AN EVALUATION AND COMPARISON OF AD HOC ROUTING ALGORITHMS FOR THE PURPOSE OF AUTONOMOUS VEHICLE CONTROL

by

Joseph Ethan Mata

A Thesis submitted to the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Auburn, Alabama
August 5, 2017

Approved by

Dr. Alvin Lim, Chair, Professor of Computer Science and Software Engineering
Dr. Cheryl Seals, Member, Professor of Computer Science and Software Engineering
Dr. David Umphress, Member, Professor of Computer Science and Software Engineering

ABSTRACT

The capability to communicating through a decentralized network with no pre-existing structure is invaluable in numerous different scenarios. In order to effectively utilize these networks, an efficient routing algorithm must be used. These algorithms allow devices to send packets through a method besides solely flooding them through the network. In this paper, the researcher simulates the operation of a remotely controlled vehicle in a static arrangement of wireless devices using the Ns2 simulation application. The OLSR, AODV, DSR, and DSDV routing algorithms are used in this simulation and were compared in various scenarios. Several follow-up simulations were performed with changes to the traffic flows. In addition to increasing the required throughput for the network, the packet sizes were altered as well to examine the impact of sending fewer larger packets. The series of simulations found that in general, AODV and DSDV routed packets with the least latency on average. Additionally, the results showed that AODV and DSR had the most reliable routing, each with over 80% of packets delivered in all simulations.

ACKNOWLEDGEMENTS

I would like to thank Dr. Lim for the advice and feedback he has provided in my studies at Auburn University. Additionally, I would like to thank the advisory committee members, Dr. Cheryl Seals and Dr. David Umphress.

I would also like to thank my parents for their support and encouragement to pursue graduate studies.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 History of Wireless Ad Hoc Networks

As wireless transmission technologies continue to increase in speed, it has become increasingly feasible for wireless ad hoc networks to be used in new scenarios and for new purposes. While the predecessor to these networks, Packet Radio Networks, was originally developed in 1972, mobile ad hoc networks underwent a significant amount of development during the 1990s. During this time, an associativity-based routing system was developed by Chai-Keong Toh, proving the feasibility of wireless ad hoc networking [16]. Since then, numerous routing protocols have been developed to improve the scalability and reliability of these networks.

1.2 Current Applications

In addition to providing localized connectivity in the absence of wireless routers, ad hoc networks have been utilized in numerous other applications. One notable use is in wireless sensor networks, where a large number of sensor devices are deployed to an area to collect data. In this system, the sensor devices cooperatively pass data to each other in order to reach a gateway node, where the data is then gathered by a separate computer.

Wireless ad hoc network functionality has also been adopted into cellular phones, in order to communicate without the use of cellular towers or data plans. One instance of this is the social messaging application FireChat. This application uses peer-to-peer transmissions to pass information through a network of mobile phones and tablets to provide communication capabilities without the use of existing infrastructure [17]. This type of application can be expected to be seen more widely as Apple Inc. has implemented a framework for ad hoc networking, MultipeerConnectivity, into their line of iPhones starting with the iPhone 7 [18]. This framework provides functionality for communicating not just across one link, but also through a network of phones. By using this technology, app developers will be capable of creating applications, which use these networks without the additional work of designing the networking protocols. In 2014, iPhones made up over 40% of smartphone subscribers, so device compatibility as well as adoption are not likely to impede the use of these types of networks [18].

1.3 Benefits of Wireless Ad Hoc Networks

Wireless ad hoc networks allow for a resilient infrastructure that can be easily deployed to a location [2]. No work must be done to configure the routing as the implemented routing protocols discover nearby nodes and create the routing table. This is beneficial, since these devices may be set up in environments where wireless antennas may be damaged or obstructed, as well as locations where the infrastructure must be set up quickly. The routing algorithms present on the network nodes allow for a reasonably quick recovery from damage to the network, and although the new route may not be as optimal, the network remains functional while the damaged or missing device is replaced. This is in part because several routing algorithms do not depend on knowledge of the entire system, which causes repairs to typically be local to the immediate area affected. This allows wireless networks to exist in areas which prevent traditional wired networks from existing, such as disaster areas. Disaster relief, in particular, is a viable application for autonomous robots [1]. Further increasing the operating range of these devices would be highly beneficial to disaster relief efforts.

Chapter 2

Motivation for Work

2.1 Introduction

As wireless communication technologies have evolved, wireless ad hoc networks have remained a useful system for connecting multiple independent devices into a network. No overarching setup is required, so these networks are by nature easily expanded and modular. Because packets sent through these networks must undergo several retransmissions as they traverse the network, they may experience increased latency in their delivery. Additionally, if a route to the destination is not known when the packet must be sent, all packets to that location must wait until one is discovered. Because of these additional potential delays, it remains imperative to select a routing algorithm, which best fits, the application in order to select routes that minimize the number of delays.

2.2 Usage in Autonomous Guided Vehicles

One promising usage of these mobile networks is to manage Autonomous Guided Vehicles (AGVs). These vehicles are robots that are capable of navigating autonomously to a given location. Although the step-by-step navigation is controlled entirely by the autonomous vehicle, high-level instructions are still required in order to set destinations and actions. As the costs for these robots decrease, they will likely become more and more present in both logistics and manufacturing companies [13]. One real-world example of this type of system is a robot being used to select items from shelves in Amazon.com. Inc.'s warehouses. These robots are used to retrieve items that had been ordered, and transport them to the packing area. Although capable of routing themselves around obstacles, by nature of filling a request, they still require the instructions as to what inventory items are needed [3].

2.2.1 Improvements to Existing Autonomous Guided Vehicle Systems

Traditional AGVs would significantly benefit from the ability to receive instructions and information over an ad hoc network. One of the primary methods for directing these vehicles is the use of metal wires beneath the floor surface that directs the vehicle's path [13]. Reconfiguring these routes requires removing and replacing the tape to set new routes, which could be avoided by the use of wireless instructions to the vehicle. Using an ad hoc network of wireless devices reduces the burden of connecting numerous wireless access points through the use of cables. This would greatly reduce the cost of reconfiguring warehouses and factories to more productive and efficient layouts. This also prevents a single router's malfunction from crippling the remainder of the network as the ad hoc network can simply reroute around the damaged device.

2.2.2 Requirements Specific to Autonomous Guided Vehicle Management

There are several unique requirements of this type of application. If this vehicle is being used for streaming data collection, such as surveillance camera footage, the routing must be quick enough to ensure that the packet reaches the control station reliably and in a timely manner. Otherwise, the information may reach the controller too late for a response to changing conditions. Furthermore, the communications to the vehicle must have guaranteed delivery, since future instructions may rely on previous commands. For example, an instruction to a robot to pick up an item for a shipment is entirely dependent on the robot being at the correct position with the correct heading. Additional verification may be required for high importance tasks such as interacting with complex objects at the vehicle’s destination, or where there must be human oversight for the decisions being made

2.3 Usage in Disaster Relief Scenarios

Another potential usage of this technology is in the area of disaster relief. With advances in technology, remotely controlled robots are being considered for use in searching rubble and debris for survivors. In areas where the infrastructure has been damaged, the usage of wireless devices to provide a temporary network would greatly expand the operating range of these systems. This kind of network setup could also be used when communicating with first-responders in an area in order to deliver important updates.

2.3.1 Requirements Specific to Disaster Relief Usage

This application would need to minimize the delay in transmitting information across the network, and also to be resilient in the event that one of the wireless relays is damaged. Especially when responding to natural disasters such as flooding where the environment is still hazardous, there is a notable risk for water damage or falling debris to damage or block the wireless relays. The changes to the network as the robot moves would be relatively small; this usage of wireless ad hoc networks would have relatively few mobile devices, so changes to the network would generally occur near the moving robots as they enter and leave the range of the stationary devices. Current robots being designed for the purpose of disaster relief are working under the assumption of very limited communication, so the usage of wireless ad hoc networks is not out of the question for this scenario [14].

2.4 Limitations of Wireless Ad Hoc Networks

While wireless ad hoc networks do provide scalable and resilient networks, some problems still remain. Particularly, the delays due to routing through a dynamic network may cause too much overall latency for certain real-time applications. In the case of the AGVs, feedback for decisions may reduce the overall efficiency of the system. Similarly, for disaster relief robots, directions for the robot must be prompt in order for the robot to act in a timely manner. If not, the robot risks making decisions that put it or the lives it is trying to save in danger. In a worst-case scenario, the environment may change enough such that the decision is no longer quick enough to be relevant. An example of this might be that an instruction to move to a location is received before the robot's report that a hazard is present can reach the controller. Both of these scenarios would suffer from slow or non-reactive routing, which can be mitigated by selecting and implementing a routing algorithm suited for the specific application.

Chapter 3

Background and Previous Work

3.1 Introduction

In a mobile wireless ad hoc network, the destination within the network is likely not within the transmission range of the source of the packet. Therefore, it is necessary to route the packet through the network of devices to reach the destination. While it is possible to transmit these packets through a flooding mechanism where each network device rebroadcasts each received packet, the use of more efficient routing algorithms allows for a more effective use of the network's resources. There are significant differences between these routing algorithms with regard to their priorities and operation. For example, some algorithms actively compute the route to all nodes within the network, while some wait until a destination is provided. This allows routes to be set in advance, although this adds some unnecessary traffic to the network. Additionally, routing information can be contained within the data packet itself, or saved in each device within the network.

3.2 DSR Operation

The Dynamic Source Routing algorithm, or DSR algorithm, sends the full route along with each packet. The routing is performed on-demand, so no routing occurs before a packet needs to be sent. When sending a packet, if there is a route to the destination, then the cached route is used. Otherwise, the device begins using the route discovery protocol [4]. As these routes are discovered, they are saved to the source's memory in a cache. During the discovery process, a route request packet (RREQ) is flooded through the network. The flooding mechanism works by requiring each device to rebroadcast received packets to the device's neighbors. Upon receipt of the RREQ packet, the previous hop is appended to the route record contained in the packet. This route record is used to track the overall route that each request packet has taken. Once the RREQ packet reaches the destination, a route reply packet (RREP) is sent back to the source of the request. If there is already a route to that device, then it is used. Otherwise, the route record is reversed and used to return the reply. Each device along the route caches the route for use at a later time. This allows each device along the route to use the established route if they need to send packets to the same location [4].

3.2.1 DSR Route Maintenance

The data link layer may encounter errors during the transmission of packets through the network. When this happens, the device that encountered the error must refresh its route cache. This is done by removing any route that depends on the link that was broken. In addition to removing the route that no longer works, any other routes that use the hop that encountered the error are also removed. To detect these errors, devices using DSR may also use acknowledgements to ensure that packets have been delivered successfully. If the connection is bidirectional, this acknowledgement may also be done passively by detecting the retransmission of the sent packet [22, p.315].

3.3 AODV Operation

Similar to DSR, the Ad-hoc On-Demand Distance Vector Routing (AODV) algorithm determines the route only when a packet is ready to be sent. However, this algorithm keeps the routing information in tables on each device rather than sending the entire route along with the packet [5]. AODV uses nearly the same method of sending RREQ and RREP packets as DSR, with some notable changes. As in DSR, RREQ packets are broadcast by devices in the network to forward it to the destination. These request packets do not have a route record; instead each device's routing table is updated with new information as RREQ packets are received by that device. Once the RREQ packet reaches the destination or a device with a current route to that destination, the RREP packet is sent back from that device to the source. As this packet returns to the source for the route, the intermediate devices set their next hop to be the source of the packet as they receive it [22, p 314].

3.3.1 AODV Route Maintenance

AODV uses a timer for each of the entries in its routing table. If the route is not used within the time limit, it is removed from the table. Additionally, when a device sees that its downstream neighbors have left its range, it removes that route from its table. When this occurs, it also alerts other devices in that route of the link failure. This propagates through the entire route, until the source is reached. This then re-triggers the route discovery process [22, p 314].

3.4 OLSR Operation

The Optimized Link State Routing algorithm (OLSR) attempts to reduce the work required by the network by establishing and maintaining routing information before route is needed. This is accomplished by each device periodically broadcasting packets to inform its geographic neighbors of itself. These packets are designated as HELLO packets. As devices learn their immediate neighbors, these neighbors are also sent in the periodic updates. Because of this, each device

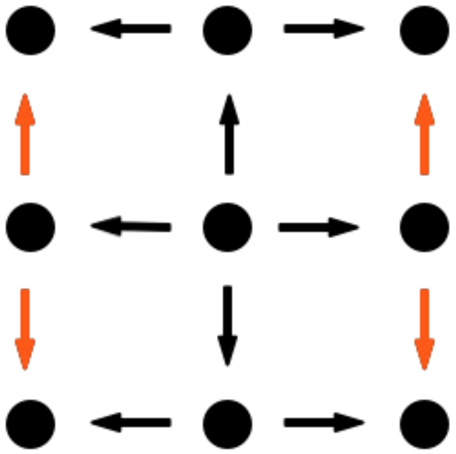

Figure 1: Example of redundant transmissions preventable through OLSR's MPR feature

will eventually learn its 1-hop and 2-hop neighbors in the network [6]. This information is used to establish Multipoint Relays (MPRs), which are designated devices that are required to forward packets to flood the network. Each device selects from its 1-hop neighbors a subset that can reach all 2-hop neighbors. The list of MPRs selected by a given device is included in its HELLO packets. By requiring only certain devices to rebroadcast packets, this reduces the strain on the network as only some of the neighbors may be needed to reach all the 2-hop neighbors of the source [6]. For example, as seen in Figure 1, the unnecessary forwards are highlighted in red. By only having the top and bottom center nodes act as MPRs and rebroadcast the packet, there is a significant reduction in transmissions. Once setup, this infrastructure is used to propagate Topology Control (TC) messages through the network. Since these TC packets are sent regularly, the MPR system provides a constant benefit through the entire time the routing algorithm is active. The TC packets contain the connected devices as well as the last hop for that connection. In addition to periodic broadcasts, these TC packets are also transmitted when a change is made to the list of MPRs a given device has chosen [6].

3.4.1 OLSR Route Maintenance

OLSR maintains its routes through the use of holding times on entries to its topology table. These entries are typically refreshed by incoming TC packets, however, if a node moves outside the range of another node, the second node will eventually remove the entry corresponding to the first entry. When this deletion of topography entries occurs, the routing table is recalculated [6].

3.5 DSDV Operation

The Destination Sequenced Distance Vector Algorithm (DSDV) is a routing algorithm that utilizes the Bellman-Ford algorithm to determine routes. This allows it to determine the best route for a packet, even if loops exist in the network [22, p 309]. In this algorithm, routing advertisements are broadcast periodically, as well as when new routing information is obtained [7]. These route broadcasts consist of the destination address, the cost of the route, a sequence number associated with the source of the information, as well as an additional sequence number associated with the broadcast itself. Each device in the network tracks information for every other device, consisting of a cost metric such as the number of hops and the route. When receiving packets, the device compares the sequence number and metric against previously held routes. The most recent route is used, and in the case of a tie, the route with the lowest metric is used [22, p 309].

3.5.1 DSDV Route Maintenance

DSDV actively updates its routes, as the periodic broadcasts ensure that the most recently discovered routes are used. The updates can be optimized in order to reduce the traffic on the network by sending information in a manner similar to many streaming video encodings. Periodic full updates are sent, which are called dumps. These allow devices that join the network to quickly learn the routing information discovered by the other devices. Because this is a significantly large update, smaller incremental packets are sent which only reflect the changes. By doing this, the algorithm still ensures that newcomers to the network obtain the full information within a certain time, while also reducing the average size of the updates [22, p 310].

3.6 Prior Work Introduction

These algorithms were selected for this comparison in order to cover the major categories of routing techniques. In addition to using both proactive and on-demand routing, the algorithms discussed also use both distance vector and link state techniques. In a distance vector algorithm, each device within the network shares its perceived distances to all other devices in the network.

Devices that receive a destination and cost then use these values to update their own table of destinations, in which they also save the origin of the message as the next hop for those routes if the message has the best cost so far. This cost may be a hop count, or may also be a measure of time required to send the packet. This updated table is then used to send out future messages to its neighbors. In this manner, eventually all the devices within the network learn the route to any other destination. In link-state routing algorithms, each device sends messages throughout the entire network that contains a list of all immediate neighbors of the author. These messages are propagated the entire network, typically through flooding. Then, when a route is needed, any given device in the network will have a map of the entire network, allowing it to select the quickest route. These algorithms are also all implemented in the NS2 simulation framework, which allows for the comparison to be performed in a reliable, deterministic manner.

3.7 First Prior Study

There have been previous comparisons on some of the aforementioned algorithms; however, they primarily are modeled after a different use scenario. In the simulation performed by Gupta and Kumar, each device in the network was told to use the random waypoint mobility model [8]. In this model, the devices periodically select a random destination and begin to move in that direction. This causes the entire network to experience changes in the topography, rather than the localized changes which would occur when only one of the endpoints of a connection is in motion. The experiment performed by Gupta and Kumar also only compares DSR, AODV, and the DSDV algorithm. These are all on-demand distance vector algorithms, which still leaves the potential for a proactive algorithm to have better performance in the vehicle control scenario. Additionally, the movement patterns of the devices within their experiment exhibits a much higher degree of mobility that would be found in a scenario with immobile base stations. Their experimental conditions would favor the algorithms that react quickly to change, while reducing the viability of algorithms that have established routes that are not expected to change.

3.8 Second Prior Study

A similar experiment was also performed by Satyam Kumar Sainy, Ravi Rai Chaudhary, and Ajay Kumar. In addition to AODV and DSR, they also considered LAR1, a location based routing algorithm [9]. This experiment also used the random waypoint model for device movement. They found that overall, LAR1 was the most reliable, followed by AODV, and with DSR having the lowest delivery ratio. However, AODV scaled better than LAR1, and much better than DSR as the number of devices increased [9]. While helpful for understanding the algorithms within the context of a fully mobile group of devices, the changes to the network are significantly different than if one device was in motion. Essentially, their network would see changes to the network links uniformly through the entire network, whereas if only one device was moving, the changes would be local to that single device.

3.9 Third Prior Study

An additional study was performed by A.A.A. Radwan, T.M. Mahmoud, and E.H. Houssein. This study also compares the AODV, DSR, and LAR algorithms. The simulation area is set to be much larger, however, the range of the devices is also extended accordingly. Two areas were used, with the sizes 1.5 km by 1 km, and 2km by 1.5 km. Within these areas, the range of the wireless devices was set to 376 meters with a capacity of 2 Mbit/sec. This simulation uses a set number of communication flows, with randomly selected endpoints. Additionally, this simulation follows the random waypoint model of mobility. The results obtained do show noticeable differences in the performance of the algorithms. They found that with a 100-device arrangement in the smaller of the two areas, LAR However, the differences in both the maximum bandwidth and mobility make a direct comparison to a static arrangement difficult. The traffic flow differences are also significant. In this study, 512 bit packets were sent at a rate of 4 per second, which is nowhere near the required throughput for video or audio streams, which can reach several megabits per second [11]. Furthermore, a small number of packets per second allows for queues to handle periods of unavailable routing. In scenarios where a large number of packets is generated, such as a video feed, the amount of storage to hold backed up packets would quickly exceed what is available [12].

Chapter 4

Methodology

4.1 Introduction

This comparison of the different algorithms is performed using ns-2, a network simulation tool. This tool was chosen due to the fact that it is able to simulate moving devices within an ad hoc network, as well as being deterministic in operation if the same seed number is used for the underlying random number generator. In comparison, comparing the algorithms in hardware adds a much higher hardware cost, and adds significantly many more points of failure, such as changing background interference, obstacles, and hardware differences. DSR, AODV, and DSDV are implemented as protocols within the ns-2 simulator by default, and OLSR can be included through use of a patch [10].

4.2 Limitations

The ns-2 simulation tool is limited in some areas, which influenced the design of these experiments. Notably, ns-2 is known to encounter memory issues when working with very large simulations. For this reason, the number of nodes, and therefore the total area of the experiment, was limited in order to ensure reliability in the simulations. Additionally, as the underlying behavior of the simulation is determined by a random number generator, the simulations were run 20 times in order to achieve a more reliable result. Ns-2 requires a rapidly increasing amount of time to complete simulations as the network size grows, so in order to evaluate the performance in larger networks, only a few simulations were run for scenarios dealing with changes to the network size.

4.3 Simulation Parameters

The geographic area simulated in this experiment is a 600-meter by 600-meter square, with 36 devices evenly placed 100 meters apart, in a grid layout with space around the edge of the grid. In this configuration, most nodes are adjacent to 4 other nodes, thus there is a network density of 4 for all devices in the center of the grid, a density of 3 for devices along the outer edges, and a density of 2 for the four corner devices. Ns-2 allows the data rate and maximum transmission range to be set, so a maximum data rate of 11 Mbit/sec and a maximum range of 100 meters were used in order to simulate 802.11 ad hoc mode. As the 802.11 specifications only requires 11 Mbit/sec for wireless ad hoc networks, this is the highest reasonable expectation for the data rate in a real-world application. This range is an absolute limit, as devices even just past this range will never receive packets. The signal frequency, transmission power, the receiver threshold and carrier sense threshold are obtained through a utility included with the ns-2 simulator that computes the required values in order to achieve a given range. In addition to the simulation which occurs in a

Simulation Area	600m x 600m, 900m x 900m, 1200m x 1200m
Device Count	37, 82, 145
Network Density	2-4
Maximum Data Rate	11 Mbit/sec
Maximum Transmission Range	100m
Control Signal Data Rate	1 Kbit/sec
Control Signal Packet Size	1000 bits
Video Signal Data Rate	1.5 Mbit/sec, 3.0 Mbit/sec, 6.0 Mbit/sec
Video Signal Packet Size	1316 bits, 2632 bits, 5264 bits
CSThresh_	1.42681e-8
RXThresh_	1.42681e-8
Pt_	0.28184
freq_	914 MHz

Table 1: Synopsis of simulation parameters

600 meter by 600 meter area, two other simulations were run in both a 900 meter by 900 meter area and a 1200 meter by 1200 meter area. In these larger simulation areas, OLSR's Hello and Topology Control packet rates were each slowed from one per second to one every 5 seconds. Table 1 provides an overall synopsis of the parameters used.

4.4 Route Overview

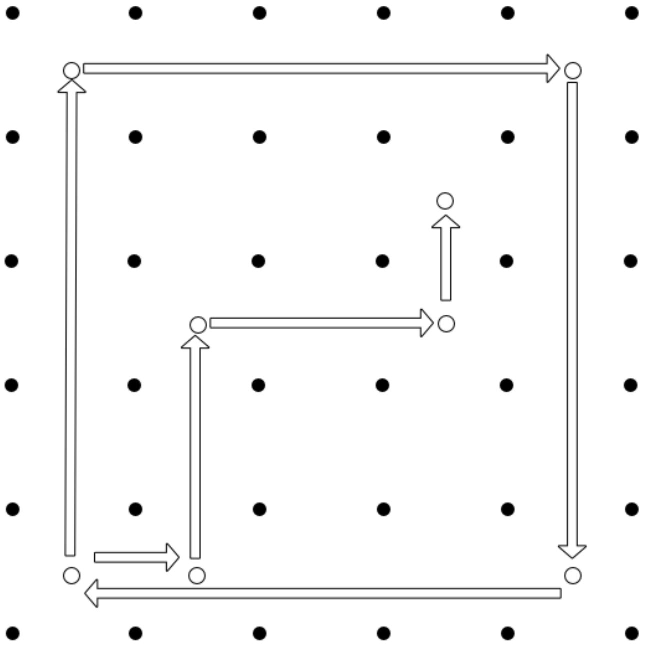

Figure 2: Overview of the simulated route

During the execution of the simulation, a node at one corner of the grid is designated the control station for the vehicle, and an additional node is placed into the grid to act as the autonomous vehicle. The corner was chosen as it provides a greater average distance to the vehicle than other nodes, magnifying the need for optimal routing. The vehicle node then follows a predefined route through the grid with stops to simulate actions, as shown in Figure 2. This route includes both long and short movements that move the vehicle through the entirety of the simulated area. First, the vehicle travels along the perimeter of the area with 10 second stops at each location. After it reaches its starting position, there is a 10 second delay before it follows a zigzagging path through the center of the grid with no delays. Upon reaching the end of the route, the simulated vehicle pauses for one minute, after which the simulation ends. All motions are performed at 5 meters per second. In order to observe the effects of larger networks, this 6 by 6 grid of nodes was extended into a 9 by 9 grid as well as a 12 by 12 grid. The route for these expanded networks follows the same general layout, with the exception of the final set of motions being extended to scale to the larger area. In these expanded simulations, the Hello and Topology Control intervals for OLSR were also lengthened from 1 second to 5 to ensure a reasonable execution time for the simulation.

4.5 Traffic Flows

One traffic flow is set to travel from the control station to the vehicle using TCP, with a data rate of 1 Kbit/sec and a packet size of 1000 bits. TCP is used for this connection because future commands to the vehicle will likely depend of the execution of previous steps. Thus, the arrival of the packets must be guaranteed. This traffic flow is small, as autonomous vehicles and robots are being designed with significantly limited communication in mind [14]. Additionally, a second traffic flow is set to travel from the vehicle node to the control node, in order to simulate audio and video traffic from the vehicle. This second traffic flow has a constant data rate of 1500 Kbit/sec and a packet size of 1316 bits, simulating the MJPEG streaming video format packet size [11] [12]. The MJPEG encoding has a higher data rate when compared to other video encodings; however, each video frame is sent separately and does not depend on the previous packets being received. Because of this, the connection is able to use UDP in order to reduce the latency in transferring packets between hops as the packets do not require guaranteed delivery. This avoids the ACKs that would be required when using TCP, which reduces the burden on the ad hoc network. If ordering is an issue in an implementation, sending a sequence number along with this packet would allow the receiver to playback the images in order.

In addition to this baseline traffic flow, the simulation was repeated with changes to both the UDP data rate as well as packet size. Each of these was adjusted separately and tested at double and four times the original value for each parameter.

4.6 Results Files

Ns-2 creates several files during its execution. The first file is the .tr file that contains a trace of the execution of the simulation. In this trace, it is possible to see the time at which a given packet is created by the agent, which is comparable to an application layer program. Each transmission between nodes is logged, and a final entry is shown when the destination agent receives the packet. By calculating the difference between these two times, the latency of a packet can be determined. Additionally, if the packet is not received by the destination, the packet has been lost along the way. By counting these lost packets and comparing this number against the overall number of packets sent, the packet delivery ratio can be found. The second optional file is the .nam file, which can be opened with nam, which is short for Network ANimator. This can be used to visualize the execution of the simulation, and to diagnose issues that may have occurred.

4.7 Packet Delivery Ratio Metric

There are several metrics that must be examined in order to understand the benefits and shortcomings of each of the different algorithms. Firstly, the percentage of packets that arrive at their destinations, the packet delivery ratio, will be used to understand how reliable the algorithm is. If this number is low, then it can be understood that the algorithm is likely unable to route packets quickly enough through the network. In ns-2, this is reflected by packets being logged in the .tr file as dropped for the reason “IFQ”. This log entry signifies that the wireless interface queue was full when the given packet was received, and that any packets received by this device will be dropped until sufficient room exists in the queue. For this simulation, since some packets depend on their predecessors being delivered, the newest packets will be dropped rather than the oldest.

4.8 Packet Latency Metric

Another metric under consideration is the delay which occurs between the time a given packet is sent and the time at which it is received. This metric can be used to detect that the algorithm often must wait for a route, or when compared to the other algorithms, that the route which is selected is longer than other routes. If this metric has a value that is too high, it is likely that this algorithm is not quick enough for real-time manual control of vehicles, and that it may be suitable only for vehicles and robots that do not require significant amounts of oversight.

Chapter 5

Results

5.1 Introduction

The resulting files from the simulations were parsed using a Python script to determine the time of origin and the time at which each packet reached the destination, if applicable. From these packet times and latencies, the latency as a function of distance from the destination can be obtained as well, as the simulated vehicle follows a fixed route with known times.

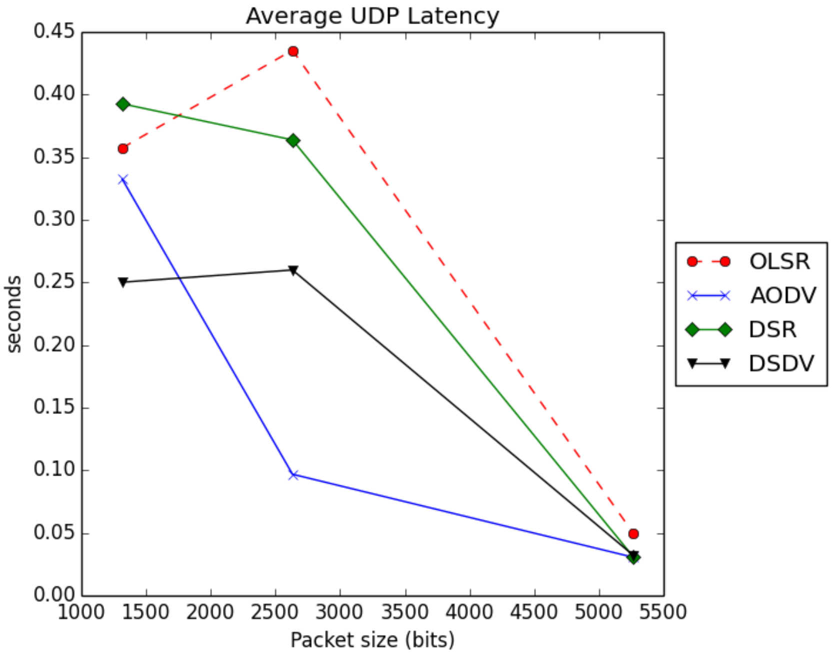

Figure 3: Average UDP latency with respect to packet size

5.2 Video Feed Latency Results

In each simulation, all four algorithms attempted to send the same number of UDP packets. Although they queued the same number of packets, the latency and delivery ratio for each of these algorithms varied noticeably. Additionally, the algorithms had different patterns of latency when comparing the delay to the distance from the packet destination.

When comparing latencies in the base scenario, DSR performed the worst, with an average delay of 0.3926 seconds. OLSR performed slightly better, at 0.3574 seconds on average. AODV and DSDV performed the best overall in terms of latency, at 0.3324 and 0.2502 seconds respectively. As the packet size was increased, this additionally reduced the number of packets being sent. As seen in Figure 3, this reduced the overall latency as fewer routing interruptions occurred due to the interface queue filling up. When sending 5264 bit packets, the latencies for AODV, DSR, and DSDV were all approximately 0.030 seconds. OLSR, however, routed packets

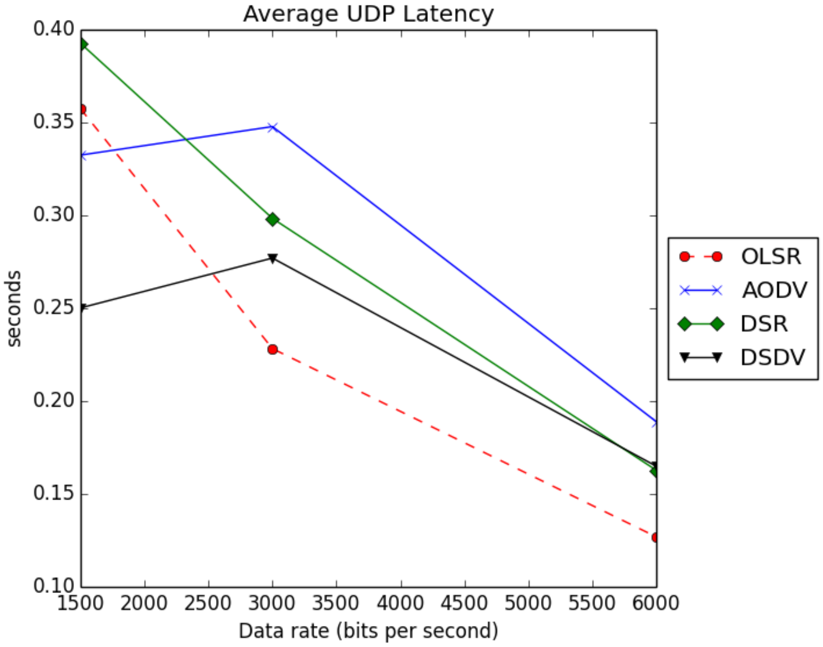

Figure 4: Average UDP latency with respect to data rate

with an average latency of 0.050 seconds.

Additionally, the latencies were observed while changing the rate at which new UDP packets were generated. As seen in Figure 4, when sending four times the number of packets, the apparent average latency dropped significantly. OLSR was able to route the packets the most quickly of the four algorithms when the rate of packets was increased, with a latency of 0.127 seconds. DSR and DSDV both routed packets at just over 0.160 seconds. AODV had the slowest routing at this data rate, with an average latency of 0.189 seconds. The significant decrease in latency overall is related to a much lower delivery ratio, and will be discussed further in the following section.

Although informative of the overall speed of the algorithm, the averages alone do not provide sufficient insight into the effectiveness of the algorithm. As the simulated vehicle ranged from 70 to over 600 meters away from the destination device, the simulation included windows of time at both close and far distances. In the following figures, the packet latency is displayed with respect to the simulation time in order to illustrate the range of latencies that occur during the simulation. The following figures reflect the results of a single experiment performed with a video feed packet size of 1316 bits, and a simulated bit rate of 1500 Kbit/sec.

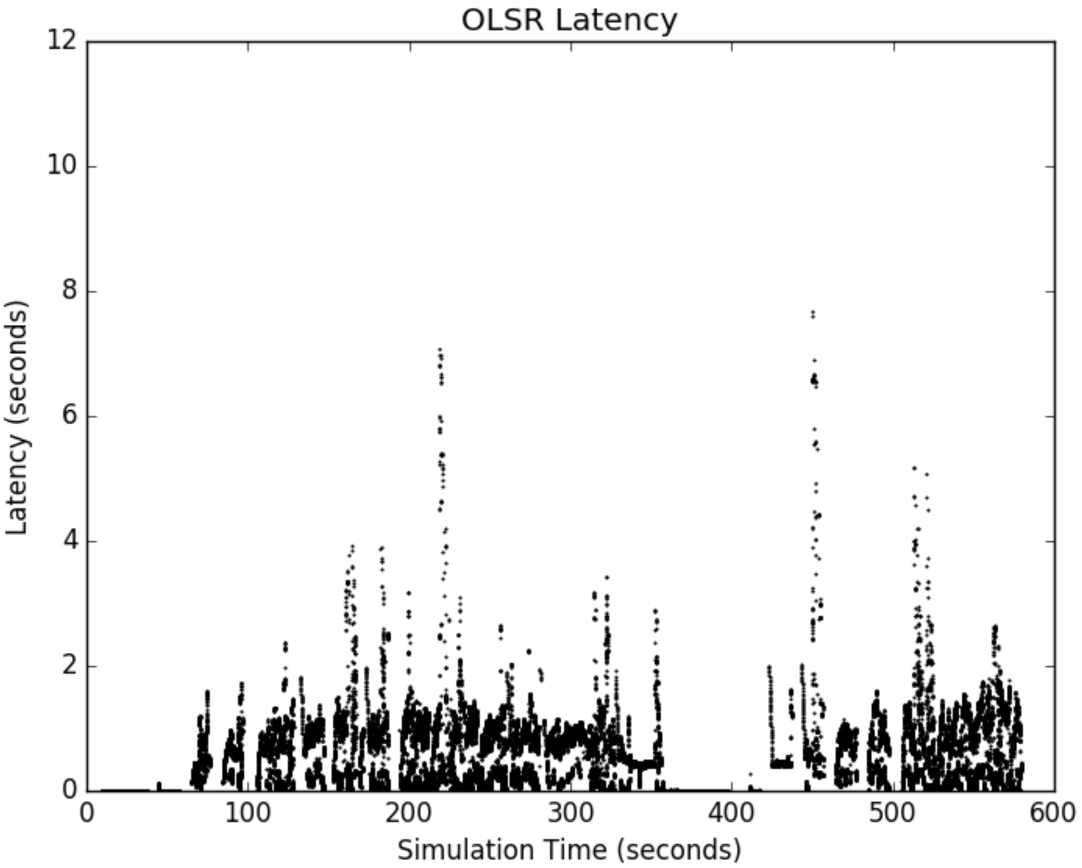

Figure 5: OLSR latency with respect to simulation time

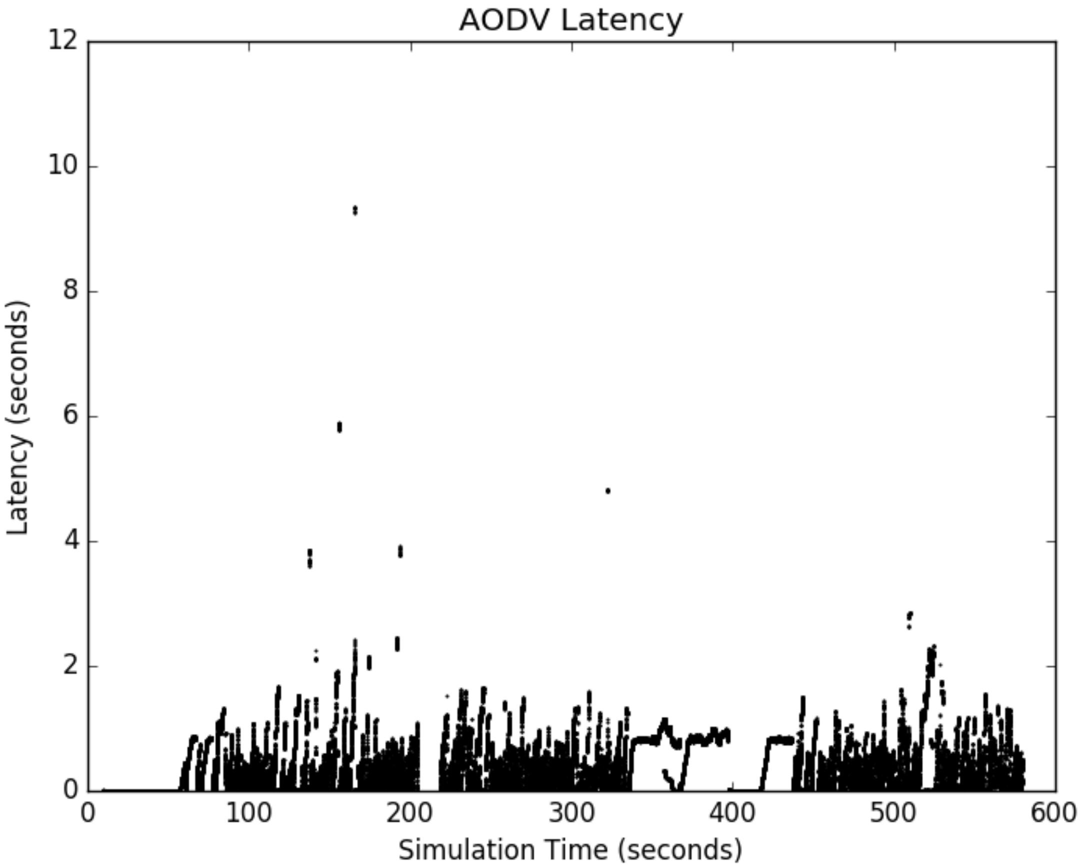

Figure 6: AODV latency with respect to simulation time

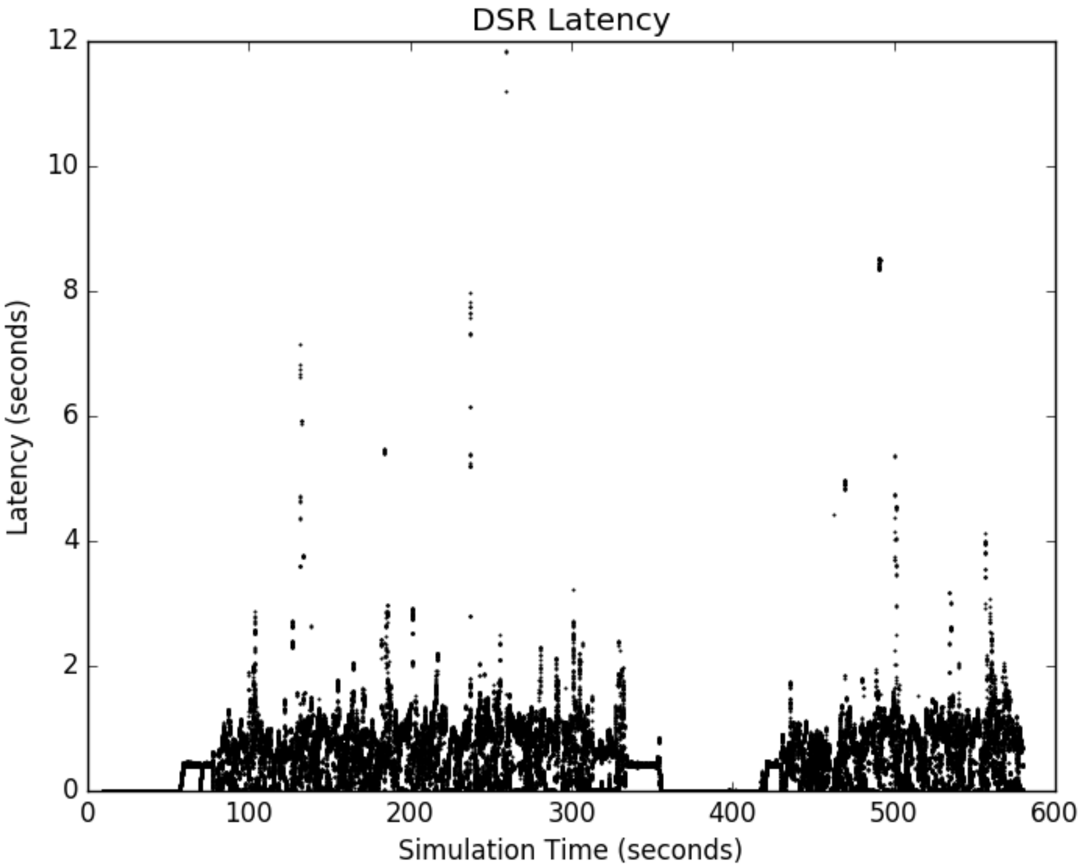

Figure 7: DSR latency with respect to simulation time

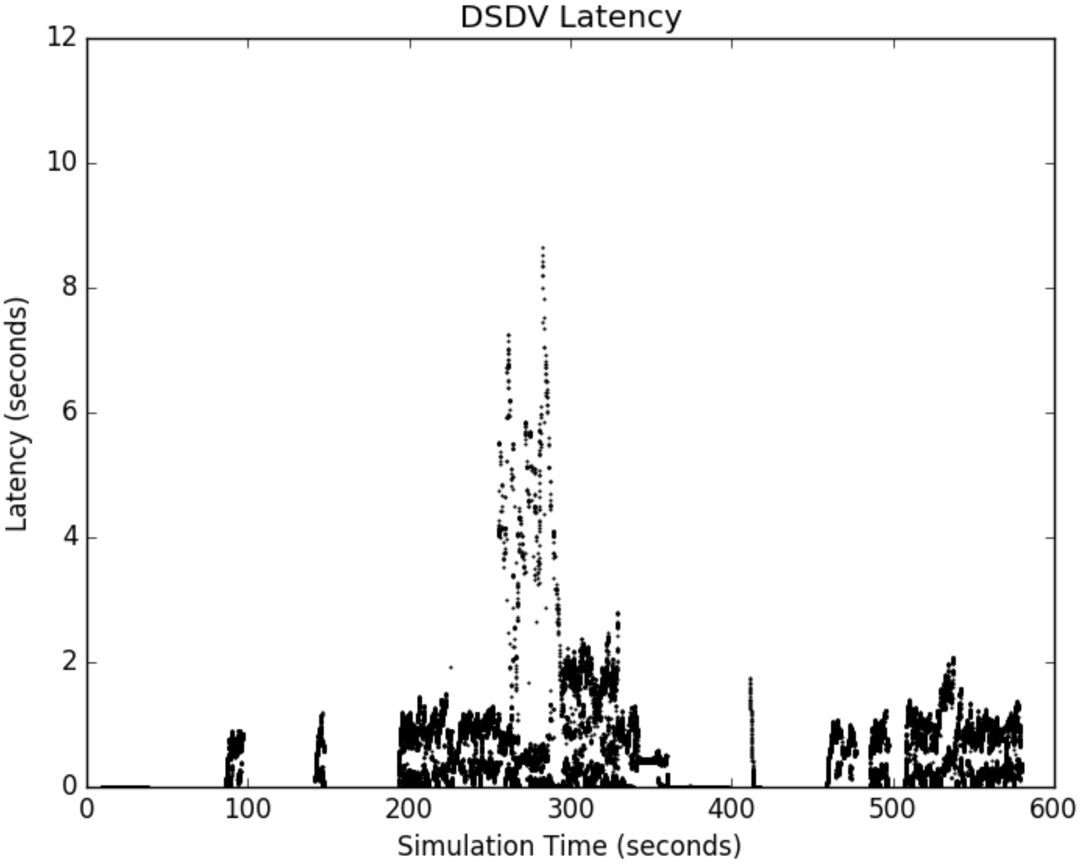

Figure 8: DSDV latency with respect to simulation time

As seen in the figures 4 through 7, routing changes lead to periods of significantly higher latencies, which appear in the figures as densely populated vertical patterns. These areas present significant problems for video streams, as receiving packets up to 8 seconds later would result in the information being essentially useless for any sort of real-time response. In Figure 6, it can be noted that although in the average case AODV is slower than DSDV, the periods of interrupted communication occur less frequently and are of a significantly smaller impact. In the case of DSDV, it can be seen that it frequently has windows of time where all sent packets are dropped. Additionally, when compared to other algorithms, AODV shows a much more consistent pattern of latencies, with much less deviation from the average overall. OLSR tended to frequently encounter routing problems that required significant amounts of time to resolve.

In these figures, the simulation route can be seen through the patterns in the latencies. At times t=0 to t=30, the simulated vehicle is immobile, and is within 1 hop of the destination. Additionally, at t=390 to t=400, the vehicle is located in the same location and is immobile for that time window. Prior to the second pause at the initial location, the vehicle is moving directly towards the packet destination. This is the cause of the greatly reduced latencies, as the device is moving along roughly the same path as the packets themselves. Between these two pauses, the vehicle gradually moves away, and at the exact center point of this time range, begins to move towards the packet destination. Especially in the case of the OLSR and DSR, this can be seen in the slightly curved shape of the latencies.

In addition to examining the packet latencies with respect to the simulation time, it is also worth considering the average latency for each distance from the destination. In the following figures, latencies were averaged together according to the distance from the control station.

These concentric slices have a width of 1 meter, and all packets within this ring were averaged together to gain an understanding of how distance affects the latencies for each algorithm.

In the figures relating distance to latency, it can be seen that there is an influence of distance on average latency. This influence does not directly correspond to the distances; doubling the distance does not double the latency. As the latency increases, the variability of the average latency increases. In Figure 7, which illustrates the latencies for DSR, the upper and lower bounds for the latency spread further as the distance increases. This algorithm had the fewest outliers, however, this trend is still visible in the other three algorithms.

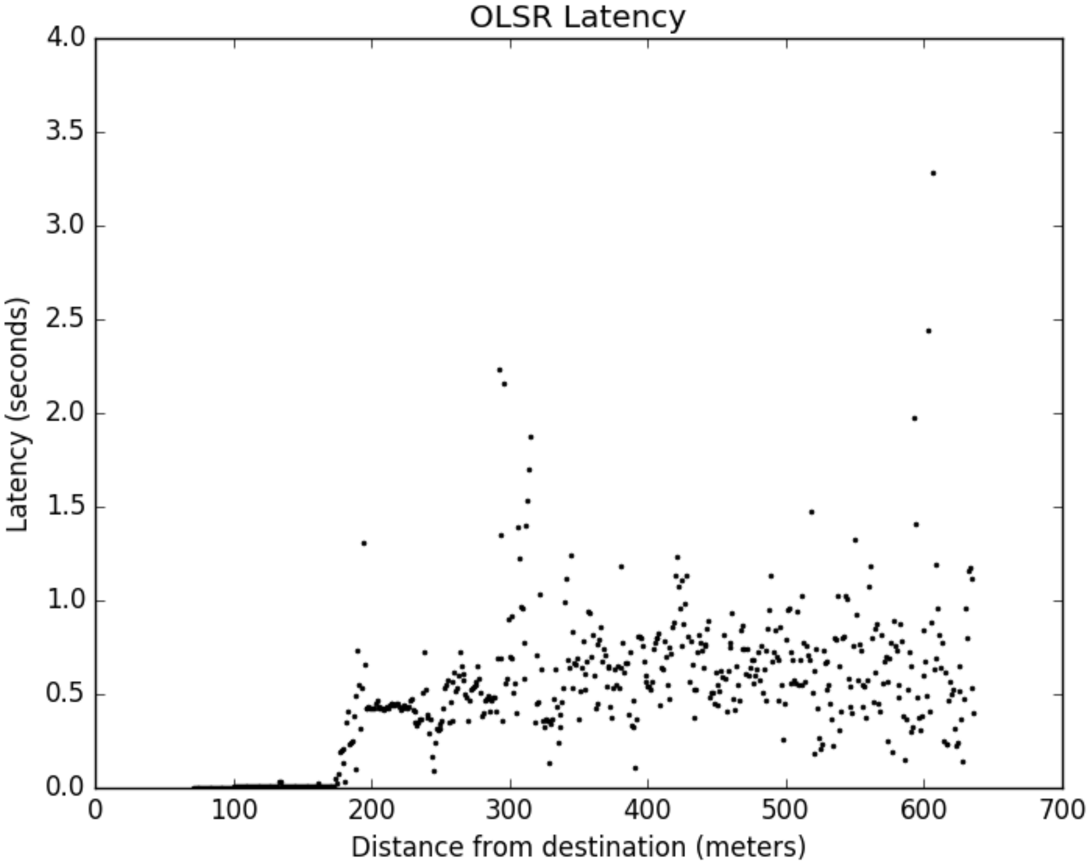

Figure 9: OLSR latency with respect to distance from packet destination

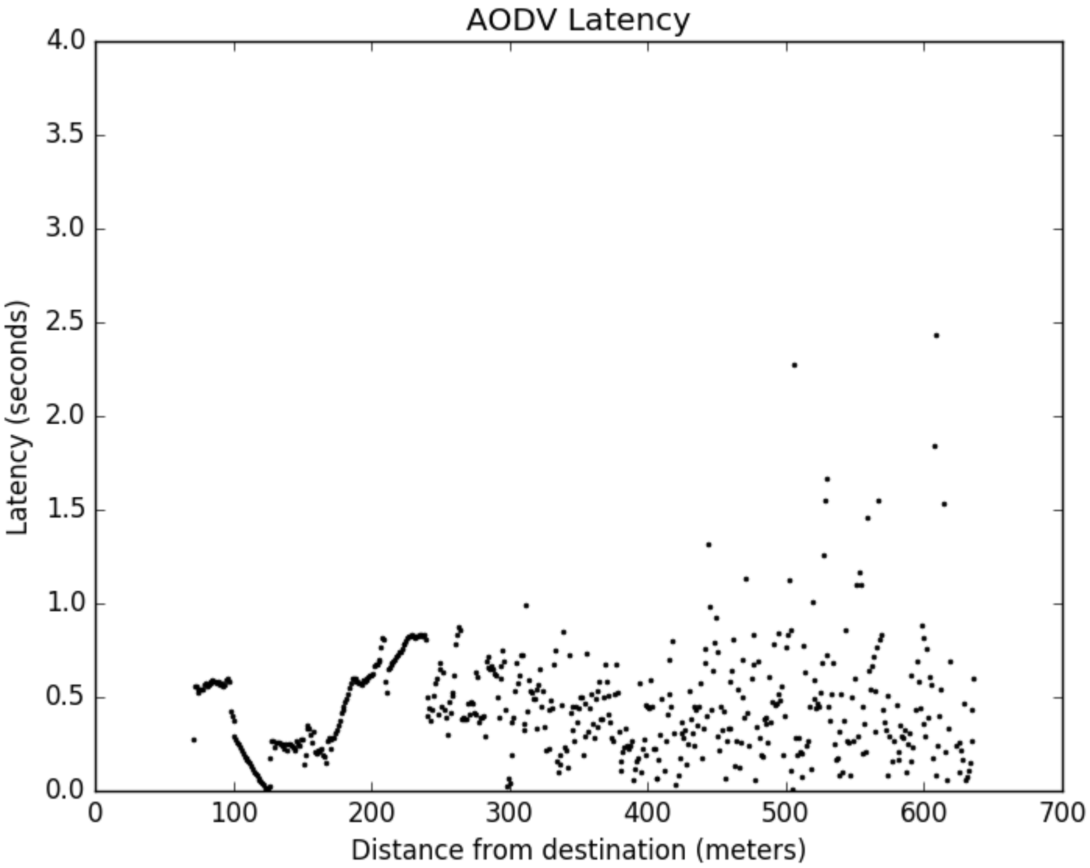

Figure 10: AODV latency with respect to distance from packet destination

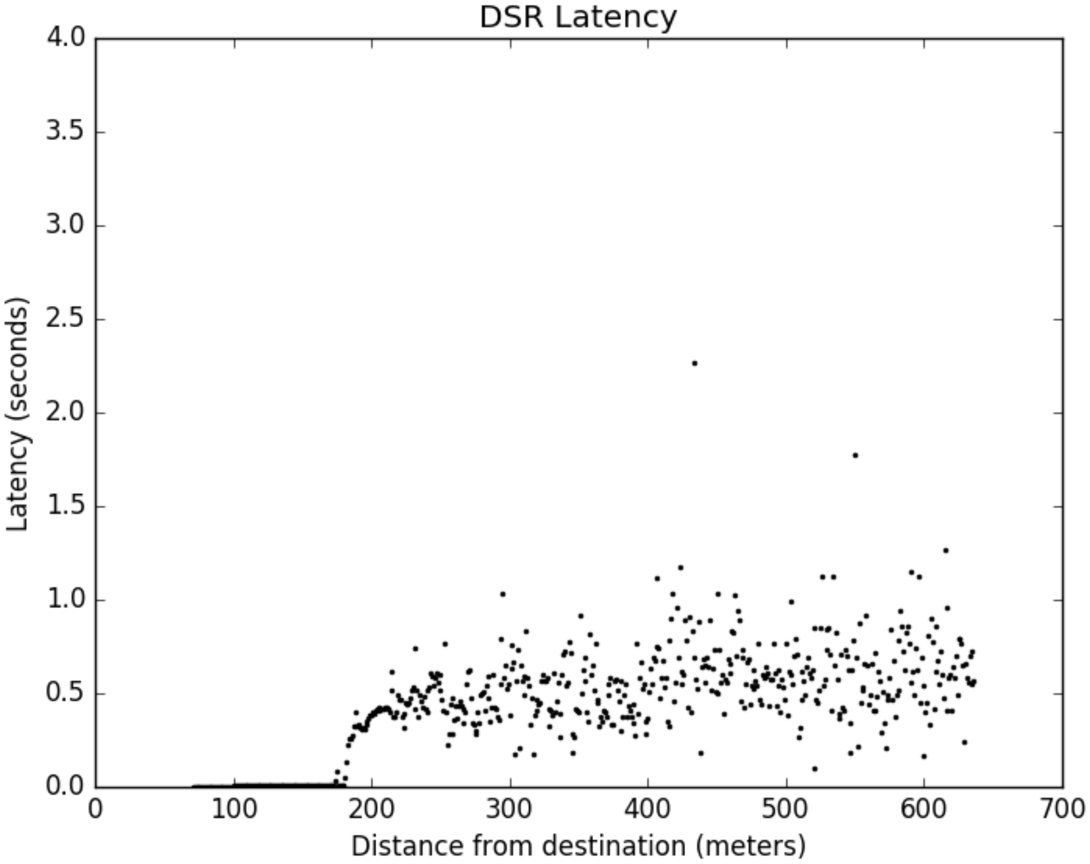

Figure 11: DSR latency with respect to distance from packet destination

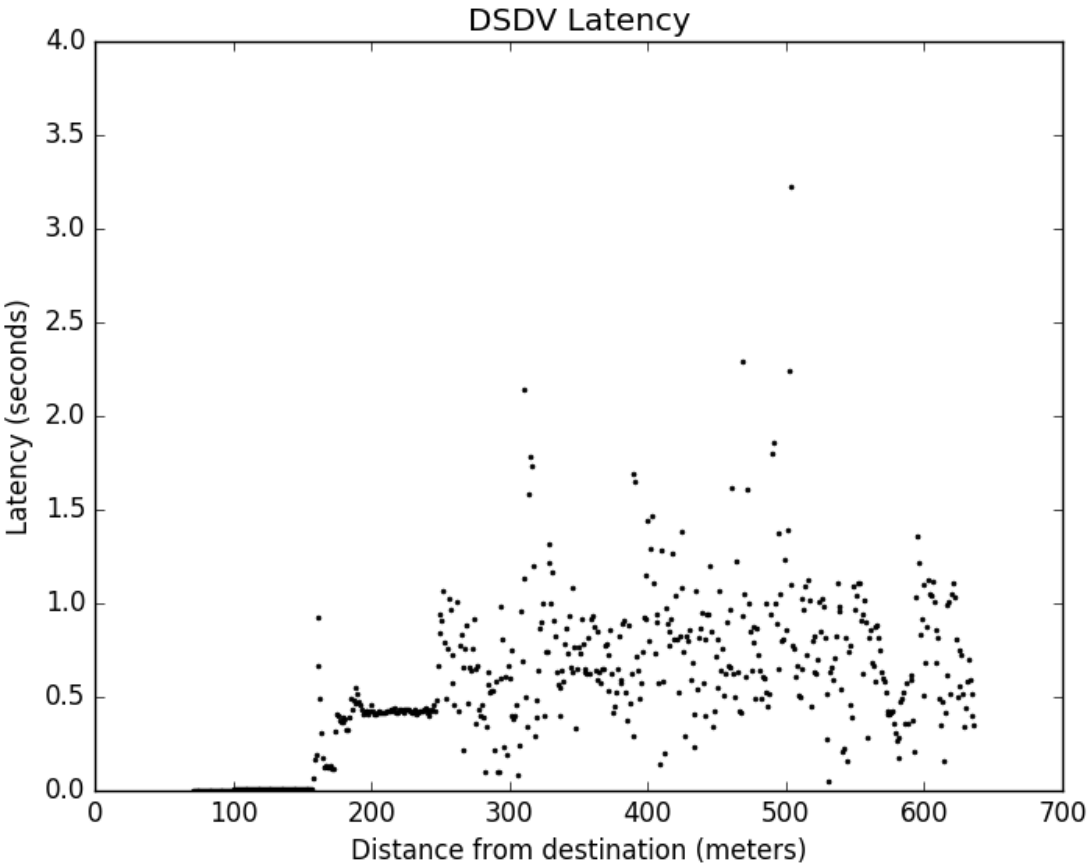

Figure 12: DSDV latency with respect to distance from packet destination

5.3 Video Feed Delivery Results

The packet delivery ratios of these algorithms also displayed a significant difference. In the original simulation, DSDV had the lowest number of delivered packets with only 53.33% of packets reaching their destination. OLSR was not much more reliable, with a 0.7352 delivery ratio. During these experiments, OLSR had a Hello and Topology Control interval of 1 second. Additionally, the UDP data rate for these experiments is 1.5 Mbit/sec. AODV and DSR provided the most reliable delivery, with ratios of 0.8103 and 0.8987 respectively. However, when sending larger packets at a lower rate, the delivery ratio increased significantly. In the case of DSR and AODV, sending packets that were 4 times the original size at a slower rate caused their delivery ratios to exceed 0.99. OLSR saw a slight increase, from 0.7352 to 0.8150. DSDV's delivery ratio was also increased from 0.5333 to 0.6275.

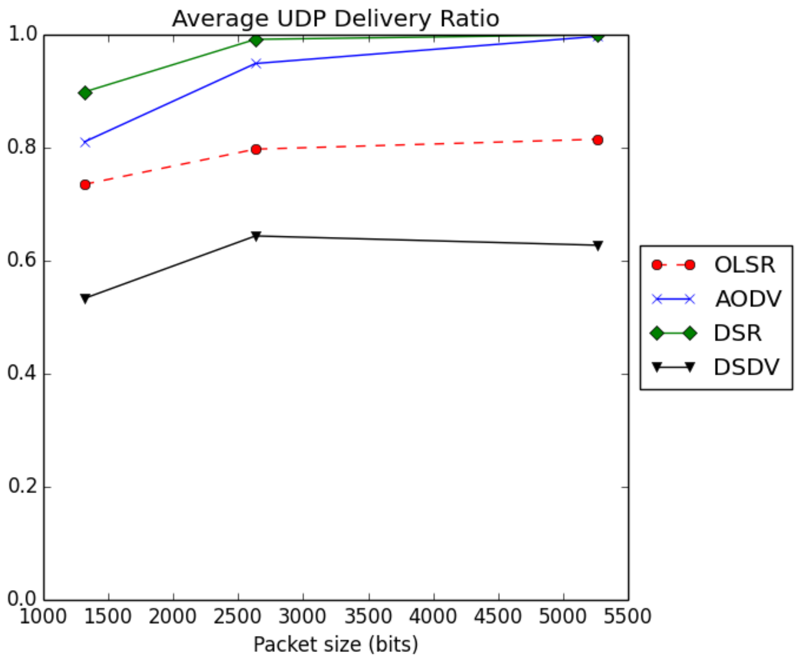

Figure 13: Average UDP delivery ratio with respect to packet size

The delivery ratio for all four algorithms was also dramatically affected by an increase in the number of packets being sent. As seen in Figure 14, as more packets are queued to be sent per second, the less likely they are to reach their destination. These experiments also used a Hello and Topology Control interval of 1 second for OLSR. This set of experiments used the baseline packet size of 1316 bits per packet. All four algorithms saw diminishing returns in scaling up their packet sending rate; AODV performed the best overall, but only had a delivery ratio of 0.3640. The other algorithms did not perform well in this situation either; OLSR had a delivery ratio of 0.3150, DSR had a ratio of 0.3355, and DSDV had a ratio of 0.2315.

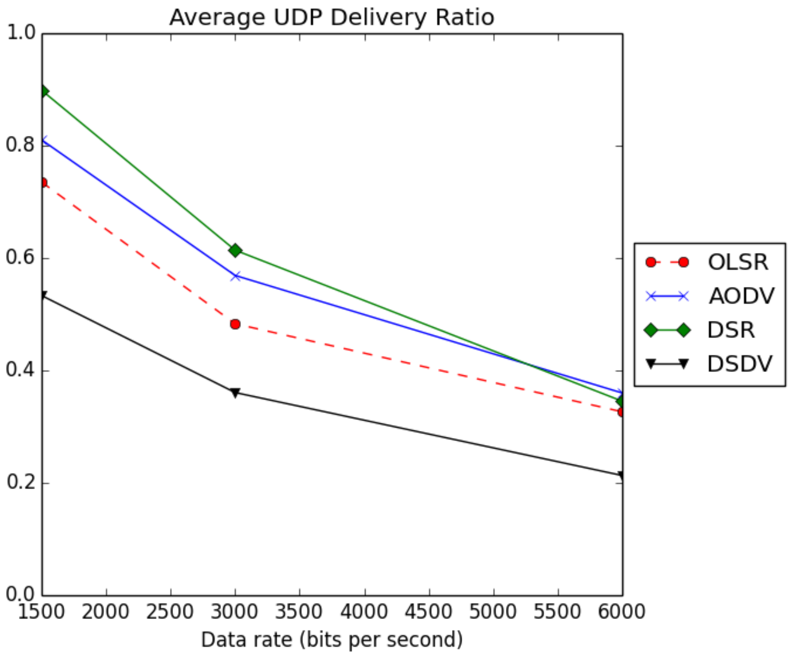

Figure 14: Average UDP delivery ratio with respect to data rate

5.4 Control Signal Latency Results

These latencies may be too high for traditional vehicle control methods; however, this is still sufficiently quick for autonomous vehicle control. For comparison, a 2 hop communication link to a geosynchronous satellite takes approximately 0.54 seconds, which would be the required number of hops to first reach the satellite then return to the vehicle [23]. In the base simulation, OLSR had the lowest latency overall, with 0.1835 seconds on average. Next, AODV performed reasonably well with a latency of 0.1884 seconds on average, followed by DSDV with a latency of 0.5875 seconds. Overall, DSR performed the worst with an average latency of 0.5946 seconds. As seen in figure 15, this ranking of latencies remained mostly consistent when varying the data rate; However, the small quantity of packets exchanged increases the difficulty

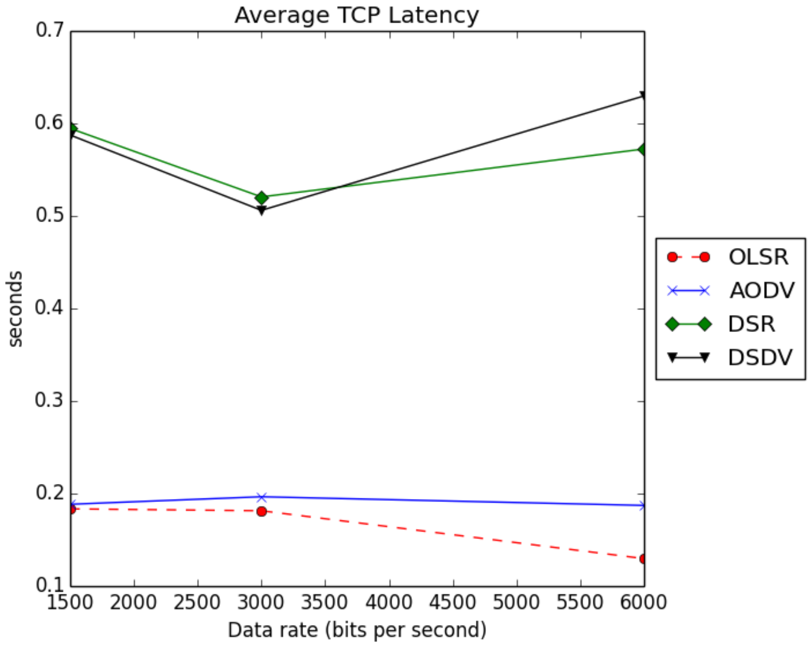

Figure 15: Average TCP latency with respect to data rate

of making strong comparisons between algorithms on the basis of their TCP performance. Similar trends were observed when evaluating the effects of packet size on the latency. Latency remained relatively stable for OLSR, AODV, and DSR, with additional fluctuations for DSDV which is likely due to the smaller sample size, as shown in Figure 16.

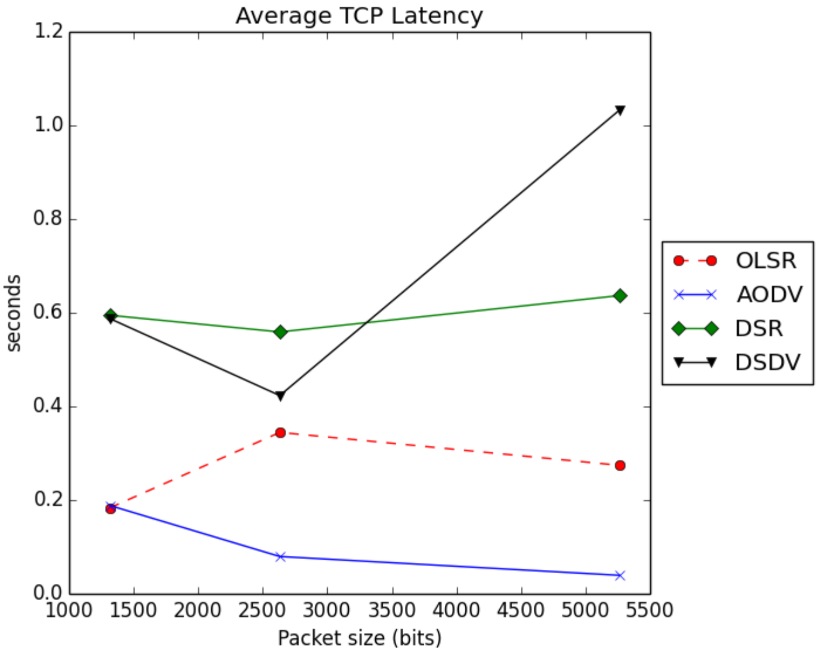

Figure 16: Average TCP latency with respect to packet size

5.5 Control Signal Delivery Results

As TCP was used for this traffic flow, every packet sent reached the destination. However, as a congestion window is part of this protocol, not every algorithm routed the same number of packets. Although 570 packets should be sent in the base simulation, approximately 70 were sent by each of the algorithms, signaling that the congestion window mechanism in TCP caused a reduction in the number of packets sent. As ns-2 uses an interface queue for the wireless transmissions, dropped packets are due to the queue being full at the time of the packet creation. When a route is not available, this queue backs up quickly in the devices between the simulated vehicle and control station, as they are required to send packets in both directions. Thus, the full queues increase the number of dropped packets, increasing the probability of a TCP packet being dropped, which then reduces the congestion window.

5.6 Scaled Network Results

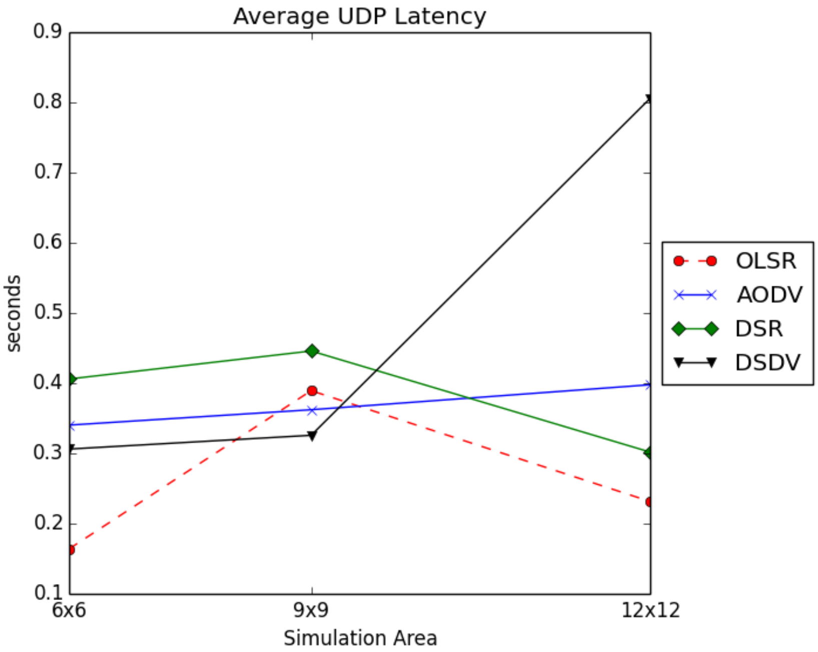

Figure 17: Average UDP latency with respect to simulation area

In this series of three simulations, the network size was increased from 36 static devices to both 81 and 144 devices. In this set of experiments, the Hello and Topology Control message interval was set to 5 seconds rather than 1 second, in order to ensure a reasonable runtime of each simulation. As seen in Figure 17, the average latencies for this simulation remained relatively stable even though the network was several times larger. Although DSDV does have a higher average latency for a network with 144 devices, this is likely due to the randomness present in the ns2 simulator. The delivery ratio shown in Figure 18 does not show any changes at this size, suggesting that a more significant difference may only be observable in networks larger than these. TCP traffic, as seen in Figure 19, also does not see drastic increases in latency when making network changes of this magnitude.

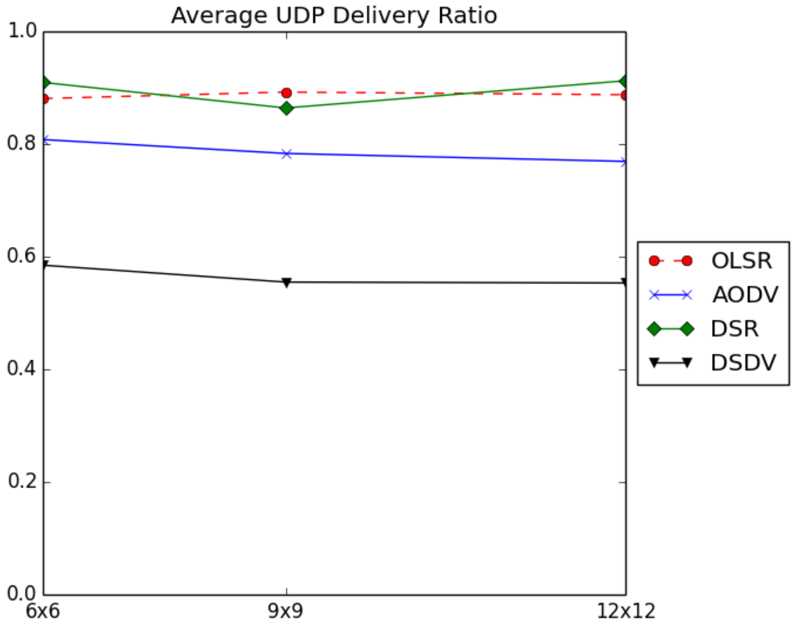

Figure 18: Average UDP delivery ratio with respect to simulation area

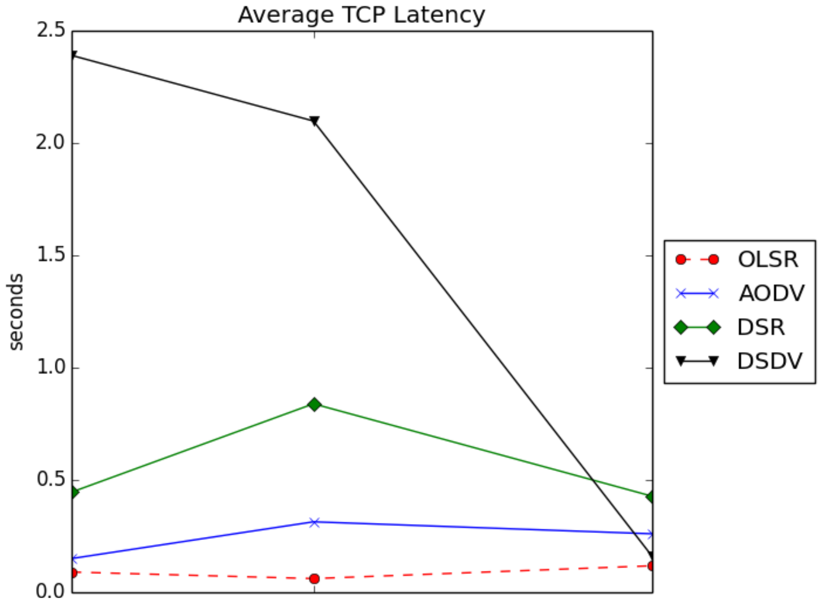

Figure 19: Average TCP latency with respect to simulation area

5.6.1 OLSR Beacon Interval

In addition to varying the packet size and data rate, OLSR was adjusted for running in the larger network to ensure a reasonable run time. OLSR was run with a Hello and Topology Control interval of 5 seconds rather than the default 1 second interval for these experiments. This reduces the speed at which the network can detect changes, but also reduces the chance that in a given time period that the links between devices can fail due to lost packets. This decision was made to reduce the computational load required for simulating the larger networks. In the series of simulations performed with the larger beacon interval, OLSR encountered less latency, likely due to fewer routing interruptions.

Chapter 6

Conclusion

6.1 Latency

These experiments point to AODV and DSDV as having the best latencies of the four algorithms when using consolidating the stream of packets into larger groups. This is especially highlighted in the simulation set with a packet size of 2632 bits. In this scenario, AODV has a latency of 96 milliseconds. However, OLSR performs the worst in this scenario, with a latency of 435 milliseconds on average. DSR is not much faster, with a delay of 363 milliseconds. DSDV performs reasonably quickly in this scenario, with 259 milliseconds of routing delay. Depending on the application, DSDV may still not perform quickly enough. When sending packets that are 5264 bits, all four algorithms perform almost equally well, each having latencies of at most 50 milliseconds.

However, when sending a much higher quantity of packets, OLSR offers the lowest latency overall. For rates of 3000 or 6000 bits per second, OLSR performs approximately 50 milliseconds faster than the next fastest algorithm. DSR and DSDV perform roughly the same at these speeds, while OLSR has the highest latency per packet.

6.2 Reliability

An equally significant measure of effectiveness in these simulations is how many packets reach their destination Regardless of the data rate or packet size parameters, DSR and AODV consistently routed more packets to their destination that OLSR and DSDV. DSR had a higher delivery ratio than AODV in almost all cases, barring the simulation set with a data rate of 6000 Kbits/sec in which the difference in the delivery ratios was 0.033. In the scenarios with larger packets, the delivery ratios of these algorithms approaches 1, with almost perfect routing with a packet size of 5264 bits. In that simulation set, DSR has a delivery ratio of 0.9998, and AODV has a ratio of 0.9970. OLSR likely sees a drop in its delivery ratio, as it constantly needs to send Hello and Topology Control messages. These use slots within the interface queue, and can either be dropped leading to slow routing or block application packets from being sent.

6.3 Application-Specific Needs

For applications that use larger packets, AODV is an effective algorithm as it is capable of a larger throughput as a combination of its speed and reliability. As the number of packets per second is a constant in all simulations performed, it is directly tied to the effective throughput. As seen in Figure 13 and Figure 14, AODV performs with nearly the same reliability as DSR. In the base simulation, these differences add up to a difference of 132.4 Kbit/second in their throughputs. This is made up for by the significantly lower latency that occurs when using AODV. In the 2632 packet scenario, this is particularly evident, although AODV proved faster in all cases. This scenario showed, on average, a difference of 363.8 milliseconds in their latencies. When dealing with applications that may demand real-time reactions with large packets, this additional reduction of delay is invaluable.

When considering applications with a high quantity of smaller packets, DSR operates noticeably faster than AODV while also maintaining a high throughput. At rates of 3000 Kbit/sec and 6000 Kbit/sec, DSR's latency was 49 milliseconds and 26 milliseconds faster, respectively. At the same time, DSR's delivery ratio was 0.045 higher than AODV in the 3000 Kbit/sec simulations, which provides 134.9 Kbit/sec more throughput than AODV. In the 6000

Kbit/sec simulations, DSR performed on average slightly worse than AODV, with a difference of 0.015 in their delivery ratios. This translates to a throughput difference of 87.66 Kbit/sec. Overall, the significantly quicker latency will be of higher impact in real-time applications than the relatively minor throughput difference.

One major factor in the operation of AODV and DSR is that they do not use periodic broadcasts to share routing information in the same way that OLSR and DSDV do. These broadcasts add additional traffic into the network, and it is possible that sudden spikes in traffic were caused by each node sending OLSR Hello and Topology Control packets as well as the periodic dumps sent by DSDV to pass along changes in routing tables. These extra packets could potentially interfere and be interfered with by the traffic sent to and from the simulated vehicle. This can cause either a failure to route packets as updates are lost, or the loss of the packets themselves due to full interface queues.

Chapter 7

Future Work

7.1 Introduction

The focus of the experiments performed was to determine the optimal routing algorithms based on the constraints that the communication must be fast and reliable. Depending on the real-world usage scenario for this algorithm as well as additional information possessed by the routing devices, controller, or vehicle, there may be different requirements for an ideal algorithm.

7.2 Power Consumption

One metric that may be more relevant in disaster relief applications than in other usages is the amount of power consumed by the different devices. The routing devices have been assumed to have sufficient power at all times; future simulations may benefit from monitoring the power used and simulating outages based on consumption from an initial power capacity.

7.3 Additional Algorithms

Other algorithms exist which depend on additional sensors and a priori information for the routing devices. In the presence of GPS sensors and map data, the LAR1 algorithm may be able to provide quick and effective routing for generally immobile networks [9].

7.4 Random Packet Loss

One limitation of the ns-2 simulator is that as long as two devices are within range of each other and the interface queues have room, packets are guaranteed to arrive at the destination. In environments with large amounts of noise or multipath interference, this is not an assumption that can be made. Additionally, these errors are much more prominent when sending larger packets. Future studies may take into account these random occurrences of packet loss so that realistic reliability can be studied in the route maintenance.

7.5 Scaling Device Count

The efficiency of ad hoc routing protocols further degrades as the network size increases. For both link-state and distance vector algorithms, the amount of information that must be relayed through the network increases quickly as the number of devices in the network grows. Future research may examine the relationship between the device count and performance in a greater capacity, such as significantly larger simulation areas.

Bibliography

[1] L. Matthies et al. “A portable, autonomous, urban reconnaissance robot”. In: *Robotics and Autonomous Systems*, Volume 40, Issues 2–3, 31 August 2002, Pages 163-172

[2] V. Garg et al. “Performance evaluation of wireless ad-hoc network for post-disaster recovery using Linux Live USB nodes”. In: *Proc. of the 11th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Abu Dhabi, United Arab Emirates, 2015

[3] N. Correll et al. “Analysis and Observations From the First Amazon Picking Challenge”. In: *IEEE Transactions on Automation Science and Engineering*, Volume PP, Issue 99, 05 October 2016, Pages 1-17

[4] D. Johnson and D. Maltz “Dynamic Source Routing in Ad Hoc Wireless Networks” In: Imielinski T., Korth H.F. (eds) *Mobile Computing. The Kluwer International Series in Engineering and Computer Science*, vol 353. Pages 153-181, Springer, Boston, MA

[5] C. Perkins and E. Royer. “Ad-hoc on-demand distance vector routing”. In: *Proc. from Second IEEE Workshop on Mobile Computing Systems and Applications*, 25-26 Feb. 1999, Pages 90-100, New Orleans, LA, USA

[6] P. Jacquet et al. “Optimized link state routing protocol for ad hoc networks”. In: *Proc. of IEEE International Multi Topic Conference,* 30 Dec. 2001, Pages 23-27, Lahore, Pakistan

[7] Ramya, Anto. "Destination Sequenced-Distance Vector Algorithm." In: *International Journal on Computer Science and Engineering*, Volume 7, Issue 11, 1 Nov 2015, Page 129

[8] M. Gupta and S. Kumar. "Performance Evaluation of DSR, AODV and DSDV Routing Protocol for Wireless Adhoc Network," *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, 2015, Pages 416-421

[9] S. K. Sainy, R. R. Chaudhary and A. Kumar. "Performance evaluation of routing protocols based on different models in MANET," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2016, Pages 1666-1670.

[10] “UM-OLSR patch” URL: https://cloudns2.wordpress.com/um-olsr-patch/

[11] “Internet connection and recommended encoding settings”. URL: https://support.ustream.tv/hc/en-us/articles/207852117-Internet-connection-and-recommended-encoding-settings

[12] “Fundamentals of Digital Video”. URL: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/pktvideoaag.html

[13] A. Pagano and M. Gyimah. *Contemporary Issues in Supply Chain Management and Logistics.* New York: Business Expert Press, LLC, 2017, Print

[14] A. Khan, “At DARPA challenge, rescue robots show future of disaster relief”. URL: http://www.latimes.com/science/great-reads/la-sci-c1-darpa-robot-challenge-20150608-story.html

[15] Bang, Ankur O., and Prabhakar L. Ramteke. "Manet: history, challenges and applications." *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Volume 2, Issue 9, (2013). Pages 249-251.

[16] Toh, Chai-Keong. "A novel distributed routing protocol to support ad-hoc mobile computing." *Computers and Communications, 1996., Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on*. IEEE, 1996.

[17] Open Garden Homepage. URL: https://www.opengarden.com/

[18] MultipeerConnectivity Documentation. URL: https://developer.apple.com/documentation/multipeerconnectivity

[19] A. Lella, “comScore Reports May 2014 U.S. Smartphone Subscriber Market Share”. URL: http://www.comscore.com/Insights/Market-Rankings/comScore-Reports-May-2014-U.S.-Smartphone-Subscriber-Market-Share

[20] I. Vijaya, P. B. Mishra, A. K. Rath and A. R. Dash, "Influence of Routing Protocols in Performance of Wireless Mobile Adhoc Network*," Proc. of 2011 Second International Conference on Emerging Applications of Information Technology*, Kolkata, 2011, Pages 340-344.

[21] Radwan, A. A. A., T. M. Mahmoud, and E. H. Houssein. "Evaluation comparison of some ad hoc networks routing protocols." *Egyptian Informatics Journal* Volume 12, Issue 2, (2011). Pages 95-106.

[22] *Introduction to Wireless and Mobile Systems.* Boston: Cengage Learning, 2010, Print

[23] G. Berlocher, “Minimizing Latency in Satellite Networks”. URL: http://www.satellitetoday.com/telecom/2009/09/01/minimizing-latency-in-satellite-networks/