

Vehicular Networks for Collision Avoidance at Intersections

2011-01-0573

Published
04/12/2011

Seyed Reza Azimi, Gaurav Bhatia, and Ragunathan (Raj) Rajkumar
Carnegie Mellon University

Priyantha Mudalige
GM Technical Center

Copyright © 2011 SAE International

doi:[10.4271/2011-01-0573](https://doi.org/10.4271/2011-01-0573)

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 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 [5,6,7,14,15,18]. 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 currently define an intersection as a perfect square box which has predefined entry and exit points for each lane connected to it. The trajectory of the vehicle crossing the intersection is supposed to be the path taken by the vehicle from the entry to the exit point. We assume that each vehicle has access to a map database that provides routing, lane and road information, in which each segment of the road has a unique identifier (ID). Intersections are also identified by unique IDs in this map database.

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. 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.

Each vehicle broadcasts CRS, NRS, current lane number, as well as the Arrival-Time and the Exit-Time, to all the other vehicles in its communication range. Vehicles are also assumed to have access to a global positioning system (GPS) with locally generated Radio Technical Commission for Maritime (RTCM-104) corrections to achieve Real-time Kinematic (RTK) solution.

Vehicles use this information to determine the other vehicles' turn types. [Figure 1](#) shows an example of this, wherein a vehicle intends on entering the intersection from the east and exiting to the south. Based on the CRS, NRS and lane number, we can figure out that the vehicle is going to make a right turn. We assume in this paper that vehicles can make different turns regardless of their current lane number but they should stay in the same lane after passing any intersection and do not switch lanes. It is relatively easy to restrict this behavior, assumed for convenience here.

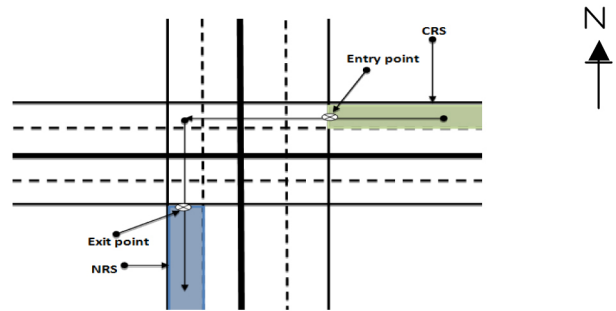


Figure 1.

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

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 2 \(a\)](#), two vehicles are within the intersection at the same time but not occupying the same space. [Figure 2\(b\)](#) 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.

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

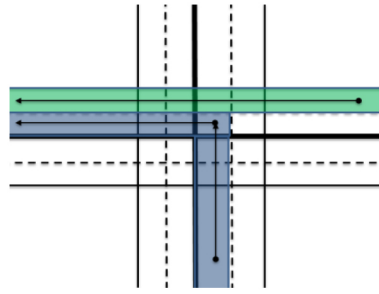


Figure 2(a)

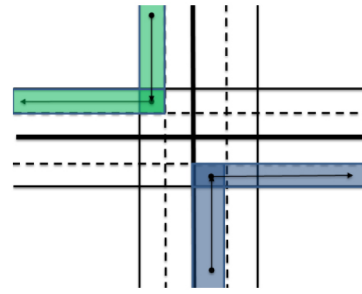


Figure 2(b)

Example scenarios in which no space conflict occurs at the intersection

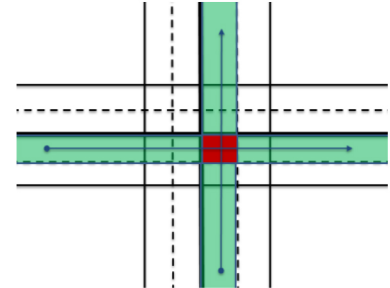
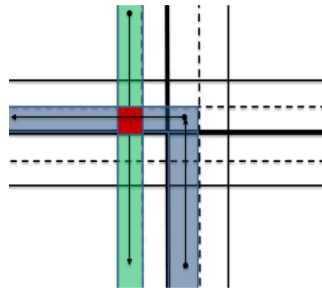
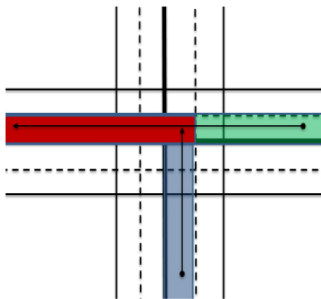


Figure 3. Three example scenarios of space conflict

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 3](#), 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.

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. In the following three protocols, we assume that all vehicles have the same shape and physical dimensions. They do not use any controller model, which means that there is no consideration about a vehicle's movement during acceleration and deceleration. The communication medium has been assumed to be perfect; therefore, no packet loss occurs.

STOP-SIGN PROTOCOL (SSP)

In this protocol, we assume that stop-signs are not physically present at the intersection but vehicles obey the stop-sign rules when they approach an intersection. Vehicles only use V2V communications. 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 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

¹Our 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.