A Dependable Detection Mechanism for Intersection Management of Connected Autonomous Vehicles

Rachel Dedinsky
Arizona State University,
660 S Mill Ave, Tempe, AZ, US
rdedinsk@asu.edu

Mohammad Khayatian
Arizona State University,
660 S Mill Ave, Tempe, AZ, US

Mohammadreza Mehrabian
Arizona State University,
660 S Mill Ave, Tempe, AZ, US

Aviral Shrivastava
Arizona State University,
660 S Mill Ave, Tempe, AZ, US

Abstract

Traffic intersections will become automated in the near future with the advent of Connected Autonomous Vehicles (CAVs). Researchers have proposed intersection management approaches that use the position and velocity that are reported by vehicles to compute a schedule for vehicles to safely and efficiently traverse the intersection. However, a vehicle may fail to follow intersection manager (IM) scheduling commands due to erroneous sensor readings or unexpected incidents like engine failure, which can cause an accident if the failure happens inside the intersection. Additionally, rogue vehicles can take the advantage of the IM by providing false position and velocity data and cause traffic congestion. In this paper, we present a new technique and infrastructure to detect anomalies and inform the IM. We propose a vision system that can monitor the position of incoming vehicles and provide real-time data for the IM. The IM can use this data to verify the trajectories of CAVs and broadcast a warning when a vehicle fails to follow commands, making the IM more resilient against attacks and false data. We implemented our method by building infrastructure for an intersection with 1/10 scale model CAVs. Results show our method, when combined with an IM dataflows, is more dependable in the event of a failure compared to an IM without it.

2012 ACM Subject Classification Computer systems organization → Embedded and cyber-physical systems

Keywords and phrases Connected Autonomous Vehicles, Intersection Management, Dependable Systems

Digital Object Identifier 10.4230/OASIcs.ASD.2019.7

Category Interactive Presentation

1 Introduction

CAVs are expected to shape the future of the automobile industry because they have the potential to minimize traffic, increase user satisfaction by enabling user autonomy, and, most importantly, increase safety for all interaction models. In order to truly be accepted by the community, it is important to have reliable infrastructure to support these vehicles.

A point of complexity in this fully autonomous system is managing traffic at an intersection. To manage traffic best, infrastructure is implemented to support intersection management.

Workshop on Autonomous Systems Design (ASD 2019).
Editors: Selma Saidi, Rolf Ernst, and Dirk Ziegenbein; Article No. 7; pp. 7:1–7:13
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Automated Intersection Management or Intersection Managers (IM(s)) as they are referred to in this paper interact with vehicles as they approach an intersection. IMs have the potential to make intersections safer and more efficient compared to the involvement of human drivers in interactions, as these IMs instruct traffic as an unbiased external system specifically designed for both safety and efficiency. Humans are biased and prone to making a greedy decision in safety-critical situations which can decrease throughput and lead to accidents. Thus, many researches regarding intersection optimization have involved the use of IMs.

Some of the most notable existing researches using IMs are Autonomous Intersection Management(AIM)[7], Crossroads[2], and Robust Intersection Management(RIM)[11]. In AIM, vehicles query the IM with a specific velocity of arrival and trajectory of arrival in order to reserve a timeslot through the intersection, which is commonly called a query-based IM(QB-IM). In Crossroads, vehicles provide their current position and velocity to the IM and request a velocity from the IM to drive safely through the intersection; this technique is commonly referred to as a velocity-assignment Intersection Management(VA-IM). VA-IM is improved in RIM; typically a VA-IM assumes constant velocity through the intersection but RIM tracks position trajectory in order to account for external disturbances. From experimentation, we know that even RIM can't account for all external disturbances, because it has no physical monitoring system to ensure that the trajectory of arrival assigned in RIM is actually achieved.

A common technique in these IMs to achieve safe operation despite uncertainty in vehicle's trajectory is to consider a safety buffer around the vehicles to ensure safe scheduling of vehicles through the intersection. Though this achieves some level of reliability by considering worst case position uncertainty of vehicles, this can still be erroneous if too much trust is placed on one system. Consider the case where a vehicle has inaccurate positioning information which is being reported to the IM. If the error of positioning is greater than the threshold safety buffer, an accident is likely to occur in the intersection. For example in AIM to achieve a safe scheduling of vehicles through the intersection, the CAVs send the IM completion times; this way the IM knows which vehicles are about to enter the intersection, are in the intersection, or have completed their trajectories through the intersection. The completion time data which the IM uses to create a schedule are based on the data the car is reporting for its position, and in the case that the car is reporting positioning outside of the threshold safety buffer value the schedule will be inaccurately computed. True safety relies on redundancy of systems to maximize the safety of the IMs.

In addition to safety, these IMs must be more efficient to be accepted by the community. A study done by University of Michigan [5] showed that connected vehicles, while improving quality of life for humans, also opens the door for cyber-attacks. Their study attacked I-SIG sponsored by the United States Department of Transportation and their results showed how drastically throughput can decrease if traffic control is not secured. Those doing research on IMs use encryption to prevent outsiders from attacking the system. Additionally, IMs are external systems and therefore unbiased because they produce the schedule to proceed through the intersection instead of vehicles possibly making greedy decisions. However, if an attacker found a way to spoof into the system, there is no way to detect a sybil attack since there is no data validation.

We propose a Detection System as an additional way to advance safe operation. The Detection System will act as a supervisory system that can verify the behavior of vehicles before and inside the intersection. This external system will supplement IMs positioning data given by the CAVs with real-time, environmental sensing data. Additionally, this system will have the capability to validate if vehicles are communicating with the IM, if a vehicle is actually entering the intersection (eg. sensing for a sybil attack), and ensuring the connected data aligns with the environmental data. In the event of an accident-prone situation, the IM can broadcast a warning to the CAVs interacting with the intersection. This solution can be applied to many intelligent intersection management schemes already implemented or in development, since it acts as an external agent whose primary focus is to identify positioning information(which can be extrapolated to velocity data using a simple distance versus time equation). This enables many IMs to have more fine-tuned positioning data and these management schemes can therefore make more informed decisions for scheduling vehicles through the intersections. The implementation of the Detection System was done by building intersection infrastructure and CAVs that are 1/10 scale.

2 Related Work

Many different intelligent autonomous intersection schemes has been proposed so far [3, 9, 12, 13]. We need to validate that the intersection management schemes are as robust, resilient, reliable, and redundant as possible.

A popular solution to this problem is known as the Autonomous Intersection Management (AIM)[7]. As vehicles approach the intersection at a constant speed, the vehicle sends a speed query to the IM and the IM replies with either a yes or a no to the request based on other requests it is managing for safe operation. If the answer from the IM is yes, the vehicle continues through the intersection at the requested constant speed; however if the answer is no the vehicle slows down and again queries the IM. Once the vehicle is through the intersection it would send a done signal to the intersection manager. This query-based approach worked well on both hybrid (both autonomously-driven and human-driven vehicles operating on the road) and completely autonomous intersections and it did not degrade the position uncertainty due to computational delay on the IM, however it increased network traffic and the amount of computation done on both the vehicle and the IM. In order to have safe operation, AIM relies on the accuracy of the data being communicated between the vehicles and the IM.

Another cutting-edge solution followed it, known as Crossroads [2], whose contribution was to have the IM calculate and assign the velocity of completely autonomous vehicles requesting to cross the intersection. This method when compared to AIM decreased network traffic and computational delay. Vehicles in this research were given a safety buffer which accounted for uncertainty in velocity and position. Network and computational delays are included in finding the round trip delay and this round trip delay is incorporated into the safety buffer. The main issue with this scheme was that it was prone to external disturbances and model mismatches (i.e. a bump in the road or wind). This is because Crossroads tracked a constant velocity through the intersection without accounting for the effect of position uncertainty resulting from these disturbances. The resulting position uncertainty leads to the possibility of inaccurate positioning data being communicated between the vehicles and the IM. This situation could eventually lead to accidents since the IM isn't aware of the vehicles' exact positioning.

The research which improved on Crossroads was Robust Intersection Management (RIM)[11], whose contribution was to track the trajectory of vehicles through the intersection. The premise was to assign a velocity of arrival(VOA) and trajectory of arrival(TOA) to vehicles who were about to enter the intersection. The tracked trajectory accounted for the time it would take the vehicle to speedup or slowdown its velocity to reach its respective VOA. This helped account for any position disturbances and therefore lead to a more accurate model for scheduling vehicles crossing the intersection. Though RIM has higher certainty of vehicle positioning, it still relies entirely on accurate data transmission from the vehicles to the IM which can be error prone as seen in Figure 1.

Since all the researches above have one point of failure for obtaining positioning information, they are all prone to attacks which involve spoofing of location. Further, the system may fail if the communication between the vehicle and the IM fails. Whether a direct attack or lack of validation of connected data, any positioning inaccuracy can be detrimental to safe operation of the IM. To improve all these models, it is extremely beneficial to create redundant systems.

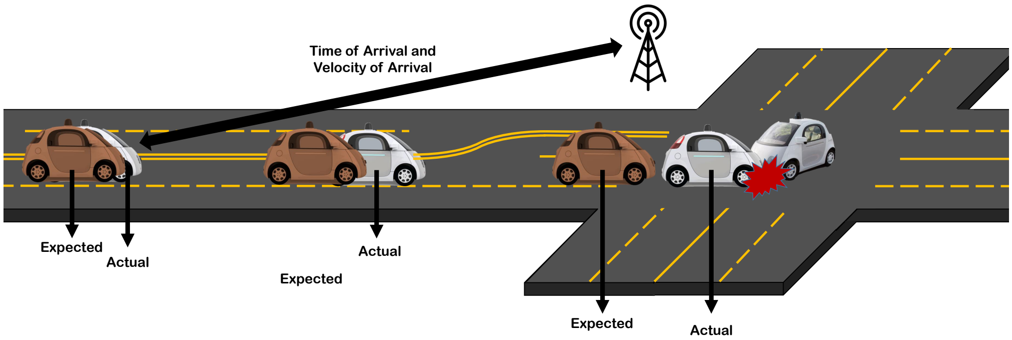

Figure 1 RIM Algorithm: A car will communicate with the IM to get a Time of Arrival (TOA) and Velocity of Arrival (VOA). However, it may not be able to meet the TOA and VOA and can cause an accident.

3 Proposed Method

The Detection System needs to be able to gain a robust understanding of the environment, and the solution implemented to find positioning data was image processing. It is important for this system to act independently from the IM to prevent interaction bias. To act independently, this system must do all processing externally from any of the IM researches, acting only as a reporting mechanism. The interaction model between the IM and Detection System uses I2C, where position determined by the Detection System is sent using I2C to the IM to be used for computation. Note that for security purposes, the Detection System is completely offline, so accessing the system can only be done physically. This adds a layer of data integrity which is dependent on the physical security to the Detection System and the cyber-security available to the IM. Because the Detection System contains four subsystems, there were four I2C communication points between the IM and the Detection System.

As described in Algorithm 1, the Detection System uses live video feed, parses the image using image processing, and alerts the IM using interrupts that a vehicle with a given identification is approaching the intersection. To do the image processing, the camera is calibrated, the image is normalized, and then the front bumper of the car is identified. Once the bumper is located in the image, the distance to the car can be found. This distance is transferred over I2C to the IM.

In Algorithm 2, the IM uses the calculated distance and identification provided by the Detection System to parse for specific packets being sent from the vehicles. Then the IM will process the environmental data and compare it to the connected vehicle data. During this comparison the IM can perceive four different scenarios: The first is that the image processing data doesn't match the connected data, the second is that a request is received but no vehicle is detected, the third is the Detection System recognizes a vehicle is approaching even though

Algorithm 1: Pseudocode for Camera platform.

```
for every Δt seconds do
    Read Image;
    Process Image ;
    Send vehicles' data to IM;
end
```

the vehicle hasn't communicated with the IM, and the last is that the connected data is the same as the environmental sensing data. These scenarios are discussed in Section 4.

Algorithm 2: Pseudocode for IM.

```
Timer_ISR (every t seconds){
    Get data from cameras;
}
if (A packet is received) then
    Process the data;
    if Scenario 1: Camera's data doesn't match the expected then
        Send back a packet to the vehicle
    end
    if Scenario 2: A request is received but no vehicle is detected then
        Change Policy e.g. MAC blocking
    end
    if Scenario 3: vehicle didn't communicate (A rouge driver) then
        Broadcast an alert to other automated vehicles
    end
end
```

4 Empirical Evaluation

The Detection System is beneficial to preserve high precision of position certainty for the IM. We implemented and tested with the four previously discussed scenarios in order to show the advantages of using the Detection System. Our testing has shown that in all four scenarios, the Detection System was able to observe and report the vehicle's location within 2 cm accuracy.

4.1 Testbed

4.1.1 The components

The Detection System created in this research involved four separate, equivalent subsystems to work together. Each subsystem's purpose was to monitor a given lane of traffic. Given a typical intersection, there are four lanes of traffic therefore four subsystems were needed for testing of the Detection System. Each subsystem was built using a Raspberry Pi 3 Model B as the processing agent, running operating system Raspbian Sketch 4.14. A Raspberry Pi V2 Camera was used for capturing the visual input to the subsystem. OpenCV 2.4.9.1 in

Python 3.5 was used for image detection. Note the interaction of all these components in Figure 2.

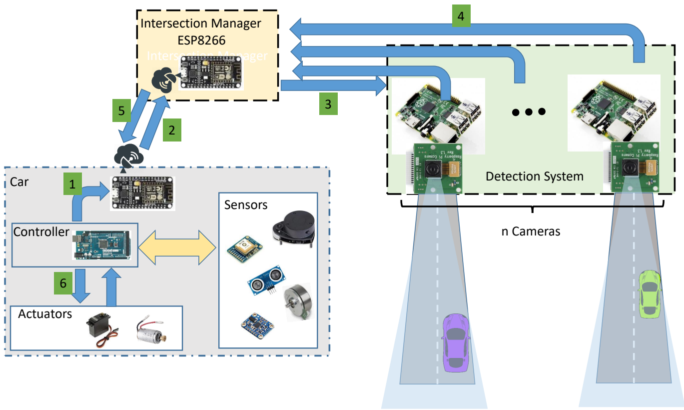

Figure 2 The IM, Detection System and vehicle interact with each other in five steps to verify the car data using the environmental sensing data.

4.1.2 Chosen IM

In order to test the Detection System, a version of RIM was used for the IM. RIM iterated on both AIM and Crossroads, and has the most accurate model for positioning data. By using the model with the most accurate positioning, the potential of the Detection System can be seen more clearly.

Briefly, RIM divides the action of a vehicle approaching the intersection into four phases. Initially, the vehicles synchronize their clocks with the IM and pass a physical synchronization line. If the synchronization is not successful the vehicle tries to synchronize again. If the synchronization is successful, then when the vehicles pass the physical transmit line they send to the IM a packet containing position, velocity, acceleration, timestamp, outgoing lane, minimum and maximum acceleration, and identification. If this transmit is unsuccessful, the vehicle slows down and tries again. If this transmit is successful, the IM calculates a trajectory of arrival and velocity of arrival for the vehicle based on the packet sent by the vehicle and the respective scheduling algorithm being used to control the vehicles which are currently interacting with the intersection. Then the IM sends this packet back to the requesting vehicle, meanwhile the vehicle is keeping a constant velocity headed toward the intersection until it receives further instruction. Finally, the vehicle creates a reference trajectory from the received packet from the IM and follows it through the intersection[11].

Using RIM described above as the basis for the IM, the Detection System was incorporated into a variant algorithm for RIM which for our purposes will be called RIM Robust. RIM Robust varies from RIM after phase two finishes. In phase three after the vehicle transmits its data to the IM, the IM verifies that the transmitted data is the same as the environmental sensing data from the Detection System.

To implement RIM Robust as seen in Figure 3, the first step was to build a model intersection with model CAVs. The vehicles were built using TRAXXAS RC Car with onboard ESP8266 boards for communication with the IM, which also used an ESP8266 board. All boards were designed using the Arduino software suite. A wooden arch was crafted for the infrastructure to support the Raspberry Pi and Camera above the 1/10 scale model of

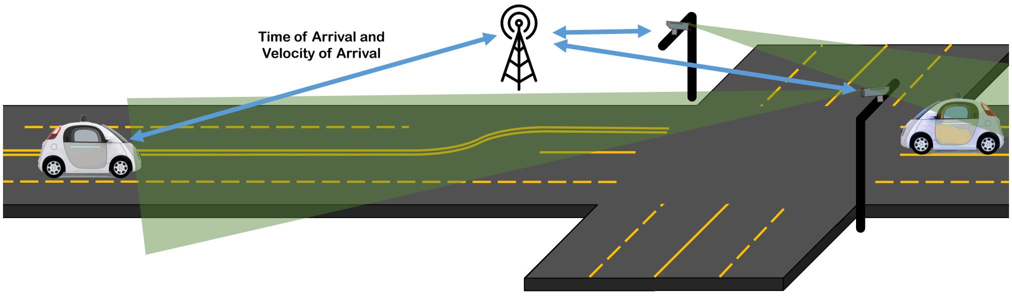

Figure 3 RIM Robust has a surveillance Detection system that monitors the position of vehicles and reports it to the IM.

intersection using TRAXXAS RC car. Given that the average traffic light is approximately 18 ft. to 22 ft. in height, our scaled, wooden infrastructure's height was 26 inches tall. The infrastructure of the model intersection can be seen in Figure 4.

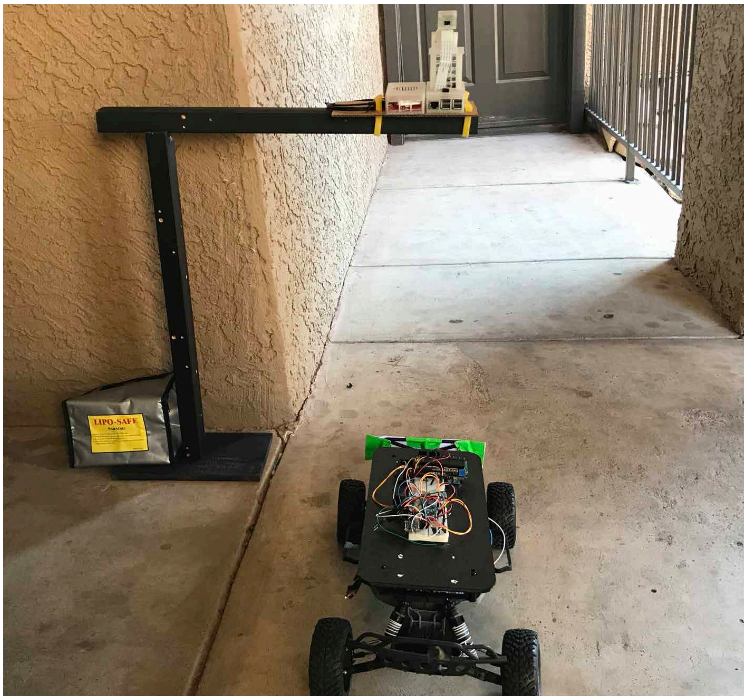

Figure 4 At one corner of the intersection, you can see one of the four subsystems of the Detection System mounted on top of the wooden infrastructure with a TRAXXAS RC Car about to drive underneath.

4.1.3 Image Processing

For image processing, the camera first had to be calibrated. It is important to note the field of view(a restriction to what is visible in the frame[1]) because it is directly related to the focal length(the distance from the focal point to the vertex of the first optical surface[10]) which is needed to understand distances in a picture. Note that depending on the angle at which the camera is pointed at the intersection, different focal lengths will result. Once the camera is calibrated and the focal length is found, the images read in could be normalized for image processing and the front bumper of the RC cars can be more easily detected. Once the bumper was detected, trackers were created for each RC car[8] which could be used to find distance by comparing with a base set of images. This base set of images defined the ratio of pixels to distance, so by comparing an identified pixel to the base set you can estimate the distance as shown in Figure 5.

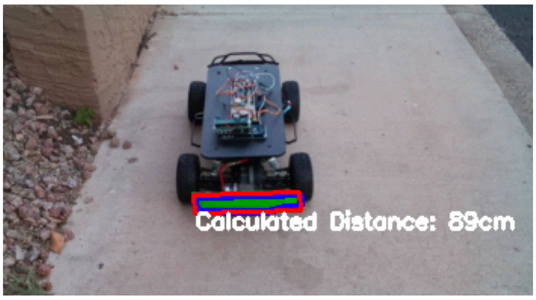

Figure 5 Detect the front bumper use the pixels to approximate actual distance from the camera to the vehicle.

4.1.4 Preliminary results

In Figure 6, the chart shows a comparison between the vehicle self-reporting error and the Detection System error. The Detection System has an error of approximately 2cm on average for true distance compared to actual distance. This uncertainty could be introduced by error in the perception algorithm itself. This error is small, though in a real-world scenario noisy perception data is likely and thus this methodology to obtain data will need to be tuned to any given environment. The results from testing show that safety of the intersection is increased because the likelihood of an accident is minimized, and the total throughput is increased because there are no accidents that result (where accidents cease throughput of cars through the intersection). Throughput here is a measure of managed vehicles divided by total wait time.

4.2 The Benefits of the Detection System

Analyzing RIM Robust, four scenarios may occur shown briefly in Algorithm 2. Note that the positioning data reported by the CAVs is prone to higher amounts of error compared to the Detection System, which has positioning data that is not prone to external disturbance/misreporting and therefore the Detection System's data overwrites the CAVs' data. In the first scenario, the environmental data is different from the reported data from the vehicle. This scenario requires the IM to go into recovery mode and update the vehicle's information packet to what is observed from the environmental data. The second scenario is that a request is received by the IM from a vehicle but the requesting vehicle is not detected by the Detection System. This implies an invalid request is coming from an attacker or an error occurred, therefore mark this user as an invalid user by implementing methods such as MAC blocking. This scenario requires that RIM Robust's policy will be updated so as to handle attackers who are trying to slow throughput or cause harm. The third scenario is that the Detection System finds a vehicle entering the intersection which didn't communicate.

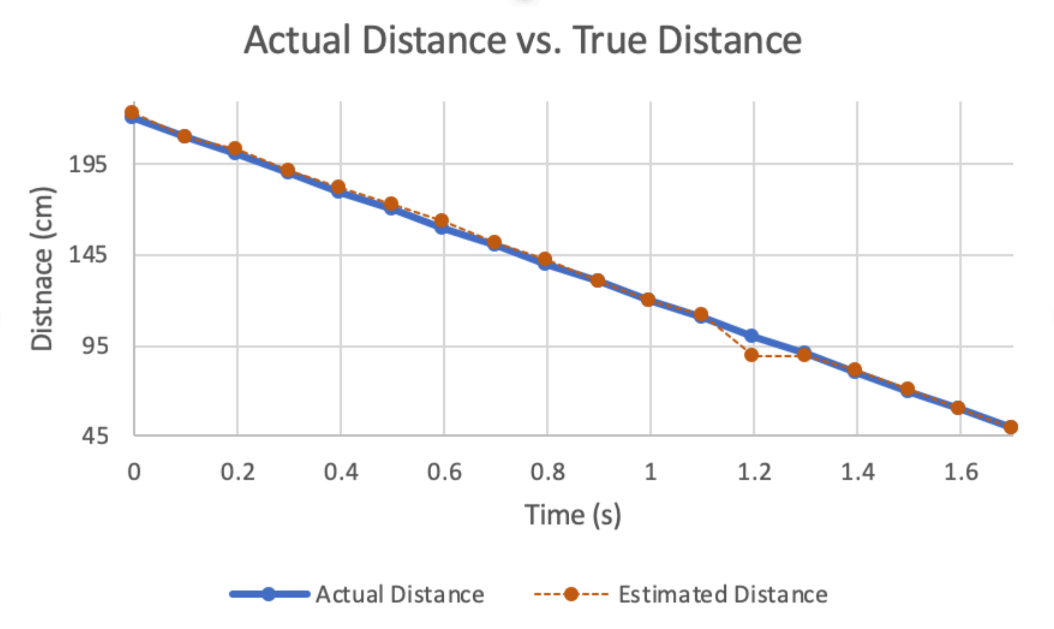

Figure 6 Observe the difference in position from the known distance to the calculated distance by the Detection System during one run of the program.

This scenario again requires RIM Robust's policy will be updated to handle a rouge driver attack, where a vehicle outside the system maliciously goes rouge. Note the interactions for the three error scenarios described here are depicted in Figure 7. Finally in the last scenario, the vehicle's data matches the Detection System's data. This requires no change to the operation of RIM and so the algorithm continues as described in RIM.

In order to test this system, RIM Robust was implemented using recovery mechanisms such as alerting CAVs of potential attacks, MAC blocking, and updating vehicle's information.

4.3 Sybil Attack case study

4.3.1 Theory

The Sybil attacker uses false data(eg. ID) to authenticate itself and sends a request to the IM for passage through the intersection[4]. The attack would cause the IM to schedule a vehicle which isn't actually present in the intersection with the intention of decreasing throughput through the intersection[6]. This leaves the intersection vulnerable in high traffic situations. To control such situations, the Detection System can monitor the traffic and determine whether or not the vehicle exists or not. If there is a vehicle detected using the environmental sensing data, the Detection System goes a step further and identifies the vehicle using the identification characteristics such as license plate. If the identification packet sent by the vehicle contains a matching license plate, the vehicle can continue communication with the intersection. If the vehicle does not match any existing, approaching vehicles in the intersection the vehicle is label as a Sybil attacker and is marked as an invalid user in the intersection using MAC blocking.

A large scale sybil attack could result in denial of service to valid vehicles entering the intersection because the sybil attacker is jamming the IM network with requests. To explain this in more detail, in a typical traffic situation there is a worst case time it takes a vehicle to go through an intersection; this would be in fast, high traffic situations. By simulating this traffic situation, the worst case downtime between requests can be approximated. The

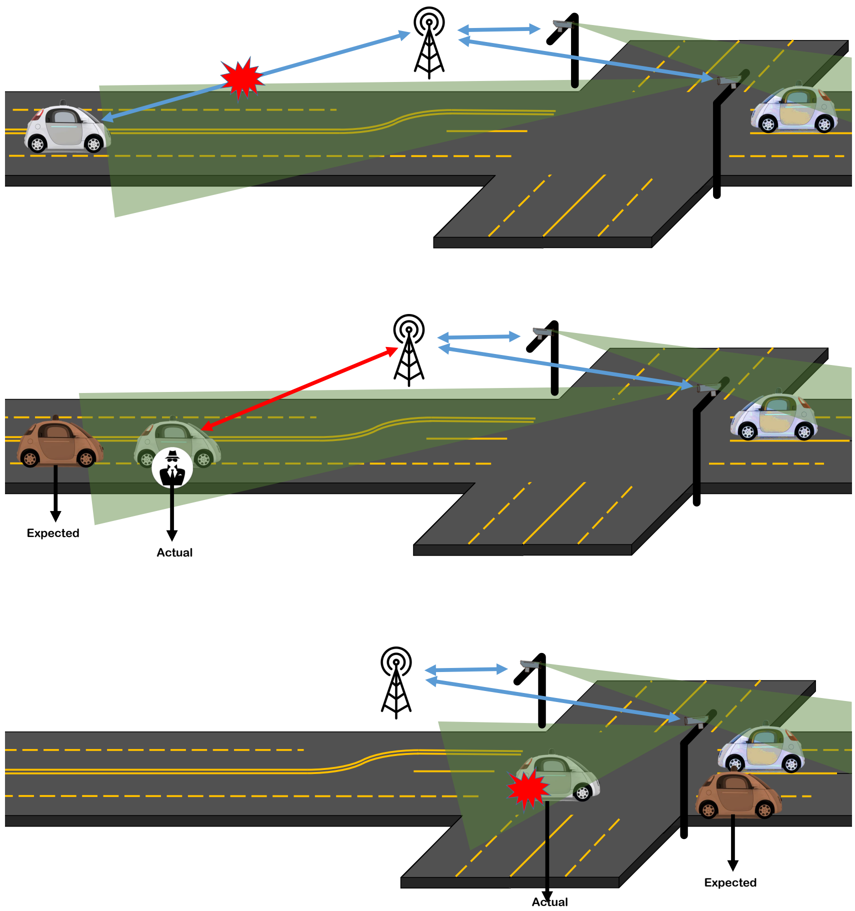

Figure 7 Three beneficial scenarios of the Detection System (from top to bottom): when the vehicle's communication systems are malfunctioning, when an attacker claims there is a vehicle nearing the intersection when there is not, and when there is a discrepancy between the connected data and environmental detection data.

worst case downtime is representative of the time interval between requests in the case where the maximum number of requests are being made in a short period of time, and therefore the worst case downtime is the smallest time interval that can occur between requests in high traffic situations. If the interval were any smaller than the determined worst case downtime it would mean that vehicles were physically overlapping which is impossible.

4.3.2 Implementation

In this experiment we used one ESP8266 microcontroller board for simulating the attacker. In the event that a sybil attack is sensed, based on the identification and information sent in the packet from the vehicle, the Detection System verifies that in the respective outgoing lane there is a vehicle that matches the identification information. If such a vehicle doesn't exist, the first step is to alert CAVs in the intersection to a potential threat and to proceed with caution. The next step is to get the MAC of the malicious requesting vehicle and use the MAC address to block it from interacting with the IM and other CAVs on the network.

4.3.3 Results

It was observed that as the number of these sybil attacks increases, the throughput of the intersection decreases drastically in RIM because RIM schedules all vehicles that make a request. However using RIM Robust, the vehicles are tracked and if a vehicle that is communicating with the IM isn't detected, it is blocked from the intersection; therefore, the only vehicles scheduled are those physically present at the intersection and there is no throughput degradation.

4.4 Vehicle failure

4.4.1 Theory

If vehicle fails to communicate with the IM, whether it be that the communication systems aren't functional or possibly the vehicle doesn't have enough power for communication, the intersection should be able to manage traffic. A distinction should be made here: a vehicle failure is not a latency issue in communication between IM and CAV. In the scenario of latency in communication and the vehicle is too close to the intersection, the vehicle would slow to a stop if necessary to initiate communication. However, in the situation where vehicles don't have the ability to communicate at all, the Detection System should be able to identify them once they are within bounds of entering the intersection.

4.4.2 Implementation

When the Detection System identifies a vehicle which has crossed the synchronization line and hasn't communicated with the IM yet, RIM Robust can plan its scheduling accordingly. Two cases may occur if the vehicle doesn't communicate with the intersection. The first is if the vehicle comes to a stop outside of the intersection. In this case vehicles already scheduled proceed normally and vehicles which are communicating to receive a schedule slow down. These vehicles wait for a notice from the IM, whose job is to observe the location of the vehicle. The second case is if the vehicle doesn't stop before reaching the intersection. In this case the car continues through the intersection without communicating to the Detection System. For this situation, the vehicles already scheduled proceed with caution and vehicles which are communicating to receive a schedule slow down until the car has passed through the intersection. Once the vehicle which is not communicating exits the intersection, normal operation proceeds. For this case study, TRAXXAS RC cars went through the intersection without communicating with the IM. This was done to observe if the Detection System recognized that cars were present in the intersection and prevent a potential accident.

4.4.3 Results

This is a necessary precaution for managing vehicles who cannot be scheduled. The broken-down vehicle situation eliminates a lane of traffic, which is extremely harmful to managing traffic properly. This case study didn't implement managing this situation beyond alerting CAVs to the situation. In the other scenario of this case study where a vehicle proceeds through the intersection without communication, it is assumed the vehicle which cannot communicate with the IM is either a potentially malicious vehicle or it is a vehicle whose communication is down. Both are a threat to safe operation, and due to the unpredictable nature of this vehicle it is safest to clear the intersection of all vehicles and let this vehicle make its decision (pass through the intersection) before normal operation begins. Comparing throughput in this scenario, RIM may cease safe operation if a rogue car causes an accident but RIM Robust ensures no accident will occur so traffic will continue normally after the rogue car is handled.

4.5 Reporting error

4.5.1 Theory

The vehicle may fail to maintain the expected trajectory of arrival or velocity of arrival, and this failure can cause a position error greater than the threshold set for an acceptable safety buffer. This leaves the intersection vulnerable to collision, so to validate this situation the Detection System monitors the vehicles' positioning data and ensures scheduling doesn't violates any safety buffer boundaries.

4.5.2 Implementation

RIM Robust was tested for this scenario by hard-coding an error in the reporting mechanism of the TRAXXAS RC Car to be plus or minus one meter from a given position. The role of the Detection System is to monitor the actual position of the vehicle and report this to the IM. If the IM recognizes a discrepancy between the reported value from the vehicle and the observed value from the environmental sensing data the IM defaults to using the data provided by the Detection System and updates both the vehicle's information and the scheduling algorithm inputs.

4.5.3 Results

No safety buffers were violated in the RIM Robust. Violation of safety buffers may lead to accidents in RIM and halt operation of the intersection entirely, but this scenario won't occur using RIM Robust.

5 Conclusion

As processes for driving continue to become more automated, it becomes pertinent to ensure safe passage during traffic. Intersections become an area of concern for safe passage, as this is the most complex area of traffic to navigate given the infinite scenarios that can occur. IMs are typically a great way to increase efficiency of traffic, but ultimately IMs are unsuccessful if they cannot direct vehicles safely in all scenarios. Many researchers using IMs to conduct traffic have one source for position reporting and therefore lack redundancy. The advantage of using the Detection System is the high level of position certainty that can be obtained. Adding the Detection System to an IM makes the intersection more resilient to attack or inaccurate reporting because it increases the reliability of the system by duplicating positioning data sources. By relaying the information obtained by the Detection System back to the IM, and combining it with the connected data between vehicles and the IM, a more accurate portrayal of the intersection is obtained and vehicles can react safely to even more scenarios than was possible before.

References

1 Patricia L Alfano and George F Michel. Restricting the field of view: Perceptual and performance effects. *Perceptual and motor skills*, 70(1):35–45, 1990.

2 Edward Andert, Mohammad Khayatian, and Aviral Shrivastava. Crossroads - A Time-Sensitive Autonomous Intersection Management Technique. In *Proceedings of The 54th Annual Design Automation Conference (DAC)*, 2017.

3 Tsz-Chiu Au and Peter Stone. Motion Planning Algorithms for Autonomous Intersection Management. In *Bridging the Gap Between Task and Motion Planning*, 2010.

4 Karl C Bentjen. Mitigating the Effects of Cyber Attacks and Human Control in an Autonomous Intersection. Technical report, AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON AFB United States, 2018.

5 Qi Alfred Chen, Yucheng Yin, Yiheng Feng, Zhuoqing Morley Mao, and Henry Xianghong Liu. Exposing Congestion Attack on Emerging Connected Vehicle based Traffic Signal Control. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS'18)*, San Diego, CA, February 2018.

6 John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

7 Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

8 Mohsen Kheirandish Fard, Mehran Yazdi, and Mohammadali MasnadiShirazi. A block matching based method for moving object detection in active camera. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, pages 443–446. IEEE, 2013.

9 Chien-Liang Fok, Maykel Hanna, Seth Gee, Tsz-Chiu Au, Peter Stone, Christine Julien, and Sriram Vishwanath. A Platform for Evaluating Autonomous Intersection Management Policies. In *Proceedings of IEEE/ACM 3rd International Conference on Cyber-Physical Systems*. IEEE Computer Society, 2012.

10 John E Greivenkamp. *Field guide to geometrical optics*, volume 1. SPIE Press Bellingham, WA, 2004.

11 Mohammad Khayatian, *et. al.* RIM: Robust Intersection for Connected Autonomous Vehicles. In *RTSS*. IEEE, 2018.

12 Joyoung Lee and Byungkyu Park. Development and Evaluation of A Cooperative Vehicle Intersection Control Algorithm under The Connected Vehicles Environment. *IEEE ITS Transactions*, 2012.

13 Florent Perronnet, Abdeljalil Abbas-Turki, and Abdellah El-Moudni. Cooperative Intersection Management: Using Mini-Robots to Compare Sequenced-based Protocols. *Les Journées Nationales des Communications dans les Transports (JNCT)*, 2013.