Article

Impact of Direction Parameter in Performance of Modified AODV in VANET

Afsana Ahamed [1] and **Hamid Vakilzadian** [2,*]

1 Department of Electrical Engineering, Arkansas Tech University, Russellville, AR 72801, USA;
aahamed@atu.edu

2 Department of Electrical and Computer Engineering, University of Nebraska-Lincoln,
Lincoln, NE 68588, USA

* Correspondence: hvakilzadian@unl.edu

Abstract: A vehicular ad hoc network (VANET) is a technology in which moving cars are used as routers (nodes) to establish a reliable mobile communication network among the vehicles. Some of the drawbacks of the routing protocol, Ad hoc On-Demand Distance Vector (AODV), associated with VANETs are the end-to-end delay and packet loss. We modified the AODV routing protocols to reduce the number of route request (RREQ) and route reply (RREP) messages by adding direction parameters and two-step filtering. The two-step filtering process reduces the number of RREQ and RREP packets, reduces the packet overhead, and helps to select the stable route. In this study, we show the impact of the direction parameter in reducing the end-to-end delay and the packet loss in AODV. The simulation results show a 1.4% reduction in packet loss, an 11% reduction in the end-to-end delay, and an increase in throughput.

Keywords: AODV; end-to-end delay; packet loss ratio; throughput; VANET

---

1. Introduction

Intelligent traffic systems (ITS) and vehicular ad hoc networks (VANETs) are established to permit communication among vehicles to decrease traffic congestion and increase safety. A VANET uses moving cars as wireless routers (nodes) to establish a mobile network for communication [1]. The network is created by applying the principles of mobile ad hoc networks (MANET) to build a wireless network for exchanging data spontaneously. In this technology, only the vehicles equipped with wireless transceivers can exchange data with neighboring vehicles to transfer data packets to destinations that are not within direct communication range.

Compared to a MANET, a VANET has higher, structured mobility and a broad coverage area. It requires little or no power and has no service fee. However, it needs continuous data exchange to update the road structure, such as the number of lanes, the number of cars on the road, the number of roadside units (RSUs), etc. In addition, a VANET requires fast, reliable environmental data for proper and safe vehicle navigation and speed control. Because of the continuously changing number of nodes and their mobility, the throughput is low in a VANET; and packet loss is high due to connection failure. The topology created in a VANET is dynamic and not uniformly distributed. Therefore, maintaining the quality of service (QoS) is crucial; and routing of the packets is a major challenge, especially when bandwidth is limited [2].

Among the routing protocols suggested for VANETs, the Ad hoc On-Demand Vector (AODV) [3] offers rapid adaptation to changes in dynamic link, has low overhead for processing and memory usage, and provides low network utilization to determine unicast routes to a destination within the network.

The AODV is an on-demand or reactive routing protocol for a wireless ad hoc network. It initiates a route discovery when it needs to transmit data packets to a destination node, and it does not have any path towards the destination node prior to transmitting the data packets. This network consists of three procedures: (1) route discovery process, (2) route message generation, and (3) route maintenance. Since the route is only created when needed, it requires less overhead as compared to proactive routing protocols. Therefore, one of the main advantages of an AODV is the low overhead required for the data packets. In an AODV, the routing information is not updated after a specific period, which is also bandwidth efficient. This approach decreases the effects of inactive routes along with the need for route maintenance for unused routes. To improve the performance of an AODV in a VANET, Abedi et al. [4] proposed the use of direction information for each node as a parameter for selecting the next hop during a route discovery phase. However, their study did not include end-to-end (E2E) delay, throughput (TH), and packet delivery ratio (PDR). The broadcasting effect to mitigate flooding problems was also not addressed.

Wang et al. [5] used fuzzy logic and fuzzy control to make a routing decision. They developed fuzzy control-based AODV (FCAR) routing protocols. By comparing AODV and FCAR, they concluded that FCAR outperforms AODV in E2E delay and packet drop percentage. One of the main caveats of this study is that the packet drop rate of FCAR increases significantly when the network size is more than 70 nodes or the speed of the nodes (cars) is more than 10 m per second. Therefore, further study is required to reduce the packet drop rate for performance improvement.

Ding et al. [6] suggested improved AODV routing protocols to enhance stability and decrease the control overhead of the route. In this study, they used two-step optimization in the route discovery and route selection. One of the main contributions of this method is that it reduces the number of broken links. However, this method does not provide satisfactory performance in terms of packet delivery ratio. In addition, it does not consider other performance matrices, such as throughput and E2E delay, when considering the performance of the proposed method.

Sun et al. [7] proposed a global positioning system (GPS)-based AODV (GBAODV) routing protocol to establish a route. They constrained the flooding of AODV routing packets using GPS devices to improve routing performance. They found that their GBAODV method reduces the network load more than the AODV method. As a result, the number of broken links and the packet loss ratio are reduced. The average E2E delay is also shorter when using GBAODV than AODV. However, when they considered different highway scenarios, the performance of the GBAODV method for packet loss was not satisfactory (more than 10%). Their study was also based on a small number of nodes (only eight nodes).

Yu et al. [8] incorporated vehicles' movement information into the route discovery process based on an AODV for VANET applications to improve the reliability of the routing protocols (more stable route). They considered the total weight of the route (based on the position metrics) (TWR) and the estimated expiration time in their proposed protocol to achieve more stable routing. They found that their proposed protocol reduces the routing load even more and ensures more stable connections. Nevertheless, their proposed protocol does not show significant improvement in the percentage of packet drops.

He et al. [9] proposed a new vehicular reliability model that uses vehicle movement information and channel state information to improve the reliability of routing in VANETs. They extended the AODV routing protocol by proposing a reliable ad hoc, on-demand distance vector routing protocol AODV-L. They compared the traditional AODV routing with AODV-L using Objective Modular Network NESTbed in C++ (OMNeT++). They found that the AODV-L routing protocol outperformed the AODV in terms of PDR and E2E delay. Nevertheless, this study only focused on rural highways.

Feyzi et al. [10] suggested fuzzy logic to improve the efficiency of routing protocols in AODV. They used direction, speed, and distance of vehicles to the destination as inputs to a fuzzy logic controller. Their proposed protocol outperformed the original AODV for average E2E and both AODV

and FCAR for the packet delivery rate. Nevertheless, the proposed protocol underperformed FCAR regarding the average E2E delay.

Wang et al. [11] proposed an efficient message routing framework to optimize the message delivery throughput from vehicles to RSUs. They developed a mathematical model to examine the asymptotic throughput scaling of VANETs. However, the throughput of their method was poor (0.16 packets/sec). To improve the QoS of the vehicular network under different road scenarios (urban, rural, and highway), optimization of performance matrices (i.e., throughput, the percentage of packet loss, E2E delay, and the network coverage) is essential.

To increase reliability and resource usage efficiency based on medium access control (MAC) protocol, Bazzi et al. [12,13], suggested applying orthogonal frequency division multiple access (OFDMA) for alert message flooding in VANETs. They compared their protocols with other benchmark MAC protocols. In [14], Karabulut et al. proposed an OFDMA-based efficient cooperative MAC (OEC-MAC) protocol for VANETS. The OEC-MAC protocol ensured an increase in throughput with a delay of 100 ms for safety messages. The network connection reliability was raised by reducing the packet dropped rate.

In our study, we modified the AODV routing protocols to include the information on speed, direction, and position of vehicles. The routing tables now include the additional direction information of the nearby vehicles. In [4], Abedi et al. used the direction parameter to select the next-hop only. In our method, the carry and forward methods are applied so that the nearest vehicle to the destination carries the information as long as the source can send the packets to those vehicles going in the same direction, thereby making the route more stable. Filtering is done in two stages to remove the unwanted nodes moving in the opposite direction.

In the above-mentioned studies, the packets were sent to the closest vehicles in all directions; but in our approach, we used a priority-based routing where the packets were routed to the nodes that were closest and were moving in the same direction. This new method uses two filtering steps and, therefore, increases the throughput to more than 10%, compared to other state-of-the-art techniques, reduced the packet loss to 2%, and improved the E2E delays as compared to other state-of-the-art methods.

2. AODV and Modified AODV Routing Algorithms

The AODV [15] routing protocol is reactive, where routes are determined only when needed. Ad hoc On-Demand Vector is bandwidth efficient because it works only on demand. Figure 1 presents the AODV routing protocol message exchanges.
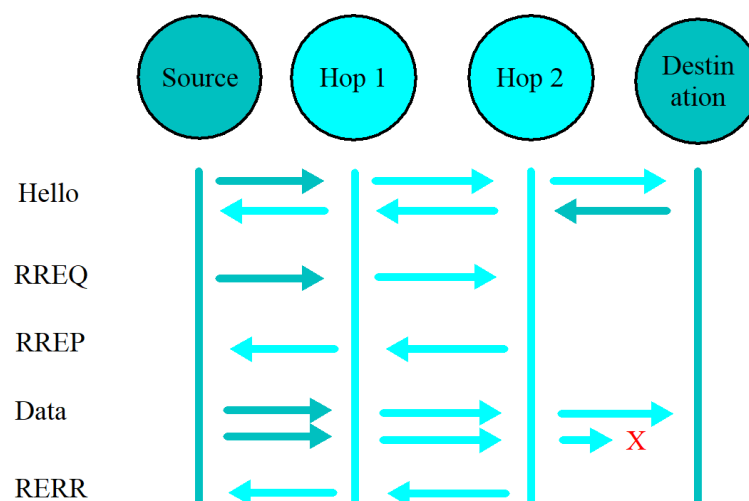

Figure 1. AODV messaging.

In this protocol, each mobile node identifies other neighborhood nodes by flooding or by receiving a local broadcast known as a Hello message. At this time, to provide an immediate response to the requester for establishing new routes, the routing tables of the neighboring nodes are modified with response time of local movements. The main objectives of this routing protocol are [15]:

1. To broadcast discovery packets (Hello) and receive acknowledgment when necessary.
2. To distinguish between general topology maintenance (route management) and local connectivity management (detection of neighborhood).
3. To inform the neighboring mobile nodes about the changes in local connection.

Hello messages [15] are utilized to detect and to observe the nearby neighbors. When Hello messages are used, every active node in the network periodically transmits a Hello message to its neighbors. These messages permit the detection of a connection or link break. The protocol uses different types of messages, shown below, to discover and maintain the links (broken links or selecting a different path). They are:

1. Route Request (RREQ)
2. Route Reply (RREP)
3. Route Reply Acknowledgement (RREP-ACK)
4. Route Error (RERR)

A source broadcasts [15] an RREQ to transfer data to an unknown destination. Therefore, a route from a node to the source is created at the intermediate node after the reception of an RREQ. If the receiving node does not receive the RREQ, the source rebroadcasts it. When the destination node is the receiving node or has a recent route from the source to the destination, it creates an RREP. As the RREP circulates, intermediate nodes generate routes to the destination. If the source receives the RREP, it registers the route to the destination in its routing table. Then it starts directing data. The source chooses the path with the shortest hop count after receiving multiple path information.

When the data is transferred from the source to the destination, nodes update their timers, which is related to the time the route is used between the source and destination to maintain the routing table. When a route is inactive (for about 2000 ms), the node invalidates the route and removes it from its routing table.

The RREP-ACK message [15] is generally sent in response to an RREP message to complete the route discovery cycle. If a link failure is detected while transferring the data, the node sends an RERR message to the source in a hop-by-hop approach. As the RERR is headed for the source, the intermediate nodes invalidate routes to inaccessible destinations. The source also invalidates the route after receiving the RERR and initiates the route discovery again.

An AODV uses a destination sequence number (DSN) [15] to avoid counting to infinity to be loop free (no duplication of packet sending). The destination includes the destination sequence number and any route information when sending messages to requesting nodes. A node broadcasts an RREQ to all network nodes to discover a route as long as it finds the destination or another node with the most recent route to the destination. The requesting node then selects the best route based on the DSN (an AODV uses a DSN to determine the most recent path to a destination). The entries in the routing table are related to sequence numbers generated by the destination. A DSN acts as a route timestamp, confirming a fresh route. When an intermediate node receives an RREQ packet, it compares its DSN with that in the RREQ packet. If the stored DSN is greater than the current DSN in the RREQ packet, the existing route is considered to be up to date. At that point, an RREP is sent back to the source, and the route (discovered) is made available.

An AODV has three main phases of operation [15]:

1. *Route discovery* is initiated when a source node needs to communicate with another neighbor node. Each node keeps two counters: broadcast_id and node_sequence_number and (so that

individual nodes can be recognized and the duplication of packets can be avoided) an RREQ. When the source node issues an RREQ, the broadcast_id is incremented. Each neighbor satisfies the RREQ by either sending back an RREP or rebroadcasting the RREQ after increasing the hop count in its neighborhood.

2. *Route maintenance* does not have any impact on the route to the destination. It reinitiates the route discovery process when the source node moves. When the destination nodes or intermediate nodes move, an RREP is sent to the affected nodes. Periodic Hello messages or link-layer acknowledgments (LLACKS) (far less latency) can be used to determine node movements. If a node is not reachable, the special RREP is sent back toward the source, containing a new sequence number and hop count.

3. *Route message generation* use a local connectivity to maintain a list of active nodes within its neighbors in one of two ways. First, if a node obtains a broadcast message from a nearby node, it updates its routing table, containing connectivity information. Second, when a neighbor does not send any packets within the Hello interval, it broadcasts a Hello message that includes its identity and its DSN.

An AODV has a routing table management system where information about neighbor nodes is updated periodically. Information about short-lived routes is also stored in the routing table, such as the routes created to store reverse paths towards nodes originating RREQs temporarily. The route entry table in the AODV [15] includes the following information on its neighbor:

- Destination sequence number
- Destination IP address
- Valid destination flag
- Next hop
- Hop count
- State and routing flags (e.g., repairable, being repaired, valid, and invalid)
- List of precursors
- Network interface
- Lifetime (i.e., deletion or expiration time of the route)

In AODV, there is no special security to prevent attacks at high latency because the route discovery is reactive and requires more time. This results in high packet loss. Since it does not allow the handling of unidirectional links, a single route request packet may result in multiple route reply packets, leading to huge routing overhead. Periodic beaconing (to determine the active/alive state of a node) leads to unnecessary bandwidth consumption, which leads to low throughput. The high mobility of the vehicles may also cause more link breakage, which increases the delay in transferring the data packets.

To maintain QoS and to overcome a flooding problem, low overhead is required when replying to a single route request. Moreover, the periodic beaconing time can be optimized depending on the size and number of nodes (cars) of the network. The route discovery phase can be modified to work more effectively with less time delay. The AODV routing protocols can be modified to give priority to a node to handle unidirectional links. Due to the highly dynamic nature of the network, the routing table can be modified and filtered to give priority to nodes based on the location information of each node.

Modified AODV

In this study, we proposed an improved AODV routing protocol for VANETs by introducing two optimization steps: (1) a route discovery phase and (2) a route selection phase. The current location, speed, and direction information about vehicles is included for performance optimization while keeping the E2E delays to a minimum. We assumed a Manhattan model in our research experiment environment. In this model, a vehicle can go straight with 50% probability and left or right, with a 25% probability each. We also assumed a highway model where the vehicles can move in the opposite direction. The AODV packets carry more information on each vehicle to keep the other vehicles more

up to date about each node on the road. We assumed automated cars with automated cruise control and wireless sensors will be used to prevent accidents. Therefore, the RREP and RREQ packets now have three pieces of information: velocity, direction, and position of each source node. The routing table in the vehicles is updated with this information so that all of the nearby vehicles can be known to each other.

This method will certainly not increase the size of a packet. However, the direction information will provide more filtering options (to make the route more stable) for the vehicles. In our model, we have assumed two different situations:

1. *Nonaccident mode:* In this case, each vehicle will update its routing table for nearby cars; but it will transmit the information to those cars which are moving in the same direction as well as all nearby RSUs within 92 m.
2. *Accident mode:* In this case, each vehicle will update the information in its routing table so that if there is an accident, the information will be sent in all directions and to nearby RSUs so that all of the vehicles can update their routing tables accordingly. Here the filtering process will be divided in two different ways. In the first half, the information will be sent to all of the vehicles within 92 m. If an accident is on the opposite side of the road, the vehicle will keep moving and updating the information for nearby vehicles. However, if it is on the same side of the road, the vehicle will stop (if it is immediate to the point of the accident and there is no free lane to move into), or it will reduce its speed and change into an available lane for safety and broadcast information.

The RREQ packet format is changed in a modified AODV so that the reserved eight bits can be used for assigning the direction in which the source is moving in terms of the degree of rotation with reference to the equator. Therefore, each node of the network has the direction information along with the location of the destination. In the case of an accident, the information can be filtered and prioritized. This information will be given to the cars moving in the same direction and close to the accident. Therefore, the accident information will be passed to the nearby nodes with more ease.

The RREQ message format is presented in Figure 2, and the fields include [15]:

Type	1 for RREQ, 2 for RREP
J	A flag which is reserved for joining
R	A flag which is reserved for repair
G	A gratuitous RREP flag when a gratuitous RREP has to be unicast to the node specific in destination IP address field
D	A destination flag when the destination is responding to the RREQ by itself
U	A flag for unknown DSN
Reserved	Source moving direction
Hop Count	The total number of hops from the source IP address to the destination IP address during the request

0									1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	1	2
Type									J	R	G	D	U	Reserved								Hop Count									
RREQ ID																															
Destination IP Address																															
Destination Sequence Number																															
Originator IP Address																															
Originator Sequence Number																															

Figure 2. RREQ message format.

The nearest node is calculated using the Pythagoras formula. If the location of node A is (λ_1, φ_1) and the location of node B is (λ_2, φ_2), then the shortest distance between them is calculated as [16]

$$d \;=\; 2R \, \sin^{-1} \sqrt{\sin^2 \frac{(\varphi_1 - \varphi_2)}{2} + \cos\varphi_1 \cos\varphi_2 \sin^2 \frac{(\lambda_1 - \lambda_2)}{2}} \tag{1}$$

where R is the earth radius, φ is the latitude, and λ is the longitude.

If the latitudes of nodes A and B are φ_1 and φ_2, respectively, the direction between them is calculated as

$$\theta \;=\; \varphi_1 - \varphi_2 \tag{2}$$

In addition, the modified AODV routing table contains the direction information column for each nearby node that will be used to filter the nodes to pass data with little or no delay. Priority will be given to those nodes which have a minimum angle θ. Here, we assume that if the angle is less than 20 degrees, then the nodes are moving in the same direction [4].

3. Simulation Environment

We assessed the performance of the proposed routing protocols in OMNeT++ and used MATLAB® to analyze the simulation data. Recently, the combination of simulation of urban mobility (SUMO) and OMNeT++ has been used to build the network communication in a VANET. We combined the network with a traffic simulator named SUMO, which is used to reproduce the network protocols in the VANET. The mobility of each node in this simulation environment is controlled by SUMO, and the position is forwarded periodically to the network simulator.

In a VANET, the outcomes of a simulation are not entirely reliable and mostly guided by the mobility model in use. In this context, Sommer et al. [17] suggested the necessity for a robust simulation environment for VANETs. This led to the advancement of Veins (vehicles in network simulation) [18]. Veins is an open-source simulation environment that uses two simulators: (1) SUMO for conducting microscopic road traffic simulation and (2) OMNeT++ for conducting the network simulation, as shown in Figure 3.
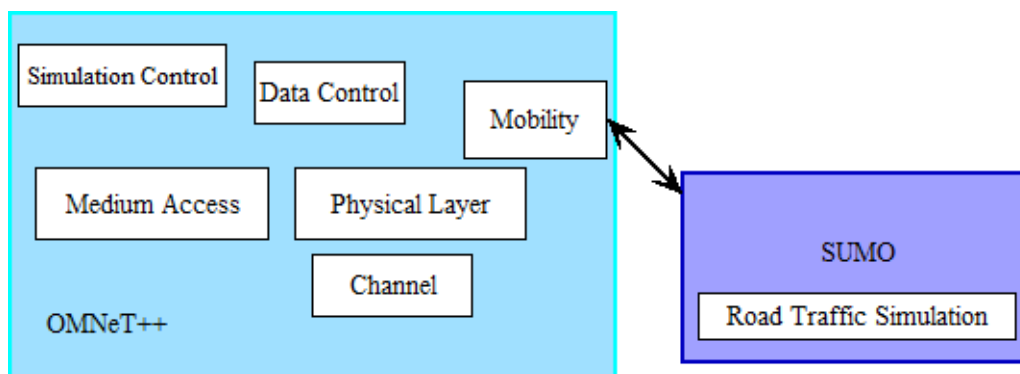
Figure 3. Objective Modular Network NESTbed in C++ (OMNeT++) with simulation of urban mobility (SUMO) [16].

The road map for VANET simulation is imported from openstreetmap.org [19] by manually choosing the required region or area and exporting to the .osm option to download a map.osm file. The resultant file is an XML file with an OpenStreetMap-defined region. We used Java Open Street Map Editor (JOSM) to edit the map.osm file manually. Apart from the road network, the imported and edited map.osm also includes other information, such as rivers, walkways, power distribution lines, bicycle routes, parks, and other features that are outside the scope of this study. A brief description of the simulation tools used in this research is given below:

3.1. SUMO

Simulation of Urban Mobility is an open-source traffic simulation software. When a vehicle or node moves through a given road network, SUMO can handle a given traffic demand and permits addressing a large set of traffic scenarios. In SUMO, each vehicle is modeled explicitly, having its own route, and moves autonomously through the system. There are also several alternatives to introduce randomness to the simulation [20].

3.2. JOSM

Java Open Street Map (OSM) Editor, a desktop application, was developed by Immanuel Scholz. It is currently maintained by Dirk Stöcker. It is an open street map for a JAVA platform that supports loading standalone GPS tracks from an OSM database to load and edit existing nodes. The imported map.osm also includes other information (e.g., rivers, walkways, power distribution lines, railway networks, and bicycle routes, etc.) apart from the road network. In this study, JOSM was used to edit the map. We delete unnecessary nodes; and, if required, we add new nodes with new routes and infrastructures to the simulation environment. Therefore, the final map.osm file contains only the road network and infrastructure that are necessary for the simulation environment [21].

3.3. OMNeT++

Objective Modular Network NESTbed in C++ is a discrete event simulator. It is used for modeling wired and wireless communication networks as well as microprocessors, multiprocessors, and other distributed or parallel computing systems. It is based on the C++ language, and it consists of different basic modules that communicate with each other by passing messages among them. These basic modules are used to create larger modules [22].

The Network Description (NED) language is used to simulate models in OMNeT++. The component modules described in this language can be assembled to create a compound (system) module. Models written in NED can be simulated by one of the network simulators, such as MiXiM [23], INET [24]/INETMANET [25], or Veins [26]. Veins is the open-source vehicular network simulator. The execution of a model by Veins is controlled by OMNeT++ while interacting simultaneously with a road traffic simulator (SUMO). Various submodules in Veins take care of setting up, running, and monitoring the simulation.

When performing intravehicular communication (IVC) evaluations, both traffic and network simulators run in parallel since they are connected via a Transmission Control Protocol (TCP) socket. This communication network protocol is known as Traffic Control Interface (TraCI) [27].

The vehicle movement in SUMO is directed by the movement of nodes in an OMNeT++ simulation environment. We have also used Plexe [28] for the movement of an automated vehicle in OMNeT++. Plexe is an extension of Veins in OMNeT++. It allows realistic and accurate simulation of the automated car. It structures real-time vehicle dynamics and several cruise control models, enabling the implementation of large-scale and mixed scenario control systems along with different networking protocols.

4. VANET Component Model

The following are some of the essential models of VANET components that were used in the simulation.

4.1. Two-Ray Interference Model

The Two-Ray Ground Reflection Model is a radio propagation model used to predict the path losses between transmitting and receiving antennas of the nodes. The received signal has two components: the line-of-sight (LOS) component and the multipath component, which is formed by a single ground-reflected wave. This model considers the impact of both the direct path and the ground reflection during information

propagation. The path loss is included to model the propagation of information in the vehicular network. The received power is

$$P_r \propto \frac{1}{d^4} \tag{3}$$

where d is the distance between the transmitting and receiving antennas.

For any nonlinear (not smooth) unobstructed stretch of road, depending on the distance, the transmission faces constructive or destructive interference for its ground reflection. The two-ray ground model only considers that path loss increase for distances over approximately 900 m. Therefore, to capture ground reflection, VANET is comprised of a Two-Ray Interference model [29].

4.2. Obstacle Shadowing Model

The signal shadowing effects are heavily impacted by radio transmission. Therefore, obtaining the real signal is important in VANET in both urban and suburban areas, where buildings and infrastructure block radio propagation. In the simulation, we include a calibrated and validated obstacle shadowing model against realistic obstacle measurement [18]. This model accurately captures the effect of blocking transmission by large buildings. In other words, while strong transmissions can be hindered by the presence of infrastructure in the line of sight, weak transmissions can be blocked by something as small as a short wall.

4.3. Adaptive Cruise Control (ACC) Model

An ACC system offers driver comfort and ease by providing cruise control in a high mobility traffic environment. Adaptive cruise control systems improve highway safety. About 94% of highway accidents occur through human error [30], while a small percentage of highway accidents are caused by equipment failure or weather conditions (such as slippery roads). Since an ACC system possibly decreases driver liability and changes driver function with an automated process, it is estimated that the implementation of an ACC system will reduce the number of crashes and hazards [28] on the road.

An ACC system with two modes of steady-state operation (i.e., speed control and vehicle spacing control) is shown in Figure 4.
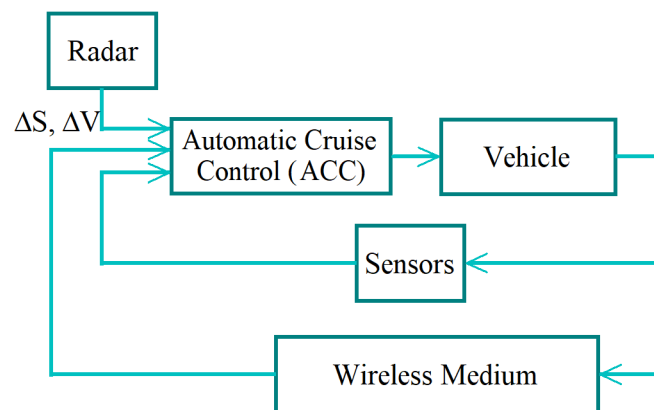

Figure 4. Adaptive cruise control (ACC) with vehicle [28].

4.4. Roadside Unit (RSU)

Roadside units are essential nodes to provide communication support between vehicles. This support may be in the form of reporting an accident, safety warnings, and traffic congestion. A model of an RSU is shown in Figure 5. Each RSU is equipped with a network interference card (NIC) and the application layer software, appl. The NIC unit also includes the AODV router.
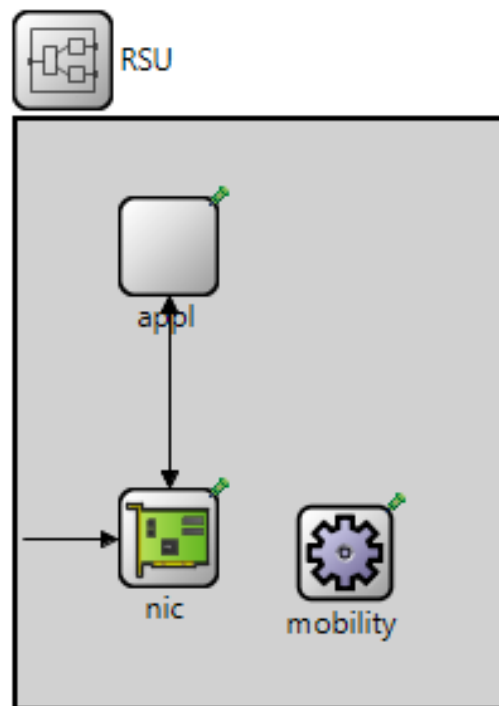
Figure 5. Roadside unit (RSU) model.

5. Results and Discussion

To compare our results with published articles, we used the map of Erlangen in SUMO to generate the traffic information. This map is used primarily in VANET simulations in most of the state-of-the-art methods. The mobility model was generated using OMNeT++, and then the simulations were run by varying the number of vehicles. The simulation was run 20 times to have a statistically significant output.

Figure 6 shows the roads and the existing infrastructure of the area. The black lines are the roads, and the red-colored areas indicate the infrastructure.

Figure 6. View of Erlangen map in SUMO.

The simulation result in OMNeT++ is shown in Figure 7, where the blue dots are the moving nodes in the same area in SUMO. A summary of the simulation parameters is provided in Table 1.

The percentage of packet loss versus the number of cars is presented in Figure 8. It can be seen from the plot that the packet loss increased with an increase in the number of cars.
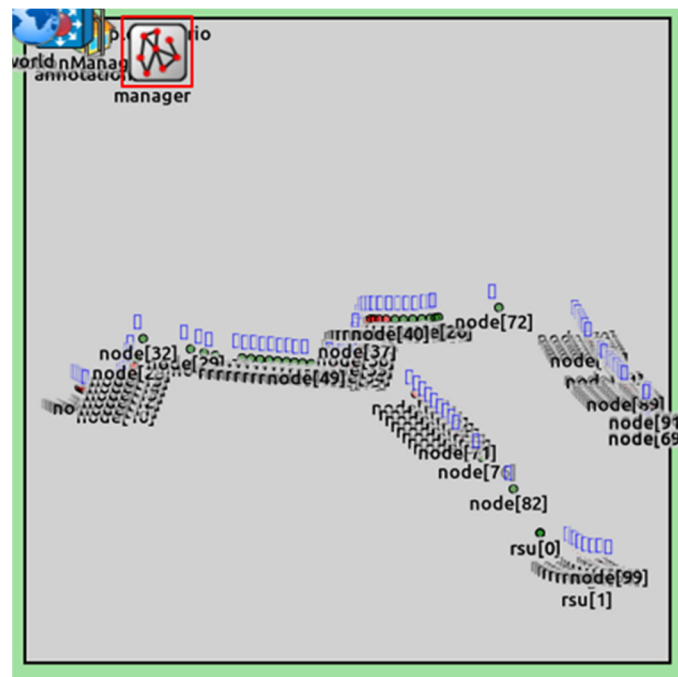
Figure 7. Simulation in OMNeT++.

Table 1. Summary of simulation parameters.

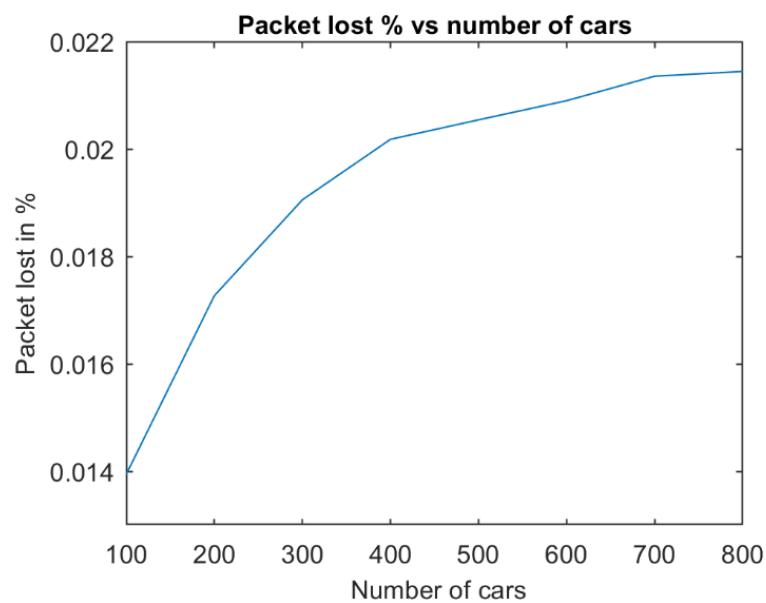| Parameters	Value
Network simulator	OMNeT++
Simulation time	1000 s
Number of nodes	100–800
Simulation coverage volume	2500 m (L) × 2500 m (W) × 50 m (H)
MAC layer	802.11 p
Propagation model	Two ray interferences
Mobility model	Random trip and Intelligent car
Number of runs	20
Velocity	20–40 mps

Figure 8. Percentage of packet loss vs. the number of cars.

The throughput vs. the number of cars is shown in Figure 9. It can be seen from this graph that the throughput increased with the number of cars due to the high network activities with more cars.
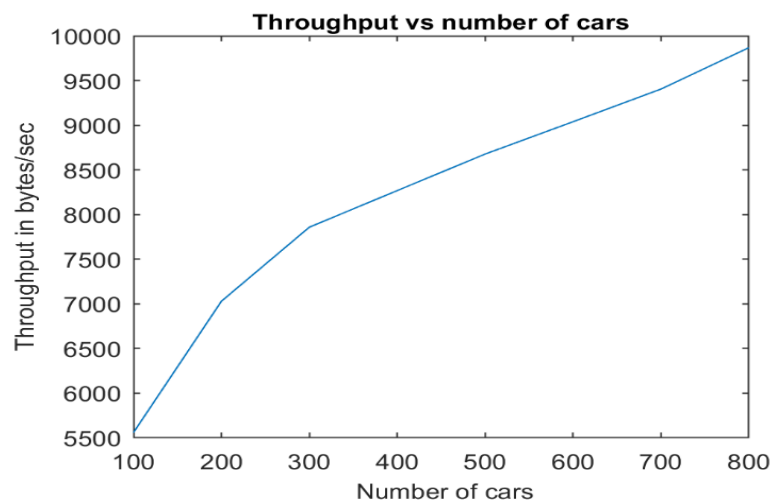

Figure 9. Throughput vs. the number of cars.

We examined the performance regarding packet loss and packet delivery ratios during the simulation; and the results are provided in Table 2 with 100 nodes and in Table 3 with 200 nodes, along with the results of other methods [4–6,9]. In both cases, the inclusion of the direction parameter outperformed the others by about 1.3% under the same environmental conditions.

Table 2. Percentage of packet loss with 100 nodes.

Methodology	Packet Loss	Packet Delivery Ratio
Our method	*1.4%*	*98.6%*
Wang et al. [5]	2.75%	97.25%
Sun et al. [7]	10%	90%

Table 3. Percentage of packet loss with 200 nodes.

Methodology	Packet Loss	Packet Delivery Ratio
Our method	*1.7%*	*98.3%*
Wang et al. [5]	3%	97%
He et al. [9]	3%	97%
Ding et al. [6]	7%	93%

The comparison of an average E2E delay is presented in Table 4. As shown, our method displayed a decrease of 11% in delay compared with the other methods.

Table 4. E2E delay performance.

Methodology	Average E2E Delay
Our method	*178 ms*
Wang et al. [5]	200 ms
Sun et al. [7]	400 ms
He et al. [9]	300 ms
Feyzi et al. [10]	225 ms
Wang et al. [11]	250 ms

The improved performance is the result of using two filtering steps in route discovery and route selection procedures. In this case, each node has the most updated routing information; therefore, it also has fewer broken links. The packet drop rate and end-to-end delay also decreased due to a lower number of broken links in a stable route. These processes also reduced the number of RREQ and RREP messages, which increased throughput.

6. Conclusions and Future Work

In this research paper, we simulated the performance analysis of VANET using OMNeT++ and SUMO by collecting the map from OpenStreetMap. We modified the AODV routing protocols to reduce the number of RREQ and RREP messages by adding direction parameters and two-step filtering. The two-step filtering process reduced the number of RREQ and RREP packets, reduced the packet overhead, and helped selection of the stable route. We also compared the performance metrics in terms of E2E delay, packet delivery ratio, and throughput with other state-of-the-art techniques.

The inclusion of direction parameters along with two-step filtering in route discovery and route selection procedures reduced the average end-to-end delays and packet loss by 11% and 1.3%, respectively, as compared to other state-of-the-art methods. We also found an improvement in throughput that was due to the combined effect of direction parameters and filtering in the same direction to find the stable route, which reduced the packet loss during communication. Therefore, the proposed method increases the QoS in VANET, providing more safety on the road.

In our future research, we will study and propose new protocols that might help to improve the performance of protocols such as dynamic source routing with a grid or geographic location service protocol for VANETs concerning road safety in real-time applications.

Author Contributions: Conceptualization, methodology, software, and preparation of original draft by A.A.; supervision, review, recommendation for changes, and editing by H.V. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

ACC	Automatic cruise control
AODV-L	AODV link expire time
AODV	Ad hoc On-demand Distance Vector
DSN	Destination sequence number
E2E	End-to-end
FCAR	Fuzzy control-based AODV routing
GBAODV	Global positioning system (GPS)-based AODV
GPS	Global positioning system
ITS	Intelligent traffic system
IVC	Intervehicle communication
JOSM	Java OpenStreetMap
LLAKS	Link-layer acknowledgments
LOS	Line-of-sight
MAC	Medium access control
MANET	Mobile ad hoc network
NED	Network description
NIC	Network interface card
OFDMA	Orthogonal frequency division multiple access
OMNeT++	Objective Modular Network NESTbed in C++
OSM	OpenStreetMap
PDR	Packet delivery ratio

PDR	Packet dropped rate
QoS	Quality of service
RSU	Roadside unit
RERR	Route error
RREP	Route reply
RREP-ACK	Route reply acknowledgment
RREQ	Route request
SUMO	Simulation of Urban Mobility
TCP	Transmission Control Protocol
TH	Throughput
TraCI	Traffic Control Interface
TWR	Total weight of the route
VANET	Vehicular ad hoc network
Veins	Vehicles in network simulation

References

1. Benslimane, A. Localization in Vehicular Ad Hoc networks. In *2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05)*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2005.

2. Ahamed, A.; Vakilzadian, H. Issues and challenges in VANET routing protocols. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 723–728.

3. Perkins, C.E.; Royer, E.M. Ad-hoc on-demand distance vector routing. In Proceedings of the WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 25–26 February 1999.

4. Abedi, O.; Fathy, M.; Taghiloo, J. Enhancing AODV routing protocol using mobility parameters in VANET. In Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, Doha, Qatar, 31 March–4 April 2008; pp. 229–235.

5. Wang, X.-B.; Yang, Y.-L.; An, J.-W. Multi-metric routing decisions in VANET. In Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, 12–14 December 2009; pp. 551–556.

6. Ding, B.; Chen, Z.; Wang, Y.; Yu, H. An improved AODV routing protocol for VANETs. In Proceedings of the 2011 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 9–11 November 2011; pp. 1–5.

7. Sun, Y.; Chen, Y.; Xu, Y. A GPS enhanced routing protocol for vehicular Ad-hoc network. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, Phuket, Thailand, 7–11 December 2011; pp. 2096–2101.

8. Yu, X.; Guo, H.; Wong, W.-C. A reliable routing protocol for VANET communications. In Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 4–8 July 2011; pp. 1748–1753.

9. He, Y.; Xu, W.; Lin, X. A stable routing protocol for highway mobility over vehicular ad-hoc networks. In Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, UK, 11–14 May 2015; pp. 1–5.

10. Feyzi, A.; Sattari-Naeini, V. Application of fuzzy logic for selecting the route in AODV routing protocol for vehicular ad hoc networks. In Proceedings of the 23rd Iranian Conference on Electrical Engineering, Tehran, Iran, 10–14 May 2015; pp. 684–687.

11. Wang, M.; Shan, H.; Luan, T.H.; Lu, N.; Zhang, R.; Shen, X.; Bai, F. Asymptotic throughput capacity analysis of VANETs exploiting mobility diversity. *IEEE Trans. Veh. Technol.* **2014**, *64*, 1. [CrossRef]

12. Bazzi, A.; Masini, B.M.; Zabini, F. On the exploitation of OFDMA properties for an efficient alert message flooding in VANETs. In Proceedings of the 2013 IEEE International Conference on Communications, Xi'an, China, 12–14 August 2013; pp. 5094–5098.

13. Bazzi, A.; Zanella, A.; Masini, B.M. An OFDMA-based MAC protocol for next-generation VANETs. *IEEE Trans. Veh. Technol.* **2014**, *64*, 4088–4100. [CrossRef]

14. Karabulut, M.A.; Shah, A.F.M.S.; Ilhan, H. OEC-MAC: A novel OFDMA based efficient cooperative MAC protocol for VANETS. *IEEE Access* **2020**, *8*, 94665–94677. [CrossRef]

15. Perkins, C.; Belding-Royer, E.; Das, S. *Ad hoc On-Demand Distance Vector (AODV) Routing*; RFC Editor: Santa Barbara, CA, USA, 2003.

16. Mahmoud, H.; Akkari, N. Shortest path calculation: A comparative study for location-based recommender system. In Proceedings of the 2016 World Symposium on Computer Applications & Research (WSCAR), Cairo, Egypt, 12–14 March 2016; pp. 1–5. [CrossRef]

17. Sommer, C.; German, R.; Dressler, F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.* **2010**, *10*, 3–15. [CrossRef]

18. Sommer, C.; Dietrich, I.; Dressler, F. Realistic simulation of network protocols in VANET scenarios. In Proceedings of the 2007 Mobile Networking for Vehicular Environments, Anchorage, AK, USA, 11 May 2007; pp. 139–143. [CrossRef]

19. OpenStreetMap Wiki Contributors, "OpenStreetMap", OpenStreetMap Wiki. 2004. Available online: http://www.openstreetmap.org (accessed on 7 August 2020).

20. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of SUMO—Simulation of urban mobility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.

21. OpenStreetMap Wiki Contributors, JOSM, OpenStreetMap Wiki. 2017. Available online: http://wiki.openstreetmap.org/w/index.php?title=JOSM&oldid=1492146 (accessed on 7 August 2020).

22. Varga, A.; Hornig, R. An overview of the omnet++ simulation environment. In Proceedings of the 1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Marseille, France, 3–7 March 2008.

23. OMNeTpp/mixim, GitHub. 2016. Available online: https://github.com/OMNeTpp/mixim (accessed on 7 August 2020).

24. INET Framework, OMNeT++. 2012. Available online: http://inet.OMNeTpp.org/ (accessed on 7 August 2020).

25. aarizaq/inetmanet-3.x, GitHub. 2017. Available online: https://github.com/aarizaq/inetmanet-3.x (accessed on 7 August 2020).

26. Sommer, C. Veins. 2006. Available online: http://veins.car2x.org (accessed on 7 August 2020).

27. Wegener, A.; Piórkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.-P. TraCI: An interface for coupling road traffic and network simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium, New York, NY, USA, 14–17 April 2008.

28. Segata, M.; Joerer, S.; Bloessl, B.; Sommer, C.; Dressler, F.; Cigno, R.L. Plexe: A platooning extension for veins. In Proceedings of the 2014 IEEE Vehicular Networking Conference (VNC), Padeborn, Germany, 3–5 December 2014; pp. 53–60.

29. Henderson, T. Two-Ray Ground Reflection Model, isi.edu. 5 November 2011. Available online: https://www.isi.edu/nsnam/ns/doc/node218.html (accessed on 7 August 2020).

30. Singh, S. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*; National Highway Traffic Safety Administration: Washington, DC, USA, 2015.