Analysis of Routing Algorithms in Various Traffic Scenarios (VANET)

Da Teng

Jiacheng Zhang

Georgia Institute of Technology

May 1, 2024

Table of Content

1. Introduction

Routing algorithms play a pivotal role in the efficient functioning of communication networks, determining the paths through which data packets are transmitted from source to destination. Among the myriad of routing algorithms, Ad-hoc On-demand Distance Vector (AODV), Destination-Sequenced Distance Vector (DSDV), and Optimized Link State Routing (OLSR) are prominent contenders, each designed with unique characteristics to address specific network scenarios. In this project, we delve into the analysis of these routing algorithms under varying traffic conditions, encompassing high and low pedestrian traffic as well as high and low vehicular traffic. The primary objective of this study is to evaluate the performance of these algorithms in terms of metrics including delay time, packet loss ratio, and number of forwarded times, discerning their suitability for deployment in real-world scenarios. By scrutinizing the performance of AODV, DSDV, and OLSR under different traffic scenarios, this project aims to provide insights into the strengths and weaknesses of each algorithm, aiding network administrators and researchers in making informed decisions regarding protocol selection and network design.

The importance of this project lies in its potential to enhance the reliability, scalability, and efficiency of communication networks, particularly in scenarios characterized by dynamic traffic conditions. By identifying the most suitable routing algorithm for specific environments, network operators can optimize resource utilization, mitigate congestion, and improve overall network performance. Furthermore, the findings of this study can inform the development of novel routing protocols tailored to address the unique challenges posed by pedestrian and vehicular traffic, contributing to the advancement of mobile ad-hoc networks (MANETs) and vehicular ad-hoc networks (VANETs). Ultimately, the knowledge gained from this research can pave the way for more resilient and responsive communication infrastructures, capable of supporting a wide range of applications, from urban surveillance to intelligent transportation systems.

2. Technical Background

2.1.1. Ad-hoc On-demand Distance Vector (AODV)

AODV is a reactive routing protocol designed for MANETs, where nodes communicate with each other without the need for a fixed infrastructure. It operates on a "route-on-demand" basis, meaning routes are only established when needed. When a node wants to send data to another node, it initiates a route discovery process by broadcasting a route request (RREQ) packet. Intermediate nodes receiving the RREQ either forward it or respond if they have a route to the destination or are the destination itself. Once the RREQ reaches the destination or a node with a valid route to the destination, a route reply (RREP) is generated and sent back to the source along the reverse path. AODV maintains routing tables at each node to store information about active routes, and routes are updated as needed based on route maintenance messages.

2.1.2 Destination-Sequenced Distance Vector (DSDV):

DSDV is a proactive routing protocol that employs the concept of distance vectors with sequence numbers to maintain routes in a MANET. Unlike reactive protocols like AODV, DSDV continuously updates routing information to all nodes in the network, regardless of whether they are actively involved in data transmission. Each node maintains a routing table containing entries for all reachable destinations along with their associated sequence numbers. Periodic updates are exchanged between neighboring nodes to disseminate routing information. Sequence numbers are used to ensure the freshness of routing information and to prevent routing loops. When changes occur in the network topology, such as link failures or node mobility, DSDV quickly converges to new routes by propagating updates throughout the network.

2.1.3 Optimized Link State Routing (OLSR):

OLSR is a proactive routing protocol designed specifically for mobile ad-hoc networks characterized by high mobility and frequent topology changes. It operates based on the concept of multipoint relays (MPRs), which are selected nodes responsible for flooding control messages efficiently. OLSR reduces control message overhead by having only MPRs forward link-state information, rather than every node in the network. Each node selects a set of MPRs based on its 2-hop neighborhood, ensuring that every node has at least one MPR within its transmission range. This reduces redundant retransmissions and conserves network bandwidth. OLSR maintains a topology database containing information about neighboring nodes and their links, from which it calculates shortest paths to all destinations using Dijkstra's algorithm. This proactive approach allows OLSR to provide low-latency routes and adapt quickly to changes in network topology.

3. Methodology

3.1. Simulation Setup

This project involves simulating various traffic scenarios using the SUMO (Simulation of Urban Mobility) software. Four different scenarios were chosen: high pedestrian traffic on the Georgia Tech campus, low pedestrian traffic on the Georgia Tech campus, high vehicular traffic in Times Square, New York City, and low vehicular traffic in Fredericksburg, Texas. To configure each simulation, parameters were set using the OSMWebwizard tool according to the desired scenario. The simulation period was set to 1000 seconds to ensure that the nodes (vehicles or pedestrians) reached an equilibrium state during the simulation, as longer periods would take too long to generate and shorter periods would not allow the nodes to stabilize.

After generating the scenarios, the SUMO software was used to convert the sumocfg file into an XML file, which was then converted into a TCL file using a built-in Python script. The TCL file was then filtered using a customized script to extract time periods of 40% and 50% of the total simulation time. This filtering process served two purposes: it significantly reduced the simulation time for the NS3 software, which was used for further analysis, and it allowed us to focus on the equilibrium state of the simulation, which is the desired state for each scenario.

By filtering out the initial and final stages of the simulation, we were able to isolate the period when the nodes were in a stable state, reflecting the desired scenario. This approach enabled us to efficiently analyze the traffic scenarios and extract meaningful insights. The use of SUMO and NS3 software, combined with our customized filtering script, allowed us to simulate and analyze complex traffic scenarios in an efficient and effective manner.

3.2. Code Setup

3.2.1. Wi-Fi PHY Layer/Channel Setup

The configuration of the Wi-Fi physical (PHY) layer and the channel is established for a network simulation utilizing the 802.11b standard to make the network compatible with older but widely adopted communication protocols.

For channel setup, two significant models are employed: the Constant Speed Propagation Delay Model and the Friis Propagation Loss Model. The former ensures a straightforward and constant signal propagation time between nodes, calculated by the distance and a fixed speed of light, which is a typical assumption in a simulation environment for ease of computation. The Friis model is a fundamental approach to representing signal strength loss over distance and especially applies to open-space scenarios.

Further, an ad-hoc network is simulated using WifiMacHelper, which sets up each node with an ad-hoc Wi-Fi Media Access Control (MAC) layer. This configuration implies a network where each node can directly communicate with others in a decentralized manner, which is typical in scenarios without fixed infrastructure, such as emergency response situations or temporary gatherings.

Mobility patterns are introduced into the simulation through the Ns2MobilityHelper, which utilizes trace files produced by SUMO—a simulation tool for urban mobility. These traces encapsulate the movement patterns of the nodes within the simulation, providing a realistic representation of node mobility. By calling the Install() method, these patterns are applied to the nodes, thereby influencing the dynamic topology of the network and allowing the simulation to mimic real-world mobile network environments.

3.2.2. IP Address Dynamic Allocation

Since the node count is based on the SUMO trace file, IP Address Exhaustion will frequently happen if the testing area is enormous and contains a lot of cars (nodes). So we introduced IP address dynamic allocation of IP addresses based on the node count derived from a SUMO trace file. It leverages the Ipv4AddressHelper class to adaptively configure the network's IP addressing scheme to accommodate the varying number of nodes that a SUMO simulation may yield.

Here is an example of implementing the IP address dynamic allocation. For a network with up to 254 nodes, it configures a standard subnet, assigning IP addresses in the range "10.1.1.0" to "10.1.1.254" with a subnet mask of "255.255.255.0". This subnetting allows for a single subnet with 254 usable IP addresses. When the node count extends to 510, the code selects a wider subnet by adjusting the subnet mask to "255.255.254.0", doubling the available addresses to encompass a range suitable for up to 510 nodes.

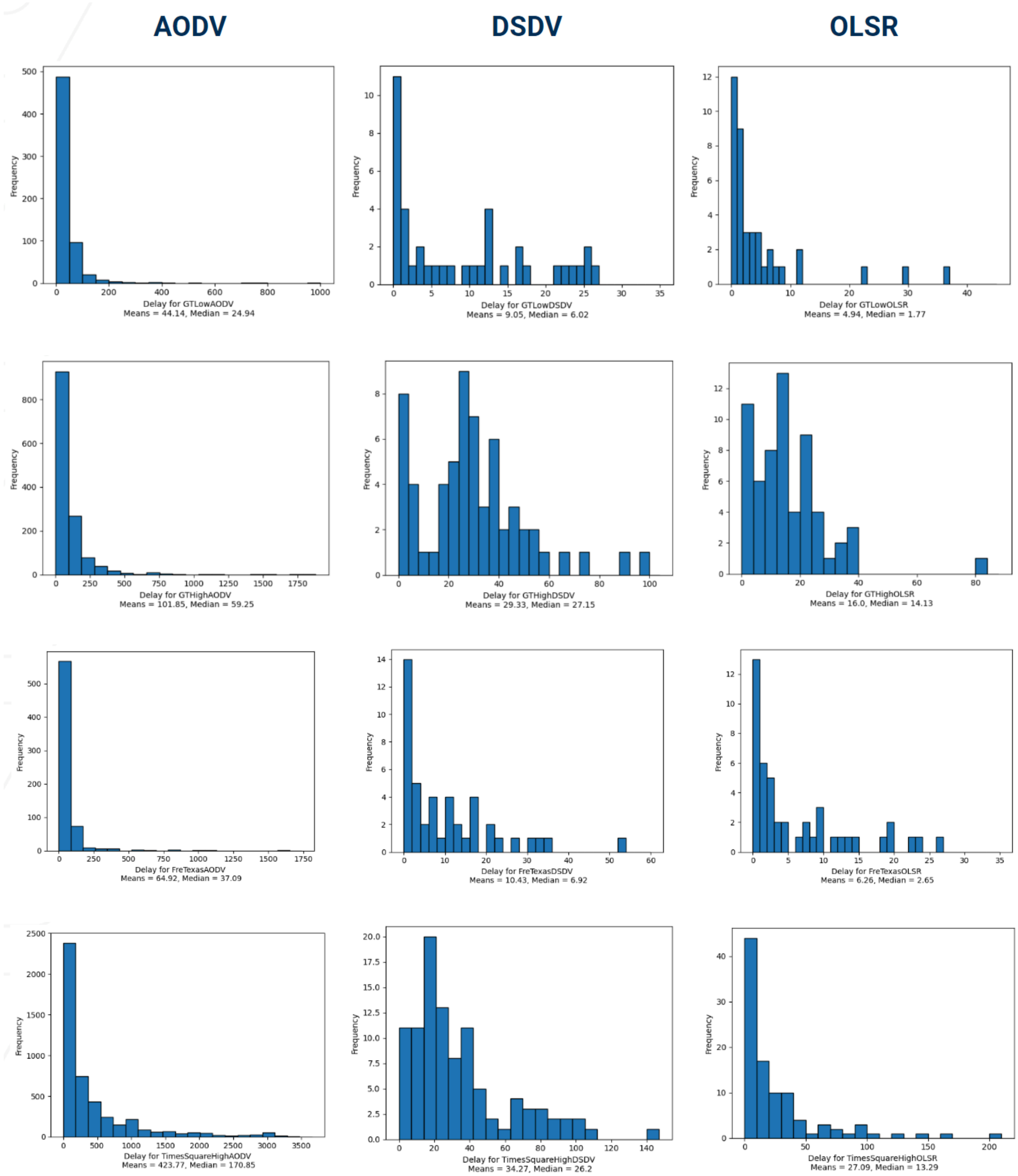
3.2.3. Packet Sending/Receiving Mechanism

We established packet communication between nodes in an ad-hoc network. Each node in the simulation is configured to send packets to its immediately preceding node, starting from the second node in the sequence. This setup is intended for performance analysis of network traffic and routing algorithms under controlled conditions.

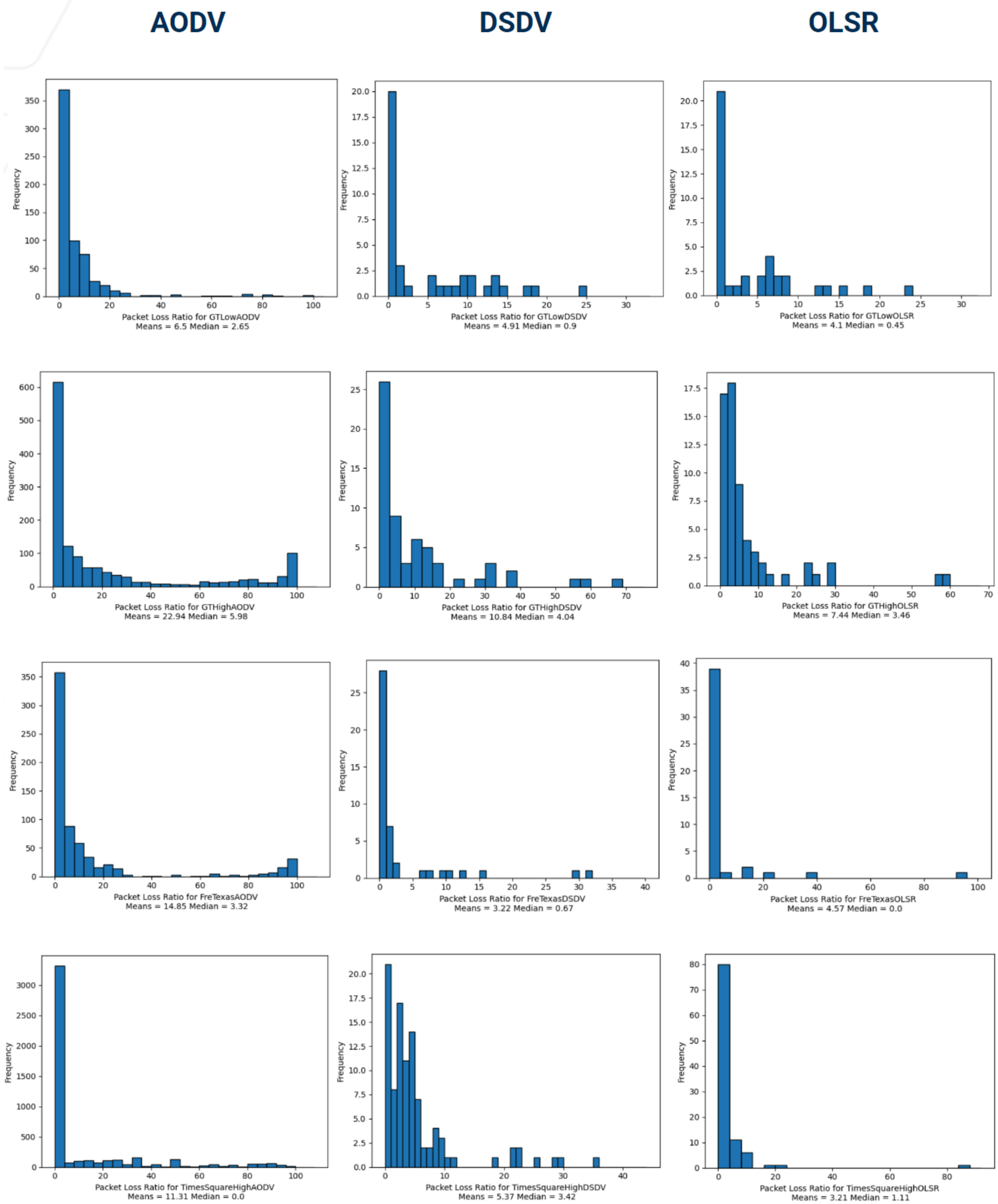
In our implementation, senderIndex and receiverIndex are defined to establish a sending relationship where each node targets the node just before it in the sequence for packet delivery. The socket sink is configured on the receiver node to accept incoming packets. This setup is crucial for capturing and processing the packets transmitted by the sender nodes. An OnOffHelper object is utilized to manage the packet-sending process. It employs the UDP protocol, setting the destination address and port for the packets. The OnTime is configured to a constant value of 1.0, indicating continuous packet transmission when the application starts running. The OffTime is set to 0.0, ensuring that once the packet transmission begins, it does not stop until the simulation ends.

4. Result

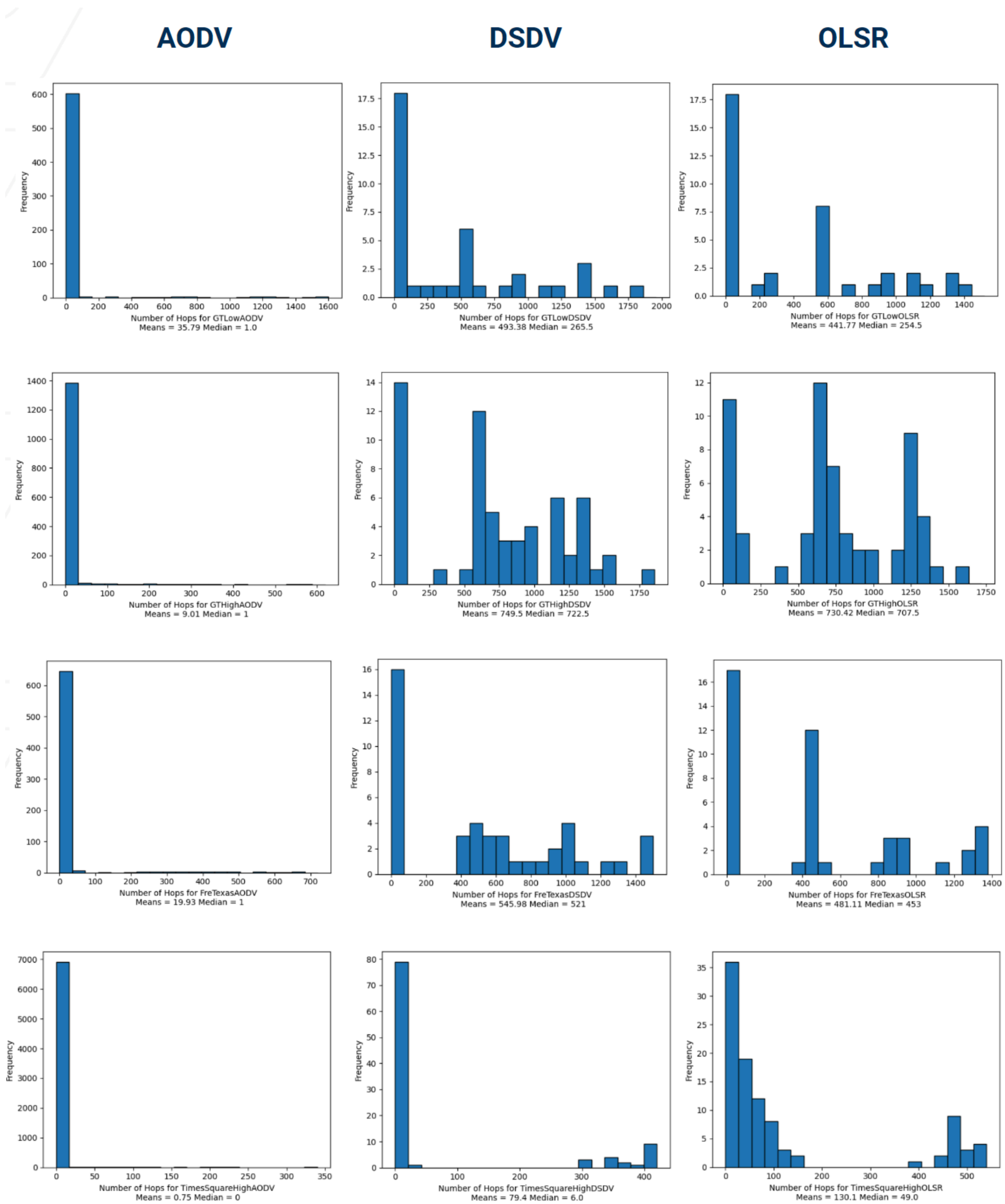
4.1. Delay



4.2. Packet Loss Ratio



4.3. Number of Hops



5. Conclusion

5.1. Delay Time Analysis

AODV consistently demonstrates higher delay times across all traffic scenarios compared to DSDV and OLSR. Particularly in high vehicular traffic scenarios, AODV exhibits significantly higher mean delay times, indicating congestion and inefficient routing.

DSDV generally shows lower delay times compared to AODV, especially in low pedestrian and low vehicular traffic scenarios. However, its delay times increase notably in high pedestrian and high vehicular traffic scenarios, albeit remaining lower than those of AODV.

OLSR exhibits the lowest delay times among the three algorithms in most scenarios, indicating its efficiency in data packet transmission. However, similar to DSDV, OLSR experiences increased delay times in high pedestrian and high vehicular traffic scenarios, albeit remaining lower than those of AODV.

5.2. Packet Loss Ratio Analysis

AODV generally exhibits higher packet loss ratios compared to DSDV and OLSR across all traffic scenarios. In high pedestrian and low vehicular traffic scenarios, AODV has the highest mean packet loss ratio, indicating its vulnerability to congestion and mobility-induced disruptions. However, in high vehicular traffic scenarios, AODV shows relatively lower packet loss ratios compared to low and high pedestrian traffic scenarios.

DSDV consistently demonstrates lower packet loss ratios compared to AODV and sometimes outperforms OLSR, especially in low pedestrian and low vehicular traffic scenarios. This suggests that DSDV's proactive approach to route maintenance and stability contributes to reduced packet loss even in dynamic network conditions.

OLSR exhibits competitive packet loss ratios, particularly in low pedestrian and high vehicular traffic scenarios, where it performs comparably to DSDV. However, OLSR's performance deteriorates in high pedestrian traffic scenarios, where it shows higher packet loss ratios compared to DSDV.

5.3. Number of Hops Analysis

AODV demonstrates varying numbers of hops depending on the traffic scenario, with the lowest mean number of hops observed in high vehicular traffic scenarios. This suggests that AODV adapts its routing decisions based on the traffic density, preferring shorter paths in high vehicular traffic scenarios to minimize delay and packet loss.

DSDV consistently shows higher numbers of hops compared to AODV and OLSR across all traffic scenarios. Particularly in low pedestrian and high pedestrian traffic scenarios, DSDV exhibits significantly higher mean numbers of hops, indicating less efficient route selection.

OLSR generally maintains lower numbers of hops compared to AODV and DSDV, especially in low pedestrian and low vehicular traffic scenarios. However, similar to AODV, OLSR exhibits the lowest mean number of hops in high vehicular traffic scenarios, indicating its adaptability to varying traffic densities.

5.4. Overall Conclusion

Overall, the analysis reveals that while AODV may offer advantages in specific traffic scenarios such as high vehicular traffic, DSDV and OLSR generally demonstrate more stable performance across diverse traffic conditions, with DSDV excelling in stability and OLSR showing efficiency in delay-sensitive applications.

6. Future Works

In the future, we plan to develop more sophisticated evaluation metrics to assess our system's performance in terms of energy consumption, bandwidth, fairness, scalability, etc. This suggests a move towards a more comprehensive and multi-dimensional analysis to optimize system performance. Additionally, we aim to expand our testing to highway vehicles to research potential applications in the transportation or automotive industries. We have generated a SUMO trace file for a specific area around i-285. However, we still need to figure out how to run the simulation efficiently in our implementation. Moreover, we propose to enhance the complexity of our sending/receiving mechanism and integrate the DSR (Dynamic Source Routing) algorithm. These enhancements aim to make our system more robust and efficient in real-world and dynamic environments. And also makes our performance analysis of different routing algorithms more comprehensive and accurate.

Reference

1. Perkins, C. E., & Royer, E. M. (2020). Ad-hoc on-demand distance vector routing. Sun Microsystems Laboratories Advanced Development Group, Menlo Park, CA.
2. He, G. Destination-sequenced distance vector (DSDV) protocol. Helsinki University of Technology, Networking Laboratory.
3. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2009). Optimized link state routing protocol for ad hoc networks. Hipercom Project, INRIA Rocquencourt.