OLBR: Optimization of Location-Based Routing protocols in Vehicular Ad Hoc Networks

*Thangam S[1], M Vivek Srikar Reddy[2], Karthik Ullas[3], N Abdul Khadhar Jilani[4], K Venkatesh[5], J Jesy Janet Kumari[6]
Dept. of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India
[1]s_thangam@blr.amrita.edu, [2]bl.en.u4cse21115@bl.students.amrita.edu,
[3]bl.en.u4cse21092@bl.students.amrita.edu, [4]bl.en.u4cse21135@bl.students.amrita.edu,
[5]bl.en.u4cse21085@bl.students.amrita.edu, [6]bl.en.r4cse21008@bl.students.amrita.edu

***Abstract*— The proposed work employs ns-3, SUMO, and NetAnim to create geographical routing in vehicular ad hoc networks (VANETs). The research attempts to assess the performance of three well-known protocols AODV, DSDV, and OLSR in VANET settings. Utilizing SUMO's realistic mobility models and ns-3's simulation capabilities, the project simulates node communication, protocol behaviour, and vehicle movement. The main purpose is to know how well various routing techniques work in dynamic vehicular situations.**

***Keywords*— Geographical Routing, VANET, node, AODV, DSDV, OLSR.**

I. INTRODUCTION

A system that improves communication between vehicles and roadside equipment is called a Vehicular Adhoc Network (VANET). It benefits like vehicle-to-vehicle (V2V) and Inter-vehicle communication (IVC). It uses a distributed ad hoc technique to enhance traffic conditions and driving safety. VANET is a component of the broader Intelligent Transportation System (ITS), which try to improve the system's overall performance. Through active communication, VANETs enable cars to recognise and react to traffic jams and accidents, ensuring driver safety. And also, high dynamics, frequent network changes are the present difficulties for VANETs. Vehicular Adhoc Networks (VANETs) have gained a lot of attention due to their unique characteristics, including distributed management, multi-hop routing, and self-configuration. Traditional protocols, such as network topology-based AODV and DSR, have no way to adapt to the dynamic changes happening in VANETs. Geographic routing, that relies on geographical information instead of topology, is as a scalable and effective solution to this issue. Geographic routing, as compared to path-based techniques, makes advantage of nodes geographic locations, doing away with the requirement for complex routing tables. There are certain issues with the common greedy forwarding method, that forwards packets to nodes nearby which are closer to the target. These issues include communication voids. Various void-handling techniques have been proposed to address this issue, providing alternatives for mobile networks like VANETs. The aim for this study is to expand the data exchange between linked vehicles in order to develop applications designed especially for Intelligent Transportation Systems (ITS). To improve communication between node in VANETs our proposed work involves different geographic routing algorithms like Greedy forwarding, GPSR (Geographical Position based Routing) and in this algorithm many parameters can influence the GPSR performance such as obstacles, the link quality, the network size, and so on. As a result, the two parameters that our suggested work primarily addresses here are the vehicle speed and the neighbour node. In particular, vehicle speed plays a critical role in data routing between vehicles. This is because traveling too quickly increases the likelihood of a link break, so it is imperative to choose a vehicle with a low speed probability to guarantee packet arrival at the destination. Furthermore, a key component of our suggested work is estimating the neighbour node's chance of being near the source; as a result, this factor would permit the node to be near the source for an efficient data delivery. The technique of fuzzy logic is used to apply these two criteria in order to determine the optimal next-hop selection, which will improve performance. In the context of a vehicular Adhoc network (VANET), packets can be safely passed from one node to another via Optimised Link State Routing (OLSR) and without any data loss by using these mentioned algorithms we can safely pass the packets from one node to another in the network without any data loss in Vehicular Adhoc Network (VANET).

II. LITERATURE REVIEW

Dynamic Trust Tokens (DTT), a recent VANET security system, use hybrid cryptography for real-time dependability assessment, which may cause data packet loss. Geographic Position-Based Routing (PBR) uses cryptographic techniques to guarantee network integrity, whereas Geographic Secure Path Routing (GSPR) uses position and path authentication to secure location-based services. To counter position spoofing risks in vehicle communication, Secure Location Verification (SLV) uses radio frequency (RF) [1]. VANET protocols were developed by researchers to guarantee lossless data transport. By utilising greedy forwarding and perimeter modes, GPSR maximises package transit while removing obstacles. By highlighting junctions, GPCR increases the efficiency of urban routing. Through beacon messages at intersections, GPSRJ+ improves GPSR in urban VANETs. By using vehicle position and direction for improved routing, PDGR, a direction-aware system, increases accuracy in same-direction travel scenarios [2]. Research which was presented at the IScV conference, assessed Geographic and Topology-Based Routing Protocols in VANETs with a focus on performance and QoS analysis in vehicular ad hoc networks. Most likely, the study investigates protocol effectiveness with an emphasis on network dependability and effective data transfer [3]. Geographic Routing Protocols in VANETs are examined in a Vehicular Communications survey, which also probably discusses their usefulness, difficulties, and future developments for effective routing and communication techniques in vehicular networks. With regard to mobility and scalability in ad hoc networks, the study intends to provide insights into improving communication efficiency and routing dependability within dynamic vehicle contexts [4].

Vehicular Ad-Hoc Networks (VANETs) have recently seen the introduction of innovative protocols such as V-GEDIR, which employ Voronoi Diagrams for loop-free routing (VD-GREEDY & CH-MFR). Junction-based Geographic Distance Routing (JGEDIR) uses a strategic, greedy distance method that takes into account the positioning of vehicles at intersections to optimize latency. Various techniques have been investigated for void handling, including Beacon-Less Routing, Greedy Perimeter Stateless Routing, SPEED-vb's 2-Hop Forwarding, BOUNDHOLE, One-hop Flooding, and Partial Source Routing [5]. This research examines Void Handling Techniques for Geographic Routing in VANET networks and is published in the International Journal of Grid & Distributed Computing. The review probably looks at ways to fill in the gaps (communication gaps) in geographic routing in VANETs in order to improve routing performance and dependability by reducing problems resulting from unconnected or underserved areas [6]. Furthermore, QAGR, an adaptive geographic routing method based on Q-learning, was presented. QAGR separates routing into two parts: ground and aerial. For context-specific information, fuzzy logic is used, and for global routing, a depth-first search technique is applied. To improve adaptation to changes in VANET topology, the ground component uses a traditional greedy technique in conjunction with a fixed-sized Q-table [7]. Additionally, trust management for security in geographic routing has been studied by researchers. An ideal Road-Side Unit setup was shown, combining triangulation-based location verification with trust-oriented secure multi-hop routing. In actual simulations, this procedure demonstrated to be more resilient and dependable than baseline techniques [8].

A QoS-aware service selection architecture designed for pervasive contexts is presented in this research. In order to improve overall quality and performance, it focuses on streamlining the processes used for service selection in these kinds of situations. [9]. The comparative analysis of the AODV and EDAODV routing protocols in VANETs under congestion conditions is explored in this paper. It evaluates their performance through thorough analysis, offering important information for vehicle network congestion control solutions [10]. This study uses an optimised trust computation model in a chimp-AODV enabled WSN to study how to identify selfish nodes that are dishonest. The paper discusses improved detection techniques that are essential to preserving wireless sensor network integrity [11]. This research offers a delay-based routing protocol designed for collision avoidance on VANETs, with the goal of optimising routing algorithms. Its main objective is to increase vehicular networks' efficiency, especially in situations where collisions are a concern [12]. This study looks at a fault-tolerant peer-to-peer multicasting overlay strategy designed for emergency data transmission in VANETs. The goal of the project is to improve the dependability of data distribution in vehicle networks in emergency conditions [13].

III. METHDOLOGY

A. Geographic Routing Framework Establishment: -

The proposed framework introduces a hierarchical network architecture tailored explicitly for Vehicular Ad Hoc Networks (VANETs). This innovative structure delineates three fundamental layers—core, distribution, and access—meticulously designed to adeptly manage the flow of data within the dynamic vehicular environment. Each layer operates synergistically, considering crucial aspects such as vehicular mobility patterns, communication ranges, and routing functionalities inspired by a spectrum of sophisticated algorithms.

At the core layer, the framework orchestrates the foundational network operations, ensuring robustness and reliability in data transmission across the VANET. The distribution layer acts as an intermediary, efficiently routing and disseminating information received from the core to the access layer, optimizing data propagation while considering the varied communication ranges inherent in vehicular movement.

The access layer, intricately connected to the vehicular nodes, facilitates direct interaction with individual vehicles. It accommodates the specific communication needs dictated by the constantly evolving vehicular dynamics. The hierarchical structure, drawing inspiration from diverse routing algorithms, aims to enhance data management efficiency, minimize latency, and adapt dynamically to the volatile nature of VANETs, thereby bolstering the overall network performance and reliability.

B. VANET Model:

Vehicular Ad Hoc Networks (VANETs) represent a transformative advancement in wireless communication systems, mirroring the interconnected nature of social networking within the realm of automobiles. These networks have emerged as a pivotal innovation in the automotive industry, revolutionizing the way vehicles communicate and enhancing their reliability while in motion. By enabling seamless communication among vehicles on the move, VANETs bolster the dependability and safety of automotive systems.

At the core of VANETs lies the fundamental principle of facilitating communication between nearby vehicles and roadside infrastructure. This synergy between automobiles and roadside devices forms the bedrock of this technology, enabling a dynamic mesh network that empowers vehicles to exchange critical information in real time. This exchange encompasses a diverse range of data, including traffic conditions, road hazards, weather updates, and other pertinent information crucial for ensuring a safer and more efficient driving experience.

The significance of VANETs is underscored by their multifaceted real-world applications, catering to both private and public vehicles. In the domain of private vehicles, these networks enhance the driving experience by providing predictive traffic information, suggesting alternate routes to avoid congestion, and enabling proactive collision avoidance systems. Meanwhile, for public vehicles like buses and emergency service vehicles, VANETs play a pivotal role in optimizing routes, ensuring timely responses to emergencies, and improving overall transport efficiency.

One of the defining features of VANETs is their ability to create a dynamic and self-configuring network without the need for a centralized infrastructure. This decentralized nature allows vehicles to form ad hoc networks, communicating directly with each other within their vicinity. Such communication is facilitated by dedicated wireless protocols, such as IEEE 802.11p, specifically designed to cater to the high-speed mobility and intermittent connectivity characteristic of vehicular environments.

The integration of VANETs into the automotive landscape represents a paradigm shift, not only enhancing the individual driving experience but also contributing significantly to the collective efficiency and safety of vehicular operations. As these networks continue to evolve and integrate with emerging technologies like autonomous vehicles and smart cities, the potential for further advancements in transportation systems and road safety grows exponentially.

C. Architecture:

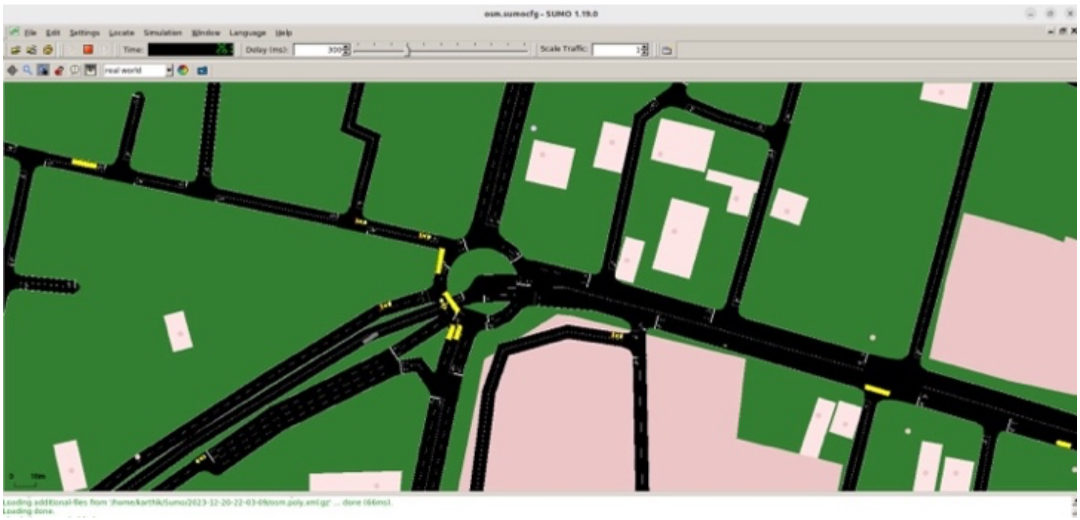


Figure 1. Working of VANET in real time scenario

Figure 1 illustrates the real-time operation of VANETs, depicted against a backdrop of a route map, showcasing the dynamic functionality within the vehicular environment. Concurrently, Figure 2 outlines the comprehensive architecture of VANETs. Here, the integral components include the data collection by Car Electronic Control Units (ECUs) and sensors, pivotal for gathering crucial information. Communication within the network is facilitated through cellular Base Transceiver Stations (BTS), enabling seamless data exchange among vehicles and infrastructure.

The system integrates various services, such as parking and traffic information, optimizing operational efficiency. Road details play a vital role in enabling vehicle adaptation to the ever-changing environment. Cooperative functions rely on WiFi and Vehicle-to-Vehicle (V2V) communication, fostering collaboration among vehicles for shared objectives. This architectural depiction underscores the intricate interplay of components and services within VANETs, emphasizing their role in enabling real-time data collection, exchange, and cooperative actions to enhance vehicular operations and safety.

Figure 2. Architecture of VANET

Vehicle-to-Vehicle (V2V) Communication:

Components Involved: OBUs of the participating vehicles.Description: V2V communication involves wireless data exchange between the On-Board Units (OBUs) of different vehicles. This communication enables vehicles to share information about their current status, such as speed, location, and direction. V2V is essential for cooperative applications like collision avoidance, traffic management, and forming ad-hoc networks among vehicles.

Vehicle-to-Infrastructure (V2I) Communication:

Components Involved: OBUs of vehicles and Road-Side Units (RSUs) connected to the infrastructure. Description: V2I communication establishes bidirectional wireless communication between vehicles and Road-Side Units (RSUs). RSUs are equipped with sensors and communication devices and are typically deployed at intersections, traffic lights, and other key points. This communication allows vehicles to receive information about traffic conditions, signal changes, and other relevant data from the infrastructure, enhancing safety and traffic efficiency.

Infrastructure-to-Infrastructure (I2I) Communication:

Components Involved: RSUs communicating with each other. Description: I2I communication involves the exchange of data between different Road-Side Units (RSUs). This communication extends the coverage of the VANET network and ensures that information can be relayed across a broader area. I2I communication is vital for maintaining a seamless and well-connected VANET infrastructure.

D. Algorithm :

Vehicle Ad Hoc Networks (VANETs) use a variety of routing techniques, including:

Various routing methodologies within Vehicular Ad Hoc Networks (VANETs) leverage different approaches to manage data transmission and optimize communication efficiency. Topology-Based routing aims to strike a balance between overhead and scalability by relying on reasonably accurate network structure information. It utilizes a predefined network structure, managing data transmission efficiently while ensuring a manageable level of scalability. This method optimizes routing decisions based on a general understanding of the network's topology. Position-Based routing focuses on position-awareness for routing decisions, utilizing GPS or location data to facilitate efficient and moderately to highly accurate routing. Leveraging vehicle positions, this approach enables effective data transmission by selecting paths based on location information, offering a reliable means of routing within the network.

Cluster-Based routing contributes to scalability by reducing control overhead. It organizes nodes into clusters, minimizing communication overhead by operating within these clusters. While providing a moderate degree of accuracy, this method enhances efficiency by allowing nodes to communicate within their respective clusters, optimizing resource utilization. Opportunistic routing capitalizes on irregular node connections, leading to erratic accuracy. This method takes advantage of sporadic or intermittent connections between nodes, allowing for routing through available connections without a predefined structure. While offering potential advantages in certain scenarios, its accuracy can vary significantly due to the opportunistic nature of connections. Traffic-Aware routing optimizes routes based on dynamic traffic statuses, considering moderate to extremely accurate traffic circumstances. This method adapts routing decisions based on real-time traffic information, aiming to minimize congestion and delays by selecting paths that consider current traffic conditions. Each routing approach caters to specific VANET scenarios, offering distinct advantages in managing data flow, scalability, accuracy, and adaptability to varying network conditions and requirements.

Algorithms	Accuracy
Positional Based	Moderate - High
Topology Based	Moderate
Cluster Based	Moderate
Traffic aware	Moderate - High
Opportunistic	Low - Moderate

Figure 3. Accuracies of Routing algorithms in VANET

Position-based routing in Vehicular Ad Hoc Networks (VANETs) relies on location information, typically obtained through GPS, to determine the forwarding of data packets. By leveraging geographical coordinates, nodes in the network optimize routing efficiency by transmitting data to the most suitable or closest locations. This approach streamlines communication within mobile networks, capitalizing on location awareness while minimizing the need to constantly update network topology data.

Figure 3 illustrates the varying accuracies of different routing algorithms within VANETs, showcasing the diverse benefits and trade-offs that impact routing precision in dynamic vehicle scenarios. Meanwhile, Figure 4 delineates the various types of algorithms employed in position-based routing, emphasizing the reliance on GPS data to direct data transfer based on node positions, thereby ensuring effective and efficient communication by forwarding packets through the network based on geographical proximity. This method enhances routing effectiveness by utilizing GPS-derived positional data to optimize data transfer and communication pathways within the dynamic landscape of VANETs.

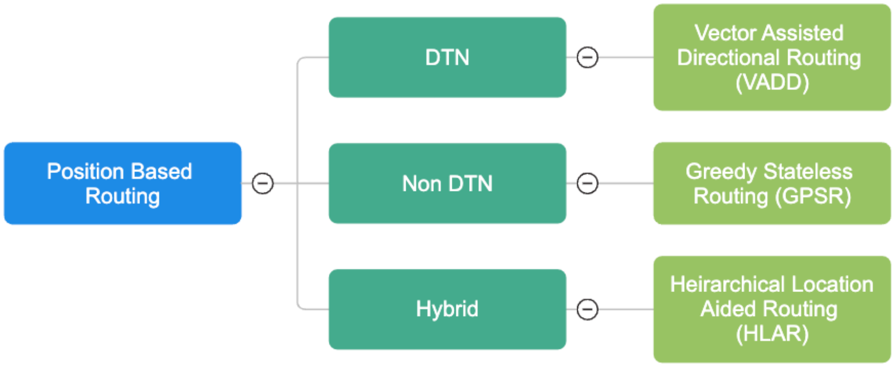


Figure 4. Types of Algorithms in Position Based Routing

Position-based routing in Vehicular Ad Hoc Networks (VANETs) capitalizes on location information, typically sourced from GPS, to determine the most efficient paths for data transmission. By utilizing geographical coordinates, nodes within the network make informed decisions regarding packet forwarding, aiming to deliver data to the nearest or most suitable locations. This strategy enhances communication in mobile networks, leveraging location awareness while reducing the need for continuous updates to network topology data.

Figure 3 illustrates the diverse accuracies associated with various routing algorithms deployed in VANETs. Each algorithm presents unique advantages and trade-offs that influence routing precision within dynamic vehicular environments. Meanwhile, Figure 4 provides a breakdown of different algorithmic approaches used in position-based routing, underscoring the reliance on GPS data to guide data transfer based on node positions. This method optimizes communication efficiency by directing packet forwarding according to geographical proximity, enhancing the effectiveness of data transmission within VANETs.

The effectiveness of position-based routing lies in its ability to leverage GPS-derived positional data for efficient and targeted communication. This approach minimizes routing overhead by relying on location awareness, enabling nodes to make informed forwarding decisions without extensive reliance on network topology updates. Ultimately, position-based routing in VANETs optimizes data transfer, facilitating effective communication pathways adapted to the dynamic nature of vehicular networks while ensuring accurate and efficient forwarding based on node positions.

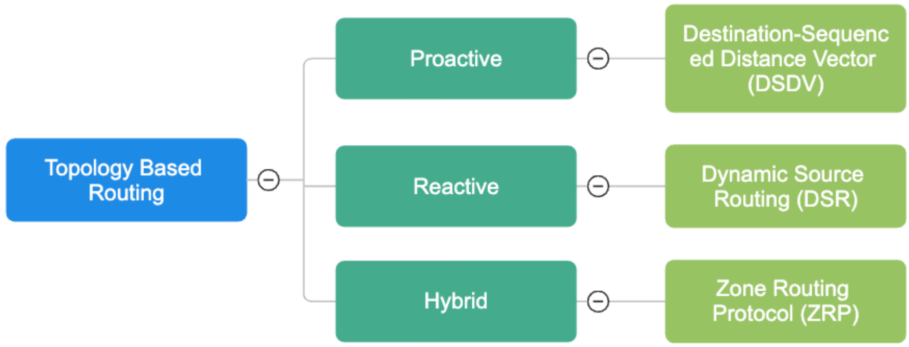


Figure 5. Types of Algorithms in Topology Based routing

Vehicular Ad Hoc Networks (VANETs) employ a cluster-based approach to streamline operations and optimize efficiency by organizing nodes into clusters. This strategic clustering minimizes control overhead and enhances network efficiency through specialized cluster-based routing techniques. At the heart of this framework are cluster chiefs, tasked with overseeing communication both within and between clusters. Their role significantly improves network management and scalability by orchestrating local communication decisions. Figure 6 exemplifies the localized decision-making process inherent in cluster-based routing. Prioritizing localized decisions ensures efficient data transmission within clusters while reducing global traffic, thereby enhancing overall routing efficiency. By delegating decision-making authority to localized cluster heads, this method enhances response times and minimizes latency, crucial in dynamic vehicular environments where swift reactions are essential.

The inherent resilience of this architecture stands out, fortifying the network against sudden and unforeseen changes. By compartmentalizing communication into clusters, the network can dynamically adapt to fluctuations without compromising the system's stability or significantly affecting global operations. This scalable approach proves exceptionally effective for vehicular communication, catering to the unique demands of VANETs. By optimizing communication pathways within clusters and between vehicles, this architecture maximizes efficiency while minimizing control overhead. Its focus on local decision-making not only boosts routing efficiency but also ensures a robust and adaptable network capable of navigating the dynamic and unpredictable nature of vehicular environments.

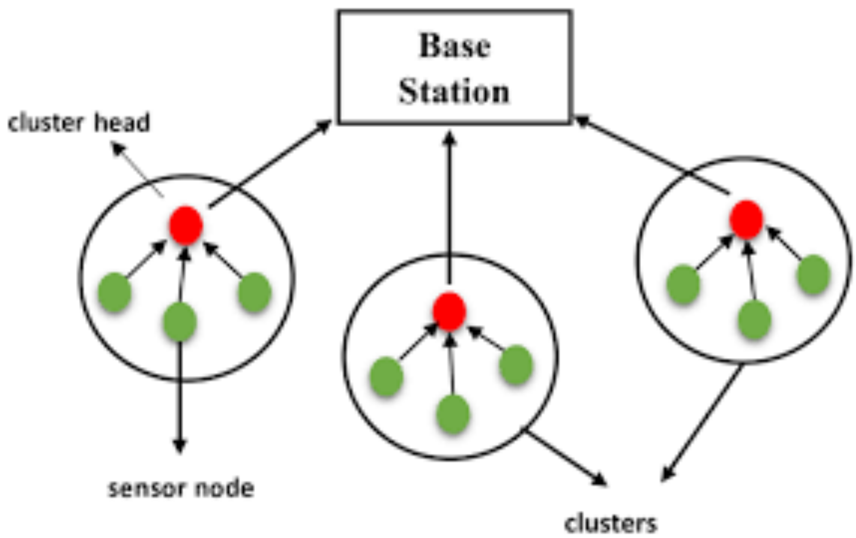


Figure 6. Working of Cluster Based Routing

The Ad-hoc On-Demand Distance Vector (AODV) protocol plays a pivotal role as a reactive routing algorithm within Vehicular Ad Hoc Networks (VANETs), specifically designed to cater to the dynamic and ever-changing nature of vehicular environments. This protocol operates on-demand, establishing routes exclusively between vehicles when essential, thereby significantly reducing routing overhead and optimizing bandwidth utilization in VANETs. Functioning as a distance-vector algorithm, AODV employs sequence numbers to maintain the currency of routing information, enhancing the efficiency of route establishment. When a vehicle aims to communicate with an unidentified location, AODV initiates a route discovery process by broadcasting route request packets among neighboring vehicles. This process continues until the destination is reached or a viable path to the destination is identified.

AODV's adaptability is a standout feature, perfectly suited for scenarios involving fluctuating connections and dynamic topologies characteristic of vehicular networks. Its reactive nature ensures that routes are established only when necessary, conserving resources and minimizing unnecessary traffic on the network. This adaptability allows AODV to efficiently handle changes in connection availability or alterations in network structures, contributing significantly to the efficiency and reliability of communication within VANETs. By dynamically creating routes on demand, leveraging sequence numbers for efficient routing data maintenance, and adapting seamlessly to changing network conditions, AODV emerges as a valuable tool in VANETs. Its ability to respond to the dynamic nature of vehicular environments makes it an integral component in enhancing communication efficiency and reliability among vehicles on the move.

The Destination-Sequenced Distance Vector (DSDV) protocol serves as a proactive routing solution tailored for Vehicular Ad Hoc Networks (VANETs), emphasizing current and consistent data transmission by regularly broadcasting routing tables among nodes. This proactive approach ensures route stability and reliability within the network, essential for stable and predictable environments. A defining feature of DSDV lies in its use of sequence numbers to maintain the freshness and accuracy of routing information. These sequence numbers enable nodes to discern the most recent paths, enhancing the efficiency of route selection and promoting stability by avoiding outdated or erroneous routes. By routinely broadcasting routing tables, DSDV aims to maintain a consistent and up-to-date network topology, crucial for establishing reliable communication paths. However, the periodic broadcasting of routing updates, while ensuring accuracy, contributes to increased control overhead within the network. This aspect becomes a limitation, particularly in highly dynamic vehicular environments where rapid changes in topology and connectivity are common. The frequent updates may strain network resources and lead to inefficiencies, making DSDV less suitable for extremely dynamic VANET scenarios.

Despite this limitation, DSDV excels in providing straightforward routing decisions and loop-free paths within stable networks. It prioritizes route consistency and stability over rapid adaptation, ensuring that nodes can make reliable routing decisions based on consistent and updated information. This aspect makes DSDV a viable solution for scenarios where the network topology undergoes gradual changes or remains relatively stable over time. In summary, DSDV's proactive routing methodology, leveraging sequence numbers for consistent data transmission, ensures route stability and loop-free paths in relatively stable vehicular environments. While its periodic updates might pose challenges in highly dynamic scenarios, DSDV remains a reliable choice for VANETs that prioritize route consistency and stability over rapid adaptation to drastic topology changes.

The Optimised Link State Routing (OLSR) protocol serves as a topology-based routing technique that stands out in Vehicular Ad Hoc Networks (VANETs), particularly in our project's geographical routing application. OLSR's proactive approach involves regularly disseminating comprehensive network topology details among neighboring nodes, ensuring the continuous maintenance and updating of routing tables. This proactive stance guarantees the availability of current and accurate route information, significantly reducing routing overhead and enabling efficient route computation within VANETs. Our selection of OLSR for geographical routing in the project is underpinned by its distinct advantages over conventional distance-vector algorithms like AODV or DSDV. OLSR's topology-based structure aligns closely with the specific demands of VANETs, especially in scenarios where vehicle mobility triggers rapid and frequent topology changes. This structural design empowers OLSR to handle dynamic network conditions more adeptly compared to the reactive approach of AODV or the periodic update model of DSDV.

The proactive nature of OLSR equips it with the agility needed to swiftly adapt to dynamic VANET conditions, fostering the creation of more robust and dependable communication channels. In contrast, reactive protocols like AODV necessitate route discovery only upon demand, potentially causing delays in adapting to topology changes. Similarly, periodic updates in DSDV might lead to outdated information during rapid topology alterations, affecting route efficiency and reliability. Furthermore, OLSR's proactive behavior significantly reduces route finding delays, a critical aspect in time-sensitive VANET applications such as real-time traffic updates or emergency alerts. Its ability to continuously maintain and disseminate up-to-date topology information allows for quicker and more efficient route establishment, facilitating timely and crucial data transmission in dynamic VANET environments. In the dynamic and rapidly evolving landscape of VANETs, OLSR emerges as the optimal choice for fulfilling the geographical routing requirements of our project. Its capacity to store and distribute current topology information seamlessly aligns with our project's needs, enabling the creation of effective routes based on the most recent network status. OLSR's proactive nature, reduced routing overhead, swift adaptation to changes, and minimized route finding delays collectively position it as the ideal protocol to ensure efficient and dependable geographical routing within our VANET project.

E. Testing and Validation:

The validation and testing methodology planned for our project involves rigorous and comprehensive testing of geographic routing protocols within Vehicular Ad Hoc Networks (VANETs). This methodology incorporates elements from referenced papers, integrating a range of strategies and measures to evaluate and enhance the efficiency, security, and performance of these protocols in diverse VANET scenarios. The cornerstone of our testing approach involves the utilization of hybrid optimization algorithms, notably the Optimised Link State Routing (OLSR), to simulate various VANET scenarios. This simulation will encompass different vehicular densities, mobility patterns, and communication ranges to comprehensively assess the geographic routing protocols' efficacy. Efficiency, security, and performance testing will be conducted under these diverse scenarios. Efficiency testing aims to gauge the protocols' ability to manage data flow, latency, and throughput, considering the impact of varying vehicular densities and mobility patterns on routing efficiency. Security measures will also be thoroughly examined to ensure robustness against potential threats and vulnerabilities within the network.

Performance validation will be crucial, involving rigorous assessments against predefined benchmarks. Parameters such as packet delivery ratio, latency, throughput, and scalability will be meticulously evaluated to validate the effectiveness of the implemented routing protocols. The objective is to ensure optimal functionality across a spectrum of VANET conditions, guaranteeing reliable and efficient data transmission. This methodology encompasses a holistic approach by aligning various elements from referenced papers. It integrates the hierarchical network structure to manage data flow efficiently, segmentation strategies for enhanced network organization, security measures to safeguard data transmission, VPN integration for secure communication, and comprehensive testing to evaluate and optimize geographic routing protocols specifically tailored for the dynamic VANET environment. The hierarchical network structure delineates core, distribution, and access layers, ensuring efficient data management considering vehicular movement and communication ranges. Segmentation strategies aid in organizing the network, optimizing resource utilization, and enhancing scalability. Robust security measures, including encryption and authentication, are implemented to safeguard data transmission against potential threats. Moreover, VPN integration ensures secure and private communication among vehicles, reinforcing the confidentiality and integrity of transmitted data. Comprehensive testing aligns with these implemented strategies, evaluating and enhancing the geographic routing protocols to ensure optimal performance, reliability, and security in diverse VANET scenarios. Overall, this methodology represents a comprehensive and systematic approach to validate, evaluate, and optimize geographic routing protocols within VANETs, incorporating elements from referenced papers to address the multifaceted challenges of vehicular environments.

IV. IMPLEMENTATION

The simulation setup utilized a combination of NS3, SUMO, and NetAnim technologies to replicate and evaluate Vehicular Ad Hoc Networks (VANETs). NS3 and SUMO were integrated to facilitate connectivity, enabling SUMO simulations to generate the initial configuration file, osm.sumocfg, which served as a foundational component for the simulations. This integration involved a connection between NS3 and SUMO, allowing for seamless data exchange and interaction between the two platforms. SUMO, a traffic simulation tool, was responsible for generating the initial configuration file, which provided essential parameters and settings for the subsequent simulation runs. Figure 7 depicted the process of simulating vehicles using SUMO within this setup. It illustrated the steps involved in configuring and initiating vehicle simulations, offering insights into how the vehicles were generated, their movement patterns, and the environment in which they operated. SUMO's capabilities allowed for the creation of a realistic vehicular environment, taking into account factors such as road networks, traffic flows, and vehicle behavior.

The collaboration between NS3 and SUMO leveraged their respective strengths, with NS3 focusing on network-level simulations and SUMO providing a detailed representation of vehicular traffic and movement patterns. NetAnim complemented these tools by offering visualization capabilities, allowing for the graphical representation of simulated VANET scenarios, aiding in the analysis and understanding of the network behaviors and outcomes. Together, the integration of NS3, SUMO, and NetAnim facilitated a comprehensive simulation environment for VANETs, combining network-level analysis, realistic traffic simulations, and visualization tools to evaluate and study the dynamics, efficiency, and performance of vehicular networks. Figure 8 shows the Selection of area and Figure 9 shows adding different vechiles.

Figure 10 depicts scenario generation within SUMO, while Figure 11 showcases the concurrent display of simulation-related information in the terminal (the origin of osmWebWizard execution). This simultaneous display in the terminal involves the presentation of simulation details and the storage of configured data into the osm.sumocfg file.

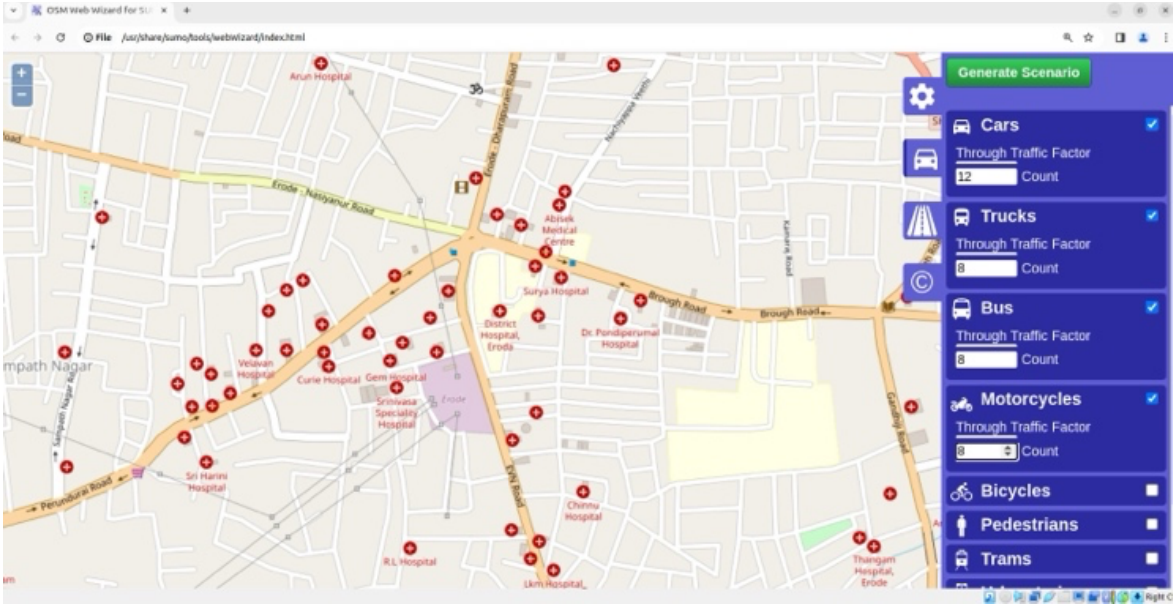


Figure 7. Simulation of Vehicles using SUMO.

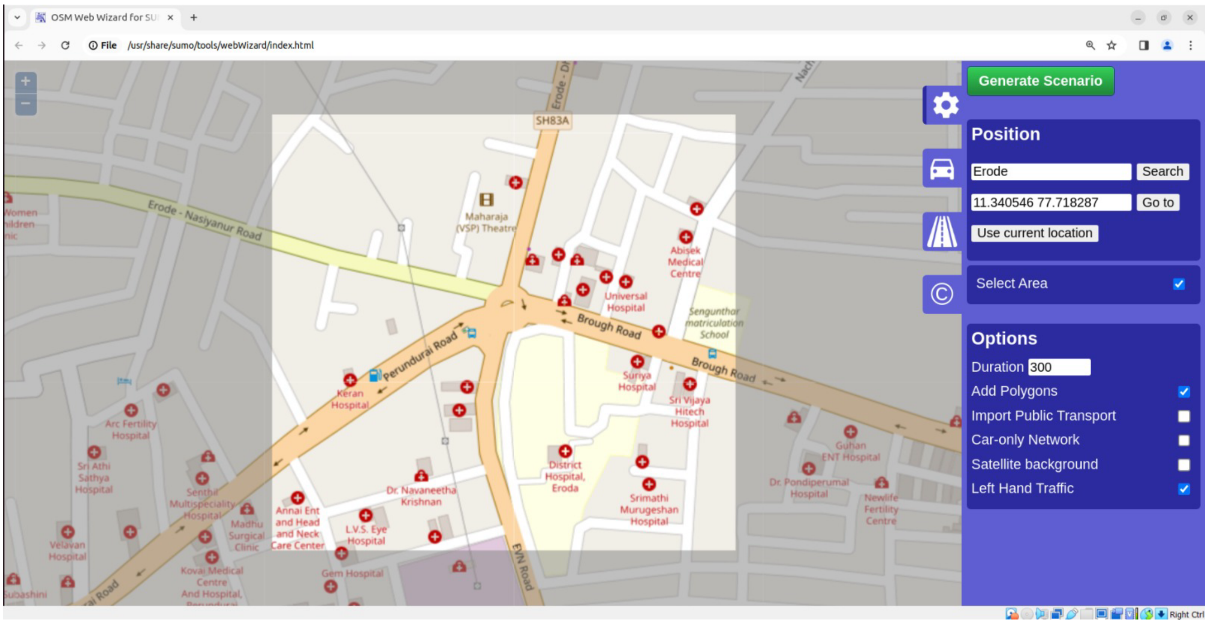


Figure 8. osmWebWizard.py tool. Selecting region and area and setting delay for simulation.

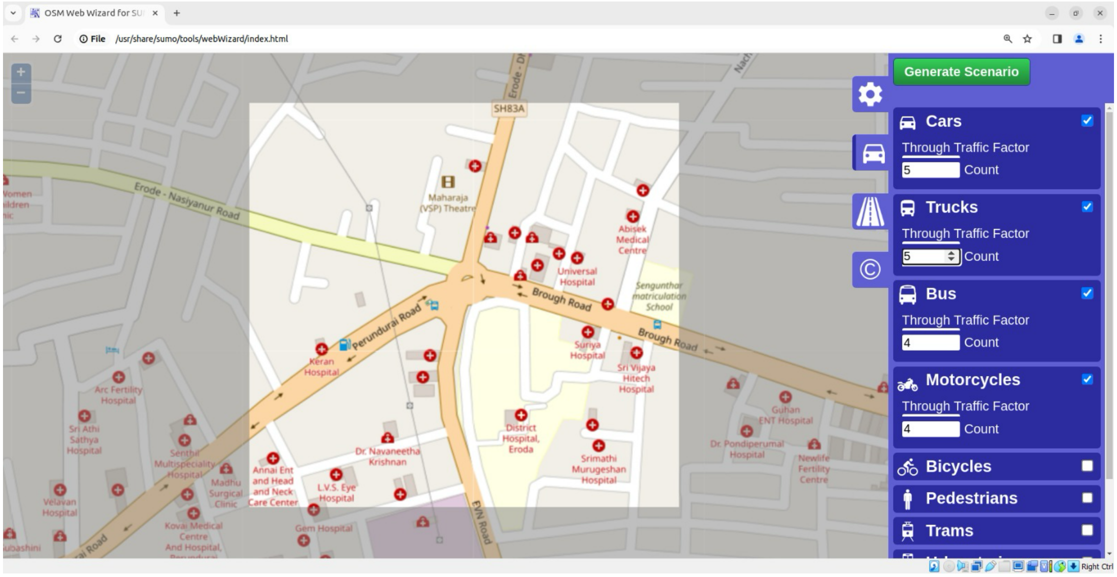


Figure 9. Adding a number of different types of vehicles

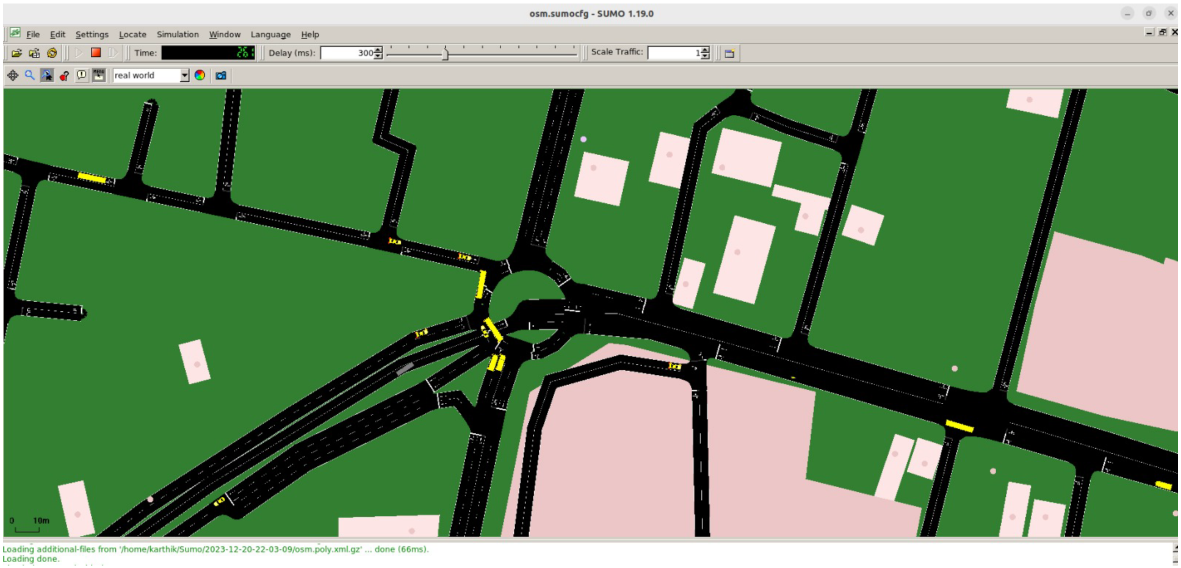


Figure 10. Simulation in Sumo after generating scenario

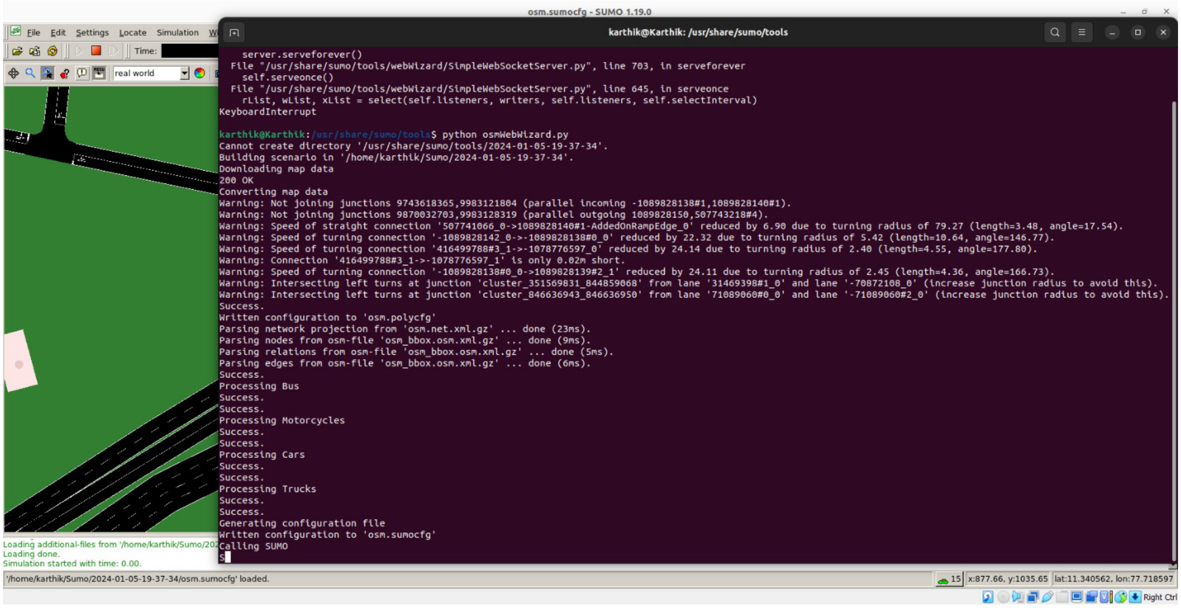


Figure 11. Terminal Displaying simulation info

After completing the simulation, the focal point shifted towards obtaining a crucial output file named vanettrace.xml, housing essential vehicle data. Within our resources, there's the osm.sumocfg file, mirroring the configuration depicted in Figure 2. Notably, Figure 12 displays the acquired vanettrace.xml file, a document rich with intricate vehicle information. This transition in focus post-simulation underscored the significance of extracting specific data encapsulated within the vanettrace.xml file. While the osm.sumocfg file maintained a semblance to the visual reference provided in Figure 2, the real crux lay in the detailed insights held within the vanettrace.xml document, represented in Figure 12. The vanettrace.xml file stands as a repository of critical vehicle metrics, likely encompassing mobility patterns, spatial coordinates, and other pertinent data points crucial for further analysis and inference. This shift in attention signifies a pivotal stage in our study, as this file's contents are anticipated to fuel subsequent analytical processes and enhance our understanding of vehicular dynamics within the simulated environment.

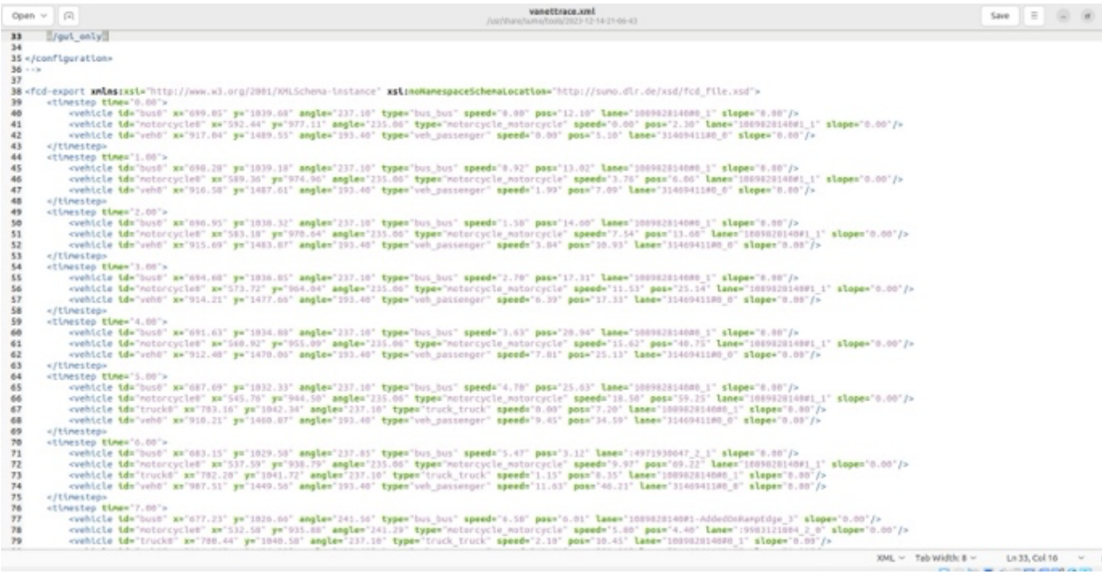
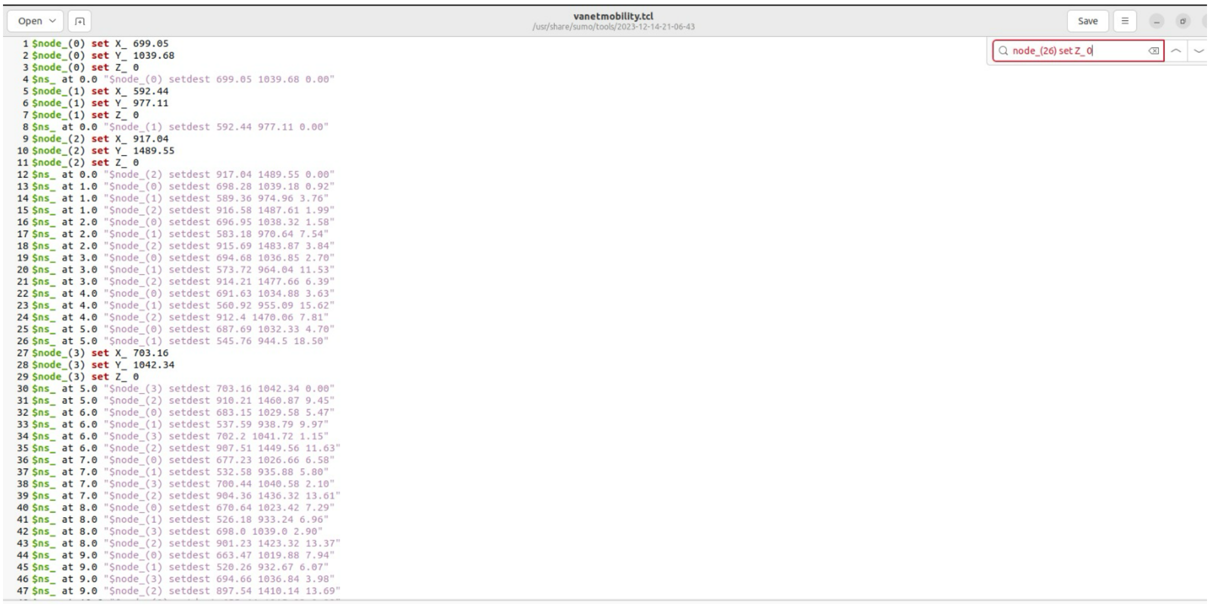


Figure 12. Vanettrace.xml file from SUMO

Using a Python Trace Exporter command, the osm.sumocfg file underwent a transformation into the compatible vanetmobility.tcl format for NS3. This conversion facilitated the adaptation of the initial osm.sumocfg file, ensuring compatibility with NS3's requirements. Figure 13 visually represents the outcome of this process, illustrating the resultant file generated by converting the Vanettrace.xml file into the NS3-compatible vanetmobility.tcl format. This evolution from the osm.sumocfg to the vanetmobility.tcl format stands as a pivotal step, aligning the simulation data with NS3's specifications. Figure 13 encapsulates the culmination of this conversion, showcasing the file essential for seamlessly integrating simulation data into NS3 for further analysis and utilization within the NS3 environment.



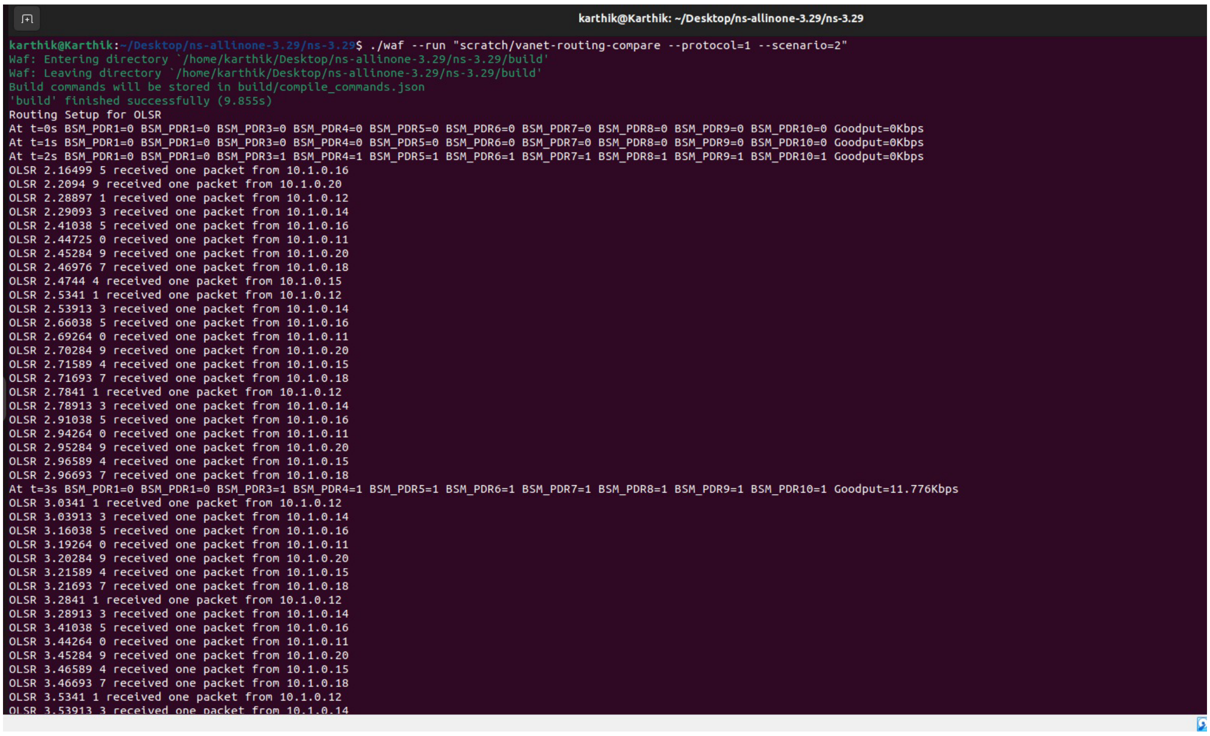

```
$node_(0) set X_ 699.05
$node_(0) set Y_ 1039.68
$node_(0) set Z_ 0
$ns_ at 0.0 "$node_(0) setdest 699.05 1039.68 0.00"
$node_(1) set X_ 592.44
$node_(1) set Y_ 977.11
$node_(1) set Z_ 0
$ns_ at 0.0 "$node_(1) setdest 592.44 977.11 0.00"
$node_(2) set X_ 917.04
$node_(2) set Y_ 1489.55
$node_(2) set Z_ 0
$ns_ at 0.0 "$node_(2) setdest 917.04 1489.55 0.00"
$ns_ at 1.0 "$node_(0) setdest 698.28 1039.18 0.92"
$ns_ at 1.0 "$node_(1) setdest 589.36 974.96 3.76"
$ns_ at 1.0 "$node_(2) setdest 916.58 1487.61 1.99"
$ns_ at 2.0 "$node_(0) setdest 696.95 1038.32 1.58"
$ns_ at 2.0 "$node_(1) setdest 583.18 970.64 7.54"
$ns_ at 2.0 "$node_(2) setdest 915.69 1483.87 3.84"
$ns_ at 3.0 "$node_(0) setdest 694.68 1036.85 2.70"
$ns_ at 3.0 "$node_(1) setdest 573.72 964.04 11.53"
$ns_ at 3.0 "$node_(2) setdest 914.21 1477.66 6.39"
$ns_ at 4.0 "$node_(0) setdest 691.63 1034.88 3.63"
$ns_ at 4.0 "$node_(1) setdest 560.92 955.09 15.62"
$ns_ at 4.0 "$node_(2) setdest 912.4 1470.06 7.81"
$ns_ at 5.0 "$node_(0) setdest 687.69 1032.33 4.70"
$ns_ at 5.0 "$node_(1) setdest 545.76 944.5 18.50"
$node_(3) set X_ 703.16
$node_(3) set Y_ 1042.34
$node_(3) set Z_ 0
$ns_ at 5.0 "$node_(3) setdest 703.16 1042.34 0.00"
$ns_ at 5.0 "$node_(2) setdest 910.21 1460.87 9.45"
$ns_ at 6.0 "$node_(0) setdest 683.15 1029.58 5.47"
$ns_ at 6.0 "$node_(1) setdest 537.59 938.79 9.97"
$ns_ at 6.0 "$node_(3) setdest 702.2 1041.72 1.15"
$ns_ at 6.0 "$node_(2) setdest 907.51 1449.56 11.63"
$ns_ at 7.0 "$node_(0) setdest 677.23 1026.66 6.58"
$ns_ at 7.0 "$node_(1) setdest 532.58 935.88 5.80"
$ns_ at 7.0 "$node_(3) setdest 700.44 1040.58 2.10"
$ns_ at 7.0 "$node_(2) setdest 904.36 1436.32 13.61"
$ns_ at 8.0 "$node_(0) setdest 670.64 1023.42 7.29"
$ns_ at 8.0 "$node_(1) setdest 526.18 933.24 6.96"
$ns_ at 8.0 "$node_(3) setdest 698.0 1039.0 2.90"
$ns_ at 8.0 "$node_(2) setdest 901.23 1423.32 13.37"
$ns_ at 9.0 "$node_(0) setdest 663.47 1019.88 7.94"
$ns_ at 9.0 "$node_(1) setdest 520.26 932.67 6.07"
$ns_ at 9.0 "$node_(3) setdest 694.66 1036.84 3.98"
$ns_ at 9.0 "$node_(2) setdest 897.54 1410.14 13.69"
```

Figure 13. Vanettrace.xml file converted to NS3 format

Modifying the routing function within the primary project code involves a deliberate adjustment where the file path of the trace file, specifically our TCL (Tool Command Language) file, is specified. This critical alteration in the routing function aims to redefine how the code processes and utilizes data, particularly by indicating the precise location of the trace file within the code structure. By explicitly referencing the file path of the trace file in the TCL format, the code gains a targeted understanding of where to access and integrate this specific dataset. This modification serves as a strategic maneuver, enabling the project code to harness and interpret the contents of the designated TCL file, presumably the vanetmobility.tcl generated from the earlier conversion process. This adjustment fundamentally reconfigures how the project code interacts with and incorporates simulation data. It establishes a direct link between the code's routing function and the transformed simulation data, facilitating seamless integration and utilization of this critical information within the project's computational framework. Figure 14 shows the code which has been used to Modify the routing function of main project code by specifying the file path of trace file as our TCL file.

```
  if (m_scenario == 1)
    {
      // 40 nodes in RWP 300 m x 1500 m synthetic highway, 10s
      m_traceFile = "";
      m_logFile = "";
      m_mobility = 2;
      if (m_nNodes == 156)
        {
          m_nNodes = 40;
        }
      if (m_TotalSimTime == 300.01)
        {
          m_TotalSimTime = 10.0;
        }
    }
  else if (m_scenario == 2)
    {
      // Realistic vehicular trace in Erode
      // "low density, 265 total vehicles"
      m_traceFile = "/usr/share/sumo/tools/2023-12-14-21-06-43/vanetmobility.tcl";
      m_logFile = "vanet.log";
      m_mobility = 1;
      m_nNodes = 25;
      m_TotalSimTime = 150.01;
      m_nodeSpeed = 0;
      m_nodePause = 0;
      m_CSVfileName = "vanet.csv";
      m_CSVfileName = "vanet2.csv";
    }
}
```

Figure 14. Modifying the main project code by specifying the file path.



```
karthik@Karthik: ~/Desktop/ns-allinone-3.29/ns-3.29
karthik@Karthik:~/Desktop/ns-allinone-3.29/ns-3.29$ ./waf --run "scratch/vanet-routing-compare --protocol=1 --scenario=2"
Waf: Entering directory `/home/karthik/Desktop/ns-allinone-3.29/ns-3.29/build'
Waf: Leaving directory `/home/karthik/Desktop/ns-allinone-3.29/ns-3.29/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (9.855s)
Routing Setup for OLSR
At t=0s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0 BSM_PDR7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=0Kbps
At t=1s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0 BSM_PDR7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=0Kbps
At t=2s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=1 BSM_PDR4=1 BSM_PDR5=1 BSM_PDR6=1 BSM_PDR7=1 BSM_PDR8=1 BSM_PDR9=1 BSM_PDR10=1 Goodput=0Kbps
OLSR 2.16499 5 received one packet from 10.1.0.16
OLSR 2.2094 9 received one packet from 10.1.0.20
OLSR 2.28897 1 received one packet from 10.1.0.12
OLSR 2.29093 3 received one packet from 10.1.0.14
OLSR 2.41038 5 received one packet from 10.1.0.16
OLSR 2.44725 0 received one packet from 10.1.0.11
OLSR 2.45284 9 received one packet from 10.1.0.20
OLSR 2.46976 7 received one packet from 10.1.0.18
OLSR 2.4744 4 received one packet from 10.1.0.15
OLSR 2.5341 1 received one packet from 10.1.0.12
OLSR 2.53913 3 received one packet from 10.1.0.14
OLSR 2.66038 5 received one packet from 10.1.0.16
OLSR 2.69264 0 received one packet from 10.1.0.11
OLSR 2.70284 9 received one packet from 10.1.0.20
OLSR 2.71589 4 received one packet from 10.1.0.15
OLSR 2.71693 7 received one packet from 10.1.0.18
OLSR 2.7841 1 received one packet from 10.1.0.12
OLSR 2.78913 3 received one packet from 10.1.0.14
OLSR 2.91038 5 received one packet from 10.1.0.16
OLSR 2.94264 0 received one packet from 10.1.0.11
OLSR 2.95284 9 received one packet from 10.1.0.20
OLSR 2.96589 4 received one packet from 10.1.0.15
OLSR 2.96693 7 received one packet from 10.1.0.18
At t=3s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=1 BSM_PDR4=1 BSM_PDR5=1 BSM_PDR6=1 BSM_PDR7=1 BSM_PDR8=1 BSM_PDR9=1 BSM_PDR10=1 Goodput=11.776Kbps
OLSR 3.0341 1 received one packet from 10.1.0.12
OLSR 3.03913 3 received one packet from 10.1.0.14
OLSR 3.16038 5 received one packet from 10.1.0.16
OLSR 3.19264 0 received one packet from 10.1.0.11
OLSR 3.20284 9 received one packet from 10.1.0.20
OLSR 3.21589 4 received one packet from 10.1.0.15
OLSR 3.21693 7 received one packet from 10.1.0.18
OLSR 3.2841 1 received one packet from 10.1.0.12
OLSR 3.28913 3 received one packet from 10.1.0.14
OLSR 3.41038 5 received one packet from 10.1.0.16
OLSR 3.44264 0 received one packet from 10.1.0.11
OLSR 3.45284 9 received one packet from 10.1.0.20
OLSR 3.46589 4 received one packet from 10.1.0.15
OLSR 3.46693 7 received one packet from 10.1.0.18
OLSR 3.5341 1 received one packet from 10.1.0.12
OLSR 3.53913 3 received one packet from 10.1.0.14
```

Figure 15. Executing the code by specifying protocol number,1-OLSR,2-AODV,3-DSDV or 4-DSR

The subsequent phase of our project marked a pivotal juncture necessitating extensive modifications within the NS3 codebase. These alterations were imperative to incorporate advanced visualization functionalities, thereby enriching our understanding of the data. Our efforts began by integrating the Net Animation header file into the code, a fundamental step essential for activating visualization features within the NS3 simulation environment. Additionally, we meticulously embedded the file path directing towards vanetmobility.tcl directly into the codebase. This deliberate step was aimed at establishing a seamless connection between the simulation data and the code execution process.

Ensuring the smooth integration of these modifications demanded further adjustments, particularly in accommodating an expanded maximum number of nodes. This meticulous fine-tuning process involved an in-depth review and comprehensive analysis of the generated mobility file. This thorough examination was instrumental in guaranteeing the configuration of optimal simulation settings, ensuring that our simulations ran with precision and accuracy. The execution of the revised code was conducted within the terminal environment using the NS3 Scratch Tool. At present, the code is operating under protocol 1 within scenario 2, specifically employing the Optimized Link State Routing (OLSR) protocol. The outcome of this execution, visualized in Figure 17, showcases the flow statistics obtained post-simulation. Leveraging our mobility.tcl file, the code successfully generated crucial output files, notably the VanetAnim.xml file. This file was meticulously designed to facilitate visualization through the Net Animation tool. Figures 17 and 18 vividly depict the final simulation, offering a visual representation of the simulated scenario. Moreover, Figure 20 provides an insightful glimpse into the IP and MAC addresses of the simulated nodes. Complementing these efforts, a comprehensive log file was crafted with meticulous detail, serving as a robust resource for debugging and troubleshooting potential inconsistencies encountered during the simulation process. This exhaustive documentation played a critical role in identifying and resolving anomalies, ensuring the integrity and accuracy of our simulation framework.

The execution of the revised code took place within the terminal environment using the NS3 Scratch Tool. At present, our code is configured to operate under protocol 1 within scenario 2, specifically employing the Optimized Link State Routing (OLSR). Figure 17 serves as a visual representation, illustrating the flow statistics gleaned following the code execution.

Leveraging our mobility.tcl file, the code successfully generated crucial output files, notably the VanetAnim.xml file, meticulously tailored for visualization purposes using the Net Animation tool. Figures 17 and 18 stand as comprehensive depictions, showcasing the final simulation displayed through the Net Animation Tool. These visuals offer a detailed and insightful representation of the simulated scenario, providing a clear and comprehensive understanding of the simulation outcomes. Furthermore, Figure 20 delves deeper, offering insights into the specific IP and MAC addresses allocated to the individual nodes within the simulation. These details play a significant role in understanding the network configuration and facilitating any necessary evaluations or comparisons within the simulated environment. The utilization of the mobility.tcl file was instrumental in generating these visual representations and detailed data outputs. The VanetAnim.xml file, specifically tailored for visualization purposes, encapsulates the simulation's dynamics, enabling a more comprehensive comprehension of the network behaviors and interactions between nodes. In summary, through the execution of the code in the specified NS3 environment and the generation of key output files, including the visualization-focused VanetAnim.xml file, our project has achieved a detailed and visual representation of the simulation outcomes. This enables a more profound understanding of the simulated network, facilitating informed analyses and insights into network behaviors and configurations.Additionally, a meticulous log file was meticulously crafted to aid in debugging and troubleshooting potential inconsistencies in the simulation. This log file serves as a robust resource, meticulously documenting the simulation process to assist in identifying and resolving any anomalies encountered during the simulation run.

The flow statistics obtained in scenario 2 using protocol 1 (OLS) with the NS3 Scratch Tool are depicted in Figure 16. Furthermore, the mobility.tcl output showcased node addresses in Figure 19, while the VanetAnim.xml file, derived from the same mobility.tcl output, facilitated visualization through Figures 17 and 18.

In summary, these series of modifications and executions in the NS3 environment were pivotal in integrating visualization features, refining simulation parameters, and documenting the simulation process comprehensively. These steps were crucial in ensuring a robust and accurate simulation framework, aiding in both analysis and troubleshooting throughout the project's development phase.

Figure 16. Flow statistics on execution.

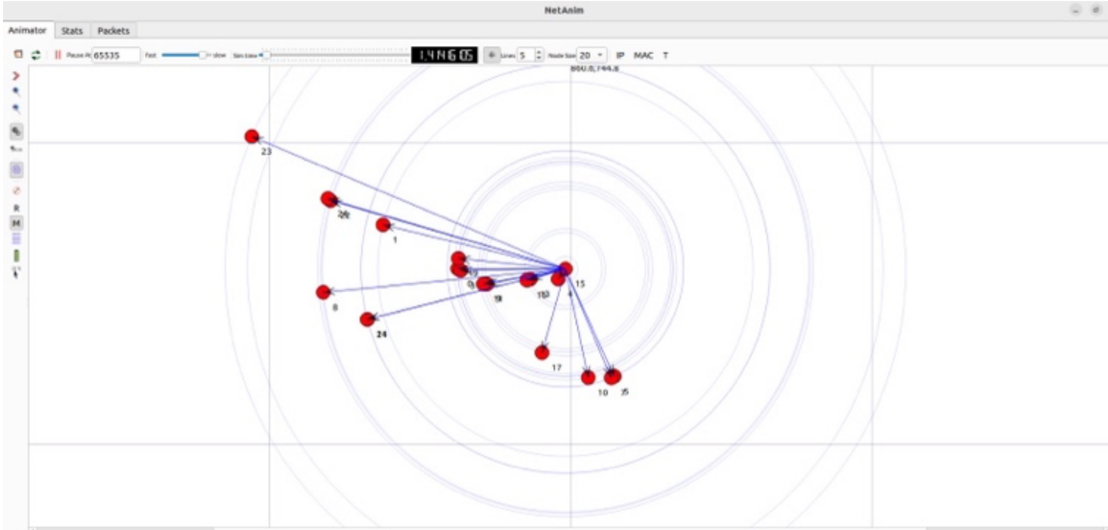


Figure 17. NetAnimation Tool Final Simulation.

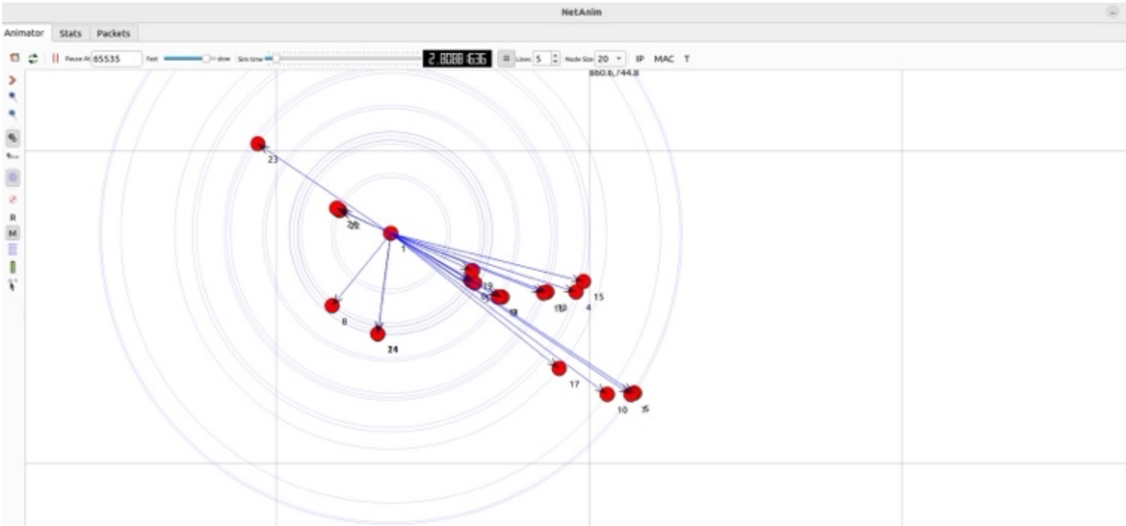


Figure 18. NetAnimation Tool Final Simulation.

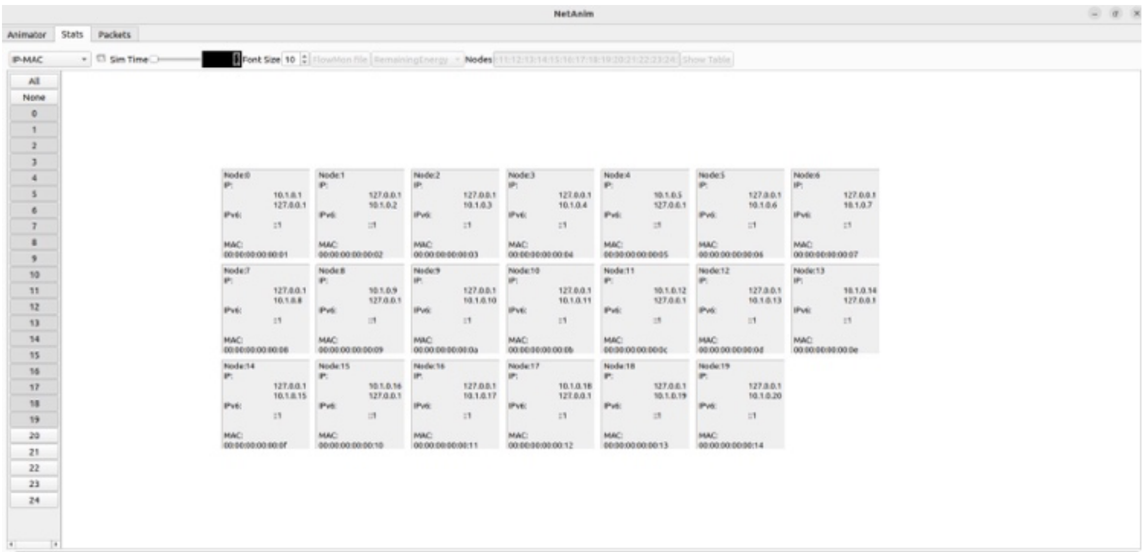


Figure 19. IP and Mac Address of nodes

Within NS3, the NetAnimation Tool emerged as a pivotal asset, offering robust visualization capabilities crucial for unraveling the complex dynamics inherent in the simulated vehicular network. This tool played an instrumental role in providing a tangible and comprehensive representation of the network's intricate behaviors and interactions. At the core of this visualization prowess stood the VanetAnim.xml file, meticulously crafted from the simulation outputs, serving as the cornerstone for these visual representations.

The NetAnimation Tool's significance stemmed from its ability to transform raw data from the simulation into comprehensible visual depictions. The VanetAnim.xml file, generated through a meticulous process using simulation data, encapsulated a wealth of information regarding the vehicular network's behaviors, movements, and interactions. This file essentially served as the blueprint for constructing an accurate and insightful visual narrative of the simulated scenario. Through the visualization facilitated by the NetAnimation Tool, stakeholders gained a clearer understanding of the vehicular network's complexities. The animations and graphical representations derived from the VanetAnim.xml file offered a tangible view of the network's behaviors, aiding in the interpretation of traffic patterns, node movements, and overall network dynamics. Overall, the NetAnimation Tool, powered by the VanetAnim.xml file, played a pivotal role in transforming raw simulation data into visually digestible and meaningful representations. This transformation was crucial in enhancing comprehension and analysis, enabling a deeper exploration and understanding of the intricate workings within the simulated vehicular network.

This file served as a dynamic portrayal of real-time node-to-node communication within the virtual network, offering a live glimpse into the simulated environment's inner workings. Its role as a visual gateway was pivotal, granting insights into the intricate interactions between cars navigating through the network. The visualization software was a crucial asset, rendering a wealth of critical data readily accessible and easily comprehensible. Notably, essential network properties like IP and MAC addresses of specific nodes were graphically depicted alongside the dynamic movements of the vehicles. This visual representation provided a granular understanding of the network's architecture and the nuanced interactions among its elements. Its value lay in its meticulous display of vital information, presenting a comprehensive overview of the network's intricacies.

Moreover, the graphical display encompassed an array of crucial metrics including throughput, latency, jitter, loss ratio, delivery ratio, transmitted and received packets, and packet losses. This extensive presentation offered detailed insights into the network's operations across various scenarios. It facilitated the identification of trends, anomalies, and bottlenecks within the network infrastructure, thereby enabling informed analysis and practical optimization strategies. Figure 20 specifically illustrates the flow-level statistics of AODV (Ad hoc On-Demand Distance Vector) protocol, providing a visual representation of the protocol's performance within the simulated environment. This level of detailed visualization not only elucidated the operational aspects of specific protocols but also facilitated a deeper understanding of their impact on the network's overall functionality and efficiency.

Figure 21 shows the flow level statistics of DSDV. Figure 22 shows the flow level statistics of OLSR. The combination of immediate visualisation and comprehensive statistical analysis enabled a comprehensive understanding of the behaviour of the simulated network. It was an effective tool for learning how different factors affected network performance, testing the simulation model, and evaluating the effectiveness of routing algorithms.

```
Flow Id:1
======
UDP 10.1.0.16/49153--->10.1.0.6/9

Tx bitrate:2.94896kbps
Rx bitrate:2.85981kbps
Mean delay:6.88319ms
Packet Loss ratio:8.28829%

timeFirstTxPacket= 1.41016e+09ns
timeFirstRxPacket= 1.41501e+09ns
timeLastTxPacket= 1.4991e+11ns
timeLastRxPacket= 1.32411e+11ns
delaySum= 3.50354e+09ns
jitterSum= 3.34545e+09ns
lastDelay= 3.50354e+09ns
txBytes= 54740
rxBytes= 46828
txPackets= 595
rxPackets= 509
lostPackets= 46
timesForwarded= 73

delayHistogram nBins:961
Index:0 Start:0 Width:0.001 Count:465
Index:1 Start:0.001 Width:0.001 Count:12
Index:2 Start:0.002 Width:0.001 Count:5
Index:4 Start:0.004 Width:0.001 Count:3
Index:5 Start:0.005 Width:0.001 Count:3
Index:7 Start:0.007 Width:0.001 Count:3
Index:8 Start:0.008 Width:0.001 Count:1
Index:9 Start:0.009 Width:0.001 Count:1
Index:10 Start:0.01 Width:0.001 Count:1
Index:12 Start:0.012 Width:0.001 Count:3
Index:13 Start:0.013 Width:0.001 Count:2
Index:18 Start:0.018 Width:0.001 Count:1
Index:19 Start:0.019 Width:0.001 Count:1
Index:22 Start:0.022 Width:0.001 Count:1
Index:45 Start:0.045 Width:0.001 Count:1
Index:211 Start:0.211 Width:0.001 Count:1
Index:251 Start:0.251 Width:0.001 Count:1
Index:461 Start:0.461 Width:0.001 Count:1
Index:501 Start:0.501 Width:0.001 Count:1
Index:710 Start:0.71 Width:0.001 Count:1
Index:960 Start:0.96 Width:0.001 Count:1

jitterHistogram nBins:961
```

```
Flow Id:2
======
UDP 10.1.0.18/49153--->10.1.0.8/9

Tx bitrate:2.94896kbps
Rx bitrate:2.93373kbps
Mean delay:0.899829ms
Packet Loss ratio:0.524476%

timeFirstTxPacket= 1.46671e+09ns
timeFirstRxPacket= 1.46976e+09ns
timeLastTxPacket= 1.49967e+11ns
timeLastRxPacket= 1.44218e+11ns
delaySum= 5.12002e+08ns
jitterSum= 6.90009e+08ns
lastDelay= 5.12002e+08ns
txBytes= 54740
rxBytes= 52348
txPackets= 595
rxPackets= 569
lostPackets= 3
timesForwarded= 59

delayHistogram nBins:85
Index:0 Start:0 Width:0.001 Count:535
Index:1 Start:0.001 Width:0.001 Count:9
Index:2 Start:0.002 Width:0.001 Count:3
Index:3 Start:0.003 Width:0.001 Count:4
Index:4 Start:0.004 Width:0.001 Count:1
Index:5 Start:0.005 Width:0.001 Count:2
Index:6 Start:0.006 Width:0.001 Count:1
Index:8 Start:0.008 Width:0.001 Count:3
Index:9 Start:0.009 Width:0.001 Count:1
Index:12 Start:0.012 Width:0.001 Count:2
Index:13 Start:0.013 Width:0.001 Count:1
Index:15 Start:0.015 Width:0.001 Count:1
Index:20 Start:0.02 Width:0.001 Count:1
Index:23 Start:0.023 Width:0.001 Count:1
Index:27 Start:0.027 Width:0.001 Count:1
Index:33 Start:0.033 Width:0.001 Count:1
Index:37 Start:0.037 Width:0.001 Count:1
Index:84 Start:0.084 Width:0.001 Count:1

jitterHistogram nBins:85
Index:0 Start:0 Width:0.001 Count:515
Index:1 Start:0.001 Width:0.001 Count:9
Index:2 Start:0.002 Width:0.001 Count:7
```

Figure 20. Flow level statistics for AODV

```
Flow Id:1
======
UDP 10.1.0.16/49153--->10.1.0.6/9

Tx bitrate:2.94898kbps
Rx bitrate:2.60807kbps
Mean delay:4.53379ms
Packet Loss ratio:17.3986%

timeFirstTxPacket= 2.16016e+09ns
timeFirstRxPacket= 2.16499e+09ns
timeLastTxPacket= 1.4991e+11ns
timeLastRxPacket= 1.40161e+11ns
delaySum= 2.21702e+09ns
jitterSum= 4.14176e+09ns
lastDelay= 2.21702e+09ns
txBytes= 54464
rxBytes= 44988
txPackets= 592
rxPackets= 489
lostPackets= 103
timesForwarded= 34

Packets Dropped:

No Route:32
TTL Expire:7

Bytes Dropped:

No Route:2944
TTL Expire:644

delayHistogram nBins:2008
Index:0 Start:0 Width:0.001 Count:467
Index:1 Start:0.001 Width:0.001 Count:14
Index:2 Start:0.002 Width:0.001 Count:1
Index:4 Start:0.004 Width:0.001 Count:2
Index:9 Start:0.009 Width:0.001 Count:1
Index:15 Start:0.015 Width:0.001 Count:1
Index:16 Start:0.016 Width:0.001 Count:1
Index:18 Start:0.018 Width:0.001 Count:1
Index:2007 Start:2.007 Width:0.001 Count:1

jitterHistogram nBins:2007
Index:0 Start:0 Width:0.001 Count:474
```

```
Flow Id:2
======
UDP 10.1.0.20/49153--->10.1.0.10/9

Tx bitrate:2.94933kbps
Rx bitrate:2.87056kbps
Mean delay:0.401788ms
Packet Loss ratio:7.9566%

timeFirstTxPacket= 2.20262e+09ns
timeFirstRxPacket= 2.2094e+09ns
timeLastTxPacket= 1.40203e+11ns
timeLastRxPacket= 1.32715e+11ns
delaySum= 2.0451e+08ns
jitterSum= 1.12563e+08ns
lastDelay= 2.0451e+08ns
txBytes= 50876
rxBytes= 46828
txPackets= 553
rxPackets= 509
lostPackets= 44
timesForwarded= 2

Packets Dropped:

No Route:0
TTL Expire:9

Bytes Dropped:

No Route:0
TTL Expire:828

delayHistogram nBins:17
Index:0 Start:0 Width:0.001 Count:493
Index:1 Start:0.001 Width:0.001 Count:8
Index:2 Start:0.002 Width:0.001 Count:3
Index:4 Start:0.004 Width:0.001 Count:1
Index:6 Start:0.006 Width:0.001 Count:1
Index:12 Start:0.012 Width:0.001 Count:1
Index:16 Start:0.016 Width:0.001 Count:2

jitterHistogram nBins:17
Index:0 Start:0 Width:0.001 Count:495
Index:1 Start:0.001 Width:0.001 Count:2
Index:2 Start:0.002 Width:0.001 Count:6
```

Figure 21. Flow level statistics for DSDV

```
Flow Id:1
======
UDP 10.1.0.16/49153--->10.1.0.6/9

Tx bitrate:2.94896kbps
Rx bitrate:2.92141kbps
Mean delay:0.355324ms
Packet Loss ratio:7.20721%

timeFirstTxPacket= 1.41016e+09ns
timeFirstRxPacket= 1.41511e+09ns
timeLastTxPacket= 1.4991e+11ns
timeLastRxPacket= 1.31161e+11ns
delaySum= 1.82992e+08ns
jitterSum= 9.34322e+07ns
lastDelay= 1.82992e+08ns
txBytes= 54740
rxBytes= 47380
txPackets= 595
rxPackets= 515
lostPackets= 40
timesForwarded= 45

delayHistogram nBins:24
Index:0 Start:0 Width:0.001 Count:501
Index:1 Start:0.001 Width:0.001 Count:12
Index:4 Start:0.004 Width:0.001 Count:1
Index:23 Start:0.023 Width:0.001 Count:1

jitterHistogram nBins:24
Index:0 Start:0 Width:0.001 Count:509
Index:1 Start:0.001 Width:0.001 Count:2
Index:4 Start:0.004 Width:0.001 Count:1
Index:22 Start:0.022 Width:0.001 Count:1
Index:23 Start:0.023 Width:0.001 Count:1

packetSizeHistogram nBins:5
Index:4 Start:80 Width:20 Count:515

flowInterruptionsHistogram nBins:7
Index:6 Start:1.5 Width:0.25 Count:1
```

```
Flow Id:2
======
UDP 10.1.0.18/49153--->10.1.0.8/9

Tx bitrate:2.94896kbps
Rx bitrate:2.93861kbps
Mean delay:0.418559ms
Packet Loss ratio:0.34904%

timeFirstTxPacket= 1.46671e+09ns
timeFirstRxPacket= 1.46954e+09ns
timeLastTxPacket= 1.49967e+11ns
timeLastRxPacket= 1.44481e+11ns
delaySum= 2.38997e+08ns
jitterSum= 1.39402e+08ns
lastDelay= 2.38997e+08ns
txBytes= 54740
rxBytes= 52532
txPackets= 595
rxPackets= 571
lostPackets= 2
timesForwarded= 38

delayHistogram nBins:15
Index:0 Start:0 Width:0.001 Count:549
Index:1 Start:0.001 Width:0.001 Count:6
Index:2 Start:0.002 Width:0.001 Count:2
Index:3 Start:0.003 Width:0.001 Count:3
Index:4 Start:0.004 Width:0.001 Count:1
Index:5 Start:0.005 Width:0.001 Count:3
Index:6 Start:0.006 Width:0.001 Count:4
Index:7 Start:0.007 Width:0.001 Count:2
Index:14 Start:0.014 Width:0.001 Count:1

jitterHistogram nBins:15
Index:0 Start:0 Width:0.001 Count:543
Index:1 Start:0.001 Width:0.001 Count:2
Index:2 Start:0.002 Width:0.001 Count:6
Index:3 Start:0.003 Width:0.001 Count:2
Index:4 Start:0.004 Width:0.001 Count:6
Index:5 Start:0.005 Width:0.001 Count:5
Index:6 Start:0.006 Width:0.001 Count:5
Index:14 Start:0.014 Width:0.001 Count:1

packetSizeHistogram nBins:5
Index:4 Start:80 Width:20 Count:571
```

Figure 22. Flow level statistics for OSLR

	AODV	DSDV	OLSR
Flow Id	1	1	1
Tx Bitrate	2.94896 kbps	2.94898 kbps	2.94896 kbps
Rx Bitrate	2.85981 kbps	2.60807 kbps	2.92141 kbps
Mean Delay	6.88319 ms	4.53379 ms	0.355324 ms
Packet Loss Ratio	8.28829%	17.3986%	7.20721%
Time First Tx Packet	1.41016e+09 ns	2.16016e+09 ns	1.41016e+09 ns
Time First Rx Packet	1.41501e+09 ns	2.16499e+09 ns	1.41511e+09 ns
Time Last Tx Packet	1.4991e+11 ns	1.4991e+11 ns	1.4991e+11 ns
Time Last Rx Packet	1.3241e+11 ns	1.40161e+11 ns	1.31161e+11 ns
Delay Sum	3.50354e+09 ns	2.21702e+09 ns	1.82992e+08 ns
Bitrate Sum	3.34545e+09 ns	4.14176e+09 ns	9.34322e+07 ns
Last Delay	3.50354e+09 ns	2.21702e+09 ns	1.82992e+08 ns
Tx Bytes	54740	54464	54740
Rx Bytes	46828	44988	47380
Tx Packets	595	592	595
Rx Packets	509	489	515

Table 1. Comparison of flow level statistics of AODV, DSDV and OLSR.

Table 1 serves as a comparative snapshot, juxtaposing the flow level statistics among AODV (Ad hoc On-Demand Distance Vector), DSDV (Destination-Sequenced Distance Vector), and OLSR (Optimized Link State Routing) protocols. This meticulous comparison offered a concise yet comprehensive overview of their respective performances within the simulated vehicular communication system. The ability to visualize data comprehensively emerged as a critical asset, enabling a deeper understanding and facilitating well-informed decision-making processes. These visual representations were instrumental in unraveling practical insights that directly contributed to enhancing the reliability and efficiency of vehicular communication systems. By juxtaposing the flow level statistics, Table 1 provided a structured comparison of crucial metrics such as throughput, latency, packet delivery, and loss ratios among the different routing protocols. This comparative analysis facilitated a nuanced comprehension of each protocol's performance in handling vehicular communication scenarios. The practical utility of this visualization lay in its capacity to aid in making informed judgments regarding protocol selection and optimization strategies. These insights guided improvements in the dependability and effectiveness of vehicular communication systems, contributing significantly to their overall functionality and performance within dynamic vehicular networks.

V. CONCLUSION

The thorough implementation and assessment of geographic routing protocols in VANETs demonstrated significant findings. By simulating vehicles with SUMO and then converting the data into NS3 format, three well-known protocols OLSR, DSDV, and AODV were executed one by one. By means of comprehensive simulations and Net Animation visualisation, comparision of statistics demonstrated differing protocol performances.

We inferred that OLSR turned out to be the most optimised protocol, demonstrating improved performance in terms of important metrics including packet loss ratio, mean delay, transmitted bytes (txbytes), jitter sum, and final delay. Its capacity to reduce latency, control packet loss, and keep delay metrics constant indicates that it is appropriate for geographic routing in VANETs. since of its strong performance, OLSR is the best option since it provides efficient and dependable communication, which is essential in dynamic road settings.

ACKNOWLEDGMENTS

The authors wish to extend their gratitude to Thangam S for invaluable guidance and support throughout the research process. The simulation and execution of results were conducted by M Vivek Srikar Reddy, Karthik Ullas, Abdul Khadar Jilani, and K Venkatesh under the guidance of Ms. J Jesy Janet Kumari.

REFERENCES

[1] Punitha, A., Manickam, J. and LEO, M., 2018. Location Verification Technique for Secure Geographical Routing in VANET.

[2] Wu, T.Y., Wang, Y.B. and Lee, W.T., 2018. Mixing greedy and predictive approaches to improve geographic routing for VANET. Wireless Communications and Mobile Computing, 12(4), pp.367-378.

[3] Smiri, S., Boushaba, A., Abbou, R.B. and Zahi, A., 2018, April. Geographic and topology based routing protocols in vehicular ad-hoc networks: Performance evaluation and QoS analysis. In 2018 international conference on intelligent systems and computer vision (IScV) (pp. 1-8). IEEE.

[4] Boussoufa-Lahlah, S., Semchedine, F. and Bouallouche-Medjkoune, L., 2018. Geographic routing protocols for Vehicular Ad hoc NETworks (VANETs): A survey. Vehicular communications, 11, pp.20-31.

[5] Hassan, A.N., Abdullah, A.H., Kaiwartya, O., Cao, Y. and Sheet, D.K., 2018. Multi-metric geographic routing for vehicular ad hoc networks. Wireless Networks, 24, pp.2763-2779.

[6] Parvin, S., Sarram, M.A., Mirjalily, G. and Adibnia, F., 2019. A survey on void handling techniques for geographic routing in VANET network. International Journal of Grid & Distributed Computing, 8(2), pp.101-114.

[7] Jiang, S., Huang, Z. and Ji, Y., 2020. Adaptive UAV-assisted geographic routing with Q-learning in VANET. IEEE Communications Letters, 25(4), pp.1358-1362.

[8] Hou, S., Hu, Y. and Tian, L., 2021. A Survey on Geographic location routing and forwarding strategies of VANET and WSN based on NDN. In Journal of Physics: Conference Series (Vol. 1738, No. 1, p. 012040). IOP Publishing.

[9] Dr. Thangam S. and Kirubakarkan, "QoS-aware Service Selection Architecture for Pervasive Environment", European Journal of Scientific Research, pp. 159- 168, 2012.

[10] Sailaja, P., Ravi, B. and Jaisingh, T., 2018, April. Performance analysis of AODV and EDAODV routing protocol under congestion control in VANETs. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 945-948). IEEE.

[11] Manjula, P. and Priya, S.B., 2022, April. Detection of falsified selfish node with optimized trust computation model in chimp-AODV based WSN. In 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC) (pp. 52-57). IEEE.

[12] Sathianadhan, N., Gangadharan, S.S. and Narayanan, G., 2014, May. A delay-based optimum routing protocol scheme for collision avoidance applications in VANETs. In 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-5). IEEE.

[13] Karanam, V., Maheswari, B.U. and Sudarshan, T.S.B., 2017, August. Overlay based fault tolerant peer to peer multicasting for emergency data communication in VANETS. In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon) (pp. 510-513). IEEE.