

Vehicular Networks for Collision Avoidance at Intersections

Reza Azimi, Gaurav Bhatia and Ragnathan (Raj) Rajkumar

Carnegie Mellon University

Priyantha Mudalige

General Motors Research

Copyright © 2011 SAE International

ABSTRACT

A substantial fraction of automotive collisions occur at intersections. Statistics collected by the Federal Highway Administration (FHWA) show that more than 2.8 million intersection-related crashes occur in the United States each year, with such crashes constituting more than 44 percent of all reported crashes [12]. In addition, there is a desire to increase throughput at intersections by reducing the delay introduced by stop signs and traffic signals. In the future, when dealing with autonomous vehicles, some form of co-operative driving is also necessary at intersections to address safety and throughput concerns.

In this paper, we investigate the use of vehicle-to-vehicle (V2V) communications to enable the navigation of traffic intersections, to mitigate collision risks, and to increase intersection throughput significantly. Specifically, we design a vehicular network protocol that integrates with mobile wireless radio communication standards such as Dedicated Short Range Communications (DSRC) and Wireless Access in a Vehicular Environment (WAVE). This protocol relies primarily on using V2V communications, GPS and other automotive sensors to safely navigate intersections and also to enable autonomous vehicle control. Vehicles use DSRC/WAVE wireless media to periodically broadcast their position information along with the driving intentions as they approach intersections. We used the hybrid simulator called GrooveNet [1, 2] in order to study different driving scenarios at intersections using simulated vehicles interacting with each other. Our simulation results indicate that very reasonable improvements in safe throughput are possible across many practical traffic scenarios.

INTRODUCTION

Current human driver-based intersections which are managed by stop signs and traffic lights are not entirely safe, based on Federal Highway Administration (FHWA) statistics [12]. Our goal is to design new methods to manage intersections, which lead to higher fewer collisions and less travel delay for vehicles crossing at intersections. Various driverless vehicles have been developed and tested at intersections, such as in the DARPA Urban Challenge [3] and General Motor's EN-V, which has been recently unveiled in Shanghai, China [4]. Our focus is to use vehicle-to-vehicle (V2V) communication as a part of co-operative driving in the context of autonomous vehicles to manage intersection traffic efficiently and safely.

Past work in this domain includes the use of Vehicle to Infrastructure (V2I) communication by having a centralized system in which all vehicles approaching an intersection communicate with the intersection manager. The intersection manager is the computational infrastructure installed at intersections and to make reservations for each approaching vehicle and manages all vehicles crossing the intersection [5, 6, 7]. Installing centralized infrastructure at every intersection is not very practical due to prohibitively high total system costs. In this work, we advocate the use of Vehicle-to-Vehicle (V2V) communications and a distributed intersection algorithm that runs in each vehicle. Our focus in this paper is on (a) designing new protocols for V2V based-intersection management, (b) extending an advanced mobility simulator for vehicles, and (c) comparing our protocols to the operational efficiency of conventional intersections with stop signs and traffic lights.

The rest of this paper is organized as follows. Section 2 introduces the collision-detection algorithm used in our proposed intersection protocols. Section 3 contains intersection protocols used to manage various intersection scenarios. This section consists of a stop-sign model, a traffic light model and three V2V-based protocols: V2V Stop-Sign Protocol (SSP), Throughput-Enhancement Protocol (TEP) and Throughput-Enhancement Protocol with Agreement (TEPA). In Section 4, we describe the implementation of our protocols using

the GrooveNet hybrid simulator, with new mobility and trip models. Section 5 contains the evaluation of our intersection protocols. Section 6 presents our concluding remarks.

COLLISION DETECTION AT INTERSECTIONS

We first identify the conditions required for two or more vehicles to collide at an intersection.

Suppose *Arrival-Time* is the time at which a vehicle arrives at an entrance of the intersection and *Exit-Time* is the time at which the vehicle exits the intersection area. If a vehicle enters an intersection while another vehicle is in the intersection area, their (*Arrival-Time*, *Exit-Time*) intervals must overlap. Two vehicles being inside the same intersection at the same time is a necessary, but not sufficient condition for a collision. In Figure 1(a), two vehicles are within the intersection at the same time but not occupying the same space. Figure 1(b) shows a scenario in which a vehicle is coming from the south and turning right while the other vehicle is coming from the north and also turning to its right. In this case, both vehicles can cross the intersection at the same time without a collision.

A collision occurs if the following conditions are all true:

1. **Same Intersection:** vehicles are at the same intersection.
2. **Time Conflict:** vehicles have overlapping (*Arrival-Time*, *Exit-Time*) intervals.
3. **Space Conflict:** vehicles occupy the same space while crossing the intersection.

If any of the above three conditions is false, then there will be no collision and vehicles can safely continue along their trajectory.

We refer to the part of the road that a vehicle is currently on as its **current road segment (CRS)**, and the part of the road that the vehicle will be moving to after the current road segment as the **next road segment (NRS)**. In the context of an intersection, CRS corresponds to the road segment that a vehicle is on before the intersection, and NRS represents the road segment that the vehicle will be on after crossing the intersection.

We assume in this paper that vehicles stay in the same lane after passing any intersection and do not switch lanes.

Our **Collision Detection Algorithm for Intersections (CDAI)** will be run on each vehicle that crosses a transaction, with information exchanged among vehicles approaching, crossing and leaving the intersection. The algorithm uses path prediction to determine any space conflicts with other vehicles trying to cross the intersection. Each lane on the road is considered to be a polygon, which starts from the previous intersection and ends at the next approaching intersection. Then, CDAI predicts the space (or region) which will be occupied by the vehicle during its trajectory. Utilizing the CRS (current road segment), current lane, and NRS (next road segment) information for each vehicle, CDAI predicts the path taken by the vehicle to cross the intersection and generates two polygons: the first polygon is related to the vehicle's CRS and current lane, and the second polygon is related to the vehicle's NRS. Each polygon's height is the length of the road between two consecutive intersections and the polygon's width is the lane width. So, for each vehicle, these two polygons together form the complete spatial region related to its path, which we refer to as its **Trajectory Box (TB)**. As illustrated in Figure 2, a collision can potentially occur if two vehicles approaching the same intersection have intersecting "TBs".

To find out if the TBs of two vehicles intersect, we use the *Separating Axis Theorem* [8, 9]. The Separating Axis Theorem states that, for a pair of convex polygons that are not in a state of collision, there exists an axis perpendicular to an edge of one of the polygons that has no overlap between the projected vertices of the two polygons. This theorem can be simplified for our purposes since we are only dealing with two-dimensional rotated rectangles. Therefore, each polygon is tested against the four axes of the other polygon and if all projections overlap, a collision is detected. An optimization on this theorem exists for two-dimensional rotated rectangles, wherein, the polygon-under-test is rotated and centered on the intersection of the *x*-axis and the *y*-axis, and hence projections need to occur for only 2 axes [13]. This solution works for any collision possibility, even for *cross-collisions* where a collision occurs between two polygons perpendicular to each other.

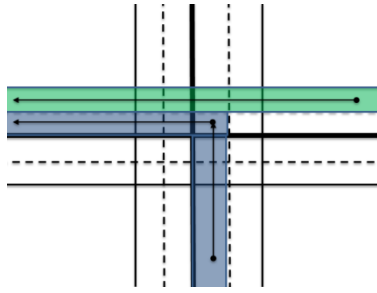


Figure 1(a)

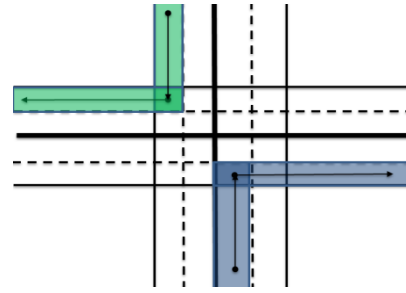


Figure 1(b)

Example scenarios in which no space conflict occurs at the intersection

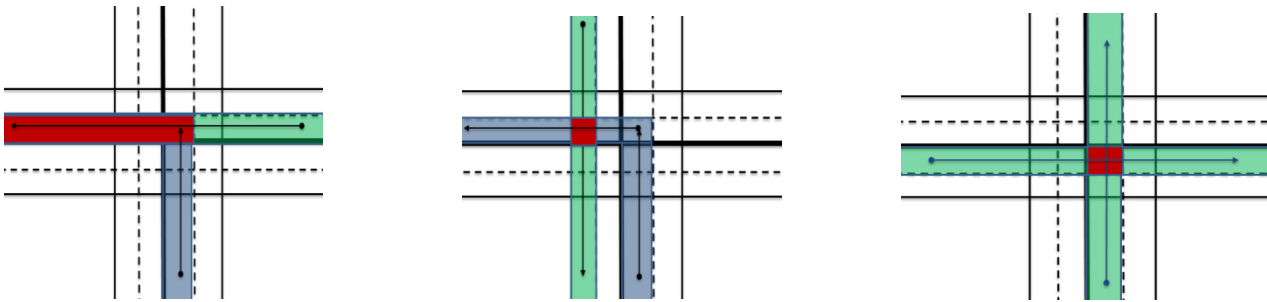


Figure 2: Three example scenarios of space conflict

If a potential collision is detected by CDAI, it uses a **priority-based policy** to assign priorities to vehicles so that an unambiguous and repeatable precedence order in which vehicles cross the intersection can be established. For prioritizing the movement of vehicles at the intersection, the “first come, first served” (FCFS) algorithm is used. Based on FCFS, the first car arriving at the intersection is the first one crossing and leaving the intersection. Even though FCFS is an efficient algorithm, it can lead to a deadlock situation in particular scenarios such as when several vehicles get to an intersection at the same time (or very close to each other). To avoid any deadlocks, three **tie-breaking policies** are applied for vehicles with the same arrival time in the following sequence:

1. Roads are categorized as *primary roads* and *secondary roads* based on the roadmap database information, and higher priority is assigned to vehicles arriving at an intersection using a primary road than vehicles arriving using a secondary road.
2. If vehicles arrive at an intersection at the same time and using the same type of (primary or secondary) road, priorities are assigned based on their trajectories and whether turns are required. Specifically, vehicles going straight have higher priority than vehicles turning right, with vehicles turning left getting the lowest priority.
3. If all the previous conditions still result in a tie among two or more vehicles, the *Vehicle ID* (VID) which is unique for each vehicle is used to break ties - the vehicle holding a higher VID is given higher priority to cross the intersection.

The CDAI decision is made within each vehicle based on the information communicated using V2V. The algorithm will alert a vehicle if it can cross the intersection safely or, if any collisions are predicted, the vehicle must stop.

INTERSECTION PROTOCOLS

In this section, we describe three protocols¹: the Stop-Sign Protocol (SSP), the Throughput-Enhancement Protocol (TEP) and Throughput-Enhancement Protocol with Agreement (TEAP). We will specify their functionality under various scenarios. The contents of messages communicated among the vehicles will be detailed in the next section.

STOP-SIGN PROTOCOL (SSP)

In this protocol, a stop sign is assumed to be present at every arrival point of an intersection. Let t_s be the minimum amount of time in seconds that a car must wait at an intersection before proceeding. When vehicles approach an intersection, they must obey the rules of a stop sign which is to wait t_s seconds even if there is no other vehicle around. The FCFS priority policy mentioned in the previous section is obeyed by each vehicle. Vehicles also use *STOP* and *CLEAR* safety messages at the intersection in order to inform other vehicles in range about their current situation and movement parameters. The following rules are applicable.

- **Sending STOP:** As a vehicle approaches an intersection, it transmits a *STOP* safety message. Any vehicles within range will receive that message. When the vehicle arrives at the intersection, it also comes to a complete stop for t_s seconds.
- **On Receiving STOP:** On receipt of a *STOP* message, a vehicle uses the CDAI scheme described earlier, except for the *Space Conflict* rule. If more than one car arrives at the same intersection and will be inside the intersection area for an overlapping interval of time, priorities will get assigned to them and the vehicle with the highest priority will cross the intersection after t_s seconds pass. Lower-priority vehicles will remain stopped even after t_s seconds, waiting to receive a *CLEAR* message.
- **Sending CLEAR:** When a vehicle crosses the intersection secondary and travels a distance defined by a threshold parameter D_{TC} , it broadcasts *CLEAR* messages indicating that the intersection is now safe to pass.
- **On Receiving CLEAR:** On receiving this message, the vehicle checks if it has stopped for at least t_s seconds and, if true, it then checks if the sender of the *CLEAR* message is the same as the sender of the *STOP* message. The FCFS, priority and tie-breaking rules are again applied. If t_s seconds have not passed as yet, the vehicle remains stopped while processing received messages to make a decision when the t_s seconds ends. If several vehicles are stopped at the intersection, by re-applying the priority policy, each vehicle decides if it should remain stopped or it can cross the intersection next as it has the highest priority among all stopped vehicles at the intersection.

THROUGHPUT ENHANCEMENT PROTOCOL (TEP)

This protocol is designed to manage intersection crossings by pure V2V communication without using any infrastructure such as stop-signs, traffic lights, sensors and cameras. The goal is to enhance the throughput at intersections without causing collisions. Vehicles again use *STOP* and *CLEAR* safety messages to interact with other vehicles. We define the throughput of an intersection based on the delay of all vehicles trying to cross the intersection. The following rules are applicable to each vehicle.

- **Sending STOP:** Every vehicle has access to its own GPS coordinates, speed and also to the map database; using these values, it computes the distance to the approaching intersection. The accuracy of this distance prediction is directly related to GPS accuracy. If the current distance of the host vehicle from the other vehicle is not greater than a threshold parameter D_{STOP} , then it starts sending periodic *STOP* messages (with the goal of informing other vehicles within range that it is getting close to the intersection). The *STOP* message will be sent with frequency f_{STOP} .
- **Sending CLEAR:** When the vehicle exits the intersection, it sends periodic *CLEAR* messages with frequency f_{CLEAR} until it travels further than a threshold value D_{CLEAR} from the exit point of the intersection. This behavior lets other vehicles know that the intersection is no longer in use by this vehicle.
- **On Receiving STOP:** On receiving a *STOP* message, the vehicle checks if all three collision conditions are satisfied. If even one of the conditions is not satisfied, then it means that the vehicle can cross the intersection without a collision with the

¹ Our protocols are inspired at least in part by Kurt Dresner's work [11]. Our focus is exclusively on V2V-based protocols, and our contributions include support for intersection management protocols in GrooveNet [1, 2], detailed evaluations and ongoing implementations in real vehicles.

sender of the STOP message. Otherwise, the vehicle acts based on the priority assigned to it using the priority policy. If it has lower priority than the sender of the STOP message, it comes to a complete stop at the intersection. Else, it has higher priority and ignores this message. In the latter case, the vehicle will have precedence at the intersection. Note that a vehicle which first started sending STOP messages may be superseded by a later vehicle due to priority considerations.

- **On Receiving CLEAR:** Each vehicle stores the information within received STOP messages which made it stop at the intersection. On receiving a CLEAR message, the vehicle checks if this message is sent from the sender of the last STOP message that has higher priority and because of which the vehicle is waiting at the intersection. This check is possible by just looking at the unique VID embedded in the message. If the VID of the CLEAR message is the same as the VID of the last processed STOP message, then the space that the vehicle needs to occupy for crossing the intersection is now clear.

Using TEP, vehicles stop at the intersection only if the collision detection algorithm predicts a collision and assigns a lower priority to them based on the messages it receives from all vehicles at the intersection. If no collision potential is detected or the highest priority is determined among contending vehicles, a vehicle can ignore other STOP messages, broadcast its own STOP messages to notify other vehicles, and cross the intersection safely. Multiple vehicles can be inside the intersection area at the same time if no space conflict occurs based on the collision detection policy's results. These rules increase the throughput of the intersection by decreasing the average delay time relative to the situation that vehicles should stop at the intersection. (We are currently studying enhancements to this protocol which will enable vehicles to slow down instead of coming to a complete stop when there are vehicles with higher priority entering the intersection. Evaluations of this scheme will be reported in the near future).

A reader might note that TEP implicitly assumes that V2V messages are not lost. While TEP will indeed work better with a very reliable wireless medium, the periodic transmission of STOP and CLEAR messages is targeted at a lossy communications medium and the protocol can tolerate *some* lost messages.

THROUGHPUT ENHANCEMENT PROTOCOL WITH AGREEMENT (TEPA)

This protocol is built on TEP and is explicitly designed to handle lost V2V messages. Additional CONFIRM and DENY messages are used to perform explicit handshaking between vehicles approaching the same intersection. Each vehicle makes its own local decision as in the previous protocols, but each vehicle announces its decision to cross the intersection by sending a CONFIRM or DENY message to either adhere to or override a decision made by another vehicle. On receiving a STOP message from another vehicle, the receiver will also send a message to acknowledge the reception of the message. The following rules are used by each vehicle *in addition to* the rules used by TEP:

- **Sending CONFIRM:** if no collision with the sender of a STOP message is predicted by CDAI, this message is sent first. It is also sent if a collision is predicted and a lower priority is assigned to the receiver of the STOP message. In this case, the receiver of the STOP message comes to a complete stop and waits for a CLEAR message.
- **Sending DENY:** If a collision is predicted and the receiver of the STOP message has a higher priority than its sender, the vehicle will send a DENY message to inform the sender of the STOP message that the latter's decision has been overridden and that this vehicle will *not* stop at the intersection.
- **On Receiving CONFIRM:** if the vehicle had sent a STOP message earlier, it has higher priority than the sender of the CONFIRM message and continues to proceed with its current decision.
- **On Receiving DENY:** if the vehicle had sent a STOP message later, it now has lower priority than the sender of the DENY message and must wait for a CLEAR message when it must re-evaluate the situation.

COLLISION DETECTION

The collision detection scheme used in our intersection protocols ensures that two vehicles will not occupy the same space at the same time while crossing the intersection. Essentially, if there is any trajectory conflict, then one of the cars will be assigned a higher priority based on the priority policy, and the other one will wait for a CLEAR message without entering the intersection area. This prevents any collision between vehicles crossing the intersection.

DISTANCE KEEPING

In order to ensure a safe distance between cars, a distance-keeping protocol known as the *Car-Following Model* is used. This model is designed to control the mobility of vehicles while moving towards and exiting the intersection. A message of type *Generic* is sent at a regular interval and contains information about a vehicle's position, current lane, as well as current and projected map DB locations. On receiving this message, each vehicle checks if it is on the same road segment and the same lane as the sender. If this is the case, then by comparing its current GPS position with the sender's position, the vehicle determines if the sender is in front or behind it. In case of being behind the sender's vehicle, the vehicle adjusts its current velocity to the speed of the vehicle in front to prevent any collision. The vehicle does not need to have the same speed as the leader vehicle unless the distance between them is less than a threshold D_{follow} . Otherwise, it can maintain its current velocity which is related to the road's speed limit.

INTERSECTION SAFETY MESSAGE TYPES

We now describe in detail the content of transmitted messages.

The STOP message contains 9 parameters:

- **Vehicle ID:** Each vehicle has a unique identification number.
- **Current Road Segment:** Identifies the current road that the vehicle is using to get to the intersection.
- **Current Lane:** Identifies the lane being used.
- **Next Road Segment:** The next road taken by the vehicle after crossing the intersection.
- **Next Vertex:** The next intersection that the vehicle is getting close to.
- **Arrival-Time:** The time at which the vehicle gets to the intersection.
- **Exit-Time:** The time at which the vehicle will exit the intersection.
- **Message Sequence Number:** A unique number for each message from a vehicle. This count gets incremented for each new message generated by the same vehicle. This helps a receiver since it only needs to process the last message received from a particular sender.
- **Message Type:** The type of the message which is STOP in this case.

The CLEAR message contains 3 parameters: **Vehicle ID**, **Message Sequence Number**, and **Message Type**: CLEAR.

The CONFIRM message contains 3 parameters: **Vehicle ID**, **Message Sequence Number**, and **Message Type**: CONFIRM.

The DENY message contains 3 parameters: **Vehicle ID**, **Message Sequence Number**, and **Message Type**: DENY.

IMPLEMENTATION

In this section, we describe the implementation of the V2V protocols and the messages described in the previous two sections. To implement and analyze intersection protocols, traffic at intersections needs to be simulated. For this purpose, we use a tool called GrooveNet [1, 2] built at Carnegie Mellon University. We first give a brief introduction to GrooveNet and describe the extensions made.

GROOVENET²

GrooveNet [1, 2] is a sophisticated hybrid vehicular network simulator that enables communication among simulated vehicles, real vehicles and among real and simulated vehicles. By modeling inter-vehicular communication within a real street map-based topography, GrooveNet facilitates protocol design and also in-vehicle deployment. GrooveNet's modular architecture incorporates multiple mobility models, trip models and message broadcast models over a variety of links and physical layer communication models. It is easy to run simulations of thousands of vehicles in any US city and to add new models for networking, security,

² GrooveNet is an acronym that stands for “Geographical Routing of Vehicular Networks”.

applications and vehicle interaction. GrooveNet supports multiple network interfaces, GPS and events triggered from the vehicle's on-board computer. Through simulation, message latencies and coverage under various traffic conditions can be studied.

New models can easily be added to GrooveNet without concern of conflicts with existing models as dependencies are resolved automatically. Three types of simulated nodes are supported: (i) vehicles which are capable of multi-hopping data over one or more DSRC channels, (ii) fixed infrastructure nodes and (iii) mobile gateways capable of vehicle-to-vehicle and vehicle-to-infrastructure communication. GrooveNet's map database is based on the US Census Bureau's TIGER/Line 2000+ database format [10]. Multiple message types such as GPS messages, which are broadcast periodically to inform neighbors of a vehicle's current position, are supported. On-road tests over 400 miles within GrooveNet have lent insight to market penetration required to make V2V practical in the real world [1].

Mobility Models

One major extension to GrooveNet that we made is the inclusion of lane information for roads. The TIGER map database has no information concerning the number of lanes along each road. We used the heuristic of adding lane information based on road-type information present in the database. GrooveNet has several mobility models, such as the Street Speed, Uniform Speed and Car-Following models. We have modified these models for our current purposes. In addition, we have also created new mobility models that support the presence of multiple lanes with vehicles now also having the ability to switch lanes. Cars can switch lanes either at randomly chosen times or using predefined starting lanes. Specifically, the new mobility models that were implemented are as follows:

1. **Stop-Sign Model:** When a simulated vehicle approaches an intersection managed by stop-signs at each entrance, it comes to a complete stop regardless of the situation of any other vehicle at the intersection. In other words, the velocity of the vehicle becomes zero even if there is no other car trying to cross the intersection. In discussions, police recommend 3 seconds of complete stopping even at an empty intersection. This stop delay will increase in proportion to the number of cars that arrived earlier at the intersection.
2. **Traffic-Light Model:** The traffic-light model follows the same basic logic as the stop sign model except that stop signs are now replaced by traffic lights. The *Green-Light Time* of the traffic light has a default value that can be changed by the user.
3. **V2V Stop-Sign Model:** This model represents the implementation of the Stop-Sign Protocol (SSP) described earlier. Each intersection in the map has a unique number which is called its *Vertex Number*. Based on the vertex number, each vehicle determines the next intersection it is approaching and also all the roads connecting at this intersection. The vehicle sends out a periodic safety message as described earlier. These messages are processed by other vehicles receiving them to know if multiple vehicles are approaching the same intersection. A priority-assignment policy decides which vehicle gets to cross the intersection first. In case of distinct arrival times, a first-come-first-served policy is used. In case of ties, tie-breaking rules are applied. Any vehicle with a lower priority comes to a stop at the intersection. The vehicle then checks if other vehicles have exited the intersection. Based on the V2V stop-sign protocol, if the vehicle should remain stopped, the velocity stays zero until its next update cycle, after which the tests are executed again. This continues until the vehicle gets the permission to cross the intersection and sets its velocity to the street speed limit.
4. **Throughput-Enhancement Model:** The Throughput-Enhancement Protocol (TEP) is implemented by this model. This model uses the complete collision detection algorithm (CDAI) including *Space Conflict*. Vehicles obey the car-following rules on the road before getting to the intersection such that their speed gets adjusted to the vehicle in front based on the information received within periodic Generic safety messages. As the vehicle arrives at the intersection, it follows the V2V-based intersection rules and uses CDAI to determine if it is safe to cross the intersection. All safety messages including STOP, CLEAR and GENERIC are sent with a frequency of 10 Hz. All safety messages utilize the same 10Hz V2V Basic Safety Message (BSM) formats defined by SAE J2735 Dedicated Short Range Communications (DSRC) Message Set Dictionary. Data elements in Part II of BSM are used to specify the type of the safety message and also encapsulate related elements defined in the previous section. The safe distance maintained between two contiguous vehicles is selected to be 10 m.
5. **Throughput Enhancement with Agreement Model:** This model is designed to use all five types of safety messages: STOP, CLEAR, CONFIRM, DENY and GENERIC. Each vehicle moves based on the car-following protocol before approaching an intersection as well as after exiting the intersection area. Vehicles follow the Throughput-Enhancement with Agreement Protocol (TEPA) rules to get to an agreement on the sequence that the vehicles at the intersection should cross and also inform each other about their decision.

EVALUATION

In this section, we present a detailed evaluation of the proposed protocols using the models added to GrooveNet. The evaluation is carried out under different types of traffic scenarios and using different kinds of intersections. We compare the different mobility models: the Stop-Sign Model, the Traffic-Light model and V2V-interaction models. Two instances of the Traffic Light model are used, one with a green light duration of 10 seconds and another with a duration of 30 seconds.

METRIC

We calculate the trip time for each simulated car under each model and compare that against the trip time taken by the car assuming that it stays at a constant street speed and does not stop at the intersection. The difference between these two trip times is considered to be the **trip delay** due to the intersection. We take the *average trip delays* across all cars in a simulation sequence as our metric of comparison.

The trip route for each car is calculated using the `DijkstraTripModel` in GrooveNet which calculates the shortest route between two points using Dijkstra's algorithm. The route is chosen with a waypoint at the intersection forcing the route to pass through the intersection. The logging mechanism in GrooveNet was modified to enable logging of start and end times of cars to measure their trip times.

SCENARIOS

Since there is a large variation in intersection types, we restrict our attention to the following three categories of intersections:

- **Four-way Perfect-Cross Intersections:** The intersection legs are at perfect right angles to the neighboring leg.
- **T-junction:** Two roads are perpendicular to each other, and one of the roads ends at the intersection.
- **Four-way Angled Intersections:** These intersections are four-way intersections where they do *not* intersect at a right angle.

We run all our simulations on 4-lane roads, with 2 lanes in each direction. The intersection type, vehicle-birthing sequence, vehicle routes and turn-types are generated offline. Each vehicle is removed from simulation when it reaches its destination. This file is then fed into GrooveNet to simulate the intersection protocols. Traffic volume is specified on a per intersection-leg basis, allowing intersection legs to have different traffic levels. Each simulation uses 250 vehicles, and each run is terminated when the last vehicle reaches its destination. The simulation model in GrooveNet was modified to prevent a vehicle from becoming active if vehicles with earlier start times are already present within 10 meters of its starting position in its lane. This feature prevents cars from starting if the lane is already completely backed up.

Our extensions to GrooveNet V2V models do not yet support message loss due to a lossy communication medium. Under this assumption, both TEP and TEP-A behave in identical fashion. Hence, in all our experiments, we will use the name V2V-Intersection as the name of the protocol – it represents both TEP and TEP-A under the assumption of a reliable communication network. We will relax this assumption in future work.

EXPERIMENTAL RESULTS

In our first experiment, we compare different protocols for a perfect-cross intersection with an equal amount of traffic volume in every lane and an equal amount of turn ratios (that is, a vehicle has equal odds of going straight or making a turn at an intersection). The results are presented in Figure 3-(a). As can be expected, the Stop-Sign model results in higher average delays than the other protocols. As the traffic volume increases above 0.1 vehicles per second, the performance of the Stop-Sign model drops dramatically and significant traffic backlog results. In contrast, both Traffic-Light models behave at a near-constant level until the traffic volume reaches 0.25 cars per second for the Traffic-Light model with a green-light time of 10 seconds and 0.3 cars per second for the Traffic-Light model with a green-light time of 30 seconds. After that, the average delay jumps until it settles down at a higher near-constant level at about 0.35 cars per second. Beyond this traffic volume, the Traffic-Light models behave the same regardless of the green-light duration as all the lanes are completely saturated and traffic is backed up significantly. The V2V Intersection model performs the best, doing very well at low traffic volumes up to 0.2 vehicles per second resulting in very negligible delay. As traffic volume increases, the average delay increases and beyond 0.3 cars per second, it performs very similar to the Traffic-Light model with a green-light time of 10 seconds. However, the overall performance improvement is about 26% as compared to the latter Traffic-Light model. Figure 3-(b) zooms into the plot of Figure 3-(a) to show a detailed comparison between the Traffic-Light models and the Intersection model, by not showing the poorly performing Stop-Sign model.

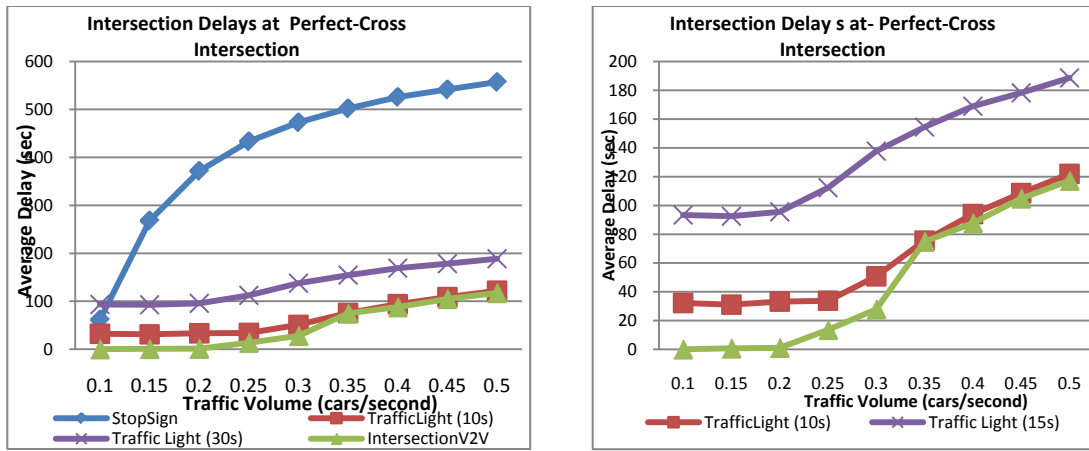


Figure 3: Delays for Perfect-Cross Intersection. Figure (a) shows all protocols. Figure (b) shows more detail w/o the Stop-Sign Protocol.

According to classical queueing theory, the average delay will asymptotically become *very* high when the arrival rate (i.e. traffic intensity) exceeds the service rate (throughput) at the intersection. This delay, however, occurs under steady-state conditions only after a considerable amount of time. Due to practical considerations, our simulations are run for finite durations, and hence capture only transient delay behaviors after overload conditions have been reached. Nevertheless, our results clearly indicate that before overload conditions are reached, the service rate (i.e. throughput) with the V2V-Intersection protocol is noticeably better than the Traffic-Light models.

We then repeated the above experiment for a T-junction and the corresponding results are shown in Figure 4. For the T-junction, the V2V-Intersection protocol has an 83% overall performance improvement over the Traffic-Light model with a 10-second green-light time, and a 94% overall performance improvement over the Traffic-Light model with a 30-second green-light time. The T-junction has fewer conflicts to deal with than at a perfect-cross intersection, resulting in less stopping at the intersection for the V2V-Intersection model leading to its much better performance than before.

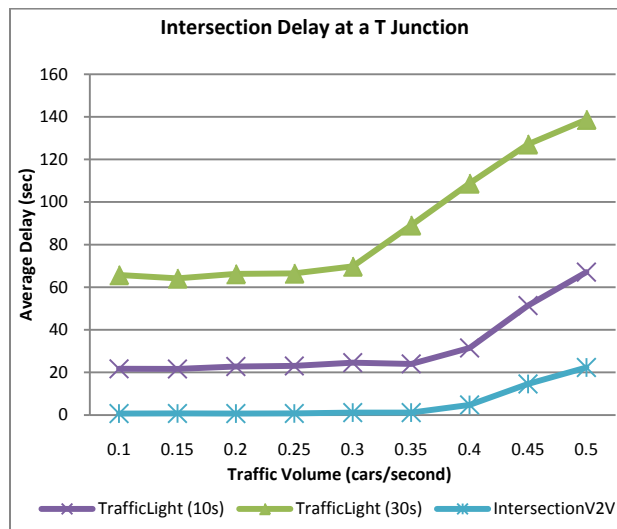


Figure 4: Delays at a T-Junction

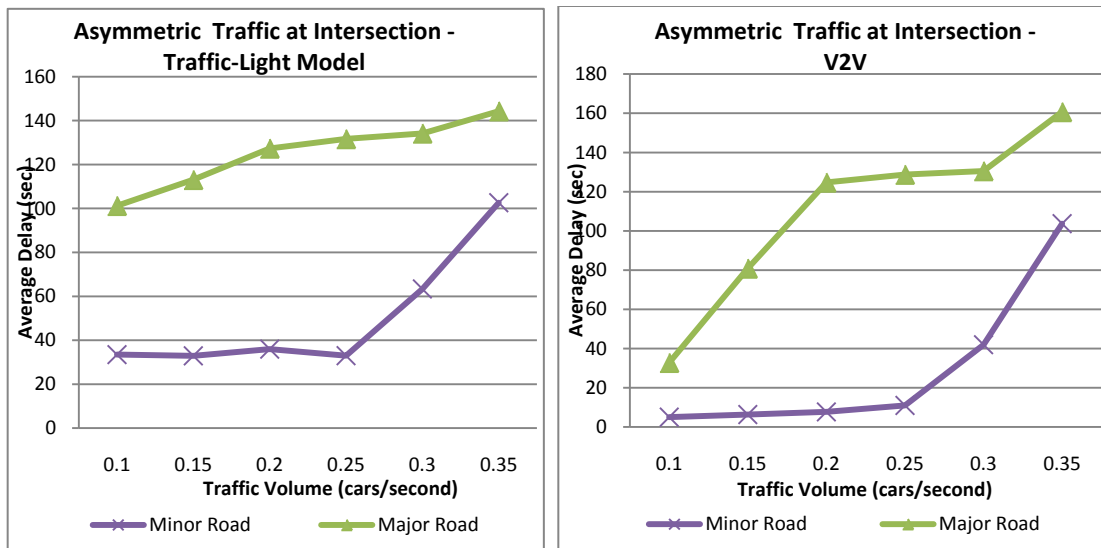


Figure 5: Delays with Asymmetric Traffic (a) Traffic-Light Model (b) V2V Intersection Model

Next, we studied the scenario where traffic varies along different intersecting roads. That is, when two roads intersect, one road has more traffic than the other. However, we still assume that both roads have the same type and hence one does not have priority over the other. The corresponding results are given in Figure 5-(a) for the Traffic-Light Model and Figure 5-(b) for the V2V-Intersection Model. Again, the V2V-Intersection Model performs better than the Traffic-Light Model.

SUMMARY/CONCLUSIONS

A substantial fraction of automotive collisions occur at intersections. Furthermore, intersections are often traffic bottlenecks contributing to significant trip delays. In this paper, our goal was to design intersection management protocols using only vehicle-to-vehicle communications to address these two core issues of safety and throughput. We believe that intersection collisions can be reduced and throughput improved significantly using only V2V protocols. Since installing wireless infrastructure at every intersection to support vehicle to intersection protocols can be prohibitively costly, a V2V-based approach seems more practical for deployment. We have described and evaluated four V2V-based protocols namely Stop-Sign, Traffic-Light, Throughput-Enhancement and Throughput-Enhancement with Agreement protocols. We have also compared these protocols to conventional stop-signs and traffic lights, and have evaluated the average delays encountered at an intersection. We extended GrooveNet [1], a sophisticated hybrid vehicular network simulator, to support these protocols. Our results indicate the potential of these new V2V-based protocols to manage intersections with minimal dependency on infrastructure. Although our protocols are designed for autonomous vehicles that use V2V communication for co-operative driving, they can be adapted to a driver-alert system for manual vehicles at traffic intersections.

LIMITATIONS

The protocols we evaluate do not take into account any controller model for the cars. Since they assume a simplistic movement model based on current speed and current heading, certain assumptions will be violated when applied to real cars, especially when considering the throughput enhancement model where we look at polygon intersections. The ability to integrate several different controller models needs to exist and their effect on the protocols needs to be studied. Currently, we also do not deal with position inaccuracies and packet losses with wireless communication. Position accuracy will affect the protocols since each vehicle depends on its position and the known position of the other vehicles to make safety-critical decisions. Wireless packet loss results in dropped messages between vehicles and this can lead to vehicles not being able to sense other vehicles around them. We also make assumptions at a global level such as the constant speed of all cars (unless they are using the car-following model), and these assumptions are made at a global level. Hence, protocol changes will also need to be made for adapting to scenarios involving different types of cars traveling at different speeds.

FUTURE WORK

We intend to overcome the limitations described above and extend the V2V protocols in the context of real cars. We are working on hybrid simulations with real and simulated vehicles to take advantage of GrooveNet's hybrid environment. We are also working extending our protocols to support enhancements which will allow a vehicle to slow down, and not come to a complete stop, at an intersection to allow another vehicle to cross. We have indeed already implemented a version of this protocol on real-world Segway robots but it is not captured in this paper. There is also ongoing work to look at integration of Vehicle-to-Infrastructure (V2I) technologies within these protocols to take advantage of static known entities at intersections.

REFERENCES

1. R. Mangharam, D. S. Weller, R. Rajkumar, P. Mudalige and Fan Bai, "GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks", Second International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), San Jose, USA. July 2006
2. R. Mangharam, J. Meyers, R. Rajkumar, D. Stancil, J. Parikh, H. Krishnan, and C. Kellum, "A Multi-hop Mobile Networking Test-bed for Telematics" Society for Automotive Engineers (SAE) World Congress. Detroit, USA. April 2005
3. DARPA. The DARPA urban challenge, 2007. <http://www.darpa.mil/grandchallenge>.
4. GM, EN-V concept vehicles at expo 2010, http://media.gm.com/content/media/us/en/news/news_detail.globalnews.html/content/Pages/news/global/en/2010/0817_env_capabilities
5. Kurt Dresner & Peter Stone (2008), Replacing the Stop Sign: Unmanaged Intersection Control, The Fifth Workshop on Agents in Traffic and Transportation Multiagent Systems. pp. 94-101, Estoril, Portugal.
6. Kurt Dresner & Peter Stone (2008), A Multiagent Approach to Autonomous Intersection Management. Journal of Artificial Intelligence Research(JAIR)
7. V. Drabkin, R. Friedman, G. Kliot, and M. Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In The 26th IEEE International Symposium on Reliable Distributed Systems, Beijing, China, October 2007.
8. Golshtein, E. G.; Tretyakov, N.V.; translated by Tretyakov, N.V. (1996). Modified Lagrangians and monotone maps in optimization. New York: Wiley. p. 6.
9. Shimizu, Kiyotaka; Ishizuka, Yo; Bard, Jonathan F. (1997). Nondifferentiable and two-level mathematical programming. Boston: Kluwer Academic Publishers. p. 19.
10. U.S Census Bureau. <http://www.census.gov/geo/www/tiger/>
11. Kurt Dresner, Ph.D. Dissertation, "Autonomous Intersection Management", University of Texas at Austin, May 2010.
12. US Department of Transportation-Federal Highway Administration Publication, National Agenda for Intersection Safety <http://safety.fhwa.dot.gov/intersection/resources/intersafagenda/>
13. Eric Meythaler, 2D Rotated Rectangle Collision <http://www.gamedev.net/reference/programming/features/2dRotatedRectCollision/>

CONTACT INFORMATION

Reza Azimi: sazimi@andrew.cmu.edu

Gaurav Bhatia: gnb@ece.cmu.edu

Ragunathan (Raj) Rajkumar: raj@ece.cmu.edu

Priyantha Mudalige: priyantha.mudalige@gm.com