The Impact of Cyberattacks on Safe and Efficient Operations of Connected and Autonomous Vehicles

Ian McManus

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Kevin P. Heaslip
Kathleen Hancock
Bryan J. Katz

July 29, 2021
Blacksburg, Virginia

Keywords: Cybersecurity, Autonomous Vehicles, Cavs, Dos, MITM, Transportation Safety, Traffic Operation, Modeling, Resilience, Risk, Impact, RSU, V2X Communication, VANET, ITS, Autonomous Vehicles, Simulation, Veins, Cyberattacks

The Impact of Cyberattacks on Safe and Efficient Operations of Connected and Autonomous Vehicles

Ian McManus

ABSTRACT

The landscape of vehicular transportation is quickly shifting as emerging technologies continue to increase in intelligence and complexity. From the introduction of Intelligent Transportation Systems (ITS) to the quickly developing field of Connected and Autonomous Vehicles (CAVs), the transportation industry is experiencing a shift in focus. A move to more autonomous and intelligent transportation systems brings with it a promise of increased equity, efficiency, and safety. However, one aspect that is overlooked in this shift is cybersecurity.

As intelligent systems and vehicles have been introduced, a large amount of research has been conducted showing vulnerabilities in them. With a new connected transportation system emerging, a multidisciplinary approach will be required to develop a cyber-resilient network. Ensuring protection against cyberattacks and developing a system that can handle their consequences is a key objective moving forward. The first step to developing this system is understanding how different cyberattacks can negatively impact the operations of the transportation system.

This research aimed to quantify the safety and efficiency impacts of an attack on the transportation network. To do so, a simulation was developed using Veins software to model a network of intelligent intersections in an urban environment. Vehicles communicated with Road-Side Units (RSUs) to make intersection reservations – effectively simulating CAV vehicle network. Denial of Service (DoS) and Man in the Middle (MITM) attacks were simulated by dropping and delaying vehicle's intersection reservation requests, respectively. Attacks were modeled with varying degrees of severity by changing the number of infected RSUs in the system and their attack success rates.

Data analysis showed that severe attacks, either from a DoS or MITM attack, can have significant impact on the transportation network's operations. The worst-case scenario for each introduced an over 20% increase in delay per vehicle. The simulation showed also that increasing the number of compromised RSUs directly related to decreased safety and operational efficiency. Successful attacks also produced a high level of variance in their impact. One other key finding was that a single compromised RSU had very limited impact on the transportation network.

These findings highlight the importance of developing security and resilience in a connected vehicle environment. Building a network that can respond to an initial attack and prevent an attack's dissemination through the network is crucial in limiting the negative effects of the attack. If proper resilience planning is not implemented for the next generation of transportation, adversaries could cause great harm to safety and efficiency with relative ease. The next generation of vehicular transportation must be able to withstand cyberattacks to function. Understanding their impact is a key first step for engineers and planners on the long road to ensuring a secure transportation network.

The Impact of Cyberattacks on Safe and Efficient Operations of Connected and Autonomous Vehicles

Ian McManus

GENERAL AUDIENCE ABSTRACT

The landscape of transportation is quickly shifting as transportation technologies continue to increase in intelligence and complexity. The transportation industry is shifting its focus to Connected and Autonomous Vehicles (CAVs). The move to more autonomous and intelligent transportation systems brings with it a promise of increased transportation equity, efficiency, and safety. However, one aspect that is often overlooked in this shift is cybersecurity.

As intelligent systems and vehicles have been introduced, a large amount of research has been conducted showing cyber vulnerabilities in them. With a new connected transportation system emerging, a multidisciplinary approach will be required to prevent and handle attacks. Ensuring protection against cyberattacks is a key objective moving forward. The first step to developing this system is understanding how different cyberattacks can negatively impact the operations of the transportation system.

This research aimed to measure the safety and efficiency impacts of an attack on the transportation network. To do so, a simulation was developed to model an intelligent urban road network. Vehicles made reservations at each intersection they passed – effectively simulating an autonomous vehicle network. Denial of Service (DoS) and Man in the Middle (MITM) attacks were simulated by dropping, and delaying vehicle's intersection reservation requests, respectively. These cyberattacks were modeled with varying degrees of severity to test the different impacts on the transportation network.

Analysis showed that severe attacks can have significant impact on the transportation network's operations. The worst-case scenario for each attack introduced an over 20% increase in delay per vehicle. The simulation showed also that increasing the number of attacked intersections directly related to decreased safety and operational efficiency. Successful attacks also produced a high level of variance in their impact. One other key finding was that a single compromised RSU had very limited impact on the transportation network.

These findings highlight the importance of developing security and resilience in a connected vehicle environment. Building a transportation network that can respond to an initial attack and prevent it from impacting the entire network is crucial in limiting the negative effects of the attack. If proper resilience planning is not implemented for CAVs, hackers could cause great harm to safety and efficiency with relative ease. The next generation of vehicular transportation must be able to withstand cyberattacks to function. Understanding their impact is a key first step for engineers and planners on the long road to ensuring a secure transportation network.

Acknowledgements:

I would like to thank all my family, friends, professors, and classmates who helped in any way along the journey of writing this thesis. This was not an easy endeavor, and I could not have done it without a plethora of support.

Table of Contents

1. Introduction

Background

The future of transportation in the United States looks a lot different than it has in the past. A country that has long depended on privately owned, gas-powered, driver dependent, and communicatively independent vehicles is headed towards a future where vehicles will ultimately be connected and drive themselves. Connected and Autonomous Vehicles (CAVs) are currently one of the main focuses of the transportation research community, with companies seeking to break into the consumer market sooner rather than later. CAVs will be able to communicate with other vehicles through Vehicle-to-Vehicle (V2V) communication and Road-Side Units (RSUs) through Vehicle-to-Infrastructure (V2I) communication. These two sources of communication will allow CAVs to have a continuously updated map of the road landscape around them, alerting them to crashes or congestion that may be occurring downstream [1]. The performance of these vehicles is widely believed to reduce crashes, deaths, congestion, and emissions in the future. The National Highway Traffic Safety Administration (NHTSA) reported that in 2018, 36,560 people were killed in motor vehicle crashes, and 94% of all severe crashes were the result of human error; both statistics are expected to be significantly reduced by the introduction of CAVs [2].

While CAVs are an emerging technology, the concept that most people consider to be an autonomous vehicle has been around for a few decades, with the first “modern” autonomous vehicle being developed in Germany at the tail end of the 20th century [3]. Autonomous vehicle research and competition increased in the United States at the beginning of the 21st century due to the United States’ Defense Advanced Research Project Agency (DARPA) Grand and Urban

Challenge where universities, vehicle manufacturers, and other companies deployed autonomous vehicle technology that is now attempting to be deployed to the consumer market [3].

Companies such as Audi, Mercedes-Benz, and Google all currently have prototypes being rigorously tested [4]. As of September 2020, Google's (Alphabet's) autonomous vehicle program, Waymo, has driven more than 20 million miles on public roads with self-driving-enabled vehicles [5]. In July 2020, Tesla's CEO Elon Musk claimed that the company was "very close" to having basic level 5 automation technology developed [6]. The Society of Automotive Engineers (SAE) has developed Standard J0316, defining the six levels of automation in vehicles, shown in Figure 1 [7].

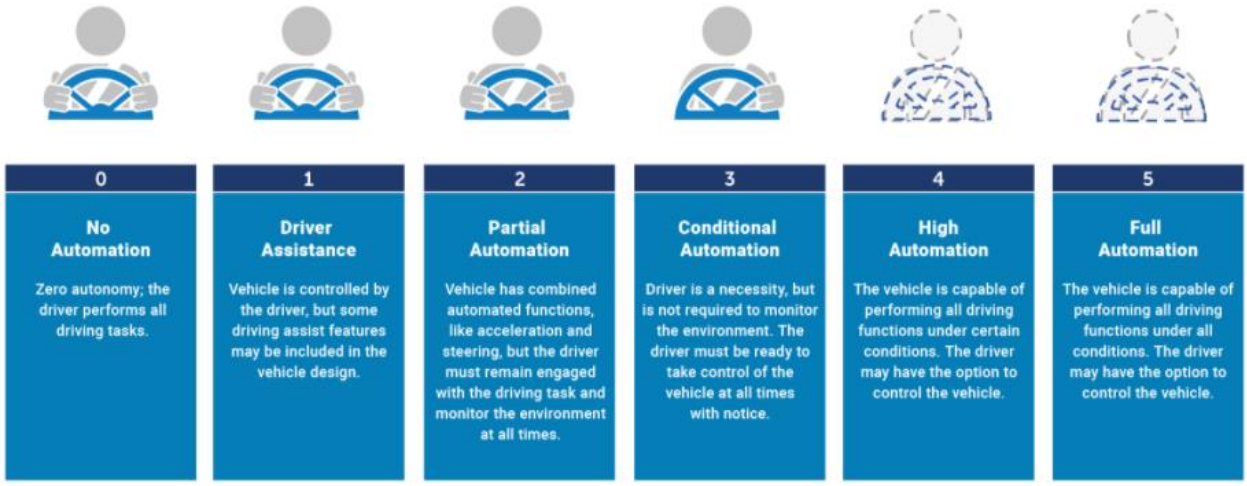


Figure 1: SAE Levels of Automation [2].

Currently, Tesla vehicles possess an autopilot feature that puts the vehicle close to a level three automated vehicle while Waymo claims to be operating at level four automation [5].

Connected and autonomous vehicles are currently being researched, developed, and tested, but they may be a long way from dominating the vehicular market share. Litman [8] claimed that autonomous vehicles would be available at the end of the 2020s, but they will only benefit the affluent portion of the population; most people will not receive the benefits until the 2050s or '60s [8]. Using a model based on the implementation of previous emerging

technologies, Lavasani et al. [4] concluded that if autonomous vehicles are available in 2025, it will take ten years (2035) for 8 million vehicles to be in the market and 35 years (2060) for the market to become fully saturated with these vehicles. While they are not currently commercially available, it might not be long before they begin streaming into the market.

In the future, vehicles will communicate with each other and roadside infrastructure through a network known as a vehicular ad-hoc network (VANET). In a VANET, vehicles are treated as nodes on the system, and roadside infrastructure remains still. Vehicles use On-Board Units (OBUs) to send messages to other vehicles and RSUs, on the infrastructure side, with all communication using wireless connections based on IEEE 802.11p radio frequencies [9]. Communication between vehicles on the road and in-place infrastructure allows for advanced warning messages and safety messages to be sent throughout the network, theoretically reducing crashes and congestion. NHTSA [10] reported that V2V communication, on its own, could prevent 75% of light vehicle crashes.

CAVs are joining the modern age, where most of everything is connected through the internet. With this modernization, CAVs are also vulnerable to hacking and cyberattacks. Anticipating and defending against these attacks is important as there is an obvious endangerment to human life if these attacks are successful. There are many possible attacks against CAVs, but this thesis will focus on Denial of Service (DoS) attacks and Man in the Middle (MITM) attacks. Al-Kahtani [11] described a DoS attack as a malicious node sending dummy messages that can overwhelm the system, jamming any further communications, thus preventing critical and necessary messages from being sent to other nodes on the network. This same study described a MITM attack as a malicious node eavesdropping on messages sent between other vehicles or infrastructure; the node then injects a malicious message into the

network [11]. These two attacks were selected as they are two relatively understood and well-known attacks that can significantly impact the transportation network [12]. Understanding these cyberattacks' impact on the transportation network is important to designing and constructing a more resilient transportation system.

Necessity for Research

This research is important to understand the possible vulnerabilities in the transportation network presented by cyberattacks on CAVs. While this research does not aim to mitigate or prevent these attacks, understanding the attacks and their impact on CAVs is a first step to developing mitigations in the future. Understanding these attacks in a transportation context can prevent loss of life in the future and build resilience to attacks that may have ulterior motives other than the loss of life. An attack on a network of CAVs can have safety and operations consequences on critical transportation routes in the US, resulting in gridlock with the possibility of death.

Research Question(s)

There is one centralized theme that summarizes the goal of this thesis; "How do cyberattacks on CAVs impact traditional transportation engineering measures?" However, this question is vast and requires a more specific focus, resulting in two more specific questions. The first question is: "How does a Denial of Service (Dos) attack impact transportation safety and operations in an urban setting?" The second question is similar but with a different attack: "How does a Man in the Middle (MITM) attack impact traffic safety and operations in an urban setting?" The specific nature of the questions allows for this research to be applied in various realistic scenarios, providing meaningful results and data analysis for engineers and decision-makers.

Future transportation safety and efficiency problems will require expertise not usually associated with civil engineering to solve them. Understanding the future of transportation in terms of cybersecurity and traditional transportation metrics will allow CAVs to be planned holistically, with the end goal of a resilient transportation system. Without this multifaceted approach, the transportation industry could develop a new, computer-reliant system entirely ignorant and exposed to cyberattacks - a system waiting to be taken down by malicious attackers.

Research Problem and General Approach

Very little is known about the impact of cyberattacks on traditional transportation engineering metrics like safety and operations. Generally, research that has been conducted on the topic was from an electrical or computer science lens which neglects the transportation network impact of the attacks. These studies have examined the impacts of attacks on the VANET itself and recommending countermeasures to specific attacks, but not translating it to transportation - or not sufficiently doing so. Studies also tend to outline the possible cyberattacks that can be performed against CAVs and often perform risk analysis on the attacks that categorize the likelihood and general impact of the attack.

This research is aimed to start to bridge the gap between these previously conducted studies and connect the results to transportation engineering. The transportation impacts of different severities of cyberattacks will be modeled and quantified using a simulation. This model aims to model cyberattacks and quantify and analyze their impact on traditional transportation measures of effectiveness (MOEs).

Underlying Objective of Research

The underlying objective of this research is to study, analyze, and present the impacts of cyberattacks on CAVs in terms of safety and operations, two traditional transportation topics.

This research aims to be one of the first to bring cybersecurity to the forefront of transportation planning and analysis. In doing so, it seeks to provide cyber impact analyses for safe and efficient operations for future transportation systems, systems that will encounter problems and attacks not faced by the transportation field currently or in the past. These contributions aim to provide another layer of resilience for the transportation network, which millions of Americans depend on every day to function and survive.

Anticipated Contributions

This thesis seeks to contribute two things: model and analyze two separate attacks on an urban environment and to quantify the impact of these attacks on the transportation network. Understanding the impact of cyberattacks on operations and safety on a roadway is important for transportation engineers and transportation agencies to plan for and develop mitigation strategies. This thesis will not present or recommend mitigation strategies, but it seeks to quantify the problem that needs to be solved. This research can serve as a basis for others who seek to develop mitigation strategies based on the modeling conducted in this study. This research also aims to provide a base for quantitative risk analysis for both of the proposed scenarios. Quantifying risk can also be crucial for governments to decide how much money to invest into cybersecurity resilience and attack mitigations. This research aims to quantify the impact and set an example of analyzing non-civil engineering fields and relating them to pressing civil engineering topics.

Document Organization

The remainder of this thesis will contain six main chapters: Literature Review, Methodology, Results, Data Analysis, Discussion, and Summary, Key Findings, Conclusions, and Future Research. The Literature Review chapter will provide background on cyberattacks on

VANETs, risk and resilience of CAVs in terms of cybersecurity, MITM attacks on VANETs, DoS attacks on VANETs, cybersecurity as it relates to transportation engineering, and the overall safety and operational features of a fully connected fleet of autonomous vehicles. The Methodology chapter will detail the process of the model setup, experiment, and subsequent analysis. The results of the simulation runs will be provided in the Results chapter, and the data analyzed in the Analysis and Discussion chapter. The Analysis and Discussion chapter will follow the Results chapter, interpret the model's findings, and summarize the key takeaways from the modeling. The Summary, Key Findings, Conclusions, and Future Research chapter will wrap up the thesis, restating the purpose and the main findings from the results and analysis and proposing potential future research to build on this thesis.

2. Literature Review

This literature review has two main goals: finding and documenting the research conducted in a broad range of categories relating to CAV cybersecurity and cyberattacks and their impact on transportation safety and operations and identifying areas where research may not be as strong. The literature review will frame the main questions to be answered by research to add to existing literature while connecting ideas that may fill in identified research gaps. By the end of the literature review, this thesis' main themes and concepts will be clarified with the appropriate level of detail.

The review has incorporated a diverse breadth of ideas sourced from many different types of literature. This chapter may be updated in the future as more information is processed.

Literature Categories

Research in CAV cybersecurity has generally focused on the computer science and network aspects. In contrast, the civil engineering research on CAVs has primarily focused on

the potential improvements in transportation network efficiency and safety. Due to the multidisciplinary nature of the topic, literature review categories for the research included topics that span both fields and any connection that has been made between them.

The topics to be covered in this literature review are as followed: Transportation Safety and Operations Improvements from the Implementation of CAVs, Cyberattacks on VANETs, Man the Middle Attacks on the VANET, Denial of Service Attacks on the VANET, Cybersecurity in Transportation Engineering, and finally Risk and Resilience of CAVs. The order that topics are presented is meant to build a working knowledge for those that are not knowledgeable in these topics. It will introduce the benefits of CAVs, followed by the possible risks, connect cybersecurity and transportation, and finally present what is known about the risk and resilience of the future transportation system. Each topic was selected to add a different perspective from existing research, the specific reasoning for each is detailed below:

- Transportation Safety and Operations Improvement from the Implementation of CAVs were selected to show the potential/expected benefits from the implementation of CAVs. This past research serves almost as a control group to the current study, giving a solid baseline of how CAVs should behave in an ideal future real-world scenario.
- Cyberattacks on VANETs will provide a background for cyberattacks that have successfully been conducted through modeling, laboratory testing, and testing in a real-world environment. This section will also describe the numerous types of cyberattacks modeled against the VANETs and the in-vehicle network itself.
- Man in the Middle Attacks on the VANET will be reviewed as it is one of the main points of research for this thesis. This section will provide a more in-depth description of the attack method and its modeled and analyzed past studies.

- Denial of Service Attacks on the VANET will be a very similar section to the previous section. These sections were broken up as each attack is unique, and an entire unit should be dedicated to the specifics of the two attacks. This attack is also a primary point of research for this thesis.
- Cybersecurity in Transportation Engineering will be reviewed to find any relevant research that has been conducted that has connected cybersecurity to transportation measures of effectiveness. As mentioned previously, this is a relatively underrepresented research topic, so it is essential to find what has been examined to add to the bank of knowledge. Studying this topic will also lead to the review and analysis of different modeling and analysis techniques that could be used or adapted for this research.
- Risk and Resilience of Connected and Autonomous Vehicles will be studied to determine what analysis has been done on the topic. One of the main goals of this research is to enable risk analysis based on the impact of attacks to build resilient transportation infrastructure in the future.

Transportation Safety and Operation Improvements from Implementation of CAVs

The impact of the implementation of CAVs has long been discussed as a utopian vision of the future. This new technology claims to solve the safety and efficiency issues long associated with the current car-oriented transportation network. The actual impacts of CAVs have been the topic of heavy research since the turn of the century, resulting in quantifiable predictions and models that show the actual benefits and drawbacks of the technology. Researchers have come to different conclusions in terms of quantifying benefits. Still, most studies agree with Ye and Yamamoto's [13] conclusion that the impacts of CAVs are a function of market penetration and the level of vehicular connectivity. Understanding these two factors

and their associated implementation timelines should give engineers, planners, and officials a better idea of benefits for future infrastructure policy.

In 2008, NHTSA reported that 93% of motor vehicle crashes were primarily caused by human factors [2]. Correlating that percentage to 2019 reported crash data [14], approximately 33,500 fatalities were attributed mainly to human behavior. Additionally, NHTSA [15] found that a full V2V adoption would prevent 439,000-615,000 crashes for light vehicles in the United States. This reduction represents a 10% improvement in total crashes compared to 2018 crash data [16]. In an exemplary implementation, CAVs would be able to eliminate all crashes and fatalities, but research has shown that this is unlikely, at least for the time being. Rahman and Abdel-Aty [17] found that, based on the predicted levels of market infiltration, safety measures benefited the most from two separate lanes. The first, a managed lane (a CAV-only lane), which accommodates connected vehicle platoons, and the second, a mixed-use lane to accommodate non-CAVS for a two-lane freeway environment. Transportation infrastructure designed for and implemented during the early stages of CAV infiltration must accommodate CAV platooning (ideally) and non-CAVs. There will likely be a heterogeneous mix of vehicle types.

The proposed safety improvements are meant to draw attention to the predicted impacts of a transition from traditional vehicles to CAVs. In reality, these improvements are difficult to model and quantify. Since NHTSA [2] reported that over 90% of severe crashes could be attributed to human error, it is logical to conclude that these crashes will decrease as the human driving element is removed. Other researchers have reported safety benefits due to traffic smoothing and efficiency resulting from improved operations and connectivity due to CAV market penetration rates [18]. To this point, a significant number of studies have been conducted

to quantify the improved traffic capacity and travel efficiency of a CAV-oriented system, avoiding making safety predictions.

The two main metrics used to quantify these efficiency improvements to the future transportation network through the implementation of CAVs have been increased road capacity (and associated metrics) and improved fuel economy through traffic smoothing. Chen et al. [19] studied the impact of designating traffic lanes on a multilane road, as either only connected and autonomous vehicles, only regular vehicles, or mixed-use on congestion and capacity. The study found that the impact was directly related to market penetration. The highest capacity and throughput for a two-lane (in one direction) freeway would have one lane dedicated to connected and autonomous vehicle platooning and another that allowed and accommodated mixed-use traffic [19]. It was also noted by Chen et al., Shladover et al., and Milanes et al. [19]–[21] that platooning could reduce the headway of vehicles from 1.5 seconds down to 0.6 seconds, further increasing capacity and congestion mitigation. Ye and Yamamoto [13] reported similar findings, which showed that reducing the time gap between vehicles, increasing market penetration, and increasing connectivity between vehicles extends the free flow density by increasing roadway capacity.

Freeways and interstates are not the only roadway classification where CAV implementation has been modeled. Researchers have extended their work into areas that frequently suffer from congestion issues, such as intersections and freeway ramps. Rios-Torres and Malikopoulos [22] found that effectively coordinating CAVs could help smooth the rapid acceleration, deceleration, and occasional stop and go nature of merging highways. In doing so, they [22] reported that fuel consumption improved around 50%, and travel time improved nearly 15% in the merge locations. Another location where vehicle connectivity seeks to provide

benefits is at urban intersections. Zhang et al. [23] conducted a simulation that showed the possibility of optimizing vehicle movement through intersections utilizing V2I and V2V communications, even finding that with the implementation of CAVs, the need for signalized intersections may become a relic of a previous generation of transportation infrastructure. Using optimization of acceleration and deceleration rate and collision avoidance as the primary constraint, Zhang et al. [23] found that fuel consumption improved by nearly 45% and travel time improved almost 40%.

Researchers have also focused on non-traditional aspects of transportation network efficiency, modeling principles like congestion avoidance and dynamic routing through V2V and V2I communications. Bauza et al., Milojevic and Rakocevic, and Yang and Bagrodia [24]–[26] modeled this, showing that broad and efficient communication between vehicles and infrastructure allowed vehicles to reroute themselves to avoid congestion. Yang and Bagrodia [26] found that dynamic routing in the event of a crash (or other disruption), when compared to a control model without it, reduced overall travel time by 37%, reduced overall system delay time by 57%, and increased vehicle throughput to their destination by 118%. Through modeling and simulation, Bauza et al., Milojevic and Rakocevic, and Yang and Bagrodia [24]–[26] demonstrated the ability of VANETs to improve future transportation efficiency at a network level, reducing congestion and emissions in urban environments and allowing for a higher range of mobility among users.

While research points to likely increases in efficiency and safety through the implementation of CAVs, it is important to note that with any emerging technology, there are downsides. Since CAVs are in the early stages of development and implementation, identifying problems and offering solutions before they can cause harm is needed for a successful rollout.

Aside from the cybersecurity risks associated with CAVs, the basis on which this research is predicated, potential societal and behavioral downsides arise from the increased access provided by CAVs. Fagnant and Kockelman [27] found that the increased comfort of CAVs may lead to an even further increase in urban sprawl while increasing total Vehicle Miles Traveled (VMT), an increase in travel that may offset or exceed the fuel economy savings by producing more harmful emissions. The introduction of CAVs may harm equity even though they have long been marketed to bridge inequities in the transportation network. Litman [8] reported that the initial implementation of CAVs in the 2030s and 2040s would mainly provide further mobility only for affluent customers. It was also determined that these vehicles would become more common and affordable in the 2050s and 2060s [8]. However, with nothing specific to promote equity and accessibility to the technology, CAVs may not initially, or ever, benefit those who need them the most.

The proposed benefits may seem to outweigh the possible drawbacks in implementing CAVs. Still, for too long, the transportation system has not done enough to promote equity. The early implementation phases of this new technology pose an opportunity for transportation decision-makers to create a system designed from its inception to promote a healthy and equitable system, providing affordable mobility and access to those who need it the most. Engineers and planners should not ignore the behavioral and societal impacts and consequences that the transportation system carries. At the same time, they should aim to create a system that increases safety and maximizes efficiency.

Cyber Attacks on the ITS and VANET

It is first essential to understand what a VANET is to understand the possible cyberattacks against it. A VANET is a variation of what is known as a mobile ad-hoc network

(MANET), with the main difference being that the mobile nodes on the VANET are vehicles and the fixed points are RSUs [28]. A VANET is a critical element that could allow the future transportation system to be truly connected through V2V and V2I communication, eliminating the notion that vehicles are "information-isolated islands" [29].

VANETs are relatively complex networks that consist of various components, each serving a pivotal purpose to make the entire network run correctly. These components can be categorized into three distinct but connected domains. The three domains are the in-vehicle domain, the Ad-Hoc domain, and the infrastructure domain, each containing different components and capabilities, serving its purpose in the network [30]. These three domains interact with each other, enabling the network to support V2V and V2I communications.

The first domain of a VANET is the in-vehicle domain. According to Al-Sultan et al. [9], there are two primary components of the in-vehicle domain, the Onboard Unit (OBU) and the Application Unit(s) (AUs). The OBU is responsible for communication between vehicles themselves and between vehicles and RSUs [28]. The primary purpose of the AU is to run applications using the communication abilities of the OBU; these units can be focused on safety or more casual, internet-based applications [9], [30]. Al-Sultan et al. [9] pointed out that an AU can be placed with the OBU as a single entity but differentiated the two as AUs can only send messages through the OBU, as the OBU is the only component in this domain that can communicate with the rest of the network. In addition to the two parts mentioned previously, Hamida et al. [30] presented that a vehicle's Electronic Control Units (ECUs), Global Navigation Satellite System (GNSS), and Trusted Platform Module (TPM) are considered in the in-vehicle domain. ECUs are tiny computers embedded within a vehicle that control the electrical systems in the car and collect data on the vehicle and environment (sensors) [30]. These small computers

can handle various functions in a vehicle, ranging from tire pressure monitors to safety-critical functions like transmission control and control of the Anti-Lock Brakes (ABS) [31]. The TPM is used to manage certificates and keys to ensure reliable communications, and the GNSS provides GPS and location information and tracking for the vehicle [30].

The second domain of a VANET is the Ad-Hoc domain, which is sometimes referred to as the Vehicle-to-Everything (V2X) environment [30]. The Ad-Hoc domain is where the V2V and V2I communications occur as individual OBUs communicate with each other and with RSUs [9]. These two components connect wirelessly through Dedicated Short Range Communication (DSRC), which is regulated through the Institute of Electronics and Electrical Engineers (IEEE) standard 802.11p: The Standard for Wireless Access in Vehicular Environments (WAVE) [29]. Using this standard, vehicles can send safety and network efficiency messages through V2V and V2I communications and use the network of RSUs to disseminate messages far throughout the Ad-Hoc network [29]. The Ad-Hoc domain is the critical domain that allows for safe and efficient CAVs in the future; it is also the most pressing domain to secure to prevent and mitigate cyberattacks on the system.

The final domain is referred to as the infrastructure domain, where RSUs connect with Trusted Third Parties (TTPs), which consist of car manufacturers, internet providers, and trusted authorities (police and emergency services) [30]. Hamida et al. [30] described the infrastructure domain as the vital link between vehicle OBUs and TTPs using RSUs as connection, providing applications, over-the-air software updates, and connecting to the internet.

Now that the architecture of a VANET has been described, the operations and characteristics of the network can be described. According to Kelarestaghi et al. [12], the main characteristics of VANET have been defined as high node mobility, dynamic topology, frequent

disconnections, an open-air transmission medium, predictability, and having sufficient energy for data storage/computation. The high node mobility is a direct result of the constant movement of vehicles within the network, making network routing a challenge [12], [28]–[32]. Similarly, the network's topology can change based on the time of the day, the density of nodes in the network, and the speed at which the nodes are moving - resulting in a large number of short communications [12], [30]. Although connections are relatively short, the nodes (vehicles) are constrained by the built environment, allowing their locations to be predictable as they should all be within the streets [12], [29].

As a result of poor weather conditions, the built environment, and the dynamic nature of the network topology, the network is susceptible to frequent disconnections [12]. The network's sensitivity to the weather conditions and built environment highlights that a VANET is transmitted in the open-air setting. Using an open-air medium makes the network sensitive to environmental factors. It leaves the network vulnerable to cyberattacks as there is no physical protection for the system[12], [30]. Due to the abundance of energy provided by a car battery and the computational power of OBUs and RSUs, a VANET can incorporate complex algorithms that may take large amounts of energy or computational power and storage [30].

In addition to these characteristics, Hamida et al. [30] also detailed two additional features, stating that the network size is unbounded and scalable and time-sensitive. A scalable network could allow a single network to cover a diverse range of transportation networks; in theory, all of NYC could be covered by a single network. It is easy to see why messages in a VANET are time sensitive. The network is responsible for delivering safety-critical information to vehicles. A delay in this delivery due to network overload or a malicious attack could have harmful, possibly fatal, consequences for those in the network.

The implementation of VANETs and CAVs is expected to change the landscape of the transportation network significantly. The enabling technology at the heart of this change will be developing VANETs and their potential applications. These applications were summarized by Cruz-Cunha and Moreira [28] as V2V warning propagation, V2V group communication, V2V beaconing, and V2I/I2V warnings. Combining these four overarching applications should increase safety and comfort for all users who depend on the transportation network while increasing its overall efficiency. Al-Sultan et al. [9] categorized communication applications as either comfort/entertainment applications or safety applications. Entertainment applications are a single category that includes using the internet, sending messages, and playing video games. In contrast, the safety category was broken down into five subcategories: intersection collision avoidance, public safety, sign extension, vehicle diagnostics and maintenance, and information from other vehicles. Bauza et al. and Milojevic and Rakocevic [24], [25] demonstrated a VANET's ability to detect congestion in the transportation network, and Yang and Bagrodia [26] showed the network's ability to account for congestion and implement dynamic routing to reduce travel time and decrease overall delay. While these specific examples do not represent an extensive list of safety and operations applications, they highlight the possibilities supported by V2V warning propagation, V2V group communication, V2V beaconing, and V2I/I2V warnings.

While CAVs and V2V/V2I communications promise to improve the future of transportation, they do not come without challenges. One considerable challenge facing this future system is the threat of cyberattacks and other malicious activities. To begin to combat and neutralize these threats, many researchers have proposed base security requirements for VANETs. Cruz-Cunha and Moreira [28] and Sumra et al. [33] identified eight security requirements for VANET: [unique] entity identification, entity authentication, attribute

authentication, privacy preservation, non-repudiation, confidentiality, availability, and data trust. The requirements listed by Hasrouny et al. [34] mostly overlapped the previous eight requirements but described liability identification in place of entity identification. The two concepts are similar and more or less serve the same purpose. The report also listed four more security requirements: access control, traceability and revocability, error detection, and network flexibility and efficiency [34]. Hamida et al. [30] named an additional condition; the system needs to be robust against external attacks, which is a requirement that is more difficult to quantify compared to others. While the last requirement may be challenging to measure, this single requirement almost summarizes the goal of this long list of security requirements in three words: "robust against attackers.". These three words are valid, but they may be an oversimplification of the requirements that the system's security should meet. Al-Kahtani and Kelarestaghi [11], [12] briefly described the five main security requirements: availability, authentication, integrity, confidentiality, non-repudiation, and privacy. A more in-depth way to summarize the goals of VANET security requirements was presented by Samara et al. [35], a VANET should ensure: correct information is being exchanged, the message carrying the data has integrity, the source can be authenticated by the message receiver, privacy among the nodes versus tracking and identification, and the system as a whole is robust. Meeting these requirements would ideally create a VANET that is resilient against attack, ensuring confidence that the connected transportation network is safe and secure for public use.

While engineers are working to ensure a safe and secure system, there are those on the other side working to attack and infiltrate VANETs. Ahmad and Adnane [36] defined an attack as "an attempt to gain illegal and unauthorized access into a system by exploiting its vulnerabilities" and stated that attacks are the result of a lengthy planning process with the hope

of personal gains for the attacker. Attackers, often referred to as adversaries, have generally been categorized by four metrics: their network membership (insider vs. outsider), their motivation (prankster vs. attacker seeking reward), their attack method (generate malicious message packets vs. eavesdropping), and the scope of their attack (a local and limited attack vs. an extended attack with a broad reach and an extensive range of control and manipulation) [12], [34]. Stellios et al. [37] also noted that an attacker's technical capability has a significant degree of influence on the impact and scope of the attack itself.

Adversaries and malicious nodes have been classified by mobility, being defined as static within the network (in a parked vehicle or near an RSU) or moving within the network (in a moving car) [36]. An adversary can pair its mobility with the vehicular mobility properties of a network (i.e., jam vs. free flow and urban vs. rural) to create four separate adversarial attack models: high vehicular mobility with a static attacker, low vehicular mobility with a static attacker, high vehicular mobility with a dynamic attacker, and finally low vehicular mobility with a dynamic attacker [36]. These four scenarios serve as a very high-level introduction to the scenarios in which an attack, cyber or physical, could occur, but it is by no means comprehensive.

To understand the impacts adversarial attacks can have on the transportation system, one must understand the potential cyberattacks and methodologies fully in addition to cyber and physical attacks that have already been carried out - whether in the real world, in a lab setting, or through modeling. One of the more challenging aspects of building CAV resistance and resilience against attacks is the variety of attack surfaces available to adversaries.

There are two main categories of threat models: physical access to the vehicle or infrastructure and attacks carried out through the open air (wireless) [31]. This categorization

may falsely imply a level of simplicity that does not show the complexity of the numerous attacks possible through these two vectors.

It is important to note that these two attack mediums are not restricted to attacks on the emerging technology that is VANETs. Vehicles and infrastructure currently in use can also be compromised through physical and wireless attacks [31]. There is a distinction between attacks conducted in real-world conditions, whether nefarious or carried out by white-hat hackers, attacks carried out in a controlled or laboratory environment, and theoretical attacks that have only been analyzed through advanced modeling software. This section will only include physical and wireless attacks that have been performed on vehicles and other transportation infrastructure that have either been successfully perpetrated or have been conducted in a controlled manner. These attacks do not include attacks that hackers can use to compromise a VANET. Those attacks generally fall under the umbrella of theoretical/modeled attacks and will be discussed later.

One of the most publicized and well-known vehicle hackings occurred in 2015 as white-hat hackers Charlie Miller and Chris Valasek were able to remotely kill the engine of a 2014 Jeep Cherokee as it drove down the highway [38]. This attack was not the first performed by the duo as they were able to manipulate a vehicle in 2013 using a cable plugged into the car's OBD-II port on one end and a laptop on the other end [39]. The 2015 attack was, however, the duo's first real-world wireless attack, with the test acting as a successful proof of concept [38]. The hacker's successful infiltration of the automobile resulted in a recall of 1.4 million vehicles by Fiat-Chrysler Automobiles and a modification of Sprint's wireless cellular network [38]. To understand how these two were able to perform these attacks, it is crucial to know how a vehicle's internal electronics operate and communicate. Most people consider modern

automobiles to be mechanical machines with electronic features (especially newer vehicles), but what they do not realize is that a modern car (2009) generally has over 70, and up to 100, ECUs [40], [41]. ECUs play an essential role in "virtually every aspect" of the operation of a vehicle, with applications ranging from controlling the breaks and transmission to controlling a passenger's heated seats [31]. All of these ECUs have an inherent need to communicate with each other and the vehicle itself. This efficient and time-critical communication is provided by a computer network referred to as the Control Area Network (CAN) [40]. The CAN operates on a multilayered network that uses a bus digital network architecture. Subsequently, the communication system is known as the CAN bus [40]. In the case of the hacked Jeep Cherokee, the vehicle had two CANs (CAN C and CAN IHS), each with unique ECUs, but also sharing critical communication functions like Bluetooth, radio, and cellular (telematics unit) [42]. Any communication between electronic components within a vehicle utilizes the CAN; this includes sensors used in newer vehicles that allow for functions like adaptive cruise control and intelligent lane assist. The interconnected nature of the ECUs via the CAN bus introduces a significant security flaw into the system. Koscher et al. [31] and Checkoway et al. [43] reported that an adversary could gain complete control of a vehicle's electronic systems by compromising a single ECU. Using the communication framework provided by the CAN bus, adversaries can study the network and its behavior and insert false message packets into the system for the vehicle and other ECUs to execute [31]. Attacks against the CAN bus can severely impact a driver's safety and the safety of other drivers in the surrounding area. Miller and Valasek demonstrated that safety-critical functions of a vehicle, such as steering and braking, could be compromised through a successful wireless attack on a vehicle's infotainment system. Similarly, Koscher et al. [31] compromised a vehicle's braking control module by bridging the vehicle's

two internal CAN networks, connecting a hacked telematics unit to safety-critical braking functions. Also, the researchers were able to write the nefarious code so that it was automatically erased, effectively creating an untraceable bug [31]. It is important to note that these tests were conducted in a controlled environment with physical access. Still, the results could easily be applied to a wireless attack using Miller and Valasek's methodology.

To further illustrate the wide range of attack surfaces on modern vehicles, Checkoway et al. [43] established three main attack mediums: physical access, short-range wireless access, and long-range wireless access.

For the most part, the physical access attacks have been focused on the On-Board Diagnostic (OBD) II port. The federally-mandated port allows technicians to diagnose internal problems within a vehicle; it will also enable adversaries to send and decode messages to the vehicle's CAN bus [31]. It would be intuitive to assume hackers would need physical access to insert a malicious device to access the internal network, but Foster et al. [44] proved that popular aftermarket Telematics Control Units (TCUs) that utilize the OBD-II port could be used as a source of entry. The study found that through tracking SMS and IP addresses (found through Google search), the TCU could be compromised and subsequently send CAN packets that resulted in compromised braking and engine functions [44]. It is noted in the study that the experiment was conducted at a close range, but the effective range is infinite. This study showed that attack surfaces traditionally only accessible through physical attack could be compromised through an over-the-air attack through popular plugins and after-market products.

The next group of attack vectors described by Checkoway et al. [43] is the short-range wireless access which includes Bluetooth, Remote Keyless Entry (RKE), Tire Pressure Monitoring System (TPMS), and more emerging technologies (V2V communication). Oka et al.

[45] found that while it is not easy, hackers can pair with a vehicle's infotainment systems (mainly external systems), giving a hacker the ability to send damaging messages to the CAN bus. Checkoway et al. [43] performed a similar attack that required the attacker to know the Bluetooth Medium Access Control (MAC) address (easily sniffable); they found that over a day if thousands of vehicles leave a parking garage, a hacker could pair with at least one vehicle in under a minute. These attacks on Bluetooth are enabled through brute force attacks on the PIN required for pairing; these PINS can be either random or fixed [43], [45].

The final grouping of attack vectors described is the long-range wireless access which consists of broadcast networks (GPS and Satellite Radio) and addressable channels (TCUs) [43]. As previously mentioned, Miller and Valasek most notoriously proved the ability to attack a vehicle through its telematics unit. Checkoway et al. [43] was able to perform a similar attack, compromising a vehicle through an attack on the TCU.

This wide range of attack surfaces has proven that an adversary can infiltrate a vehicle's internal communication system through many vectors, ranging from sensors to telematics units. To ensure a higher standard of safety in modern vehicles and future CAVS, researchers and manufacturers must put network and cybersecurity at the forefront of vehicle and sensor development to ensure the security of the future transportation system.

Modern vehicles are not the only part of the transportation network with a history of physical and remote attacks. Kelarestaghi et al. [46] showed that hackers, notoriously "The Sun Hacker," could wirelessly infiltrate roadside Variable Message Signs (VMS) and modify them to display any message desired, whether humorous or to display false information. Hackers were able to access these signs by cracking the passcode for the device's Virtual Private Network

(VPN) [46]. The attacks led the FHWA to release guidelines for strong passwords and other network security features.

While this may seem like a harmless prank, two main takeaways have apparent negative implications. The first being that such a prank could distract drivers from the road, which compromises road safety and could potentially lead to vehicle crashes [46]. The second consequence may not be an issue right now but could lead to significant safety implications in the future. The report presented that a compromised VMS could allow hackers to access the ITS network as a whole [46]. Similarly, Cerrudo [47] theorized that variable speed limit signs could also be compromised through weak wireless network security, thus granting access to the network at large.

Variable and changeable transportation infrastructure systems are not the only roadside infrastructure that has been shown to be susceptible to malicious intent. Traffic signals have been the target of one real-world attack and one study/proof of concept in the past. In 2006, two Los Angeles city traffic engineers illegally accessed the city's wireless network that controlled the wireless operation of traffic signals in the city [48]. While this incident is often referred to as hacking, the employees simply used an authorized user's login to access the program; the act was more along the lines of identity theft than hacking. All that being said, once they had access, the engineers changed the signal timing at four critical intersections throughout the city, creating a delay that lasted for several days [49]. The attackers extended the red time for the mainline traffic and allocated excess green time to the side roads. The attack resulted in massive gridlock around these intersections; one intersection backed up to Los Angeles International Airport (LAX) [48]. This attack shows the consequences of hacking, as a malicious actor could recreate this attack by imitating an authorized user to access the system.

Cesar Cerrudo demonstrated that an attacker does not need access to a traffic control system to control traffic signals [47]. With less than $100 worth of hardware, Cerrudo was able to compromise sensors that are used at over 50,000+ intersections throughout US major cities (NYC, SF, LA) and cities throughout the World (UK, Canada, Australia) [47]. In another case study, Ghena et al. [50] teamed with Cerrudo to examine and attack real-world traffic signals that utilized wireless communication in Michigan. They found that the network lacked basic network security. No encryption or authentication process had been implemented. It used generic and default login credentials, resulting in relatively easy access to the network from an adversary [50]. This real-world case study proved what was already known by Cerrudo. Weak wireless network security could result in a compromised network, allowing an adversary to manipulate the signal timings [47], [50]. The Malfunction Management Unit (MMU) within a signal controller restricts the scope of harm attackers can do, preventing dangerous signal combinations hardcoded in the system. Even this fail-safe did not prevent denial of service attacks (all lights remain red) and signal timing changes [50], similar to those carried out in the LA signal attack. In another study [51], Cerrudo also claimed that compromised sensors could act as a malicious worm, compromising other connected sensors through the wireless network. Like the previous infrastructure attacks, this aspect could allow an attacker to inflict a broad scope of attacks against connected devices on the network.

In his findings, Cerrudo also claimed that these attacks are challenging to detect [52] and pointed to a more significant security issue in the industry: a “lack of security consciousness in the entire field.” The blind eye from the industry sector could make these attacks easier to perform against various traffic signal devices and systems used throughout the World [50]. These wireless attacks have the potential to create widespread congestion and potentially lead to more

vehicle crashes. They also showed that the attacks performed by the LA traffic engineers could be achieved by hackers wirelessly and with relative ease. Cerrudo brought his findings to the USDOT, but he claims they were not worried about it as they had “worse things to worry about” [52].

As wireless technology has developed rapidly in vehicles and transportation infrastructure, network security has lagged, opening the door for a range of malicious attacks [50]. The attacks currently able to be performed against modern vehicles and infrastructure are concerning to present-day wireless security and present a scary possibility of future attacks. If the claims made by Kelarestaghi et al. [46] are correct, these infrastructure attacks can be used as a gateway to the broader ITS network; hackers could compromise future V2V and V2X systems by compromising the intelligent infrastructure in place that has already been proven to be vulnerable. Koscher et al. [31] agreed with this concept, introducing the idea that the implementation of V2V and V2X communications will present an even more broad attack surface, making vehicles even more challenging to secure. It will be nearly impossible to protect future VANETs if the existing infrastructure that V2X will connect in the future is already so easily compromised.

The previous sections have detailed a long list of physical and wireless attacks hackers could perform against modern vehicles and already-in-place intelligent transportation infrastructure. The following sections will describe the attacks that have been modeled and theorized against the future transportation network components, including VANETs, V2V communications, and V2I communications.

Similar to attacks that have been performed against modern vehicles, attacks against VANETs can be rooted in a physical or wireless attack. However, a vast majority of research has

focused on wireless attacks. While previously discussed attacks focused on attacking ECUs within a vehicle network, the research on future VANET attacks has focused more on disrupting the network and, going forward, will be referred to as cyberattacks. Al-Kahtani [11], Kelarestaghi et al. [12], Cruz-Cunha and Moreira [28], and Hamida et al. [30] categorized these cyberattacks by which of the five VANET security goals they target: availability, authentication, integrity, confidentiality, non-repudiation, and privacy. It is important to note that of the at least twenty-two attacks reported in Kelarestaghi et al. [12], an overwhelming majority of them fell within at least two categories. The following section will give a brief overview of each attack, grouping them by the security requirement they compromise. Even though many attacks fall into multiple categories, they will only be assigned one group in this report to avoid redundant information. The following list is a thorough exploration of potential attacks, but it is not completely exhaustive.

Availability

- Denial of Service (DoS): An attacker prevents important messages from either being sent or received from vehicles or RSUs, by jamming the network. The attacker could overload the network or specific channels to prevent message transmission. It is important to note that this attack will be a primary focus of this research [11], [12], [28], [30], [33]–[36], [53], [54].
- Distributed Denial of Service (DDOS): Instead of one malicious node jamming the network/communication channels, multiple malicious nodes exist within the transportation network, sending out large numbers of compromised messages [12], [30], [33], [34], [54].
- Jamming: An attacker introduces noisy communications into, or overloads, the system to increase interference and reduce communication capabilities. Attackers can perform this attack against wireless and wired components and vehicle sensors [12], [30], [34], [36].

- Greedy Behavior: A node does not follow timing protocols and sends messages at its own will with the intent of producing favorable results for the adversary [12], [28], [30].
- Spamming: Sending spam messages through the network to introduce latency into the network, using up bandwidth, and slowing network communication [11], [12], [30], [33].
- Blackhole: A node, or group nodes, attract network traffic by advertising the shortest routing path. The information packets are dropped as there may be no legitimate nodes in the destination area, or the malicious nodes purposely do not accept the packets being sent, effectively causing a dead spot in the routing path [11], [12], [30], [34].
- Malware/Spam: An attacker uses updates to RSUs and OBUs to insert spam, viruses, and other bugs. These attacks decrease network efficiency and increase delay [11], [12], [30], [34], [36].
- Timing: An attacker purposefully delays the transmission of a critical message [11], [12], [30], [33], [34].

Authentication and Identification

- Man in the Middle (MITM): A malicious node pretending to be an authenticated user intercepts messages between RSUs and OBUs and can either delay the message, spoof/falsify the message, or drop the message altogether. It is important to note that this attack will be one of the attacks investigated in this research [11], [12], [30], [33], [34], [36], [55].
- Sybil: An attacker generates multiple pseudonyms, causing legitimate nodes to assume heavy traffic or believe false locations of aliases - degrading traffic and network operations [11], [12], [28], [30], [34]–[36].
- Replay: A malicious node impersonates an authorized user by replaying an earlier message or inserting an old message to manipulate the contents or outcomes [12], [30], [34]–[36].

- Wormhole / Tunneling: Two malicious nodes create a tunnel through the network to transmit and broadcast data packets. The tunnel can make two authentic nodes think they are close on the network while not being near the other geographically [11], [12], [30], [34], [36].

Confidentiality and Privacy

- Eavesdropping: Attackers can gain access to the network and can extract sensitive data from packets such as location and identifiable information [12], [28], [30], [36].
- Tracking: An adversary can capture, track, map, and record a vehicle's location [12], [28], [34].
- Information Gathering: An attacker listens to the network long enough to gather valuable information such as identity, location, traffic data, etc. [28], [30], [33].

Integrity and Data Trust

- Spoofing: An attacker introduces fake signals, tricking sensors or nodes into reporting fake or false information. They are often performed against sensors, GPS systems or performed to impersonate legitimate nodes [11], [12], [30], [34], [36].
- Gray hole: Variant of a black hole attack, instead of dropping information packets, malicious nodes send their packets in place of legitimate messages [12], [30], [34].
- Message Tampering: An attacker fabricates or alters the contents or delivery time [12], [30], [34], [35].
- Message Suppression: An attacker drops select messages to support their goal. The attacker can use these messages again later (replay attack) [12], [30], [34], [35].
- Bogus Information: An attacker inserts false information into the network for personal gain or degrades network/traffic performance [11], [34], [36].

- Illusion: An adversary can transmit false information to neighboring vehicles, giving the illusion of a non-existent traffic condition such as a jam or a crash [11], [12], [34].

Non-Repudiation and Accountability

- Cryptographic Replication: An attacker obtains, then copies, an actual key or certificate to hide their true identity while gaining access to the network [12], [30], [34], [54].
- Masquerading: An attacker appears like another valid vehicle then uses this access to produce false information to disrupt the network and achieve their objectives [11], [12], [30], [33], [34].
- Impersonation: A malicious actor uses legitimate nodes to send messages on their behalf, using the authentication of other vehicles, and then hides in the network [11], [12], [28], [34], [36].
- Repudiation: An attacker performs an attack using a valid user's ID, making it difficult to track or prove the attack's origin and the perpetrator's identity [12], [30], [34].

It is important to note that many attacks listed above fit into more than one category of attack. Attacks were grouped by their primary or most common targeted security requirement. While DoS and MITM attacks were briefly mentioned in this section, the following two sections will provide more details on the selected attacks. Since these two attacks are the main focus points for this research, it is essential to understand how they are conducted, the vulnerabilities they exploit, and the consequences they carry.

Man in the Middle Attacks on the VANET

While there are multiple goals of a man-in-the-middle attack, at its core, it can be defined as a malicious actor (vehicle) intercepting a message between two nodes in a VANET [11]. MITM attacks compromise many of the VANET's core security requirements, including

authentication, integrity, confidentiality, and non-repudiation [12]. Ahmad et al. [55] also found the possibility of using MITM attacks as a method of eavesdropping, adding privacy to the list of VANET security requirements violated. The wide range of uses and consequences of this attack has made it a research priority in the past and shows why it continues to be one. Ahmad and Adnane [36] categorized a MITM attack with a possible likelihood to occur, resulting in medium impact and presenting a significant risk to the public.

Adversaries can perform MITM attacks in two main ways, either take advantage of non-encrypted messages being sent through the system or enter the network through an insecure wireless network [36]. An attack generally has three active attack goals: to delay a message transmission, drop a message altogether, and modify/falsify a message, but can also be used for passive attacks such as eavesdrop [55]. Hackers can use MITM attacks to attack the wireless network itself and the supporting infrastructure [34].

Due to a MITM attack's ability to directly prevent authentic messages or alter the time they are delivered; these attacks carry severe consequences to the network. These attacks seek to compromise the vehicle connectivity in a CAV network and, if completed successfully, can directly prevent the safety and efficiency benefits promised by the system [55]. As a result of tampered messages, message delay, and packet losses, MITM attacks were found to prevent safety-critical messages, like collision avoidance warnings, from reaching legitimate nodes in the system, resulting in safety degradation [55]. Without reliable and authentic communications between infrastructure and vehicles and among vehicles themselves, a VANET is prone to poor system performance, resulting in decreased safety and operations. These attacks seek to compromise the basis that a CAV system is built on, which makes understanding the impacts of these attacks critical for developing the resilient critical infrastructure of the future.

Denial of Service Attacks on the VANET

A denial of service attack can be simply defined as an attack that prevents or inhibits an authenticated user's ability to access the VANET [33]. DoS attacks seek to compromise the availability of the VANET to users, one of the leading security requirements for a successful implementation of the VANET [28]. This category of attack is of particular interest for research, as Kelarestaghi et al. [56] stated that an availability attack on an in-vehicle network had a very high likelihood to occur as well as a very high impact rating. Ahmad and Adnane [36] took this notion a step further, showing that a DoS attack against a VANET was likely, with a high impact, and presented a critical risk. All three designations were the highest possible for each category in the reported threat analysis.

Three different methods of performing a DoS attack have been defined through research by Sumra et al. [33] and Ahmad and Adnane [36]. The three methods are as follows: overwhelming the network by transmitting random signals that occupy all of the network's resources while reducing efficiency, jamming the network by sending large amounts of messages that restrict communication ability, and dropping packets sent through V2I and V2V communication - effectively losing the messages. These attack methods seek to compromise three targets, as described by Cruz-Cunha and Moreira [28] and Kim et al. [57], the network's protocols and communication, the network's computational ability, and the infrastructure that supports the network. Hackers can launch DoS attacks with relative ease as no network penetration is needed. An adversary can remotely launch an attack against a network vulnerability to carry out the attack [57]. An adversary can use one node to launch an attack or can use a network of compromised nodes spread out over a large area to launch more than one

DoS attack at the same time [54]. These coordinated and more damaging attacks are known as Distributed Denial of Service (DDoS) attacks.

Since DoS attacks have been described as relatively easy and familiar attacks, it is essential to understand their impact on a VANET and CAV system. Kim et al. [57] defined four levels of influence; nuisance, network degradation, network disruption, and a disabled network, with each successive categorization being more damaging to the network. Ekedebe et al. [32], modeled a DoS attack and found that the results directly diminished the proposed improvements to safety and efficiency of CAVs by reducing the system's ability to provide reliable and efficient communication. Parkinson et al. [54] claimed that V2I attacks are a more disruptive type of attack due to the larger area covered by this service; however, Ekedebe et al. [32] showed that V2I communications are more resilient than V2V communication due to its high bandwidth requirement. While studies have shown the potential crippling impacts of these attacks, Biswas et al. [58] presented that neither the sender nor the receiver will be aware of the attack as it occurs and may not be privy to the attack until the consequences are imminent and irreversible.

Combining the relative ease of attack, attack anonymity, and extreme consequences of a possible attack make DoS and MIMA attacks against the VANET especially dangerous and problematic. Modeling and understanding the real-world impacts of such attacks is important for prevention, mitigation, and resilient planning for engineers in the near future.

This research seeks to quantify the effects of a DoS and MIMA attack on transportation safety and operations to understand the consequences of each attack on the transportation network.

Cybersecurity in Transportation Engineering

Due to the multidisciplinary nature of the research mentioned above, few researchers have focused on cybersecurity attacks on CAVs related to transportation engineering. The existing research tends to be split between examining the impact of the attacks on the VANET communications and its implications on high-level MOEs and the other research focusing more on the effects of an attack on more traditional, yet focused, transportation MOEs. This research that has already been performed tends to fall short in investigating the transportation aspect. Research on the topic already completed is detailed below to detail previous conclusions while showing the knowledge gaps within the field.

ITS and low-level autonomy are not new topics within transportation, with aspects of ITS implemented for years and modern cars come equipped with driver-assist functions like lane assist and cooperative adaptive cruise control (CACC). Even with these topics being relatively current and more understood aspects of a connected vehicle future, little in-depth research has been done to quantify their risk related to cybersecurity. Ganin et al. [59] modeled resilience by quantifying delay associated with disruptions to ITS links and nodes (smart intersections and smart segments). The study found that delay was increased by nearly 20% by disruptions at ITS-controlled intersections [59]. The results demonstrated that even a low-level connected network that only incorporates ITS is vulnerable to disruptions due to attacks, resulting in significant delays within the transportation system. In the same vein of modeling attacks against existing technologies, Amoozadeh et al. [60] demonstrated the impacts of message falsification and radio jamming attacks on a CACC enabled vehicle platoon. The attacks proved that compromising CACC in vehicle platoons reduces acceptable gaps between vehicles (efficiency) and reduces each vehicle's speed in the platoon [60]. These two studies stand to show that not-so-distant

infrastructure and technology are susceptible to cyberattacks, attacks that have significant implications on transportation safety and operations.

Ahmad et al. [55] tested the effects of MITM attacks on VANETs, seeking to quantify the impact different concentrations of malicious nodes had on the VANET's behavior. The results showed that delayed messages, dropped messages, and tampered messages significantly hampered the network's ability to efficiently and effectively transmit message packets between vehicles and RSUs [55]. As a result of hampered network communication abilities, the researchers reported that the attack would compromise transportation safety and operations. A similar study performed by Grover et al. [61] demonstrated the impact of a position forging attack on the VANET's packet transmission and quantifying the effect on average vehicle speed. This study also showed that this attack against the VANET resulted in packet transmission interference, dropped packets, and a significant reduction in average vehicle speed [61]. While these two studies somewhat demonstrated how cyberattacks could impact transportation safety and operations, they focused more on VANET operations.

Two studies were found that quantified the impact of cyberattacks against CAVs in terms of transportation MOEs. Ekedebe et al. [32] studied the effect of a jamming DoS attack on a transportation network following a crash and the subsequent traffic backup. Similarly, Garib et al. [62] demonstrated how a vehicular botnet could impact traffic operations and travel times.

The two studies mentioned above focused their research on different aspects of traffic network operations. Still, neither presented legitimately quantifiable impacts on safety. Ekedebe et al. [32] set up a simulation where upstream traffic was slowed/stopped due to a crash. A jamming DoS attack was then implemented on downstream vehicles, ranging from 0% to 100% communication jam. These communications jamming attacks aimed to hamper the compromised

vehicles' ability to dynamically reroute around the traffic jam. The study showed that more vehicles remained on the jammed route as the attack reduced communications, resulting in even more delay, increased travel times, increased emissions, and decreased average speed for the compromised vehicles [32]. Garib et al. [62] used a different tactic to lure compromised vehicles onto already congested street segments, using a bot network to send falsified messages. Using different concentrations of bots, researchers demonstrated that this attack could trick vehicles into thinking a route was clear when in reality, the road was already congested with traffic and other compromised vehicles [62]. These message falsification attacks resulted in a drastic increase in trip time, delay, and congestion while significantly reducing the average speed of vehicles in the network [62].

The results are important to note from previously conducted research, but the analysis methods are also crucial in shaping the research methodology of this research. Garib et al. [62] and Ahmad et al. [55] both used Veins to model their research. Veins combines SUMO (a traffic simulator) and OMNET++ (a discrete event simulator) to allow researchers to model VANET attacks and their impact on transportation networks [63]. Similarly, Ekedebe et al. [32] modeled their work using VSimRTI - which combines SUMO and the VSIM RTI communication simulator - a similar concept to Veins. Amoozadeh et al. [60] used an open-source, developed in-house simulator VENTOS (Vehicular Network Open Simulator) to model disruptions to CACC communications. Grover et al. [61] used NCTUns-5.0, which focuses more on modeling the impact of attacks on the VANET, not the transportation network. Finally, Ganin et al. [59] utilized a series of system models, probabilities, maps, and mathematical formulas to define and analyze different attack scenarios. It is essential to understand the best practice of modeling in past research to apply it correctly for current and future research opportunities.

As shown above, the research connecting cybersecurity and transportation operations and safety exist but is filled with knowledge gaps. These knowledge gaps are the main forces that have defined and shaped the research detailed throughout this thesis, producing the goal of quantifying the impact of cyberattacks in transportation. This research seeks to connect cybersecurity to transportation through modeling and analyzing these attacks and show the potential consequences of an attack against the system.

Transportation System Risk and Resilience of CAVs

In a holistic view of the future transportation system, understanding the risks of cyberattacks and the system's resilience in response is critical to ensure user safety and overall efficient system operations in the case of an attack. Some terms need to be understood to model and quantify CAV risk and resilience. The National Institute of Standards (NIST) Special Publication 800-30 defines risk as, "A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of (i) the adverse impacts [of the event] … and (ii) the likelihood of occurrence" [64], while President Obama's Presidential Policy Directive PPD-21 defines resilience as, "The ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions … deliberate attacks, accidents, or naturally occurring threats and incidents" [65]. Understanding the risk of and the resilience against cyberattacks forces engineers, planners, and policymakers to formulate an attack prevention plan and a response and impact mitigation plan in the case of an attack.

As mentioned in a previous section, previous studies have conducted risk assessments on various aspects of ITS. Kelarestaghi et al. [46] assessed the risk of hacked variable message signs, Kelarestaghi et al. [56] evaluated the risk of in-vehicle network attacks, and Ahmad and Adnane [36] assessed the risk of attacks against a VANET. Both Kelarestaghi et al.'s studies

[46], [56] used the risk assessment methodology defined in the NIST Special Publication 800-30, while Ahmad and Adnane [36] employed the approach described by the International Standards Organization (ISO) standard 27005. While these approaches differ in their path, one thing remains constant between the two - risk is defined as a function of the likelihood and impact of an attack [64] [66]. The NIST Special Publication outlines six steps when conducting a risk assessment: identify threat sources, identify threat events, identify system vulnerabilities, determine the likelihood of the event, determine the adverse impacts, and determine the risk (as a function of likelihood and impact) [64]. A thorough understanding of risk and the impact of a risk event, such as a cyberattack, is critical for engineers and planners to attempt to prevent an attack or mitigate the effects of a successful attack.

By definition, similar to risk, a system's resilience to an attack is a function of the impact from a threat event (attack); the more severe and degrading the attack, the harder it is to bounce back. For cyberattacks, resilience could be defined as the ability for the VANET to recover or for the transportation system to recover. For this research, system resilience, and resilience in general, will refer to the transportation system's ability to heal, not the VANET's. Few researchers have focused on resilience, but Ganin et al. [59] modeled the resilience of urban ITS systems. The research concluded that a balance between "resilience and smartness" is needed to increase system functionality during threat events [59]. Further research is necessary to figure out how to maximize resilience in a connected vehicle system, but understanding how to develop an appropriately resilient system will allow CAVs to function at an acceptable level during and after a cyberattack.

To develop a robust and long-lasting transportation system that can support the implementation and growth of CAVs, engineers must understand the risk and resilience of the

system related to cyberattacks. It is crucial to understand the potential impacts that different threat events can have on the system. Modeling and quantifying the effects of an attack allows experts to make policy and engineering decisions that seek to prevent the attack in the first place, mitigate the damage done by a successful attack, and minimize the system recovery time after an attack has occurred.

Conclusion

This literature review aimed to review previous research conducted on the cybersecurity of connected and autonomous systems and the impact on the transportation system. Due to the multidisciplinary nature of this research, the review covered a wide variety of topics, including topics from the point of cybersecurity, issues exclusive to transportation, and the few cases that have connected them. The depth and diversity of review were necessary to form a complete understanding of the multiple different aspects of each discipline and how they impact and integrate into this research.

This review identified multiple knowledge gaps and provided a framework for the primary research questions in this report. This research seeks to add knowledge on the impacts of cyberattacks on the safety and efficiency of CAVs in transportation. Delivering knowledge and analysis in this currently scarcely researched topic will help develop a safe, robust, and efficient transportation network. Connecting and exploring the two engineering fields before the widespread implementation of CAVs and connected infrastructure allows engineers and experts to predict, prevent, and mitigate the negative impacts of cyberattacks, adding a level of resilience to the future network.

3. Methodology

Overview

Before simulation could occur, a detailed research plan and roadmap had to be developed. The following goals were taken into account to develop the research plan.

- Conduct an experiment that can answer the developed research questions.
    - "How does a DoS attack impact traffic operation and safety in an urban setting?"
    - "How does a MITM attack impact traffic operation and safety in an urban setting?"
- Develop a simulation that accurately models CAV movement and VANET communication.
- Create an adaptable model that can be used in different scenarios to produce consistent results.
- Create and model DoS and MITM attack scenarios and record results.
- Perform data analysis on the different attack scenarios to quantify their impact using transportation measures of effectiveness.

It was essential to create a robust methodology that addressed and accomplished the stated objectives. First, the expectations of the model had to be established. The baseline goal of the simulation was to model basic vehicle flow in a connected vehicle environment. For this, the simulation would be expected to incorporate developing technologies such as CAVs, RSUs, and V2I, V2V, and V2X communication into traffic simulation. By doing so, the impact of cyberattacks on these communication technologies could be visualized, recorded, and analyzed from a traffic engineering perspective. The goal of this simulation was not to quantify the traffic

flow benefits of connected and autonomous vehicles. It was assumed that these technologies had been implemented. The model was developed to assess the impact of cyberattacks on the system.

With the primary goal established, simulation software with traffic and communication capabilities were selected. Veins was chosen as the simulation software with the best ability to fulfill the simulation's primary goal [63]. The software was prevalent throughout the literature review. Veins was also used in the previous study conducted by Virginia Tech which simulated cyberattacks on an intelligent intersection - further establishing its credibility. Veins is open-source software that incorporates and connects OMNeT++ and SUMO. OMNeT++ is a C++-based network simulator (V2X communications) [67], and SUMO is a microscopic traffic network simulator [68]. The Veins architecture shown below in Figure 2 allows the two separate simulations to run in parallel and communicate with each other, allowing for the traffic flow simulation of a real-world connected vehicle environment [63].

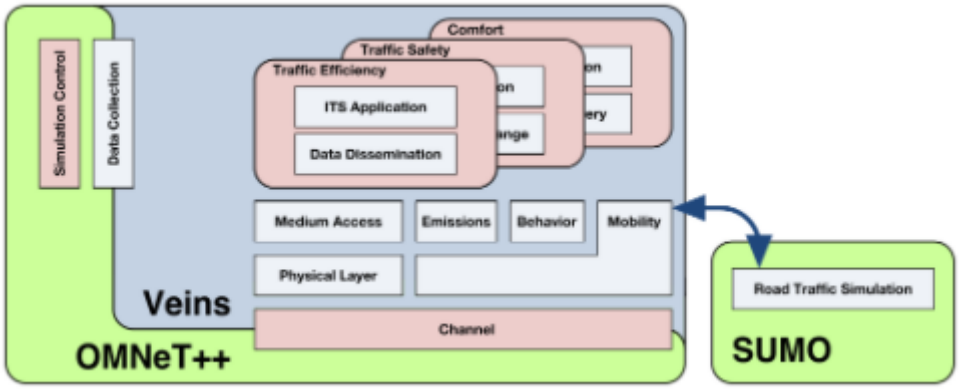


Figure 2 - Veins Simulation Architecture [63]

Traffic Simulation Development

With the simulation software selected, the next step was to determine and create the traffic simulation side of the model, including network geometry, traffic volumes, and routing. SUMO was used exclusively for this step. For both attacks, it was decided that the most impactful results would come from a simulation of an urban environment. Some of the variables

considered were the emergence of smart cities in research, high volume of intersections and vehicular interaction, smart infrastructure deployment in urban settings, and the general uniformity of an urban street network. A traditional Manhattan grid represents a relatively simple and scalable street network but a very realistic example of existing street networks worldwide and was used for this study.

The network created for this project was north-south oriented, meaning a majority of traffic originates from the northbound and southbound links, with smaller amounts of traffic originating from east and westbound links. The grid consists of six northbound and southbound travel lanes and three eastbound and westbound travel lanes, resulting in eighteen intersections within the grid network. Links were designed at 800 meters (approximately half a mile) to create equally spaced intersections at realistic intervals. All of the internal links consist of two 400-meter links connected in the middle. The network is shown below in Figure 3.

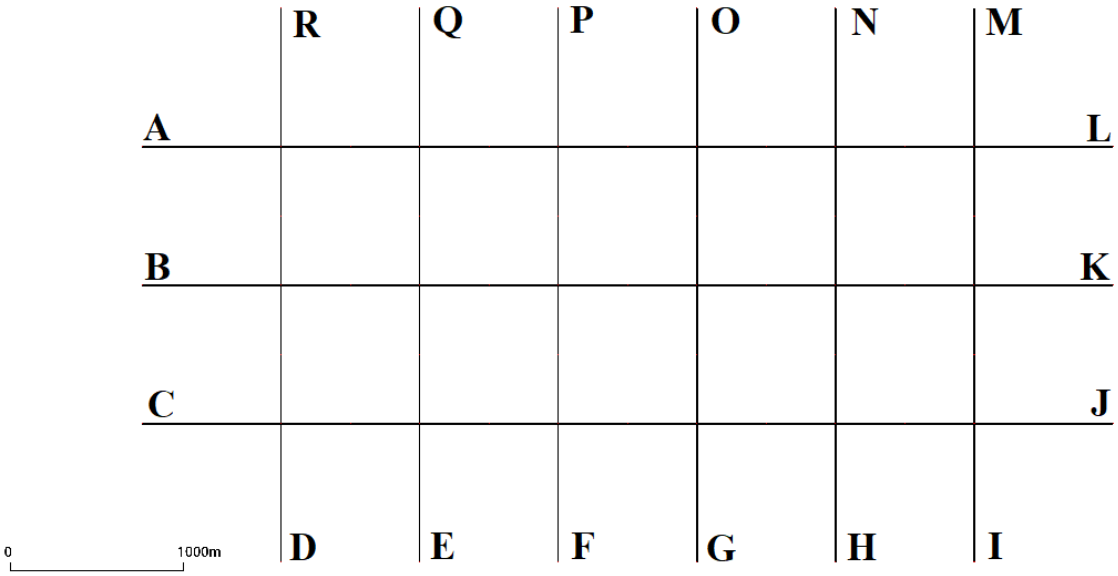


Figure 3 - Simulation Network Grid

Each link has two travel lanes, one in each direction to allow two-way flow on all connections. Each lane has a unique terminus, acting as origin and destination links for vehicular traffic. Once vehicles have reached their route terminus, they leave the simulation.

All links have equal intersection priority and were modeled as four-way stop-controlled intersections. This intersection control method is irrelevant for the Veins simulation as the vehicles make intersection reservations with the RSU which dictate right of way and intersection arrival and passage times. The simulation does not require vehicles to stop at intersections. The router uses this four-way intersection control to optimize routing decisions (discussed later).

With the network established, the next step was to add vehicular traffic to the network for simulation. The amount of traffic in the network was not incredibly important. The volume selected needed to balance not overwhelming the simulation while also producing enough vehicle interaction to run a realistic simulation of urban traffic flow. Before traffic volumes were determined, origin and destination data needed to be created; each traffic origin link would produce trips that ended at each of the other destination links. Traffic produced at A would end at B, C, D, E, …, P, Q, and R. This origin-destination pattern was used for each subsequent link, resulting in trips originating at every link ending at every other link in the network. The selected pattern encourages maximum vehicle interaction within the network. With more vehicle interactions, the consequences of a cyberattack can be more clearly seen and analyzed. This may result in an exaggerated impact, but the results are expected to be accurate as urban traffic is heavily characterized by large amounts of vehicular interaction and dense traffic streams of conflicting traffic crossing at signalized intersections.

Traffic volumes now had to be generated and routed using the defined origin-destination data. Equal traffic streams were assigned for all origin links except for links A, C, J, and L.

These four select links were assigned one-third of the main traffic generators. Less traffic originated from these links to reduce vehicle interactions at the corners of the network and emphasize internal links of the network. For the remaining fourteen links, twelve vehicles per hour were generated for each destination. Each origin link has seventeen destinations resulting in 204 vehicles per hour per main origin link, totaling 2,856 vehicles per hour for the fourteen main origin links. The minor links produced four vehicles per hour per destination (68 vehicles per hour per link). The total traffic generation for the entire network was 3,128 vehicles per hour. The vehicle distribution per hour is shown below in Figure 4. This simulation was built to be applied in a real-world scenario; in future iterations of research recorded network traffic could be easily applied to the simulation for a result specific to real world conditions.

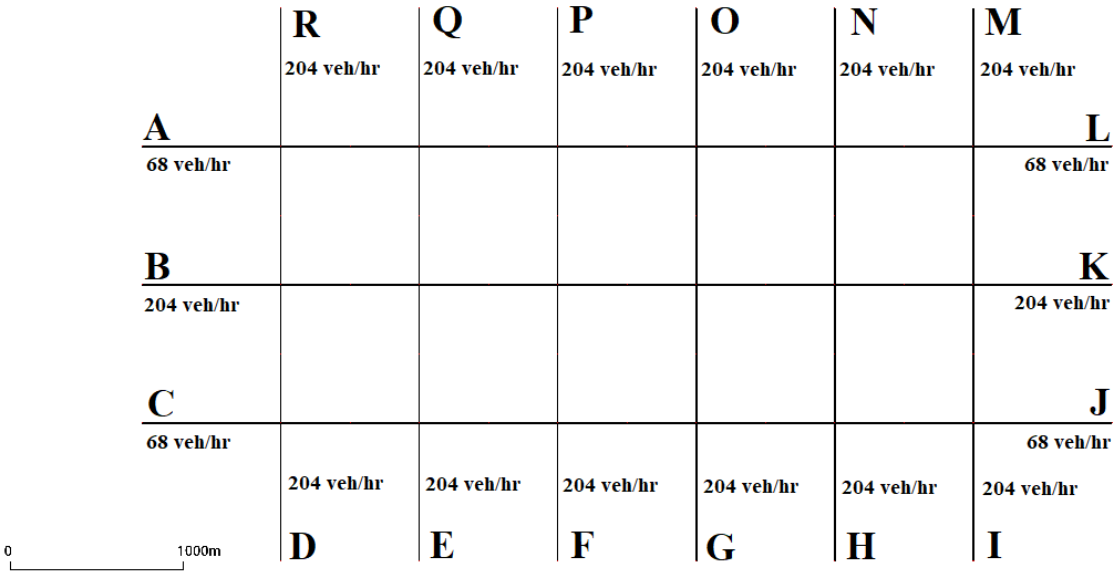


Figure 4 - Network Volume Distribution

Now the vehicles had to be routed through the network to create the base of the simulation. With the volume of vehicles and size of the network, manually routing was deemed to be beyond a reasonable scope. SUMO offers various routing possibilities, including an iterative optimization routing technique referred to as Dynamic User Assignment (DUA) and,

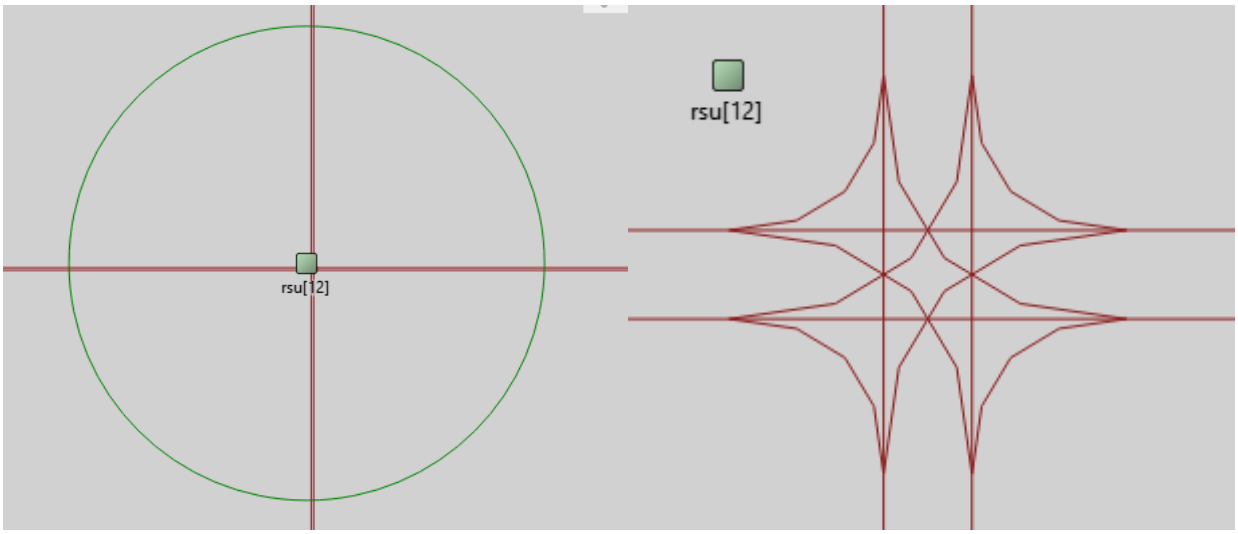
more specifically, Dynamic User Equilibrium (DUE) [68]. The duaIterate python script iterates through simulation runs to find a system equilibrium and assigns vehicle routes such that each vehicle uses a course with the least travel distance [68]. An extension to this script was used that randomized the departure times of the assigned traffic volumes within their flow definition. The randomizer generated non-uniform traffic origin times, emulating random arrivals to the simulation network expected in a connected vehicle system versus coordinated traffic signal platoons. It is worth noting that this optimization was done within the established Sumo network - meaning it used four-way stop intersection control versus the first in, first out reservation method used in the Veins simulation. This discrepancy was overlooked as it was not deemed to impact the actual simulation as the four-way stop and Veins method both used FIFO methodologies.

Roadside Unit Placement and Development

Roadside units provide critical communication needed for the implementation of connected and autonomous vehicles. RSUs were placed at each intersection to coordinate movement through the intersection - resulting in 18 separate and independent RSUs. Separate and independent are important distinctions to make to describe the functionality of the RSUs. In the simulation, the RSUs have one purpose: to regulate traffic through their respective intersections. They do not provide theoretical functionality such as dynamic routing in the case of a crash or other advantages that can be implemented using interRSU communication. Modeling these functionalities was out of this project's scope and said benefits had been demonstrated in studies previously mentioned.

Within the simulation, each intersection was controlled by a single RSU. Each RSU was placed in the northwest corner of the intersection. However, placement was arbitrary; placement

of the RSU would not impact the simulation. Vehicles arriving from the North or West would receive slight preference due to the NW location of the RSU - meaning they would make their reservations slightly earlier. This slight difference in priority did not impact the operation of the simulation. It was unavoidable as placing an RSU in the dead center of the intersection was not plausible or realistic. The RSUs had a communication radius of 350 meters, eliminating any possible communication overlap but produced a 100-meter zone in the middle of each internal link where a vehicle would not receive any communication from an RSU. This "dead zone" had no negative impact on the simulation as the vehicles operated as usual until they reached the communication radius of the next RSU on their route. The communication radius could be expanded to approximately 390 meters and still perform at the same efficiency as 350 meters. An example of RSU placement and radius of communication is shown below in Figure 5.



(a) (b)

Figure 5 - (a) RSU Radius of Communication (b) RSU Placement at Intersection

Vehicle Behavior

To correctly model the theoretical driving behavior of a CAV, a new vehicle class had to be created on the SUMO side of the simulation. The vehicle type "CAV" was created using the Krauss car following model [69]. Three parameters were adjusted to model the theoretical

perfect driving behavior of CAVs. Speed deviation was set to “0.0,” and speed factor was set to “1.0,” meaning that a vehicle would travel at its set speed 100% of the time and deviate from that speed 0% of the time. The sigma factor of the model was set to “0,” which means that the driver imperfection value is set to zero, representing perfect driving behavior [69]. The driver's behavior had no impact on vehicles’ interactions approaching and entering intersections as the RSU controlled these movements.

Simulation Assumptions

The simulation was developed to model a connected vehicle environment and quantify the results of cyberattacks against it. Since this type of environment has not been implemented yet, several assumptions had to be made to develop the simulation. These assumptions and their impact on the simulation are detailed below.

- Crash avoidance techniques of advanced vehicles were not incorporated in the simulation. Systems like RADAR and LIDAR could be used as a redundancy to CAV communications to avoid collisions in a connected environment. Full autonomy was also assumed in this aspect, meaning passengers in the vehicle would not be able intervene in the case of an emergency. This resulted in higher crashes than would have occurred if such systems were possible. While the crash numbers may be more emphasized, the conservative approach to the simulation demonstrates the true baseline results of an unprotected connected vehicle environment. Redundancies such as these would be necessary when developing a cybersecurity resilience plan.
- A few assumptions were made in the collision response modeling for the simulation. The first of these assumptions was how traffic would flow through the intersection following a crash. The way the simulation was configured assumed that vehicles would be able to

flow through the intersection at a severely reduced rate. This scenario represents a crash at an intersection where vehicles could still navigate the intersection, instead of a scenario where the intersection was completely shut down. This assumption allows for a consistent collision response from traffic, unlike the more variant nature of crash response in a real-life setting. Particularly severe crashes could shut down an intersection's operations completely at a real intersection, but in this simulation, the response is kept consistent to allow consistent comparisons to be made between scenarios. Different crash response severities could be added upon further iterations of research.

- The second collision response assumption made was the delay response speed and delay time. The delay speed was selected to be around five mph and the delay response time was selected to be five minutes. These two selections were made to simulate actual delay in the event of a collision that reduced intersection throughput. These assumptions were made directly following the previous assumption. Slowing down the vehicles for a period of time following the collision creates a consistent collision response for all crashes in the simulations. The assumption introduces delay in the simulation much like in a real-world scenario but does so in a different approach. It simulates the delay itself instead of the traffic flow theory that results in delay. Again, the delay response is something that could be built upon in future research iterations.
- Another assumption lies with the volume of vehicles in the simulation. This volume was arbitrarily created to best recreate an urban environment but could be easily adapted to incorporate a city's traffic counts. The traffic flow and vehicle volumes were kept consistent throughout each simulation scenario. This assumption resulted in consistent

vehicle interaction throughout the simulation, much like one would expect to see in an urban environment, where vehicle routing on a day-to-day basis is relatively consistent. While this assumption was made, the flow and volumes could be adapted depending on the client's preferences.

- The base scenario represents an assumption that without any cyberattacks, vehicles will operate perfectly and result in minimal delay and no crashes. The simulation setup prevents crashes from occurring under normal operations of the RSU reservation systems. The base scenario may not be how vehicles operate in the future as the technology has not yet been developed yet, however it serves as a control group for comparison for the attack scenarios. As technology progresses and CAV planning and implementation continues, the simulation setup could be changed to meet the changing simulation needs.
- The final assumption made during simulation setup was that the attacks themselves were able to be carried out by a perpetrator. Again, the purpose of the simulation was not to simulate the attacks themselves, but to simulate how the transportation system was impacted by them. Along with this assumption it was the assumption that no cybersecurity countermeasures had been implemented prior to the attacks. These assumptions allowed for a very controlled scenario setup, where the number of RSUs and the attack success rate could be manually adjusted by the researcher. This element of fine-tuned control allows the research to focus on the impacts of the developed scenarios instead of the actual attacks themselves. This control also allows for a wide variety of scenarios to be tested going forward, simply by changing the two main independent variables.

Base Model Development

The Veins model used for this research was an extension of a simulation model developed for the by a Virginia Tech research team. The model was developed in OMNeT++ and was based on constant communication between RSUs and vehicles within the communication radius. The simulation begins with each RSU sending out beacon messages every half a second. The vehicle accepts the beacon message and sends out a message directed at the RSU requesting a reservation at the intersection upon entering into the communication radius. The RSU accepts this message, checks for conflicting reservations from other cars, sends back a delay time (zero if no delay is needed) and the total calculated time for the vehicle to reach the intersection. The vehicle is added to the RSU's reservation list. Using the total time from the RSU, the vehicle then performs calculations and adjusts its speed accordingly to pass through the intersection safely; once the vehicle exits the intersection, it then speeds back up to its maximum speed. Suppose the delay is zero, the vehicle proceeds without adjustment. This brief overview is a simplified summary of how communications occur within the simulation. The following section will go into considerable detail.

Base Simulation Behavior

The following description of the simulation is based on one vehicle's interaction with one RSU. When the simulation runs on a large scale, the logic is the same, just with interactions and delays occurring in larger quantities.

Once the simulation begins, both the vehicle and RSU run initialization functions - initializing specific variables such as assigning a PSID and setting counters to zero (not a fully comprehensive list). The initialization function is essential for the RSU as it establishes a connection to SUMO through a protocol that enables communication through the Traffic Control

Interface (TraCI) [70]. The RSU also sets data parameters and specifies using the IEEE 1609.4 standard for Wireless Access in Vehicular Environments (WAVE), which defines MAC and PHY layers for communication [71]. After the initialization, the RSU sends a self-message to prompt a beacon to be sent out. After receiving this self-message, the RSU creates and fills a control message (beacon) to be sent out while also specifying another self-message to be sent after 0.5 seconds (the beacon interval).

To send a wireless beacon out into the environment, the RSU creates a WAVE service message upon receiving its self-message. Once the message is made, it must be populated with various information, including identifying information and relevant directional information. These messages are specified to be sent on the IEEE 802.11 p defined channel 178, generally reserved for safety-critical control messages [72]. Since the RSU has not contacted a vehicle at this point and there are no reservations or delays to report, these initial beacons act more as a "hello" message versus a service message.

As a vehicle enters the RSU's communication radius, it will encounter a beacon and read the message's contents. The vehicle's first function is to record the PSID of the RSU from which it received a message. The vehicle will perform this every time a message is received and store a running RSU PSID list of recent beacons received. At the beginning of the simulation, this function simply records the PSID, but as the simulation progresses, the functionality increases, further detailed later. The vehicle will then iterate through the adjustment list sent by the RSU, checking if the vehicle's PSID was included in these adjustments. Since this is the first interaction between the RSU and vehicle, there will be no adjustment for the vehicle, and it subsequently sends itself a different self-message. Upon receiving this self-message, the vehicle will check if it is currently following directions from an RSU. Since this is the first interaction, it

will not be. The vehicle will then create a safety message to request a reservation for the intersection that it is approaching. The message to be sent is populated by the vehicle, including the time it sends the message, its current position, acceleration, distance, desired turning movement at the intersection, PSID, the first critical point at the intersection, and current speed. The information included in this message is crucial for proper intersection operation. The following paragraph will detail how this information is found and calculated.

The first piece of information, the time the vehicle sends the message, is reported directly by the OMNeT++ simulation. Next, utilizing the TraCI communication, the OMNeT++ simulation can report the vehicle's current position and direction through the SUMO simulation and report it back in OMNeT++ units.

The second user-defined function called in populating the vehicle's message is the route conversion function. This function uses a vehicle's direction, current road ID, and future road ID to perform calculations and determine the desired turning movement of the vehicle. The direction is reported to OMNeT++ by SUMO through the TraCI communications, as is a vector listing the road IDs on a vehicle's route. During initialization, an integer variable named "futureIndex" is created and set to a value of 1. When the route conversion function is called, the value of the street ID vector is indexed at 1 (0 is the first street), returning the value of the second street in the route, representing the receiving lane. Once a vehicle interacts with a new RSU, the previously mentioned RSU identification function will be called and add 2 to this index. The next time the route conversion function is called, it will return the fourth street in the vector, representing the receiving road of the second intersection. This process is repeated for every new RSU the vehicle encounters. The approach street at the intersection is reported through the TraCI protocol. The

names of the roads were chosen with integer values to use the street names in calculations to register the desired turning movement. An example intersection is shown below in Figure 6.

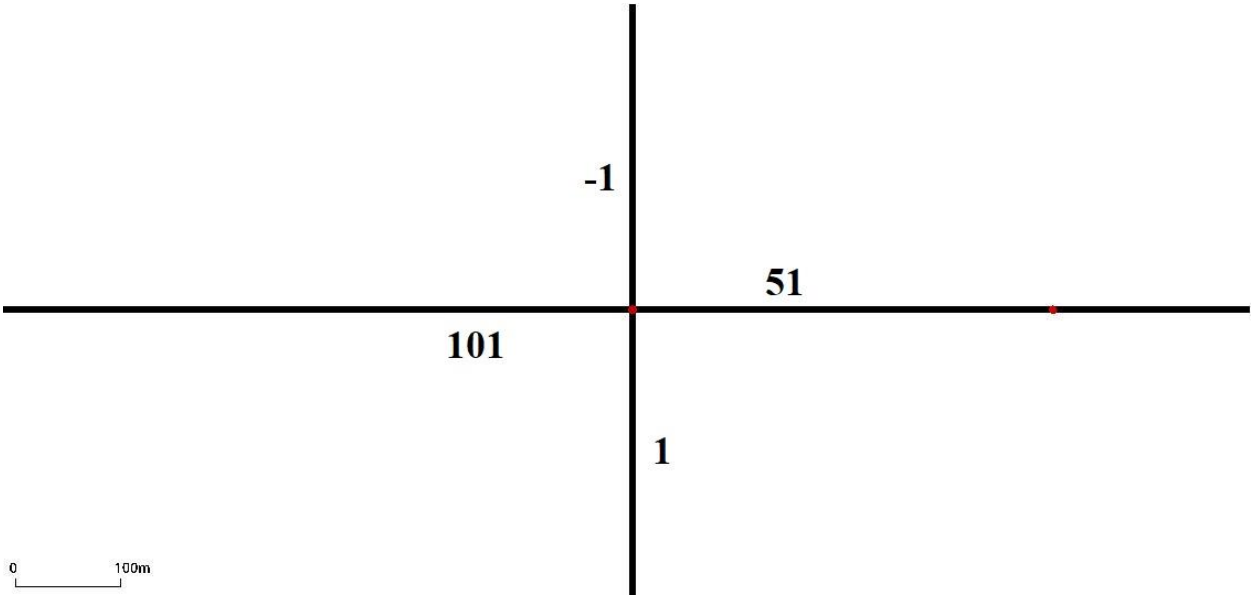


Figure 6 - Intersection Approaches - Labeled

With the approach and receiving road names identified, simple addition or subtraction is done. The result of this arithmetic, paired with the vehicle's travel direction, is used by the function and returns an integer that correlates to a specific turn movement (NBR, SBL, WBT, etc.). An example of the logic used is shown below in Table 1, using the northbound approach.

Table 1 - Sample Turn Movement Logic and Calculation

Travel Direction	Approach Road ID	Receiving Road ID	Arithmetic	Result	Turning Movement
North (0°)	1	101	Receiving - Approach	100	NB Left
North (0°)	1	-1	Receiving + Approach	0	NB Through
North (0°)	1	51	Receiving - Approach	50	NB Right

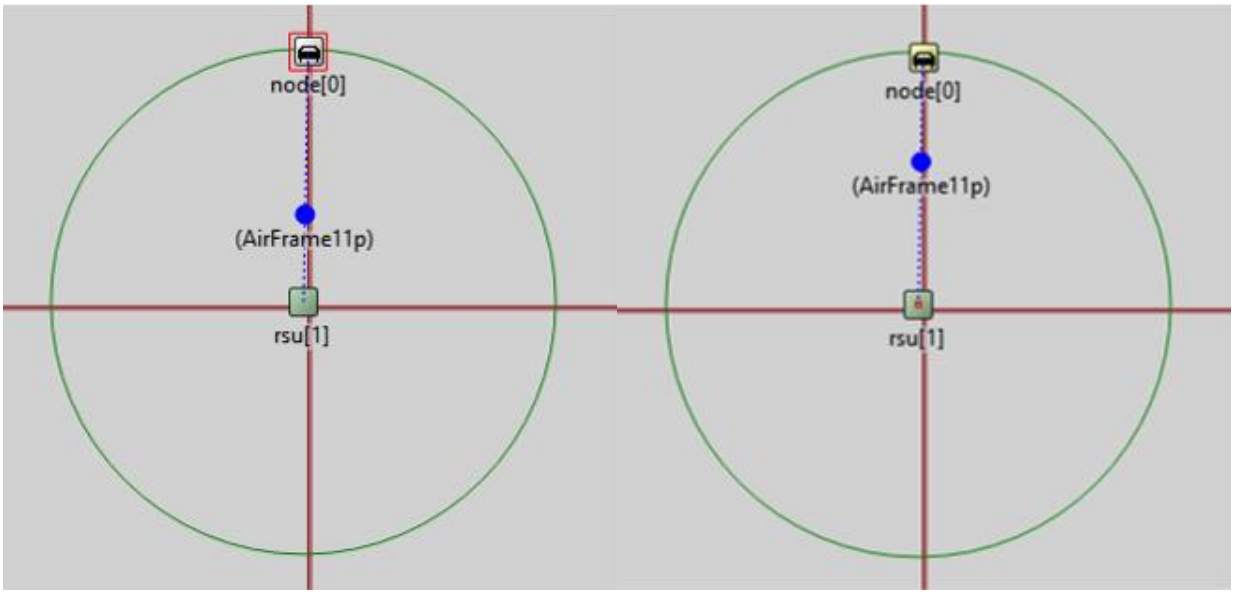
For each of the remaining three travel directions, the arithmetic may vary, but the overall logic of using road IDs to calculate the desired turning movement remains the same.

The next piece of information added to the message is the vehicle's PSID, which is assigned randomly at initialization and simply stored and recalled throughout the simulation.

To calculate the earliest possible arrival time (under non-delayed travel conditions) for a vehicle at an intersection, the RSU must know the coordinates of the first critical point along the vehicle's path through the intersection. The reservation system for the RSU is based on a series of defined critical points that define possible paths through the intersection. This system is detailed further in a later section. The vehicle performs a user-defined function to calculate the first critical point, then adds the information to the safety message to be sent out to the RSU. The get first point function does not take in any arguments; instead, it uses the TraCI mobility capabilities to retrieve the vehicle's coordinates at the current simulation time. With the location known, the function divides the x and y coordinates by 800 and rounds down to create x and y factor variables. These variables determine how many streets away, in both directions, the current vehicle is from the origin intersection (the intersection with the lowest x and y coordinates). Since each intersection is a multiple of 800 meters apart from the origin intersection, using travel direction, x-factor, and y-factor, the vehicle can calculate which intersection it is approaching and thus the coordinates of the first critical point before an intersection. The vehicle then takes this information and adds it to its message to be sent to the RSU.

The final user-defined function used is the fastest arrival function. This function calculates the fastest arrival time for the vehicle to reach the first critical point if the vehicle is traveling below the route velocity. The time reported in this function is generally only used by the RSU if the intersection is recovering from a crash and the vehicles arriving have been slowed by the delay. Equations 6.0 - 6.3 are used to calculate this time**.**

The last information to be included in the vehicle's message to the RSU is the vehicle's current speed. The speed is found using the TraCI mobility communications and is used in the earliest arrival time calculations mentioned above. Once all of the information is filled in for the message, the vehicle sends out the safety message on channel 178 to the RSU and other vehicles nearby. The other vehicles ignore this information while the RSU takes the data for reservations and calculations. Once this message is sent, the vehicle turns yellow, indicating that it has sent a request for a reservation and is waiting for a response from the RSU. This initial interaction is shown in Figure 7.



(a) (b)

Figure 7 - (a) Initial RSU Beacon Sent (b) Vehicle Sending Intersection Request to RSU

Once the RSU receives the message from a vehicle, it adds the message to a vector of safety messages and notes that a reservation needs to be made. The next time the RSU receives a self-message, it will populate the message similar to previously described, but this time it will perform an extra function since a reservation needs to be made. This extra function, the update pathing function, is the main driver for the RSU's ability to control the intersection. The function

is iterative, meaning that it loops through all of the received safety messages and performs the same process for each.

Before performing reservation calculations, the RSU performs two checks to ensure that reservations are made correctly. The first function takes the vehicle's distance from the RSU and sorts the requests. This sorting function ensures that the RSU is making reservations in the first in, first out framework that is desired. Second, the RSU checks the acceleration of the vehicle as it iterates through the received safety messages. The iteration loop will break if a vehicle is performing heavy deceleration. At this point, any reservations already made will be sent out, but any vehicle behind the decelerating vehicle will not receive a reservation. Once the vehicle is no longer decelerating, reservations will be made for itself and vehicles behind it in the queue. This deceleration check is to ensure a vehicle reaches a steady velocity before making a reservation. This instance occurs mainly when a vehicle reaches the end of a vehicle queue when an intersection is recovering from a crash. Once these two checks are complete, the RSU will perform a series of calculations to complete the reservation process.

First, the RSU defines the vehicle's desired turning movement. Using a user-defined function, the RSU notes all of the intersection points that a vehicle will pass through along the route into a single vector. The intersection is made up of sixteen points, shown below in Figure 8, that allow the RSU to make reservations based on the desired path of a vehicle.

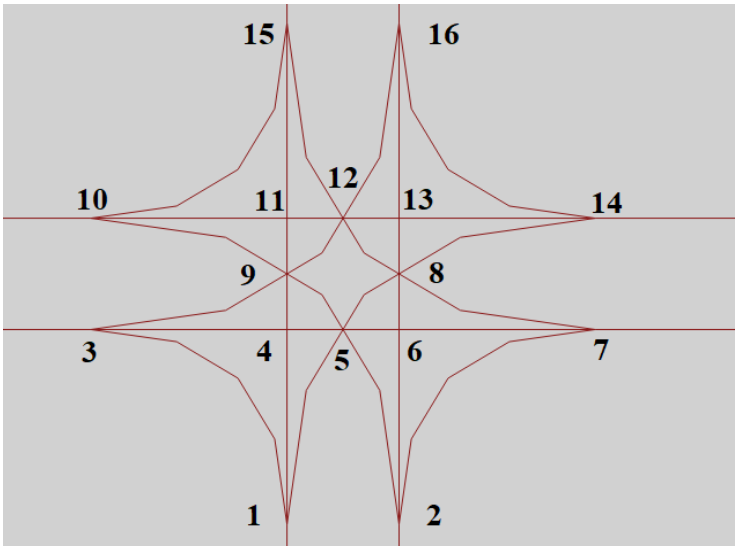


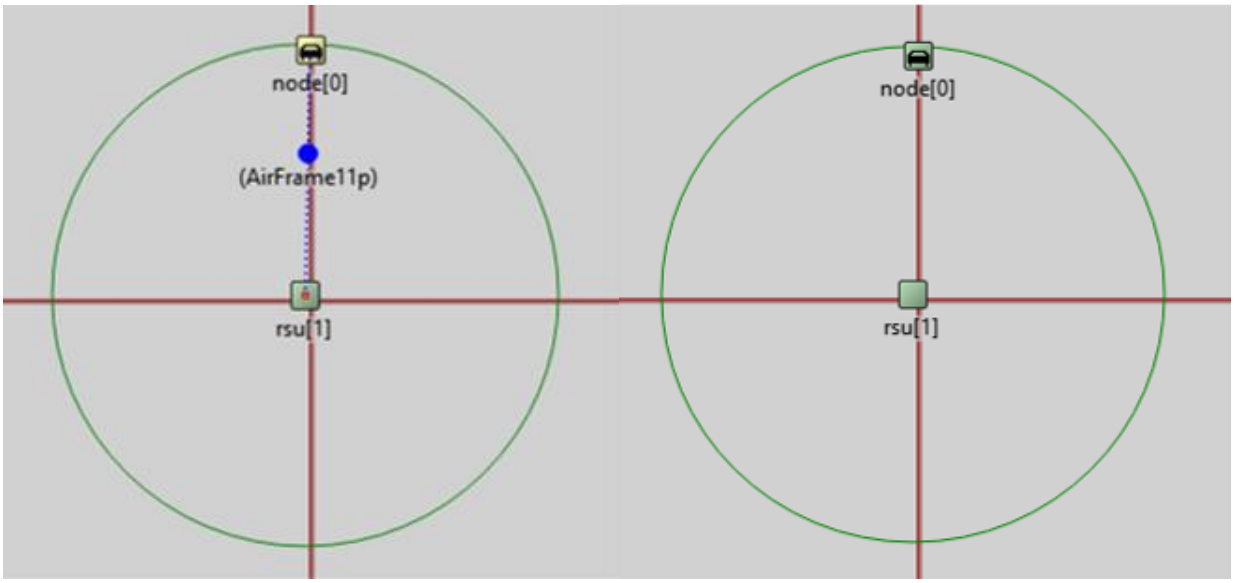
Figure 8 - Intersection Critical Path Points

Based on the figure above, a northbound left turn would pass through points 2, 6, 8, 12, 11, and 10. Once the route has been determined, the RSU creates a series of times representing when a vehicle will enter and leave a point within the intersection and assigns the route a speed based on the turning movement. The RSU then performs the earliest arrival time function mentioned earlier (Equation 2.1), which takes in the first critical point coordinates, the vehicle's location, route speed, and vehicle's speed. If the vehicle is traveling below its route velocity, the RSU will collect the fastest arrival data from the vehicle's safety message. With this information calculated/collected, the RSU then loops through the points along the route and adds the earliest arrival time to the enter and exit times for the vehicle's critical points. The RSU iterates through points along the route and checks for reservations made by previous vehicles. The RSU records the earliest available time for each point. Suppose the earliest arrival time is before (or within ± 2.0 seconds of) the previous reservation. In that case, the RSU knows that it must delay the current vehicle's passage time to not interfere with existing reservations. Each point along the route is checked, and the highest required delay is recorded as a required offset. If there is no need for delay, zero seconds are added for the vehicle.

If a delay is required, the RSU performs a series of checks and adds time to ensure proper clearance between vehicles. It first checks if the constraining reservation was made by a vehicle traveling in the same direction. If so, 1.75 seconds are added to the reservation, with an additional 0.25 seconds if the prior vehicle performs a right turn. Right turning vehicles have the lowest turning velocity, thus requiring extra time for passage. The next check occurs if the vehicle and the constraining reservation are both through moving vehicles. This situation only requires 1.75 added seconds as through movements have the largest velocity. Finally, suppose neither of these two conditions is true. In that case, a blanket delay of three seconds is added to the maximum offset to ensure a smooth and proper passage without the risk of a collision within the intersection. For each point along the vehicle's path, a reservation is made and recorded containing the vehicle's entry time and exit time for that point, along with the vehicle's PSID for identification purposes. After looping through the points, the RSU adds the required delay, total time, and vehicle PSID to a vector. After iterating through the safety messages received by the RSU, it clears the messages received and removes any reservations at points that have already passed in time. With the adjustment vector updated and added to the control message, the RSU adds any other relevant information (PSID and channel), sends the message out to any vehicles nearby, and schedules a self-message to restart the process for any new vehicles entering the radius of communication.

The vehicle receives the message sent from the RSU and checks to confirm that it is a control message. Once the type of message is established, the vehicle checks the identity of the sender's RSU. The vehicle then iterates through the attached adjustments, looking to find adjustments with an attached PSID that matches the vehicle's PSID. Assuming that the RSU sent an adjustment for the vehicle (including a zero-second adjustment), the vehicle will note that it is

following directions from an RSU and will change its color to green (for visual indication). A vehicle receiving a reservation from and beginning to follow its directions is shown in Figure 9.



(a) (b)

Figure 9 - (a) RSU Sending Reservation (b) Vehicle Following RSU's Directions

If the time adjustment is zero and the vehicle is making a through, the function finishes, and the vehicle carries along without any adjustments. A turning vehicle will schedule a self-message to slow down to its route velocity as it enters the intersection using Equation 2.1. If the adjustment is not zero, the vehicle must gather relevant information and perform necessary calculations to ensure clear passage through the intersection. The vehicle performs previously mentioned functions to determine route information (path and speed) and the coordinates of the first critical point along the path. The vehicle calculates the distance between its current position and the first critical point of its course. The adjustment and total time sent by the RSU are also collected from the message for calculations.

All of the following calculations described are derived from standard kinematic motion equations. The complete derivations are shown in Appendix A. The first series of calculations involve defining the maximum delay a vehicle can incur without decelerating past the final

velocity. This maximum delay variable, defined in Equation 1.0, is the difference between the fastest and longest possible times a vehicle could take to reach the first critical point from its current position.

$$t_{\text{Max Delay}} = t_{\text{Longest Arrival}} - t_{\text{Fastest Arrival}} \quad \text{Equation 1.0}$$

The longest arrival time can be logically described as the time it would take a vehicle to reach the first critical point if it were to begin the braking process immediately. Conversely, the fastest arrival time could be described as the time it would take a vehicle to reach the first critical point if it proceeded at the maximum allowed speed until the very last moment when it could safely decelerate to the needed intersection speed. These calculations are dependent on whether the vehicle is required to slow down (route velocity < velocity) or speed up (route velocity > velocity). The derived formulas for each are shown in Equation 2.0 and Equation 2.1, and Equation 3.0 and 3.1, respectively, below.

$$t_{\text{Longest Arrival}} = \frac{Distance}{Vend} + \frac{Vend}{2decel} + \frac{(Vstart)^2}{2decel\ Vend} - \frac{Vstart}{decel} \quad \text{Equation 2.0}$$

$$t_{\text{Fastest Arrival}} = \frac{Distance}{Vstart} - \frac{(Vend)^2}{2decel\ Vstart} - \frac{Vstart}{2decel} + \frac{Vend}{decel} \quad \text{Equation 2.1}$$

The fastest arrival for a vehicle traveling below its route velocity when it receives directions from the RSU has a unique velocity profile. This vehicle could theoretically accelerate past its route velocity (max of 13.4 m/s) and slow down just before reaching the intersection. This velocity profile produces a more complex equation, but the overall logic remains the same.

$$t_{\text{Longest Arrival}} = \frac{Distance}{Vstart} - \frac{Vstart}{2\ accel} - \frac{(Vend)^2}{2accel\ Vstart} + \frac{Vend}{accel} \quad \text{Equation 3.0}$$

$$t_{\text{Fastest Arrival}} = \frac{Vmid}{2accel} - \frac{Vstart}{accel} + \frac{Distance}{Vmid} + \frac{(Vstart)^2}{2accel\ Vmid} - \frac{Vmid}{2\ decel} + \frac{Vend}{decel} \quad \text{Equation 3.1}$$

The maximum delay variable is dependent on the speed the vehicle is traveling and the final speed of the vehicle entering the intersection (the route velocity). Within the simulation, the

vehicle will perform a test to determine whether it will need to accelerate, decelerate, or remain at a constant speed before the intersection. The acceleration will either be set to a maximum deceleration value or max acceleration value if acceleration is required. These values are shown in Equation 2.3 and Equation 3.3, respectively.

$$decel = -4.5\ m/s^2 \quad \text{Equation 2.3}$$

$$accel = 2.6\ m/s^2 \quad \text{Equation 3.3}$$

The next series of calculations are based on a vehicle that enters the communication radius at the maximum speed (13.4 m/s), and its route velocity is also maximum speed (through movement). Since it received a delay command from the RSU, the vehicle needs to introduce delay by slowing down for some time, to a middle velocity, before accelerating back to its intersection speed before arrival at the first critical point. These calculations are also used when a vehicle's calculated maximum delay (Equation 1.0) is less than the delay required by the calculation. This case occurs if the vehicle is forced to introduce delay to extend its travel time to delay its entry into the intersection.

The first step in the calculation process for the vehicle is to calculate the velocity that it needs to decelerate to introduce delay. This formula is shown below in Equation 4.0 with its member functions after (Equations 4.1, 4.2, and 4.3).

$$Velocity = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \quad \text{Equation 4.0}$$

$$a = -0.3034 \quad \text{Equation 4.1}$$

$$b = \frac{Vend}{accel} - \frac{Vstart}{decel} - TotalTime \quad \text{Equation 4.2}$$

$$c = Distance + \frac{(Vstart)^2}{2decel} - \frac{(Vend)^2}{2accel} \quad \text{Equation 4.3}$$

The vehicle decelerates to the desired speed immediately, using the deceleration rate shown in Equation 2.1. Once the vehicle is traveling at its calculated velocity, it must calculate the time it

needs to remain at this speed before accelerating to its intersection velocity. The time is calculated using Equation 4.4 and is the sum of the time the vehicle spent decelerating and the time it should remain at the lower speed.

$$Time\ Accel\ =\ TotalTime - \frac{Vend}{accel} + \frac{Vmid}{accel} \qquad \text{Equation 4.4}$$

The next series of calculations only occurs if the vehicle receives a reservation when traveling slower than its route velocity. The previous velocity profile applies to both vehicles traveling faster or slower than their route velocity. The first possible velocity profile occurs when a vehicle has entered the communication radius and must slow down as it is at the end of a queue of vehicles and must decelerate. Once the vehicle has finished decelerating, this vehicle follows a velocity profile that maintains its speed until it is safe to accelerate to its route velocity. It then maintains its route velocity to arrive at its assigned arrival time. This velocity is calculated using Equations 5.0, 5.1, and 5.2 - shown below:

Equation 5.0

$$T_3 = (Distance\ - Vstart * TotalTime + \frac{Vend*Vbegin}{accel} - \frac{(Vstart)^2}{2*accel} - \frac{(Vend)^2}{2*accel}) / (Vend - Vstart)$$

$$T_2 = \frac{Vend - Vstart}{accel} \qquad \text{Equation 5.1}$$

$$T_1\ =\ TotalTime\ -\ T_2 - T_3 \qquad \text{Equation 5.2}$$

In this scenario, the vehicle will travel at its beginning speed for *T1* seconds, accelerate for *T2* seconds, then maintain its route velocity for *T3* seconds until it passes the intersection where it can accelerate to its maximum speed (if needed).

The next velocity profile is valid for vehicles that receive their reservation when they are traveling below the route velocity, and there is enough time and space to accelerate past (or equal to) its needed velocity prior to decelerating as it reaches the intersection. This velocity profile is

based on the vehicle's earliest arrival time shown in Equation 3.1 above. The calculations are based on Equation 4.0 with member functions 6.1, 6.2, and 6.3.

$$Velocity = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \quad \text{Equation 4.0}$$

$$a = 0.3034 \quad \text{Equation 6.1}$$

$$b = -\frac{Vend}{decel} - \frac{Vstart}{accel} - TotalTime \quad \text{Equation 6.2}$$

$$c = Distance + \frac{(Vstart)^2}{2accel} - \frac{(Vend)^2}{2decel} \quad \text{Equation 6.3}$$

In this profile, the vehicle will immediately accelerate to its directed velocity then maintain this velocity until it has reached the calculated time to decelerate, shown in Equation 6.4.

$$Time\ Decel = TotalTime - \frac{Vend}{decel} + \frac{Vmid}{decel} \quad \text{Equation 6.4}$$

The final velocity profile for a vehicle traveling below its route velocity occurs when the vehicle can speed up but not up to or past its route velocity. Meaning the vehicle will have to accelerate to a middle velocity, maintain that velocity, then accelerate again as it approaches the intersection to reach its route velocity. This velocity calculation is shown in Equation 6.5.

$$Velocity = \left(Distance - \frac{(Vstart)^2}{2*accel} - \frac{(Vend)^2}{2*accel}\right) / \left(TotalTime + \frac{Vstart}{accel} - \frac{Vend}{accel}\right) \quad \text{Equation 6.5}$$

Similar to previous profiles, the vehicle will accelerate immediately and maintain velocity until it reaches the time to accelerate calculated in Equation 6.6.

$$Time\ Accel = TotalTime - \frac{Vend}{accel} + \frac{Vmid}{accel} \quad \text{Equation 6.6}$$

The final main velocity is the inverse of the previous one. It occurs when a vehicle needs to decelerate to a middle velocity, maintain the velocity, then decelerate once again before reaching the intersection. This velocity is calculated using Equation 7.0.

$$Velocity = (Distance + \frac{(Vstart)^2}{2*decel} - \frac{(Vend)^2}{2*decel}) / (TotalTime + \frac{Vstart}{decel} - \frac{Vend}{decel})$$ Equation 7.0

The vehicle will immediately decelerate, maintain its directed velocity, then decelerate again to reach the intersection on time. This time to decelerate is calculated with Equation 7.1.

$$Time\ Decel = TotalTime - \frac{Vend}{decel} + \frac{Vmid}{decel}$$ Equation 7.1

There exist two equations for vehicles to use in case of a crash where there is not enough time or space to make a proper speed adjustment to a reservation. These equations are included in Appendix A as Equation 8.0 and Equation 8.1, respectively. Due to their extremely limited use in the simulations, they were not included in this chapter for brevity.

Once the vehicle has reached its appropriate time to accelerate or decelerate, it will send itself a self-message to speed up to the intersection velocity. After receiving the self-message, the vehicle will determine the message not to be a safety message and check to see if it needs to accelerate. In this case, the vehicle needs to accelerate to match the intersection speed. The vehicle will either speed up or slow down to match the intersection speed based on its current speed. After achieving the desired speed, the vehicle will send another self-message after navigating the intersection. The vehicle will receive this message and accelerate to the maximum speed after exiting the intersection.

After exiting the intersection, the vehicle will continue to travel along its route, receiving messages from the RSU until it leaves the communication radius. These messages are ignored by the vehicle as they do not contain any relevant information for the vehicle. Once a vehicle reaches a new RSU's communication radius, the vehicle will perform the critical RSU identification function that re-initializes the vehicle as it approaches a new intersection. The function checks the RSID of the new RSU. Since it will be a different identity from the previously received messages, the vehicle will stop following the command from the previous

RSU. This adjustment will allow the vehicle to communicate with the new RSU and perform the cycle described above to make reservations for the new intersection. The vehicle will also change color to indicate that it is waiting for a reservation. Finally, the vehicle will change the vehicle's acceleration indicator back to the default false, so it can accelerate adequately to reach its desired route velocity when the time comes.

Each vehicle in the network will perform these steps with all RSUs it encounters along its route. The overall logic is identical for each RSU, allowing for a scalable network and consistent traffic behavior throughout the built network.

Running the Simulation

The simulation parameters needed to be established before the simulation could be executed. First was run time; the simulation runs for 1500 seconds to simulate 20 minutes of traffic with a 5-minute warmup period to fill in the network. The second was the number of simulations to be performed for each scenario. This number was initially chosen to be five simulations per scenario. Nineteen scenarios were developed and are described in a later section. The vehicles operate at a maximum speed of 13.4 m/s (approximately 30 mph) in free-flow conditions. After a collision, a delay period ensues, the delayed vehicles are slowed down to 2.24 m/s (about 5 mph). This delay speed was selected as it introduces significant delay but prevents gridlock within the system and the possibility of queue spill back into another RSU's radius of communication. This delay period is meant to imitate traffic operations consistent with vehicular crashes in an urban setting. The rationale will be further explained in the following sections.

Attack Model Development

To maintain a high level of control and consistency, both attacks (DoS and MITM) were modeled to show the attack's impact versus the actual method of attack. The model assumes that

the attacker could bypass vehicle authentication and gain the appropriate access to launch an attack. For the DoS attack, the model simulates a safety message from a vehicle to an RSU being dropped based on a probability of an attack succeeding. Similarly, the MITM attacks modeled the delay of acceleration/deceleration messages based on a likelihood of attack success. The actual attack methods were not simulated due to the scope of the research and to have a higher level of control on the simulation and attack success rates. Both attacks are modeled as static attacks against dynamic mobility - meaning the attacker is not moving, but the victims are.

For both attacks, crashes between vehicles are expected. However, SUMO was designed to simulate a perfect network. SUMO allows the user to hard code what defines a collision and how crashes are handled within the simulation. SUMO reports a collision for both attacks and the base simulation when the gap between vehicles is zero (inferring a collision has occurred). The vehicles come to a stop where they have crashed and are removed from the simulation after one second. Surrounding traffic is delayed following the collision for five minutes - simulating a to-scale traffic response one would expect in a real-world collision environment. The network collision response will be discussed following the attack models.

Denial of Service Attack Model

While there are many different versions of a DoS attack, an attack that drops safety-critical communications from the vehicle to the RSU was chosen. Since the attack's impact is being modeled, the attack is modeled simply within the existing simulation model. The attack occurs within the function that accepts self-messages within a vehicle. When the vehicle reaches a new RSU's communication radius, it will receive an initial message from the RSU and then send out a self-message to populate and send out a reservation request for the upcoming intersection. A drop variable is manually set (between zero and 0.75) depending on the attack

scenario. If a random number generated by the vehicle is less than the drop variable, the attack will occur. For example, if a drop variable is 0.10, there is a 10% chance that a DoS attack will occur. If the random number is less than 0.10, the attack will occur. Otherwise, the simulation will carry on as usual.

If the attack occurs, there are two main functions to represent the impact of the DoS attack. First, the vehicle marks that it is following the directions of an RSU. This change tricks the vehicle into thinking that it received a legitimate message containing intersection directions from an RSU. The second function is to delete the populated self-message, effectively erasing the request from existence. Secondary functions change the color to indicate that the vehicle was a victim of an attack and note how many times the vehicle has been attacked for analysis.

The impact of the attack effectively skips the reservation process between the vehicle and the RSU. Without an attack, the vehicle would send a reservation request to the RSU, the RSU would send back reservation information, and the vehicle would make the relevant calculations. Since the second two of these steps do not occur under the DoS attack, the vehicle effectively blindly enters the intersection, assuming that it is clear, and no delay is needed.

Man in the Middle Attack Model

Similar to DoS attacks, there are many variations of a MITM attack. For this model, the impact of the MITM attack is a delay in the intersection approach process. The attack is performed by randomly delaying the time that it takes a vehicle to send itself an internal message to speed up to the required intersection speed after receiving a reservation from the RSU. The compromised vehicle has an authentic reservation at the intersection but will enter the intersection after its allotted time, depending on the random delay introduced. The delay is a random integer between two and five seconds. The MITM attack is also dependent on a random

number generator and a manually set probability. If the random number is less than the probability, the attack will occur. The compromised vehicle will turn red to indicate it has been attacked. A delayed vehicle indicator is set to yes (discussed later), and the delayed message is counted for tracking purposes. The MITM attack cannot impact a vehicle with no intersection delay as it never sends a self-message to speed up to the intersection speed.

The goal of the MITM is to impact the timing of an intersection through random internal message delays. Even though the MITM attack is more subtle than the DoS attack, significant collisions within the intersection are still expected. The attack would not be noticeable to either the vehicle or the RSU as they both believe the vehicle has made a legitimate reservation.

Collisions in the Simulation

Collisions within intersections are expected to occur when either of the simulated attacks is successful. Since SUMO models a perfect vehicle driving scenario, it has to be told what a crash is within the network. For this simulation, a crash occurs when two vehicles are less than or equal to zero feet apart from each other, meaning that they have collided. Once a collision occurs, the vehicles involved in the crashes are removed from the intersection after one second.

Due to the nature of the simulation, an overwhelming majority of the crashes are angle or turning movement crashes. Victims of DoS attacks approach the intersection blindly, but without a proper reservation from the RSU. This means that the infected vehicles can slow down if there is a slower vehicle in front of it, basically eliminating rear end crashes in a DoS environment. For MITM attacks, rear end crashes are possible but rarely occurred. Vehicles in this attack setting make legitimate reservations and will accelerate to their reservation time regardless of vehicles around them. This introduces the possibility of a rear end crash if a legitimate vehicle is trailing a delayed vehicle, where the delayed vehicle has not yet accelerated due to the attack.

Even with this possibility, most crashes that occurred in the simulation were more severe angle and turning movement crashes.

The crash scenario introduces two large assumptions made for the simulation environment. First, it is assumed that there are little to no vehicle sensors that introduce collision avoidance, and the vehicle is fully autonomous so there is no possibility of driver intervention. This simulation strictly relies on V2I and V2V communication for collision avoidance through intersection reservations. However, during DoS attacks the infected vehicles approach the intersection without direction from the RSU. This fact allows them to avoid a crash if there is enough time and space to properly react. The vehicle's reaction is defined by the driving model in SUMO. MITM attacks do not have this same ability, since the vehicles have legitimate reservations, the infected vehicles act strictly based on the reservations made. The second assumption is that the vehicle is removed soon after the collision occurs. This allows traffic to flow through the intersection. Allowing traffic to flow at a delayed rate is how the simulation simulates delay. The assumption was made to simulate an intersection collision that did not completely obstruct the intersection, allowing a low flow rate of vehicles to pass. The collision response process is described below.

Collision Incident Management Model

The way the surrounding vehicles react to a crash within an intersection is crucial to measure the operational impact of the crash. Delay is modeled within the simulation for a length of five minutes. Five minutes was selected as an appropriate delay time as it simulates the significant increase in delay caused by a crash. Still, it does not overtake a majority of the twenty-minute simulation with delay. There is also a five-minute recovery period, allowing the network to recover without the threat of another attack. The delay time and recovery times were

chosen to simulate consistent collision reaction in the simulation where the crash could be cleared relatively quickly and navigated by passing vehicles. Upon further development of the simulation, the delay and recover could be more refined to further improve the results of the model. The delay mechanism is detailed below.

An internal collision checking message is sent within a vehicle every time it receives a control beacon from an RSU (every 0.5 seconds). The vehicle's velocity is checked upon receiving the internal message. If the speed is equal to zero, a crash has occurred. The only time a vehicle would be at zero velocity is if it was involved in a collision in the simulation. Each vehicle in the crash indicates it has been in a crash, records it in a count, and then sends a new type of message to surrounding vehicles (within the communication radius) and the intersection's RSU. The vehicles and the RSU both process the message but perform unique actions as a result. Each receives two messages, one from each involved vehicle, but only the first is processed. The second message is ignored to prevent duplicate responses.

The RSU receives the collision message and immediately indicates that a crash has occurred and notes the time of the crash. Any messages sent by the RSU during the delay period will notify any new vehicles entering its radius of the current delay. All reservations made at the intersection are then cleared, creating a blank reservations list for when vehicles eventually make reservations during or after the delay. The new reservations will not be impacted by previous reservations whose end time has not passed yet. The final and most important step the RSU takes is to create and schedule a speed-up message. After the five-minute delay period, the speed-up message is sent to vehicles indicating that the delay period is over, and traffic can begin to recover. Before this message, all vehicles operate at the 2.24 m/s delay speed. The RSU must hold this message until after the delay instead of vehicles initially impacted by the crash. These

vehicles could possibly make their way through the intersection during the delay, making it difficult to indicate the end of the delay. The RSU also keeps an internal track of a subsequent five-minute network recovery rate. During the delay and recovery periods, vehicles cannot be compromised through cyberattacks. Additionally, vehicles that may have been compromised but have not reached the intersection have their attack wiped out and begin to operate normally again. These steps prevent a successive chain of collisions during the recovery period of the intersection. After the recovery period, vehicles become subject to cyberattacks again.

The vehicle reacts to the initial crash message in a completely different manner. First, the vehicle checks if it is within 75 meters of the intersection. If so, the vehicle will drop its reservations, mark itself within the intersection, and send a self-message to make a new reservation. All reservations during the delay period have their distance to the intersection factored into them. Factoring the distance ensures that the vehicles closest to the RSU get to make the first reservation. These messages are also sorted based on the distance by the RSU for redundancy. The delay essentially creates a slow-moving queue. If the second vehicle in the line made a reservation before the first, the simulation would throw a runtime error. The second vehicle cannot reach the intersection before the first. The 75-meter threshold allows vehicles approaching the intersection to accelerate to make a reservation and react accordingly to navigate the intersection safely.

If a vehicle receives the initial collision message and is not within 75 meters of the intersection, its first step is to slow down to the delay velocity immediately. To simulate an urban delay, the vehicles slow to 5 mph (2.24 m/s - the delay velocity) until it reaches 75 meters of the intersection, or the delay period ends. A vehicle outside of the distance threshold indicates that it is following directions so as not to receive any new directions from the RSU. It also indicates

that it has been impacted by the delay and changes its color to orange. This vehicle will also be cleared of any previous attack at the intersection and prevented from being attacked by the current RSU. All attacks performed against vehicles impacted by the initial delay are cleared. Removing a prior attack in a delay situation prevents a cascading series of collisions resulting from the initial collision. A vehicle attacked during the delay would almost certainly create another crash. The second crash would create an infinite loop of collision, recovery, then immediately another collision. As previously stated, vehicles are also immune to attacks during the recovery phase as well. The vehicle is susceptible to attack once it reaches a new RSU or the recovery period ends.

After the crash, the RSU continues to send beacons to vehicles every 0.5 seconds. During the delay period, these messages do not make reservations, for the most part, and act as refresher messages for the vehicles receiving them. Upon receiving a message from the RSU, a vehicle will check if it is within 75 meters of the intersection. If so, the vehicle will make a reservation and react as it usually would. The vehicle is effectively moving past the delay and carrying on along its route. If the vehicle is not within 75 meters, the next step is to check if the vehicle has just entered the communication radius. If so, the vehicle will slow down to the delay velocity and indicate the impact of the delay, becoming the last vehicle in the queue. Next, the vehicle checks if it is in the recovery phase. Again, if it is, it will not be subject to a cyberattack. During the recovery period, vehicles make reservations as usual but are immune to attacks. The final aspect of the delay-specific vehicle refreshers is to check if the message received from the RSU contains an emergency indicator. The RSU sends a single emergency speed-up message after the delay period has ended to indicate that vehicles can now make reservations without restriction.

The delay-specific reservation process is slightly different from the normal process. The main difference is the vehicle's message to make a reservation and how this reservation is prompted. When the delay period is over, the RSU sends out the emergency speed-up message to vehicles within its radius. Vehicles within 75 meters are exempt as they are already following directions. The remaining vehicles, all of which are impacted by the delay, receive and process the speed-up message. Vehicles send an internal speed-up message. The internal messages are delayed by a small distance-based factor, ensuring that vehicles in the front of the queue receive reservations first. They also set an indicator that they can now speed up - dependent on their impending reservation. Once the vehicle's internal message is received, it sends out a new external message to the RSU. The RSU collects all of the messages, sorts the messages by distance (redundancy to ensure proper order), adds the reservations to a message, and sends it to the vehicles to make necessary adjustment calculations. The RSU adds that this message is an "emergency" message and contains reservations for the delayed vehicles. Each vehicle receives the message and notes that it includes the special "emergency" reservations. This indication allows the vehicles to take in the message and perform reservation calculations as they usually would. If a vehicle is left off of the RSU's delay response due to reservation size constraints (10 reservations), it sends a self-message. This self-message prompts a request to the RSU for a reservation as it usually would. Any vehicle that received the RSU's speed-up message is not subject to further attacks, and subsequent vehicles will not be until after the recovery period. This immune period ensures that a new attack is only performed against a fully operational intersection, preventing a cascading impact.

Analysis Methodology

The overall analysis methodology was to develop an experimental plan that conveys how the actual simulation and results were set up and processed. The simulation environment was set up to emphasize the attacks and their results to identify the impact of the attacks. After the simulations were run, proper data analysis was conducted to analyze the outcomes of the experiment and their possible real-world consequences.

Experimental plan

Nineteen unique scenarios were developed before testing to measure the impact of a wide variety of attack situations. Four independent variables were created to form the scenarios: attack type, the chance of attack success, number of RSUs attacked, and ID of RSUs attacked. The orientation of RSUs attacked was randomly selected using a random number generator. The initial random orientation was consistent within all runs in each scenario. Each scenario was run five times to ensure an adequate sample size. The scenarios are shown below in Table 2.

Table 2 - Unique Attack Scenarios

Attack	Scenario	Scenario Group	# of Infected RSUs	Attack Success Rate %	Infected RSU(s)	Runs	Total Runs
None	0	0	--	--	--	5	5
DoS Attack	1	1	1	25%	7	5	10
	2			50%	1	5	15
	3			75%	4	5	20
	4	2	6	25%	4, 8, 9, 11, 16, 17	5	25
	5			50%	0, 5, 8, 9, 10, 15	5	30
	6			75%	0, 1, 4, 5, 13, 14	5	35
	7	3	12	25%	0, 1, 2, 3, 6, 7, 8, 10, 11, 12, 15, 17	5	40
	8			50%	0, 2, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16	5	45
	9			75%	0, 1, 2, 6, 8, 9, 10, 11, 12, 13, 14, 15	5	50
MITM Attack	10	4	1	25%	8	5	55
	11			50%	9	5	60
	12			75%	10	5	65
	13	5	6	25%	1, 4, 7, 9, 11, 14	5	70
	14			50%	1, 4, 10, 11, 12, 14	5	75
	15			75%	1, 6, 10, 14, 15, 16	5	80
	16	6	12	25%	0, 2, 5, 6, 7, 9, 10, 13, 14, 15, 16, 17	5	85
	17			50%	0, 1, 3, 4, 5, 7, 8, 11, 13, 14, 15, 17	5	90
	18			75%	1, 2, 3, 4, 6, 9, 10, 11, 12, 15, 16, 17	5	95

Measures of Effectiveness

The overall goal of this research was to measure the impact of cyberattacks on the transportation network. Measures of effectiveness had to be established to track between scenarios to quantify the results. The three main transportation MOEs collected were travel delay per vehicle, number of crashes, and system throughput (vehicles). Delay was calculated as the

average time spent in the simulation per vehicle per run. Vehicles spend more time within the network if a delay impacts them than if it did not. The difference in travel time between each attack scenario and the base scenario would indicate the delay incurred by vehicles due to collisions.

A vehicle involved in a collision noted the collision for data collection. Following the conclusion of the simulation, the total number of vehicles involved in a crash and the total number of crashes could be summarized.

Finally, network throughput could be measured as the number of vehicles in the network at a specific time. Similar to the delay calculation, as collisions impact more vehicles, more vehicles will remain in the network. The difference between the base case throughput can be compared to each scenario to determine the reduction in network efficiency.

These three MOEs serve to quantify the impact on the safety and efficiency of the different cyberattack scenarios. It is important to note that travel time and throughput data was only analyzed for vehicles that had exited the simulation at the conclusion of the simulation. This was done to prevent a skew from vehicles entering the network right before the end of the simulation.

The simulation also recorded and output the total number of successful cyberattacks and the total number of infected vehicles traveling through intersections. The difference in these values lies in the fact that if a delay impacted an infected vehicle, it reset its attack status and became immune to further attacks at the intersection. If the status was reset, the vehicle would not be compromised as it passes through the intersection. Both of these values are important to represent how widespread cyberattacks can be in the different simulation scenarios.

Finally, visualizations for each scenario were created using the average speed per vehicle per run. This value was calculated using the average speed of each vehicle in the network at each timestep. The analysis provided a visualization of the impact an attack has on the speed and implied efficiency of the network.

To ensure data validity and track statistical significance, minimum required number of runs and variance were calculated for each scenario. T-tests and Chi-Squared tests were performed to compare the data's average and variance against the base scenario. A 95% confidence interval was used for all appropriate calculations.

The validity of the results were tested using the Federal Highway Administration's equation (Equation 9.0) [73] - the minimum number of scenario repetitions were calculated.

$$n_{minimum} = \left(\frac{t_{n-1,95\%} S}{ex}\right)^2 \qquad \text{Equation 9.0}$$

where,

$n_{minimum}, n$ is the minimum required scenario runs, and the initial number of runs
$t_{n-1,95\%}$ is the *t* stat for *n-1* degrees of freedom for the 95% confidence interval
s, x is the standard deviation and mean of the initial runs
e is the tolerance error

The calculations were performed using a standard 95% confidence interval to ensure validity. The assumed minimum number of simulation repetitions was five for all scenarios. All test scenario's results were analyzed and all calculations resulted in a minimum of five repetitions for each scenario. No additional repetitions were required for any of the scenarios. These calculations show that an adequate number of repetitions were performed for each scenario resulting in valid and repeatable data.

Data Analysis Methods

OMNeT++ produces two types of data: scalar and vector. Scalar data is recorded throughout the simulation for each vehicle and recorded as a summary value. For example, a

vehicle's time spent within the simulation network would be a scalar value. On the other hand, vector data is recorded as a value at every time step a vehicle is in the network. An example of this would be a vehicle's speed. Vehicle speed is recorded at every time step and paired with the time step at which it was recorded. The simulation used a time step value of 0.01 seconds.

Two analysis tools were used to process the results of the simulation. The python-based software Jupyter Notebook was used for vector analysis, while Microsoft Excel's pivot tables were used for the scalar outputs. Jupyter Notebook was used for its vector processing power. The application incorporates various processing and graphing programs to perform robust data analysis and visually present the results. Pivot tables were used for scalar values as they provide powerful summary functions suited ideally for the analysis needed for the simulation.

Scope

The scope of this research is important to understand to properly digest the results and analysis. The research is meant to provide a baseline scenario for attacks in a connected vehicle environment. Scenario 0, the base scenario, provides an environment where communication and vehicles behave in a manner expected with the implementation of CAVs and RSUs. This simulation incorporates many assumptions and limitations, mostly based on a connected environment that does not exist yet. Simulating the two attack scenarios presents a baseline, worst-case scenario for these cyberattacks. The impact of these attacks was quantified through a variety of transportation safety and efficiency measures of effectiveness. The results of the simulations should present a clear idea to decision makers of what the consequences would be of implementing V2V and V2I communication without proper resilience countermeasures. The simulation only incorporates communication capabilities and does not use other technologies

such as vehicle sensors during the simulation. This simulation is not meant to model cybersecurity countermeasures or study the impact of the implementation of CAVs.

Limitations

Since this simulation was developed to model a baseline scenario in a CAV environment, it has a number of limitations, mostly based on the assumptions described in previous sections and the inherent uncertainty of the implementation of CAVs. One of the main limitations lies in fact that no countermeasures or direct collision avoidance measures were implemented. This first limitation highlights the baseline nature of the experiment; however, the limitation produces a higher rate of crashes and delay and needs to be developed more for more accurate results. The second large limitation lies in the simulation's modeling of overall CAV behavior. Since the technology has not been largely implemented yet, it is nearly impossible to know exactly what some aspects of the service will look like. For this research, collision avoidance and collision response fall under this limitation. These types of assumptions were made based on previous research and general consensus among experts. The final main limitation lies in the collision response and delay during the simulation. The response is consistent and controlled for each crash, which does not properly simulate the exact nature of traffic flow. The nature of this limitation allows for easy comparisons to be made, but moving forward could be adapted to provide a more realistic crash response.

4. Results

Overview

After running the simulation a total of 95 times, nearly 600 GB of raw data was produced for analysis. As mentioned in a previous section, vehicle throughput, vehicle delay, crashes, total attacks, and effective attacks were the main points of analysis. The raw data for each run of each

scenario is presented below. In the Data Analysis and Discussion section, the proper analysis will be performed. This chapter exists to present the raw data before any discussion.

Base Scenario Results

Scenario 0 – No Attack

Table 3 - Scalar Results - Scenario 0

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,048	326.2	0	0	0
2	1,046	325.6	0	0	0
3	1,049	326.0	0	0	0
4	1,047	325.8	0	0	0
5	1,048	326.1	0	0	0
Average	1,048	325.9	0	0	0

Scenario 0 represents the base scenario for other scenarios to be compared. This scenario represents a fully functional CAV environment with vehicles interacting with RSUs with no risk of cyberattack. With the simulation environment set, the simulation emulates a perfect scenario where there is not a possibility of crashes, and the network operates at maximum efficiency.

Denial of Service Attack Results

Scenario 1 – DoS Attack – 1 RSU – 25% Success Rate

Table 4 - Attack Scenario 1

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
1	DoS	25%	1	7	5

Table 5 - Scalar Results - Scenario 1

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,046	333.4	1	35	32
2	1,042	331.0	2	43	42
3	1,034	332.0	2	16	14
4	1,047	332.4	1	28	26
5	1,036	332.3	2	18	15
Average	1,041	332.2	2	28	26

Scenario 1 represents the first scenario where a cyberattack is being conducted against the network. The DoS attack is performed against RSU #7 with a 25% success rate. RSU #7 represents an internal intersection - meaning that traffic neither originates nor terminates at this RSU. The scenario resulted in an average of two crashes per run and a decrease in overall efficiency from the base scenario. An average of twenty-six effective attacks were successful at this intersection throughout the scenario. This scenario (along with scenario 10) represents the lowest intensity of a cyberattack within the research.

Scenario 2 – DoS Attack – 1 RSU – 50% Success Rate

Table 6 - Attack Scenario 2

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
2	DoS	50%	1	1	5

Table 7 - Scalar Results - Scenario 2

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,047	333.6	1	65	63
2	1,040	333.7	2	45	38
3	1,038	333.5	2	37	32
4	1,050	333.7	1	61	58
5	1,045	333.5	2	53	43
Average	1,044	333.6	2	52	47

Scenario 2 also represents a DoS attack against a single RSU, but with a 50% chance of attack success. The difference in success rate can be seen in the total attack and net attack results as they are nearly doubled from the previous attack scenario. These attack numbers are dependent on the success rate, but also the RSU selection and relative traffic at the intersection during attack times. This attack scenario attacks RSU #1 which is an edge RSU - meaning it has origin and destination traffic as well as pass-through traffic. The throughput of the network slightly increased from the previous scenario with a small increase in delay, and the same number of crashes.

Scenario 3 – DoS Attack – 1 RSU – 75% Success Rate

Table 8 - Attack Scenario 3

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
3	DoS	75%	1	4	5

Table 9 - Scalar Results - Scenario 3

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,033	332.4	2	53	44
2	1,033	332.8	2	52	43
3	1,034	335.2	2	20	7
4	1,051	330.6	2	87	81
5	1,046	331.1	2	79	70
Average	1,039	332.4	2	58	49

Scenario 3 is the last member of the first grouping of scenarios with a single compromised RSU and a 75% attack success rate. The attack was performed against RSU #4, which shares the edge characteristics of the previous scenario. While the success rate of this scenario was three times higher than scenario 1, the number of attacks did not increase at the same rate. Scenario 3 had the lowest throughput of the first group of attacks, but only had the second highest delay. This scenario averaged two crashes per run, equaling previous averages.

Scenario 4 – DoS Attack – 6 RSUs – 25% Success Rate

Table 10 - Attack Scenario 4

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
4	DoS	25%	6	4, 8, 9, 11, 16, 17	5

Table 11 - Scalar Results - Scenario 4

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,019	354.2	7	166	151
2	1,010	356.9	7	159	150
3	1,034	347.0	7	141	135
4	1,032	357.8	9	134	129
5	1,029	356.4	8	159	150
Average	1,025	354.5	8	152	143

Scenario 4 represents the first major jump in the number of infected RSUs for the DoS attacks, going from one to six. The infected RSUs were randomly selected, however in this scenario all of the, except RSU #8, fall on the eastern half of the network. RSU #4 is the only isolated intersection in this scenario as RSU #8 and #9 share a connection and RSUs #11 and #16 both share a connection with RSU #17. The number of attacks represent a nearly 5.5x increase from scenario 1, which shares an attack success rate. The operational and safety impacts from the increase in RSUs is clear from looking at the average data compared to the first scenario group.

Scenario 5 – DoS Attack – 6 RSUs – 50% Success Rate

Table 12 - Attack Scenario 5

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
5	DoS	50%	6	0, 5, 8, 9, 10, 15	5

Table 13 - Scalar Results - Scenario 5

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	994	366.2	10	193	160
2	1,004	354.0	10	303	266
3	1,025	363.3	11	278	250
4	1,025	352.0	7	397	376
5	1,014	358.4	11	247	221
Average	1,012	358.8	10	284	255

Attack scenario 5 is the second scenario in scenario group two with six infected RSUs operating with a 50% chance of attack success. RSU #0, #5, and #15 are on the edge of the simulation environment, with RSU #0 and #5 representing corner intersections. The results show a steady increase in successful attacks from the previous scenario which is expected as the attack success rate increases. The operational efficiency decreased from the previous scenario as the average number of crashes increased.

Scenario 6 – DoS Attack – 6 RSUs – 75% Success Rate

Table 14 - Attack Scenario 6

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
6	DoS	75%	6	0, 1, 4, 5, 13, 14	5

Table 15 - Scalar Results - Scenario 6

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	967	370.6	11	199	165
2	992	365.5	9	334	299
3	997	367.1	10	310	276
4	988	366.6	10	259	215
5	984	369.4	12	234	197
Average	986	367.8	10	267	230

Scenario 6 is the last of the second grouping of attacks with six compromised RSUs operating at a 75% attack success rate. All of the infected RSUs are on the edge of the simulation environment, with RSUs #0 and #5 acting as the northwest and northeast corners. The average number of attacks decreased from the previous intersection while the operational efficiency decreased. This trend implies that more vehicles were impacted by a crash delay and were immune to cyberattacks during the subsequent delay and recovery periods. The average crash total remained steady from the previous scenario.

Scenario 7 – DoS Attack – 12 RSUs – 25% Success Rate

Table 16 - Attack Scenario 7

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
7	DoS	25%	12	0, 1, 2, 3, 6, 7, 8, 10, 11, 12, 15, 17	5

Table 17 - Scalar Results - Scenario 7

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	990	379.8	16	340	320
2	961	380.9	15	365	346
3	973	369.8	16	365	344
4	971	370.3	11	386	375
5	977	379.3	16	374	351
Average	974	376.0	15	366	347

Scenario 7 is the first DoS attack scenario with twelve total infected RSUs, each operating at a 25% attack success rate. The compromised intersections included internal, edge, and corner intersections representing the widespread nature of the attack. All of the data measures were significantly impacted by the increase in infected RSUs throughout the network.

Scenario 8 – DoS Attack – 12 RSUs – 50% Success Rate

Table 18 - Attack Scenario 8

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
8	DoS	50%	12	0, 2, 5, 7, 9, 10, 11, 13, 14, 15, 16	5

Table 19 - Scalar Results - Scenario 8

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	951	384.4	19	600	548
2	969	386.7	17	638	585
3	915	386.2	19	552	493
4	947	399.8	19	504	459
5	967	396.7	17	524	474
Average	950	390.8	18	564	512

Scenario 8 represented twelve compromised RSUs operating with a 50% DoS attack success rate. The widespread nature of the attack encompassed the three main intersection categories, internal, edge, and corner. The increase in attack success rate deteriorated the network in all of the main measured data fields. This damage was particularly clear in the number of attacks, representing a sharp increase from the previous scenario.

Scenario 9 – DoS Attack – 12 RSUs – 75% Success Rate

Table 20 - Attack Scenario 9

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
9	DoS	75%	12	0, 1, 2, 6, 8, 9, 10, 11, 12, 13, 14, 15	5

Table 21 - Scalar Results - Scenario 9

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks=
1	955	393.3	21	513	458
2	923	387.1	23	530	482
3	970	389.1	20	548	490
4	952	397.8	18	525	470
5	976	398.5	22	540	485
Average	955	393.2	21	531	477

The last, and most severe, of the DoS attack scenarios is scenario 9. This scenario models twelve infected RSUs, throughout the network, operating with a 75% attack success rate. Delay and crashes were more significant in the previous scenario, although throughput and number of attacks improved. Similar to scenario 6, this implies that more vehicles were impacted by delay throughout the network. During extensive delay and subsequent intersection recovery, the impacted vehicles at an intersection cannot fall victim to the attacks.

Man in the Middle Attack Results

Scenario 10 – MITM Attack – 1 RSU – 25% Success Rate

Table 22 - Attack Scenario 10

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
10	MITM	25%	1	8	5

Table 23 - Scalar Results - Scenario 10

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1032	333.8	2	11	10
2	1048	333.8	1	42	39
3	1047	334.2	1	26	23
4	1040	330.7	1	42	42
5	1030	327.3	1	38	38
Average	1,039	332.0	1	32	30

Scenario 10 is the first, and least severe, of the MITM attacks that were simulated. The scenario models a single compromised RSU with a 25% attack success rate. The victim, RSU #8 is a centrally located internal RSU. The attack resulted in a decrease in efficiency from the base scenario, however it only averaged one crash per scenario run. Only the first run had a second crash. The comparable DoS attack scenario averaged two crashes per run.

Scenario 11 – MITM Attack – 1 RSU – 50% Success Rate

Table 24 - Attack Scenario 11

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
11	MITM	50%	1	9	5

Table 25 - Scalar Results - Scenario 11

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,45	331.6	1	66	61
2	1,030	330.5	2	41	41
3	1048	331.6	2	61	58
4	1,043	331.2	2	42	38
5	1,048	331.9	1	70	69
Average	1,042	331.4	2	56	53

Scenario 11 modeled a single infected RSU with a 50% attack success rate. Similar to the previous scenario, the victim RSU was centrally located in the interior of the network. The overall operational results were very similar to the previous scenario with a slight increase in

throughput and delay. The average crashes per run doubled to two. The number of successful attacks also significantly increased as the MITM attack success rate rose.

Scenario 12 – MITM Attack – 1 RSU – 75% Success Rate

Table 26 - Attack Scenario 12

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
16	MITM	75%	1	10	5

Table 27 - Scalar Results - Scenario 12

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,038	332.7	2	50	40
2	1,040	331.9	2	50	42
3	1,038	332.7	2	49	40
4	1,038	329.4	1	99	95
5	1,039	334.3	2	28	20
Average	1,039	332.2	2	55	47

The final scenario in the fourth scenario group was scenario 12, modelling a MITM attack against a single RSU with a 75% success rate. The impacted RSU for this scenario was also located at an internal intersection but less centrally located. Both operational metrics were worsened by the increased likelihood of success while the average number of crashes remained the same from the previous scenario. The total number of attacks decreased with the increase in attack success rate, likely caused by more vehicles being protected from attacks by crash delay and intersection recovery times.

Scenario 13 – MITM Attack – 6 RSUs – 25% Success Rate

Table 28 - Attack Scenario 13

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
11	MITM	25%	6	1, 4, 7, 9, 11, 14	5

Table 29 - Scalar Results - Scenario 13

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,032	350.3	8	238	225
2	1,003	361.7	7	162	149
3	1,005	347.4	6	207	196
4	992	349.5	7	200	191
5	1,027	352.8	7	213	194
Average	1,012	352.3	7	204	191

Scenario 13 represents the first significant jump in the number of infected RSUs for the MITM attacks with six total. These RSUs represent both internal and edge intersections on the network. These RSUs were operating with a 25% attack success rate. The results of this simulation show a clear jump in negative safety and operations impacts as a result of adding in more infected RSUs to the system. The number of successful attacks also significantly increased.

Scenario 14 – MITM Attack – 6 RSUs – 50% Success Rate

Table 30 - Attack Scenario 14

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
14	MITM	50%	6	1, 4, 10, 11, 12, 14	5

Table 31 - Scalar Results - Scenario 14

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1,037	361.7	8	308	283
2	1,021	360.7	10	261	236
3	1,025	354.7	8	322	301
4	1,012	361.2	7	310	292
5	1,009	355.9	11	302	273
Average	1,021	358.9	9	301	277

Scenario 14 simulated six infected RSUs operating with a 50% MITM attack success rate. These six infected RSUs encompassed the three main locations of intersections. With the increase in attack success rate, the number of attacks, crashes, and delay all increased. However, this scenario had a higher vehicle throughput than the previous scenario. This outcome represents

the variability of the simulation and how different scenarios can produce unexpected results depending on internal simulation factors.

Scenario 15 – MITM Attack – 6 RSUs – 75% Success Rate

Table 32 - Attack Scenario 15

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
15	MITM	75%	6	1, 6, 10, 14, 15, 16	5

Table 33 - Scalar Results - Scenario 15

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	993	367.8	8	335	294
2	979	366.2	9	344	300
3	1,002	366.2	10	414	377
4	988	364.9	10	343	299
5	1,000	366.7	10	422	374
Average	992	366.4	9	372	329

The last scenario in the fifth scenario grouping is scenario 15, representing six compromised RSUs operating at a 75% MITM attack success rate. The orientation of the impacted RSUs represent internal and edge intersections. The increase in the attack success rate resulted in an increased negative impact on all recorded measures from the previous scenario except for crashes. This trend shows that the same number of average crashes in a network can impact the network operations as a result of differences between variables like traffic volume and RSU placement.

Scenario 16 – MITM Attack – 12 RSUs – 25% Success Rate

Table 34 - Attack Scenario 16

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
16	DoS	25%	12	0, 2, 5, 6, 7, 9, 10, 13, 14, 15, 16, 17	5

Table 35 - Scalar Results - Scenario 16

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	1002	375.0	15	398	380
2	969	364.6	12	460	441
3	1003	369.6	14	469	449
4	986	376.2	11	408	396
5	980	366.3	14	421	394
Average	988	370.3	13	431	412

Scenario 16 is the first scenario in the last grouping, representing an increase in infected RSUs to twelve and operating with a 25% MITM attack success rate. The widespread attack included intersections from all three categories. The increase in the number of RSUs resulted in decreased safety and efficiency with a slight increase in number of attacks from the previous scenario. The slight increase is likely due to the increase in crashes, delay, and recovery and corresponding vehicle protections.

Scenario 17 – MITM Attack – 12 RSUs – 50% Success Rate

Table 36 - Attack Scenario 17

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
17	MITM	50%	12	0, 1, 3, 4, 5, 7, 8, 11, 13, 14, 15, 17	5

Table 37 - Scalar Results - Scenario 17

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	959	385.2	18	639	603
2	959	382.7	20	675	636
3	965	373.5	16	800	770
4	980	391.4	19	564	513
5	972	388.4	17	677	611
Average	967	384.2	18	671	627

Scenario 17 represents twelve infected RSUs operating at a 50% MITM attack success rate. Impacting all three intersection types, the attack scenario produced significant negative

impacts on operation and safety, even including a dramatic increase in successful attacks. The large increase in attacks goes against the general observed trend mentioned in previous scenarios.

Scenario 18 – MITM Attack – 12 RSUs – 75% Success Rate

Table 38 - Attack Scenario 18

Scenario	Attack	Chance of Success	# of RSUs	Impacted RSU(s)	Runs
18	MITM	75%	12	1, 2, 3, 4, 6, 9, 10, 11, 12, 15, 16, 17	5

Table 39 - Scalar Results - Scenario 18

Run #	Total Thru Vehicles	Sim Time/Veh (s)	Crashes	Total Attacks	Net Attacks
1	965	396.4	18	750	679
2	951	396.8	21	660	586
3	944	393.5	18	698	626
4	962	395.9	19	767	681
5	962	396.3	21	666	579
Average	957	395.8	20	708	630

The final scenario of the MITM attacks is scenario 18 with twelve infected RSUs operating with a 75% attack success rate. This scenario represents the worst-case scenario for a MITM attack. The results tend to agree with this, continuing the trend of increasingly severe impacts on traffic operations and safety. The results from this scenario are closely comparable to the worst-case scenario for DoS attacks. Scenario 18 represents the highest number of attacks for any scenario in the simulation, showing a very high success rate and subsequently very significant impacts on the transportation network.

5. Data Analysis and Discussion

Overview

For organization, the task of analysis was divided into three main sections: “DoS Attack Scenarios Analysis”, “MITM Attack Scenarios Analysis”, and “Combined Data Analysis”. The goal of this division was to independently look at the attacks compared to themselves first, then

compare the main takeaways from each attack to each other to understand how the attacks compare.

The main findings from the analysis will be highlighted in the final section of this chapter to reiterate the importance of the outcomes.

Time spent in the simulation per vehicle (delay per vehicle) was the main point of focus for this analysis as a whole. This metric demonstrated how crashes and subsequent delays impacted the travel time of every vehicle in the network. Focusing on this metric gave the best idea to the audience of how the various scenarios impacted the transportation network as a whole. While the other reported variables will be touched on in minor detail, the delay gave the best indication of the network's performance. It was less dependent on probability and timing than the other metrics.

Denial of Service Attack Scenarios Analysis

The first calculations performed were performed to test the variance for each of the scenarios. The variance shows the variability in results per scenario. A higher variance means a broader range of results. Variance in this instance was calculated using the simulation time per vehicle for each run in each scenario. T-tests and Chi-Squared tests were also conducted for all scenarios using average simulation time per scenario, with each being compared to the base scenario results. The T-test was used to analyze the difference in the average results between each scenario and the base, while the Chi-Squared test compared the variance. The results of these tests are shown below in Table 40.

Table 40 - DoS Statistical Analysis

Scenario	Variance	T-Test Results	Chi^2 Test Results
0	0.06	--	--
1	0.74	3.4 E-05	0.962
2	0.01	5.7 E-09	0.924
3	3.23	1.2 E-03	0.953
4	19.15	1.3 E-04	0.013
5	36.12	2.5 E-04	0.002
6	4.40	1.2 E-06	2.01 E-05
7	30.07	3.3 E-05	7.51 E-08
8	48.68	3.1 E-05	2.50 E-13
9	25.82	7.5 E-06	2.73 E-14
Average	**16.80**	**1.8 E-04**	**0.317**

Two main points are clear from Table 40. The first point comes from the T-tests, which compares the mean value for each scenario (using average simulation time per vehicle) and compares it to the base. Every scenario produced a significantly smaller test result than the 95% confidence interval threshold of 0.05. This test shows no overlap between the results of the different scenarios compared to the base scenario.

The second point to be made from this small bit of data is that the variance increases significantly after scenario 3, except for scenario 6. The results of the Chi-Squared tests confirm this as only Scenarios 1, 2, and 3 produced higher test results than the 95% confidence interval threshold. This relationship is significant as it shows that a higher number of infected RSUs and an increased likelihood of success impact the outcome's variance. Scenarios 4, 5, and 6 contain six infected RSUs, and Scenarios 7, 8, and 9 contain twelve. It can also be inferred that the number of infected RSUs has a higher impact on the variance of the results than the likelihood of attack. There is no clear trend among the scenario groupings that shows the likelihood of attack impacts the variance.

Variance in the transportation realm is expected, and the identified result makes sense when put into a modern transportation context. If one were to imagine one reckless driver in a group of cars, the variance of its impact would be expected to be relatively low. If the scenario were different and a more significant proportion of vehicles were being operated by reckless drivers, the outcome would likely be more destructive with a higher degree of chaos.

The variance calculations also tell a story of reliability and resilience in the transportation network. If a small number of RSUs in a network are impacted by a DoS attack, authorities and engineers can be confident in the attack outcome. While the attack may produce crashes and network delay, there is a high confidence level that the impacts will be similar to other attacks. Knowing the impact of a single compromised RSU can allow decision-makers to plan and introduce redundancies and resilience countermeasures into the system. On the other hand, the data shows that the higher numbers of infected RSUs directly challenge this idea. The two larger scenario groups (six and twelve infected RSUs) have widely variant results. A more significant variation in traffic disruption from a widespread attack against multiple RSUs is inherently harder to plan against and design countermeasures for.

From the variation calculations, it is already clear how a widespread attack can introduce a high level of chaos into the transportation network. Decision-makers must understand these variance calculations to understand how to mitigate future attacks and build a robust connected vehicle environment.

The next series of calculations were performed to quantify the actual impact of the DoS attacks on vehicle travel time. As mentioned previously, the delay added to each vehicle in the simulation is the most meaningful measure of an attack's impact on the transportation network.

Table 41 shows each scenario's average simulation time per vehicle and provides simple comparisons to the base scenario to show the relative impact of each attack.

Table 41 - DoS Scenarios - Relative Delay

Scenario	Avg. Sim Time / Vehicle (s)	Change from Base Scenario (s)	% Change from Base Scenario
0	325.9	--	--
1	332.2	6.3	1.9%
2	333.6	7.7	2.4%
3	332.4	6.5	2.0%
4	354.5	28.5	8.8%
5	358.8	32.8	10.1%
6	367.8	41.9	12.9%
7	376.0	50.1	15.4%
8	390.8	64.8	19.9%
9	393.2	67.2	20.6%
Average	**359.9**	**34.0**	**10.4%**

The takeaway from the above calculations is that, similar to the variance calculations; there is a clear impact threshold once six or more RSUs are infected. From scenario 3 to scenario 4, there is a twenty-two-second increase in delay, and the delay gets increasingly worse with each of the successive scenarios. The worst-case scenario (scenario 9) presents an over twenty percent increase from the baseline scenario. While that may only represent a sixty-seven-second increase, this simulation represents a small urban environment on a compressed time sample, meaning that percentage applied to an average commuter on a larger time scale would correspond to a significant loss of time. It is also important to note that these delays are measured per vehicle and represent an overall loss of time for the entirety of the network. A twenty percent loss of efficiency for the network as a whole is considerable - especially if the attack were to be conducted during peak hours. On a microscopic level, vehicles that have been delayed at the compromised intersections would feel a considerably more significant amount of delay. The delay introduced to them was enough to impact the efficiency of the network as a whole

drastically. Figure 10 provides a graphical representation of the correlation between the number of compromised RSUs and the average simulation time per vehicle.

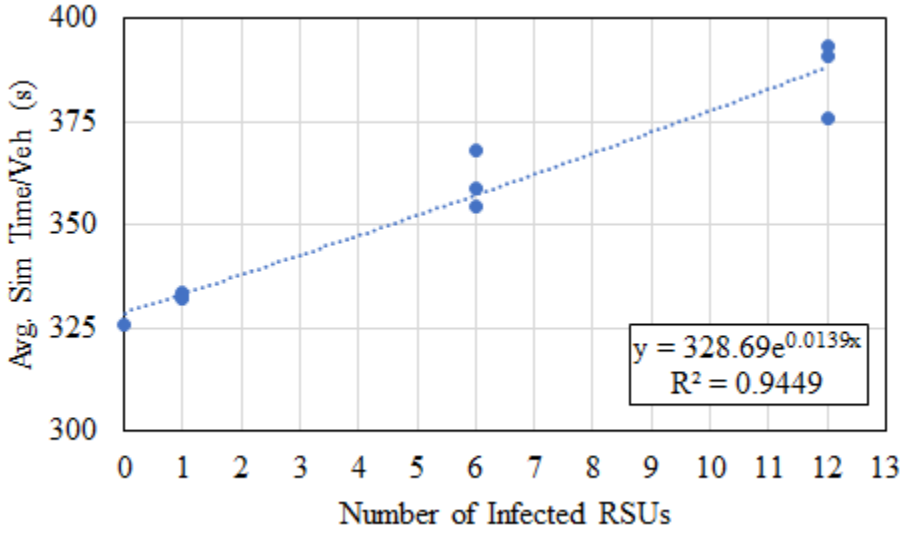


Figure 10 - DoS Infected RSUs vs. Travel Time

The exponential curve demonstrates a heavy correlation between the number of infected RSUs and travel time (R^2 = 0.95). The high correlation shows that as the attack scenarios get more severe, their impact on network delay also becomes more severe at an exponential rate. These results directly support the conclusions drawn from the variance calculations that as the severity and number of compromised RSUs increase, the consequences become increasingly challenging to manage. One of the early takeaways from the data analysis so far is that limiting the initial damage of a DoS attack is one of the most crucial steps in maintaining the transportations network's integrity in a compromised setting.

Another important aspect of transportation network efficiency is throughput - how many vehicles are making it through the simulation during each of the scenarios. Although the throughput data output from the simulation is somewhat circumstantial in terms of the numbers themselves, the data serves to show overall trends on an efficiency level between the scenarios. Figure 11 serves to point out the overall trend in vehicle throughput for each attack severity.

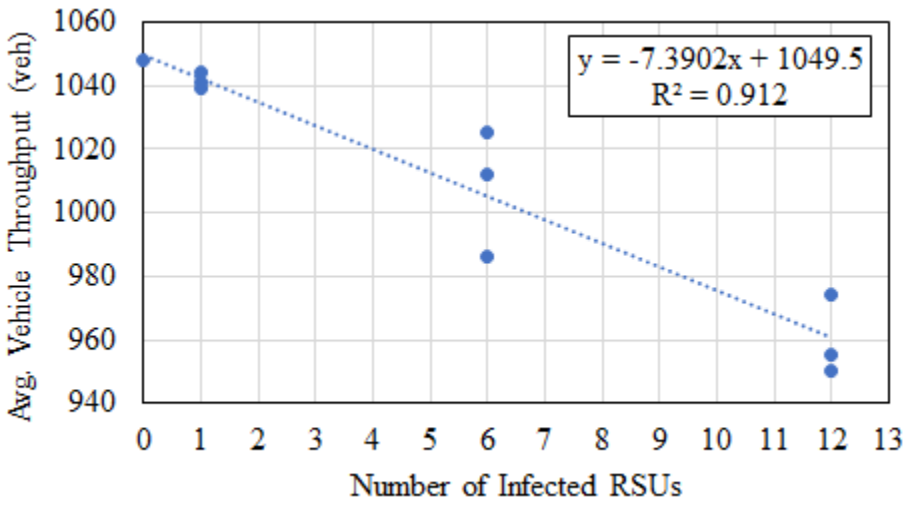


Figure 11 - Infected RSUs' Impact on Average Throughput - DoS

The figure confirms a strong correlation between increased scenario severity and decreased operational performance (R^2 = 0.91). It can also be seen that scenarios 1 - 3 remain relatively consistent in terms of throughput. In contrast, the others consistently suffer, indicating that a threshold for increasingly poor operational performance exists for the subsequent scenarios. The space between each data point for scenarios with more than one infected RSU further highlights the increase in variance as attack severity increases.

So far, the data analysis has been focused on efficiency. While efficiency is the most meaningful measure produced by the simulation, safety is also an extremely important aspect of transportation and one of the stated main focuses of the research. Like the calculated delay, crashes were counted for each DoS scenario and compared to the base scenario in Table 42.

Table 42 - DoS Scenarios - Crashes

Scenario	Avg. Crashes
0	0
1	2
2	2
3	2
4	8
5	10
6	10
7	15
8	18
9	21
Average	**10**

Similar to previous findings, average crashes per scenario increases with each increase in the number of infected RSUs. There is still a relatively clear threshold after scenario 3 where crashes increase suddenly, but not as severe as the delay calculations. The correlation between the increase in infected RSUs and crashes is rather apparent and expected as introducing more compromised RSUs increases infected vehicles entering intersections throughout the network, resulting in more compromised vehicle interactions and thus crashes. Keeping with the so far running theme, damage to the network becomes significantly worse once attacks are more widespread throughout RSUs in the network.

Examining the scenario groupings, it appears that the likelihood of attack success may have an impact on the result of the simulation for the first time. For scenarios 4 through 9, an overall trend of increased crashes appeared as the scenarios increased in intensity. This trend will be further investigated in a subsequent paragraph.

For vehicle occupancy safety purposes, it is important to describe how vehicles crash under the DoS attack scenarios. Due to the nature of the attack in the simulation, an infected vehicle never makes a reservation at the intersection. The vehicle assumes the intersection is clear and can go full speed into the intersection with no interruptions. However, the vehicle is

still capable of adjusting to traffic in its immediate vicinity. This primarily means that if a vehicle in front of it was slowed down (due to a properly made reservation), it would not rear-end it; it would simply slow down to its safe following velocity. With that being said, the infected vehicle would still enter the intersection blindly without regard to other approaching vehicles. A vehicle could perform emergency braking to avoid a crash, but often there is not enough time, so a crash would occur. With all that being said, crashes tend to be angle or turning crashes between two vehicles on conflicting paths. Both of these crash types have a high crash severity. If, in the future, CAVs act as shared shuttles with higher occupancy than most vehicles today (mostly single occupancy), these directional crashes could result in higher injury rates during a DoS attack. Even at urban speeds, directional crashes can have a significant human impact, only to be compounded by vehicle occupancy numbers. While crashes are inherently bad, DoS attacks all but eliminate rear-end crashes, thus raising the human cost of any crash that occurred during these attacks.

As mentioned previously, the impact of a DoS attack's effectiveness has not yet been explored. Logically, it would be expected that for each scenario grouping, an attack's consequences would be worse for each increase in attack success rate. Table 43 investigates this assumption by comparing average simulation time per vehicle and average crashes for each scenario compared to the previous scenario.

Table 43 - DoS Results Compared by Scenario

Scenario	Avg. Sim Time / Vehicle (s)	% Change from Previous Scenario	Average Crashes	%Change from Previous Scenario
0	325.9	--	0	--
1	332.2	1.9%	2	--
2	333.6	0.4%	2	0.0%
3	332.4	-0.4%	2	0.0%
4	354.5	6.6%	8	300.0%
5	358.8	1.2%	10	25.0%
6	367.8	2.5%	10	0.0%
7	376.0	2.2%	15	50.0%
8	390.8	3.9%	18	20.0%
9	393.2	0.6%	21	16.7%
Average	**359.9**	**2.1%**	**10**	**51.5%**

Table 43 above further corroborates the overall idea that as an attack scenario worsens (more RSUs or a higher chance of success), the attack's impact is more pronounced on the network. While the overall trend is true, there is no consistent increase in impact for each increase in intensity. The impact tends to be more severe, but the increase in severity appears to be random. It is also important to note that the percent increase for the individual MOEs are not comparable, as they are in different units and have different magnitudes of impact. An increase in two crashes is statistically more significant than an increase in two seconds of travel time.

With the correlation between an attack's probability of success and impact established, the question now becomes whether the number of RSUs is more or less impactful on the network's performance. Similar to the previous analysis, Table 44 seeks to compare scenario groups (grouped by the number of infected RSUs) and how the number of compromised RSUs impacts the safety and efficiency of the network.

Table 44 - DoS Results Compared by Scenario Group

Scenario Grouping	Avg. Sim Time / Vehicle (s)	% Change from Previous Scenario	Average Crashes	%Change from Previous Scenario
0	325.9	--	0	--
Scenarios 1-3	332.7	2.1%	2	--
Scenarios 4-6	360.4	8.3%	9	366.7%
Scenarios 7-9	386.6	7.3%	18	92.9%
Average	**359.9**	**5.9%**	**10**	**229.8%**

The number of infected RSUs has a higher impact on traffic operations than the likelihood of success of the respective scenarios. As previously mentioned, crashes are assumed to rise significantly with the increase of infected RSUs. In turn, as crashes rise as does delay within the network. A scenario with six infected RSUs operating at a seventy-five percent attack rate will have less impact on the system than a scenario with twelve infected intersections operating at a twenty-five percent attack rate. The impact of each scenario group can be seen in Figure 12, which plots the average velocity per vehicle per timestep over time. This statistic is more for visual representation than data analysis, but it provides a visual representation of how vehicles operate throughout the simulation.

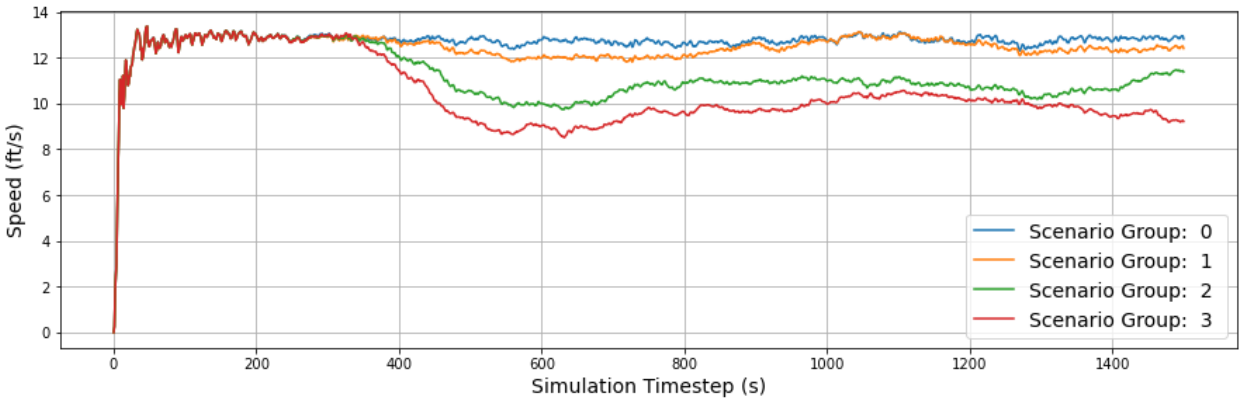


Figure 12 - Vehicle Speed per Time Step - DOS Groups

The figure shows two significant dips in scenario groups 2 and 3, further emphasizing that the damage to the network compounds seriously as attack severity increases. The two

scenario groups feel the impacts from the attack throughout the entirety of the simulation as they can never fully recover to normal operations.

Overall, a few main points have emerged from preliminary data analysis of DoS attack scenario data. The first point is that a clear threshold of impact exists between scenarios 0 - 3 and scenarios 3 - 9. Put into words, as a DoS attack is more widespread throughout the network, the more pronounced and severe its impacts become. This is a predictable result as more infected RSUs result in more crashes, therefore, more delay. The main takeaway from this conclusion is that the primary mitigation strategy for a DoS attack is to limit its initial effectiveness to prevent a widespread network attack. Keeping with this theme, it was also clear that the variance of results increases significantly as the attack scenarios grow more severe. The impact of one infected RSU is relatively easy to plan resilience concepts against, but as the attack severity increases, this becomes more difficult. The high variance makes specific resilience concepts challenging to implement as the attack's impact can vary so much with each attack. The variance also impacts the reliability of the transportation network for those who depend on it. Finally, it was found that both likelihoods of success and scenario groupings impact the severity of the results. Scenario grouping is a better indicator of severity as a more widespread attack is inherently worse on the system, regardless of an attack's likelihood of success.

Man in the Middle Attack Scenarios Analysis

Similar to the DoS data analysis, the first step in analyzing MITM attack results was to analyze the variance and other statistical calculations of the results produced from each scenario. Table 45 shows the variance, T-test results, and Chi-Squared test results.

Table 45 - MITM Statistical Analysis

Scenario	Variance	T-Test Results	Chi^2 Test Results
0	0.06	--	--
10	8.77	1.0 E-02	0.955
11	0.29	2.0 E-06	0.978
12	3.21	1.3 E-03	0.959
13	31.11	4.4 E-04	0.025
14	10.75	2.2 E-05	0.002
15	1.09	2.9 E-08	4.9 E-05
16	26.47	4.1 E-05	3.8 E-06
17	46.80	4.4 E-05	9.8 E-11
18	1.73	1.2 E-08	2.1 E-15
Average	**14.47**	**1.4 E-03**	**0.325**

Once again, the T-tests were calculated to be under the 95% confidence threshold for all scenarios, showing that the results would never overlap with the base scenario.

The variance of the MITM attack scenarios is also significant to note. Similar to the previous attack, the Chi-Squared value for Scenarios 10, 11, and 12 came back above the threshold. This means that these are the only scenarios that decision-makers could be confident in forecasting the resulting impact on the network. For the other attacks, the results could vary drastically, making resilience planning extremely difficult. The general trend shows an increase in variance; however, the trend is not exactly clear cut. Figure 13 provides a visual representation of the degrees for each attack scenario.

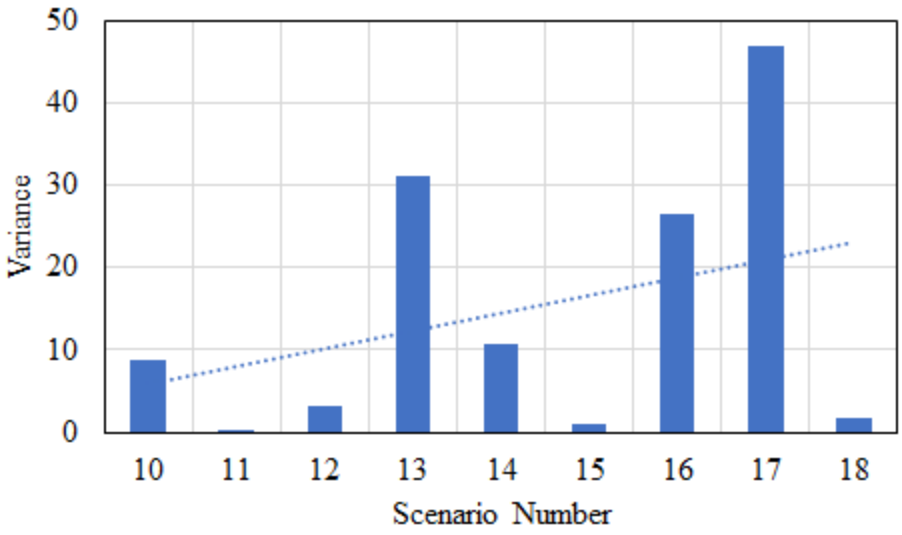


Figure 13 - MITM Attack Variance by Scenario

While the general trend for MITM attack variance generally increases with increased attack severity, this analysis does not tell the entire story. The increasing trendline is characterized by relatively low variance with extreme spikes from scenarios 13, 16, and 17. If these three extreme cases are removed, the average variance is only 3.80 per scenario (excluding the base scenario), a relatively small variance given the simulation's setup and the inclusive average of 14.47.

MITM attacks appear to have a low variance in results, aside from a small number of highly chaotic scenarios, which means that the impacts of a MITM attack could be easily predicted and planned for. However, the extreme cases of variance and the calculated Chi-Squared values make the picture significantly less clear as these cases can not be ignored. These extreme cases make the job of decision-makers increasingly tricky when deciding on how to mitigate against MITM cyberattacks actively. The possibility of highly variant results broadens the scope of protecting the network against an attack and introduces a level of chaos that should not be ignored. Suppose the extreme situations are not correctly accounted for. In that case, a high-impact attack could bypass any existing countermeasures and significantly impact the

transportation network - effectively defeating any resiliency concepts introduced for the "average" scenario.

While the variance is essential for understanding the outcome of an attack, knowing how an attack will impact, the network is also important. Table 46 notes the average simulation time per vehicle in each MITM attack scenario and compares the results against the base scenario.

Table 46 - MITM Attack Scenarios - Relative Delay

Scenario	Avg. Sim Time / Vehicle (s)	Change from Base Scenario (s)	% Change from Base Scenario
0	325.9	--	--
10	332.0	5.1	1.8%
11	331.4	5.4	1.7%
12	332.2	6.3	1.9%
13	352.2	26.4	8.1%
14	358.8	32.9	10.1%
15	366.4	40.4	12.4%
16	370.3	44.4	13.6%
17	384.2	58.3	17.9%
18	395.8	69.8	21.4%
Average	**358.2**	**32.2**	**9.9%**

Similar to the previous analysis, a clear impact threshold appears after scenario 12, once the number of infected RSUs jumps from one to six. This threshold is expected as an increase in infected RSUs directly relates to an increase in negative impacts on the network. After this threshold, each successive scenario carries an increasingly more significant impact on the delay felt by the average vehicle in the network. With the delay, the worst case occurs during scenario 18 with an over twenty percent increase in average travel time per vehicle. As previously mentioned in the DoS section, an over twenty percent increase in travel time could significantly impact the transportation network during peak hours and in a larger scale environment. The main takeaway from this analysis is that after scenario 12, each attack scenario contributes more and more significant delays to the network. This takeaway can be seen in Figure 14**.**

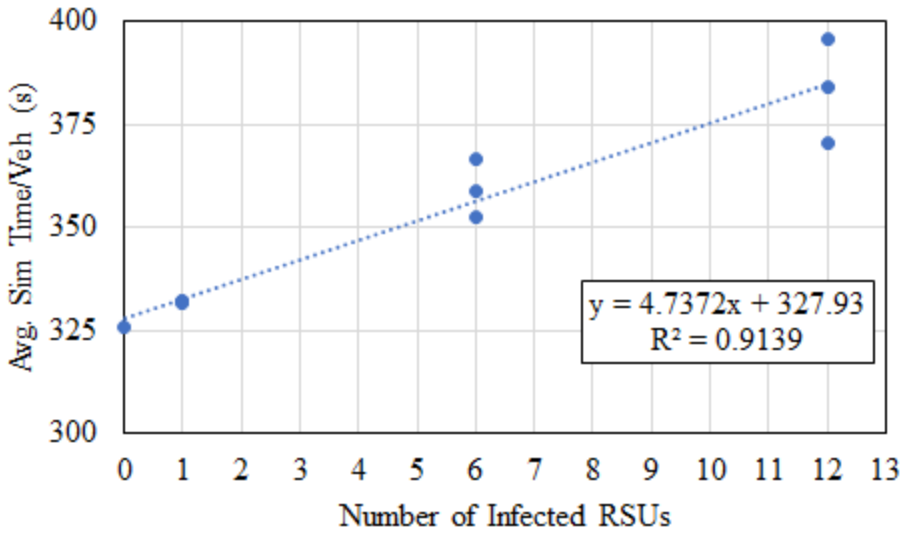


Figure 14 - MITM Infected RSUs vs. Travel Time

The figure presents a high linear correlation ($R^2 = 0.91$) between an attack's severity and the impact the attack has on travel time. This correlation further demonstrates the need to limit an attack's initial success, as a more widespread success in attacks results in exponentially more significant impacts on the network.

Average vehicle throughput is also an important indicator of network performance under the different attack scenarios. While vehicle throughput is not as good of an indicator in the simulation environment, it is still important to highlight network efficiency trends from a macroscopic level. Figure 15 shows the impact of each attack scenario on overall throughput.

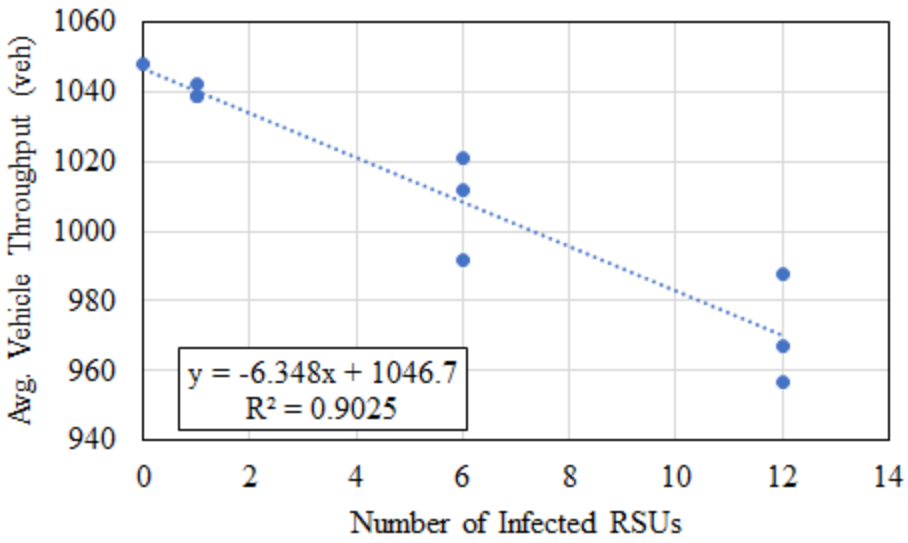


Figure 15 - Average Throughput per Number of Infected RSUs - MITM

The figure shows a clear correlation between attack severity and a decrease in average vehicle throughput with an R^2 value equal to 0.90. The figure also demonstrates the threshold discussed previously, with continued higher impacts with the increase in severity after scenario 12, once more RSUs are infected. The throughput data agrees with the delay data in that it is clear that the network is significantly impacted more as attack severity increases.

In addition to network performance, safety data needs to be analyzed for these attack scenarios. To do so, Table 47 compares average crashes per scenario.

Table 47 - MITM Scenarios - Crashes

Scenario	Avg. Crashes
0	0
10	1
11	2
12	2
13	7
14	9
15	9
16	13
17	18
18	20
Average	**9**

As expected, the safety data further enforces the idea of a threshold of damage from a widespread attack versus an isolated attack. Crash results are relatively consistent for scenarios 10, 11, and 12, but after, almost every increase in attack severity results in a higher crash count.

As mentioned in the DoS section, the crash scenarios for this simulation are different from that of the real world. However, the MITM crash scenarios do slightly differ from those of DoS attacks. The nature of the MITM attack in the simulation results in a vehicle's final acceleration to the intersection being delayed. This delay opens the door back up for the possibility of rear-end crashes as a vehicle trailing a delayed vehicle could accelerate to make its given reservation time and make contact with the delayed vehicle. In this case, the trailing vehicle would assume the delayed vehicle to have already passed through the intersection and result in a crash. However, through observation, most crashes resulting from these MITM attack scenarios were angle and turning crashes. Even with the possibility of rear-end crashes, the sentiment from the previous section remains the same. The crashes resulting from these attacks are likely to have a higher human cost than a traditional urban intersection that primarily deals with rear-end collisions. Higher occupancy and higher angle crash likelihood must be taken into account when considering the safety of a compromised intersection.

To understand how the different crash scenarios impact the outcome of the simulations, it is essential to compare the results on a case-by-case basis and a group-by-group basis. The average change between each scenario was first calculated and shown in Table 48 for comparison.

Table 48 - MITM Results Compared by Scenario

Scenario	Avg. Sim Time / Vehicle (s)	% Change from Previous Scenario	Average Crashes	%Change from Previous Scenario
0	325.9	--	0	--
10	332.0	1.8%	1	--
11	331.4	-0.2%	2	100.0%
12	332.2	0.3%	2	0.0%
13	352.2	6.1%	7	250.0%
14	358.8	1.8%	9	28.6%
15	366.4	2.1%	9	0.0%
16	370.3	1.1%	13	44.4%
17	384.2	3.8%	18	38.5%
18	395.8	3.0%	20	11.1%
Average	**358.2**	**2.2%**	**9**	**59.1%**

Furthermore, the table above continues to reinforce two main points about the MITM attack data. The first being that the most significant jump from both data sets comes between scenarios 12 and 13. This supports the idea that a threshold exists between these two scenarios and is further reinforced by the trend of increased negative impact with each successive scenario (except for crashes in scenario 15). These two analysis points show that once the threshold is reached, a slippery slope develops where any increase in intensity further impacts the system.

The next question is whether scenario groupings have more of an impact than looking at the data from a scenario-to-scenario level. Table 49 seeks to do just this by comparing the percent change between scenario groupings.

Table 49 - MITM Results Compared by Scenario Group

Scenario Grouping	Avg. Sim Time / Vehicle (s)	% Change from Previous Scenario	Average Crashes	%Change from Previous Scenario
0	325.9	--	0	--
Scenarios 10-12	331.8	1.8%	2	--
Scenarios 13-15	359.2	8.3%	8	400.0%
Scenarios 16-18	383.5	6.8%	17	104%
Average	**358.2**	**5.6%**	**9**	**252.2%**

After analyzing the two separate groupings of data, it is clear that the change between scenario groupings causes a more significant change in the outcomes than simply switching between the individual scenarios themselves. This is expected as introducing more infected RSUs into the network results in more widespread disruption to traffic operations. More so than increasing the likelihood of a successful attack with a constant number of RSUs. To further understand the impact of each scenario group, Figure 16 displays the average vehicle speed for each time step in the simulation.

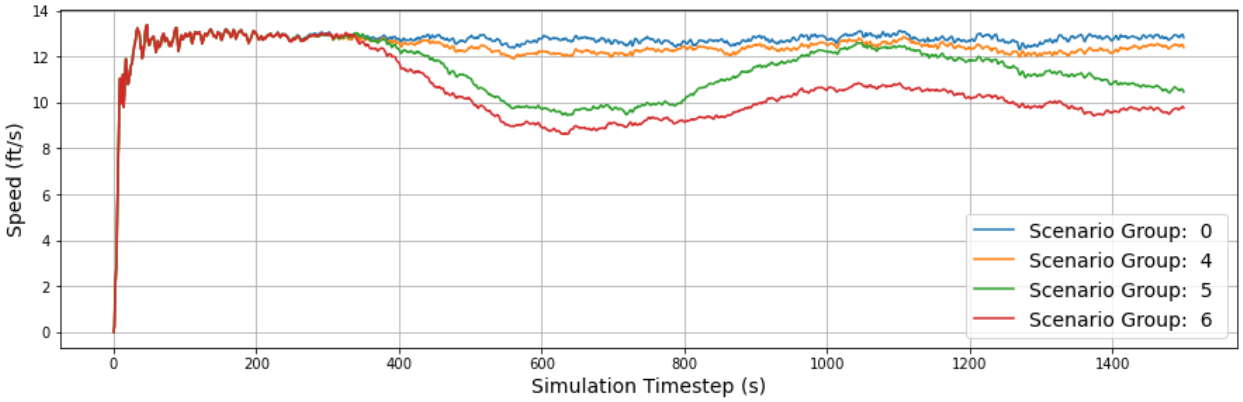


Figure 16 - Vehicle Speed per Time Step - MITM Groups

The figure clearly shows the significance of the impact scenario groups 5 and 6 have on the normal traffic operation during the simulation. There is a significant dip in speed that bottoms out just after 600 seconds. The plot shows the network recovering as speeds increase, but it is clear that scenario 6 never recovers due to how widespread the attacks are.

The overall conclusions from the MITM attack data analysis are very similar to the ones made for DoS data analysis. One of the main themes so far in terms of data analysis is that increasing the number of RSUs has the most direct and significant impact on the operations and safety of the transportation network. This impact can be seen throughout the analysis in scenarios 13 - 18, where performance steadily decreases with each iteration of attack severity. One of the main points these attacks seem to diverge in data analysis is their variance per scenario. MITM

attacks had a relatively low variance except for three scenarios with seemingly random spikes in variance. While the lower variance would benefit planners and decision-makers, the spikes in different scenarios present a troubling problem. It appears that at least some scenarios, seemingly without cause, can have extreme variance - meaning significantly different efficiency results from different trial runs within a scenario.

While this sudden change in variance is troubling, one of the main takeaways, again, is that the most critical resilience strategy is to limit an initial MITM attack before it can spread throughout the network. The more widespread an attack becomes, the more damage it will do to network efficiency and passenger safety.

Combined Data Analysis

Now that the two attacks have been compared independently, it is now important to compare the two attacks against each other at a high level to make general conclusions on them. This analysis will compare data already presented to show the similarities and differences between the impact of the two attacks - consisting primarily of average and summary data.

Keeping with the theme of the previous sections, the average variance for each of the attacks will be compared first, in Table 50.

Table 50 - Average Variance Comparison

Attack	
DoS	**MITM**
16.80	14.47

The average variance for each attack is considerable, but they achieved these averages in different manners. For the DoS attacks, the variance increased as the attack scenarios became more severe. On the other hand, MITM attacks achieved their high variance through sporadic spikes in variance from three main scenarios. Aside from these three attacks, the variance for MITM attacks would have been considerably lower. However, these attacks occurred, and the

results have to be accounted for. Overall, the average variation did not change much from attack to attack. In fact, for both attacks, only the scenarios with one infected RSU achieved a Chi-Squared score within the 95% confidence interval threshold. This shows that although their path to high variance may be different, either way, the more severe attack scenarios result in high variance.

With the two attacks having similarly high variance calculations, a challenge arises for planners and decision-makers to develop robust countermeasures for these attacks. Depending on the volume, an attack can have a wide range of impacts on the network's operations. As previously mentioned, modern traffic collisions behave similarly. A crash during rush hour at a busy urban intersection will have a considerably different impact than a crash in the middle of the day on a side street. The high variance in network impact makes things challenging for modern traffic engineers and will continue to do so for traffic engineers in a connected environment. The main conclusion drawn from this comparison is that a severe DoS attack and a severe MITM attack are extremely hard to predict and plan for confidently. Decision-makers must develop a resilience plan that takes the unpredictable nature of the impact of these attacks into account.

Variance tells a story of how different the impact of each attack could be. However, it is important to compare key measures of effectiveness from the simulations themselves with the simulation data. Average simulation time per vehicle has been the primary statistic for analysis throughout this chapter so far. Table 51 will compare the statistics for each of the related scenario groups for each of the attacks. The simulation time for each will also be compared to the base scenario for reference.

Table 51 - Average Simulation Time per Vehicle Comparison

Attack					
DoS			MITM		
Scenario Group	Sim Time / Vehicle (s)	% Change from Base	Scenario Group	Sim Time / Vehicle (s)	% Change from Base
0	325.9	--	0	325.9	--
1	332.7	2.1%	4	331.8	1.8%
2	360.4	10.6%	5	359.2	10.2%
3	386.6	18.6%	6	383.5	17.6%
Average	**359.9**	**10.4%**	**Average**	**358.2**	**9.9%**

The first trend is how similar the average result for each scenario group and the overall average results are. The DoS attacks have an average of 1.7 seconds more simulation time per vehicle, which results in a 0.5% more delay relative to the base scenario. For all intents and purposes, the numbers are equal. This is an important finding as MITM attacks are often thought of as less impactful than a DoS attack. The above findings find very little difference between a well-executed severe DoS attack and a well-executed severe MITM attack. Finding that the two have nearly identical efficiency impacts puts more pressure on engineers and decision-makers to develop a robust and resilient system against a variety of cyberattacks. A well-positioned adversary could use various attacks to compromise the network, and picking and choosing which to focus on could result in significant efficiency impacts on the network.

Next, the safety implications of each of the attacks need to be compared. Similarly, Table 52 compares the average crashes for each attack and their corresponding scenario groupings. No comparison will be made against the base scenario as there were no crashes in this scenario.

Table 52 - Average Crash Comparison

Attack			
DoS		**MITM**	
Scenario Group	**Average Crashes**	**Scenario Group**	**Average Crashes**
0	0	0	0
1	2	4	2
2	9	5	8
3	18	6	17
Average	**10**	**Average**	**9**

Similar to operational impacts, DoS and MITM attacks have nearly identical safety implications. On average, one less crash occurs in a MITM attack than a DoS, but that difference is not significant at scale. One of the main differences in the crashes that can be made is how vehicles crash in the attacks. Rear-end crashes are effectively eliminated under the DoS attack methodology, resulting in only angle and turning movement crashes. On the other hand, there is still a possibility during a MITM attack for a rear-end collision, but most crashes occur at an angle or during a conflicting turning movement. Compared to modern urban intersections, the increase in these severe crash types introduces a more significant human risk when a crash occurs. Angle and turning movement crashes are more likely to result in more serious vehicle damage and an increased chance of serious injury or death. With an expected higher occupancy for CAVs in the future, these more severe crashes could result in a high injury rate among passengers in a vehicle.

Safety is one of the critical aspects of modern traffic engineering and is thought to be a problem solvable by implementing CAVs. However, safety engineering in the future is more likely to be focused on cybersecurity aspects of transportation than human behavior. Without a proper understanding of the possible safety impacts of these attacks, engineers can not properly design a system that ensures the safety of those depending on the connected environment for transportation. Understanding that DoS and MITM attacks can have nearly identical negative

impacts on safety is an excellent first step to take in developing a cyberattack resilient CAV network.

The final means of comparing the impacts from the two simulated attacks are visually based more so than statistics based. Figure 17 provides the average speed per vehicle per timestep over simulation time for each of the scenario groups. This provides a visual representation of how operations are impacted during the attack simulations.

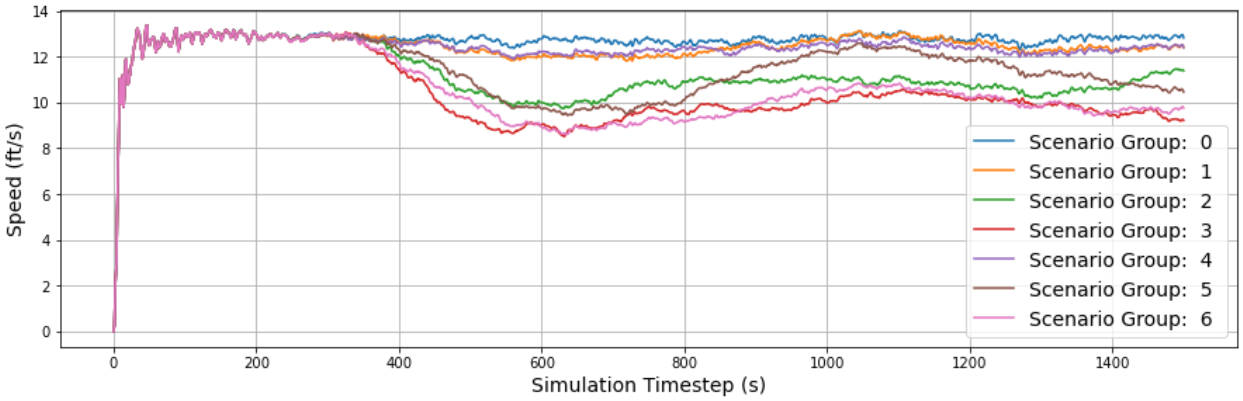


Figure 17 - Average Speed per Time Step - Scenario Groups

Each of the attacks' corresponding scenario groups (groups 1 and 4, groups 2 and 5, and groups 3 and 6) behave similarly. They experience very similar dips and recovery in speed, with one main exception. Scenario group 5 recovers to a much higher speed on average than scenario group 2, with its peak around 1,100 seconds. The figure also reiterates the theme that scenario groups 1 and 4 behave very similarly to the baseline scenario, especially compared to the other attack groups. This graph visually represents the operational impacts of the widespread attacks shown in scenarios 2, 3, 5, and 6. These groups have significantly higher drops in speed and generally fail to recover, at a network scale, during the simulation.

The final comparison between the two attacks is shown in Figure 18 as a simple comparison between the average speed per vehicle over time for each attack as a whole.

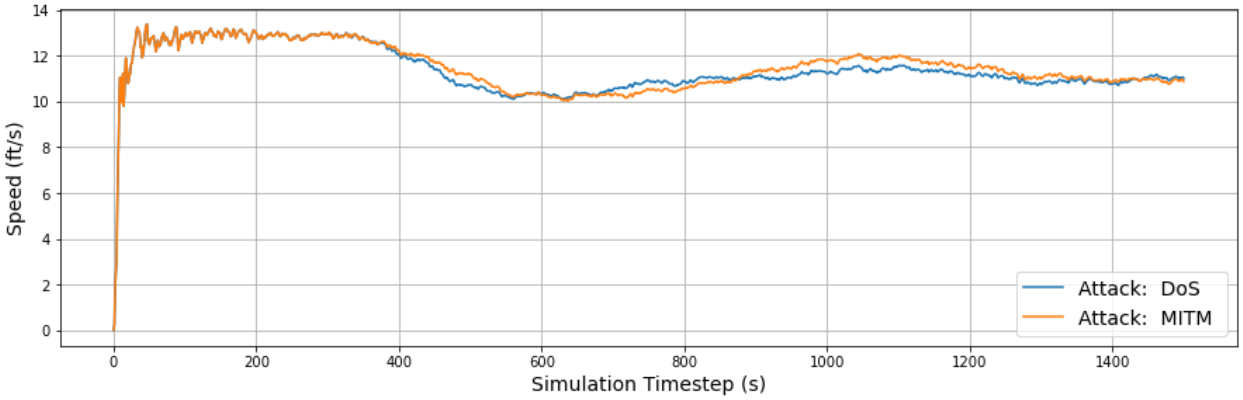


Figure 18 - Average Speed per Time Step - Attack Groups

The only visible difference between the two curves occurs after the 1,000 second time step. The graph shows that a network that is the victim of MITM attacks, on average, has a higher ability to recover than a network that has experienced a DoS attack. This difference is slight and does not take away from the overall similarity each attack's impact has on the network.

A conclusion can be drawn between the simulated DoS and MITM attacks through multiple means of comparison - their impact on network safety and efficiency are nearly identical. This is an important concept to understand in developing a robust transportation network against cyberattacks. While this research focused on two significant cyberattacks, it has also shown an incredible variety of attacks an adversary can use to compromise the network in one way or another. Understanding the impact of these attacks is the first step in understanding the risk they pose to the network. Effectively understanding the possible risk of these attacks can help engineers gain a step on nefarious actors in the battle for cybersecurity.

6. Summary, Key Findings, Conclusions and, Future Work

Summary

Throughout this thesis, many aspects of the impact of cyberattacks on the transportation network have been covered. This analysis includes background information, a literature review,

methodology, simulation development, limitation, results, data analysis, and key findings. Over the course of this thesis, a theme of vulnerability has emerged. The future of transportation is sure to consist of intelligent transportation systems and CAVs interacting with each other to take the human element out of the vehicular transportation system. These improvements promise safety and operational improvements but come with their own downsides. A long list of tested and theorized cyberattacks against these connected systems was created through an in-depth literature review. In a world where ITS is emerging in different ways, from variable message signs to intelligent traffic signals, hackers have already found ways to compromise these systems. These attacks carry with them the possibility of decreased network safety and efficiency. In the same vein, hackers have also found ways to exploit existing vehicles through sensors and communication mechanisms. Regardless of where humans fall in the transportation system, whether it be as a driver or a passenger in a CAV, the threat against them will always be human. Whether the danger comes from a distracted driver running a red light or a human launching a cyberattack to cripple a city, the transportation system that most humans depend on will always have to be human proof.

Key Findings

This section summarizes the key takeaways from the data analysis and presents them as bullet points for the most accessible consumption.

- The negative operational and safety impacts from DoS and MITM cyberattacks are nearly identical.
    - Engineers must understand that these attacks carry similar and equally significant impacts on the network when building a system to resist these attacks.

- A threshold exists that corresponds to the number of infected RSUs and the subsequent negative impact of the attack. More infected RSUs correspond to more damage.
    - A higher number of successfully compromised RSUs is more impactful and carries more severe consequences than less compromised RSUs with a higher chance of a successful attack on vehicles.
- Impacts from both attacks carry with them high levels of variance.
    - It is challenging for decision-makers to be confident of how severely an attack will negatively impact their system until after the attack has occurred.
- The worst-case scenario for safety and operational efficiency is a widespread DoS or MITM attack.
    - A focus of resilience planning should be to detect and limit the spread of an initial attack. Attacks below the stated threshold will have negative impacts but will be limited compared to the base scenario and occur with a low level of variance. A widespread attack can have severe safety and efficiency implications with a very poor level of subsequent operational recovery.

Real World Application

The overall goal of this research was to provide a baseline to transportation planners on the impacts of cyberattacks on the transportation network. The results of this research and the key findings can be of importance to planners at all levels, but specifically for state and municipal DOTs. The findings of this research help to show decision makers what implementing smart infrastructure in urban environments without putting appropriate thought into cybersecurity prior to implementation. With no resilience countermeasures put into place, it is likely that any successful cyberattack will have the impact on traffic operations demonstrated

throughout this research. This baseline scenario represents the worst-case scenario in terms of cybersecurity readiness and resilience and is important to consider. Using these results, state and local DOTs should realize that it is irresponsible to implement operational and safety-critical infrastructure without first developing a cybersecurity resilience and mitigation. Without these plans in place, the new infrastructure will provide a risk to those who depend on it.

Conclusions

This research aimed to quantify the impact that DoS and MITM cyberattacks have on the safety and operations of a connected transportation system. To do so, a model was adapted to simulate these attacks and measure their impact. The key findings from this simulation are again listed below:

- DoS and MITM attacks carry with them very similar safety and operational impacts on the network.
- A threshold exists for the number of infected RSUs where increased attack severity correlates to an increase in negative impacts on the system.
- DoS and MITM attacks have significant variance in their impact, making predicting the exact impact of these attacks difficult.
- The worst-case attack scenario is a widespread cyberattack of either kind, emphasizing immediate threat response to mitigate initial attacks spreading through the network.

The overarching goal for this research is that the results can be used in a risk assessment while planning for a robust and resilient connected transportation network in the future. The results show what could happen if cybersecurity is ignored through ITS and CAV implementation. Understanding the impact an attack is first step in planning against it, and without a proper understanding, a resilience plan is essentially useless against the impact. Proper

cyberattack risk management and mitigation will be essential to ensure the safety and operation of the next generation of vehicular transportation. Without it, people will be at the hands of nefarious actors as they attempt to go about their daily lives. This research is just one step in the more significant movement to ensure that the connected environment is indeed human proof.

Future Work

The high-level nature of the research conducted aimed to provide the quantitative impact of two separate cyberattacks on the safety and operational efficiency of a CAV transportation network. This research is intended to act as a base level of information to inform future research. With the results and conclusions gained from this research, a variety of potential future research could be conducted, further expanding on this topic and adding more detail to it. These future research opportunities are detailed below:

- Incorporate more attack scenarios to fine tune the results and understand where exactly the transportation begins to experience debilitating effects from attacks
- Calibrate the simulation to emulate real world traffic scenarios and networks. Reducing arbitrary factors in the research can further narrow the results to be used for resilience planning.
- Introduce scale to the simulation in both size, complexity and, traffic volumes
- Develop resiliency countermeasures for these attacks at both a cybersecurity and transportation level - further understanding the interaction of these two fields.
- Further focus on and model transportation system recovery in the aftermath of a successful cyberattack(s) on the transportation network.

The conducted research opens the door for a wide variety of future research, but the future research should remain focused on the topics of resilience put forward throughout this

thesis. Further understanding the interactions between cyberattacks and their impacts on the transportation network could result in finding solutions and developing a resilient transportation network for the future.

References

[1] "Intelligent Transportation Systems - ITS Program Overview," *United States Department of Transportation*. https://www.its.dot.gov/factsheets/ITSJPO_overview.htm (accessed Apr. 20, 2021).

[2] "Automated Vehicles for Safety," *NHTSA*, Sep. 07, 2017. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety (accessed Jan. 24, 2021).

[3] P. Davidson and A. Spinoulas, "Autonomous Vehicles - What Could This Mean For The Future Of Transport?," p. 15, 2015.

[4] M. Lavasani, X. Jin, and Y. Du, "Market Penetration Model for Autonomous Vehicles on the Basis of Earlier Technology Adoption Experience," *Transportation Research Record*, vol. 2597, no. 1, pp. 67–74, Jan. 2016, doi: 10.3141/2597-09.

[5] Waymo LLC, "Waymo Safety Report," https://storage.googleapis.com/sdc-prod/v1/safety-report/2020-09-waymo-safety-report.pdf, Sep. 2020.

[6] B. G. Sun Yilei, "Tesla 'very close' to level 5 autonomous driving technology, Musk says," *Reuters*, Jul. 09, 2020. Accessed: Apr. 20, 2021. [Online]. Available: https://www.reuters.com/article/us-tesla-autonomous-idUSKBN24A0HE

[7] SAE International, "Standard J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles." Jun. 2018.

[8] T. Litman, "Implications for Transport Planning," p. 45, Jun. 2020.

[9] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, pp. 380–392, Jan. 2014, doi: 10.1016/j.jnca.2013.02.036.

[10] NHTSA, “FMVSS No 150 Vehicle to Vehicle Communication Technology for Light Vehicles.pdf.” U.S. Department of Transportation, Federal Highway Administration, Office of Traffic Operations, Nov. 2016.

[11] M. S. Al-kahtani, “Survey on security attacks in Vehicular Ad hoc Networks (VANETs),” in *2012 6th International Conference on Signal Processing and Communication Systems*, Gold Coast, Australia, Dec. 2012, pp. 1–9. doi: 10.1109/ICSPCS.2012.6507953.

[12] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip, and R. Gerdes, “Survey on Vehicular Ad Hoc Networks and Its Access Technologies Security Vulnerabilities and Countermeasures,” p. 22.

[13] L. Ye and T. Yamamoto, “Modeling connected and autonomous vehicles in heterogeneous traffic flow,” *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 269–277, Jan. 2018, doi: 10.1016/j.physa.2017.08.015.

[14] NHTSA, “Preview of Motor Vehicle Traffic Fatalities In 2019,” US DOT, DOT HS 813 021, Oct. 2020. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813021

[15] NHTSA, “Federal Motor Vehicle Safety Standards; V2V Communications.pdf,” US DOT, NHTSA, NHTSA-2016-0126, Jan. 2017. [Online]. Available: https://www.govinfo.gov/content/pkg/FR-2017-01-12/pdf/2016-31059.pdf

[16] NHTSA, “Traffic Safety Facts Annual Report,” US DOT, Jun. 2020. [Online]. Available: https://cdan.nhtsa.gov/tsftables/National%20Statistics.pdf

[17] M. S. Rahman and M. Abdel-Aty, “Longitudinal safety evaluation of connected vehicles’ platooning on expressways,” *Accident Analysis & Prevention*, vol. 117, pp. 381–391, Aug. 2018, doi: 10.1016/j.aap.2017.12.012.

[18] L. Ye and T. Yamamoto, “Evaluating the impact of connected and autonomous vehicles on traffic safety,” *Physica A: Statistical Mechanics and its Applications*, vol. 526, p. 121009, Jul. 2019, doi: 10.1016/j.physa.2019.04.245.

[19] D. Chen, S. Ahn, M. Chitturi, and D. A. Noyce, “Towards vehicle automation: Roadway capacity formulation for traffic mixed with regular and automated vehicles,” *Transportation Research Part B: Methodological*, vol. 100, pp. 196–221, Jun. 2017, doi: 10.1016/j.trb.2017.01.017.

[20] S. E. Shladover, D. Su, and X.-Y. Lu, “Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow,” *Transportation Research Record*, vol. 2324, no. 1, pp. 63–70, Jan. 2012, doi: 10.3141/2324-08.

[21] V. Milanes, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, “Cooperative Adaptive Cruise Control in Real Traffic Situations,” *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 1, pp. 296–305, Feb. 2014, doi: 10.1109/TITS.2013.2278494.

[22] J. Rios-Torres and A. A. Malikopoulos, “Automated and Cooperative Vehicle Merging at Highway On-Ramps,” *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 4, pp. 780–789, Apr. 2017, doi: 10.1109/TITS.2016.2587582.

[23] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, “Optimal control and coordination of connected and automated vehicles at urban traffic intersections,” in *2016 American Control Conference (ACC)*, Boston, MA, USA, Jul. 2016, pp. 6227–6232. doi: 10.1109/ACC.2016.7526648.

[24] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, “Road traffic congestion detection through cooperative Vehicle-to-Vehicle communications,” in *IEEE Local Computer Network Conference*, Denver, CO, USA, Oct. 2010, pp. 606–612. doi: 10.1109/LCN.2010.5735780.

[25] M. Milojevic and V. Rakocevic, “Short paper: Distributed vehicular traffic congestion detection algorithm for urban environments,” in *2013 IEEE Vehicular Networking Conference*, Boston, MA, USA, Dec. 2013, pp. 182–185. doi: 10.1109/VNC.2013.6737608.

[26] Y. Yang and R. Bagrodia, “Evaluation of VANET-based advanced intelligent transportation systems,” in *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking - VANET ’09*, Beijing, China, 2009, p. 3. doi: 10.1145/1614269.1614273.

[27] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations,” *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, Jul. 2015, doi: 10.1016/j.tra.2015.04.003.

[28] M. M. Cruz-Cunha and F. Moreira, Eds., “Overview of Security Issues in Vehicular Ad-Hoc Networks,” in *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*, IGI Global, 2011. doi: 10.4018/978-1-60960-042-6.

[29] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, “Connected Vehicles: Solutions and Challenges,” *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014, doi: 10.1109/JIOT.2014.2327587.

[30] E. Hamida, H. Noura, and W. Znaidi, “Security of Cooperative Intelligent Transport Systems: Standards, Threats Analysis and Cryptographic Countermeasures,” *Electronics*, vol. 4, no. 3, pp. 380–423, Jul. 2015, doi: 10.3390/electronics4030380.

[31] K. Koscher *et al.*, “Experimental Security Analysis of a Modern Automobile,” p. 16.

[32] N. Ekedebe, W. Yu, H. Song, and C. Lu, “On a simulation study of cyber attacks on vehicle-to-infrastructure communication (V2I) in Intelligent Transportation System (ITS),” Baltimore, Maryland, United States, May 2015, p. 94970B. doi: 10.1117/12.2177465.

[33] I. A. Sumra, I. Ahmad, H. Hasbullah, and B. S. Iskandar, “Behavior of attacker and some new possible attacks in Vehicular Ad hoc Network (VANET),” p. 8.

[34] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, “VANet security challenges and solutions: A survey,” *Vehicular Communications*, vol. 7, pp. 7–20, Jan. 2017, doi: 10.1016/j.vehcom.2017.01.002.

[35] G. Samara, W. A. H. Al-Salihy, and R. Sures, “Security Analysis of Vehicular Ad Hoc Nerworks (VANET),” in *2010 Second International Conference on Network Applications, Protocols and Services*, Alor Setar, Kedah, Malaysia, Sep. 2010, pp. 55–60. doi: 10.1109/NETAPPS.2010.17.

[36] F. Ahmad and A. Adnane, “A Novel Context-Based Risk Assessment Approach in Vehicular Networks,” in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Crans-Montana, Switzerland, Mar. 2016, pp. 466–474. doi: 10.1109/WAINA.2016.60.

[37] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, “A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3453–3495, 2018, doi: 10.1109/COMST.2018.2855563.

[38] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, no. S 91, 2015.

[39] A. Greenberg, “Hackers Reveal Nasty New Car Attacks--With Me Behind The Wheel (Video),” *Forbes*, Jul. 24, 2013. https://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-me-behind-the-wheel-video/ (accessed Jan. 27, 2021).

[40] J. Stoltzfus, “ECUs and the Controller Area Network - The Modern Car,” *Techopedia.com*, Aug. 07, 2020. https://www.techopedia.com/your-car-your-computer-ecus-and-the-controller-area-network/2/32218 (accessed Jan. 27, 2021).

[41] R. Charette, “This Car Runs on Code - IEEE Spectrum,” *IEEE Spectrum: Technology, Engineering, and Science News*, Feb. 01, 2009. https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code (accessed Jan. 27, 2021).

[42] D. C. Miller and C. Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle,” p. 91, Aug. 2015.

[43] S. Checkoway *et al.*, “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” p. 16.

[44] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, “Fast and Vulnerable: A Story of Telematic Failures,” p. 9.

[45] D. K. Oka, T. Furue, L. Langenhop, and T. Nishimura, “Survey of Vehicle IoT Bluetooth Devices,” in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Matsue, Japan, Nov. 2014, pp. 260–264. doi: 10.1109/SOCA.2014.20.

[46] K. B. Kelarestaghi, K. Heaslip, M. Khalilikhah, A. Fuentes, and V. Fessmann, “Intelligent Transportation System Security: Hacked Message Signs,” *SAE Int. J. Cybersecurity*, vol. 1, no. 2, pp. 75–90, Jun. 2018, doi: 10.4271/11-01-02-0004.

[47] C. Cerrudo, “Hacking US (and UK, Australia, France, etc.) Traffic Control Systems,” *IOActive*, Apr. 30, 2014. https://ioactive.com/hacking-us-and-uk-australia-france-etc/ (accessed Jan. 27, 2021).

[48] S. Bernstein and A. Blankstein, “Key signals targeted, officials say,” *Los Angeles Times*, Jan. 09, 2007. https://www.latimes.com/archives/la-xpm-2007-jan-09-me-trafficlights9-story.html (accessed Jan. 27, 2021).

[49] S. Grad, “Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced,” *LA Times Blogs - L.A. NOW*, Dec. 01, 2009. https://latimesblogs.latimes.com/lanow/2009/12/engineers-who-hacked-in-la-traffic-signal-computers-jamming-traffic-sentenced.html (accessed Jan. 27, 2021).

[50] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, “Green Lights Forever: Analyzing the Security of Traffic Infrastructure,” p. 10, 2014.

[51] C. Cerrudo, “Hacking US traffic control systems,” in *Proc. DEFCON*, 2014, pp. 1–5.

[52] C. Cerrudo, “An emerging US (and world) threat: Cities wide open to cyber attacks,” *Securing Smart Cities*, vol. 17, pp. 137–151, 2015.

[53] J. Blum and A. Eskandarian, “The threat of intelligent collisions,” *IT Prof.*, vol. 6, no. 1, pp. 24–29, Jan. 2004, doi: 10.1109/MITP.2004.1265539.

[54] S. Parkinson, P. Ward, K. Wilson, and J. Miller, “Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges,” *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 11, pp. 2898–2915, Nov. 2017, doi: 10.1109/TITS.2017.2665968.

[55] F. Ahmad, A. Adnane, V. Franqueira, F. Kurugollu, and L. Liu, “Man-In-The-Middle Attacks in Vehicular Ad-Hoc Networks: Evaluating the Impact of Attackers’ Strategies,” *Sensors*, vol. 18, no. 11, p. 4040, Nov. 2018, doi: 10.3390/s18114040.

[56] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip, and R. Gerdes, “Intelligent Transportation System Security: Impact-Oriented Risk Assessment of In-Vehicle Networks,” *IEEE Intell. Transport. Syst. Mag.*, pp. 1–1, 2019, doi: 10.1109/MITS.2018.2889714.

[57] Yeongkwun Kim, Injoo Kim, and C. Y. Shim, “A taxonomy for DOS attacks in VANET,” in *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, Incheon, South Korea, Sep. 2014, pp. 26–27. doi: 10.1109/ISCIT.2014.7011862.

[58] S. Biswas, J. Misic, and V. Misic, “DDoS attack on WAVE-enabled VANET through synchronization,” in *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, USA, Dec. 2012, pp. 1079–1084. doi: 10.1109/GLOCOM.2012.6503256.

[59] A. A. Ganin, A. C. Mersky, A. S. Jin, M. Kitsak, J. M. Keisler, and I. Linkov, “Resilience in Intelligent Transportation Systems (ITS),” *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 318–329, Mar. 2019, doi: 10.1016/j.trc.2019.01.014.

[60] M. Amoozadeh *et al.*, “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving,” *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 126–132, Jun. 2015, doi: 10.1109/MCOM.2015.7120028.

[61] J. Grover, V. Laxmi, and M. S. Gaur, “Attack models and infrastructure supported detection mechanisms for position forging attacks in vehicular ad hoc networks,” *CSIT*, vol. 1, no. 3, pp. 261–279, Sep. 2013, doi: 10.1007/s40012-013-0025-1.

[62] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla, “Congestion Attacks to Autonomous Cars Using Vehicular Botnets,” presented at the Workshop on Security of Emerging Networking Technologies, San Diego, CA, 2015. doi: 10.14722/sent.2015.23001.

[63] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Trans. on Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011, doi: 10.1109/TMC.2010.133.

[64] Joint Task Force Transformation Initiative, “Guide for conducting risk assessments,” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-30r1, 2012. doi: 10.6028/NIST.SP.800-30r1.

[65] The White House, “Presidential Policy Directive -- Critical Infrastructure Security and Resilience,” *whitehouse.gov*, Feb. 12, 2013. https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil (accessed Jan. 27, 2021).

[66] “ISO 27005:2018.” International Organization of Standardization. Accessed: Apr. 19, 2021. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:27005:ed-3:v1:en

[67] “What is OMNeT++?” https://omnetpp.org/intro/ (accessed Jul. 12, 2021).

[68] P. A. Lopez *et al.*, “Microscopic Traffic Simulation using SUMO,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, Nov. 2018, pp. 2575–2582. doi: 10.1109/ITSC.2018.8569938.

[69] S. Krauß, “Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics,” 1998.

[70] “TraCI - SUMO Documentation.” https://sumo.dlr.de/docs/TraCI.html (accessed Jul. 12, 2021).

[71] “IEEE Std 1609.4-2016 (Revision of IEEE Std 1609.4-2010) IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation,” p. 94.

[72] S. Eichler, “Performance Evaluation of the IEEE 802.11p WAVE Communication Standard,” in *2007 IEEE 66th Vehicular Technology Conference*, Baltimore, MD, USA, Sep. 2007, pp. 2199–2203. doi: 10.1109/VETECF.2007.461.

[73] Federal Highway Administration, “Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software 2019 Update to the 2004 Version.” Apr. 2019.

APPENDIX A – Calculation Derivations

Note: All velocity profiles are for visual representation and are NOT to scale

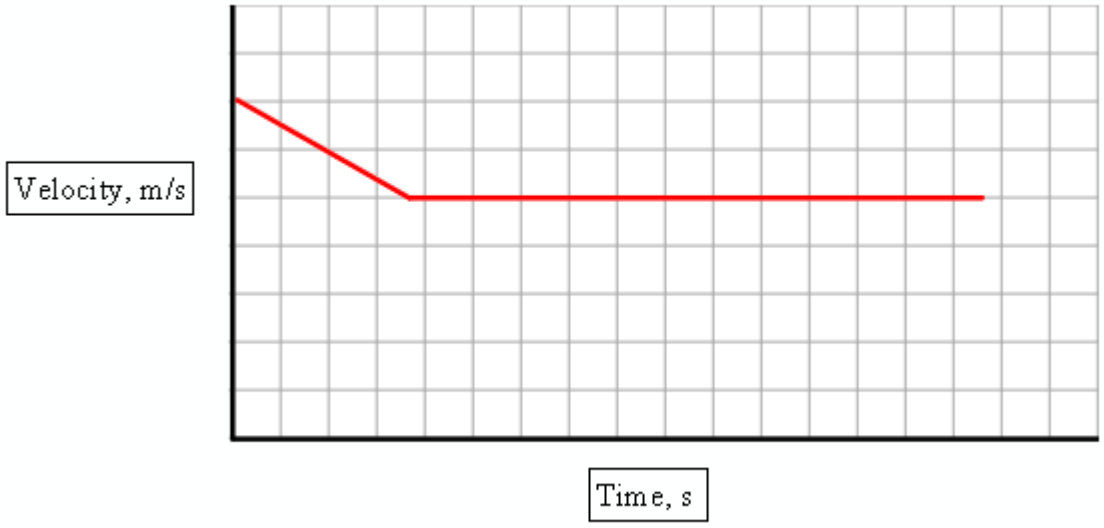
To begin the calculation derivations, the maximum allowable delay a vehicle can incur without decelerating pas the final velocity must be calculated. This value is a vehicle's longest arrival time minus the vehicle's shortest arrival time.

$$t_{Max\ Delay} = t_{longest\ arrival} - t_{shortest\ arrival} \qquad \text{Equation 1.0}$$

The longest arrival and shortest arrival times will differ based on the vehicle's current speed and that speeds relation with its route velocity. The first set of derivations are based on a vehicle making a reservation at or above its route velocity.

Equation 2.0

The longest arrival time will be defined by a vehicle immediately decelerating to its route velocity and maintaining this speed until the intersection. This velocity profile is defined below:



Fi

rst Leg: Deceleration

$$V_{end} = V_{start} + decel * t_1 \qquad 1.0$$

$$d_1 = \frac{V_{start} + V_{end}}{2} * t_1 \qquad 1.1$$

First Leg: Straightaway

$$d_2 = V_{end} * t_2 \qquad 1.2$$

Continuity:

$$t_{longest\ arrival} = t_1 + t_2 \quad (1.3)$$

$$d_t = d_1 + d_2 \quad (1.4)$$

Derivation:

1.0 $$t_1 = \frac{v_{end} - v_{start}}{decel} \quad (1.5)$$

1.1, 1.5 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{start}}{decel}\right) \quad (1.6)$$

$$d_1 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} \quad (1.7)$$

1.4 $$d_2 = d_t - d_1 \quad (1.8)$$

1.7, 1.8 $$d_2 = d_t - \frac{{v_{end}}^2}{2decel} + \frac{{v_{start}}^2}{2decel} \quad (1.9)$$

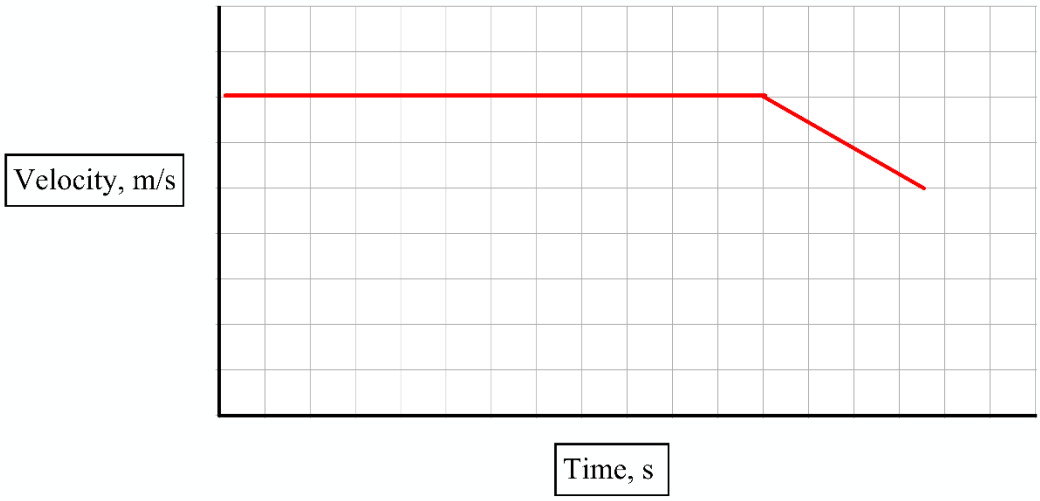
1.2 $$t_2 = \frac{d_2}{v_{end}} \quad (1.10)$$

1.3, 1.5, 1.10 $$t_{longest\ arrival} = \frac{d_2}{v_{end}} + \frac{v_{end} - v_{start}}{decel} \quad (1.11)$$

1.9, 1.11 $$t_{longest\ arival} = \frac{d_t}{v_{end}} + \frac{{v_{start}}^2}{2decel * v_{end}} + \frac{v_{end}}{2decel} - \frac{v_{start}}{decel} \quad \textbf{Eqn. 2.0}$$

Equation 2.1

This vehicle's fastest arrival time will be defined by maintaining its entry velocity until the last moment prior to the intersection when it must slow down to enter at its route velocity. The velocity profile is shown below:



First Leg: Straightaway

$$d_1 = v_{start} * t_1 \quad (2.0)$$

Second Leg: Deceleration

$$v_{end} = v_{start} + decel * t_2 \quad (2.1)$$

$$d_2 = \frac{v_{start} + v_{end}}{2} * t_2 \quad (2.2)$$

Continuity:

$$t_{fastest\ arrival} = t_1 + t_2 \quad (2.3)$$

$$d_t = d_1 + d_2 \quad (2.4)$$

Derivation:

2.1 $$t_2 = \frac{v_{end} - v_{start}}{decel} \quad (2.5)$$

2.2, 2.5 $$d_2 = \left(\frac{v_{start}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{start}}{decel}\right) \quad (2.6)$$

$$d_2 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} \quad (2.7)$$

2.4 $$d_1 = d_t - d_2 \quad (2.8)$$

2.7, 2.8 $$d_1 = d_t - \frac{{v_{end}}^2}{2decel} + \frac{{v_{start}}^2}{2decel} \quad (2.9)$$

2.0 $$t_1 = \frac{d_1}{v_{start}} \quad (2.10)$$

2.3, 2.5, 2.10 $$t_{fastest\ arrival} = \frac{d_1}{v_{start}} + \frac{v_{end} - v_{start}}{decel} \quad (2.11)$$

2.9, 2.11 $$t_{fastest\ arival} = \frac{d_t}{v_{start}} - \frac{{v_{end}}^2}{2decel * v_{start}} - \frac{v_{start}}{2decel} + \frac{v_{end}}{decel} \quad \textbf{Eqn. 2.1}$$

Equation 1.1

$$t_{Max\ Delay} = t_{longest\ arrival} - t_{fastest\ arrival} \quad \text{1.0}$$

1.0, Eqn. 2.0, Eqn. 2.1

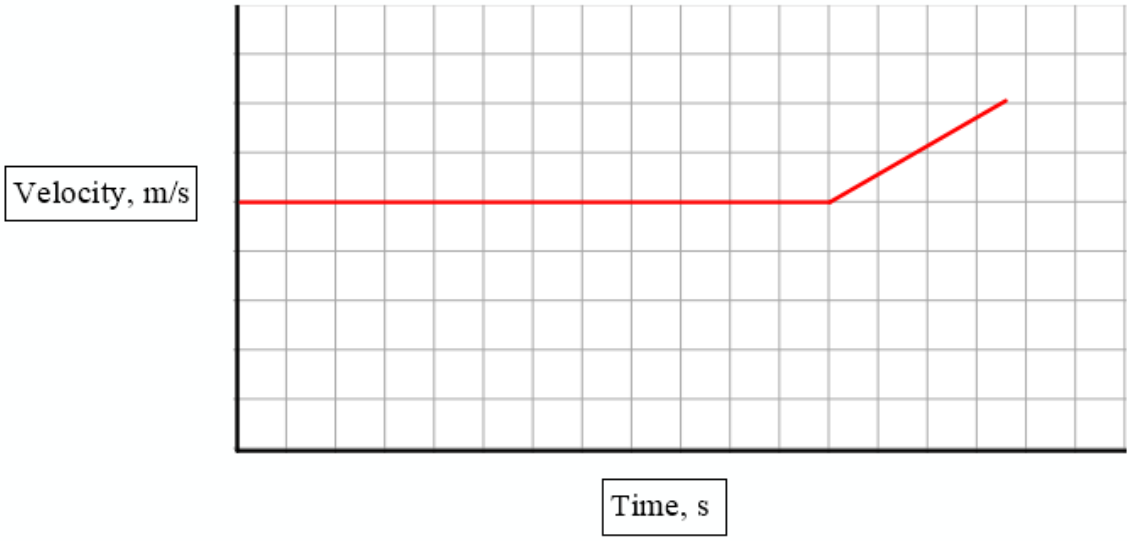
$$t_{Max\ Delay} = \left(\frac{d_t}{v_{end}} + \frac{v_{start}^{\ 2}}{2decel * v_{end}} + \frac{v_{end}}{2decel} - \frac{v_{start}}{decel}\right) - \left(\frac{d_t}{v_{start}} - \frac{v_{end}^{\ 2}}{2decel * v_{start}} - \frac{v_{start}}{2decel} + \frac{v_{end}}{decel}\right)$$

$$t_{Max\ Delay} = \frac{d_t}{v_{end}} - \frac{d_t}{v_{start}} - \frac{v_{end}}{2a_{dec}} - \frac{v_{start}}{2a_{dec}} + \frac{v_{start}^{\ 2}}{2a_{dec}v_{end}} + \frac{v_{end}^{\ 2}}{2a_{dec}v_{start}} \quad \textbf{Eqn. 1.1}$$

The second set of derivations pertains to a vehicle that has received directions while traveling below its intended route velocity.

Equation 3.0

The longest arrival time for a vehicle in this scenario would be for the vehicle to maintain its speed until prior to the intersection, then accelerate to its route velocity to reach the intersection at the appropriate time. The velocity profile for this scenario is shown below:



First Leg: Straightaway

$$d_1 = v_{start} * t_1 \quad (3.0)$$

Second Leg: Acceleration

$$v_{end} = v_{start} + accel * t_2 \quad (3.1)$$

$$d_2 = \frac{v_{start} + v_{end}}{2} * t_2 \quad (3.2)$$

Continuity:

$$t_{longest\ arrival} = t_1 + t_2 \quad (3.3)$$

$$d_t = d_1 + d_2 \quad (3.4)$$

Derivation:

3.1

$$t_2 = \frac{v_{end} - v_{start}}{accel} \quad (3.5)$$

3.2, 3.5 $$d_2 = \left(\frac{v_{start}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{accel} - \frac{v_{start}}{accel}\right)$$ 3.6

$$d_2 = \frac{{v_{end}}^2}{2accel} - \frac{{v_{start}}^2}{2accel}$$ 3.7

3.4 $$d_1 = d_t - d_2$$ 3.8

3.7, 3.8 $$d_1 = d_t - \frac{{v_{end}}^2}{2accel} + \frac{{v_{start}}^2}{2accel}$$ 3.9

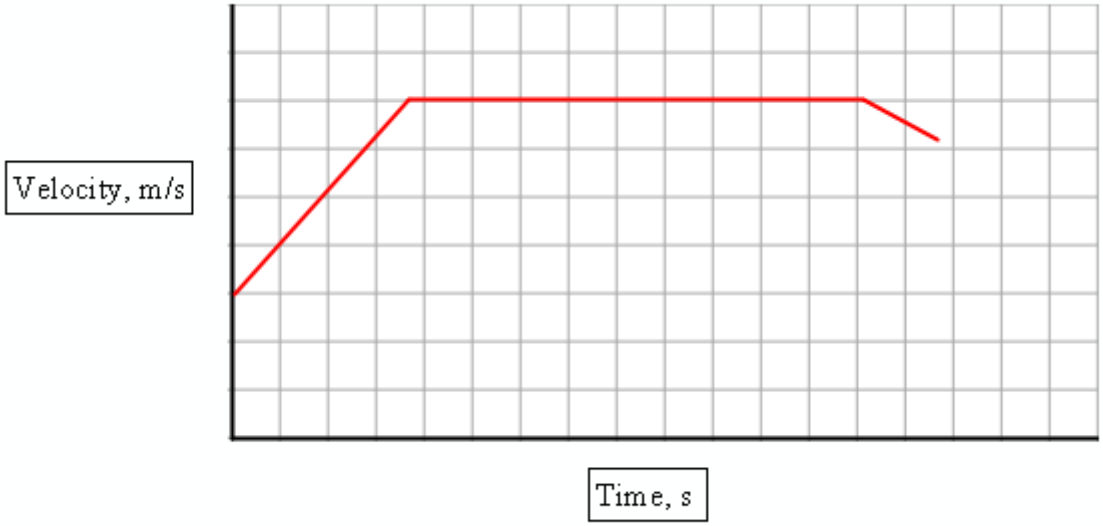
3.0 $$t_1 = \frac{d_1}{v_{start}}$$ 3.10

3.3, 3.5, 3.10 $$t_{longest\ arrival} = \frac{d_1}{v_{start}} + \frac{v_{end} - v_{start}}{accel}$$ 3.11

3.9, 3.11 $$t_{longest\ arival} = \frac{d_t}{v_{start}} - \frac{{v_{end}}^2}{2accel * v_{start}} - \frac{v_{start}}{2accel} + \frac{v_{end}}{accel}$$ **Eqn. 3.0**

Equation 3.1

The overall idea remains the same, but the fastest arrival time for this vehicle is different as it is allowed to accelerate up to / past its route velocity and decelerate prior to the intersection if necessary. The velocity profile is shown below:



First Leg: Acceleration

$$v_{mid} = v_{start} + accel * t_1 \quad (4.0)$$

$$d_1 = \frac{v_{start} + v_{mid}}{2} * t_1 \quad (4.1)$$

Second Leg: Straightaway

$$d_2 = v_{mid} * t_2 \quad (4.2)$$

Third Leg: Deceleration

$$v_{end} = v_{mid} + decel * t_3 \quad (4.3)$$

$$d_3 = \frac{v_{mid} + v_{end}}{2} * t_3 \quad (4.4)$$

Continuity:

$$t_{fastest\ arrival} = t_1 + t_2 + t_3 \quad (4.5)$$

$$d_t = d_1 + d_2 + d_3 \quad (4.6)$$

Derivation:

4.0 $$t_1 = \frac{v_{mid} - v_{start}}{accel}$$ 4.7

4.1 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{mid}}{2}\right)\left(\frac{v_{mid}}{accel} - \frac{v_{start}}{accel}\right)$$ 4.8

4.8 $$d_1 = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel}$$ 4.9

4.3 $$t_3 = \frac{v_{end} - v_{mid}}{decel}$$ 4.10

4.4 $$d_3 = \left(\frac{v_{mid}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{mid}}{decel}\right)$$ 4.11

4.11 $$d_3 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel}$$ 4.12

4.6 $$d_t = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel} + V_{mid} * t_2 + \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel}$$ 4.13

4.13 $$t_2 = \frac{d_t}{v_{mid}} - \frac{v_{mid}}{2accel} + \frac{{v_{start}}^2}{2accel * v_{mid}} - \frac{{v_{end}}^2}{2decel * v_{mid}} + \frac{v_{mid}}{2decel}$$ 4.14

4.5, 4.7, 4.10, 4.14 $$t_{fastest\ arrival} = \frac{d_t}{v_{mid}} - \frac{v_{start}}{accel} + \frac{v_{mid}}{2accel} + \frac{{v_{start}}^2}{2accel * v_{mid}} - \frac{{v_{end}}^2}{2decel * v_{mid}} + \frac{v_{end}}{decel} - \frac{v_{mid}}{2decel}$$ **Eqn. 3.1**

Equation 1.2

$$t_{Max\ Delay} = t_{longest\ arrival} - t_{fastest\ arrival} \quad \text{1.0}$$

1.0, Eqn. 3.0, Eqn. 3.1

$$t_{Max\ Delay} = \left(\frac{d_t}{v_{start}} - \frac{{v_{end}}^2}{2accel * v_{start}} - \frac{v_{start}}{2accel} + \frac{v_{end}}{accel}\right) - \left(\frac{d_t}{v_{mid}} - \frac{v_{start}}{accel} + \frac{v_{mid}}{2accel} + \frac{{v_{start}}^2}{2accel * v_{mid}} - \frac{{v_{end}}^2}{2decel * v_{mid}} + \frac{v_{end}}{decel} - \frac{v_{mid}}{2decel}\right)$$

$$t_{Max\ Delay} = \frac{d_t}{v_{start}} - \frac{{v_{end}}^2}{2accel * v_{start}} + \frac{v_{end}}{accel} - \frac{v_{mid}}{2accel} + \frac{v_{mid}}{2decel} + \frac{v_{start}}{2accel} - \frac{{v_{start}}^2}{2accel * v_{mid}} + \frac{{v_{end}}^2}{2decel * v_{mid}} - \frac{v_{end}}{decel} - \frac{d_t}{v_{mid}} \quad \textbf{Eqn. 1.2}$$

With the maximum delay calculated, now velocity calculations can be made.

Equation 4.0 – 4.4

The first velocity possibility occurs when a vehicle needs to introduce delay into its trip to reach the intersection at the appropriate time. This occurs when the directed time is longer than the max delay and can occur at any velocity relative to the final velocity. The velocity profile is shown below:

First Leg: Deceleration

$$v_{mid} = v_{start} + decel * t_1 \quad (5.0)$$

$$d_1 = \frac{v_{start} + v_{mid}}{2} * t_1 \quad (5.1)$$

Second Leg: Straightaway

$$d_2 = v_{mid} * t_2 \quad (5.2)$$

Third Leg: Acceleration

$$v_{end} = v_{mid} + accel * t_3 \quad (5.3)$$

$$d_3 = \frac{v_{mid} + v_{end}}{2} * t_3 \quad (5.4)$$

Continuity:

$$t_{total} = t_1 + t_2 + t_3 \quad (5.5)$$

$$d_t = d_1 + d_2 + d_3 \qquad 5.6$$

Derivation:

5.0 $$t_1 = \frac{v_{mid} - v_{start}}{decel} \qquad 5.7$$

5.1 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{mid}}{2}\right)\left(\frac{v_{mid}}{decel} - \frac{v_{start}}{decel}\right) \qquad 5.8$$

5.8 $$d_1 = \frac{{v_{mid}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} \qquad 5.9$$

5.3 $$t_3 = \frac{v_{end} - v_{mid}}{accel} \qquad 5.10$$

5.4 $$d_3 = \left(\frac{v_{mid}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{accel} - \frac{v_{mid}}{accel}\right) \qquad 5.11$$

5.11 $$d_3 = \frac{{v_{end}}^2}{2accel} - \frac{{v_{mid}}^2}{2accel} \qquad 5.12$$

5.6 $$d_t = \frac{{v_{mid}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} + V_{mid} * t_2 + \frac{{v_{end}}^2}{2accel} - \frac{{v_{mid}}^2}{2accel} \qquad 5.13$$

5.13 $$t_2 = \frac{d_t}{v_{mid}} - \frac{v_{mid}}{2decel} + \frac{{v_{start}}^2}{2decel * v_{mid}} - \frac{{v_{end}}^2}{2accel * v_{mid}} + \frac{v_{mid}}{2accel} \qquad 5.14$$

5.5, 5.7, 5.10, 5.14 $$t_{total} = \frac{v_{mid}}{2decel} - \frac{v_{mid}}{2accel} - \frac{v_{start}}{decel} + \frac{d_t}{v_{mid}} + \frac{{v_{start}}^2}{2decel * v_{mid}} - \frac{{v_{end}}^2}{2accel * v_{mid}} + \frac{v_{end}}{accel} \qquad 5.15$$

5.15 *Set to 0 and multiply by V_{mid} to isolate terms* 5.16

5.16 $$0 = \frac{{v_{mid}}^2}{2decel} - \frac{{v_{mid}}^2}{2accel} - \frac{v_{start} v_{mid}}{decel} + d_t + \frac{{v_{start}}^2}{2decel} - \frac{{v_{end}}^2}{2accel} + \frac{v_{end} v_{mid}}{accel} - t_{total}\, v_{mid} \qquad 5.17$$

5.17 $${v_{mid}}^2\left(\frac{1}{2decel} - \frac{1}{2accel}\right) = a \qquad \textbf{Eqn. 4.1}$$

5.17 $$v_{mid}\left(-\frac{v_{start}}{decel} + \frac{v_{end}}{accel} - t_{total}\right) = b \qquad \textbf{Eqn. 4.2}$$

5.17 $$d_t + \frac{{v_{start}}^2}{2decel} - \frac{{v_{end}}^2}{2accel}) = c \qquad \textbf{Eqn. 4.3}$$

$$v_{mid} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad \textbf{Eqn. 4.0}$$

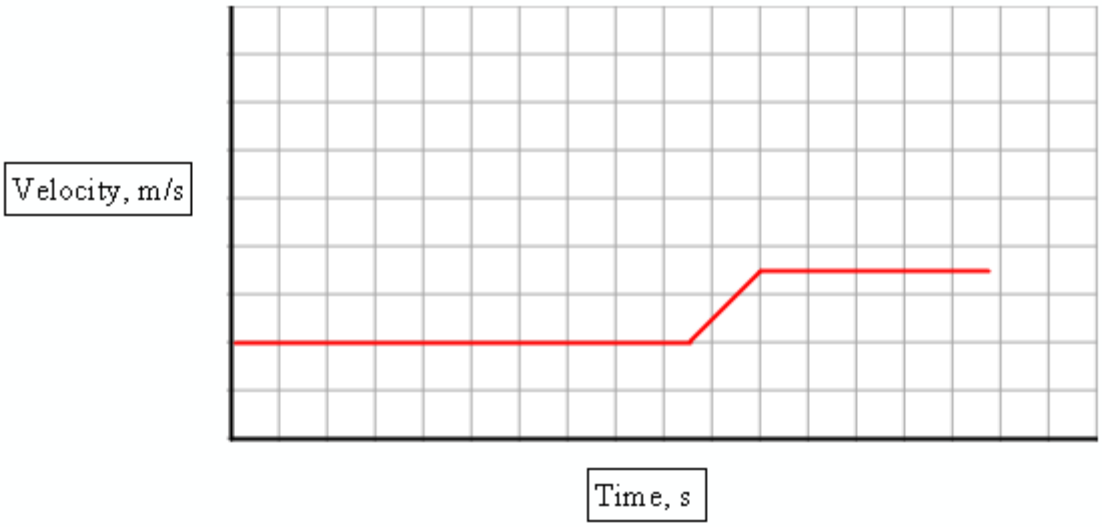
The vehicle will receive a self-message to accelerate to its route velocity just prior to arriving at the intersection. This time will be equal to the time it takes to reach the intersection minus the time it takes for the vehicle to accelerate. The formulation is shown below.

4.5, 4.10 $$t_{accelerate} = t_{total} - \frac{v_{end}}{accel} + \frac{v_{mid}}{accel}$$ **Eqn. 4.4**

The next series of three velocity profiles only occur if a vehicle receives directions and its is traveling below the required route velocity.

Equation 5.0

If a vehicle enters the radius of communication during the recovery period, there is a chance this vehicle will reach the end of the vehicle queue and be forced to slow down to match traffic. A reservation cannot be made while heavy deceleration is occurring. However, once the vehicle has slowed down it can make a reservation. If the vehicle is not required to slow down further (Equation 4.0), it will maintain its speed for a stretch, then accelerate to its route velocity when safe to do so and maintain this speed until it reaches the intersection at its reservation. The velocity profile for this maneuver is shown below:



Since there is no velocity calculated in this velocity profile, the main calculation is for time to accelerate.

First Leg: Straightaway

$$d_1 = v_{start} * t_1 \quad (6.0)$$

Second Leg: Acceleration

$$v_{end} = v_{start} + accel * t_2 \quad (6.1)$$

$$d_2 = \frac{v_{start} + v_{end}}{2} * t_2 \quad (6.2)$$

Third Leg: Straightaway

$$d_3 = V_{end} * t_3 \quad (6.3)$$

Continuity:

$$t_{total} = t_1 + t_2 + t_3 \quad (6.4)$$

$$d_t = d_1 + d_2 + d_3 \quad (6.5)$$

Derivation:

6.0 $$t_1 = \frac{d_1}{v_{start}} \quad (6.6)$$

6.1 $$t_2 = \frac{v_{end} - v_{start}}{accel} \quad (6.7)$$

6.2, 6.7 $$d_2 = \left(\frac{v_{start}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{accel} - \frac{v_{start}}{decel}\right) \quad (6.8)$$

6.8 $$d_2 = \frac{{v_{end}}^2}{2accel} - \frac{{v_{start}}^2}{2accel} \quad (6.9)$$

6.3 $$t_3 = \frac{d_3}{v_{end}} \quad (6.10)$$

6.4, 6.6, 6.7, 6.10 $$t_t = \frac{d_1}{v_{start}} + \frac{v_{end}}{accel} - \frac{v_{start}}{accel} + \frac{d_3}{v_{end}} \quad (6.11)$$

6.11 $$d_1 = t_t v_{start} - \frac{v_{end} v_{start}}{accel} + \frac{{v_{start}}^2}{accel} - \frac{d_3 v_{start}}{v_{end}} \quad (6.12)$$

6.3, 6.5, 6.12 $$d_t = t_t v_{start} - \frac{v_{end} v_{start}}{accel} + \frac{{v_{start}}^2}{2accel} - t_3 v_{start} + \frac{{v_{end}}^2}{2accel} + t_3 v_{end} \quad (6.13)$$

6.13 $$t_3 = \frac{d_t - t_t v_{start} + \frac{v_{end} v_{start}}{accel} - \frac{{v_{start}}^2}{2accel} - \frac{{v_{end}}^2}{2accel}}{v_{end} - v_{start}} \quad \textbf{Eqn. 5.0}$$

6.1 $$t_2 = \frac{v_{end} - v_{start}}{accel} \quad \textbf{Eqn. 5.1}$$

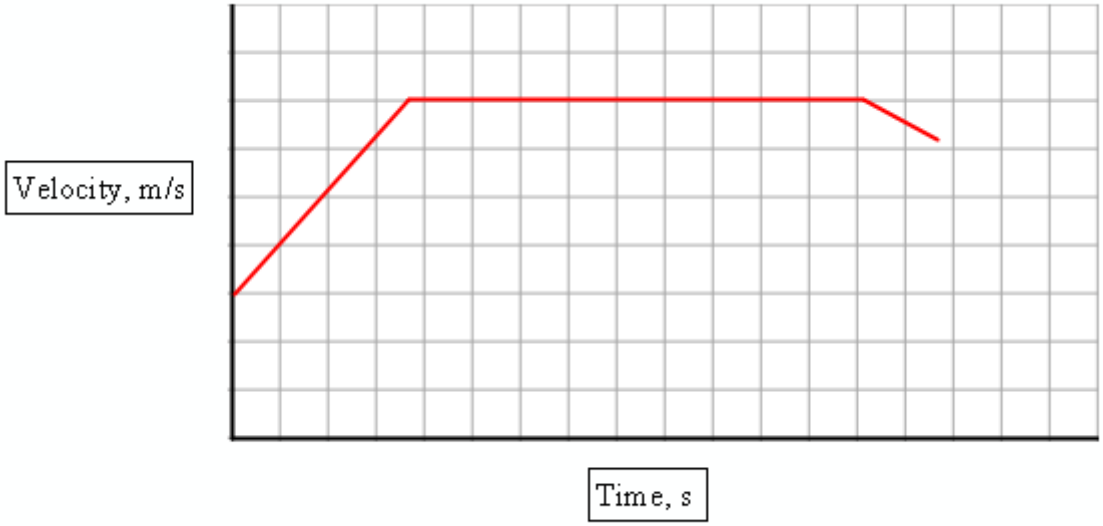
Eqn. 5.0, $$t_1 = t_t - t_2 - t_3 \quad \textbf{Eqn. 5.2}$$

Eqn. 5.1, $$t_1 = time\ to\ accelerate$$

6.4

Equation 6.1 -6.4

The next velocity scenario occurs when a vehicle has the time, space, and directed velocity to accelerate up to / past the directed velocity then decelerate to its route velocity if needed. The velocity profile is identical to that used for Equation 3.1 and shown below:



First Leg: Acceleration

$$v_{mid} = v_{start} + accel * t_1 \quad (7.0)$$

$$d_1 = \frac{v_{start} + v_{mid}}{2} * t_1 \quad (7.1)$$

Second Leg: Straightaway

$$d_2 = v_{mid} * t_2 \quad (7.2)$$

Third Leg: Deceleration

$$v_{end} = v_{mid} + decel * t_3 \quad (7.3)$$

$$d_3 = \frac{v_{mid} + v_{end}}{2} * t_3 \quad (7.4)$$

Continuity:

$$t_t = t_1 + t_2 + t_3 \quad (7.5)$$

$$d_t = d_1 + d_2 + d_3 \quad (7.6)$$

Derivation:

7.0 $t_1 = \frac{v_{mid} - v_{start}}{accel}$ 7.7

7.1 $d_1 = \left(\frac{v_{start}}{2} + \frac{v_{mid}}{2}\right)\left(\frac{v_{mid}}{accel} - \frac{v_{start}}{accel}\right)$ 7.8

7.8 $d_1 = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel}$ 7.9

7.3 $t_3 = \frac{v_{end} - v_{mid}}{decel}$ 7.10

7.4 $d_3 = \left(\frac{v_{mid}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{mid}}{decel}\right)$ 7.11

7.11 $d_3 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel}$ 7.12

7.6 $d_t = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel} + v_{mid} * t_2 + \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel}$ 7.13

7.13 $t_2 = \frac{d_t}{v_{mid}} - \frac{v_{mid}}{2accel} + \frac{{v_{start}}^2}{2accel * v_{mid}} - \frac{{v_{end}}^2}{2decel * v_{mid}} + \frac{v_{mid}}{2decel}$ 7.14

7.5, 7.7, 7.10, 7.14 $t_t = \frac{d_t}{v_{mid}} - \frac{v_{start}}{accel} + \frac{v_{mid}}{2accel} + \frac{{v_{start}}^2}{2accel v_{mid}} - \frac{{v_{end}}^2}{2decel * v_{mid}} + \frac{v_{end}}{decel} - \frac{v_{mid}}{2decel}$ 7.15

Set to 0 and multiply by V_{mid} to isolate terms

7.15 $0 = \frac{{v_{mid}}^2}{2accel} - \frac{v_{start} v_{mid}}{accel} + d_t + \frac{{v_{start}}^2}{2accel} - \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel} + \frac{v_{end} v_{mid}}{decel} - t_t v_{mid}$ 7.16

7.16 $v_{mid}{}^2\left(\frac{1}{2accel} - \frac{1}{2decel}\right) = a$ **Eqn. 6.1**

7.16 $v_{mid}\left(\frac{v_{end}}{decel} - \frac{v_{start}}{accel} - t_{total}\right) = b$ **Eqn. 6.2**

7.16 $\left(d_t + \frac{{v_{start}}^2}{2accel} - \frac{{v_{end}}^2}{2decel}\right) = c$ **Eqn. 6.3**

$v_{mid} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ **Eqn. 4.0**

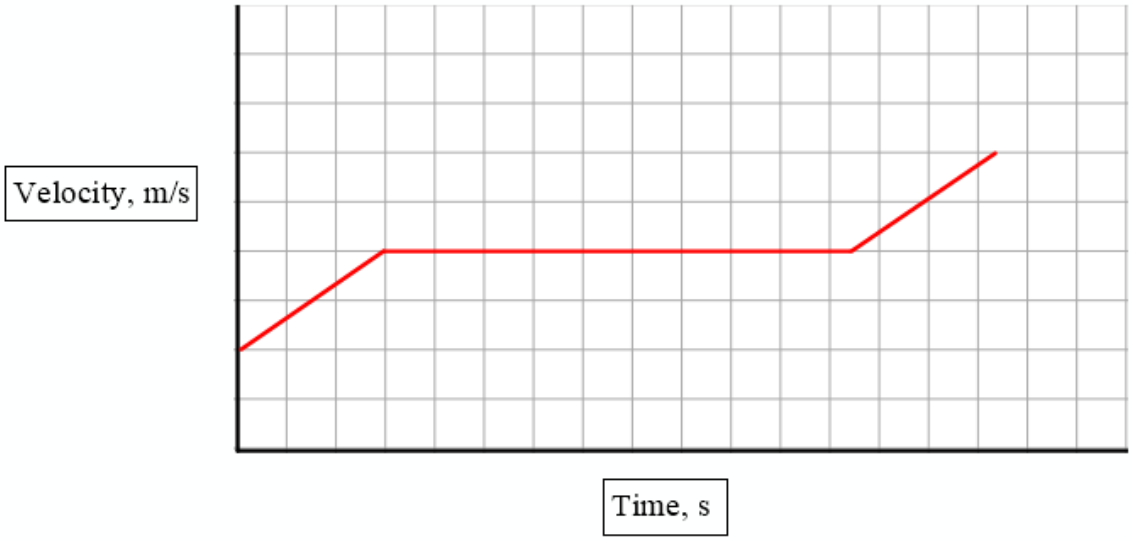
The vehicle will receive a self-message to decelerate to its route velocity just prior to arriving at the intersection. This time will be equal to the time it takes to reach the intersection minus the time it takes for the vehicle to decelerate. The formulation is shown below.

7.5, 7.10

$$t_{decelerate} = t_{total} - \frac{v_{end}}{decel} + \frac{v_{mid}}{decel}$$ **Eqn. 6.4**

Equation 6.5 – 6.6

The last of the velocities possible under the presented velocity scenarios occurs when a vehicle receives a direction and can accelerate, but the velocity is not higher than its route velocity, meaning it will have to accelerate a second time. This velocity profile is shown below:



First Leg: Acceleration

$$v_{mid} = v_{start} + accel * t_1 \quad (8.0)$$

$$d_1 = \frac{v_{start} + v_{mid}}{2} * t_1 \quad (8.1)$$

Second Leg: Straightaway

$$d_2 = v_{mid} * t_2 \quad (8.2)$$

Third Leg: Acceleration

$$v_{end} = v_{mid} + accel * t_3 \quad (8.3)$$

$$d_3 = \frac{v_{mid} + v_{end}}{2} * t_3 \quad (8.4)$$

Continuity:

$$t_t = t_1 + t_2 + t_3 \quad (8.5)$$

$$d_t = d_1 + d_2 + d_3 \quad (8.6)$$

Derivation:

8.0 $$t_1 = \frac{v_{mid} - v_{start}}{accel}$$ 8.7

8.1 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{mid}}{2}\right)\left(\frac{v_{mid}}{accel} - \frac{v_{start}}{accel}\right)$$ 8.8

8.8 $$d_1 = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel}$$ 8.9

8.3 $$t_3 = \frac{v_{end} - v_{mid}}{accel}$$ 8.10

8.4 $$d_3 = \left(\frac{v_{mid}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{accel} - \frac{v_{mid}}{accel}\right)$$ 8.11

8.11 $$d_3 = \frac{{v_{end}}^2}{2accel} - \frac{{v_{mid}}^2}{2accel}$$ 8.12

8.6 $$d_t = \frac{{v_{mid}}^2}{2accel} - \frac{{v_{start}}^2}{2accel} + v_{mid} * t_2 + \frac{{v_{end}}^2}{2accel} - \frac{{v_{mid}}^2}{accel}$$ 8.13

8.13 $$t_2 = \frac{d_t}{v_{mid}} + \frac{{v_{start}}^2}{2accel v_{mid}} - \frac{{v_{end}}^2}{2accel v_{mid}}$$ 8.14

8.5, 8.7, 8.10, 8.14 $$t_t = \frac{d_t}{v_{mid}} - \frac{v_{start}}{accel} - \frac{{v_{end}}^2}{2accel v_{mid}} + \frac{{v_{start}}^2}{2accel v_{mid}} + \frac{v_{end}}{accel}$$ 8.15

Set to 0 and multiply by V_{mid} to isolate terms

8.15 $$0 = d_t - \frac{v_{start} v_{mid}}{accel} - \frac{{v_{end}}^2}{2accel} + \frac{{v_{start}}^2}{2accel} + \frac{v_{end} v_{mid}}{accel} - t_t v_{mid}$$ 8.16

8.16 $$v_{mid} = \frac{d_t + \frac{{v_{start}}^2}{2accel} - \frac{{v_{end}}^2}{2accel}}{t_t - \frac{v_{end}}{accel} + \frac{v_{start}}{acel}}$$ **Eqn. 6.5**

The vehicle will accelerate just prior to reaching the intersection to reach its stated velocity. This time calculation is shown below.

8.5, 8.10 $$t_{accelerate} = t_{total} - \frac{v_{end}}{accel} + \frac{v_{mid}}{accel}$$ **Eqn. 6.6**

Equation 7.0 and 7.1

The final main velocity profile is the inverse of the one calculated in Equation 6.6. The vehicle will receive its directions, decelerate, and then decelerate a second time to reach the intersection at its allotted time. The velocity profile is shown below:

First Leg: Deceleration

$$v_{mid} = v_{start} + decel * t_1 \quad (9.0)$$

$$d_1 = \frac{v_{start} + v_{mid}}{2} * t_1 \quad (9.1)$$

Second Leg: Straightaway

$$d_2 = v_{mid} * t_2 \quad (9.2)$$

Third Leg: Deceleration

$$v_{end} = v_{mid} + decel * t_3 \quad (9.3)$$

$$d_3 = \frac{v_{mid} + v_{end}}{2} * t_3 \quad (9.4)$$

Continuity:

$$t_t = t_1 + t_2 + t_3 \quad (9.5)$$

$$d_t = d_1 + d_2 + d_3 \quad (9.6)$$

Derivation:

9.0 $$t_1 = \frac{v_{mid} - v_{start}}{decel} \quad (9.7)$$

9.1 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{mid}}{2}\right)\left(\frac{v_{mid}}{decel} - \frac{v_{start}}{decel}\right) \quad (9.8)$$

9.8 $$d_1 = \frac{{v_{mid}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} \quad (9.9)$$

9.3 $$t_3 = \frac{v_{end} - v_{mid}}{decel} \quad (9.10)$$

9.4 $$d_3 = \left(\frac{v_{mid}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{mid}}{decel}\right) \quad (9.11)$$

9.11 $$d_3 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel} \quad (9.12)$$

9.6 $$d_t = \frac{{v_{mid}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} + v_{mid} * t_2 + \frac{{v_{end}}^2}{2decel} - \frac{{v_{mid}}^2}{2decel} \quad (9.13)$$

9.13 $$t_2 = \frac{d_t}{v_{mid}} + \frac{{v_{start}}^2}{2decel v_{mid}} - \frac{{v_{end}}^2}{2decel v_{mid}} \quad (9.14)$$

9.5, 9.7, 9.10, 9.14 $$t_t = \frac{d_t}{v_{mid}} - \frac{v_{start}}{decel} - \frac{{v_{end}}^2}{2decel v_{mid}} + \frac{{v_{start}}^2}{2decel v_{mid}} + \frac{v_{end}}{decel} \quad (9.15)$$

Set to 0 and multiply by V_{mid} to isolate terms

9.15 $$0 = d_t - \frac{v_{start} v_{mid}}{decel} - \frac{{v_{end}}^2}{2decel} + \frac{{v_{start}}^2}{2decel} + \frac{v_{end} v_{mid}}{decel} - t_t v_{mid} \quad (9.16)$$

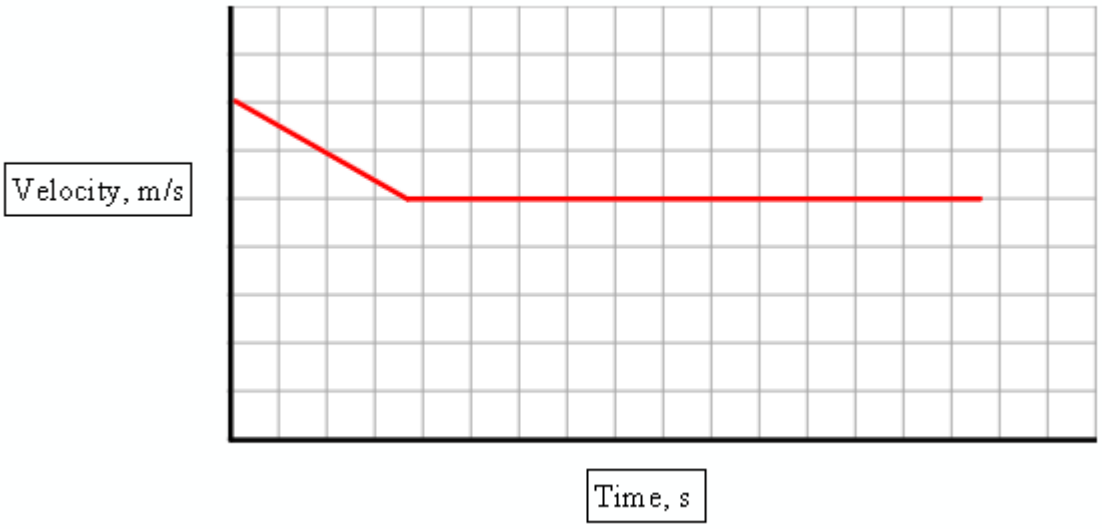
9.16 $$v_{mid} = \frac{d_t + \frac{{v_{start}}^2}{2decel} - \frac{{v_{end}}^2}{2decel}}{t_t - \frac{v_{end}}{decel} + \frac{v_{start}}{decel}} \quad \textbf{Eqn. 7.0}$$

The vehicle will accelerate just prior to reaching the intersection to reach its stated velocity. This time calculation is shown below.

9.5, 9.10 $$t_{decelerate} = t_{total} - \frac{v_{end}}{decel} + \frac{v_{mid}}{decel} \quad \textbf{Eqn. 7.1}$$

Equation 8.0 – 8.3

The final possible velocity occurs in the instance when a collision occurs, and a vehicle does not have the proper time or space to make a proper adjustment. In this instance, a vehicle will slow down to a calculated velocity to make sure they arrive at the intersection at the correct time. This is a rare occasion, and the worst-case scenario, but this emergency velocity allows a vehicle a quick reaction in the instance of a collision in front of it. The vehicle will accelerate once it has reached the intersection to its max allowed velocity. This velocity curve is identical to Equation 2 (see below), but the end velocity is not equal to the route velocity.



First Leg: Deceleration

$$v_{end} = v_{start} + decel * t_1 \quad (10.0)$$

$$d_1 = \frac{v_{start} + v_{end}}{2} * t_1 \quad (10.1)$$

First Leg: Straightaway

$$d_2 = v_{end} * t_2 \quad (10.2)$$

Continuity:

$$t_{total} = t_1 + t_2 \quad (10.3)$$

$$d_t = d_1 + d_2 \quad (10.4)$$

Derivation:

10.0 $$t_1 = \frac{v_{end} - v_{start}}{decel}$$ 10.5

10.1, 10.5 $$d_1 = \left(\frac{v_{start}}{2} + \frac{v_{end}}{2}\right)\left(\frac{v_{end}}{decel} - \frac{v_{start}}{decel}\right)$$ 10.6

$$d_1 = \frac{{v_{end}}^2}{2decel} - \frac{{v_{start}}^2}{2decel}$$ 10.7

10.2,10.4, 10.7 $$d_t = \frac{{v_{end}}^2}{2decel} - \frac{{v_{start}}^2}{2decel} + v_{end}t_2$$ 10.8

10.8 $$t_2 = \frac{d_t}{v_{end}} - \frac{v_{end}}{2decel} + \frac{{v_{start}}^2}{2decel}$$ 10.9

10.3, 10.5 $$t_t = \frac{v_{end}}{2decel} - \frac{v_{start}}{decel} + \frac{d_t}{v_{end}} + \frac{{v_{start}}^2}{2decel}$$ 10.10

Set to 0 and multiply by V_{end} to isolate terms

10.10 $$0 = \frac{{v_{end}}^2}{2decel} - \frac{v_{start}v_{end}}{decel} + d_t - \frac{{v_{start}}^2}{2decel} - t_t v_{end}$$

7.16 $${v_{end}}^2\left(\frac{1}{2decel}\right) = a$$ **Eqn. 8.1**

7.16 $$v_{end}\left(-\frac{v_{start}}{decel} - t_{total}\right) = b$$ **Eqn. 8.2**

7.16 $$\left(d_t - \frac{{v_{start}}^2}{2decel}\right) = c$$ **Eqn. 8.3**

$$v_{end} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$ **Eqn. 4.0**