Internet of Autonomous Vehicles

Puya Fard
Computer Engineering
California State University,
Fresno
Fresno,CA

Alvin Nguyen
Computer Engineering
California State University,
Fresno
Fresno, CA

Anthony Michael Herrick
Computer Engineering
California State University
Fresno
Fresno, CA

Anthony Sandoval
Computer Engineering
California State University,
Fresno
Fresno, CA

Travis Anthony Davison
Computer Engineering
California State University, Fresno
Fresno,CA

***Abstract*— Understanding the complexities and issues surrounding the network capabilities when it comes to deploying Autonomous Vehicles within limited parameters.**

Keywords*— Basic Safety Messages (BSM), Internet of Autonomous Vehicles (IoAV), Internet of Things (IoT), Quality of Service, (QoS), Vehicle to Infrastructure (V2I), Vehicle to Vehicle (V2V), Vehicular Ad Hoc Networks (VANETs), Wave Short Message Protocol

I. Introduction

The Internet of Autonomous Vehicles is one of the leading fragments for the IoV. There has been a vast amount of development in trying to create a Full Automation Autonomous Vehicle or a Level 5 using different protocols in a network [1]. The factor of reliability and safety are two of the most important pieces when it comes to an autonomous vehicle that heavily relies on the infrastructure of the network. Every nanosecond matters and a network has to be able to receive and transmit every piece of data that is coming in from a vehicle at all times to ensure safety. As traffic changes every second around a vehicle, the network has to keep track of not only the vehicle but all the gadgets inside as this data is instrumental to the function of the network.

There are many different protocols when it comes to deploying an autonomous vehicle on the network. The majority of the protocols have many specific use cases while others can be used in a variety of networks. To enforce safety and understand the risk of these specific protocols the QoS over a wireless network is handled by the BSM architecture in order to transmit and detect data over the network if any of its packets were not received correctly making this a priority.

II. Protocols

A. *Wave Short Message*

This protocol supports collision avoidance. The technology relies on standards based interoperability. Format of WSM protocol: variable-length header and variable-length payload. These Services can be completed by IPv6 or WSMP protocol stack.

B. *Collision Avoidance*

Some of these protocols are:
Inertial Measurement Unit (IMU), map matching algorithm, Dedicated Short Range Communication(DSRC) which its subtree contains Wave Short Message protocol that is already covered in this report, and sensing technology with traditional GPS. However, the assumption in these protocols is that all the vehicles are capable of communicating with each other in order to provide a reliable collision avoidance system.

C. *Dynamic Source Routing*

This protocol can configure itself without human interface in order to complete the routing protocol for wireless networks. Two of the main components are Route Discovery and Route Maintenance.
Route Discovery: calculates the most optimum route for a transmission between the source-destination.
Route Maintenance: Makes sure that the transmission route remains optimum and loop-free as if conditions in the network might change at any given time.

D. *Ad-hoc On-demand Distance Vector*

Vehicular Ad-Hoc Networks (VANETs) is an efficient routing strategy which is focused on enhancing the packet delivery ratio in all cases, including the random packets. This protocol is from the family of Mobile Adhoc Networks transformed into a system where vehicle to vehicle and RSU becomes data is broadcasted among the vehicles. These packets of data serve the purpose of a transportation correspondence framework that intermingle information and GPS communication and management in order to lessen the roads turned dynamically as a car is in motion on the road..This protocol has two routing algorithms: Route Discovery and Route Maintenance.

E. *Destination-Sequenced Distance Vector routing*

Destination-Sequenced Distance Vector (DSDV) is a protocol that uses hop counts in order to calculate routing while proactively filling up a look up table for each node in the network. The routing table is created by each node sharing their distance with their neighbors. In this way, a collection of distance vector tables are created among the neighbor nodes. There are two main mechanisms used in DSDV routing updates: Periodic updates, which broadcast their routing table every 15s, and Trigger updates which are considered as mini updates among the periodic updates and its sent out whenever a new routing table needs to re-create its routing table based on incoming packets.

III. Network Architecture

A. *Peer-to-peer*

The implementation of a Peer-to-peer network would be vehicles within a close proximity having the ability to communicate with one another. With this ability they can send key information like GPS location, and Speed to notify cars that are traveling closely, and with that information programs can be written so cars can alleviate crashing into one another. With this network architecture, we presume that the traffic flow will be improved, while still dense, but collision avoidance will be mitigated.

B. *Cloud*

A cloud based implementation would include all vehicles communicating with a cloud based server. They will receive information about slow downs or dense traffic and be able to effectively route around it. With this, traffic on surface streets might increase, and because of this, there may be more slowdowns, and more collisions on surface streets. We believe the congestion would be slower here because of the travel time from the cars to the cloud server.

C. *Peer-to-peer and Cloud*

A combination of the two Cloud and Peer-to-peer will most likely lead to best results for our purposes. Combining the routing ability from the cloud with the traffic flow, and awareness that comes with the Peer-to-peer system will lead to better traffic flow with less collisions. The downside of this architecture would be the added complexity.

IV. Simulations

A. *Network (Packet Tracer)*

Packet tracer will be used to create the network parameters in order to run our next simulation using MATSIM as well as make comparisons between the networks. We will measure the speed of the implemented network. Which will give an accurate view of the delay that will be present along with how much data can be sent. Along with this, the overhead of the network and other network parameters will serve as a good benchmark for comparison for the different implemented architecture as a whole. The idea is to use this delay and data to control the traffic conditions with MATSIM. A comparison between the different architectures will be done in packet tracer, but most of the value will come from traffic observations.

B. Physical (MATSIM &Eclipse, Alt. VIA)

The physical simulation will be accomplished using Matsim in conjunction with either Eclipse or VIA as an alternative. In doing the physical simulation, our focus is placed on traffic flow, routing, collisions, and route time. With these metrics, in conjunction with the benchmarks that were found in the network simulation, we will be able to very accurately compare and contrast the different network implementations. Traffic flow and collisions are our main focus here in doing this simulation. Routing, and route time is fairly closely tied to traffic flow, with better routing, traffic will flow faster, and with faster flowing traffic, route time will be decreased. With these being our main focus, we will be able to see how much safer a certain implementation is, and the time saving capabilities for it.

V. Plans For Phase II

A. Implementation

To start off work will be divided amongst the group members who will be assigned to which architecture. The main focus of these architecture implementations is to examine the deley within each of the network architecture types. Using Packet tracer will give us a better idea of how the networks can be implemented and which one is the most viable to use in an actual implementation. One person or two people will also work on the MATSIM implementation, learning how MATSIM works and creating our scenarios. This will be assigned to one person, but can change to two people depending on how much the workload in MATSIM ends up becoming. The ideal schedule for our project right now is to finish the packet tracer networks first then finish building the traffic scenario. If the all three network architectures end up being behind then we would have to test the traffic conditions with manual inputs while the packet tracer people catch up. If only one or two of the architectures ends up with errors then we can easily use the one working architectures for testing in MATSIM, these are only scenarios where things go wrong and we encounter issues so we can have a basic backup plan.

B. Testing

After the testing phase will be the implementation of the different network architectures. Firstly, we will test the finished versions of our architectures in packet tracer, and if there are no errors then we will move to the physical simulation in MATSIM. Although, if there are errors we will not be able to test in MATSIM with actual data. If the implementation of MATSIM is faster, before we get real data from the three architectures we can test MATSIM out using hard coded inputs. This plan of the phase can be a lot longer depending how many errors we encounter and how often we have to go back to the implementation phase to fix the errors. Once everything is properly functioning then we will record our data to get ready for the presentation phase. We have completed most of the research in phase I of the project, but moving to phase two will require more research on how we can put the network together with the physical implementation.

C. Presentation

The presentation phase will be less planned out because we will have to follow the rubric for the final. The general idea is that we will probably have a powerpoint presentation that includes our findings. The findings could include: how fast the cars get out of congestions, speed of the communications across the network, and the efficiency of the three network architectures compared to each other. We will also include videos of the traffic simulations to show the network and how fast the communication can actually be.

C. Hypothesized results

Our main hypothesis is that the combination of peer-to-peer and cloud based architecture will have the best traffic flow, speed and less collisions because the two systems can make up for each other's faults that may occur. The second best architecture we predict would be the peer-to-peer connection since these cars would be able to communicate faster than to the cloud and send each of the key information like GPS, speed, and sensor data. We hypothesize that this will not help too much when it comes to congestion, but the collision avoidance will have a lower percent rate than the cloud based architecture. We believe the cloud based architecture to be the weakest of the three systems because each car has to send packets to the cloud based server and it takes more time this way. The packets have to be sent from each car, then the server has to manage those packets, and finally send the packets back to the automated vehicles to perform.

References

[1] *Automated vehicles for safety*. NHTSA. (n.d.). Retrieved April 8, 2022, from https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[2] An Enhanced Ad Hoc on Demand Distance Vector Routing Protocol for Vehicular Ad Hoc Networks (VANET's) (March 19, 2019). Singhal, Parveen Kumar and Chaubey, V.K. Retrieved April 8, 2002, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3355137