(A) SRS (B) OEB (C) R²P (D) 5G-FSDN (E) FBRS

As depicted in Fig. 14, generally, with low mobility and a small number of sources (10 sources), the competing protocols can deliver most of the delivered data packets (around 92–99%). However, as the number of sources increases (up to 60 sources) as well as increasing the source speed, it is found that the packet delivery ratio gradually decreases mainly in R²P, OEB, and SRS. Considering FBRS, it is shown in Fig. 14 (E) that it outperforms all other competitors. As NRT continuously delivers up-to-date routing information to the cluster nodes. R²M algorithm maintains crashed routes automatically without rolling back, asking for alternative routes.

Moreover, the GRO algorithm deals with communication requests from other clusters using shared information of the Fog and cloud servers without any delay. So, it gives nodes the ability to communicate with outside nodes simultaneously immediately.

Considering Fig. 14 again, in high mobility (e.g., using little pause time), it is indicated that: 5G-FSDN and FBRS overcome SRS, OEB, and R²P in almost all cases. On the other hand, R²P outperforms SRS but in most cases.

4.2. Routing overhead (RO)

To keep up-to-date information about network topology, routing algorithms generate small-sized packets called routing packets (such as hello messages). Routing packets do not carry any application with specific content as data packets do. However, both routing and data packets share the same network bandwidth most of the time. Accordingly, routing packets are considered as an overhead for the network. This overhead generated by routing packets is called routing overhead. A suitable routing protocol should generate as small routing overhead as possible.

The routing overheads were compared against the maximum mobility speed and depicted in Fig. 15 for all competing protocols. It is noted that routing overheads increase with the increase in sources' mobility speed. In Fig. 15, as the sources' mobility is increased, it is noted that SRS has lower overheads according to OEB. In a small number of sources (10 sources), the behavior is similar to all competing protocols regardless of the average speed of sources. It can be concluded that SRS has lower routing overheads than 5G-FSDN and OEB.

Fig. 15 The routing overhead against different average speeds and number of sources.
(A) SRS (B) OEB (C) R²P (D) 5G-FSDN (E) FBRS

Moreover, as illustrated in Fig. 15, FBRS outperforms the other competing protocols as it has minimal routing overheads. The reason is that it efficiently tracks the changes simultaneously at all the time in the network topology. Further, as in FBRS, the selected route for data transmission is the most reliable route: routing overheads are minimized since these routes' maintenance decreases. Fig. 15 illustrates the routing overhead for the tested techniques considering different numbers of sources. Also, by increasing the number of sources, the routing overhead increases. So, increasing the number of sources will undoubtedly increase the cluster routing packets (traffic).

4.3. Normalized routing load (NRL)

The normalized routing load can be defined as the fraction of all control packets sent by network nodes over the number of all received data packets at the destination nodes. It is the ratio between all routing packets sent over the network to all data packets received. The performance of the proposed FBRS in terms of the normalized routing load is illustrated in Fig. 16.

Fig. 16 The Normalized Routing Load against different average speeds and number of sources.
(A) SRS (B) OEB (C) R²P (D) 5G-FSDN (E) FBRS

As depicted in this figure, it is evident that FBRS overcomes SRS, OEB, R²P, and 5G-FSDN in most cases. On the other hand, OEB outperforms SRS in most cases, primarily by increasing mobility sources. Further, 5G-FSDN overcomes OEB in cases of low average speed with a low number of sources. From another point of view, as illustrated in Fig. 16, It is noted that the other protocols introduce higher NRL than FBRS. FBRS ranks and offers the most reliable path for transmitting data. Also, with the R²M algorithm in FBRS, there is no need for route maintenance while sending data and minimizes the normalized routing load.

4.4. Average end-to-end delay (AETE)

Generally, VANETs are delay-sensitive and require some delay guarantees. Unfortunately, offering and maintaining delay guarantees is a challenging task in highly dynamic networks such as VANETs. Although several studies have been introduced to solve the routing problem in VANETs, most of them have focused on throughput guarantees. Hence, they have concentrated on finding the shortest route for transmitting data among nodes, while among recent studies, few have proposed solutions for delay guarantees. We claim FBRS achieves delay guarantees not only in VANETs but also in all wireless multi-hop networks. The idea behind FBRS is to select the most reliable

route to carry data between the source and destination. Such a perfect choice from the beginning helps the network to get rid of frequent disconnections and, accordingly, no need for retransmitting lost data.

The guarantees that the data arrival on time as FBRS depends on a priority estimation of the most reliable route. As illustrated in Fig. 17, delay in SRS is higher than OEB, R²P, 5G-FSDN, and FBRS, especially with the increase in the number of sources (60). In all cases of real-time packet delivery, FBRS is the best choice. On the other hand, SRS can quickly respond with a route. It spends a long delay when a route is rebuilt.

Fig. 17 Average End-to-End Delay of the compared algorithms against different average speeds and number of sources.
(A) 20 sources. (B) 60 sources.

4.5. Throughput

Generally, throughput is the rate of production. Throughput is described in the routing subject as the number of packets received successfully in the unit time. Many routing protocols had been proposed for VANETs. However, most of them suffer from degraded throughput due to the network's high dynamics. The throughput of compared routing protocols (e.g., SRS, OEB, R²P, 5G-FSDN, and FBRS) is measured through this section. As depicted in Fig. 18, FBRS overcomes OEB, SRS, R²P, and 5G-FSDN in data packets transmitting successfully through the considered period.

Fig. 18 The Throughput of the compared algorithms against different number of sources.

5. Conclusion

In this paper, a new Fog-Based Routing Strategy (FBRS) for VANETs is introduced. FBRS merges both proactive and reactive routing approaches. Hence, it gains the advantages of both. According to the most reliable path, this paper's originality is that FBRS employs an efficient and stable route discovery mechanism, mining, and ranking routes. Also, it presents a global communication system using Fog and Cloud computing servers.

A unique real-time maintenance mechanism in which nodes do not need to request another path at path crashes. FBRS has been compared against the current VANET's routing protocols. Experimental results have proven that FBRS overcomes the compared protocols as it has much higher performance in terms of throughput, bandwidth consumption, and delay time.

6. References

- [1] Y. R. B. Al-Mayouf *et al.*, "Efficient and stable routing algorithm based on user mobility and node density in urban vehicular network," *PloS one*, vol. 11, no. 11, pp. 1-24, 2016.
- [2] M. Rath and B. Pati, "Communication Improvement and Traffic Control Based on V2I in Smart City Framework," *International Journal of Vehicular Telematics and Infotainment Systems (IJVTIS)*, vol. 2, no. 1, pp. 18-33, 2018.
- [3] M. Vijayaraj and S. Srinivasan, "Dynamic speed adaptive classified (D-SAC) data dissemination protocol for improving autonomous robot performance in VANETs," *Automatika*, vol. 60, no. 5, pp. 535-546, 2019.
- [4] L. Sarakis *et al.*, "802.11 p-Based VANET Applications Improving Road Safety and Traffic Management," in *Emerging Innovations in Wireless Networks and Broadband Technologies*: IGI Global, 2016, pp. 135-168.
- [5] S. Glass, I. Mahgoub, and M. Rathod, "Leveraging MANET-based cooperative cache discovery techniques in VANETs: A survey and analysis," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2640-2661, 2017.
- [6] M. Zhu, J. Cao, Z. Cai, Z. He, and M. Xu, "Providing flexible services for heterogeneous vehicles: An NFV-based approach," *IEEE network*, vol. 30, no. 3, pp. 64-71, 2016.
- [7] A. Walker and M. Radenkovic, "GPSR-TARS: congestion aware geographically targeted remote surveillance for VANETs," in *2017 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, 2017, pp. 1-6: IEEE.
- [8] I. Ahmad *et al.*, "VANET-LTE based heterogeneous vehicular clustering for driving assistance and route planning applications," *Computer Networks*, vol. 145, pp. 128-140, 2018.
- [9] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2377-2396, 2015.
- [10] W. B. Jaballah, M. Conti, and C. Lal, "A Survey on Software-Defined VANETs: Benefits, Challenges, and Future Directions," *arXiv preprint arXiv:1904.04577*, 2019.
- [11] S. S. Kumar and Y. J. W. N. Palanichamy, "Energy efficient and secured distributed data dissemination using hop by hop authentication in WSN," *Wireless Networks*, vol. 24, no. 4, pp. 1343-1360, 2018.
- [12] H. Wang *et al.*, "Architectural Design Alternatives based on Cloud/Edge/Fog Computing for Connected Vehicles," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2349-2377, 2020.
- [13] D. D. Adriano, C. Montez, A. G. Novaes, and M. J. E. Wingham, "DMRVR: Dynamic Milk-Run Vehicle Routing Solution Using Fog-Based Vehicular Ad Hoc Networks," *Application of Wireless Sensor Networks in Monitoring*, vol. 9, no. 12, p. 2010, 2020.
- [14] A. Ullah, X. Yao, S. Shaheen, and H. J. I. T. o. I. T. S. Ning, "Advances in Position Based Routing Towards ITS Enabled FoG-Oriented VANET-A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 828-840, 2020.
- [15] A. V. Sutagundar, A. H. Attar, and D. I. J. W. P. C. Hatti, "Resource allocation for fog enhanced vehicular services," *Wireless Personal Communications*, vol. 104, no. 4, pp. 1473-1491, 2019.
- [16] F. Y. Okay and S. Ozdemir, "Routing in fog-enabled IoT platforms: A survey and an SDN-based solution," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4871-4889, 2018.
- [17] Y. Deng, Y. Chen, Y. Zhang, and S. Mahadevan, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment," *Applied Soft Computing*, vol. 12, no. 3, pp. 1231-1237, 2012.
- [18] M. H. Eiza and Q. Ni, "An evolving graph-based reliable routing scheme for VANETs," *IEEE transactions on vehicular technology*, vol. 62, no. 4, pp. 1493-1504, 2013.
- [19] M. Enayattabar, A. Ebrahimnejad, and H. Motameni, "Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment," *Complex & Intelligent Systems*, vol. 5, no. 2, pp. 93-100, 2019.

- [20] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune, "Geographic routing protocols for Vehicular Ad hoc NETWORKS (VANETs): A survey," *Vehicular Communications*, vol. 11, pp. 20-31, 2018.
- [21] A. Di Maio, M. R. Palattella, and T. Engel, "Performance Analysis of MANET Routing Protocols in Urban VANETs," in *International Conference on Ad-Hoc Networks and Wireless*, 2019, pp. 432-451: Springer.
- [22] M. Selvi, S. S. Kumar, S. Ganapathy, A. Ayyanar, H. K. Nehemiah, and A. J. W. P. C. Kannan, "An Energy Efficient Clustered Gravitational and Fuzzy Based Routing Algorithm in WSNs," *Wireless Personal Communications*, vol. 104, pp. 1473-1491, 2020.
- [23] K. Thangaramya, K. Kulothungan, S. Indira Gandhi, M. Selvi, S. Santhosh Kumar, and K. J. S. C. Arputharaj, "Intelligent fuzzy rule-based approach with outlier detection for secured routing in WSN," *Soft Computing*, vol. 24, pp. 16483-16497, 2020.
- [24] C. P. Dubey, V. Kumar, B. Sharma, and G. Kaur, "Shortest Path Algorithm for Distributed VANET using Grid Computing," in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2018, pp. 118-121: IEEE.
- [25] A. I. Saleh, S. A. Gamel, and K. M. Abo-Al-Ez, "A reliable routing protocol for vehicular ad hoc networks," *Computers & Electrical Engineering*, vol. 64, pp. 473-495, 2017.
- [26] J. Luo, Q. Pan, and Z. He, "VANET middleware for service sharing based on OSGI," *Computer Science and Information Systems*, vol. 12, no. 2, pp. 729-742, 2015.
- [27] J. Sun, Q. Gu, T. Zheng, P. Dong, and Y. Qin, "Joint communication and computing resource allocation in vehicular edge computing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, pp. 1-13, 2019.
- [28] J. Grover, A. Jain, S. Singhal, and A. Yadav, "Real-time vanet applications using fog computing," in *Proceedings of First International Conference on Smart System, Innovations and Computing*, 2018, pp. 683-691: Springer.
- [29] L. Nassar, M. S. Kamel, and F. Karray, "VANET IR-CAS for Safety ACN: Information retrieval context aware system for VANET automatic crash notification safety application," *International Journal of Intelligent Transportation Systems Research*, vol. 14, no. 3, pp. 127-138, 2016.
- [30] A. I. Saleh, "A scalable-adaptive snack routing strategy (SRS) for semi-administrated mobile ad hoc networks (SAMANETs)," *International Journal of Wireless Information Networks*, vol. 18, no. 4, pp. 257-279, 2011.
- [31] D. Tian, J. Zhou, Y. Wang, H. Xia, Z. Yi, and H. Liu, "Optimal epidemic broadcasting for vehicular ad hoc networks," *International Journal of Communication Systems*, vol. 27, no. 9, pp. 1220-1242, 2014.
- [32] A. A. Khan, M. Abolhasan, and W. Ni, "5G next generation VANETs using SDN and fog computing framework," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018, pp. 1-6: IEEE.
- [33] B. Ahmed, A. W. Malik, T. Hafeez, N. J. E. J. o. W. C. Ahmed, and Networking, "Services and simulation frameworks for vehicular cloud computing: a contemporary survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 4, 2019.
- [34] D. Krajzewicz, J. Erdmann, M. Behrisch, L. J. I. j. o. a. i. s. Bieker, and measurements, "Recent development and applications of SUMO-Simulation of Urban MObility," *International Journal For Innovative Engineering and Management Research*, vol. 5, no. 3&4, 2012.
- [35] M. K. Hasan and O. Sarker, "Routing protocol selection for intelligent transport system (ITS) of VANET in high mobility areas of Bangladesh," in *Proceedings of International Joint Conference on Computational Intelligence*, 2020, pp. 123-135: Springer.
- [36] D. Lin, J. Kang, A. Squicciarini, Y. Wu, S. Gurung, and O. J. I. T. o. M. C. Tonguz, "MoZo: A moving zone based routing protocol using pure V2V communication in VANETs," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1357-1370, 2017.

Appendix

Flowchart 2 Cluster Formation and Data Gathering.

Figures

Figure 1

Network Routing Layers.

Figure 2

Proposed Fog-Based Routing Strategy (FBRS).

Figure 3

Cluster divided into level example.

Figure 4

route discovery message Flags.

Figure 5

Cluster nodes divided into levels.

Figure 6

Example of Handshaking time between three nodes A, B, and C.

Figure 7

Converting Cluster formation (a) into UCG formation (b) according to nodes' levels and relations.

Figure 8

Fuzzy membership input set. (a) Handshaking Time. (b) Direction. (c) Average speed. (d) Average Acceleration.

Figure 9

Fuzzy output membership set.

Figure 10

Example on computing route reliability.

Figure 11

Fuzzy input set values of S-A relation example. (a) Handshaking Time (b) Direction (c) Average speed (d) Average Acceleration.

Figure 12

The output membership.

Figure 13

Packet delivery ratio of the compared algorithms against pause time.

Figure 14

Packet delivery ratio against different average speeds and number of sources. (A) SRS (B) OEB (C) R2P (D) 5G-FSDN (E) FBRS

Figure 15

The routing overhead against different average speeds and number of sources. (A) SRS (B) OEB (C) R2P (D) 5G-FSDN (E) FBRS

Figure 16

The Normalized Routing Load against different average speeds and number of sources. (A) SRS (B) OEB (C) R2P (D) 5G-FSDN (E) FBRs

Figure 17

Average End-to-End Delay of the compared algorithms against different average speeds and number of sources. (A) 20 sources. (B) 60 sources.

Figure 18

The Throughput of the compared algorithms against different number of sources.