Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 33–42

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2021)
November 1-4, 2021, Leuven, Belgium

A Smartphone VANET Based Forward Collision Detection System

Jacob Speiran*, Elhadi M. Shakshuki

Jodrey School of Computer Science, Acadia University, Nova Scotia, Canada

Abstract

Though a topic of heavy theoretical research, in the real-world Vehicle Ad Hoc Networking (VANET) faces one great challenge: lack of hardware adoption. There are few vehicles manufactured today that have the On–Board Units (OBUs) necessary for vehicular networking. Most cities around the world are not yet equipped with Road Side Units (RSUs), which are required in many VANET schemes for basic functionality. Current literature tends to assume that the existence of RSUs, OBUs and other network infrastructure will become commonplace in the near future. This is simply not feasible from a monetary and manufacturing perspective, especially for rural or poorer areas.

In order to deal with this lack of hardware new approaches must be considered. One possible approach is to use smartphones in place of dedicated OBU hardware. To this end, a basic platform for a smartphone–based peer–to–peer VANET system is explored [1]. Surprisingly, no real–world applications have yet been tested on it. In order to investigate the effectiveness of this platform in a real–world scenario and whether this solution to the hardware problem is viable, this paper is aimed to implement a forward collision detection system in Veins simulator. To test the feasibility of a smartphone VANET system in the real–world and demonstrate its feasibility, we utilized a real–world data gathered from published articles.

Keywords: VANET; Smartphone

* Corresponding author.
E-mail address: 144775s@acadiau.ca

10.1016/j.procs.2021.12.208

1. Introduction

It is largely agreed that V2X (Vehicle to Everything) VANET applications can improve the safety of vehicles significantly. However, it is impossible to benefit from this safety improvement without the accompanying infrastructure. Retrofitting vehicles is incredibly costly. Additionally, RSUs and other infrastructure do not yet exist to enable vehicles to communicate. With this in mind, a new solution to the hardware problem must be found so that we can begin to enjoy the benefits of VANET technology as soon as possible.

One possible solution to this issue is the use of smartphones as nodes in a VANET. At this point in time, the vast majority of people have a smartphone. Each smartphone is capable of using a GPS to determine location, direction and speed. They are also capable of short–range communication, using different technologies such as WiFi Direct on Android devices. In light of these benefits, the smartphone may be a perfect choice to use as an on–board unit.

Galeana-Zapién, H., et al. have developed a smartphone-based platform for secure multi-hop message dissemination in smartphone VANETs [1]. The developed platform is tested in a real–world application and they obtained some experimental results in terms of distance and speed of packet transfer. This proves that the VANET does in fact work at a basic level. However, can it be used in a more complex, safety critical application? Does such a platform hold up in scenarios where lives are potentially on the line? This has yet to be tested.

2. Roads

The roads that we use everyday are incredibly dangerous and unpredictable. Vehicles are constantly coming and going, turning, accelerating and decelerating. Rear end collisions, where a moving vehicle strikes the vehicle in front of it, are a major source of injury and death on roads. These kinds of collisions occur when a driver is forced to decelerate rapidly due to some dangerous condition on the road. Any vehicles following the decelerating vehicle often have little time to react, especially on highways where vehicles are moving extremely fast over the speed limit. In some cases, multi vehicle pileups can occur. One vehicle brake abruptly in order to avoid some hazard on the road, and the vehicle(s) behind it cannot react in time and collides with it. The vehicle behind that one collides with him, and so on in the form of a chain reaction. This can cause major vehicle damage, injuries and death. It can also bring entire highways to a standstill.

How do such multi vehicle collisions happen? Let us assume a decelerating vehicle ***A*** is followed by two vehicles, ***B*** and ***C***. While ***B*** can easily view ***A*** decelerating and may have time react, ***C***'s vision of ***A*** is obstructed by ***B***. This means that ***C*** may still strike ***A*** or ***B***. This issue is compounded when many vehicles are following one another, leading to multi vehicle pileups. On major highways, hundreds of vehicles may be present- and a single crash can quickly lead to a large–scale tragedy.

In many cases, it is possible to avoid these collisions if the driver had a few more milliseconds to react. According to Hoque, F. and Kwon, S. [2], if a driver has an extra 500 ms to react in an emergency situation, 60% of chain collisions are avoidable. To this end, they developed forward collision warning system. When a vehicle meets a certain deceleration threshold, the warning system outputs a packet to all nearby neighbours, warning them that an abrupt deceleration is occurring. These warnings can then be delivered to drivers, giving them more time to react. It is this type of system that we would like to develop to test the feasibility of smartphones for VANET use.

3. Proposed system architecture

As a basis for our proposed system, we will be making use of the smartphone–based platform as defined in [1]. In [1], Android devices make use of WiFi Direct to form a peer–to–peer vehicular network. The internal GPS of each smartphone, and optionally an external GPS, are used to track location, longitude, latitude, speed and direction. Using WiFi direct and GPS a basic V2V VANET can be established. To deal with the issue of security, a novel security method, Asymmetric Certificateless Signature Scheme (ACSS) is presented. This method allows for cryptographic security that removes the need for a third party, thus bypassing the issue of external services such as RSUs or Cloud infrastructure for the verification of messages.

Asymmetric security methods are highly important, as the main draw of the smartphone approach is the ability to bypass the need for external hardware such as RSUs. Our other two algorithms are chosen to address two problems in the smartphone VANET: efficient packet forwarding and efficient selection of packets to decrypt.

4. Packet forwarding

Our chosen packet forwarding algorithm is two-way Intelligent Broadcasting with Implicit Acknowledgement (2I-BIA), as detailed in [2]. In selecting a packet forwarding method, we are concerned with limiting the amounts of redundant packets on the network in order to improve network efficiency. Smartphones are not incredibly efficient or powerful network devices; they lack processing power and signal range. Therefore, having as few packets on the network as possible is incredibly important. To this end, 2I-BIA uses an implicit acknowledgment mechanism in order to limit packets on the network. In general, packets in a VANET are forwarded from vehicle to vehicle in a multi-hop manner. The originating vehicle sends a packet to its immediate neighbours, and each neighbour then forwards the packet to its neighbours and so on until the packet reaches all nodes in the entire network. Generally, in order to save time and network resources, acknowledgment packets are not used in VANET environments. Instead, packets are periodically re-transmitted in order to achieve maximum network saturation.

However, this is an inefficient solution. For instance, consider a vehicle ***A*** in front of a vehicle *B*. If ***A*** broadcasts a packet to ***B***, and ***B*** successfully receives the packet and begins to repeat it, there is no need for ***A*** to continue broadcasting in the direction of ***B*** as ***B*** will already be passing this packet on to its neighbours behind. However, ***A*** will continue rebroadcasting the packet in case the packet was not received, inundating ***B*** with unnecessary information, which ***B*** will continue to repeat. In the standard VANET scheme, ***B*** has no way to tell ***A*** it has received the packet.

With 2I-BIA, this issue is resolved by treating packets that have already been seen as implicit acknowledgments. When a vehicle in the network receives a packet that it has already broadcast, it treats this as an acknowledgment that its neighbors have received the packet and stops broadcasting that packet. However, it only stops broadcasting if the packet has come from the opposite direction. This ensures that the emergency message has spread behind the vehicle that sent it, that is, down the line of vehicles from the originator. In our example above, when ***B*** receives the packet from ***A*** and begins to rebroadcast, it will send a copy of ***A***'s own packet back to ***A***. When ***A*** receives this copy, it will recognize that it has come from the opposite direction of itself. It will stop broadcasting its packet as it knows its neighbours have received this message, since they are repeating sending back the same packets that it has already sent. This is particularly effective in our own area of interest, highways, where vehicles are positioned in linear columns.

Towards this end, our proposed algorithm in pseudocode is presented in Algorithm 1. In this algorithm, we used two lists. The first list is used for storing packets for possible future rebroadcasting and named *to_rebroadcast*. The second list is used for storing packets that are seen and acknowledged and named *acknowledged*. In this way, messages that are acknowledged do not get re-added to the rebroadcast list.

Algorithm 1: …

1. list to_rebroadcast
2. list acknowledged

3. **if** new warning message {
4. add to rebroadcast list
5. } **else** {
6. **if** warning message is in rebroadcast list {
    i. compare directions of the new message with the previous
    ii. **if** its from the opposite direction {
    iii. *//this is an acknowledgment, the message has reached the vehicles behind us*
    iv. move it to the acknowledged list, stop rebroadcasting
    v. } **else** {

vi. ignore and keep rebroadcasting
vii. }
7. } **else if** warning message is in acknowledged list {
i. ignore it, we've already acknowledged this message
8. }
9. }

5. Decrypting packets efficiently

Our next algorithm deals with inefficiency in decryption. The Asymmetric Certificateless Signature Scheme security method presents a problem: verifying packets is mathematically intensive and thus takes some time. In [1], it was found that at the lowest bit level of security (80–bit security), receiving and processing a packet takes on average 336.28ms. If each packet takes this long to process, only one or two can be processed in the 500 milliseconds required for them to be of any use in helping a driver avoid an accident.

Smartphones are not powerful devices, and thus decryption of data is expected to be slow. In order to combat this inefficiency a packet optimization algorithm is employed. Instead of decrypting packets in a First–In–First–Out order, packets are instead ordered based on a variety of factors. In our proposed system, we will utilize Relative Time Zone (RTZ) Priority algorithm presented in [7]. The main aim of this algorithm is to prioritize packets that pose the greatest threat to the receiving vehicle. Under this algorithm, the road is divided into four quadrants around each receiving vehicle ***V***, as shown in Figure 1. The quadrants are assigned as follows:

- The bottom–right quadrant contains vehicles that are ahead of ***V*** traveling in the same direction (Quadrant 1).
- The bottom–left quadrant contains vehicles that are behind ***V*** traveling in the same direction (Quadrant 2).
- The top–right quadrant contains vehicles that are coming towards ***V*** traveling in the opposite direction (Quadrant 3).
- The top–left quadrant contains vehicles that have already passed ***V*** traveling in the opposite direction (Quadrant 4).

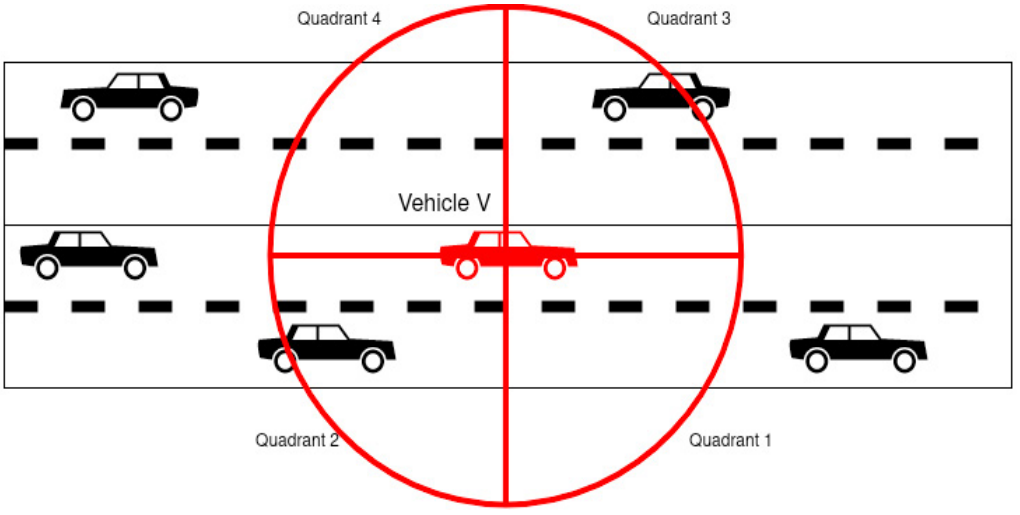

Fig. 1. RTZ Quadrants

The algorithm prioritizes messages from quadrants one, two and three over quadrant four as vehicles in quadrant four are passed the vehicle ***V*** and heading away from it, meaning they pose little danger. On top of this, a zone system is implemented, putting vehicles in zones surrounding ***V***, starting with a "danger zone" very close to the

receiving vehicle ***V***. Packets are prioritized at a zone level, meaning the closer to ***V*** a message arrives from the higher it will be placed on the list.

Finally, packets are ordered on a relative time basis. This refers to the relative time calculated for a vehicle to reach the receiving vehicle ***V***. Using all these metrics, packets are essentially ordered on how likely they are to cause an emergency situation for a receiving vehicle ***V***. In this way, messages from dangerous vehicles are decrypted first.

At a high level our proposed architecture emphasizes efficiency and limiting the number of packets traveling on the network at any given time. By limiting redundant packets and only decrypting high priority packets, we can hopefully bypass the issue of smartphone inefficiency.

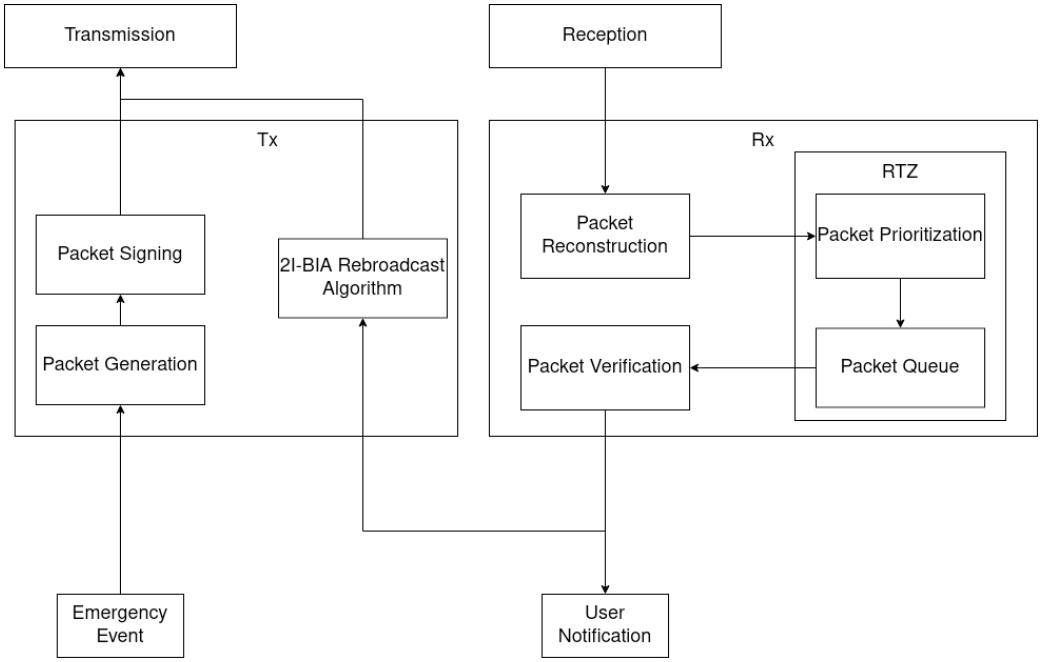

Fig. 2. Architecture overview

6. Experimental Method

To demonstrate the feasibility of the proposed approach an experiment was performed. Our experiment is executed in OMNeT++, Veins and SUMO simulators. The simulation code is implemented in C++ using the OMNeT++ simulator, SUMO and the Veins framework. Each vehicle has its own code that's run at each time step of the simulation. This code can perform actions on various event triggers, such as receiving a message, updating vehicle position and so on. Our experiment is designed to test the basic functionality of our proposed system. The experiment is organized as follows:

There are ten vehicles driving in a straight line down a highway. The highway is a 5000-meter single lane highway headed in one direction. The vehicles are numbered 0 through 9, with 0 at the front and 9 at the back. The vehicles travel at 27.78 meters per second, which equates to roughly 100 kilometers an hour. The vehicles are spread evenly along the route at roughly 30 meters from one another. The distance between vehicles as well as each vehicle individual speed varies, as SUMO modifies vehicle speeds based on a variety of different factors. These factors include road speed, driving style/aggression and following distance calculations based on the speeds of drivers around each vehicle.

After 31 seconds have passed in the simulation, vehicle 0 has some sort of emergency and is forced to reduce speed to a quarter of its current speed (25km/hour, or 6.94 meters per second). At the same instance it sends a

message to all vehicles in range informing them of the emergency. This message is to be propagated using our forward collision detection system from vehicle to vehicle, informing them of the sudden slow–down of vehicle

There are a few things to note about this experiment. The 30-meter distance between vehicles means that, due to the maximum limit of 100 meters on messages in our system, the front vehicle, i.e. vehicle 0 can only reach the two vehicles behind it with its messages. In order for messages to arrive at vehicles further back, they must be propagated successfully by our message propagation algorithm.

7. Results

To begin with, the system works as expected. In each run there are nine vehicles that need to receive the emergency warning message, vehicles 1 through 9. Vehicle 0 is sending the message, and therefore does not receive it. In our experiments, we performed 20 runs in total. Therefore, 180 emergency warning messages should have been received if the message was propagated to all vehicles in the column. This is, in fact, the case. All 180 vehicles did receive the message as planned, which means the system is propagating messages successfully.

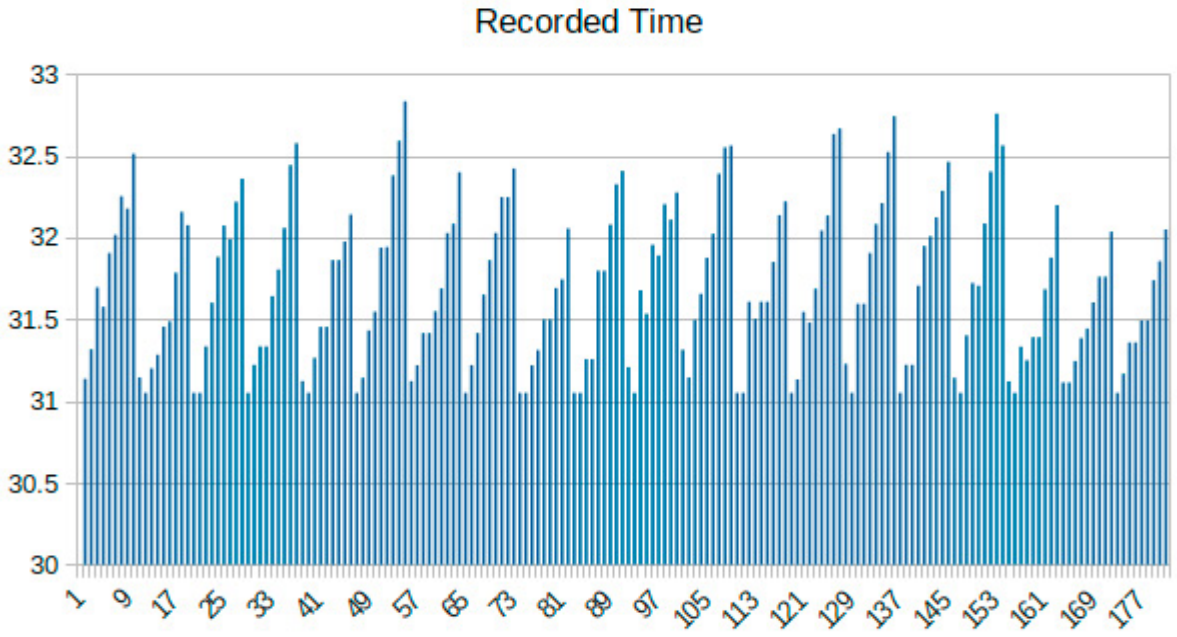

Fig. 3. Reception time of vehicles across runs

Of more interest is the timing of messages received. How long did it take, following the accident being first recorded, for vehicles to receive the crash message? Vehicle 0 begins propagating the emergency warning message at t = 31.05 seconds. Figure 3 presents the recorded time, in seconds, that each vehicle received the warning message. The x axis represents each vehicle along each run – the first 9 entries are vehicles 1-9 in run 1, the second 9 entries are vehicles 1-9 in run 2 and so on. The Y axis represents time taken to receive the message. As shown in Figure 3, emergency warning messages were received roughly between 31.05 seconds and 33 seconds. On average, the reception time for each vehicle was lowest with vehicle one and increased with each subsequent vehicle. This behaviour is expected, as messages are propagated to vehicles at the back of the vehicle column slower than vehicles near the front, where the accident originated.

The maximum time of reception is 32.8314 seconds, and the minimum time of reception is 31.0502 seconds. On average, messages are received at 31.59 seconds. Generally, the maximum reception time is recorded by vehicle 9, the last vehicle in the column, and the minimum reception time is recorded by vehicle 1. In the slowest possible case emergency message propagated the entire network in 1.78 seconds. The average amount of time taken for any vehicle to receive the emergency warning message is 0.539 seconds.

The average time of 0.539 seconds exceeds our target time frame of 0.5 seconds. However, this average is of all vehicles. This includes vehicles that are very far away from the vehicle that originated the accident. For instance, vehicle 9 at accident time is found in run one in 313 meters away. At 27.7778 meters per second (100km/hour) vehicle 9 reaches vehicle 0 in 11.26 seconds, assuming it does not dramatically change its speed. These vehicles are often outside the range where a rapid response to the accident is needed. Closer vehicles had a much better average reception time. Vehicle 5 is found at accident time in 175 meters away. If we take the average reception time of the first 5 vehicles, for instance, the result is 31.315 seconds. This means that the messages are received 0.265 seconds after the accident occurred – within the half second time frame.

Packet drops were modeled based on data gathered in [1]. Packets dropped by this model are counted in each run, as well as packets successfully received. Using this data, we can see how great the effect of sub-optimal Android WiFi hardware has on network propagation. Across 20 runs, there are 1276 packets dropped and 1830 are received, giving a total count of packets sent as 3106. In other words, 41% of packets are lost overall, which is quite acceptable for our purposes.

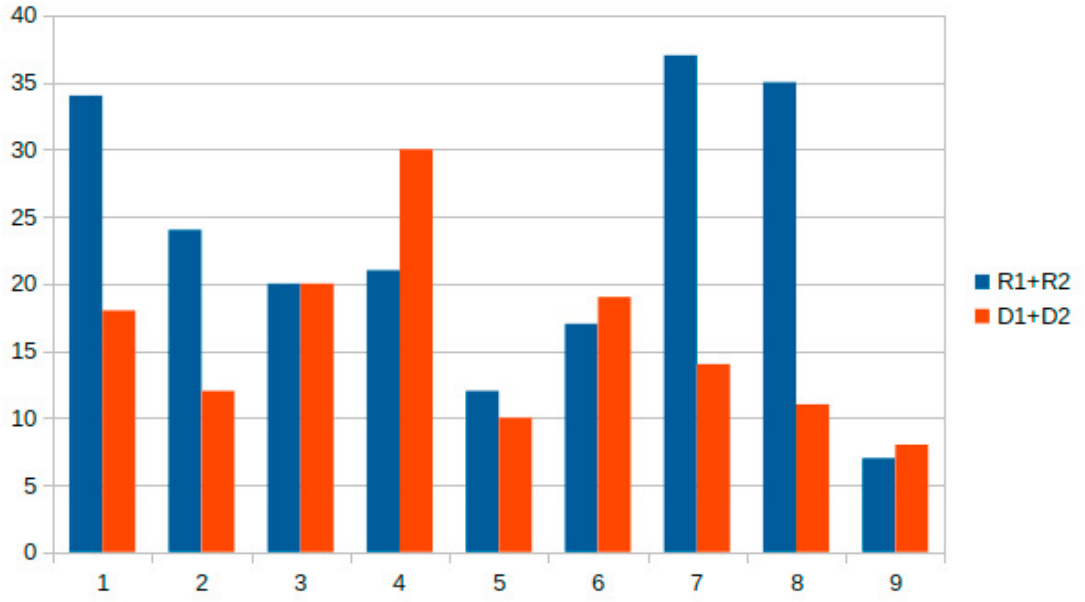

Fig. 4. Packets received/dropped in runs 1 and 2

In Figure 4, the data for received/dropped packets for runs 1 and 2 is displayed. The blue bars display received packets from run 1 and run 2, and the orange bars display dropped packets from run 1 and run 2. By looking at more than one run together, a trend is visible. While vehicles 1 and 2 successfully receive more packets than they drop, vehicles 2 through 6 seem to drop as many or more packets than they receive successfully. Vehicles 7 and 8 receive much more successfully, and vehicle 9 seems to drop as many as it receives. From this data we can theorize that the vehicles in the "middle" of the column (vehicles 2 to 6) have a difficult time establishing proper communications. This may be due to the fact that they are outside the initial range of vehicle 0. Vehicles 1 and 2 will successfully receive the messages from vehicle 0 as they are within the initial transmission range; whereas, the vehicles in the

middle must receive them repeated in a multi-hop fashion. Multi-hop is more prone to failing than a single hop transmission.

Vehicles 1 and 2 are generally in more danger than vehicles further down the line. Vehicle 2 especially is in a dangerous spot, as its vision to the emergency experiencing vehicle may be blocked by vehicle 1, but it is still very close to vehicle 0. The fact that neither vehicles 1 nor 2 lose many packets indicates that they are informed in a reliable manner of a potential emergency. Overall, packet loss seems less of a problem than initially we thought.

For this experiment we look at the effect of the limited transmission range of WiFi Direct. WiFi Direct can reach up to 100 meters. Generally, more range in a wireless network is always better, but of interest is whether the 100-meter limit causes serious packet loss. In our case, OMNET allows messages to arrive at unlimited range. In order to simulate WiFi Direct, if a message came from outside the 100-meter range, we drop it. In total, 6940 messages are dropped due to distance across 20 runs between 10 vehicles per run. On average, each vehicle dropped 34.7 messages that would have arrived if the WiFi capabilities of the smartphone has more range. But, how much more range would be necessary?

At what ranges does these messages arriving from? Figure 5 shows the average distances of messages dropped for exceeding the 100 meters transmission range.

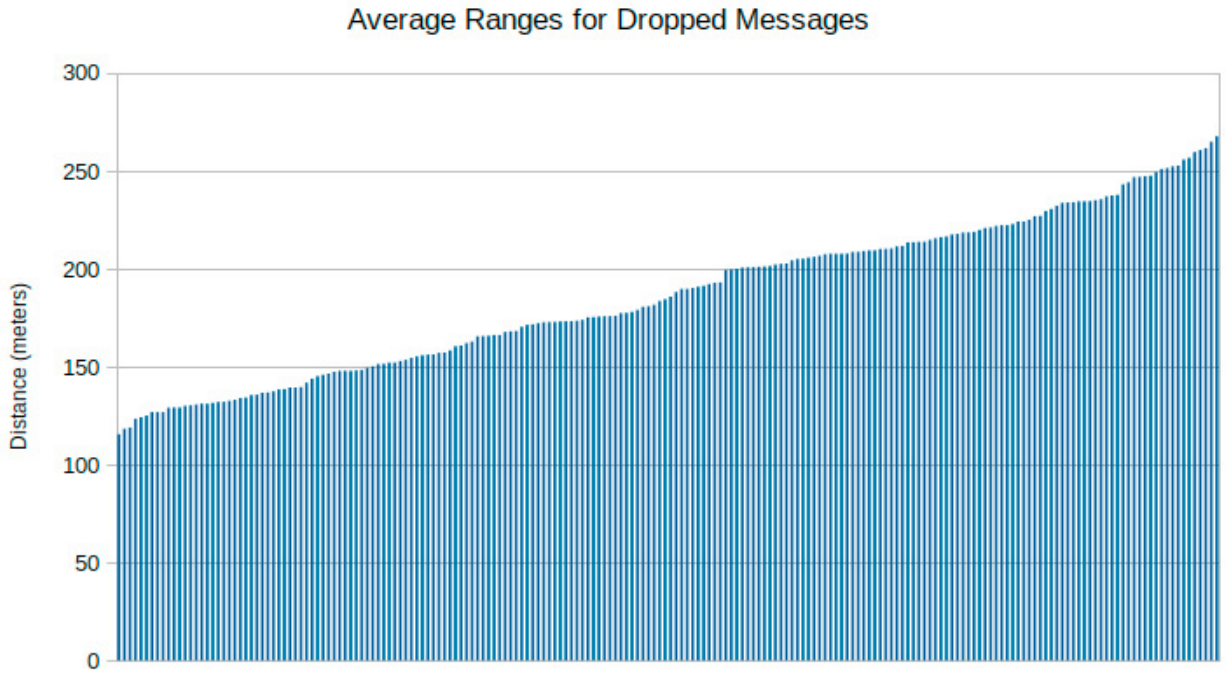

Fig. 5. Average ranges for dropped messages

From Figure 5, it can be seen that messages dropped are evenly distributed in the 120 meter to 280 meter range. That is to say, 34.7 messages per vehicle would not have been lost if WiFi Direct offered an extended range up to 280 meters. The WAVE standard for VANETs uses hardware with a range of 300 meters. Therefore, on dedicated hardware, these messages would have been received successfully, as opposed to the smartphone, where they were lost. However, this did not effect the operation of our system. As was concluded above the system functions and in most cases is fast enough to get the emergency warning messages to the drivers on time. This shows that it would be an advantage to have further range, but it is not entirely necessary.

It should be noted that we did not model processing time or encryption, decryption, or message signing times. We did not model processing time as in most cases it is almost negligible. In [1], it was found that the processing time per packet was about 6 to 8 milliseconds, using complex algorithms and larger packets than we used in our proposed system. This amount of time does not significantly affect the 500 ms reaction time goal. It can also be assumed that our system will be even faster, since packets are smaller than the ones used in [1]. Though a method similar to the packet drop algorithm is considered to factor in encryption/decryption/packet signing times, in the end it was chosen not to model these. This is because the authors of [1] used larger packets in their implementations than we used in our implementation, as well as being somewhat unclear as to how they obtained the numbers and results they did with the impact of their security scheme. Implementing the security scheme itself was considered in order to model its effect on the overall findings. However, this method is abandoned as such systems are beyond the scope of this paper.

We are still able to consider the effects of encryption/decryption/message signing, however. For instance, the average reception time found in the experiment for the first five vehicles following the accident originator is 265 milliseconds. This can be interpreted as follows: given the reaction time window aimed for is 500 milliseconds, the system should be able to encrypt/decrypt/packet sign within 235 milliseconds or less. In this manner, the message is received and cryptographically verified within the 500 millisecond required window.

We are still able to consider the effects of encryption/decryption/message signing, however. For instance, the average reception time found in the experiment for the first five vehicles following the accident originator was 265 milliseconds. This can be interpreted as follows: given that the reaction time window aimed for was 500 milliseconds, the system should be able to encrypt/decrypt/packet sign within 235 milliseconds or less. In this manner, the message will be received and cryptographically verified within the 500 millisecond window required.

8. Conclusions and future work

The problem of hardware adoption in the field of VANETs is a major one. Though hardware does exist, it is not widespread, neither in vehicles nor in infrastructure. In order for VANET schemes to be considered by major manufacturers and governing bodies, their benefits must be proven, and in order for such benefits to be proven large scale adoption is required. This presents a “chicken and egg” problem, especially in rural areas or large countries such as Canada, where installation of hardware like RSU’s will almost never happen due to the great distances involved.

In this paper, we have considered a new approach to VANETs – the use of smartphones as OBUs. A useful test application has been chosen based on a variety of parameters aimed at proving the smartphone platform to be useful. An architecture was proposed that intended to address the shortcomings of smartphones. Finally, the proposed architecture was implemented in a simulator in order to test it in a real–world setting.

In this paper, a set of experiments proved that in a normal, fairly basic setting, the system functions as intended. On a basic level, messages were received as expected by all vehicles in the network. Time taken to receive emergency messages fell within 500 millisecond benchmark, except for vehicles fairly distant from the emergency experiencing vehicle. Packet dropping was found much less severe issue than previously thought. Generally, the closest vehicles receive packets the most successfully, but the losses by more distanced vehicles do not impede the function of the system. Finally, the limited range of a smartphone was found irrelevant. Vehicles travelling tightly on a highway, tightly enough that they must be informed of an emergency very quickly are, generally, close enough that they will receive the information they need in a timely manner. More range would be beneficial, but it is not entirely necessary. Overall, the smartphone system can be seen to warrant further investigation. Fundamentally it works well.

This work has found that the smartphone VANET platform is worth exploring in further detail. It has shown that safety critical applications are feasible on a smartphone VANET, and further that applications of greater complexity should be explored for the smartphone VANET. The smartphone VANET has shown more than a “toy solution” to the VANET hardware problem. The smartphone VANET is a feasible alternative to dedicated hardware.

There are many ways this work can be extended in the future. Our forward collision system is intended primarily for highway use, which is a fairly simple use case. This is because highways are straight and travel in a single direction on either side. Perhaps expanding this system to work in more complex road systems, such as those found in cities, would be beneficial. Another further expansion to the work would be to develop further applications for the smartphone VANET that are not forward collision detection systems. A myriad of applications exists for VANETs, safety applications or otherwise, and now that the platform has shown to be feasible, implementing more applications would be very interesting. For instance, a traffic information system that passes traffic data between cars and updates a local map service on the smartphone would be interesting. This would entail new challenges, since non-safety data and safety data would be using the same service to send messages, meaning that some sort of prioritization would have to take place.

Acknowledgement

The authors would like to thank Acadia University and Natural Sciences and Engineering Research Council (NSERC) of Canada for their support and funding this research.

References

[1] H. Galeana-Zapién, M. Morales-Sandoval, C. A. Leyva-Vázquez, and J. Rubio-Loyola, “Smartphone-based platform for secure multi-hop message dissemination in vanets,” Sensors, vol. 20, no. 2, p. 330, 2020.

[2] F. Hoque and S. Kwon, “An Emergency Packet Forwarding Scheme for V2V Communication Networks,” The Scientific World Journal, vol. 2014, pp. 1–7, 2014.

[3] “Vehicle-to-Vehicle Communication,” NHTSA, 18-Dec-2019. [Online]. Available: https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication. [Accessed: 05-Feb-2021].

[4] “Vehicle-to-Vehicle (V2V) Communications for Safety,” Intelligent Transportation Systems - Vehicle-to-Vehicle (V2V) Communications for Safety. [Online]. Available: https://www.its.dot.gov/research_archives/safety/v2v_comm_progress.htm. [Accessed: 05-Feb-2021].

[5] R. Frank, W. Bronzi, G. Castignani, and T. Engel, “Bluetooth Low Energy: An alternative technology for VANET applications,” 2014 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 2014.

[6] “Mobile Operating System Market Share North America,” StatCounter Global Stats. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/north-america. [Accessed: 05-Feb-2021].

[7] S. Banani, S. Gordon, S. Thiemjarus, and S. Kittipiyakul, “Verifying Safety Messages Using Relative-Time and Zone Priority in Vehicular Ad Hoc Networks,” Sensors, vol. 18, no. 4, p. 1195, 2018.

[8]: Najm, W., Koopmann, J., Smith, J., Brewer, J., & John A. Volpe National Transportation Systems Center (U.S.). (2010, October 01). Frequency of target crashes FOR IntelliDrive safety systems. Retrieved March 01, 2021, from https://rosap.ntl.bts.gov/view/dot/12066