Reliable and Efficient Broadcast Routing Using Multipoint Relays Over VANET For Vehicle Platooning

Xing Wang, Alvin Lim

Department of Computer Sciences and Software Engineering
Auburn University, AL 36849 USA

Abstract: In this paper, we design and implement a reliable broadcast algorithm over a VANET for supporting multi-hop forwarding of vehicle sensor and control packets that will enable vehicles to platoon with each other in order to form a road train behind the lead truck. In particular, we use multipoint relays (MPRs) for packet transmission, which leads to more efficient communication in a VANET. We evaluate the performance based on simulation by running a platooning simulation application program, and show that with MPRs, the communication in the VANET to form a road train is more efficient and reliable.

Index Terms: vehicle network, ad-hoc network, vehicle platoon, road train, routing, broadcast, MPR

1. Introduction

1.1. Background

A mobile ad hoc network (MANET) is the cooperative wireless network formed by a collection of mobile hosts (MHs) without the support from fixed infrastructure or centralized access point. As a special subcategory of MANET, the vehicular ad hoc network (VANET) is built up of moving vehicles in which each participating vehicle serves as a wireless router or node, and in turn creates a network with a wide range. Due to the increasing demand of road safety and convenience, VANET is emerging rapidly and has many applications, such as vehicle collision warning, security distance warning, driver assistance, cooperative driving, cooperative cruise control, dissemination of road information, internet access, map location, automatic parking, driverless vehicles, etc.

In this project, we design and implement a VANET for self-driving vehicles using a cooperative adaptive cruise control (CACC) platooning application.

As compared to other MANETs, VANET has some unique characteristics, some are attractive while others make it challenging in real applications. For example, in other MANETs, the energy constraint often needs to be one of the most important design optimization criteria since many ad hoc devices rely on batteries or other limited power source. In contrast, this is not an issue in VANETs as modern vehicles could afford powerful computing, communication and sensing capabilities. In most MANETs, nodes are free to move in arbitrary direction and thus the network topology is highly unpredictable. On the contrary, vehicles are allowed to move within only one or two directions on the same road, moreover, road and traffic information is usually available from positioning systems and map-based technologies (e.g., GPS), all make predicting vehicles' movements possible, and leave room for modeling the vehicles' behavior and further optimizing routing through data. However, VANETs have their own characteristics of extremely dynamic topology. Vehicular network will be frequently partitioned and dynamic nature of traffic may result in large inter vehicle gaps in sparse network. The vehicles move at high speed and change their...
position constantly in the dynamic scenarios. Network topology changes frequently as the link between the nodes connect and disconnect very often, and the frequent disconnection occurs more often in sparse network. We also note that in most VANETs, roadside infrastructure is also a vital component, and both types of communication, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), are significant requirements for building an advanced intelligent transportation system (ITS). Figure 1 shows the two types of communication in VANET, in which the roadside infrastructure assists to create a better share of information between vehicles. In this project, however, we will not consider the V2I component for simplicity purpose, and focus on the V2V communication.

The introduction of automation into vehicles in a connected vehicular environment is potentially transformative [16]. To facilitate the research and implementation of automation in vehicles, a classification system based on six different levels (ranging from fully manual to fully automation) was published in 2014 by Society of Automotive Engineers (SAE) International, an automotive standardization body, as “J3016, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems”, which was then updated in 2016, called “J3016_201609” [49]. The first automation driving support system, the conventional cruise control (CCC), allows drivers to drive at a certain speed. In recent years, with the improvement of vehicle control technology, the adaptive cruise control (ACC) system allows a vehicle to drive behind a leader at a certain distance, which improves roadway capacity and traffic safety as well as fuel efficiency [61]. This system is designed to further improve the driving experience such as comfort. Enabling the connectivity between vehicles and roadside infrastructure, ACC application could be extended to form a platoon known as cooperative adaptive cruise control (CACC) [62] and belongs to level 2 automation defined by SAE. With the shared information between vehicles, the CACC allow vehicles in a platoon to maintain smaller headway compared to ACC [64]. We will discuss the platooning system in Section 2.
1.2. Objective and Research Questions

The objective of this project is to design and implement a reliable broadcast algorithm over a VANET for supporting multi-hop forwarding of vehicle sensor and control packets for a CACC system that will enable vehicles to platoon with each other in order to form a road train behind the lead truck. In particular, we aim to use multipoint relays (MPRs) for packet transmission. We expect using MPRs would lead to more efficient communication in a VANET. For comparison purpose, a reliable broadcast algorithm needs to be implemented as well. We will evaluate their performance based on simulation by running a platooning simulation application program.

To better understand the goal, we will first investigate what is known today about how to obtain better communication in platooning, this will be achieved by studying the existing research literature to address the following research questions:

- What characterizes existing literature on platooning?
- What are the communication methods for platooning in literature?
- Which routing methods have been used in literature?
- What solution have been proposed in literature to improve communication?
- Is MPR transmission better than regular flooding in a vehicle platoon?

1.3. Outline

This paper is organized as follows: Section 2 presents an overview of vehicle platooning, which serves as the background of our project. Section 3 discusses routing protocols as applied to VANETs, especially those relevant to the implementation in this project. Section 4 describes the design of the models. In Section 5, we provide the details for implementation and environment setup. Section 6 shows the experimental results in our simulation. And finally, we conclude in Section 7.

2. Vehicle Platooning

Vehicle platooning can be illustrated as a chain of vehicles travelling together at a given speed and distance. It is a concept in the automotive industry that mainly aims at increasing the road capacity and improve the travel efficiency as well as safety. Under cooperative driving, automated vehicles can be organized in tightly controlled groups, also called platoons (or road trains). Throughout this report, we will use platoon and road train interchangeably.

Figure 2 shows an illustration of vehicle platoon as a SARTRE demo (see Section 2.1), in which several vehicles travel along the highway as a single unit. These vehicles follow one another with a small spacing and are “linked” through some communication and control mechanism. The first vehicle in the platoon, the lead truck, continuously provides the others, the following vehicles, with necessary information upon which the platooning could be executed continuing.

2.1. Overview of Platooning Deployments

The idea of automated highway traffic was presented by General Motors way back to 1940 World’s Fair in New York, and the early implementation, in which vehicle platooning plays an important role, can be traced back to 1980s in California [28]. Ever since then, many projects and experiments have shown that platooning is feasible in practice. Here we provide a brief overview of several famous platooning projects over the world, mainly cited from [8] as well as the corresponding project reports.

**SARTRE**

Safe Road Trains for the Environment (SARTRE) is a European Commission Co-Funded FP7 project to investigate and trial technologies and strategies for the safe platooning of road vehicles. The three-year project was launched in 2009. The research and development was carried out by several European auto manufactures with Volvo at the lead. A first practical test successfully took place in December 2010. [48]
The SARTRE definition of platooning implies that the platoon is led by a vehicle which is manually driven by a professional driver, such as a truck. This driver must have a valid licence and is assumed to have additional training for leading a platoon. The following vehicles (trucks and passenger cars) are under automated longitudinal and lateral control of the lead vehicle, so than these drivers are hence able to undertake other tasks such as using mobile phones. The following vehicles may join or leave the platoon dynamically, e.g. leave on arrival at the desired destination. In SARTRE, a car can only be a following vehicle in a platoon.

The project aims to explore technology for platooning on roads without changes to the infrastructure and that it is safe enough to allow mixing with other users of public roads. Expected advantages of platooning include a reduction in fuel consumption, increased safety, traffic efficiency, increased driver convenience and comfort. [7]

One of the key characteristics of SARTRE is that its platooning application requires that V2V communication is used in addition to local sensors in each vehicle. With V2V implies that data can be sent directly from the source rather being indirectly measured locally with sensors. Using only local sensors can lead to lateral and longitudinal instability, increasing oscillations, and unsafe behaviour of the platoon. V2V communication allows sharing of local vehicle signals such as speed and sensor data among vehicles in the platoon. The shared signals are used in the control algorithms of the platoon. [6]

In January 2011, SARTRE made the first successful demonstration of its platooning technology at the Volvo Proving Ground near Gothenburg, Sweden, in which a lead truck was followed by a car. In January 2012, SARTRE made a second demonstration in Barcelona, Spain, in which a lead truck was followed by three cars driven entirely autonomously at speeds of up to 90 km/h (56 mph) with a gap between of at most 6 m (20 ft).

**PATH**
The Partners for Advanced Transit and Highways (PATH) program in California has been involved in the research of connected and autonomous vehicles for over 30 years. The PATH research
on automated platoons was initially motivated by the need to produce a significant increase in the capacity of a highway lane, so that increases in travel demand could be accommodated with a minimum of new infrastructure construction. The PATH kinematic studies of highway capacity showed that it could be possible to increase passenger car lane capacity by a factor of two to three over today’s capacity if the vehicles were driven in platoons of up to ten cars [7]. The gaps between platoons would be long enough to ensure that even in the worst crash hazard condition, with maximum deceleration, a following platoon would be able to stop without hitting the last vehicle of the forward platoon, even while there could be low-speed crashes between vehicles within the forward platoon. An extensive modelling and simulation study of crash safety and capacity showed the advantages of the platoon mode over individual automated vehicles [8]. The PATH studies have been based on the assumption that all vehicles would be automated, including the first vehicle of the platoon, in order to maximize efficiency and remove the potential for driver errors to cause crashes.

PATH first tested the longitudinal control of a 4-car platoon at 4 m separation at highway speeds in 1994, and then developed the 8-car automated platoon for the National Automated Highway System Consortium (NAHSC) public demonstration in August, 1997. Over 1000 visitors were given passenger rides in these platoon vehicles, which did a variety of manoeuvres including lane changing and joining and leaving the platoon as well as normal car following, all under completely automatic control. All of the intelligence, providing for sensor signal processing, V2V communication and coordination, and lateral and longitudinal control, was done on a single Pentium computer running at a 166 MHz rate, so this did not require high computational power. The gaps between the vehicles within the platoon were maintained with a 20 cm RMS error, which is small enough that vehicle occupants felt as if they had a mechanical coupling to the preceding car, while also maintaining a smooth ride quality for comfort [46].

Another focus of the PATH platooning was on heavy trucks, mainly because of the potential for energy saving. Forming trucks in automated platoons of 3 individuals could enable a capacity of about 1500 trucks per lane per hour, which is twice the capacity achievable with trucks driven individually. The PATH experiments on truck platoons have shown the technical feasibility of driving two trucks at a gap of 3 m and three trucks at a gap of 4 m between trucks. These experiments also showed direct fuel consumption savings of about 5% for the lead truck and a range of 10% to 15% for the following trucks under conservative estimations. [11]

Energy ITS

2008-2012 Energy ITS (Japan) was a Japanese national project for International Task Force (ITS) on Vehicle Highway Automation, funded by Japanese Ministry of Economy, Trade and Industry. It aimed at energy saving and global warming prevention with ITS technologies, and has two themes: an automated truck platoon and an evaluation method of effectiveness of ITS on energy saving. The lateral control is based on the lane marker detection by the computer vision, and the longitudinal control is based on the gap measurement by radar and lidar in addition to the V2V communications [59]. See Figure 3.

The project succeeded in developing automated truck platooning technologies for operating four trucks with an inter-vehicle spacing of 4 meters and traveling at a speed of 80 km/h, and achieved to reduce energy by about 15 % (measurement) by the aerodynamic drag reduction, and CO₂ by 2.1 % along an expressway (simulation) when the 40 % penetration in heavy trucks by the roadway capacity increase [60].

Others

The Federal Highway Administration in 2013 funded two research projects in heavy truck platooning (without steering automation). One is led by Auburn University, while the other is led by California Department of Transportation, with UC Berkeley and Volvo Trucks.
2.2. Benefits of Vehicle Platooning

As we mentioned as the main goals for each of the famous deployment, vehicle platooning offers a number of benefits. First of all, most platooning projects aim towards the reduction of fuel consumption for Heavy Duty Vehicles (HDV). HDV platooning for emission reduction and energy efficiency is intensively studied. It is known that driving at a short inter-vehicle spacing to a vehicle ahead results in a reduced fuel consumption and drivers are doing so today with increased stress levels. Since up to 1/4 of the fuel consumption for a typical HDV can be spent on overcoming the aerodynamic drag, forming in a platoon can significantly lower the air drag. Even the leading vehicle takes advantage of platooning at short inter-vehicle distances, due to reduced adverse aerodynamic effects. An illustration of this effect is given in [10], which shows that trailing HDVs traveling at 80 km/h experience a 21% fuel reduction when the distance between the vehicles is 10 m, while the fuel reduction is 16% for an intervehicle distance of 16 m. The fuel reductions for the same vehicles and distances traveling at 60 km/h are approximately 16% and 10%, respectively. Therefore, it is clear that platooning has the potential to provide substantial economic benefits for individual haulage companies. Moreover, since vehicles account for a large proportion of total carbon emissions (20% according to [14], and 1/4 comes from HDVs), lower fuel consumption can substantially make progress toward carbon reduction goals and achieve clear environmental gains.

Apart from reducing fuel consumption, platooning potentially offers other important gains. As vehicles are operated at close distances, the existing road infrastructure can be exploited more effectively and capacity can be increased. For passenger cars, it is argued that a capacity increase of 200% can be achieved [3]. In addition, platooning can provide a major improvement in the utilization of existing road capacity by significantly reducing road congestion.

Because human drivers are generally not capable of safely maintaining the close distances as required by platooning, it is clear that automated driving technologies are needed. The increased level of automation provides a second advantage of cooperative transport technologies since automation is generally believed to have a positive effect on road safety [4]. Namely, automatic systems can usually react more quickly to dangerous situations and can exploit additional information resulting from the communication and cooperation between vehicles. Moreover, the majority of traffic accidents are caused by human errors. Speeding, distractions, fatigue and even drunk driving are some of the reasons. In fact, it was reported that the drivers are partly or fully responsible for over 90% of the traffic accidents [23]. And increased automation has a strong potential to mitigate this factor. Moreover, autonomous driving system and vehicle platooning can greatly improve driver's comfort, in the way that the drivers may delegate certain driving tasks, allowing the driver to shift his/her focus, this can relief stress and especially fatigue for long trips.
This in turn can improve the driving safety.

We summarize the benefits of vehicle platooning we discussed above:

- Improvement of energy efficiency and transportation profitability
- Reduction of environmental impact
- Improvement of road capacity
- Improvement of safety and comfort, etc.

### 2.3. Architecture of Vehicle Platooning System

This section mainly cites [55]. A simple hierarchical framework for decomposing the implementation of a vehicle platooning system (or indeed any intelligent transportation system (ITS)), as shown in Figure 4, has been established and found to be very useful for decades of research and development work, however, it is worthwhile to note that there is a significant spreading from each layer to the next. For instance, a large urban region might have a single network layer, but potentially hundreds of links. Each link could have dozens of clusters of closely coordinated vehicles, each of which could have a dozen or more vehicles [55].

The network layer maintains desired cruising speed throughout the network (in the absence of incidents or failures) by metering access to the network. The link layer takes care of issues such as managing flows around incidents, balancing traffic across lanes, metering the entry rate of vehicles to prevent upstream propagation of congestion shock waves, and assigning suitable speed limits to each lane. Starting from the coordination layer, the emphasis shifts towards individual vehicles and a single platoon of vehicles rather than the aggregate traffic flows. At the coordination layer, vehicle trajectories are planned, information about traffic and road conditions is exchanged among vehicles, and manoeuvres are coordinated among adjacent vehicles, such as lane changing, joining and leaving a platoon, etc. The regulation layer is responsible for the control of vehicle motions, such as steering, acceleration, braking and spacing relative to the preceding vehicles, often the classic closed-loop control is applied. Finally, the physical layer is where in-vehicle sensing of position relative to the lane and other vehicles, the actuation of steering, engine, brakes, are performed. Most of the components used at the physical layer were adopted from development of sensor, computer vision, and actuator technologies.

In this project, our design and implementation will not consider managing the traffic flow, but only pay attention to a single platoon, thus the network layer and link layer are not our concern. Nor shall we focus on the lower layer physics and control, but in order to better understand the
vehicle behaviors in a platoon, the platooning control functionality is briefly discussed in Section 2.4.

2.4. Platooning Control Functionality
The main regulation layer functions are lateral and longitudinal control of vehicle motions, as well as maneuver coordination of vehicle behaviors in platooning. We discuss these functionality in more detail below.

2.4.1. Lateral Control
The primary functionality of lateral control is keeping the vehicle following the lane center, typically requires solving a closed-loop regulator problem to steer the vehicle. Lateral control is in addition concerned with lane changing, when the vehicle needs to switch from following the center of one lane to following the center of the next lane.

2.4.1.a. Lane Tracking
Lane tracking is the functionality that keeps a vehicle centered in a lane. It is a regulator control problem, in which the primary design considerations involve the fundamental trade-off between tracking accuracy and ride quality. Higher tracking accuracy enhances driving safety and makes it possible to save roadway construction cost by reducing the lane width, but it can also lead to harsher, jerkier lateral vehicle motions. Challenges to lane tracking include difficult vehicle dynamics, and particularly non-linear speed-dependent response characteristics.

For steering a vehicle towards the center of a lane, it is important to know the current lateral position of the vehicle. Several technologies have been considered to measure this position, such as permanent-magnet tracking, computer vision and Differential Global Positioning System (DGPS), an enhancement of GPS that provides a higher accurate location service.

2.4.1.b. Lane Changing
Lane changing is the functionality to steer a vehicle from the current lane to an adjacent lane. Changing lanes is considerably more challenging than simply following a lane, since it involves more complicated vehicle dynamic behaviour as well as the change of front target and coordination between vehicles.

In order to support this task, a magnet tracking system can be used. Magnet-based lane changing can be done either by following a trail of magnets between the lanes at fixed locations, similar to railroad switch that guides a train to an adjacent track (“infrastructure guided”); or by dead reckoning between the magnet trails in the center of the two lanes (“free lane change”), which involves using a lateral displacement estimator, based on acceleration and yaw rate measurements, combined with trajectory planning. And recently, the magnet infrastructure requirements is generally removed due to the progress of computer vision in recognizing the lanes.

2.4.2. Longitude Control
Longitudinal control of a road vehicle means controlling its speed and its separation from the preceding vehicle, requiring activation of engine, brake and possibly transmission actuators on the vehicle. Analogous to lateral control, the primary requirements are tracking accuracy and ride comfort, which must be traded off against each other. Ride comfort is important for driver and passenger acceptance but is also closely related to the ability of the control system to save fuel and to reduce emissions. While tracking accuracy is important for maintaining safety and for enabling vehicles to operate in close proximity to each other so that they can increase lane capacity and reduce aerodynamic drag (thereby also saving fuel and emissions).

Longitudinal control implementations were typically based on use of radar sensors to measure vehicle-to-vehicle distance, but some systems have also used computer vision and more specialized image-processing sensor systems. The differences among systems based on these alternative sensors, however, are not as significant as the differences between the autonomous
vehicle following systems that rely only on their own sensors and that rely on cooperative control. In a cooperative vehicle following system, the controllers typically augment sensor data with wireless communication in order to actively collect state information (position, speed, acceleration and maneuver information) of other vehicles. These types of controllers are not only able to maintain string stability under a constant time gap, like autonomous controllers, but also under constant distance gaps. As these constant distance gaps are independent of the operating speed of the vehicles, the cooperative controllers are capable of keeping the headway small, even at high speeds. This is one of the reasons why cooperative controllers, as used in CACC systems, can achieve a much higher road capacity.

2.4.3. Maneuver Coordination

Besides longitudinal and lateral control, vehicle platooning involves the coordination of vehicle maneuvers. These maneuvers are typically the formation and splitting of platoons, the merging of traffic streams and coordination of changing lanes. This research mainly concerned with the joining and leaving a platoon. In the following sections we limit our description to these maneuvers.

2.4.3.a. Joining Platoon

Joining is the term used when a vehicle (or more) is combined in a platoon. This situation can be easily extended to the situation where two platoons are combined into a single one in general. Figure 5 shows an example where vehicle 4 accelerates and joins at the back of a platoon (i.e., vehicles 1 to 3). We note that this is not the only way of joining a platoon: a vehicle may also join the platoon in the middle or in front of a platoon.

2.4.3.b. Leaving Platoon

A vehicle or a number of vehicles may leave the platoon and need not be bound by the rules for maintaining the platooning configuration. An example of this situation is illustrated in Figure 6 where the last vehicle (i.e. vehicle 4) leaves the platoon. We also note that a vehicle in any position of the platoon may leave.

By coordinating the maneuvers among the involved vehicles, an improvement of the efficiency could be enabled. Such automatic coordination depends heavily on wireless communication. An introduction to wireless communication in vehicular environments is provided in the next section.
2.5. Control Models

2.5.1. Vehicle Longitude Dynamics

The following equation is widely employed to model the nonlinear longitudinal dynamics

\begin{align*}
\dot{x}_i(t) &= v_i(t) \\
\dot{v}_i(t) &= \frac{\eta_{T,i}}{r_i} - c_i^A v_i^2 - m_i g f \\
\tau_i \dot{T}_i(t) + T_i(t) &= T_{i\text{des}}(t)
\end{align*}

in which \(x_i(t)\), \(v_i(t)\) denote the (longitude) position and velocity of vehicle \(i\), respectively. \(m_i\) is the vehicle mass, and \(r_i\) is its tire radius. \(\eta_{T,i}\) denotes the mechanical efficiency of driveline, \(c_i^A\) denotes its aerodynamic drag coefficient, \(g\) is the gravitational acceleration, \(f\) is the rolling resistance coefficient; \(\tau_i\) is the inertial delay of vehicle longitudinal dynamics; \(T_i(t)\) denotes the actual driving/braking torque, while the desired quantity is \(T_{i\text{des}}(t)\). The position and velocity of the leading vehicle are denoted by \(x_0(t)\) and \(v_0(t)\), respectively.

This nonlinear model is close to the real vehicle dynamics and some studies directly use it for platooning control. However, explicit performance limits are rather difficult to analyze with given spacing policy and communication topology. Rather, linear models are more frequently used for tractable issues.

As the simplest case, the single integrator model which takes the vehicle speed as the input variable to be controlled, i.e., \(\dot{x}_i(t) = u_i(t)\). While it significantly simplifies the theoretical analysis on controller design, however, the single integrator is unrealistic since controlling of the speed introduces a lot of instability. An improvement of that is to use the double integrator, i.e.,

\begin{align*}
\dot{x}_i(t) &= v_i(t) \\
\dot{v}_i(t) &= a_i(t) \\
\dot{a}_i(t) &= \frac{1}{\eta_i} a_i + \frac{1}{\eta_i} u_i
\end{align*}

where \(\eta_i\) is the internal actuator dynamics parameter.
We can write it in the state space form:

\[
\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t)
\]

\[
x_i(t) = \begin{bmatrix} x_i \\ v_i \\ a_i \end{bmatrix}, \quad A_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\eta_i} \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

Alternatively, the corresponding transfer function can be written as

\[
X_i(s) = \frac{1}{s} Y_i(s), \quad Y_i(s) = \frac{1}{s} Z_i(s)
\]

\[
Z_i(s) = H_i(s) U_i(s)
\]

in which \(X_i, Y_i, Z_i\) are the Laplace transforms of vehicle \(i\)'s position, velocity, and acceleration, respectively; \(U_i\) is the control input, and \(H_i\) is a linear Single-Input-Single-Output (SISO) transfer function which is strictly proper. Clearly, this model has two integrators and a lower order inertial delay, which leads to some fundamental limitations for certain platoons.

In vehicle platooning, all vehicles in a platoon have the same objective to follow a leading vehicle with a certain distance, which is the acceptable/comfortable inter-vehicle distance determined by the spacing policy. There are two major spacing categories that adopted by existing car-following models [16]. The first category has a fixed desired inter-vehicle distance [44], while the second one is the so-called “velocity dependent spacing” category, which determines the inter-vehicle distance based on vehicle velocity. The latter is more commonly used, in particular, for a vehicle \(i\) in the platoon, the velocity dependent spacing policy relates the desired inter-vehicle distance with the time headway as

\[
d_{r,i} = r_i + h_{d,i} v_i
\]

where \(d_{r,i}\) is the desired inter-vehicle distance between vehicle \(i\) and the one in front of it, \(r_i\) is the standstill distance, \(h_{d,i}\) is the headway-time constant, representing the time that it will take the vehicle to arrive at the same position as its predecessor when \(r_i = 0\), and finally, \(v_i\) is the velocity of vehicle \(i\). The actual distance \(d_i\) between two consecutive vehicles is then given by

\[
d_i = x_{i-1} - (x_i + L_i),
\]

where \(x_i\) is the position of vehicle \(i\), and \(L_i\) is its length. The local control objective, which is referred to as vehicle following, can now be defined as regulating the spacing error to zero, where the error is defined as

\[
e_i = d_i - d_{r,i}.
\]

The control of the collective behavior of a platoon is based on every vehicles’ mutual awareness of their states (e.g., inter-vehicle distance, vehicle speed and acceleration, as we mentioned above), which is achieved by inter-vehicle sensing and communication. The information provided by inter-vehicle sensing and communication serves as an important input to the controllers in each vehicle, thus having a significant impact on the platooning behavior. In literature, the connectivity of information exchange between vehicles is captured by the so-called information flow topology, and affects the platoon performance such as string stability, stability margin, and coherence behavior. These performance measures can often be used as objectives or constraints to form the control as an optimization problem.

There are plenty of information flow topologies studied by researchers. Early-stage platoon control is mainly based on radar-sensing, and a vehicle can only obtain information about its nearest neighbors, i.e., immediately preceding and following vehicles. Typical topology types include the Predecessor Following (PF) and Bi-Directional (BD) topologies, as shown in Figure 7 (a) and (c), respectively. Nowadays, with V2V communication via technologies such as IEEE 802.11p-based DSRC and the emerging 5G solutions, various topologies become feasible since a
vehicle can communicate with vehicles beyond its immediate surroundings. More types of topology have been derived, such as the Predecessor-Following Leader (PFL), Bi-Directional Leader (BDL), Two Predecessor-Following (TPF), and Two Predecessor-Following Leader (TPFL). See Figure 7. Since information flow topology has a significant impact on the behavior of a platoon, it is important to adapt the topology (e.g., by controlling the transmission power of communications) based on the need of platoon control.

The most commonly used topology is predecessor following (PF) where the following vehicle only receives communication signal from its predecessor. As an example, Figure 8 shows a block diagram of a vehicle in a CACC platoon under this type, in which the controller can be seen as two parts: feedback ACC ($C_{i,ACC}$) and feed-forward CACC ($C_{i,CACC}$). Both the CACC and ACC inputs can be seen as acceleration signals that act as an input to the vehicle dynamics. The conventional ACC controller is usually a simple proportional derivative (PD) or proportional integral derivative (PID) controller. The controller is designed to process the wireless communication signal. A feed-forward filter is used to process the acceleration data that are transmitted from the preceding vehicle to obtain the control input. Under the predecessor-leader following topology, the controller takes two error signals (i.e., the spacing error with its predecessor and the spacing error with the lead vehicle) into account since the subject vehicle has communication with both its predecessor and lead vehicle in the platoon. PID controller is also applied in the controller design. Bidirectional topology allows the subject vehicle communicate with its adjacent vehicles, and the controller uses the front and rear spacing error as the feedback signals.

The environment for platooning and autonomous driving technologies always has uncertainties. The control systems need to have the ability to deal with stochastic disturbance and risks. The stochasticity can have various sources, such as the drivers or the system itself. [15] proposed a reinforcement learning approach for the cooperative adaptive cruise control. The authors modeled the environment as a Markov Decision Process (MDP), and incorporated the stochastic game theory into the system to improve the CACC performance. Their simulation results show that small disturbances are damped through the platoon. And in [63], the uncertainties in the communication network and sensor information were studied. The uncertainty parameters and delays were approximated to be Gaussian distributed, and this model was used to calculate the minimal time headway for safety.
2.6. Vehicular Wireless Communication

There are several technologies that can support the wireless vehicle-to-vehicle (V2V) and vehicle-
to-infrastructure (V2I) communication such as IEEE 802.11 (WiFi), IEEE 802.16 (WiMAX), and
Dedicated Short Range Communications (DSRC). The maturity of these technologies are one of
the keys for enabling vehicle platooning. The wireless communication between vehicles that get
involved in platooning forms a vehicular ad hoc network (VANET), which is our primary focus in
this project.

In VANETs, the IEEE 802.11p standard is emerging as a popular standard for such applications.
The IEEE 802.11p standard is an amendment to the IEEE 802.11 standard which was proposed
for Wireless Access in Vehicular Environments (WAVE). The IEEE 802.11p is licensed from 5.850
GHz to 5.925 GHz (the so called “5.9 GHz band”) in the United States [18]. The standard is
basically designed to enable communication in mobile environments e.g. V2V and V2I communi-
cation. The IEEE 802.11p physical layer is similar to that of the IEEE 802.11a standard. However,
to cope with the high-speed vehicles, the bandwidth of the IEEE 802.11p is reduced to the half of
the IEEE 802.11. Its Medium Access Control (MAC) protocol is inspired by the MAC of the IEEE
802.11e standard with some modifications to make it more suitable for mobility. Similar to the IEEE
802.11e, the IEEE 802.11p uses the Enhanced Distributed Channel Access (EDCA) mechanism.
The EDCA mechanism allows the high-priority traffic to have a higher chance of acquiring the
channel access and being transmitted compared to the low-priority traffic [2].

Besides the IEEE 802.11p, WAVE also contains the standard of IEEE 1609, which is the upper
layer standard. IEEE 1609 completes the WAVE by its sub-detail standards, for instance, IEEE
1609.2 standard is responsible for the communication security; IEEE 1609.3 standard covers
the WAVE connection setup and management. IEEE 1609.4 standard that is based on the IEEE
802.11p Physical (PHY) layer and Medium Access Control (MAC) layer supplies operation of high-
level layers across multiple channels. As shown in Table [1] within IEEE 802.11, DSRC is known
as IEEE 802.11p, which amend the IEEE 802.11 on MAC and PHY layers [33].
3. Routing Protocols in VANET

3.1. Classification of VANET Routing Protocols

Routing is the process of selecting a path for traffic in a network, or between or across multiple networks. Routing is usually performed by a dedicated device called a router. As we mentioned in Chapter 1, due to the high mobile and dynamic nature of VANETs, the reliability and efficiency of communication between vehicles critically relies on routing protocol, and various types of routing protocols have been proposed by researchers. They can be characterized into different classifications, as in [19], [40], [56], [16], etc. Here we adopt a widely acknowledged one, as show in Figure 9 in which the VANET routing protocols are divided into five categories.

![Fig. 9: Classification of Routing Protocols](image)

3.1.1. Position Based Protocols

Position based protocols require the Global Positioning System (GPS) assistance to obtain the location of the vehicle as well as that of its neighbors. A source communicates with the destination by using geographical positions as well as its network address, and does not need any routing table. Therefore, it is suitable for highly dynamic environment. However, the satellite signal could become weak when the vehicle gets into the area such as tunnels. Examples of such protocols include the Greedy Perimeter Coordinator Routing (GPCR) [37], Greedy Perimeter Stateless Routing (GPSR) [27], Geographical Source Routing (GSR) [36], etc.

3.1.2. Broadcast Based Protocols

Broadcast is a frequently used routing method in VANETs, such as sharing traffic, weather, emergency, road condition among vehicles, and delivering advertisements and announcements. Broadcast is also used in unicast routing protocols (routing discovery phase) to find an efficient route to the destination. When the message needs to be disseminated to the vehicles beyond the transmission range, multi-hop is used.

The simplest way to implement a broadcast service is flooding in which each node re-broadcasts messages to all of its neighbors except the one it got this message from. Flooding guarantees the message will eventually reach all nodes in the network.

Broadcast based protocols use broadcasting mechanism, i.e., flood data packets to all the available nodes in the entire VANET. Different relay selection techniques are used to reduce the message overhead. Examples of broadcast base protocols include the BROADCOMM Routing [32], Distributed Vehicular Broadcast Routing (DV-CAST) [58], Position Aware Reliable Broadcasting Protocol (POCA) [17], Density Aware Reliable Broadcasting Protocol (DECA) [38], Urban Multi-Hop Broadcast protocol (UMB) [30], etc.
3.1.3. Cluster Based Protocols
Cluster based protocols divide the network into clusters so that each cluster consists of vehicles with same characteristics such as the speed and direction. If a vehicle node needs to communicate with a node within the cluster then the data will follow a direct path as it is considered to be a local communication. However, if the vehicle node wants to communicate a node which is outside the cluster, it must use its cluster head for reaching the destination. Clustering for Open Inter Vehicular Communication Network (COIN) [9] is the prime example of this type of protocols.

3.1.4. Geo-Cast Routing
Geocast refers to the delivery of packets to a group of destinations in a network identified by their geographical locations. In geo-cast routing protocols, the primary objective of geo-cast routing is to deliver the packet from a source node to all other nodes with a specified geographical region, which is defined as the Zone of Relevance (ZOR). Most geo-cast routing methods are implemented as a multicast service within the ZOR. However, to override the simple flooding of the geo-cast message from the source to the ZOR, a forwarding area called Zone of Forwarding (ZOF) is used to restrict the flooding inside it. Examples of geo-cast routing include GeoGRID [35], Inter-Vehicle Geo-cast Routing (IVG) [5], RObust VEhicular Routing (ROVER) [29], Distributed Robust Geo-cast Routing (DRG) [26], etc.

3.1.5. Topology Based Protocols
Topology based routing protocols use links information which stored in the routing table that exists in the VANET to forward data packet from source to destination. The topology based routing is usually further categorized into three types, the proactive or table-driven protocols, the reactive or on-demand protocols, and hybrid protocols. Figure 10 illustrates this classification and provides several common routing protocol examples for each category.

![Diagram of Ad-hoc Routing Protocols](image)

**Proactive Routing Protocols**
In proactive routing protocols, each node continuously calculates and update the routes to every other node in the network. Routing information is periodically transmitted throughout the network for each node to maintain the routing table consistency, thus the proactive scheme is also called table-driven routing. When needed, a proactive protocol immediately provides the required routes to all reachable nodes in the network, with the minimal initial delay for transmission. However, for highly dynamic network topology, the proactive schemes require a significant amount of signaling traffic and power consumption to keep routing information up-to-date and reliable. Examples of certain proactive routing protocols are Destination-Sequenced Distance Vector (DSDV) [43], Optimized Link State Routing (OLSR) [13], etc.

**Reactive Routing Protocols**
In reactive routing scheme, the protocol attempts to discover route only “on-demand”. For this purpose, a node initiates a route discovery process through the network. This process is completed
once a route is determined or all possible permutations have been examined. Once a route has been established, it is maintained by a route maintenance phase until either the destination becomes inaccessible along every path from the source or until the route is no longer desired. In Reactive schemes, nodes maintain the routes to active destinations. A route search is needed for every unknown destination. This type of protocols reduce signaling traffic overhead thus save bandwidth and power consumption, but at the cost of increased latency when no route to the destination is available. Common examples of reactive routing protocols are Dynamic Source Routing (DSR) [25], Ad-Hoc On-Demand Distance Vector (AODV) [42], Temporarily Ordered Routing Algorithm (TORA) [39], Dynamic Manet On Demand (DYMO) [57], etc.

**Hybrid Protocols**

Hybrid protocols are composites of reactive and proactive routing, in the hope of taking advantage of both schemes. The main idea behind such protocols is to initiate route-discovery on demand but at a limited search cost. Common protocols based on hybrid approach includes Zone Routing Protocol (ZRP) [21], Fisheye State Routing (FSR) [41], etc.

### 3.2. Review of Common Routing Protocols in VANETs

#### 3.2.1. DSR

**Dynamic Source Routing (DSR)** is a reactive routing protocol based on source routing, which means that the source node explicitly lists the complete path to the destination in the header of each packet it sends. Therefore, each host in the network needs to determine the best route to others. DSR consists of two phases: route discovery and route maintenance. In route discovery phase, the source node broadcasts to all its neighbors within a wireless transmission range a Route Request (RReq) packet, which contains: request ID (in order to eliminate duplicate), target destination's address, and the route record. Upon receiving RReq, the destination node generates a Route Reply (RRep) packet that includes a list of addresses received in RReq and transmits it back through the same path to the source. **Route maintenance** is the mechanism for detecting any broken route, which is required in DSR due to lack of periodic routing updates. When a node determines a problem at a hop, it will send Route Error packet which contains the addresses of both ends of the problem hop to the source. The source then can try to use any other known route to the destination in cache, or it can discover again to find a new path. All nodes get involved would need to update their route cache and send broadcast packets for discovering new routes, thus DSR could cause the so-called “broadcast storm” problem when the network topology changes.

There are several ways of optimization can be done in DSR to reduce the amount of overhead packets. For example, RRep can be piggybacked on another RReq packet sent to the targeted source. Another optimization is to allow intermediate node that receives RReq to respond with RRep if it knows a route to destination, furthermore, each node will delay for a time period proportional to its distance to destination before replying from its cache, to avoid the congestion caused by many intermediate nodes replies. Also, exponential backoff can be used in network partitions, so that the rate of initiating route discoveries can be limited and the “broadcast storm” can be reduced.

#### 3.2.2. DSDV

**Destination-Sequenced Distance Vector (DSDV)** is an extension of distance vector routing, which relies on the distributed Bellman-Ford algorithm to find the minimum-cost spanning tree. DSDV is a proactive table-driven routing scheme for VANET, its basic idea is that the destination nodes rather than intermediate ones provide the route freshness indication. In order to solve the famous distance vector problem, i.e., “count-to-infinity”, sequence number is added in each route entry. Each node preserves a routing table which stores a route to every possible destination in the network as well as the distance (i.e., the number of hops) to the destination. Each such entry in the table is marked with a sequence number assigned by the destination node.

When a node moves to a new location, it discovers its new neighbors, and then sends new
distance vector with higher sequence number which indicates the freshness. The route with the most recent sequence number is always used, and among those with the same sequence numbers, the route with fewest number of hops is used. Eventually, the new routing information can be propagated to other nodes and flooded to the entire VANET. In DSDV, any updates in the routing tables are periodically broadcast in the network to maintain table consistency. The amount of traffic generated by these updates can be huge, especially at high mobility.

3.2.3. TORA

Temporary Ordered Routing Algorithm (TORA) is a loop-free and highly adaptive distributed routing algorithm based on the concept of link reversal routing. In TORA, when a node needs a route to a destination, it broadcasts a query packet which contains the address of the destination. The query packet is propagated until it reaches the destination or an intermediate node with a route to the destination. The corresponding destination or intermediate node then broadcasts an update packet listing its height to destination. As the update packet is propagated through the network, each node receives it needs to update its height. In this way, each node constructs the direct acyclic graph (DAG). In TORA, when an invalid route is discovered, the node will look at all its neighbors to find a neighbor with the smallest height, and transmit an update packet. And if a node detects a network partition, it will generate a clear packet, reset the routing state, and removes invalid routes. The advantage of TORA is that it allows a route to all presented node in the network and the disadvantage is that the route maintenance is difficult in VANET. Whenever there is a change in the network, TORA broadcasts the update packets, which will result in high routing overhead.

3.2.4. AODV

Ad-Hoc On-Demand Distance Vector (AODV) uses a modified version of broadcast route discovery mechanism from DSR. Instead of source routing, however, AODV relies on dynamically establishing route table entries at intermediate nodes. Like in DSDV, sequence number is included when updating routes. But each node maintains a monotonically increasing sequence number counter to supersede stale cached routes, which is different from DSDV. Thus AODV is often described as a combination of DSR and DSDV. And it is claimed to be the most commonly used routing protocol in VANET [19].

In AODV, when a source node tries to send data to the destination node, it starts the route discovery process by broadcasting Route Request (RReq) to all its neighbors. Upon receiving RReq, if the neighbor nodes lack information pertaining to the request, they forward the packets to their neighbors. This process continues until RReq reaches either the destination node or the node who knows the path to destination, which will then send back to the source node a Route Reply RRep packet containing the number of hops to destination as well as its most recently seen sequence number from destination. Every node that forwards RReq will cache a reversed route for itself back to the source, and every node that forwards RRep will create a forward route to destination. In this way, each node only needs to specify the next hop, rather than the entire route as in DSR. Note that the RRep packets are sent through the reversed path. And when a link is down, the upstream nodes will receive unsolicited RRep containing an infinite metric for the destination, the source node then needs to rediscover a new route.

3.2.5. OLSR

Optimized Link State Routing (OLSR) is a protocol that is based on traditional link state algorithm in which every node maintains the network topology information by exchanging link state information with others periodically. The advantage of this protocol is that it applies the MPR (Multipoint Relays) strategy to reduce the size of control information as well as the amount of rebroadcasting nodes.

In OLSR, each node periodically broadcasts its HELLO messages containing information about its neighbors with bidirectional link as well as other known neighbors. The exchange of these HELLO messages between (1-hop) neighbors permits each node to learn the knowledge of its
neighbors up to 2 hops. Each node then maintains a neighbor table, which records the information about its 1-hop neighbors as well as the link status (which could be unidirectional, bidirectional, or MPR), and a list of 2-hop neighbors that these 1-hop neighbors give access to. Each entry in the neighbor table has an associated holding time, upon expiration of which that entry is no longer valid and thus removed. The neighbor table also has a sequence number, which is incremented every time a node selects or updates its MPRs. We will illustrate the procedure of MPR selection in Section 3.3.

TC messages are sent periodically by each node \( N \) to declare its MPR Selector set, i.e., a list of neighbors who have selected the sender node \( N \) as a MPR. A sequence number is also associated with the TC message. A node with an empty MPR Selector set (i.e., none of its 1-hop neighbors has selected it as an MPR) may not generate TC messages. In OLSR, only MPRs are allowed to generate TC messages, and only MPRs are responsible for retransmission of TC messages.

The information diffused in the network by TC messages helps each node to build its topology table. Each node maintains a topology table, and upon receiving a TC message, the following procedure may be executed:

- if there exists an entry of the topology table has the same destination with TC:
  - if the topology table has higher sequence number, the TC message will be ignored;
  - if the TC message has higher sequence number, then the corresponding entry in the topology table will be replaced by a new entry recorded from the TC message;
  - if the topology table and the TC message have the same sequence number, then the holding time of that entry will be refreshed;
- otherwise (i.e., all entries in topology table have different destination from TC), a new entry is recorded in the topology table.

With the neighbor table and the topology table, each node in the network would calculate and maintain a routing table, which determines the route to other destinations. The entries in the routing table consist of the destination address, next hop address, as well as the distance (i.e., number of hops) to destination. If any change occurs in either the neighbor table or the topology table, the routing table is re-calculated to update the route information about each known destination in the network.

3.3. Multipoint Relays (MPRs)

The idea of multipoint relays (MPRs) is to reduce the flooding of packet by reducing the duplicate retransmission of broadcast packets. Unlike in regular flooding where each node is responsible to retransmit a packet it receives, only selected MPRs are allowed for retransmission. Therefore, the route is a sequence of hops through the MPRs from source to destination. As in Figure 11b, the blue circles represents MPRs, and only them are allowed to retransmit the packet originated from the center node.

Each node \( N \) in the network selects and maintains its own MPRs from its 1-hop neighborhood, the MPRs are selected such that it covers all symmetric 2-hop neighbors of \( N \). Those 1-hop neighbors of \( N \) which are not selected as its MPR receives and processes messages from \( N \), but are not allowed to retransmit broadcast messages gathered from \( N \).

The MPRs of \( N \), denoted as \( MPR(N) \), is then an arbitrary division of the symmetric 1-hop neighborhood of \( N \) which comply with the following aspect: every node in the symmetric strict 2-hop neighbors of \( N \) must have a symmetric link towards \( MPR(N) \). The tinier the MPRs, the less control traffic overhead outcome from the routing protocol. Respective node also takes care of information about the neighbors that have selected it as an MPR. This set is called the MPR Selector set. A node obtains the information from recurring HELLO messages received from its 1-hop neighbors. The MPRs change over time as the topology of network changes (i.e., when a node selects another MPR set) and will soon be indicated by the selector nodes in their HELLO
3.4. Literature Review on Analysis of VANET Routing Protocols

There are many comparative analysis of VANET routing protocols in the literature, the compared protocols and the simulation environment vary a lot, and the results and conclusions are quite different. Here we only list a few of recent studies, mainly on some of the topology based VANET routing protocols we mentioned above.

[22] analyzed the performance of OLSR, AODV and DSDV protocols in a VANET network and examined the impact of varying mobility, density and pause time on the functionality of these protocols. They demonstrated that AODV have better performance in terms of throughput and packets delivery rate, whereas OLSR have best performance in terms of packet delivery.

In [1], the authors evaluated AODV, DSR and OLSR performance in urban scenarios under varying metrics such as node mobility, vehicle density, and with varying traffic rates. The results showed that clustering effects created by cars aggregating at intersections have remarkable impacts on evaluation and performance metrics, and OLSR performs best in most of the simulated traffic situations. [12] compared DSDV, OLSR and AODV using NS-3, and established that among the routing protocol used, OLSR proves to be the best in terms of highest packet delivery rate and lowest packet loss rate for every node, also has the least overhead.

[51] also compared DSDV, OLSR and AODV using NS-3, but under several different environment settings, they showed that the DSDV and OLSR have better performance than AODV for low-
density and low-speed scenarios, and as the network density or nodes’ velocity increases, the OLSR protocol performs better than the other two.

[52] came up with different conclusion, in which the performance of AODV, DSDV and OLSR were compared and evaluated in a safety application scenario using NS-3, and the simulation results show that DSDV could adapt to the requirement of safety application consistently, while OLSR and AODV failed to achieve the expected performance.

AODV has been reported to perform better than DSDV and OLSR in VANETs more often. For example, again using NS-3 and SUMO (Simulation of Urban Mobility), [20] claimed that AODV gives best results in high mobility environment, after comparing with OLSR and DSDV. [65] drew the same conclusion after comparing with DSR. [50] came out with the conclusion that AODV is far better than OLSR as it gives very good result with high speed vehicles. Similar conclusion was drawn in [45] by using NS-2. Another performance evaluation of topology based routing protocol (DSDV, AODV and LSRG) was done in [31] and reported AODV as the best.

Other routing protocols have also been claimed to perform better. In [24], the authors explored 8 different routing protocols, including proactive (DSDV, OLSR, FSR), reactive (AODV, DYMO, DSR), hybrid (ZRP) and position-based protocols (GPSR), by using NS-2 with vehicle mobility under urban traffic generated from VanetMobiSim, and concluded that the geographic routing protocols perform better in VANET since the position information can be useful. And [54] concluded that hybrid protocols always work well in ad-hoc environment and specially with high speed nodes, after the comparative study between GPSR and ZRP.

4. Design
In this project, sensor and control packets will be transmitted over an emulation of the wireless vehicular ad-hoc network (VANET) that uses an efficient ad-hoc routing protocols. The emulation of the mobile VANET is implemented over a fixed network. The distributed tux machines at Auburn University are used to simulate the experiments. Wireless nodes are executed on different computers, and communicate with each other through the network.

4.1. Vehicle Platooning Application
A simple road train application program will be implemented to simulate the control for a lead truck and multiple following vehicles, which will enable them to participate in automatic platoon by providing the following functions.

- Form a platoon behind a lead truck. A vehicle may initiate forming a platoon with a lead truck where the lead truck will determine the road train parameters, e.g. road train speed and direction. Once the platoon is formed, the road train control application will ensure that the following vehicle and the lead truck will travel at the correct speed to maintain the platoon configuration, i.e. the distance of the gap between the following vehicle and lead truck is within a certain range (e.g., between 10 to 20 meters). The actual speed of each vehicle in a road train may be different.
- Join an existing platoon. A vehicle that receives vehicle sensor information from other vehicles in the road train may join the road train. Once it joins the road train, it will maintain the road train configuration as above, i.e. it must maintain a spacing from the front and back vehicle of a certain distance (e.g., between 10 to 20 meters).
- Leave a platoon. A vehicle may leave a road train and need not be bound by the rules for maintaining the road train configuration.

4.1.1. Packet Propagation Loss Model

4.2. Vehicle Behavior and Packet Categories

4.2.1. Vehicle Information Packet Broadcasting
Every 10 milliseconds, each vehicle will broadcast a packet which containing its own driving information. The datagrams of the vehicle information packets contain the following:
• GPS position
• Velocity
• Acceleration
• Brake control
• Gas throttle

Upon receiving the vehicle information packets from other vehicles, each vehicle would check if the incoming packet is the newest one it has received from the source vehicle (by checking the sequence number). If that is the case, it would update its own knowledge of the source vehicle if necessary. Also, by checking its MPR selector table, it will know whether to forward/retransmit the received packets with the newest sequence number to all its neighbors, aka every other vehicle in its transmission range (i.e., 100 m), except the one from which it receives the packet (i.e., the previous hop) or the source. The retransmission will be performed if the receiver is selected as an MPR of the previous hop.

4.2.2. Platooning Behavior and Related Packets

The CACC system needs to be simple, reliable and trustworthy, its primary objective is to maintain a certain gap in the road train. Each following vehicle could join or leave the road train leading by the lead truck by sending JOIN or LEAVE requests. The design of the joining and leaving procedures is one of the key building blocks in our project.

The lead truck sends its driving information to the following vehicles so that the following vehicles in the road train can duplicate its behavior and stay in the road train. The following vehicles send their driving information to the lead truck, so that the lead truck can keep track of their positions, and allow other vehicles to join or leave the road train. The sequence number and the IP address are included in the packet header, and in addition to those NORMAL packets for exchanging the vehicle information, we further define five different types of special packets for controlling the platooning behavior, they are JOIN, LEAVE, ACK_JOIN, NOTIFY, and OK.

The following vehicles can send JOIN or LEAVE request to the lead truck, and the lead truck should reply an ACK for that request. The JOIN request can only be sent when a following vehicle is not in the road train, i.e., its vehicle state is FREE. After it sends a JOIN, the vehicle state becomes FORM, which indicates that this vehicle is waiting for the ACK_JOIN message from the lead vehicle. To avoid collision while merging into the middle of the platoon, it is only allowed to join the road train when the lead truck makes sure other vehicles have prepared enough space for it to merge and sends it an ACK_JOIN message. After it receives ACK_JOIN and replies with ACK_JOIN, the vehicle state becomes FOLLOW, which indicates that this vehicle has joined the road train, and its driving is controlled by following another vehicle in the road train. It is only allowed to make LEAVE request when it is in the road train, i.e. the vehicle state is FOLLOW. This is like a state machine (See Figure 12). Note that in our design, the ACK_JOIN messages sent by the lead truck and by the following vehicle have totally different functionality, the details are described in Section 4.2.3. While the connection established, the lead and following vehicles communicate their driving information via NORMAL packets.

4.2.3. Join the Platoon

The specific procedure for the following vehicles to join the road train is as follows:

- Initially, only the lead truck stays in the road train. The following vehicles could send request to join the road train at any time if it is not yet in.
- The lead truck then checks the information of the following vehicle that contains in the packet it receives, and determines whether to accept or decline the request of joining the road train, based on if the following vehicle is at the same direction. Once the lead truck decides the following vehicle could have the chance to join the road train, it would then determine which position in the road train the requesting vehicle should join, that is, put the requesting vehicle into the road train right after the nearest vehicle ahead of it, based on the vehicles’ current
GPS information. If no vehicle in the road train is ahead of it, then it should slow down and follow the lead truck.

- If the requesting vehicle were asked to join to the end of the road train, it will then receive an ACK_JOIN message from the lead truck, which tells it which specific vehicle it should follow. If the lead truck determines that the requesting vehicle to join in the middle of the road train, it will inform other vehicles to leave enough space for the joining vehicle first. Only after the vehicle it informed replied an ACK message when enough space has been maintained and thus it is safe to join, the lead truck then sends an ACK_JOIN to the requesting vehicle.

- Once the following vehicle which initiated a JOIN request receives an ACK_JOIN from the lead truck, it would then accelerate to catch up the platoon, or slows down for waiting the platoon, until reaches a desired location so that it keeps an appropriate distance from the vehicle in front of it in the road train, it would then moves to the right lane of the road, set the speed and acceleration in order to follow the vehicle in front of it, upon the information received from that vehicle most recently. Remark that to simplify the model, we assume the acceleration/de-acceleration to the desired speed for catching up or slow down is instantaneous, so is the braking and moving to the right lane and joining the platoon.

- After joining the road train, the requesting vehicle sends an ACK_JOIN to the lead truck. The lead truck then update its own knowledge about the road train topology, and could be able to process the next JOIN or LEAVE request.

The above procedures are depicted in Figure 13 and Figure 14 which illustrate different situations whether the following vehicle is about to join to the end or middle of the road train, respectively.
4.2.4. Leave the Platoon

For a following vehicle to leave the road train, the procedure is much simpler. The following vehicle $k$ sends a \texttt{LEAVE} message to the lead truck, without the need of receiving any response, it then moves to the left lane, and is no longer be controlled by the road train. To increase the reliability of the communication, we let the leaving vehicle sends \texttt{LEAVE} message for three times. Once the lead truck receives the \texttt{LEAVE} packet, it needs to inform the vehicle ($k+1$) in the road train which was following the left vehicle $k$ to follow the one previously was ahead the left one, (i.e., $k+1$ should follow $k-1$ instead of $k$ after $k$ leaves), by sending a \texttt{NOTIFY} message to vehicle $k+1$.

This information can be calculated from the road train topology table cached in the lead truck, and upon receiving the \texttt{LEAVE} message, the lead truck need to update its knowledge about the road train topology. The informed vehicle, as well as all other vehicles behind it, then speed up a little and fill up the space. Note that the only information that the lead truck notifies the informed vehicle is which vehicle in the road train it should follow, and the informed vehicle changes its speed based on its own knowledge about other vehicles’ driving information from corresponding \texttt{NORMAL} packets it received.

4.3. Packet Forwarding Protocols

In order for a vehicle to learn about the position, speed and acceleration of road train and other cars, each vehicle and the lead truck must transmit its sensor data containing its GPS position, speed and acceleration at a constant period of 10 to 100 milliseconds. These sensor data are then forwarded multi-hop. In this section, we will design two schemes of broadcasting, one uses
Fig. 14: Procedure for a following vehicle joining the middle of the road train

MPRs while the other one does not. Both algorithms aims at reliably sending information packets to other vehicles. The network broadcast protocols will be built on top of the simple UDP (User Datagram Protocol) socket interface that emulates the wireless link layer. The structure is shown as in Figure 15. Note that the wireless link layer is not part of this project, but represents protocols that could conceptually be added or substituted.

4.4. Reliable Broadcast Algorithm

We first design a reliable broadcast algorithm (RBA) on top of the link layer. Initially, each vehicle will generate a vehicle sensor packet depending on its dynamics and forwards it to all its neighbor nodes through its RBA layer. Eventually the packet will be forwarded through multiple hops and reach all other vehicle destination addresses. The RBA layer of the intermediate nodes performs the task of forwarding packets reliably from one node to another. The packet header must contain at least the following fields:

1) Sequence number
2) Source address
3) Previous hop (address)
Each vehicle node will run the program for the RBA as follows. In the RBA layer, each vehicle maintains a cache table of the broadcast packets that it has received from every other source node in the network, with the largest sequence number of the packets that it has received from the source up to that time point, i.e. the cache table contains information on broadcast packets that the node has forwarded previously. Now for a node with address $addr_i$, consider the case it receives a packet from another node with address $addr_{prev}$, which can be examined from the “Previous Hop” field specified in the packet header, it will first check the source address $addr_s$ of the received packet. If the packet source is itself ($addr_s = addr_i$), the packet will be discarded; otherwise it then compares the sequence number $sn_p$ of the packet with the largest sequence number $sn_s$ it cached for source $addr_s$ in its cache table. If there is no $sn_s$ record in the cache table, or the sequence number in the cache table is smaller than that of the received packet, i.e. $sn_s < sn_p$, which means this packet is the newest from source $addr_s$ for node $addr_i$, and needs to be forwarded. The node with $addr_i$ will then update its cache table, in which the sequence number from source $addr_s$ is replaced by $sn_p$, and the broadcast number $bn$ for that packet is set to 1. In addition, it changes the “PrevHop” field of the packet to its own address $addr_i$, and forwards a copy of the packet to each of its neighbors except the one from which it received this packet, i.e., $addr_{prev}$. Eventually the packet is sent to all the reachable wireless nodes.

To ensure the reliability of the broadcast protocol, we also design a re-broadcasting mechanism for increasing the redundancy, in case the previous broadcast packet was lost. With re-broadcasting, if the sequence number in the cache table is the same as that of the received packet, i.e. $sn_s \geq sn_p$, the cache table will also be updated, by adding one to the broadcast number of that packet. And that packet may be forwarded depending on the probability determined by the broadcast number, which in fact records the number of times the same packet has been re-broadcasted. If the packet has been broadcast once, there is a 50% probability of re-broadcasting it to its neighbors; if it has been broadcasted twice, then the probability is 25%, and so on.

4.5. Broadcasting with MPRs

Alternatively, a broadcast protocol using MPRs on top of the link layer will be designed and implemented. In this protocol, each vehicle continuing broadcasts Hello messages which contain...
its own one hop neighbors as well as their link status every 20 milliseconds, and we assume that any other vehicles could receive the packets only when their distance is within 100 m. The link status of each one hop neighbor could be either uni-directional, bi-directional, or MPR.

Upon receiving HELLO messages from others, each vehicle is able to learn the knowledge of its neighbors up to two hops. After sensing the neighbors, the HELLO message receiver then could have enough information to select its MPRs from its bi-directional linked one hop neighbors, by knowing all its two hop neighbors (as well as their access through nodes). The HELLO message receiver then could update all the neighborhood information in its neighbor table, which contains its one hop neighbors, the status of the link with these neighbors, and a list of two hop neighbors that these one hop neighbors give access to, as well as a sequence number which specifies the most recent MPR set that the local node keeping this neighbor table has selected. Every time a node selects or updates its MPR set, this sequence number is incremented to a higher value. Moreover, on reception of HELLO messages, each vehicle is able to know is this one hop neighbor has selected it as an MPR or not. Based on that, it can construct a MPR selector table, so that it will know whether to retransmit the received packets (other than HELLO messages).

Every broadcast message coming from these MPR Selectors of a node is assumed to be retransmitted by that node. This set can change over time, which is indicated by the selector nodes in their HELLO messages.

4.5.1. Neighbor Sensing and MPR Selection

After receiving HELLO messages, the receiver vehicle (let us denote it as $R$) might need to update its neighborhood information. First of all, we need to check whether the sender (we denoted it as $S$) was $R$’s two-hop neighbors previously, it needs to be removed if it is the case. In any case, $S$ should then be added into $R$’s one-hop neighbors if it was not there, and the link will be set as uni-directional if $R$ has not been declared as $S$’s one hop neighbor in this HELLO message. On the other hand, in the HELLO message, if $R$ is already $S$’s one hop neighbor, then the link for $S$ will be set as bi-directional.

Next, we need to update $R$’s two-hop neighbors information. Note that each of $R$’s two-hop neighbor ($M$) will maintain a set to record through which $R$’s one-hop neighbor, $M$ could get access through and reach $R$. Also note that the received HELLO contains the link status of all $S$’s one-hop neighbors, upon which $R$’s MPR selector table could be updated. Moreover, if $S$ was already a bi-directional one-hop neighbor for $R$, then there might exist some $M$ which use $S$ as an access through point to get to $R$. But if now the bi-link between $S$ and $M$ was unlinked (as specified in the received HELLO), in that case, we need to remove $S$ from $N$’s access through set, and we need to traverse the two-hop neighbors and do this for all possible $M$’s. If any such $M$ has empty access through set after removing (i.e., $S$ was the sole access through point for $M$ to reach $R$), it will also be cleaned from $R$’s two-hop neighbors.

We also need to traverse for each of $S$’s one hop neighbor $N$, which could be a candidate for $R$’s two hop neighbor. If $N$ is already a two hop neighbor of $R$, we need to update $N$’s access through set by adding source if necessary. Otherwise, $N$’s access through set will be newly constructed and contains only $S$, with that information, $N$ is then added as a new two-hop neighbor for $R$.

If there is any change, $R$’s MPRs need to be selected again. We use heuristics for MPR selection. First we sort the two hop neighbors by the number of one-hop neighbors (called access-through set $A$, in which the element is denoted as $A$) they connected with (note that the connection must be bidirectional). Then we allow the one with the fewest links with the access-throughs ($A$’s) chooses its MPR first, by randomly choose an element in $A$; Then the one with second-fewest $A$’s; and so on, until all two-hop neighbors are traversed.

Lastly, we update the one-hop neighbors by labeling their link status. If a one-hop neighbor has been selected as an MPR, we label its link status as MPR. However, if a one-hop neighbor was an MPR previously, but has not been selected this time, we need to degrade its link status as bi-directional.

A typical neighbor table would be like in Table II, which is obtain from our experiment for 10
nodes in the VANET.

### 4.5.2. OLSR Protocol for Special Packets

Those 7 types of special packets for joining and leaving the road train are the only packets use the optimized linked state routing (OLSR) protocol, since they are the only packets that specify the destination. The OLSR protocol is based on MPR selection, which we have already illustrated. The MPRs will send topology control (TC) messages to all its one hop neighbors every 30 milliseconds, in which only its own MPR selector table is declared. The sequence numbers of TC just use the sequence number of the neighbor table in that vehicle, so even new TC messages are generated, if no neighborhood information is changed, the sequence number would not increase.

The TC messages are used for every vehicles in the network to update their topology tables. Each vehicle maintains its own topology table, which has an entry for each of the other vehicles it knows in the network. Each entry contains some information about the initiator of the TC messages, which includes the address, MPR selectors, the largest sequence number of received TC messages from the initiator, and a holding time. Upon receiving an TC message, the corresponding entry is updated only when the TC message is larger than the one saved in the topology table. And if the two sequence numbers are equal, the holding time of that entry will be refreshed.

The topology table is used for updating the routing table, from which the vehicle could calculate which the next hop is for sending or forwarding those 7 types of special packets. The calculation of next hop requires keeping track of a reversed path from the destination through its MPRs, to the MPRs’ MPRs if necessary, until one of the vehicle’s one hop neighbor. Bread First Search is used for the calculation.

By doing above procedures, R’s neighborhood information as well as MPR selector table have been updated. If there is any change, the neighbor table sequence number will be increased, and the routing table will also be updated. The neighborhood information will be used for sending and forwarding packets, and also would be written into new HELLO messages. The MPR selector table would be transmitted to other vehicles if it is an MPR.

### 4.6. Network Configuration

The status of each packet sending and receiving would be recorded into a configuration file. A configuration file helps us to look at the link state between the vehicles, also the status of each sent or received packet. The vehicles write information into the configuration file, but do not track and utilize information from it. The configuration file looks like as in Table III.

### 4.7. Simulation set-up

To simulate different number of vehicles in a road train, the program needs to run on different tux machines. As there is no physical access to tux machines, remote login is needed. Multiple remote login windows may be necessary. The lead truck should always be instantiated first, and

| Node 4’s One-Hop Neighbors |
|---------------------------|
| **Node ID** | **Status** |
| --- | --- |
| 3 | MPR |
| 5 | MPR |
| 6 | BI |
| 7 | BI |
| 8 | BI |
| 9 | BI |

| Node 4’s Two-Hop Neighbors |
|---------------------------|
| **Node ID** | **Access Through (MPR)** |
| --- | --- |
| 1 | 3 |
| 2 | 3 |
| 10 | 5 |

**TABLE II: Exemplary Neighbor Table**
when the program starts to run, the lead truck will first erase the configuration file if that exists, and then write its own information into the file. The configuration file helps to track the existing nodes in the simulation environment, as well as their link states. Only after the lead vehicle were added into the environment, other vehicles, aka following vehicles, could enter the simulation. Each of the vehicle would have a unique node ID, starting from 1 for the lead truck, and incrementally generated for every vehicles when they were added. Each vehicle continues broadcasting HELLO messages which can be received by all other vehicles within a certain transmission range (i.e., 100 meters) with a loss rate. Upon that, each vehicle could maintain a neighbor table and select MPRs. Each vehicle continues sending packets containing information about its status to all its one hop neighbors, among which only MPRs will retransmit the received vehicle information packets. In this way, all other vehicles are able to sense the information by receiving the packets if they are within the transmission range of any vehicle. Following vehicles can make JOIN and LEAVE request depending on if they are in the road train, through optimized linked state routing (OLSR) protocol. Note that all packet are sent and received among vehicles by using use UDP protocol.

5. Implementation

5.1. Vehicle Model

A vehicle class is established to describe the attributes and behavior information of a vehicle, while the behavior of the lead truck and other following vehicles are quite different, we have two different subclasses to distinguish them. As they also have many functionality in common, Vehicle will be implemented as an abstract class, from which two different classes LeadingTruck and FollowingVehicle are extended. The utility methods are written separately in a handler class for vehicle parameters, includes keeping track of their GPS, controlling of their appropriate driving behavior, performing I/O operations with the configuration file, etc.

5.2. Socket Programming

For networking communication, we directly use the UDP in java.net. To write and retrieve information in the packets, a network handler class is written. Since a vehicle object contains all information that we want to send, we exploit the Object Output Stream to write the vehicle object into a Byte Array Output Stream. After flushing into a packet, all information is contained as a byte array in the packet to be sent. A benefit for doing that is while receiving the packet and retrieving the information, we can simply reverse the above procedure through Byte Array Input Stream and Object Input Stream, so that the dissembled information is a vehicle instance, and the received vehicle can easily track the information and do operations between two vehicles. Moreover, HELLO packets are sent using broadcast scheme, while other packets are sent to either one specific or all of the one hop neighbors via point-to-point communication. Note that we also implemented the without-broadcast version, and that could work on a single local machine. Broadcast only works on distributed machines, e.g., tux machines.

5.3. Multithreading

Multiple threads are run for each of the vehicle for the parallel jobs that needs to be done:

- Each vehicle maintains a broadcast thread for broadcasting the HELLO messages. In broadcasting scheme, every vehicle could receive every packets, however, as a simulation, many
of them should be discarded, if the packets are from the vehicle out of the 100 meters transmission range. Moreover, the loss rate is assumed to be proportional to the distance between the vehicles that communicate. We implemented these functionality in the receiver end to discard packets.

- Each vehicle maintains a *sensor updating* thread for updating the vehicle’s information for itself (such as GPS, speed, acceleration, etc.), as well as initiating the regular packet every 10 ms, and sending the vehicle information packet to all its one hop neighbors.
- Each vehicle keeps a server socket and this thread listens for all the packets it could receives. Each type of packets has a different processing scheme as in separate methods.
  - All received *HELLO* messages need to be processed since they do not have sequence numbers, and upon the neighborhood information of the source, the neighborhood and topology information of the receiver might need to be updated. Moreover, no further retransmission of *HELLO* messages is required. Upon receiving other packets, sequence numbers will be compared with the corresponding one in the vehicle’s cache.
  - If it is a new *normal* or *TC* packet, the source vehicle information then might be utilized by the receiver, and the packet might be further forwarded if such a receiver is a selected MPR.
  - And other types of special packets will be utilized or forwarded depending on the destination information in the packet as well as the MPR selector table and *routing table* in the receiver. *Bread First Search* is used for the calculation of the next hop.

- Upon receiving each packet (after discarding those due to out of transmission range and because of loss rate), it may be forwarded. Two different approaches are implemented. One is start a specific thread to forward this particular packet to all its neighbors, and soon close this thread, upon every arrival of packet that should be forwarded. The other is the receiving thread adds every packet that should be forwarded into a queue, and a separate thread specializes in forwarding packets from the queue in FIFO manner. Note that before forwarding a packet, the header should be modified, so that the previous hop is changed to this vehicle’s address (or, simply the node ID, in our implementation).
- Each vehicle has a separate thread to do the I/O operations with the configuration file. Since the I/O operations are so expensive, we tried to minimize its work. We used concurrency locks for preventing the conflict. The read operation then could be done frequently, as we set to every 50 ms. However, we must reduce the frequency of the write operation on the configuration file, and we set it to every 5 seconds.
- Each following vehicle would have a keyboard listening thread to catch the *JOIN* and *LEAVE* request, so that our request could be made by the user at any time, so that our simulation is just like in the real world circumstances. The thread then pass the variable values changed by the event handler to the process, and the procedure routines start.

### 5.4. Simulation Settings for Vehicle Behavior

In the simulation application, we assume the lead truck is 10 meters long, and each following vehicle is either 5 or 10 meters long (5m if it is a car, 10m if it is a truck).

The program would run with one lead truck and up to 10 following vehicles. The wireless nodes are distributed in any area large enough (about 10,000 meters of highway) for the following vehicles and lead truck to travel over a 5 minutes simulation period. All the following vehicles and the lead truck travel in one direction in a highway, e.g. towards East, and that the highway consists of only two lanes in the same direction. The width of each lane is assumed to be 5 meters. The x-coordinate of the highway is from 0 to 10,000 meters. When the simulation starts, the lead truck is on the right lane with x-coordinate of 10 meter and all the (intended) following vehicles are on the left lane from x-coordinates of 0 to 300 meters in the 10,000 meter length of the highway. At no time can the following vehicles and the lead truck overlap each other. At the beginning of the 5 minutes simulation period, all the following vehicles will use the automated program to form a
road train behind the lead truck, and then maintain the road train formation with a pre-specified spacing, i.e. 10-20 meters between the lead truck and the first following vehicle and between any two following vehicle. They will maintain the road train by tracking the position, speed, acceleration and brake control of the lead truck.

The leading truck would try to keep move smoothly at the speed of 30 m/s, but with a small perturbation of accelerating between $-1 \text{ m/s}^2$ to $1 \text{ m/s}^2$. And it would always be the head of the road train.

**Reduced Speed Zone in Simulation**

At the x-coordinate of 4000 meters of the highway, all the vehicles reduce their speed to 20 meter/sec due to a reduced speed zone. All the following vehicles will receive the change of speed from the sensor data packet broadcasted by the lead vehicle and reduce their own speed to 20 meter/sec.

At the x-coordinate of 5000 meters of the highway, all the vehicles would resume their speed to 30 meter/sec (average) since they have passed reduced speed zone. All the following vehicles will receive the change of speed from the sensor data packet broadcasted by the lead vehicle and increase their own speed to 30 meter/sec (average).

At the final phase before the end of the 5 minutes simulation period, all the following vehicles must leave the road train, travel independently of the lead truck and move to the left lane.

**5.5. Start the Simulation**

A main method in the `Simulation` class starts the simulation. The user needs to specify the type of the vehicle to be either “leading” or “follow”, and initial x-coordinate and initial speed arguments are optional. A leading truck should always be instantiated before any following vehicle.

**6. Experimental Results**

The experiments are conducted on the distributed tux computers at the university, with the number of vehicles in the VANET to be 2, 4, 6, 8, and 10, respectively.

The results are reported based on the following 3 performance metrics:

- **End-to-End Latency**: The delay in delivering a packet to the destination which is inclusive of all kinds of delay, i.e. sum of arrival time minus sum of send time divided by total number of connections.
- **Average Throughput**: This is the measure of the rate at which data being successfully delivered over a communication channel, and is usually measured as the total size of received packets with the unit time.
- **Packet Loss Rate**: The ratio of data packets lost to those generated by the sources, i.e. \((\text{number of packets received}) / (\text{number of packets sent})\). Thus, in the results, the sequence number of packets were sent is traced, and the number of received packets is counted, finally their ratio \(PDR\) (Packet Delivery Ratio) is calculated, and \((1 - PDR)\) is the packet loss rate.

**6.1. Latency**

One critical criteria for VANET is the end-to-end latency, which should be acceptable so that vehicles can timely receive information and do corresponding actions. For instance, if a car ahead breaks unexpectedly and make an emergency stop needs to be made, every fraction of a second counts. Therefore, we want to see what the latency is under different scenarios. Unlike in previous two projects, using MPRs, the source vehicle which initiates a packet will never receive this packet again, since only MPRs retransmit to their one-hop neighbors except the sender. In this case, we intentionally let the follow vehicles retransmit some of the normal packets back to the leading truck if the packets are directly from the leading truck (i.e., both the source and previous hop of the received packet are the leading truck). In this way, the leading truck could track the round trip time of these packets, and calculate it as the average latency.
In Figure 4, we compared the latency by using MPR for retransmission we implemented in this project and using flooding we implemented in the previous project. We plotted the latency time with different number of vehicles in the road train. Intuitively, as the number of vehicles in the road train grows, the number of packets flooded in the network should grow, so that every node should handle more packets. As a result, latency should grow. As we can see in our figure, with the growing number of vehicles, the average latency in this network of both MPR and flooding grows. This result conforms our expectation. Remark that in our plot, the MPR latency is higher than the flooding in general. However, we would argue that the reason is that we implemented the flooding retransmission as broadcast, and used point-to-point socket connection for MPR retransmission. The broadcast thread would always be there, but every time we wish to generate a point-to-point connection, we need to establish a new thread and close it after packet is sent. In this way, we would say that the higher latency is due to opening new threads, it does not mean MPR retransmission approach performs worse. Moreover, we can see that the latency grows exponentially for flooding as the number of vehicles increases, while the latency grows sub-linearly for the MRP retransmission. We would expect MPR retransmission performs better than flooding as more numbers in a VANET.

6.2. Throughput

As shown in Figure 17, with growing number of vehicles in the road train, more packets will be sent and retransmitted, and the throughput in the network should grow. The figure we plot conforms to the trend that we expected. Moreover, the exponential growth pattern for the flooding can be
more clearly seen in Figure 17, while using MPR to re-transmit, the throughput is much smaller, even with frequent extra \texttt{HELLO} messages sent to neighbors periodically. Flooding consumes too much traffic for possible unnecessary retransmission, and using MPRs in VANET is much more efficient.

6.3. Packet Loss Rate

As with any wireless system, packet loss is always a constant concern. In our experiments we found that packets loss also increased with growing number of vehicles in the road train. The result is plotted in Figure 6. The two approaches have similar growth rate of the average loss rate as the number of vehicles increases. That illustrates that the loss rate is mainly due to the distance-based loss function we deployed in the settings, and has little concern with the network traffic. The stable difference between the two approaches might because that the neighborhood information of using MPR through \texttt{HELLO} messages is more accurate than flooding.

7. Conclusions

In this paper, we implemented a VANET with a leading truck and multiple following vehicles, and simulated the transmission with different distances between the vehicles. We improved the retransmission by choosing relay points, and that alleviate the flooding problem. We measured the performance with different number of vehicles in the network, and compared with the results we obtained using broadcast flooding. We conclude that with MPR, the communication in the VANET to form a road train is more efficient and reliable.
References

[1] Mohamed Ben Ahmed, AA Boudhir, M Bouhorma, and K Ben Ahmed. Performance study of various routing protocols in vanet case of study. International Journal of Future Generation Communication and Networking, 7(6):231–240, 2014.

[2] Irfan Al-Anbagi and Hussein T Mouftah. Wave 4 v2g: Wireless access in vehicular environments for vehicle-to-grid applications. Vehicular Communications, 3:31–42, 2016.

[3] Assad Alam, Bart Besselink, Valerio Turri, Jonas Martensson, and Karl H Johansson. Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency. IEEE Control Systems, 35(6):34–56, 2015.

[4] Muhammad Alam, Joaquim Ferreira, and José Fonseca. Intelligent transportation systems: Dependable vehicular communications for improved road safety, volume 52. Springer, 2016.

[5] Abdelmalik Bachir and Abderrahim Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. In The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring., volume 4, pages 2456–2460. IEEE, 2003.

[6] Carl Bergenhem, Erik Hedin, Daniel Skarin, et al. Vehicle-to-vehicle communication for a platooning system. Procedia-Social and Behavioral Sciences, 48(2012):1222–1233, 2012.

[7] Carl Bergenhem, Qihui Huang, Ahmed Benmimoun, and Tom Robinson. Challenges of platooning on public motorways. In 17th world congress on intelligent transport systems, pages 1–12, 2010.

[8] Carl Bergenhem, Steven Shladover, Erik Coelingh, Christoffer Englund, and Sadayuki Tsugawa. Overview of platooning systems. In Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012), 2012.

[9] Jeremy Blum, Azim Eskandarian, and Lance Hoffman. Mobility management in ivc networks. In IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683), pages 150–155. IEEE, 2003.

[10] Christophe Bonnet and Hans Fritz. Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing. Technical report, SAE Technical Paper, 2000.
[11] Fred Browand, John McMahon, and Charles Radovich. Fuel saving achieved in the field test of two tandem trucks. 2004.
[12] Tushar Singh Chouhan and Rajvardhan Somraj Deshmukh. Analysis of dsdv, olsr and aodv routing protocols in vanets scenario: Using ns3. In 2015 International Conference on Computational Intelligence and Communication Networks (ICICN), pages 85–89. IEEE, 2015.
[13] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.
[14] CE Delft. Marginal abatement cost curves for heavy duty vehicles. background report, 2012.
[15] Charles Desjardins and Brahim Chaib-Draa. Cooperative adaptive cruise control: A reinforcement learning approach. IEEE Transactions on intelligent transportation systems, 12(4):1248–1260, 2011.
[16] Kakan C Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrru Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cac). IEEE Transactions on Intelligent Transportation Systems, 17(2):491–509, 2016.
[17] Elena Fasolo, Roberto Furiniato, and Andrea Zanella. Smart broadcast algorithm for inter-vehicular communication. In Proceedings of the Wireless Personal Multimedia Communications Symposium (WPMC), 2005.
[18] Song Gao, Alvin Lim, and David Bevly. An empirical study of dsrc v2v performance in truck platooning scenarios. Digital Communications and Networks, 2(4):233–244, 2016.
[19] Muhammad Rizwan Ghor, Ali Safa Sadiq, and Abdul Ghani. Vanet routing protocols: Review, implementation and analysis. In Journal of Physics: Conference Series, volume 1049, page 012064. IOP Publishing, 2018.
[20] Parul Gupta and Yogesh Ghaba. Performance analysis of routing protocols in vehicular ad hoc networks for cbr applications over udp connections. International Journal of Engineering and Computer Science, 3(6):6418–6421, 2014.
[21] Zygmunt J Haas, Marc R Searcy, and Prince Samar. The zone routing protocol (zrp) for ad hoc networks. 2002.
[22] Barkouk Hamid and En-Naimi El Mokhtar. Performance analysis of the vehicular ad hoc networks (vanet) routing protocols aodv, dsdv and olsr. In 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), pages 1–6. IEEE, 2015.
[23] L. R. Houbert. A study on platoon formations and reliable communication in vehicle platoons. Master’s thesis, University of Twente, 2012.
[24] Zineb Squalli Houssaini, Imane Zaimi, Mohammed Ouammi, and Said El Alaoui Ouattik. Comparative study of routing protocols performance for vehicular ad-hoc networks. International Journal of Applied Engineering Research, 12(13):3867–3878, 2017.
[25] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Mobile computing, pages 153–181. Springer, 1996.
[26] Harshvardhan P Joshi, Mihail L Sichitiu, and Maria Kihl. Distributed robust geocast multicast routing for inter-vehicle communication. In Proceedings of WEIRD workshop on WiMax, wireless and mobility, volume 921, 2007.
[27] Brad Karp and Hsin-Lung-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, pages 243–254. ACM, 2000.
[28] Pooja Kavathekar and YangQuan Chen. Vehicle platooning: A brief survey and categorization. In ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pages 829–845. American Society of Mechanical Engineers, 2011.
[29] Maria Kihl, Mihail Sichitiu, Ted Ekeroth, and Michael Rozenberg. Reliable geographical multicast routing in vehicular ad-hoc networks. In International Conference on Wired/Wireless Internet Communications, pages 315–325. Springer, 2007.
[30] Gökhân Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, pages 76–85. ACM, 2004.
[31] Vimlesh Kumar, Anurag Singh Baghel, and Priyank Mishra. Performance evaluation of dsdv, aodv and lgmr protocol in ad-hoc networks. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pages 4261–4266. IEEE, 2016.
[32] Kevin C Lee, Uichin Lee, and Mario Gerla. Survey of routing protocols in vehicular ad hoc networks. In Advances in vehicular ad-hoc networks: Developments and challenges, pages 149–170. IGI Global, 2010.
[33] Bo Li, Mahdieh Sadat Mirhashemi, Xavier Laurent, and Jinzi Gao. Wireless access for vehicular environments. https://pdfs.semanticscholar.org/8702/4b92ac23dc6e1a4322f760b77e93941826ae.pdf, 2010.
[34] Shengbo Eben Li, Yang Zheng, Keqiang Li, Yujia Wu, J Karl Hedrick, Feng Gao, and Hongwei Zhang. Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities. IEEE Intelligent Transportation Systems Magazine, 9(3):46–58, 2017.
[35] Wen-Hwa Liao, Yu-Chee Tseng, Kuo-Lun Lo, and Jang-Ping Sheu. Geogrid: A geocasting protocol for mobile ad hoc networks based on grid. 1(2):23–32, 2000.
[36] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fussler, Dagmar Hermann, and Martin Mauve. A routing strategy for vehicular ad hoc networks in city environments. In IEEE IV2003 Intelligent Vehicles Symposium, Proceedings. (Cat. No. 03TH8683), pages 156–161. IEEE, 2003.
[37] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. Geographic routing in city scenarios. ACM SIGMOBILE mobile computing and communications review, 9(1):69–72, 2005.
[38] Nawut Na Nakorn and Kultida Rojviboonchai. Dea: density-aware reliable broadcasting in vehicular ad hoc networks. In ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pages 598–602. IEEE, 2010.
[39] Vincent Douglas Park and M Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In Proceedings of INFOCOM`97, volume 3, pages 1405–1413. IEEE, 1997.
[40] Bijan Paul, Md Ibrahim, Md Bikas, and Abu Naser. Vanet routing protocols: Pros and cons. arXiv preprint arXiv:1204.1201, 2012.
[41] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In 2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record, volume 1, pages 70–74. IEEE, 2000.

[42] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[43] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (ddsv) for mobile computers. In ACM SIGCOMM computer communication review, volume 24, pages 234–244. ACM, 1994.

[44] Andréa A Peters, Richard H Middleton, and Oliver Mason. Leader tracking in homogeneous vehicle platoons with broadcast delays. Automation, 50(1):64–74, 2014.

[45] Vongpasith Phouthone and Wang Dong. Simulation based and analysis of routing protocols for vanet using vnetmobisim and ns-2. Int. J. Comput. Eng. Technol, 6:32–41, 2015.

[46] Rajesh Rajamani, Han-Shue Tan, Boon Kail Law, and Wei-Bin Zhang. Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. IEEE Transactions on Control Systems Technology, 8(4):695–708, 2000.

[47] Dr. Sunil Soni Ravi Shrimali. Improved olsr protocol for vanet. International Journal of Advance Engineering and Research Development, 4, 2017.

[48] Tom Robinson, Eric Chan, and Erik Coelingh. Operating platoons on public motorways: An introduction to the sartre platooning programme. In 17th world congress on intelligent transport systems, volume 1, page 12, 2010.

[49] SAE Sae levels of driving automation. https://www.sae.org/misc/pdfs/automated_driving.pdf. Accessed: 2016-11-20.

[50] Gamal Sallam and Ashraf Mahmood. Performance evaluation of olsr and aodv in vnet cloud computing using fading model with sumo and ns3. In 2015 International Conference on Cloud Computing (ICCC), pages 1–5. IEEE, 2015.

[51] Eduardo El Akkari Sallum, Guilherme dos Santos, Mario Alves, and Max Mauro Santos. Performance analysis and comparison of the dsdv, aodv and olsr routing protocols under vanets. In 2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST), pages 1–7. IEEE, 2018.

[52] Ganis Zulfa Santos and Moonsoo Kang. Performance analysis of aodv, dsdv and olsr in a vanets safety application scenario. In 2012 14th International Conference on Advanced Communication Technology (ICACT), pages 57–60. IEEE, 2012.

[53] Arif Sari, Onder Onursal, and Murat Akkaya. Review of the security issues in vehicular ad hoc networks (vanet). International Journal of Communications, Network and System Sciences, 8(13):552, 2015.

[54] Aji Setiabudi, Amalia Ayu Pratiwi, Doan Perdana, Riri Fitri Sari, et al. Performance comparison of gprs and zrp routing protocols in vanet environment. In 2016 IEEE Region 10 Symposium (TENSYMP), pages 42–47. IEEE, 2016.

[55] Steven E Shladover. Automated vehicles for highway operations (automated highway systems). Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 219(1):53–75, 2005.

[56] Faizan Ali Siddiqui, Liu Qingling, Zhang Chaozhu, and Saifullah Adnan. Critical review of topology based routing protocols in vanets.

[57] Christoph Sommer and Falko Dressler. The dymo routing protocol in vanet scenarios. In 2007 IEEE 66th Vehicular Technology Conference, pages 16–20. IEEE, 2007.

[58] Özcan K Tonguz, Nawaporn Wiatpongphan, and Fan Bai. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. IEEE Wireless Communications, 17(2):47–57, 2010.

[59] Sadayuki Tsugawa. An overview on an automated truck platoon within the energy its project. IFAC Proceedings Volumes, 46(21):41–46, 2013.

[60] Sadayuki Tsugawa, Shin Kato, and Keiji Aoki. An automated truck platoon for energy saving. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4109–4114. IEEE, 2011.

[61] Ardalan Vahidi and Azim Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. IEEE transactions on intelligent transportation systems, 4(3):143–153, 2003.

[62] Bart Vani Arem, Cornelle JG Van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. IEEE Transactions on Intelligent Transportation Systems, 7(4):429–436, 2006.

[63] Willem H van Willigen, Martijn C Schut, and Leon JHM Kester. Approximating safe spacing policies for adaptive cruise control strategies. In 2011 IEEE Vehicular Networking Conference (VNC), pages 9–16. IEEE, 2011.

[64] Qing Xu and Raja Sengupta. Simulation, analysis, and comparison of acc and cacc in highway merging control. In Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, pages 237–242. IEEE, 2003.

[65] Junhu Zhang and Zhenhua Sun. Assessing multi-hop performance of reactive routing protocols in wireless sensor networks. In 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), pages 444–449. IEEE, 2016.