

## Article

# Efficient IoT-Based Formal Model for Vehicle-Life Interaction in VANETs Using VDM-SL

Sidra Iqbal <sup>1,\*</sup>, Nazir Ahmad Zafar <sup>1,\*</sup>, Tariq Ali <sup>1,\*</sup> and Eman H. Alkhamash <sup>2</sup>

<sup>1</sup> Department of Computer Science, Sahiwal Campus, COMSATS University Islamabad, Sahiwal 57000, Pakistan

<sup>2</sup> Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; Eman.kms@tu.edu.sa

\* Correspondence: sidra196iqbal@gmail.com (S.I.); nazafar@gmail.com (N.A.Z.); tariqali@cuisahiwal.edu.pk (T.A.)

**Abstract:** VANETs have gained much attention from both industry and academia because of their characteristics, such as dynamic topology. There are various applications of VANETs that are classified on the basis of safety, efficiency, commercial usage, and productive areas. This paper presents an IoT-based formal model for vehicle-life integration enabling RSUs with the help of different approaches. We have developed a model that uses vehicle scenarios in smart transportation systems so that quick data transmission is provided between the source and destination vehicles. Further, fog-based RSUs provide a wide range to communicate with hospitals and emergency vehicles to deal with emergency situations. All the appropriate entities are connected to ensure a consistent traffic flow for the arrival of an emergency vehicle in emergency places. The UML, graph theory, and VDM-SL formal technique are used to represent this system. To model the network and discover appropriate paths for V2V communication, graph theory is applied. The system requirements are designed using a UML diagram. The VDM-SL, an object-oriented model-based formal technique, was utilized for this modeling procedure. This approach assures the safety and accuracy of systems by detecting flaws early in the design process. It also gives an exceptionally important answer to an issue and increases trust in the software's quality.

**Keywords:** VANETs; IoT; fog-based RSUs; formal methods; verification and validation



**Citation:** Iqbal, S.; Zafar, N.A.; Ali, T.; Alkhamash, E.H. Efficient IoT-Based Formal Model for Vehicle-Life Interaction in VANETs Using VDM-SL. *Energies* **2022**, *15*, 1013. <https://doi.org/10.3390/en15031013>

Academic Editor: Gianpiero Colangelo

Received: 2 December 2021

Accepted: 21 January 2022

Published: 29 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Vehicular ad hoc networks (VANETs) technology has been a prominent focus of research because of its dynamic topology and high mobility. To retrieve vehicle status and traffic information, VANETs support vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-infrastructure (I2I) communication. The information-sharing mechanisms have enabled the applications for transportation efficiency, reduced traffic congestion, and enhanced road safety. It also provides the convenience of works between vehicles and infrastructure to ensure a comfortable journey [1]. Many people lose their lives every year, and millions of dollars are consumed on medical bills as well as insurance costs due to road accidents. These large amounts of dollars have led to the improvement of a wide range of new transport technologies that can enable a comfortable, safe driving experience as well as offer extra value-added services to both drivers and passengers [2]. VANETs include many applications for providing various facilities such as traffic management, traffic information, driving safety, route optimization, automation of traffic-related functions, and better efficiency through IoT. In VANETs, the security requirements also included that are authentication, availability, authorization, confidentiality, integrity, anonymity, privacy, and traceability. VANETs applications have captured the attention of academia as well as automobile industries that enable vehicles to connect with the internet to obtain real-time traffic information.

### 1.1. VANETs with IoT

The internet of things (IoT) is an application of smart nodes in a network that detects, interprets, processes as well as reacts to data within the time needed. The ability to incorporate sensing systems into actual environments helps smart environments to be converted. In real life, though, there is very little realistic application of the IoT infrastructure. Researchers are recommending several conventional approaches that do not tackle this issue properly [3]. Because of their growing relevance in developing a smart transportation system, the usage of IoT in VANETs has received a lot of attention in recent years. Smart city modelling is an example of IoT-based applications in which sensor and actor nodes are connected in a network to perform tasks including sensing the environment, collecting and analyzing data, and giving relevant information [4]. The applications of IoT in smart vehicles include maintaining the environment, parking, smooth flow of vehicle movements, and protecting lives. In the area of road safety and transportation efficiency, IoT can produce great impacts on society. IoT is used in VANETs to increase efficiency by providing different communication types such as vehicle-to-sensor (V2S) and vehicle-to-personal device (V2P/V2D) communication [5]. The internet of vehicles (IoV) is used to describe when vehicles are linked to the internet and act as a component of an ad hoc network. With the development of IoT, it is emerging as a revolutionary paradigm in the cellular and mobile networking markets, resolving modern collaboration and interaction technologies. The IoV was born out of the VANET, which is a network of various road transport entities such as cars, pedestrians, sidewalks, parking lots, and community facilities that allows for real-time connectivity [6].

### 1.2. Comparison between Fog-Based RSU and Traditional RSU

Wireless sensors have been used to sense the information in the real-time environment, and actors have been used to perform the actions as required. The RSUs communicate with the vehicles in the vehicular ad hoc networks paradigm. Traditional RSUs store information about the smart vehicles that pass through their transmission range. The RSU sends an alert message to all other RSU's and vehicles in its proximity if an accident occurs. However, we expanded the RSU's scope in traffic data, and each RSU now acts as a fog node. Fog-enabled RSUs have a range of 1000–1500 m, while standard RSUs only have a range of 400–600 m [7,8]. They can provide different services according to their facility to reduce the death rate and maintain traffic flow smoothly. By using the sensor, the fog-based RSUs efficiently gather data and then organize them in an efficient manner, store and then process the data to avoid delays in sensitive tasks [9].

### 1.3. Formal Methods

Formal techniques are mathematically based procedures used in computer systems to define, create, and verify software and hardware under a variety of conditions [10]. Because of their distinct features, formal modeling is commonly utilized in life-critical systems. When we create a formal specification for a system, it is (1) unambiguous, (2) consistent, and (3) complete [11] since the syntax employed in formal methods can only have one meaning. Another compelling argument for adopting formal methods is that they take more work in the early stages of software development, which decreases requirements errors by forcing a thorough study of the requirements. Because testing does not provide us with enough assurance, we must offer a formal demonstration that ensure the software is right.

### 1.4. Contribution

In this paper, we will propose an IoT-based formal model for vehicle-life interaction in VANETs using VDM-SL, which provide emergency services by the arrival of an emergency vehicle at accident locations. Wireless sensors will be used to monitor the real-time situation, and actors will be used to perform actions as required, minimizing the average waiting time, reducing the traffic congestion issues and making communication effective by using

IoT. Further, a unicast routing protocol is utilized to convey messages from the source to destination vehicles. The proposed system is utilized different approaches such as; IoT, Graph theory, Unified Modeling Language (UML), and formal methods with the proof of concept. UML diagram is utilized to portray the system’s requirements comprehensively. The developed graph-based model was utilized to present the network as well as the appropriate route. The formal methods are mathematical techniques supported by many tools that offer effective ways to model, design, and analyze the system. The formal methods in terms of VDM-SL (Vienna Development Method-Specification Language) are used to define the formal modeling of the system in a systematic way, and the VDM-SL toolbox is utilized to provide the validation as well as verification of the model [10].

An abstract representation of the system is described in Figure 1. It displays different techniques such as VANETs, IoT, UML diagrams, graph-based models and formal methods that we have used in our system. It shows the emergency services by the arrival of an emergency vehicle at accident locations. Moreover, our fog-based strategy encompasses objects such as emergency vehicles, hospitals, RSUs, automobiles, and others who are directly or indirectly related to fog-enabled RSUs in order to give improved comfort, safety, and smooth driving. Our fog-based RSU provides emergency services like mobile RSU. It will be covered most road sections as much as possible in cities and on highways between cities. IoT will be adopted in which sensors will be used to monitor the real-time environment, and actors will be used to take action in case of any emergency. This work is displayed in Figure 1 by utilizing the emergency service module diagram as well as the UML diagram. It also represents the unicast scenarios for simple V2V communication, there are several options for a destination spot, but two of them are utilized in our system. Further, this work represents a unicast routing module diagram and graph-based model, which describe the behavior of the module. After this, these models easily translated into a formal model using VDM-SL. The VDM-SL toolbox will be used to convert the formal model into a validated model.

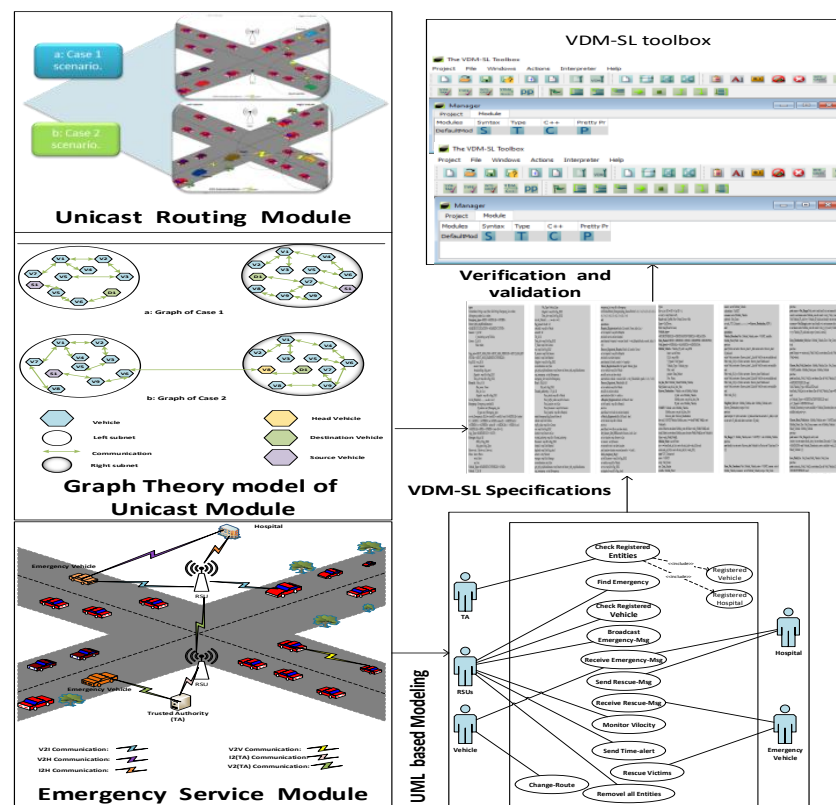


Figure 1. Abstract Representation of the System.

The following are the key contributions of this paper:

- I. We adopt a fog-based protocol and specify this protocol in a formal method approach, which provides emergency services with the arrival of an emergency vehicle at accident locations.
- II. We also adopt a unicast routing protocol to send a message from one vehicle to another, which is focused on V2V communication.
- III. The network of V2V communication will be represented using graph theory.
- IV. Unified Modeling Language (UML) is utilized to model and capture different components of a system.
- V. The formal language VDM-SL is utilized to provide the specification of the system, and a VDM-SL toolbox is used to check the validation of the proposed model.

The rest of this paper is organized as follows. In Section 2, we present the related work. We proposed system methodology and algorithms in Section 3. Section 4 presents the formal methods and formal specifications of the system. The model analysis is provided in Section 5. The conclusion of this paper is presented in Section 6. Some future challenges are present in Section 7.

## 2. Related Work

This section discusses efforts to avoid traffic collisions and accidents, including an emphasis on the need to find an ambulance near the accident site. Through dispatching an ambulance to the scene of a collision, wounded patients will be sent to hospitals as soon as possible. The literature analysis is split deeper into the various suggested schemes for delivering emergency care with the least amount of delay. In addition, the analysis of different scenarios of unicast routing for VANETs is performed by utilizing different approaches.

### 2.1. IoT and VANETs

VANETs are made up of vehicle-to-vehicle as well as vehicle-to-infrastructure communications focused on vehicles with roadside units, and they serve a range of essential applications for smart transportation systems. In order to provide effective broadcast services for VANETs, medium access control (MAC) is essential [12]. In VANETs, the IoT is commonly considered for use in smart transportation systems. The widespread use of various sensors in intelligent vehicles, in particular, has opened up intriguing chances for improving VANET routing efficiency [13]. In Ref. [14], the authors introduced the concept of IoT and VANET in an accident detection system in a smart city. The proposed model reduces the problem of medical assistance for patients. The authors proposed an intelligent vehicle network system in [15] for smart cities, which requires ad hoc path choice based on real-time data collected from nearby vehicles. They use Android-based Wi-Fi direct-enabled smartphones as integrated devices in cars. To introduce an intelligent transportation system, they used a VANETs architecture. The IoT connects thousands of objects, which may be things related to human or non-living objects. As transport networks involve the use of energy, increased efficiency, and waste reduction with the support of IoT [5]. Communication becomes secure using IoT-based VANETs and also reduced road accidents. IoV by integrating IoT with VANETs uses real-time environment data exchange between V2X through wireless communication devices based on fog computing or edge computing the cyber-physical systems' application [6].

### 2.2. Message Dissemination and Authentication Scheme

In [16], the automobiles are classified into clusters based on their driving directions. The information processing for each cluster is the responsibility of the cluster head. The bring and forward scheme is used by nodes that receive the emergency call. According to simulation results, the packet propagation ratio rises as the number of nodes rises. To facilitate the use of Long-Term Evolution (LTE) networks in data transfers, many strategies for managing VANETs have been proposed. Clustering cars and organizing the network through clusters is one of the most general and successful approaches [17]. In this paper, the

authors introduce an algorithm for disseminating reports about real-time traffic situations in VANETs. Using this process, each vehicle decides when to distribute reports, how many to distribute, and which reports are distributed on a local level [18]. The authors propose a distributed clustering-based vehicle-to-roadside communication protocol, in which vehicles join clusters using a coalitional game approach, and stable clusters are built using a fuzzy logic algorithm based on multiple metrics of vehicle velocity, moving pattern, as well as signal quality between vehicles. Each vehicle is guided by a reinforcement learning system that uses game theory to determine the most efficient path for overall network performance [19]. Vehicles are grouped with sensors, as well as sinks are placed along the path to collect or/and transmit data to the sensor, according to a draft authentication protocol for an autonomous vehicle system. The customer keeps track of the vehicles by gathering data from sinks with analyzing it before acting [20]. The transmitting of messages in an open-access environment, such as the VANET, raises the most serious and difficult security concerns. Protection in the VANET includes authentication, data confidentiality, data privacy, data availability, and nonrepudiation. The authentication schemes in VANET are adopted because they play a vital role in safe communication. The major task of the paper is to include a taxonomy of authentication systems, as well as to review their methods, benefits, drawbacks, efficiency, and scope [21]. The authors suggest a randomized authentication protocol based on homomorphic encryption that allows each vehicle in a VANET to create any number of authenticated identities in order to achieve complete anonymity. According to the suggested protocol, cars, including peer vehicles, transportation services, authentication servers, and other devices, cannot be managed by a single entity [22]. In this paper, the existence of resource-constrained nodes should be addressed by a robust authentication approach. To achieve both of these goals, the authors propose a lightweight multi-factor authentication and privacy-preserving protection solution for VANETs. It employs a combination of physically unclonable functions (PUF) and one-time complicated pseudo-identities as authentication variables. Apart from this, by decentralizing the certificate authority's (CA) wide precinct into regional domains as well as gaining comprehensive control of domain keys, it removes the strong reliance on the device key [23].

### 2.3. Emergency Vehicle's Route Clearance

In the modern age, a lot of accidents occur every day due to rapid motorizing. It is a big challenge to identify the accident location to provide emergency services, such as an ambulance or first aid. If there is a lot of traffic, rescue services cannot get to their target on time, which cause the loss of lives. Recently, the authors suggested a visual emergency vehicle priority system for VANETs based on sensors [24]. The auditory sensors connected to RSUs detect the existence of an emergency vehicle as well as measure the distance by using various distance measurement algorithms from the road intersection in the proposed scheme. The information is submitted to the traffic control center, which will change the signals of traffic to get emergency services out of the congested traffic lane as quickly as possible. The authors suggested a signaling scheme to clear the way for an ambulance or other emergency services [9], in which the green light is turned on for the emergency vehicle's path and does not turn red before the emergency vehicle exits the route. In [25], the authors proposed an automatic accident detection prototype model using IoT and VANET Network deployed in the vehicle with the help of sensors that detect accidents as well as emergency levels. The proposed model supports delivering medical aid at the correct time after detecting the accident. The main objective of the internet of things and VANETs is to prioritize emergency vehicles. The vehicular sensor networks (VSNs) concept in an IoT-based vehicular paradigm, focusing on the security aspects and, authors discussed the VSN's design features for reliability, relevant communication technologies, robustness, and their security concerns [26].

Advanced technology collaboration can help to decrease fatalities, save lives, and handle emergency situations. Drones, IoT devices, the ad hoc network, the SAR team,

and the disaster center unit are all part of the planned network architecture in [27]. Their proposed scenario is appropriate for all stages of emergencies. Drones with IoT capabilities play a critical role in disaster preparedness. Sensors on the drone board, for example, are utilized to gather physical data such as water level, vibration, and displacement in a disaster region, as well as transmit data to the disaster center unit. The camera is also used to make forecasts of the impacted region. The network connectivity, i.e., latency, load, and throughput, is used to evaluate performance.

#### 2.4. Routing-Based Scheme for Accident Prevention and Avoidance

This high volume of traffic raises a number of issues, the most serious of which is the effective distribution of emergency communications. Given the dynamic nature of VANETs, one of the most challenging jobs is distributing the message across the network. The transmission storm problem, the hidden node problem, and packet collision are the three key issues. Researchers have suggested several models to address several forms of emergency alert propagation scenarios. This paper not only addresses numerous suggested approaches focused on IoT, intelligent transportation systems, fog computing, priority signaling, software-defined networks and clustering strategies but also explores several new contributions to emergency message propagation in VANETs. They have attempted to investigate the most recent innovations in emergency message distribution over 5G networks [33]. In Ref. [34], the authors recommend combining the fog node with all other organizations on the path to improve the vehicle's capabilities in the event of an emergency. Emergency vehicles, fire trucks, workplaces, hospitals, shops, relatives, mechanics, any stations, protection forces, and those that are directly or indirectly involved with fog-enabled RSUs, vehicles as well as traditional RSUs, are all integrated into the fog-based solution to provide greater convenience, safe travel, and a smooth driving. The critical literature reviews about a routing-based scheme for accident prevention are described in Table 1.

**Table 1.** Literature reviews of Routing-based scheme for accident prevention.

Reference	Year	Tools/Approaches	Advantages of Methods	Description	Disadvantages and Limitations
[16]	2017	NS-2	<ul style="list-style-type: none"> <li>Results can be quickly obtained and tested easily</li> <li>Complex scenarios also tested easily</li> </ul>	<ul style="list-style-type: none"> <li>Reduced accidents on highway and also reduced end-to-end delay</li> </ul>	<ul style="list-style-type: none"> <li>No formal verification and validation</li> <li>To model real system is complicated</li> </ul>
[28]	2017	NS-3 SUMO	<ul style="list-style-type: none"> <li>Contains an abundance of modules</li> <li>Provide point-to-point wireless connection between a network</li> </ul>	<ul style="list-style-type: none"> <li>Delivered the current condition of victims before arrival the victims in hospital with audio and video.</li> <li>Under three evaluations parameters comparative analysis of the standards IEEE 802.11</li> </ul>	<ul style="list-style-type: none"> <li>The required quality of service QoS cannot achieved</li> <li>Not calculate the accuracy of the model utilizing formal methods</li> </ul>
[24]	2016	GPS module IDLE Arduino	<ul style="list-style-type: none"> <li>Broad range of libraries and low cost</li> <li>Easily used not need to external expertise</li> </ul>	<ul style="list-style-type: none"> <li>Hospital delivered an emergency vehicle at the accident area to provide medical services at shortest time</li> </ul>	<ul style="list-style-type: none"> <li>Ignored traffic congestion at accident area.</li> <li>No experimental result</li> <li>No Formal verification and validation</li> <li>Required too much efforts in scheduling</li> </ul>

Table 1. Cont.

Reference	Year	Tools/Approaches	Advantages of Methods	Description	Disadvantages and Limitations
[29]	2015	NS-2	<ul style="list-style-type: none"> <li>Results can be quickly obtained more ideas can be tested easily in shortest time</li> </ul>	<ul style="list-style-type: none"> <li>In VANET cluster delivered the emergency alert message in shortest time.</li> </ul>	<ul style="list-style-type: none"> <li>No Formal verification and validation</li> <li>Bugs are unreliable and to model real system is complicated</li> </ul>
[30]	2015		<ul style="list-style-type: none"> <li>Centralized management for network</li> <li>provide security</li> </ul>	<ul style="list-style-type: none"> <li>Create a framework for VANETS which is beneficial for both fog-computing and SDN</li> </ul>	<ul style="list-style-type: none"> <li>Formal verification and validation is not provided</li> <li>Significant amount of latency and lake in maintenance</li> </ul>
[31]	2015	SUMO NS-2 MOVE	<ul style="list-style-type: none"> <li>Not required costly equipments and easily tested in short time</li> </ul>	<ul style="list-style-type: none"> <li>Reached emergency vehicle at shortest time</li> <li>Delivered message from emergency vehicle to traffic signal</li> </ul>	<ul style="list-style-type: none"> <li>No comparative performance result</li> <li>Simulation based and not find the correctness of the system model</li> </ul> Complicated structure, lightly maintained and not seen significant development
[32]	2019	NS-2, VanetMobiSim	<ul style="list-style-type: none"> <li>Support multiple protocols and many algorithms support in routing and queuing</li> </ul>	<ul style="list-style-type: none"> <li>Multi-hop-clustering approach is proposed by using BFS algorithm</li> <li>Improve the VANET's performance</li> <li>Minimize the rate of message control</li> </ul>	<ul style="list-style-type: none"> <li>Simulation based</li> <li>Not calculate the accuracy of the model utilizing formal methods</li> <li>Complicated structure and unreliable bugs</li> </ul>

### 2.5. Related Work of Unicast Protocols

A unicast protocol is proposed in [35], which is specifically for VANETs. For the sake of understanding the problem, the whole network is constructed using graphs. Formal Methods are used to characterize the proposed model. Formal Methods (FM) is a novel statistical approach for proving the model's correctness and accuracy. For VANETs, various routing protocols have been developed in an attempt to increase the network's routing options [36]. They will be able to build new driver or passenger-oriented services and improve road safety according to VANET. The authors propose a novel condition-based communication-based routing method for extremely complex networks in this article. Instead of sending addresses, a message is sent with certain retransmission or receipt conditions. They demonstrate that thanks to the complex assessment of the situations, this approach can effectively accommodate the high dynamic of vehicular networks [37].

### 2.6. Literature Review of Formal Methods

The formal methods are implemented in different areas of life to measure the model accuracy as well as increase the efficiency of the model. In life-critical tasks, formal methods provide the errors free model before the implementation of the system model in a real-time environment. Formal methods utilized in every life-critical task such as border protection system [38] in which the authors proposed wireless sensors and actor networks (WSANs)-utilizing formal methods, railway interlocking system [39], in which formal analysis of safety properties are presents about moving block interlocking, the authors proposed smart healthcare system utilizing formal methods [40], and guidance system with intelligent traffic monitoring system [41] also utilized formal methods. In Refs. [42,43], the authors proposed MAHSNs that used VDM-SL to show the working of the system as well as verify the accuracy of the model. In Ref. [35], the authors suggested a smart algorithm for an efficient, structured unicast routing protocol. The authors only focused on V2V communication, not on V2I communication. In Ref. [44], the authors used VDM-SL

to model and monitor the AODV traffic-based flooding mechanism for mobile ad hoc networks systems.

In the field of software engineering, system validation is a major problem. There is a danger of loss of life if there is an error in the safety and security crucial system's specification. In most earlier studies, costly computer-based simulation approaches were utilized to assess models. Simulation approaches fall short of establishing a model's total accuracy because the number of test cases necessary to achieve the appropriate degree of confidence grows exponentially. The purpose of this study is to provide an efficient IoT-based formal model of vehicle life-interaction utilizing VANETs. Various approaches are used in our model in which UML is used to define the flow of the system. Wireless sensors and actors are used for monitoring and taking action as required. The formal method in terms of the VDM-SL is used to define the system formally, and the VDM-SL toolbox is used to define validation as well as analysis of the proposed model.

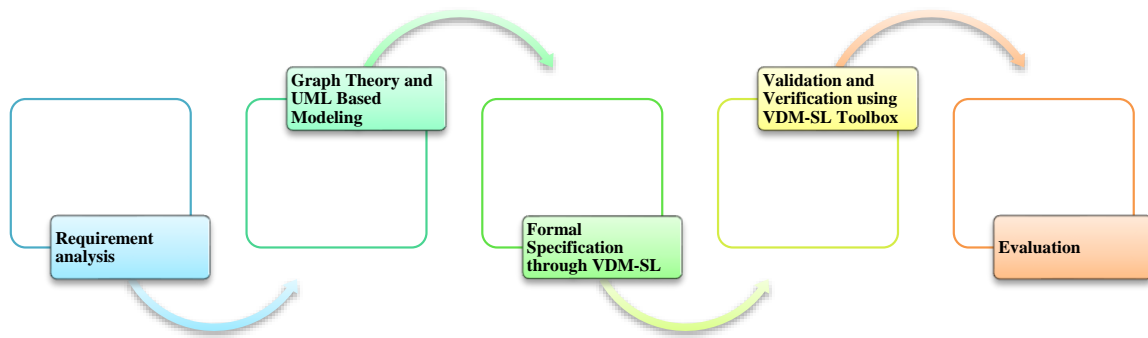
### 3. Methodology

Worldwide, road accidents are one of the main problems that become the reason for injuries as well as deaths due to fast motorization. Passengers, drivers, and systems face a lot of issues such as privacy, security, robbery, and payment due to our present transportation systems. The main cause of these issues in this system is a lack of infrastructure, which includes a lack of roadside sensors, automotive safety features, connectivity, and internet access. According to the literature assessment, many V2V-based ways of dealing with emergency circumstances produce congestion, and as a result, an emergency response does not come on time, potentially resulting in death. While several protocols rely solely on vehicle-to-roadside units (V2I) communications to alleviate vehicle congestion, they must consider network bandwidth, which can cause network congestion and an increase in deaths [45]. Furthermore, several researchers proposed the idea of SDN and fog computing, in which information is communicated to nearest vehicles by using V2V and V2I communication. Most of the work is simulation-based, not finding the correctness of the system model. In Refs. [46,47], The authors presented a fog computing-based protocol for providing emergency services with minimal latency in emergency situations. Their main focus was on V2I communication, not on V2V communication, and simulation-based, which did not provide the accuracy of the model. So, we have built a comprehensive formal model of Vehicle-life interaction using VANETs and formal logic techniques.

#### 3.1. Architecture of the System

Many design decisions made in nature are related to system architectures; these decisions are generally commercial and nontechnical. We are building an efficient IoT-based formal model for vehicle-life integration allowing fog-based roadside units (RSUs) with the use of several techniques such as VANETs, graph theory, UML, as well as VDM-SL formal methods in this study. Figure 2 depicts a flow of the proposed techniques: After analyzing the system's requirements, IoT will be adopted in which sensors will be used to monitor the real-time environment, and actors will be used to take action in case of violation. Then UML diagrams will be used to describe the informal behavior of the model. Because of its use as a data format and computing capability, graph theory is frequently used. VDM-SL may also be used to turn a graph-based model into formal specifications. The formal model will be converted into a verified model using the VDM-SL toolbox. Two separate modules are incorporated in our suggested model: a unicast routing module for V2V communication and an emergency service module.



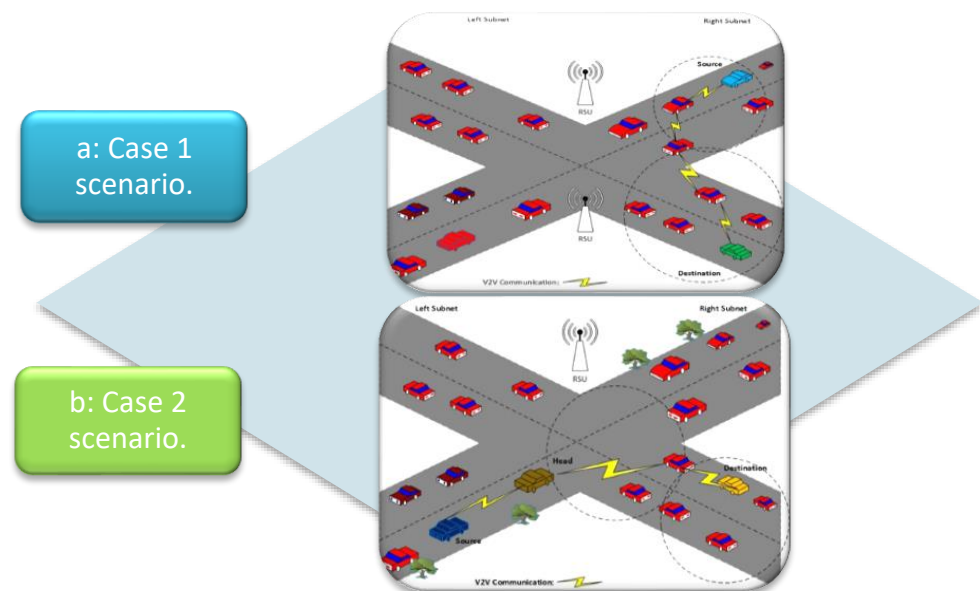


**Figure 2.** The Flow of System Methodology.

### 3.2. Unicast Routing Module for V2V Communication

A data packet that can be easily transmitted from only one source to a single target in a network is known as unicast communication. With known destination topology, the developed protocol handles unicast mode, which means that the destination's position is known and determined using partitioning. In the case of V2V communication, there are several options for a destination spot. The choices for V2V communication utilizing unicast protocol are as follows: (1) In source vehicle as well as a destination vehicle in the single subnet either left subnet or in the right subnet. (2) If the source and destination cars are on different subnets, the destination vehicle must be in the right subnet if the source is in the left, and vice versa.

Figure 3 shows the above-mentioned scenarios. In case 1 scenario 'a', relay node establishes wireless connections between the source and destination nodes by carrying packets from the source and passing them on to the destination. Since the destination node's location is identified, only certain nodes with the same lane flags as the destination vehicle are involved.



**Figure 3.** V2V Communication, Unicast Scenarios of (a) and (b) Cases.

In Figure 3 case 2 scenario 'b', depicts V2V communication between the source node as well as the target node in the two halves of the network via a head node. We will show you how to choose a head vehicle. In scenario two, the head node creates several zones to pick a peripheral and found target vehicle.

The given Algorithm 1 demonstrates a procedure about how to communicate a vehicle with another vehicle in a network called VANETs. The unicast procedure's algorithm is seen below. The algorithm's first step is to set up the proposed topology for the vertically partitioned network. The addresses of the vehicles engaged in the correspondence are then updated in the 'str' and 'dest' variables. "Vehicle-Direction" would be labeled for both the source and destination vehicles to determine their location. The source vehicle will use "Neighbour-Search" to decide if the source and destination vehicles are in the single subnet. If the "Neighbour-Search" function returns TRUE, a 2-hop scope is established, then the destination is searched within the scope, and if the destination vehicle is located in the scope, the path from source to destination is registered, as well as an acknowledged message is sent to the root vehicle that is the start point. If the destination vehicle is not identified in the scopes, a peripheral vehicle is selected at the edge of the previous scope and makes a loop on the procedure mentioned in the seventh point until the destination is not found. If the algorithm "Neighbour-Search" returns FALSE, the head is a car that is heading into the middle. If the target vehicle is in the left subnet, a scope is generated for that subnet, and if the destination is located within that scope, the route is registered, and an acknowledged message is received. If the destination cannot be found within that scope, step 17 will be repeated before the destination can no longer be found.

---

**Algorithm 1:** V2V Communication using Unicast Algorithm
 

---

```

1. Procedure Unicast
2. Input: V2V Comm (Communication)
3. if Veh1 Comm Veh2 then
4.   Str ← Veh1 & Dest ← Veh2
5.   Vehicle-Direction (KDT, Str, VB, Middle) & Vehicle-Direction (KDT, Dest, VL, Middle)
6.   Neighbour-Search (Str, Dest, L_S, R_S)
7.   if Neighbour-Search (Str, Dest, L_S, R_S) == TRUE
8.     2H- Zone-Creationn (Str, Subnet)
9.     Scope-Destination-Srch(Dest, Scope)
10.    if Scope-Destination-Srch (Dest, Scope) == TRUE
11.      Save-Path (Dest, Str, Scope)
12.      Acknowledged(Path)
13.      EXIT
14.    else if Scope-Destination-Srch (Dest, Scope) == FALSE
15.      Choose-Peri-Veh (Scope)
16.      Loop from step 8 to step 9
17.   if Neighbour-Search (Str, Dest, L_S, R_S) == FALSE
18.     Choose-Head (Str, Middle)
19.     Move hv to Middle
20.     if Str in set L_S
21.       2H_Zone-Creation (hv, R_S)
22.       if Scope-Destination-Srch (Dest, Scope) == TRUE
23.         Save-Path (Dest, Str, Scope)
24.         Acknowledged(Path)
25.         EXIT
26.       else if Scope-Destination-Srch (Dest, Scope) == FALSE
27.         Choose-Peri-Veh(Scope)
28.         Loop from step 21 to step 26
29.     else if Str in set R_S
30.       2H_Zone_Creation (hv, L_S)
31.       if Scope-Destination-Srch (Dest, Scope) == TRUE
32.         Save-Path (Dest, Str, Scope)
33.         Acknowledged(Path)
34.         EXIT
35.       else if Scope-Destination-Srch (Dest, Scope) == FALSE
36.         Choose_Peri_Veh(Scope)
37.         Loop from step 31 to step 35
  
```

---

### Graph Theory Model of Unicast Module in V2V Communication

The complete network is represented using graph theory, as well as to create structural models. It depicted the whole network of unicast routing modules (nodes) and the connection of things (edges) to build a network.

Figure 4 shows how graph-based topology improves the speed efficiency of vehicles between source and destination junctions, allowing each vehicle to reach its destination without wasting time, delays, or fuel. This model shows the two scenarios of V2V communication (scenario ‘a’ and scenario ‘b’). In first scenario both source and destination in a single subnet. They can communicate easily without involving any head node. However, in the other scenario of the unicast module, both source and destination vehicles at the different subnet via a head node. We will show you how to choose a head vehicle. In scenario two, the head node creates several zones to pick a peripheral and found target vehicle. VDM-SL may also be used to turn a graph-based model into a formal model.

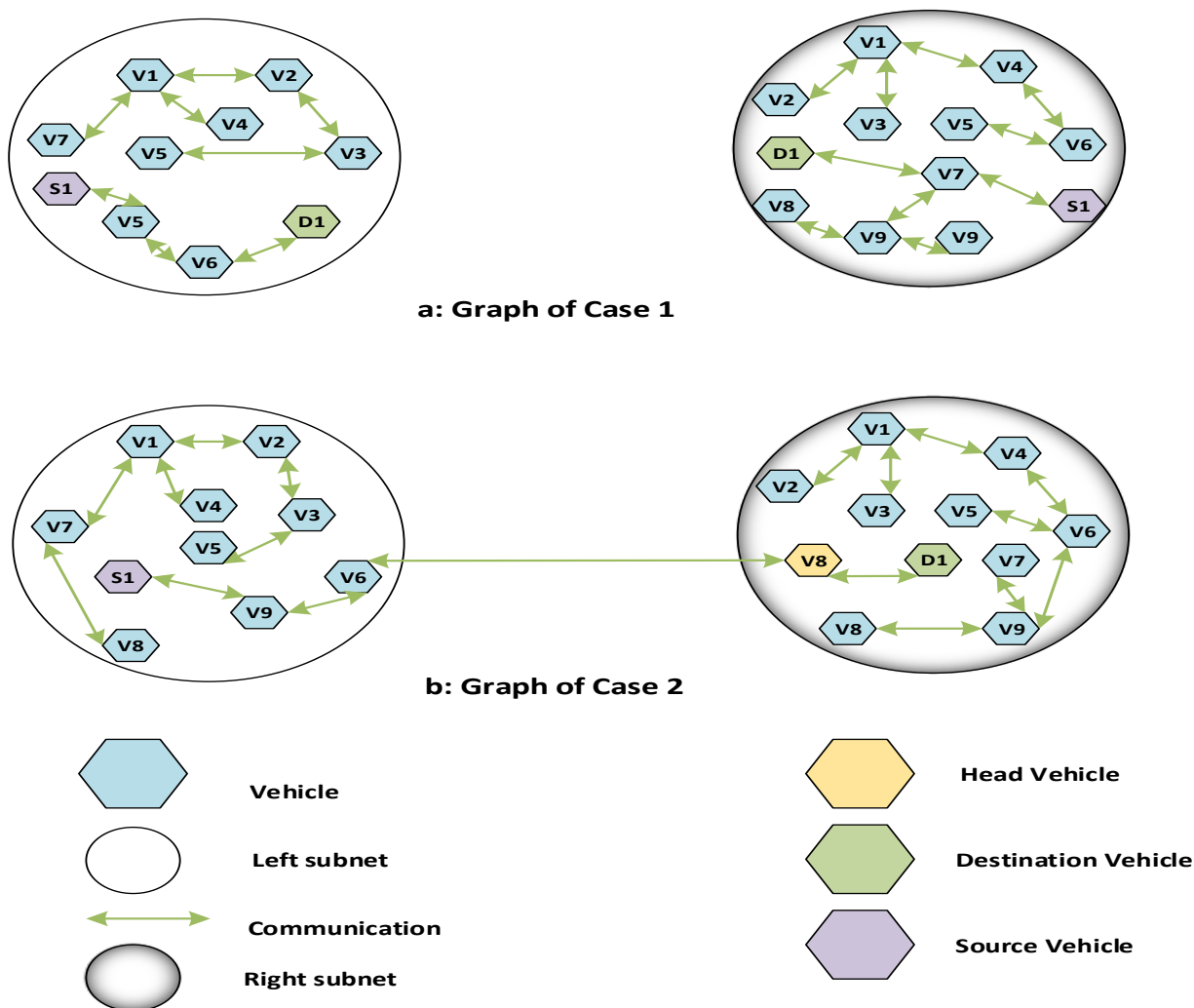


Figure 4. Graph-based Model of Unicast Topology of Two Cases (a) and (b).

### 3.3. Emergency Service Module

The warning alert is broadcast to the closest fog-based RSU as well as the vehicle in the situation of any emergency occurrence. Fog-based RSU sent an emergency alert to the hospital, along with path and location information. The hospital then double-checks its services before sending a rescue call to the emergency vehicle. The ambulance vehicle examines the direction as well as the destination ID and route information for the crash. Figure 5 depicts the fog-based RSU’s message distribution. The fog-based RSU has sent out

an alert message to other fog-based RSUs and the vehicles, instructing them to change their routes so that the path could be safe for the arriving emergency car to arrive on time at the crash scene and rescue the victims. This further decreases the possibility of future mishaps. When an emergency/accident happens, Algorithm 2 sends out an emergency response and checks the message from it. If the message is classified as an emergency, the fog-based RSU sends a message to the other fog-based RSUs. Other drivers can quickly change lanes as a result of this, and it also unicasts a warning to the registered ambulance and then alerts the emergency vehicle to save the patients. After the rescue procedure, the fog-based RSU received a message from the hospital. Then the fog-based RSU shows the normal flow of traffic by removing all entities. The formal method in terms of the VDM-SL is utilized to define the system formally, and the VDM-SL toolbox will be used to describe validation and analysis of the proposed model in given sections.

### UML Based Model of Emergency Service Module

The Unified Modeling Language (UML) is widely used, and it contains a semi-formal style that may be used to depict system requirements in detail. At the beginning of the development process, the UML is the recommended modeling language for system portrayal. It is possible to achieve first stage model functional performance evaluation using UML and formal models by combining formal techniques and formal models with UML. If the erroneous system performance is not handled quickly, it can lead to a slew of additional issues that obstruct system function and, as a consequence, the system ends up costing a lot of money.

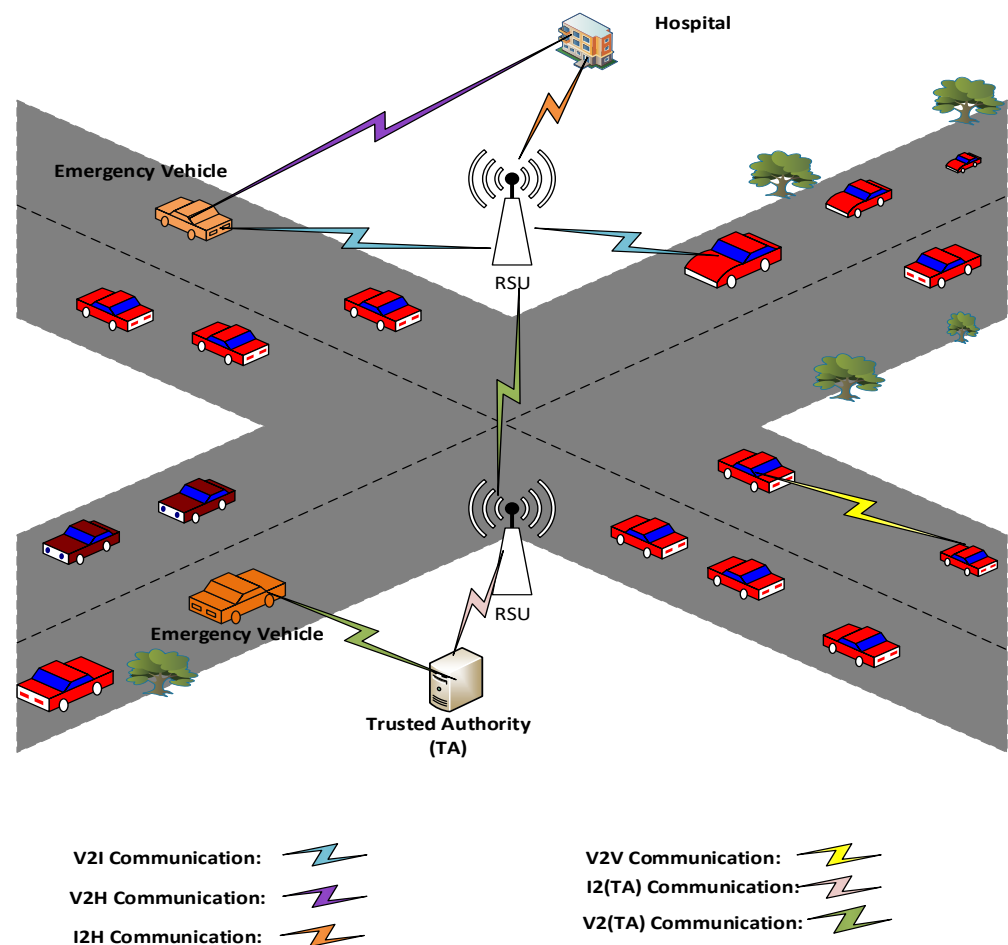


Figure 5. Emergency Case scenario.

**Algorithm 2:** Broadcast Emergency Message Algorithm

---

```

1. Procedure System Communication (Fog-RSU, Vehicle, Hospital);
2. Input: Emer(Emergency);
3. if Message = TRUE then
4.     The Fog-RSU checks a V-id;
5.     if a V-id registered then
6.         if Msg-type = Emer then
7.             Broadcast Emer_Msg to Vehicles & Fog-RSUs;
8.             RSU sends Emer_Msg to Hospital;
9.             if Hospital = registered then
10.                RSU sends Emer_Msg to the Hospital;
11.                Hospital checked Msg-type & V-id;
12.                Hospital covey Rescue-Msg to Emer_Vehicle;
13.                if Emer_Msg = Emer_Vehicle then
14.                    Emer_Vehicle extract Rescue_Msg;
15.                    RSU monitors velocity & wait for some period;
16.                    if RSU send Time-alert, after some Time_period then
17.                        Emer_Msg Broadcast to Emer_Vehicle when it is in the range
of RSU;
18.                    else
19.                        Do Nothing;
20.                    end
21.                else
22.                    Emer_Msg Broadcast to other vehicle to change their route;
23.                end
24.                else
25.                    Rescue_Msg extracts by Emer_Vehicle;
26.                    Monitors velocity by RSU also wait for some Time-period;
27.                end
28.            else
29.                Emer-Msg Broadcast to other vehicle to change their route;
30.                Victims Rescue by Emer_Vehicle;
31.                if Rescue_Msg = TRUE then
32.                    RSU Remove all entries;
33.                    Rescue_Msg send to previous RSU and Vehicles;
34.                else
35.                    Send to Trusted-Authority
36.                end
37.            end
38.        else
39.            Send to Trusted-Authority
40.            if V_id is registered then
41.                Send verify msg to RSU;
42.            else
43.                Contact to Trusted-Authority;
44.            end
45.        end
46.    else
47.        Do Nothing;
48.    end

```

---

Using the UML, a use case diagram may describe the specifics of the system's users and their interactions with it. The four primary components of the use case diagram are the system boundary, the actor, the association, and the use case; also, the use case diagram depicts the link between actors and use cases. Figure 6 shows a use case diagram that demonstrates the system's capabilities. The RSU, TA, hospital, vehicles, and emergency vehicles are the characters. RSUs are used to find emergency services if an emergency

occurs, broadcast emergency messages, send a time alert message as well as remove all the entities after the rescue procedure is completed. Trusted authority (TA) is used to check the authorized entity in the system, including vehicles and hospitals. The hospital received an emergency occurrence message from RSU and then sent a rescue message to the emergency vehicle. After the rescue procedure is complete, it informs the RSU. The emergency vehicle is used to rescue the victims from the emergency occurrence location and then inform the hospital.

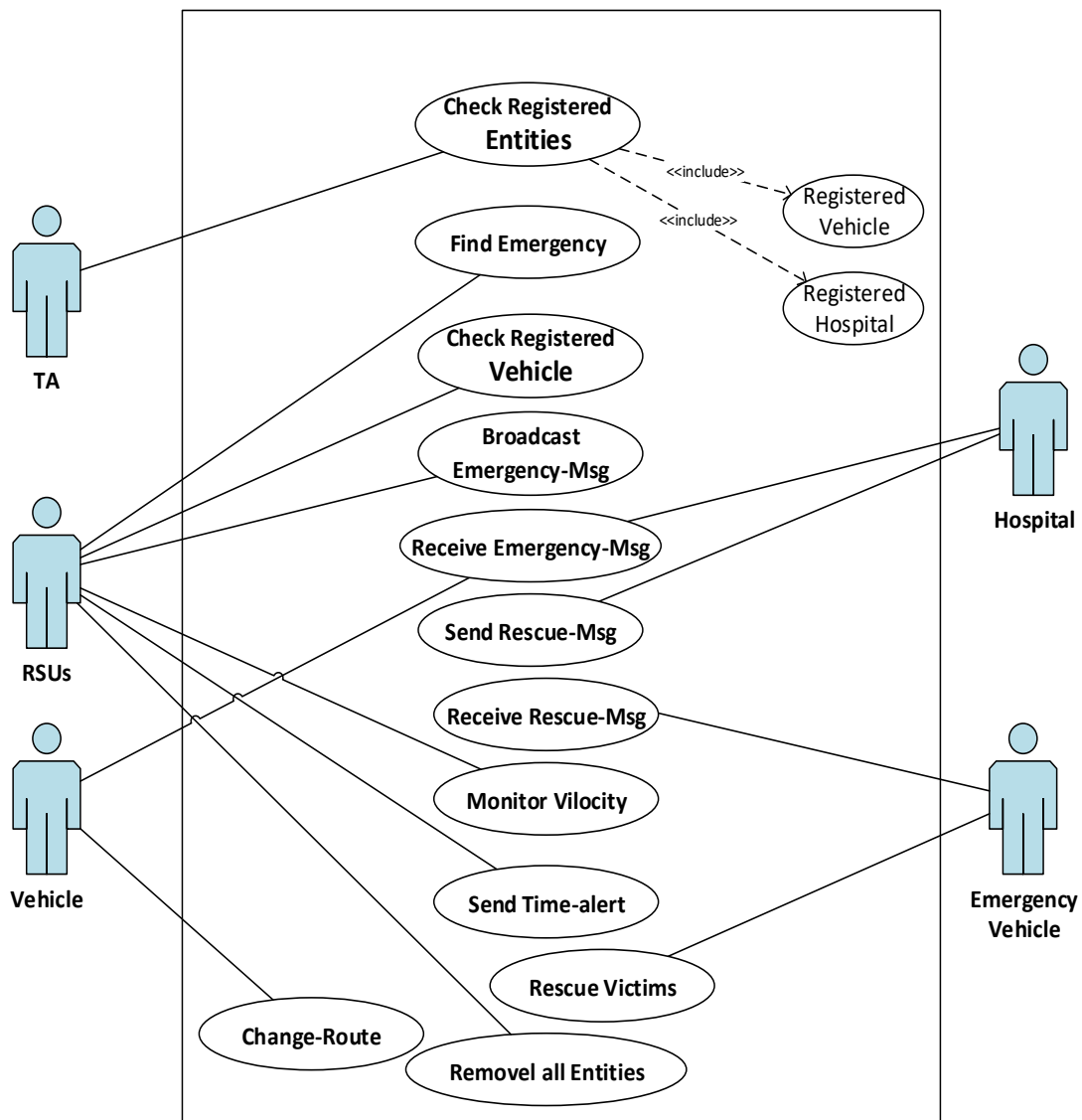


Figure 6. Use case Diagram of Emergency Service Module.

4. Formal Methods and Formal Specifications

The VDM-SL specification of the proposed Algorithm given below, in which different objects, operations, and functions described.

4.1. Unicast Routing Module’s Specifications for V2V Communication

The Vehicle\_type, Veh\_Speed, and Data\_Packet are declared as enumeration types. The position declared as Point up and down declared as Bit and Id as a token. A composite object is MobileVehicle that has eight fields, i.e., Vehicle\_IP, limit, R\_Id, V\_speed, Vehicle\_type, Dist, packet, and Pos.

---

**types** Bit = a inv BT == BT = 0 or BT = 1; a = int; b = int; Point = a\*b; Road= real; Up=Bit;  
 Pos = Point; Down = Bit; Lane = Up\*Down; Ad\_hoc\_Net = Mobile\_Vehicle\*Mobile\_Vehicle; Rdd= map  
 Road to Lane; Net\_Conn = seq of Ad\_hoc\_Net;  
**Vehicle\_type**= <SOURCEVEHICLE>|<DESTINATIONVEHICLE>|<RELAYED>;  
**Data\_Packet**= <SENT>|<RESEND>|<SEND>|<DROPPED>|<RECEIVED>;  
**Veh\_Speed** = <NORMAL>|<MAXIMUM>|<MINIMUM>;  
**MobileVehicle** :: Vehicle\_IP\_Add : seq of Bit  
 limit : set of Point  
 R\_Id : seq of Bit  
 V\_Speed : Veh\_Speed Vehicle\_Type : Vehicle\_type  
 Dist : real  
 packet : Data\_Packet  
 Pos : Point;

---

‘Known Destination’ has four fields i.e., Vehicle, L\_link, Mobile\_conn, and R\_link which is a composite object.

---

**KnownDestination** :: Vehicle : set of Mobile\_Vehicle  
 L\_link : set of Mobile\_Vehicle  
 Mobile\_conn : seq of Ad\_hoc\_Net  
 R\_link : set of Mobile\_Vehicle;

---

VANET has three fields i.e., Vehicle, Mobile\_conn, and Known\_dest, which is a composite object, and the middle point is declared as a set of points.

---

**VANET**:: Vehicle : set of Mobile\_Vehicle  
 Mobile\_conn : seq of Ad\_hoc\_Net  
 Known\_dest : Known\_Destination  
 inv mk\_VANET(Vehicle, Mobile\_conn,-) == forall Va, Vb in set Vehicle & exists Mnet in set elems  
 Mobile\_conn & Mnet = mk\_( Va, Vb) and forall Mnet in set elems Mobile\_conn & exists Va, Vb in set  
 Vehicle & Mnet = mk\_( Va, Vb); Middle\_Point = set of Point inv x == forall mk\_(a, b) in set x & mk\_(a,b)  
 = mk\_(0,b) and exists mk\_(a1,b1) in set x & mk\_(a1,b1) = mk\_(0,0);

---

The formal specification of the system’s state is shown below. The state incorporates all of the system’s necessary attributes that will be used in the unicast process, and the state’s invariant will be used to communicate.

---

**state** V2V\_Unicast of  
 netw : VANET  
 scop : Net\_Conn  
 res : Data\_Packet  
 middle : Middle\_Point  
 nearer : set of MobileVehicle  
 subnetbase : VANET  
 nonearer: set of MobileVehicle  
 pathveh : Net\_Conn  
 inv mk\_V2V\_Unicast(-,-,-,-,-,-,-) == KnownDestinationV2V () end

---

There are various operations used in the unicast routing protocol. In known destination topologies, the vehicle direction operation is used to change the vehicle’s position in the header. Vehicle Direction operation is used to find the exact as well as current location in V2V communication.

---

*operations*

*VehicleDirection*(Veh : MobileVehicle, netw : VANET, middle : Middle\_Point) Path : Lane

*pre true*

*post* (Veh in set netw. KnownDestination. L\_link union netw. KnownDestination. R\_link and exists Veh in set netw. KnownDestination. L\_link & Veh.Pos in set middle and Path = mk\_(1,0)) or (Veh in set netw. KnownDestination.Vehicle and exists Veh in set netw. KnownDestination. L\_link & Veh.Pos not in set middle and Path = mk\_(0,1)) or (Veh in set netw. KnownDestination.Vehicle and exists Veh in set netw. KnownDestination. R\_link & Veh.Pos in set middle and Path = mk\_(1,0)) or (Veh in set netw. KnownDestination.Vehicle and exists Veh in set netw. KnownDestination. R\_link & Veh.Pos not in set middle and Path = mk\_(0,1));

---

The ‘Neighbor Search’ operation is utilized to search for a source as well as a destination in which the source and destination are scanned to see whether they are in the same subnet or not. When the start node and target node both are in the same subnet (it may be right subnet or left subnet), then the algorithm will return true; otherwise, it will return false.

---

*NeighborSrch* (str : MobileVehicle, dest : MobileVehicle, netw : KnownDestination) output : bool

*pre true*

*post* output <=> (str in set netw. L\_link and dest in set netw. L\_link) or (str in set netw. R\_link and dest in set netw. R\_link);

---

The function NetRange determines which of the input node’s neighbors are within its coverage.

---

*NetRange* (V : MobileVehicle, netw : VANET) Y : set of MobileVehicle

*pre true*

*post* forall y in set netw. KnownDestination.Vehicle & y.Pos in set V.limit and Y = {y};

---

The protocol for creating a zone is defined in this algorithm. For this routing method, a two-hop scope creation is proposed. The method returns a scope, which is a collection of connections connecting a vehicle to all of its neighbours within its two-hop radius.

---

*ZoneNetCreationn*(Veh : MobileVehicle, netw : VANET, nearer : set of MobileVehicle, nonearer : set of Mobile\_Vehicle) scope : Net\_Conn

*pre true*

*post* nearer = NetRange(Veh, netw) and forall y in set nearer & exists conn1 in set elem netw.Mobile\_conn & conn1 = mk\_( Veh, y) and Veh.Vehicle\_IP\_Add <> y.Vehicle\_IP\_Add and forall y in set nearer & nonearer = NetRange(y,netw) and forall yv in set nonearer & exists conn2 in set elems netw.Mobile\_conn & conn2 = mk\_(y,yv) and y.Vehicle\_IP\_Add <> yv.Vehicle\_IP\_Add and scope = [conn1, conn2];

---

When the destination vehicle exists in the scope, then the below Zone Destination Search function returns true; otherwise, it returns false.

---

*ZoneDestinationSrch*(dest : MobileVehicle, Zon : Net\_Conn) Output: bool

*pre true*

*post* Output <=> exists mk\_(Veh1,Veh2) in set elems Zon & (Veh1= dest) or (Veh2=dest);

---

A scope/zone is passed as an input to the peripheral selection function, which returns a vehicle at the scope’s limit as an output. The peripheral node is a node at the zone’s edge that creates a two-hop scope around the destination node if it is not included in the scope. This vehicle must have a link to the source or be inside the domain, according to the method’s first point. Then, because it will be in that scope for a longer period of time, the slowest car will be chosen as a peripheral.



---

```

ChoosePeriVehZone(dest : MobileVehicle, Zon : Net_Conn, netw : VANET, middle : Middle_Point)
peri : MobileVehicle
pre true
post forall mk_(Veh1, Veh2) in set elems Zon & Veh1.Vehicle_Type = <SOURCEVEHICLE> and
Veh2.Vehicle_Type = <RELAYED> and forall mk_(Veh2, pv) in set elems Zon & Veh2.Vehicle_Type
=<RELAYED> and pv.Vehicle_Type <> <DESTINATIONVEHICLE> and pv.V_Speed = <MINIMUM>
and VehicleDirection( pv, netw, middle) = VehicleDirection(dest, netw, middle) and peri = pv;

```

---

When the start node and target vehicle are not in the single subnet, the head vehicle selection function is employed. The head node is then picked as a node with a high speed that is travelling towards the middle point from the start node subnet. A head vehicle is a node that communicates with the source node.

---

```

ChooseHeadVehicle(srt : MobileVehicle, netw : VANET, middle : Middle_Point, Zon : Net_Conn,
nearer : set of MobileVehicle) Head_Vehicle : MobileVehicle
pre true
post nearer = NetRange(srt, netw) and (exists y in set nearer & mk_(srt, y) in set elems Zon and
y.V_Speed= <MAXIMUM> and VehicleDirection(y, netw, middle) = mk_(1,0) and Head_Vehicle = y);

```

---

Save Path method keeps track of the route from the source to the target node. Essentially, the path from source to target node may be employed in the backtracking technique, which is a destination to source operation.

---

```

SavePath(Zon : Net_Conn) Path_Vehicle : Net_Conn
pre true
post exists mk_(Veh1, Veh2) in set elems Zon & Veh1.Vehicle_Type = <SOURCEVEHICLE> and
Veh2.Vehicle_Type = <RELAYED> and exists mk_(Veh3, Veh4) in set elems Zon & Veh3.Vehicle_Type
=<RELAYED> and Veh4.Vehicle_Type = <DESTINATIONVEHICLE> and forall mk_(Veh2, Veh3) in set
elems Zon & Veh2.Vehicle_Type = <RELAYED> and Veh3.Vehicle_Type = <RELAYED> and Path_Vehicle
= [mk_(Veh1, Veh2), mk_(Veh2, Veh3), mk_(Veh3, Veh4)];

```

---

If a path exists between source and destination, then the acknowledged operation sends an acknowledgment message to the source.

---

```

Acknowledged(pv : Net_Conn) Veh_Msg : Data_Packet
pre true
post exists mk_(Veh1, Veh2) in set elems pv & Veh1.Vehicle_Type = <SOURCEVEHICLE> and
Veh2.Vehicle_Type = <RELAYED> and exists mk_(Veh3, Veh4) in set elems pv &
Veh3.Vehicle_Type = <RELAYED> and Veh4.Vehicle_Type = <DESTINATIONVEHICLE> and forall
mk_(Veh2, Veh3) in set elems pv & Veh2.Vehicle_Type = <RELAYED> and Veh3.Vehicle_Type
=<RELAYED> and (pv = [mk_(Veh1, Veh2), mk_(Veh2, Veh3), mk_(Veh3, Veh4)] and Veh_Msg =
<RECEIVED> or pv <> [mk_(Veh1, Veh2), mk_(Veh2, Veh3), mk_(Veh3, Veh4)] and Veh_Msg =
<RESEND> );

```

---

The given formal specification is for the suggested V2V communication in which all the above functions are used. The complete description of the 'Known Destination V2V' is defined in the methodology section, in which we have described the complete procedure of vehicle-to-vehicle communication.

---

```

KnownDestinationV2V() output : bool
ext wr res : Data_Packet
rd netw : VANET
wr pathveh : Net_Conn
wr nonearer : set of MobileVehicle
rd middle : Middle_Point
wr nearer : set of MobileVehicle
wr scop : Net_Conn
pre true
post exists Veh in set netw.Vehicle & Veh.Vehicle_Type =<SOURCEVEHICLE> and
exists Veh1 in set netw.Vehicle & Veh1.Vehicle_Type =<DESTINATIONVEHICLE> and (NeighborSrch(
Veh, Veh1, netw. KnownDestination) = true and
nearer = NetRange (Veh, netw) and forall vn in set nearer & nonearer = NetRange(vn,netw) and
scop = ZoneNetCreationn (Veh, netw, nearer, nonearer) and (ZoneDestinationSrch (Veh1, scop) = true and
pathveh = SavePath(scop) and res = Acknowledged (pathveh)) or
(ZoneDestinationSrch (Veh1, scop) = false and
exists pv in set netw.Vehicle & pv = ChoosePeriVehZone (Veh1, scop, netw, middle) and
nearer = NetRange(pv,netw) and
forall vn in set nearer & nonearer = NetRange(vn, netw) and
forall vn in set nearer & nonearer = NetRange(vn, netw) and
scop = ZoneNetCreationn (pv,netw,nearer,nonearer) and
ZoneDestinationSrch (pv, scop) = true and pathveh = SavePath(scop) and
res = Acknowledged(pathveh) or res = <DROPPED>) and
output = true) or (NeighborSrch(Veh, Veh1,netw. KnownDestination) = false and
exists hv in set netw.Vehicle & hv = ChooseHeadVehicle(Veh1,netw,middle, scop, nearer) and
scop = ZoneNetCreationn(hv, netw,nearer,nonearer) and
((ZoneDestinationSrch(Veh1,scop) = true and pathveh = SavePath(scop) and
res = Acknowledged (pathveh) or (exists pv in set netw.Vehicle & pv = ChoosePeriVehZone (Veh1, scop,
netw, middle) and nearer = Net_Range(pv,netw) and (forall vn in set nearer & nonearer = NetRange (vn,
netw)) and
scop = ZoneNetCreationn(pv,netw,nearer,nonearer) and (ZoneDestinationSrch(pv,scop) = true and
pathveh = SavePath(scop) and res = Acknowledged(pathveh) or
res = <DROPPED> andoutput = true)))));

```

---

#### 4.2. Emergency Service Module's Specifications

VDM-SL is used to formalize the functionality as well as the specification of emergency messages broadcast by fog-RSU. VDM-SL, a well-known modeling and specification language, is used to specify its properties. The Emergency\_type and Sensor\_info\_anyMiscellaneous are declared as enumeration types. Name, emergency location, isemergency and location declared as token and Id as a string. The hospital is a composite object with four fields: hospital id, name, location, and message alert.

---

##### types

```

Name=token; String = seq of char; Id= String; Loc = token; Emergency_Loc = token; isAny_Emergency
=token; Emergency_type= <FIRE> | <MEDICAL> | <OTHER>; Sensor_info_anyMiscellaneous=
<EMEGENCYLOCATION> | <EMEGENCYTYPE>;

```

```

Hospital :: Hos_id : Id

```

```

Hos_name : Name

```

```

Hos_loc : Loc

```

```

Msgalert : map Id to FogBasedRSU

```

```

inv mk_Hospital(i, -, -, -)== len i <= 8;

```

---

The emergency object is a composite object with three fields, namely, emergency recorded, emergency location, and type. The invariant was also used, which defined the number of emergencies if they occurred. When an emergency occurs, it must check the location and type of emergency.

---

```

Emergency:: Emergency_recorded:Id
E_location: set of Emergency_Loc
E_type: set of Emergency_type
inv mk_Emergency(-,Le,Lt)== (card(Le) = card(Lt)) and (({<MEDICAL>} subset Lt \ {<FIRE>} \
{<OTHER>} or {<FIRE>} subset Lt \ {<MEDICAL>} \ {<OTHER>} or {<OTHER>} subset Lt \
{<MEDICAL>} \ {<FIRE>}) and {<MEDICAL>,<FIRE>,<OTHER>} inter Lt <> {});

```

---

Sensors, FogBasedRSU, Message and Time also defined as composite objects which have different number of fields.

---

```

Sensor :: S_Id: Id
S_monitoring: set of Vehicle;
Msg_sent=<SENT_MSG_RSU>|<SENT_MSG_VEHICLE>|<SENT_MSG_HOSPITAL>|
<SENT_MSG_EMERGENCYVEHICLE>;
FogBasedRSU:: rsu_Id: Id
sensors: Sensors
broadcastMsg: Msg_sent
Msgalert : map Id to FogBasedRSU
Time_alrt: map Id to FogBasedRSU;
Message:: RSU_id: FogBasedRSU
Msg_type: Emergency;
Hours= nat; Mint= nat; Sec= nat;
Time:: hours: Hours
mint: Mint
sec: Sec;

```

---

Enumeration types are specified for Vehicle\_Type. The vehicle object is defined as a composite object with three fields: vehicle id, vehicle type, and message alert.

---

```

Vehicle_Type= <EMERGENCYVEHICLE>|<VEHICLE>;
Vehicle :: T_Id: Id
Veh_Type: Vehicle_Type
Msgalert : map Id to FogBasedRSU
Time_alrt: map Id to FogBasedRSU
inv mk_Vehicle( i, -, - )== len i <= 8;

```

---

The fog cloud is also defined as a composite object that has twelve different fields with different data types. This composite object is used to store the data from fog-based RSUs.

---

```

FogCloud:: Rsuld : Id
vehicleId : map Id to Vehicle
sensorId : Id
TA_id: Id
Time_alrt: map Id to FogBasedRSU
Video_Traffic: map Id to Camera
rsu: map Id to FogBasedRSU
R_sensors: map Id to Sensors
hospital : map Id to Hospital
Msgalert : map Id to FogBasedRSU
currentlocation: set of Loc
get_info_anyMiscellaneous: map Sensors to Sensor_info_anyMiscellaneous
any_emergency : set of isEmergency
emergency_list: map Id to Emergency;
Road :: Rd_Id: Id
Rd_rsu1: FogBasedRSU;

```

---

The Trusted Authority object is a six-field composite object that includes the trusted authority identity, view vehicle, view traffic video, view location, view RSUs sensors, and view hospital. The trusted authority can view the mentioned information using an identifier from the fog cloud.

---

```

TrustedAuthority :: TA_id: Id
View_vehicle: map Id to Vehicle
View_traffic_video: map Id to Camera
View_location: set of Loc
View_Rsusensor : map Id to Sensors
View_hospital : map Id to Hospital;

```

---

State, actions, and reusable functions make up the dynamic portion. The properties established above in the form of composite objects are included in the formal definition of the proposed module's state. As demonstrated below, the function `init` is also used to initialise the composite objects.

---

```

state EmergencyMsg_RescueVictim of
vehicle: map Id to Vehicle
traffic_video: map Id to Camera
rsu: map Id to FogBasedRSU
location: map Sensors to Loc
trusted_authority: map Id to TrustedAuthority
Rsusensor : map Id to FogBasedRSU
hospital : map Id to Hospital
fogcloud: map Id to FogCloud
sensors : set of Sensor
currentlocation: set of Loc
get_info_anyMiscellaneous: map Sensors to Sensor_info_anyMiscellaneous
any_emergency : set of isAny_Emergency
emergency_list: map Id to Emergency
init EM==EM=mk_EmergencyMsg_RescueVictim({|->}, {|->},{|->},{|->},{|->},{|->}, {|->}, {|->},{},
{},{|->},{}, {|->}) end

```

---

In the next part, various operations are defined in the specification.

The first operation of the emergency module is hospital registration which takes hospital id, name, and location as input. `Post_condition`: use the union operator and add the pair of hospital id, name, and location in a map. `Pre_condition`: we have to use the `dom` operator to check that the hospital ID, which we take as an input, is not in the domain of the hospital map.

---

```

operations
HospitalRegistration(hosIn: Id, namIn: Name, locIn: Loc )
ext wr hospital : map Id to Hospital
pre hosIn not in set dom hospital
post hospital = hospital ~ munion{ hosIn |-> mk_Hospital(hosIn, namIn, locIn, {|->})};

```

---

`Post_condition`: used the domain deletion operator in the operation of 'Removed Registered Hospital'. `Pre_condition`: we have used the `dom` operator to check that the hospital ID which we take as input has to be in the domain of the hospital map.

---

```

RemovedRegisteredHospital (hosIn: Id, namIn: Name)
ext wr hospital : map Id to Hospital
pre hosIn in set dom hospital
post hospital = { hosIn, namIn }<-: hospital~;

```

---

The formal specification of the ‘Vehicle Registration’ operation consists of two parameters vehicle id and vehicle type. Pre\_condition: we have used the dom operator to check that the vehicle Id, which we take as an input, is not in the domain of the vehicle map. Post\_condition: use the union operator to add the pair of vehicle id and type in the map.

---

```
VehicleRegistration(inputId: Id, typeIn: Vehicle_Type)
ext wr vehicle: map Id to Vehicle
pre inputId not in set dom vehicle
post vehicle = vehicle ~ munion{ inputId |-> mk_Vehicle(inputId, typeIn, { |-> }, { |-> })};
```

---

The formal specification of the ‘Remove Registered Vehicle’ operation consists of one parameter, which is vehicle id. Pre\_condition: used the dom operator to check that the vehicle Id which we take as input has to be in the domain of the vehicle map. Post\_condition: we have used the domain deletion operator.

---

```
RemoveRegisteredVehicle(inputId: Id)
ext wr vehicle: map Id to Vehicle
pre inputId in set dom vehicle
post vehicle = { inputId }<-: vehicle ~;
```

---

The ‘isHospital Registered’ operation checks whether or not the hospital is registered.

---

```
isHospitalRegistered(hosIn: Id) Result : bool
ext rd hospital : map Id to Hospital
pre true
post Result <=> hosIn in set dom hospital;
```

---

The ‘isVehicle Registered’ operation checks whether or not the vehicle is registered.

---

```
isVehicleRegistered(idIn: Id) Result : bool
ext rd vehicle: map Id to Vehicle
pre true
post Result <=> idIn in set dom vehicle;
```

---

This operation gave us the exact number of emergency vehicles that are registered.

---

```
getNoOfRegisteredEmer_Vehicle() Result: nat
ext rd vehicle: map Id to Vehicle
pre true
post Result = card {ev | ev in set rng vehicle & ev.Veh_Type =<EMERGENCYVEHICLE>};
```

---

This operation is also available to check the connected vehicles that are registered.

---

```
getNoOfRegisteredVehicle() Result: nat
ext rd vehicle: map Id to Vehicle
pre true
post Result = card {tv | tv in set rng vehicle & tv.Veh_Type =<VEHICLE>};
```

---

‘Set Sensors On RSU’ operation is used to monitor the vehicles. This operation takes a sensor as well as a location as an input where we have to set the sensor on RSU. Pre\_condition: used to check the sensor, which we take as an input, is not in the domain of the location map. Post\_condition: when the pre-condition is true, then the sensor place in the mapping by using set notation.

---

**SetSensorsOnRSU**(inputS: Sensors, locIn: Loc)

*ext* wr location: map Sensors to Loc  
 wr sensors : set of Sensor  
*pre* inputS not in set dom location  
*post* location = location munion{ inputS |-> locIn };

---

The ‘Emergency AlertMsg’ operation is used to sense the emergency conditions from VANETs, and the emergency alert message is sent to the other RSUs and vehicles.

---

**EmergencyAlertMsg**()

*ext* wr vehicle: map Id to Vehicle  
 rd Rsusensor : map Id to FogBasedRSU  
 wr rsu: map Id to FogBasedRSU  
 wr hospital : map Id to Hospital  
 wr fogcloud: map Id to FogCloud  
*pre* true  
*post* exists msg in set dom fogcloud & fogcloud (msg).Msgalert = Rsusensor and vehicle (msg).Msgalert = Rsusensor and rsu (msg).Msgalert = Rsusensor;

---

The ‘Send Traffic Data’ operation is used to send the video of the traffic as well as emergency conditions to the cloud for further use.

---

**SendTrafficData**()

*ext* rd traffic\_video: map Id to Camera  
 wr fogcloud: map Id to FogCloud  
*pre* true  
*post* exists vd in set dom fogcloud & fogcloud (vd). Video\_Traffic = traffic\_video;

---

In emergency situations, the ‘Time Alert’ operation is utilised to sense the time-alert from vehicular ad hoc networks. The RSU sends a time-alert message to the preceding RSU to redirect the emergency vehicle after detecting the time-alert from the environment where the sensors are positioned in fog based RSU.

---

**TimeAlert**()

*ext* wr vehicle: map Id to Vehicle  
 rd Rsusensor : map Id to FogBasedRSU  
 wr rsu: map Id to FogBasedRSU  
 wr fogcloud: map Id to FogCloud  
*pre* true  
*post* exists tm in set dom fogcloud & fogcloud (tm). Time\_alrt = Rsusensor and vehicle (tm). Time\_alrt = Rsusensor and rsu (tm). Time\_alrt = Rsusensor;

---

The ‘GetInfo for Rescue Victim’ operation is used to give the location where the sensor was deployed. When this procedure is used, it can return information about the accidents that are taking place so that rescuers can be informed in the event of an accident. The operation has a pre-condition that checks to see if the information for this accident has already been applied, and if it returns true, the post-condition will be executed, and the new accident’s information will be added to the cloud.

---

**GetInfoForRescueVictim**(sensorIn: Sensors, getInfo: Sensor\_info\_anyMiscellaneous)

*ext* wr get\_info\_anyMiscellaneous: map Sensors to Sensor\_info\_anyMiscellaneous  
 wr fogcloud: map Id to FogCloud  
*pre* (getInfo not in set rng get\_info\_anyMiscellaneous)and(sensorIn in set dom location)  
*post* get\_info\_anyMiscellaneous = get\_info\_anyMiscellaneous munion{ getInfo |-> sensorIn };

---

Get emergency operation is used to get detailed information about the emergency. This operation takes an emergency ID as an input and returns the desired emergency as an output.

---

```

getEmergency(inputId: Id)
ext rd emergency_list: map Id to Emergency
pre true
post emergency_list = { inputId }<: emergency_list;

```

---

The 'View Cloud By TA' data operation is used by the trusted authority who wants to view the data from the fog cloud. This operation takes an identifier as input and shows the information about emergency vehicles, traffic video, location, and sensors detail which deploy in RSUs.

---

```

ViewCloudDataByTA(taId : Id)
ext rd fogcloud: map Id to FogCloud
wr trusted_authority: map Id to TrustedAuthority
pre exists x in set dom fogcloud & fogcloud(x). TA_id = taId
post exists x in set dom trusted_authority & trusted_authority (x). View_vehicle = fogcloud (x). vehicleId
and trusted_authority (x). View_traffic_video = fogcloud (x).T_Video and trusted_authority (x).
View_location = fogcloud (x).currentlocation and trusted_authority (x). View_Rsusensor = fogcloud (x) .
R_sensors and trusted_authority (x). View_hospital = fogcloud (x) .hospital;

```

---

The complete description of the 'Broadcast Emergency Msg' is defined in the methodology section, in which we have described the complete procedure of the emergency services module.

---

```

BroadcastEmergencyMsg(msg : Id) Registered: bool
ext wr message : map Id to Message
wr vehicle: map Id to Vehicle
wr rsu: map Id to FogBasedRSU
rd trusted_authority: map Id to TrustedAuthority
rd Rsusensor : map Id to FogBasedRSU
rd hospital : map Id to Hospital
rd fogcloud: map Id to FogCloud
pre exists ms in set dom message & message (ms). Msg_id = msg
post if Registered =true and exists m in set rng message & m. Msg_type = <EMERGENCY> then
exists r in set rng rsu & r.broadcastMsg= <SENT_MSG_RSU> and
exists r in set rng rsu & r.broadcastMsg= <SENT_MSG_HOSPITAL> and
exists r in set rng rsu & r.broadcastMsg= <SENT_MSG_VEHICLE> and
exists r in set rng rsu & r.broadcastMsg= <SENT_MSG_EMERGENCYVEHICLE>
else Registered = false and if Registered =true and
exists msg in set dom fogcloud & fogcloud (msg).Msgalert = Rsusensor then hospital (msg).Msgalert =
Rsusensor and vehicle (msg).Msgalert = Rsusensor and rsu (msg).Msgalert = Rsusensor else Registered =
false;

```

---

The analysis of the given specifications is described in the next section of model analysis by utilizing the VDM-SL toolbox.

## 5. Model Analysis by VDM-SL Toolbox

The formal specification investigated by existing strategies available in VDM-SL toolbox to prove the correctness of the system. The VDM-SL toolbox was used to analyze formal specifications. As illustrated in the output tables, our formal model includes a standard template. Types, specifically composite types, are specified first. The state is defined second. The third category is declared functions. Functions are used to conduct operations on state and variables. The toolbox includes syntax and type checking, a pretty

printer, C++ code generation, an interpreter, an integrity inspector, dynamic verification, and a range of other validations and analysis tools. The syntax and type checkers in the VDM-SL toolbox assess the static and dynamic models that have been produced. The V2V communication utilizing unicast protocol, and emergency services modules formally described by using VDM-SL toolbox. The detail description of the model defined in given Tables 2 and 3.

**Table 2.** Results of Unicast Module.

Composite Objects	Operations and Functions	Syntax Check	Type Check	Pretty Print
MobileVehicle	VehicleDirection	✓	✓	✓
	NeighborSrch	✓	✓	✓
	NetRange	✓	✓	✓
KnownDestination	ZoneNetCreation	✓	✓	✓
	ZoneDestinationSrch	✓	✓	✓
	ChoosePeriVehZone	✓	✓	✓
VANET	ChooseHeadVehicle	✓	✓	✓
	Save Path	✓	✓	✓
Time	Acknowledged	✓	✓	✓
	KnownDestinationV2V	✓	✓	✓

**Table 3.** Results of Emergency Service Module.

Composite Objects	Operations and Functions	Syntax Check	Type Check	Pretty Print
Emergency	HospitalRegistration	✓	✓	✓
Hospital	RemovedRegisteredHospital	✓	✓	✓
Sensors	VehicleRegistration	✓	✓	✓
FogBasedRSU	RemovedRegisteredVehicle	✓	✓	✓
Message	isHospitalRegistered	✓	✓	✓
Time	isVehicle Registered	✓	✓	✓
Vehicle	getNoOfRegisteredEmerVehicle	✓	✓	✓
FogCloud	getNoOfRegisteredVehicle	✓	✓	✓
TrustedAuthority	SetSensorsOnRSU	✓	✓	✓
	EmergencyAlertMsg	✓	✓	✓
	SendTrafficData	✓	✓	✓
	TimeAlert	✓	✓	✓
	GetInfoForRescueVictim	✓	✓	✓
	NewEmergencyDeduct	✓	✓	✓
	getEmergency	✓	✓	✓
	ViewCloudDataByTA	✓	✓	✓
	BroadcastEmergencyMsg	✓	✓	✓

## 6. Conclusions

The internet of things (IoT) is an application of smart nodes in a network that detects, interprets, processes as well as reacts to data within the time needed. In vehicular ad hoc networks, the suggested approach takes into consideration human and automobile activity. It also ensures a constant flow of traffic, which not only decreases fuel use but also improves living standards. The bulk of current research on algorithm performance analysis depends on testing and simulation-based techniques, which may not guarantee correctness. This study's formal approaches are more effective in confirming the model's correctness. The VDM-SL is used to improve the specification's accuracy and to do a thorough analytical assessment [48]. VSM-SL also has an important interaction with graph theory and UML diagrams. This is due to the fact that graph-based and UML models were simple to convert into formal models. The unicast procedure also described in which



vehicle-to-vehicle communication was discussed formally. We have utilized the formal techniques VDM-SL approach to ensure the system's correctness. It has been observed that formal techniques are a more dependable and tangible approach to system development than other approaches because formal methods allow for the investigation of problems at an early stage in the system's development. It raises and improves the system's quality and bridges the semantic gap between system design and implementation.

## 7. Future Work

VANET has received most of the attention from researchers who are trying to make it more intelligent, stable, as well as healthy because of its special characteristics. More attention will be paid in the future to the design and implementation of protocols in the network. We will employ model checking approaches to improve and extend our research work. The described problem that needs to be solved is significantly essential as it will make our research more confident, improved and efficient. Further, the mobile RSU will be used to deal with the unmanned aerial vehicles that arrive at a specific place of an accident or emergency on the road and help in liquidation.

**Author Contributions:** Conceptualizations, S.I. and N.A.Z.; Introduction, S.I., N.A.Z. and T.A.; Related work, S.I., N.A.Z., T.A. and E.H.A.; Our Contributions, S.I., N.A.Z., T.A. and E.H.A.; Methodology and System Architecture, S.I., T.A. N.A.Z. and E.H.A.; Formal Methods and Formal Specification, S.I., T.A. and E.H.A.; Model Analysis Using VDM-SL toolbox, T.A.; S.I. and E.H.A. and N.A.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to acknowledge Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest.

## References

1. Jiang, Y.; Shi, M.; Shen, X.; Lin, C. BAT: A robust signature scheme for vehicular networks using binary authentication tree. *IEEE Trans. Wirel. Commun.* **2008**, *8*, 1974–1983. [[CrossRef](#)]
2. Sumanth, G.; Prabodh, C. A Survey on Security in VANETS and Applications. *Int. Res. J. Eng. Technol.* **2016**, *3*, 431–436.
3. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [[CrossRef](#)]
4. Sotres, P.; Santana, J.R.; Sánchez, L.; Lanza, J.; Muñoz, L. Practical lessons from the deployment and management of a smart city internet-of-things infrastructure: The smartsantander testbed case. *IEEE Access* **2017**, *5*, 14309–14322. [[CrossRef](#)]
5. Hatim, S.M.; Elias, S.J.; Awang, N.; Darus, M.Y. VANETS and Internet of Things (IoT): A discussion. *Indones. J. Electr. Eng. Comput. Sci.* **2018**, *12*, 218–224. [[CrossRef](#)]
6. Kumar, S.; Singh, J. Internet of Vehicles over Vanets: Smart and Secure Communication using IoT. *Scalable Comput. Pract. Exp.* **2020**, *21*, 425–440. [[CrossRef](#)]
7. Benslimane, A. Optimized dissemination of alarm messages in vehicular ad-hoc networks (VANET). In Proceedings of the IEEE International Conference on High Speed Networks and Multimedia Communications, Toulouse, France, 30 June–2 July 2004; Springer: Berlin, Germany, 2004; pp. 655–666.
8. Doukha, Z.; Moussaoui, S. Dissemination of an emergency message in a vehicular ad hoc network. In Proceedings of the 2011 International Conference on Communications, Computing and Control Applications (CCCA), Hammamet, Tunisia, 3–5 March 2011; IEEE: Manhattan, NY, USA, 2011; pp. 1–6.
9. Ullah, A.; Yaqoob, S.; Imran, M.; Ning, H. Emergency message dissemination schemes based on congestion avoidance in VANET and vehicular FoG computing. *IEEE Access* **2018**, *7*, 1570–1585. [[CrossRef](#)]
10. Wing, J.M. A specifier's introduction to formal methods. *Computer* **1990**, *23*, 8–22. [[CrossRef](#)]
11. Presti, S.L.; Butler, M.; Leuschel, M.; Booth, C. A trust analysis methodology for pervasive computing systems. In *Trusting Agents for Trusting Electronic Societies*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 129–143.

12. Nguyen, V.; Khanh, T.T.; Oo, T.Z.; Tran, N.H.; Huh, E.-N.; Hong, C.S. A cooperative and reliable RSU-assisted IEEE 802.11 p-based multi-channel MAC protocol for VANETs. *IEEE Access* **2019**, *7*, 107576–107590. [[CrossRef](#)]
13. Bello-Salau, H.; Onumanyi, A.J.; Abu-Mahfouz, A.M.; Adejo, A.O.; Mu’azu, M.B. New discrete cuckoo search optimization algorithms for effective route discovery in IoT-based vehicular ad-hoc networks. *IEEE Access* **2020**, *8*, 145469–145488. [[CrossRef](#)]
14. Kaur, M.; Malhotra, J.; Kaur, P.D. A VANET-IoT based Accident Detection and Management System for the Emergency Rescue Services in a Smart City. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; IEEE: Manhattan, NY, USA, 2020; pp. 964–968.
15. Zaheer, T.; Malik, A.W.; Rahman, A.U.; Zahir, A.; Fraz, M.M. A vehicular network-based intelligent transport system for smart cities. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719888845. [[CrossRef](#)]
16. Ramakrishnan, B.; Selvi, M.; Nishanth, R.B.; Joe, M.M. An emergency message broadcasting technique using transmission power based clustering algorithm for vehicular ad hoc network. *Wirel. Pers. Commun.* **2017**, *94*, 3197–3216. [[CrossRef](#)]
17. Nie, L.; Wang, H.; Gong, S.; Ning, Z.; Obaidat, M.S.; Hsiao, K.-F. Anomaly Detection Based on Spatio-Temporal and Sparse Features of Network Traffic in VANETs. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; IEEE: Manhattan, NY, USA, 2019; pp. 1–6.
18. Zhong, T.; Xu, B.; Szczurek, P.; Wolfson, O. Trafficinfo: An algorithm for vanet dissemination of real-time traffic information. In Proceedings of the 5th World congress on Intelligent Transport Systems, London, UK, 2–4 July 2008; Citeseer.
19. Wu, C.; Yoshinaga, T.; Ji, Y.; Zhang, Y. Computational intelligence inspired data delivery for vehicle-to-roadside communications. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12038–12048. [[CrossRef](#)]
20. Mohit, P.; Amin, R.; Biswas, G. Design of authentication protocol for wireless sensor network-based smart vehicular system. *Veh. Commun.* **2017**, *9*, 64–71. [[CrossRef](#)]
21. Manvi, S.S.; Tangade, S. A survey on authentication schemes in VANETs for secured communication. *Veh. Commun.* **2017**, *9*, 19–30. [[CrossRef](#)]
22. Kang, J.; Lin, D.; Jiang, W.; Bertino, E. Highly efficient randomized authentication in VANETs. *Pervasive Mob. Comput.* **2018**, *44*, 31–44. [[CrossRef](#)]
23. Alfidhli, S.A.; Lu, S.; Chen, K.; Sebai, M. Mfsvp: A multi-factor secured and lightweight privacy-preserving authentication scheme for vanets. *IEEE Access* **2020**, *8*, 142858–142874. [[CrossRef](#)]
24. Nellore, K.; Hancke, G.P. Traffic management for emergency vehicle priority based on visual sensing. *Sensors* **2016**, *16*, 1892. [[CrossRef](#)]
25. Khaliq, K.A.; Qayyum, A.; Pannek, J. Prototype of automatic accident detection and management in vehicular environment using VANET and IoT. In Proceedings of the 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Malabe, Sri Lanka, 6–8 December 2017; IEEE: Manhattan, NY, USA, 2017; pp. 1–7.
26. Alazzawi, M.A.; Lu, H.; Yassin, A.A.; Chen, K. Efficient Conditional Anonymity with Message Integrity and Authentication in a Vehicular Ad-Hoc Network. *IEEE Access* **2019**, *7*, 71424–71435. [[CrossRef](#)]
27. Alsamhi, S.H.; Almalki, F.A.; Al-Dois, H.; Shvetsov, A.V.; Ansari, M.S.; Hawbani, A.; Gupta, S.K.; Lee, B. Multi-Drone Edge Intelligence and SAR Smart Wearable Devices for Emergency Communication. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6710074. [[CrossRef](#)]
28. Ramakrishnan, B.; Nishanth, R.B.; Joe, M.M.; Selvi, M. Cluster based emergency message broadcasting technique for vehicular ad hoc network. *Wirel. Netw.* **2017**, *23*, 233–248. [[CrossRef](#)]
29. Gokulakrishnan, P.; Ganeshkumar, P. Road accident prevention with instant emergency warning message dissemination in vehicular ad-hoc network. *PLoS ONE* **2015**, *10*, e0143383.
30. Truong, N.B.; Lee, G.M.; Ghamri-Doudane, Y. Software defined networking-based vehicular adhoc network with fog computing. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; IEEE: Manhattan, NY, USA, 2015; pp. 1202–1207.
31. Jayaraj, V. Emergency vehicle signalling using VANETS. In Proceedings of the 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, USA, 24–26 August 2015; IEEE: Manhattan, NY, USA, 2015; pp. 734–739.
32. Senouci, O.; Aliouat, Z.; Harous, S. MCA-V2I: A Multi-hop Clustering Approach over Vehicle-to-Internet communication for improving VANETs performances. *Future Gener. Comput. Syst.* **2019**, *96*, 309–323. [[CrossRef](#)]
33. Ghazi, M.U.; Khattak, M.A.K.; Shabir, B.; Malik, A.W.; Ramzan, M.S. Emergency message dissemination in vehicular networks: A review. *IEEE Access* **2020**, *8*, 38606–38621. [[CrossRef](#)]
34. Guerrero-Ibanez, J.A.; Zeadally, S.; Contreras-Castillo, J. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wirel. Commun.* **2015**, *22*, 122–128. [[CrossRef](#)]
35. Iqbal, Z.; Saeed, T.; Zafar, N.A. Effective formal unicast routing for VANETs. In Proceedings of the 2017 Fifth International Conference on Aerospace Science & Engineering (ICASE), Islamabad, Pakistan, 14–16 November 2017; IEEE: Manhattan, NY, USA, 2017; pp. 1–6.
36. Ding, Y.; Wang, C.; Xiao, L. A static-node assisted adaptive routing protocol in vehicular networks. In Proceedings of the Fourth ACM international workshop on Vehicular Ad Hoc Networks, Online, 10 September 2007; pp. 59–68.

37. Ducourthial, B.; Khaled, Y.; Shawky, M. Conditional transmissions: Performance study of a new communication strategy in VANET. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3348–3357. [[CrossRef](#)]
38. Afzaal, H.; Saeed, T.; Iqbal, Z.; Zafar, N.A. VDM-SL-Based Model of Border Protection System using WSANs. In Proceedings of the International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, 28–29 October 2016; pp. 1–6.
39. Zafar, N.A.; Khan, S.A.; Araki, K. Towards the safety properties of moving block railway interlocking system. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 5677–5690.
40. Tahir, H.M.; Nadeem, M.; Zafar, N.A. Specifying electronic health system with Vienna development method specification language. In Proceedings of the 2015 National Software Engineering Conference (NSEC), Rawalpindi, Pakistan, 17 December 2015; IEEE: Manhattan, NY, USA, 2015; pp. 61–66.
41. Latif, S.; Afzaal, H.; Zafar, N.A. Intelligent traffic monitoring and guidance system for smart city. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; IEEE: Manhattan, NY, USA, 2018; pp. 1–6.
42. Khan, S.A.; Zafar, N.A.; Ahmad, F. Extending promotion to operate controller based on trains operation. *Int. J. Phys. Sci.* **2011**, *6*, 7262–7270.
43. Afzaal, H.; Zafar, N.A. Robot-based forest fire detection and extinguishing model. In Proceedings of the 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, Pakistan, 1–2 November 2016; IEEE: Manhattan, NY, USA, 2016.
44. Saeed, T.; Iqbal, Z.; Afzaal, H.; Zafar, N.A. Formal modeling of traffic based flooding procedure of AODV for Mobile Ad hoc Networks. In Proceedings of the 2016 International conference on emerging technologies (ICET), Islamabad, Pakistan, 18–19 October 2016; IEEE: Manhattan, NY, USA, 2016; pp. 1–6.
45. Jin, D.; Shi, F.; Song, J. A traffic flow theory based density adopted emergency message dissemination scheme for vehicular ad hoc networks. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), Siem Reap, Cambodia, 12–14 January 2015; IEEE: Manhattan, NY, USA, 2015; pp. 57–62.
46. Feroz, B.; Mehmood, A.; Maryam, H.; Zeadally, S.; Maple, C.; Shah, M.A. Vehicle-Life Interaction in Fog-Enabled Smart Connected and Autonomous Vehicles. *IEEE Access* **2021**, *9*, 7402–7420. [[CrossRef](#)]
47. Tehseen, A.; Zafar, N.A.; Ali, T.; Jameel, F.; Alkhamash, E.H. Formal Modeling of IoT and Drone-Based Forest Fire Detection and Counteraction System. *Electronics* **2022**, *11*, 128. [[CrossRef](#)]
48. Zafar, N.A. Formal specification and analysis of take-off procedure using VDM-SL. *Complex Adapt. Syst. Model.* **2016**, *4*, 1–26. [[CrossRef](#)]