2020

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

EVALUATION OF V2V AND V2I IN HIGH IMPACT LOW FREQUENCY EVENTS

Rickie Cashwell

A Research Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
and
Congdon School of Supply Chain, Business Analytics, and Information Systems

University of North Carolina Wilmington

2020

Approved by

Advisory Committee

Gulustan Dogan _____

Ulku Yaylacicegi Clark _____

Sudip Mittal
_____
Chair

Accepted By

_____
Dean, Graduate School

TABLE OF

RESEARCH GOALS

The primary research goal of this project is to extend Gupta and Sandhu's work and explore a simulated environment that follows the safety aspect of the ITS vision. The UNCW school shooting scenario signifies this vision. Because safety is not as widely studied as say mobility (ex. Traffic alerts) or environmental (vehicle emissions) factors, the goal here is to explore this area by analyzing a ITS response in a school shooting scenario in a real-life environment. Exploring this aspect of ITS involves a qualitative analysis of the ITS impact in a University setting. User input from authorities can help provide a better understanding of the step by step process of a school shooting event. Due to the high level of impact of such an event, a quick response time of the system is required.

OUTLINE OF COMPLETED THESIS

Introduction discusses intelligent transportation system's (ITS) vision and mission objective. V2V (vehicle to vehicle) and V2I (vehicle to infrastructure) is used as the primary communication model between objects. These models can provide benefits for users in a high impact low frequency event (HILF). Current Market trends by region is also displayed here. This relevant Background work relating to cloud computing and architecture in V2V andV21 communication is discussed in Review of Literature. Due to security issues, the use of a public key infrastructure is needed here. Methodology goes into more detail on architecture layers and their function. An additional use case in a HILF event is shown here. A proof of concept is realized through an evaluation of police responses and the inner workings of autonomous vehicles.

INTRODUCTION

As the rise of IOT devices increase, the dilemma of quick secure connections between such devices becomes more apparent. Connections between IOT devices can provide several benefits. This is especially true for vehicles. Vehicles communicating with each other through IOT devices has become so prevalent as a concept that it is dubbed as the internet of vehicles (IOV). Our goal is to provide an intelligent system that provides safety, security, and a smart traveling experience for users operating a transportation device. This goal aligns with vision of the intelligent transportation system (ITS) association. One such use case of an ITS may involve a user receiving local crime activity alerts or sending requests for a mechanic when they have a busted tire. These ideas seem very beneficial but how feasible are they in practice? With fast moving objects on the road, how are we able to effectively enable accurate, low latency connections while deterring unauthorized attackers on the road? Gutpa and Sandhu proposed a dual communication method of vehicle to infrastructure (V2I) and vehicle to vehicle (V2V) approach instead of just straight peer-to-peer connection.

These two proposed innovations have the potential to change of what we know as vehicle communication. Vehicles will be able to exchange useful parameters and aggregate it to drive intelligent decisions. These decisions will be transferred to users via application interface. Some useful parameters include location, speed, and direction. V2I can provide different use cases by

connecting to roadside infrastructure (RSI). These infrastructures can work with vehicles in providing other useful information locally quickly. One example is a work zone ahead alert. These devices will communicate using dedicated short-range communication to quickly interreact with such devices. Data transmission between objects using DSRC exchange data through packets called basic safety messages (BSM). Using DSRC, a 360-degree proximity BIS exchange is possible. Also, because of the speed of DSRC, messages will be sent up to 10 exchanges per seconds.

The issue of security is quite clear in the aspect of V2I and V2V. A multitude of entities can become an infrastructure. This can be anywhere from traffic light to a parking lot. This raises obvious security concerns. Also, a vehicle has potentially hundreds of separate electrical components (ECU) with many lines of code. This leaves a very broad attack space for a malicious user to take advantage of. Some cyber-attacks include spoofing location, stealing user private data, and tampering with an object's firmware.

To address security issues, a system that will build a degree of trust between entities will have to implemented. approach is the use of access-based controls (ABAC) to authorize users. Using an authorization mechanism like ABAC prevents malicious behaviors like fake messages and keep track of rogue vehicles. An attribute-based access control system is an effective method for authorizing users for BSM exchange. This system provides a solution for vehicles to safely connect to anonymous devices. Attributes can authorize a vehicle such as location, speed, type of object.

To enforce ABAC authorization, Gupta and Sandhu takes advantage of cloud and cloudlet computing to get the communication efficiency objects need [3]. Cloudlets serve as a mini cloud datacenter with a defined geographic range. Motionless objects like infrastructures can serve as the edge of a cloudlet. Using this implementation, communication can exist through

certain layers and processing in others. V2V communication can pass through a middleware object where message filtering take place and that is where the edge comes in. For a secure connection to take place, security policies can exist as a gateway from the edge to the cloud. Attribute based access controls are supplemented to a policy model to filter out unauthorized users. Since there is such a wide attack space this method seems most feasible.

The advantage of using a fog or cloudlet infrastructure over just a central cloud is for two reasons: reduced latency, and more secure messages between objects. Messages are more secure because direct peer to peer connections are not being made. An attacker would also have a much more difficult time finding the recipient of the sender. This security feature would also promote user privacy and anonymity. Also, because certificates are being validated at the infrastructure level, certificates can be revoked, blocking access to that edge infrastructure. Because the geographic location of cars changes constantly, latency can become a huge issue for a model built on one central cloud. Certain messages could be sent and received much sooner or later depending on how close they are to the cloud. Having a more decentralized approach helps spread out the updates more quickly and processing can be carried out more efficiently at the edges. Having too many processes and functions at one point can lead to central point of failure. This combined with the potential loss of internet connectivity can ruin overall performance.

When building these models for V2V and V2I, we must extend our use cases from just traffic congestion monitoring, assisted driving, or traffic alerts. ITS is a vision that offers safety and security to drivers, not just convenience. We must look into the benefits of monitoring and alerting high impact low frequency events (HILF). HIFL include events such as: crime shootings, car crashes, or natural disasters. One scenario may involve an escaped convict incident at a university. In this event, the local police would alert all Connected and Autonomous Vehicles (CAV) in the associated cloudlet through Infrastructure to vehicle (I2V)

communication. The alert would discourage CAV to enter the area, and a forced parking event may occur.

To create a robust ITS for HILF, the system must respond quickly with events and strict policies must be put into place for forced behaviors. This is especially true when enforcing forced behavior like forcing a vehicle to shut down or park. There must be fail-safe mechanism in place to prevent potential mistakes in the system. For example, what would happen if a mistake would occur, like forcing a vehicle out of range of a HILF event to shut down. Also, in such a high impact situation, the requests received and sent must be quick. Latency concerns can hinder a necessary immediate response.

ABAC will help determine which policies to enforce. In the above example, A HILF event is triggered and attributes of local CAVs are relayed to the cloud for processing. Based on the attributes of the CAVs like distance and direction, policy processing takes place and the local police will receive instructions. Depending on traffic, actions will be taken place to navigate CAVs in the impacted area to a safer one. This navigation scheme helps move CAVs out of an impacted area and at the same time allow the police easier access in.

ITS MARKET TRENDS AND GROWTH

Nowadays, the amount of smart autonomous vehicles on the road is increasing across the world. As this number grows, it is becoming increasingly difficult to implement a worldwide ITS infrastructure. Despite this, transportation agencies are preparing for the future. The lack of an infrastructure leads to a rise in traffic congestion which in turn leads to a negative impact to the environment due to vehicle emissions. The implementation of an ITS infrastructure can also provide benefits in safety. Around 36,600 deaths were reported as traffic fatalities on U.S. roads in 2018. Due to these issues, the market forecast of implementing an ITS infrastructure is likely to surge [6]. The global transportation market is valued around USD 1643.8 million in 2018. By 2026, the projected value of the market is around USD 8474.2 million.

Government funding is necessary for meeting the demands of this highly expensive project. By region, the Asia pacific is leading the ITS market and is expecting to dominate over the forecast period (2019-2026). China is leading the market in this region due to huge government funding. South Korea is the second largest country to fund the deployment of an ITS in this region. The second largest market of ITS is in Europe and this region is forecast to grow

significantly during the 7-year period. The U.K. is the current leader in funding the transportation segment of ITS. Native America is projected to have a steady growth due to Canada's implementation of ITS. Latin America is also projected to have a steady growth thanks to Mexico and Brazil's support for an ITS implementation. Industry leaders are also in the mix by providing hassle free ITS services. Some industry key players are: Denso Corporation (Aichi, Japan), Hitachi (Tokyo, Japan) and Siemens ag (Munich, Germany) [6].

REVIEW OF LITERATURE REVIEW AND ANALYSIS

According the U.S. department of Transportation, the recommended infrastructure needed for authorizing and authenticating BSM messages while keeping the data confidential is the PKI. The root certificate authority, called the Security Credentials Management System or SCMS, will maintain a circle of trust between messages and verify authorized messages are received by their intended recipients. This is done through the use of issuing digital certificates. BSM that have a signed certificate by the root or intermediary certificate authority (CA) are considered trusted. This is to help prevent malicious hacks such as a man in the middle attack, because most likely they will not have a signed certificate. Some more examples of a cyber-attack to consider include: object impersonation to send fake BSM, driver spoofing, stealing personal information, and sending fake crash alerts. In an attempt to keep user data private from attackers, personal information is kept only as needed then discarded and the information is only kept in short geographic location. Asymmetric key exchange combined with digital certificates allows for authentication, authorization, confidentiality, and integrity of BSMs.

To ensure this system works correctly, each BSM will have a signed certificate attached, which is signed using the private key of the CA. Before V2I and V2V communication occur, a vehicle must be registered into the SCMS to gain access to certificates; this occurs as soon a vehicle enters a cloudlet. The root issues a certificate giving access of the vehicle to certain cloudlets. New certificates will automatically be issued when the moving vehicles moves into range of an unknown cloudlet. There are different types of certification such as identification and enrollment. Compromised or rogue vehicles can be revoked of their certificate and remembered in a certificate revoked list (CRL). Each vehicle will be issued 20 certificates to sign every week

and BSMs will be rotated every 5 minutes. Each vehicle will receive a new set of 20 certificates every 100 minutes.

While using the PKI provides some advantages, like strengthening security aspects like confidentially and authentication, some additional work on potential pitfalls are needed. Latency and bandwidth issues can emerge if there are too many untrusted vehicles in a CRL. This is because every time a vehicles certificate is reevoked, all vehicles has to update their CRL. With 256B to 800B certificates being distributed per year and 17 million vehicles in circulation, these issues are concerning. Also, while the sender is verified, false BSMs can still be exchanged by a malfunctioning vehicle.

Software architectures in the past have been proposed to identify objects and its connection to other objects or the cloud. Gupta and Sandhu proposed a four-layer architecture to help visualize the flow of connections between entities [2]. This architecture introduces clustered objects at bottom layer. This layer contains objects and objects of objects called clustered objects. A virtual entity is introduced to keep the state of an object or clustered object, which resides one layer above the object layer. Using amazon web service shadow, a constant state is kept, allowing for other layers to continuously have something to communicate with, even without internet connection. The cloud/application layer controls the processing needed in communication. Virtual objects can communicate to the cloud through an edge infrastructure which is the edge of a cloudlet. The edge infrastructure act as intermediary between objects to help filter out unwanted messages. The communication runs from the bottom starting at the object layer, then the state is changed in the virtual layer, then the virtual layer communicates with the cloud for processing. Then the processing result funnels down back to the object layer where the user is notified through an application.

Attributes are preferred as the primary method to facilitate access controls. ABAC provides fine grained authorization capabilities in a dynamic and distributed setting. Role based access controls are more suitable in an enterprise setting where there aren't many potential unknown entities. Administrators are able to easily assign a limited set of roles to users in a set location. Thus, roles are not as suitable for a system involving many geographical domains unlike attributes.

METHODOLOGY

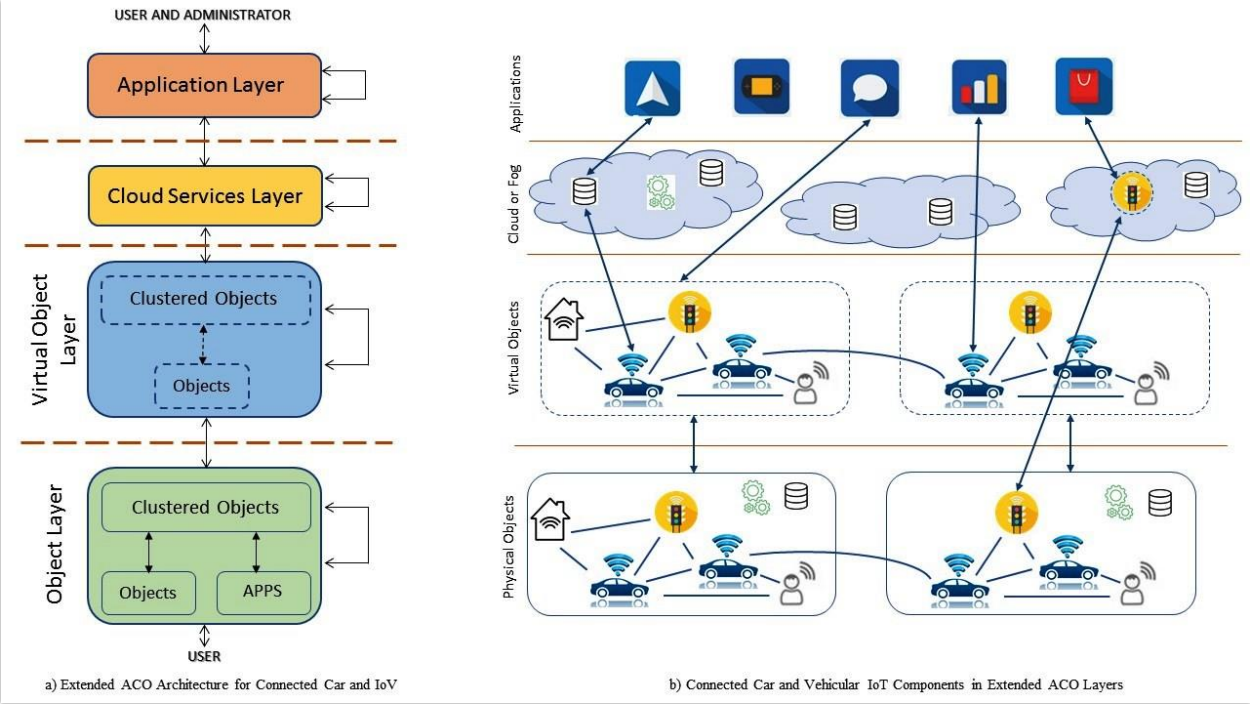

Figure 1: 4 Layer Cloud Architecture

To keep track of a highly sophisticated system of communication between fast moving objects with multiple sensors and actuators, an abstract model is introduced to help break down processes in each moment in time. Handling Different service models provided by the cloud (SaaS, PaaS, and IaaS) and different communication methods (WIFI, Bluetooth, MQTT) can be overwhelming. The importance of breaking down the problem is also due the issue of heterogeneity in between devices. An object in one vehicle may provide completely different functionality and features than another. For example, an infotainment system provides a different purpose and functionality than say a dashboard monitoring system. Also, not just vehicles and the cloud is communicating with each other, but also between the sensors and actuators in a vehicle. There is also the concern of security. With such a broad attack space, it is not practical to allow vehicles to communicate with each other or the cloud directly without any security

measures in place. Due to these concerns, this devised architecture consists of four layers: the object layer, virtual layer, cloud services layer, and the application layer. Figure 1, shows the layers and multiple avenues of communication that take place.

The layer that holds information of the physical objects is the object layer. Access control attributes can accumulate here and be sent to the cloud for processing. Mostly, the attributes gathered here are sent to the edge for a policy checking. This layer also keeps track of objects within the object (clustered objects) such as an engine control system, or a tire monitoring system. Generally, the purpose of this layer is to send accumulated attributes to the upper layers to maintain state and enforce policies. Communication between individual applications in an object or clustered object can be achieved through ethernet or a controlled area network (CAN). Individual sensors (ECU) or objects communicate to each other through wireless local area network standards like WIFI, Bluetooth, or Zigbee. Layer communication occurs through networking protocols such as HTTP or MQTT.

With a vehicle potentially having multiple objects communicating with each other, it is imperative to design a model that helps facilitate communication between devices irrespective of their functionality. There comes the idea of having a virtual state. The AWS shadow cloud service provides the means to capture the state of each object. A hierarchal data file system like JSON can be used record this state. The great thing about this is that the erratic movement of communication between objects doesn't matter. The state is recorded as an ECU and then when communication happens, only the virtual entities are communicating with each other. The JSON will be updated and sent to the virtual layer. For example, sensor 1 will communicate with virtual state 1 then virtual state 1 will connect to virtual state 2 then finally communicate to sensor 2. Another beneficial aspect of maintaining state between objects is keeping connectivity. Although the internet connection may go out, connectivity is always maintained through virtual states.

Meaning, there is always a state available for analysis for the other layers, so a dedicated connection is not required.

The cloud or fog level is where the edge resides and where the AWS/Google cloud services are. The cloud layer is generally responsible for securing connections between clustered objects and housing the Big Data. Policy enforcement is enabled at the cloudlet level with AWS Greengrass technology. Shadowed clustered objects first connect here for this reason. Even individual sensors can interact directly with the fog (and bypassing their state) for processing, however, because of latency, heterogeneity, and connectivity issues discussed before, a virtual layer is still worthwhile to implement to maintain state of these objects.

This layer uses the help of the AWS IOT and the Google cloud IOT core to form a gateway between the lower layer and the cloud one. The cloud layer enforces secure connections using policies created by the administrator layer and even the user. A user can, for example, ask to not to receive certain notifications even though they match the requirement of the related group through their attributes. A public key infrastructure-based certificate is also used to enable encrypted messages to safely reach their destination, which is first issued at this layer (more info on this in the next section). To minimize latency issues certain alerts are exclusive to the central cloud and the fog. To clarify, both the central cloud and the fog can receive updates from the virtual layer. An alert that is located in a certain geographic location may benefit more passing through the fog than through the central cloud if the cloud is not in range for example. The admin layer focuses on interfacing users with applications to generate policies for the edge. Most user applications are supported on the cloud, so the admin layer's functions go hand in hand with the cloud layer.

The advantage of using a fog or cloudlet infrastructure over just a central cloud is for two reasons: reduced latency, and more secure messages between objects. Messages are more secure

because direct peer to peer connections are not being made. An attacker would also have a much more difficult time finding the recipient of the sender. This security feature would also promote user privacy and anonymity. Also, because certificates are being validated in the layer, certificates can be revoked, blocking access to that edge infrastructure. Because the geographic location of cars changes constantly, latency can become a huge issue for a model built on one central cloud. Certain messages could be sent and received much sooner or later depending on how close they are to the cloud. Having a more decentralized approach helps spread out the updates more quickly and processing can be carried out more efficiently at the edges. Having too many processes and functions at one point can lead to central point of failure. This combined with the potential loss of internet connectivity can ruin overall performance.

The impact of just using edge computing has its drawbacks, however. The edge itself does not have the processing power to hold and perform analytics on big data. The central cloud can provide the data analytics needed for large amounts of data. It is estimated that 800 zettabytes of data will be produced by users in 2020 and this estimate may increase as the popularity of IOT devices grow [4]. Also, IOT devices like moving vehicles generate an enormous amount of data. Thus, a hybrid model for cloud computing is needed. The edge can handle preprocessing and relaying of data while the central cloud handles analytics.

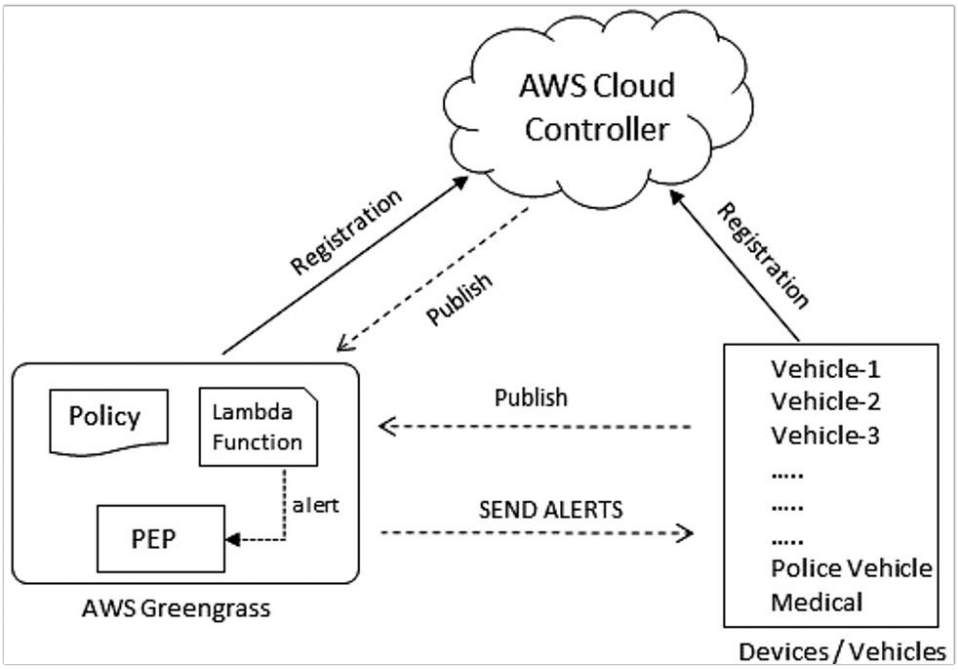

Figure 2: System Architecture

Figure 2 shows the system architecture and the core technologies in play. In this hybrid model, the central cloud functions, such as registration and data analytics, are handled by the AWS Cloud controller technology. AWS green grass provides local cloud services, so a group of cloudlets can be a considered a group of green grasses. Greengrass implements functionality needed for cloudlets such as policy evaluation and enforcement.

HILF ATTRIBUTES AND USE CASES

When thinking about HILF events, there needs to be a quick response and the correct data has to be sent to law enforcement. Authorities should be able to identify specific vehicle attributes such as license plate or vehicle color. For simplicity, attributes can be narrowed down at the edge such as just vehicle types. This can help reduce latency by giving only information needed to authorities. For example, during a HILF even, law enforcement will not need notifications on a nearby car with a tire slip. They just need the vehicle type and what security status is of the vehicle. A vehicle that has been compromised through a cyber-attack may be pick upped by the edge and be labeled as Rogue. Thus, a rogue vehicle should not be able to send alerts, but authorities should receive info on who these rogue vehicles are. This process of identifying vehicle type and security status should be quick and dynamic, but we shouldn't overload authorities with unimportant notifications.

Another use case of HILF involves a shooting at a university. A shooter would first be observed by an object (roadside infrastructure or vehicle) and alerts will be sent to the cloud. Because we have a hybrid architecture of the edge and central cloud, alerts will be received quickly by the edge (if the vehicle is unregistered, alerts will be sent to the central cloud first then, after registration, to the edge) for processing. Attributes are narrowed down and policy will be carried out using AWS Greengrass. AWS Greengrass is a service that uses lambda functions to evaluate policies in the edge. AWS Greengrass is used as the crux for policy evaluation (A group of edge devices can also be considered an AWS Greengrass group). Afterwards, the correct authorities are notified based on geographic area, responsibilities, and security status.

Relevant information is received by the authorities on the object like object type and color. Finally, notifications are relayed to vehicles in the surrounding area and a behavior is enforced such as forced parking or automatic steering of a vehicle.

In a scenario such as a weather disaster, this hybrid cloud system could also prove to be beneficial. For example, the National Weather Service (NWS) sends active alerts that includes the type of weather disaster and the risk level. Other weather observations such as precipitation or temperature are also included. The NWS offer message dissemination through email or text. Using this system, what if HILF weather alerts by the NWS can be sent to the edge or vehicle? The only issue here is how the edge can acquire attributes from the weather alert to function. Luckily NWS uses a data linking format call JSON-LD to publish information to the web. Using a JSON format enables attributes to be sent to the edge. Like the shooting scenario, after acquiring the correct attributes from the HILF weather alert, the edge can alert vehicles in the local area and a forced behavior can occur.

POLICE RESPONSE

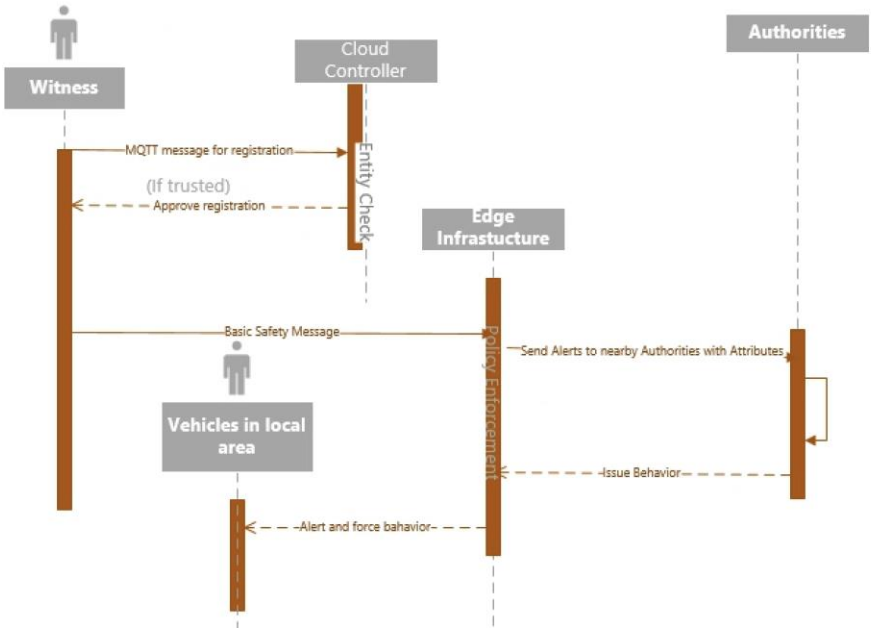

Figure 3: HILF Sequence

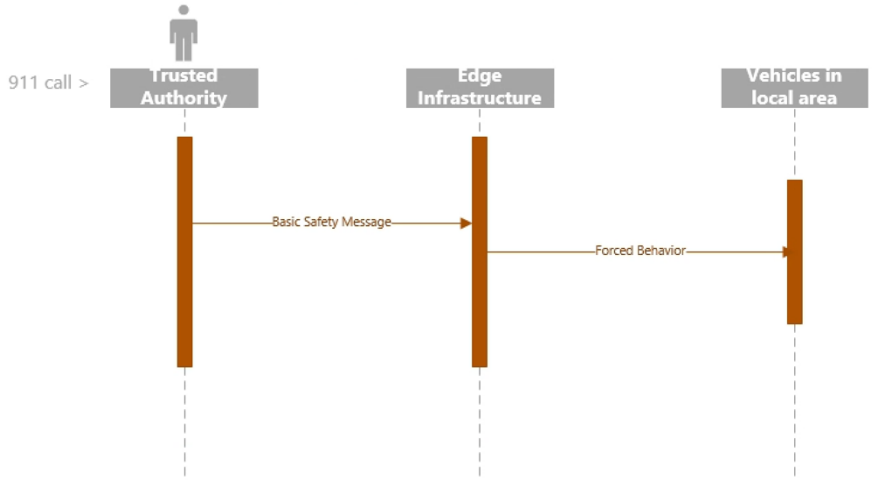

Figure 4: 911 call scenario

To realize the proof of concept in this system under a HILF event, two things must be considered. The first is understanding law enforcement's protocol and how it relates to their

response time. According to UNCW policy for an active shooter event, the University Police is generally responsible for response of incidents, investigation of reports or criminal occurrence, and assistance of other agencies relating to emergencies on UNCW owned or leased properties [7]. For example, as soon as a shooting alerts has been made by a witness, an immediate response will be executed by the University Police. Once an emergency is confirmed to take place, notifications will be distributed across campus unless this hinders apprehension of the shooter or puts others in danger. It is stated that the delay in sending notifications will be minimized as much as possible. Like other universities like Miami and PennState, police officers are trained to respond to shootings as quickly as possible. These Universities follow similar police shooting protocol with Wilmington University in regards to police response timing [9][10]. Notifications will be distributed through email or text messages. Our scenario appears to fit nicely with this typical law enforcement response. The only difference is how messages will be transmitted (notifications to vehicles instead of smart phones or computers).

Since messages are being transmitted directly to local responders, the response time, due to local enforcement protocol, should be quick. Calling 911 instead can lead to a delayed response time due to the time it takes for local dispatch to locate local responders can take several minutes. Each minute can have an impact on the event so time is of the essence. Software solutions like SecureTech Wave allows for immediate responses due to direct contact with local responders by connecting with their primary radio frequency [8]. Local dispatcher does not have to be the middleman for such an event. Thus, having a system that connects with local responders directly is beneficial for this use case.

Figure 3 shows the sequence of events from a witness to the police authorities. Firstly, registration is performed to check if the object is a valid entity (note: Registration is only required for newly detected objects from cloudlets. If the witness operating the object is already

registered, then this step is skipped, and the object/vehicle will interact directly with the cloudlet). MQTT is the chosen networking protocol because it provides a lightweight, high response time, and ease of use for message delivery. The AWS cloud controller will analyze attributes received from the vehicle and verify that the entity is valid for the system. If the entity is trusted, the cloud controller will publish (using MQTT) local cloudlet the vehicle can subscribe to.

Communication between cloudlets and objects is achieved through Dedicated Short Range Communication (DSRC) technology. Data packets exchanged through this technology, are called BSM, which are authenticated and validated through PKI. With a 360-degree proximity and 300-500-meter range, a BSM will be sent out from the witness to the edge where policy enforcement takes place. Alerts are then transmitted to local law enforcement. Certain attributes are also sent out to help the authorities identify the target like vehicle color or direction. In accordance to UNCW policy, authorities will send alerts back to the edge. Once local law enforcement is authorized through edge polices, behaviors are issued out to local vehicles like forced parking or shutdown.

Figure 4 shows an event where a 911 call is made from a witness to a trusted authority. A more typical scenario may involve a witness that is not located in a vehicle like in figure 3. In this sequence of events, registration for entity check is unneeded for the authority since they are already verified by the 911 dispatcher. From here, the interaction between the authorities and the edge is the same as in figure 3, where a behavior is being issued and then being enforced by the edge.

AUTONOMOUS VEHICLES IN HILF

Once a message is transmitted to a vehicle from a local edge infrastructure, the vehicle must process this information to perform a forced behavior. For a forced behavior to occur, the vehicle in question must be an autonomous vehicle. Thus, a basic understanding of how an autonomous vehicle functions would be beneficial. One thing to note here is that current autonomous vehicles leverage artificial intelligence (AI) and deep learning to acquire certain functions like path learning or perceiving its environment. In a HILF scenario where forced shutdown occurs, there is not much AI needed since the behavior is already verified by the edge and learning is not needed to force shutdown a car. However, in a HILF event where forced parking is enacted, AI and deep learning is required to guide the vehicle away from the impact zone and safely find a parking space.

This section covers hardware and two general architectures for decision making, learning and path planning of autonomous vehicles. At a high-level view, there are two architectures that encompass how an autonomous vehicle interacts with its environment. The first architecture is End2End communication where sensors are directly connected to control outputs. The second architecture uses AI and deep learning algorithms which is devised of four components. Grigorescu defines these four components as deep learning methods. These components are: perception and localization, high-level path planning, behavior arbitration, and motion

controllers. A vehicle about to travel on a given path must first perceive and understand its surroundings. Secondly, the vehicle must plan a high-level path to the destination. Thirdly, the vehicle must have a local path plan like merging into a lane. Finally. A motion controller reacts to errors and corrects them if things go unplanned.

End2End involves deep learning algorithms, such as convolutional neural networks (CNN), to map sensors directly to steering commands. Unlike the pipeline structure used in the four components, high level images or map are used as input for direct processing. For example, NVIDA's implementation of end-to-end learning involves passing a raw image to a CNN where road features are detected. This approach involves deep learning training use real world data captured by a camera. Another approach for designing an End2End system is using deep reinforcement learning (DRL) using simulations. Sallab, Abdou, Perot, & Yogamani (2017b) uses DRL in a car racing simulation game called TORC to determine steering commands.

CONCLUSION

ITS is a vision that provides safety, mobility, and environmental safety for drivers. It is no wonder that so many areas across the world are investing in building an ITS infrastructure. Implementing an ITS infrastructure reduces traffic collisions and congestion leading to more rich fuel free environment. Smart vehicle communication is categorized as two ideas: V2V and V2I. Using dedicated short-range communications, we can take full advantage of V2V and V2I to provide benefits such as collision or blind spot detection. To help realize these benefits, a system must be put into place. Gupta and Sandhu propose a cloud/edge hybrid solution that aligns with the ITS vision. This proposed system is designed around V2V and V2I to provide safety, mobility, and a smart driving experience for drivers.

Using this system also provides a more secure transmission of messages through the implementation of the edge. Security is essential; a typical autonomous vehicle has millions of lines code with many sensors, actuator, and applications, giving a broad attack space for a potential attacker. The edge can be used for security with ABAC while the cloud and cloudlets provide cloud computing services such data warehousing and analytics. PKI enrollment, used by the cloud, is also be used to establish trust between users to negate certain malicious attacks such as the infamous man in the middle attack.

The hybrid/edge cloud system architecture constitutes of four layers. These layers are there to help abstract or conceptualize vehicle and edge communication using cloud technology. The bottom layer is the object layer where objects or clustered objects reside. This layer is responsible for sending attributes to the higher layers. The layer above the object is the virtual layer. The virtual layer acts as a backup for objects. Vehicles communicate directly to each other's backup or shadow to maintain a consistent flow of data. If an internet outage were to occur, vehicles could connect to a backup if it's available. The cloud layer handles policy

enforcement with the cloudlets and data analytics and PKI in the central cloud. The top layer, the applications layer, is responsible for interfacing applications directly with users.

While Gupta and Sandhu explored many use cases with their system, most of these use cases explored here are more low impact and high frequency. The main goal and contribution this paper provides is exploring high impact low frequency events. Some HILF events explored include: a university shooting and a weather disaster. The edge capabilities allow communication between a variety of vehicles. A witness at a school shooting can securely interact with police authorities and a forced response can be enacted. Any autonomous vehicle can connect with the edge. A police authority receiving a 911 call, can for example, connect with the edge to issue a forced response. Using AI, a car can be steered clear out of the affected area.

Policy enforcement involves evaluating JSON formatted attributes. The National Weather Service has an API that uses a JSON data link format to process attributes to relay the weather forecast. Using the same data format, the National Weather Service API can connect to edge and pass on or receive attributes. Location coordinates passed from the vehicle can be sent to the API for the weather forecast. Weather alerts can also be received if a region, such as a state, is specified.

In future work I could incorporate more high impact events. The COVID pandemic maybe a possible area to explore. The edge can notify users the local statistics of the pandemic. For example, the edge can alert users the total or new cases in the local area. The Our World In Data (OWID) website provides JSON attributes such as total deaths, cases, and cases per million. Another area that could be explored is detecting hazardous material. Hazardous materials such as toxic or combustible substances can be detected by local authorities and warnings can be sent out.

REFERENCES

[1] M. Gupta, J. Benson, F. Patwa, and R. Sandhu, "Dynamic groups and attribute-based access control for next-generation smart cars," in *Proc. of ACM CODASPY*. New York, NY, USA: ACM, 2019 [Online]. Available: http://doi.acm.org/10.1145/3292006.3300048

[2] M. Gupta and R. Sandhu, "Authorization framework for secure cloud assisted connected cars and vehicular internet of things," in *Proc. of ACM SACMAT*, 2018.

[3] M. Gupta, J. Benson, F. Patwa and R. Sandhu" Secure V2V and V2I Communication in Intelligent Transportation using Cloudlets", 2020.

[4] Sychyk, Anzhela. "Edge Computing vs Cloud: Which Is More Efficient?" *TechTalks*, 18 Feb. 2020, bdtechtalks.com/2020/02/14/cloud-vs-edge-computing/.

[5] ITS Research Fact Sheets - Benefits of Intelligent Transportation Systems. (n.d.). Retrieved from https://www.its.dot.gov/factsheets/benefits_factsheet.htm

[6] "Automotive & Transportation." *Intelligent Transportation Systems Market Size, Share & Growth, 2026*, Fortune Business Insight, Mar. 2020, www.fortunebusinessinsights.com/intelligent-transportation-system-market-1020.

[7] "University of North Carolina Wilmington" 2018 "CRIME REPORTING, TIMELY WARNING, AND EMERGENCY RESPONSE PROCEDURES" 05.505,https://uncw.edu/policies/documents/05.505%20crime%20reporting%20and%20emergency%20response.pdf

[8] "Critical Incident Notification System: SecureTech Systems." *Secure Tech Wave*, www.securetechwave.com/.

[9] "Penn State Policies." *Active Attacker Policy | Penn State Policies*, policy.psu.edu/policies/sy41.

[10] "In This Section." *Shooter - Emergency Procedures - Miami University*, miamioh.edu/campus-safety/emergency-procedures/shooter/index.html.

[11] Grigorescu, S, Trasnea, B, Cocias, T, Macesanu, G. A survey of deep learning techniques for autonomous driving. *J Field Robotics*. 2020; 37: 362– 386. https://doi-org.liblink.uncw.edu/10.1002/rob.21918

[12] Sallab, A. E., Abdou, M., Perot, E., & Yogamani, S. (2017b). Deep reinforcement learning framework for autonomous driving. *CoRR*.