Intersection Management using Vehicular Networks

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

Priyantha Mudalige
General Motors

ABSTRACT

Driving through intersections can be potentially dangerous because nearly 23 percent of the total automotive related fatalities and almost 1 million injury-causing crashes occur at or within intersections every year [1]. The impact of traffic intersections on trip delays also leads to waste of human and natural resources. Our goal is to increase the safety and throughput of traffic intersections using co-operative driving.

In earlier work [2], we have proposed a family of vehicular network protocols, which use Dedicated Short Range Communications (DSRC) and Wireless Access in Vehicular Environment (WAVE) technologies to manage a vehicle's movement at intersections Specifically, we have provided a collision detection algorithm at intersections (CDAI) to avoid potential crashes at or near intersections and improve safety. We have shown that vehicle-to-vehicle (V2V) communications can be used to significantly decrease the trip delays introduced by traffic lights and stop signs. In this paper, we investigate the use of more realistic controller models and higher concurrency to improve V2V intersection protocols for autonomous driving at intersections. We quantify the throughput enhancements due to the use of V2V under various driving conditions, while maintaining safe passage through intersections.

1. INTRODUCTION

Based on the statistics collected from on Federal Highway Administration (FHWA), a significantly high number of vehicle crashes occur at intersections which are currently managed by stop signs and traffic lights. The future of transportation points towards autonomous driving as a way of reducing fatalities and optimizing traffic flow management. Within this context, we advocate the use of vehicular networks to design distributed protocols in which vehicles are able to interact with each other using vehicle-to-vehicle (V2V) communication. Our goal is to enhance safety while decreasing the delay introduced by stop signs or traffic lights by using our V2V-based intersection management protocols.

Vehicle-to-Infrastructure (V2I) communication is an approach that has been used to address the intersection problem in prior work in this domain [3,4,5,6,7]. As the word infrastructure implies, the system mainly consists of an intelligent and powerful computational and communicational unit which would be installed at each intersection to communicate with all approaching vehicles and manage traffic by reserving a safe time-space passage through the intersection for each vehicle. This approach is not very practical because of the high cost and inertia of installing the intersection manager at each intersection. Another drawback of such a centralized system is that if the intersection manager fails, crossing the intersection could be chaotic and dangerous, similar to signal breakdown at a busy intersection. In our previous work, we proposed a V2V intersection protocol, called Throughput Enhancement Protocol (TEP) to manage the traffic through intersections [2]. Our focus in this paper is to (a) improve our V2V-based intersection management protocol and introduce a new collision detection algorithm for intersections, (b) extend an advanced mobility simulator for vehicles and investigate the use of controller model to test the performance of our protocols under realistic driving conditions, and (c) compare the operational efficiency of our improved protocol with our previous work and conventional traffic lights.

The rest of this paper is organized as follows. Section 2 presents the *collision detection algorithm for intersections (CDAI)* which have been used in our new intersection protocol. Section 3 describes the controller model used in realistic mobility models. Section 4 contains a brief summary of our previous V2V-based protocol: *Throughput Enhancement Protocol (TEP)*, and a full description of our new V2V-intersection protocols: *Concurrent Crossing-Intersection Protocol (CC-IP)* and Maximum *Progression-Intersection Protocol (MP-IP)*. In Section 5, we present the evaluation of our protocols using a new hybrid emulator, AutoSim. Section 6 presents our concluding remarks and future work.

2. COLLISION DETECTION AT INTERSECTIONS

In this paper, we define an *intersection* as a perfect square box which has predefined *entry and exit points* for each lane connected to it. The intersection area is considered as a grid which is divided into small *cells*. Each cell in the intersection grid is associated with a unique identifier. Figure 1 shows an intersection with two lanes entering the intersection grid from all four directions.

We make the following assumptions. Each vehicle has access to a digital map database that provides road and lane information. Intersections are identified by unique identifiers (IDs) on this map. Intersections have well-defined approach and exit lanes. Vehicles also have access to a global positioning system (GPS) with locally generated Radio Technical Commission for Maritime (RTCM-104) corrections to obtain a Real-time Kinematic (RTK) solution in order to achieve reliable lane-level vehicle positioning. Such GPS augmentation can be made available by local base stations or through commercial service providers.

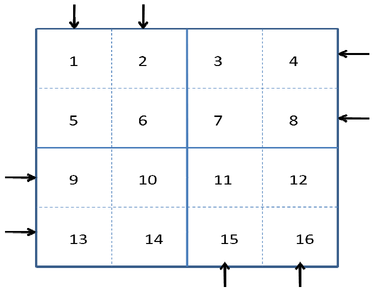

Figure 1, Intersection grid divided into small cells

In the context of an intersection, we define the **current road segment (CRS)** as the road segment that a vehicle is on *before* the intersection, and the **next road segment (NRS)** represents the road segment that the vehicle will be on *after* crossing the intersection. We use an offline table, which we refer to as the **Trajectory Cells Table (TCT)** to determine the cells which will be occupied by the vehicle while crossing the intersection area. TCT uses the CRS, NRS and the lane number as inputs, and returns a list of cell numbers, which will be referred to as **Trajectory Cells List (TCL).** The TCL is sorted based on the order of the cells along the vehicle's trajectory. For example, the first element in the list is reserved for the first cell of the intersection which will be occupied by the vehicle when it enters the intersection grid. Figure 2 shows an illustration of this, wherein a vehicle attempting to enter the intersection from the east and exiting to the west, based on the CRS, NRS and current lane, the vehicle uses the TCT to update its TCL dynamically. In this case, the initial value of the TCL will be *[8, 7, 6, 5]*.

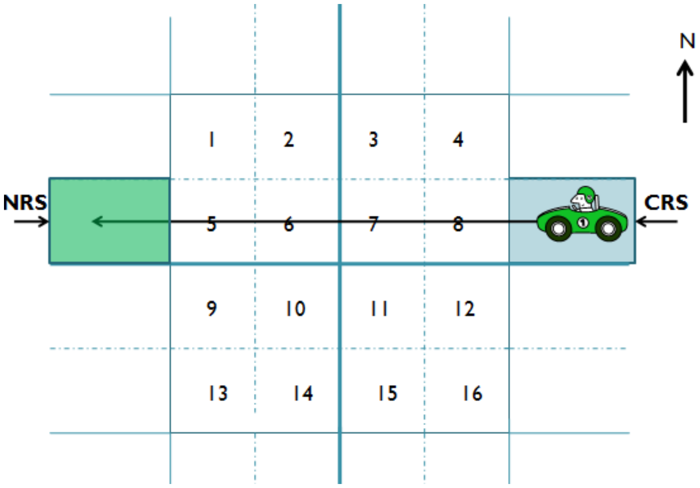

Figure 2, Illustration of TCL, CRS and NRS

In order to update the Trajectory Cells List (TCL) accurately, each vehicle should be aware of the current cell it is occupying inside the intersection grid. As we mentioned before, all vehicles are equipped with GPS devices and have access to the digital map database as well as the intersection's coordinates. Therefore, each vehicle is able to use this information to map the current GPS coordinates to its current cell number. The current cell number will be then used to update the TCL and will be broadcast to surrounding vehicles as part of the basic safety message (BSM) [11].

To update the TCL, each vehicle determines its current cell number using GPS to map to cell number. If the vehicle detects that it has not entered the intersection box, it does not modify the TCL. In this case, the TCL contains the full list of cell numbers which will be occupied by the vehicle while crossing the intersection area. If the vehicle is inside the intersection box, then it uses the current cell number and modifies the TCL as follows. As the trajectory cells list is sorted, using the current cell number, the vehicle can tell which cells have already been crossed and what the next cells along its trajectory are. So, the vehicle updates the TCL by removing cell numbers that have been completely passed and the new TCL contains the current cell number and the remaining cells of the trajectory. Figure 3 shows an example, wherein a vehicle is crossing the intersection, and updating the TCL based on its location outside or inside the intersection grid.

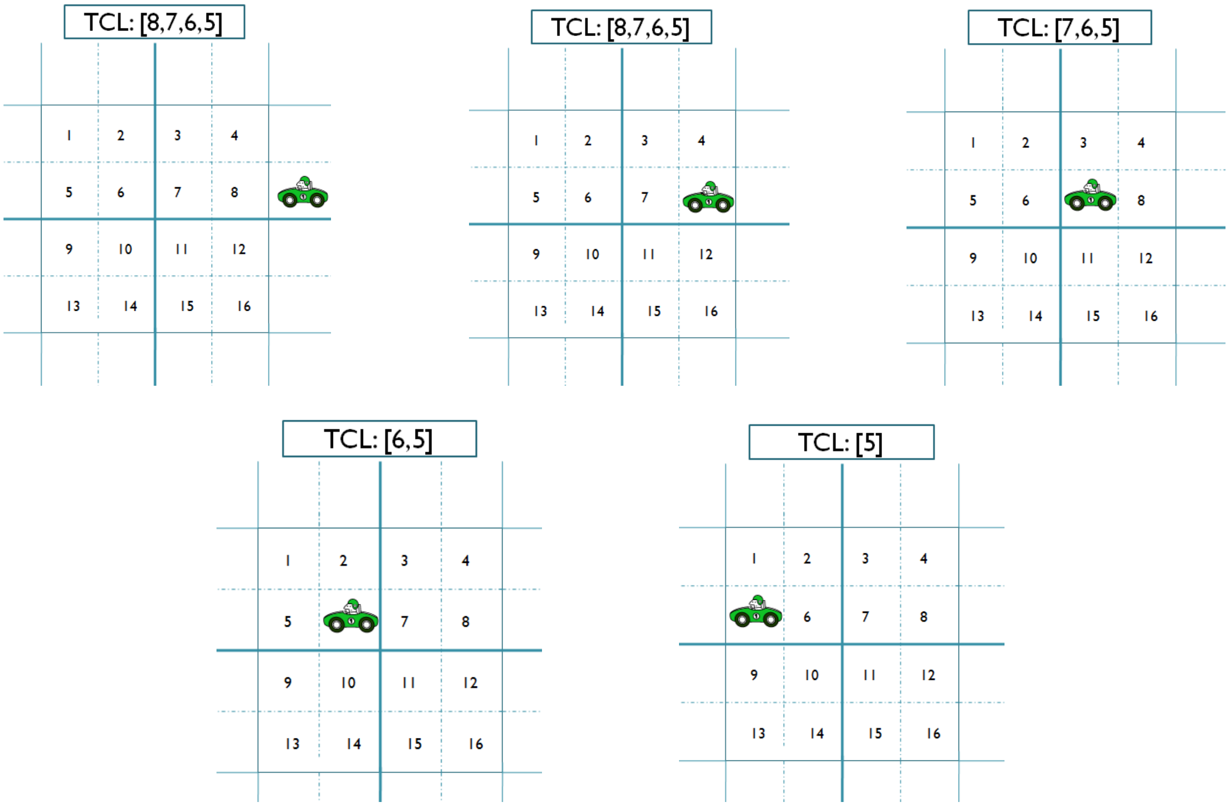

Figure 3, Sequence of updating the TCL

A collision occurs inside the intersection area if two or more vehicles have time and space conflicts. In other words, vehicles get to a potential collision if they have overlapping (Arrival-Time, Exit-Time) intervals and they occupy at least one common cell along their trajectories through the intersection. If any of these conditions is false, then there will be no conflicts and vehicles can continue along their trajectories safely.

Our **Collision Detection Algorithm for Intersections** (CDAI) runs on all vehicles, using the information obtained from received safety messages, which are broadcast by surrounding vehicles. The algorithm uses the Trajectory Cells Lists of the sender and the receiver of the safety messages and by comparing the two lists, it determines if there is any common cell along their trajectories while crossing the intersection. If a potential collision is detected by CDAI, the algorithm returns the first conflicting cell number which we refer to as the **Trajectory-Intersecting Cell (TIC).**

Figures 4(a) and 4(b) show two scenarios in which two vehicles are inside the intersection area at the same time but they have no TIC as they do not occupy any common cell along their transaction inside the intersection grid. In this case, both vehicles can cross the intersection at the same time without any collision. As illustrated in Figures 5(a) and 5(b), a potential collision may occur when two vehicles are going to be inside the intersection at the same time and having cell conflict. Figure 5(b) shows a scenario in which one vehicle is approaching the intersection from the east and attempting to exit to the west, while the other vehicle is entering from the south and also exiting to the west. These two vehicles share more than one intersection cell (cell number 6, 7 and 5) along their trajectories through the intersection area. In such cases where there are multiple conflicting cells, the CDAI returns the first TIC, which is cell number 7 for this example.

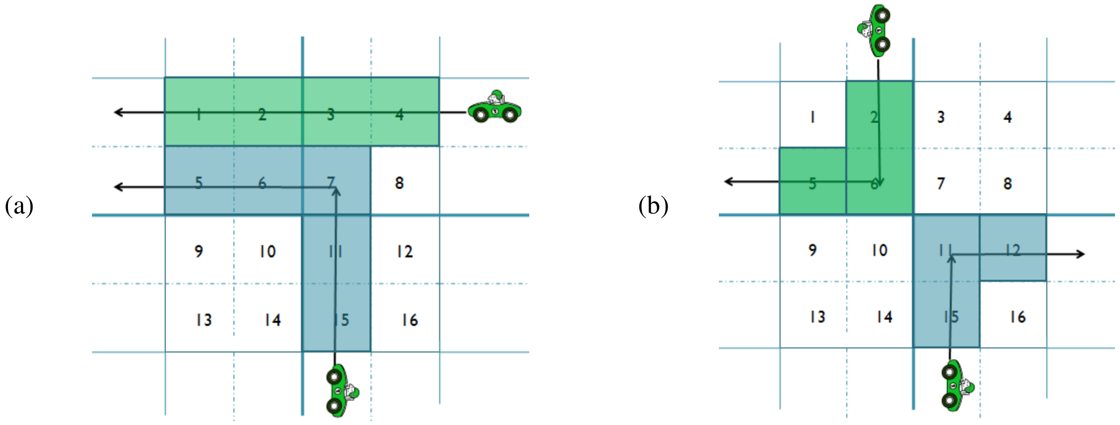

Figure 4, Example scenarios in which no space conflict occurs at the intersection

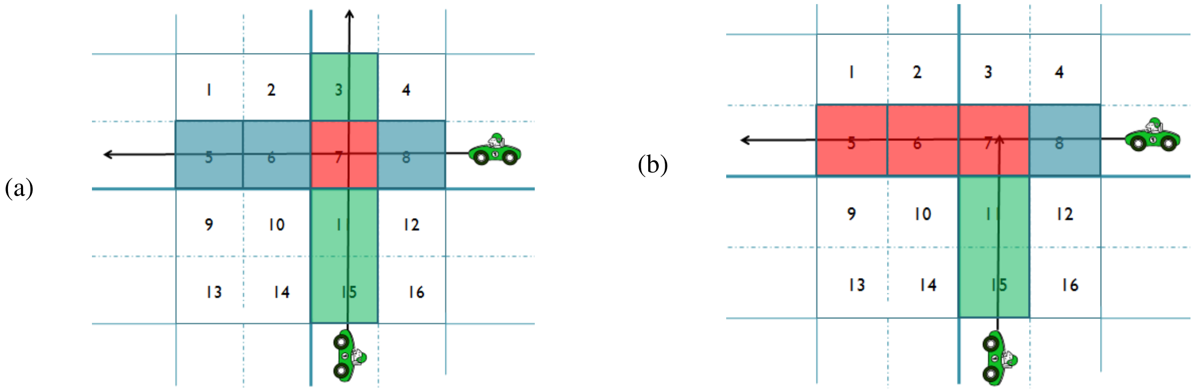

Figure 5, Example scenarios of space conflict

In the case that a potential collision is detected by CDAI, the **"first come, first served" (FCFS)** algorithm is used to assign priorities to vehicles. Based on FCFS, a vehicle, which gets to the entrance of the intersection with a lower arrival time value, gets to cross the intersection before other vehicles with higher arrival times. To avoid any deadlock situation, in which two or more vehicles have the same arrival time, tie-breaking rules apply. If vehicles arrive at an intersection at the same time, our **priority policy** assigns higher priorities to vehicles entering the intersection using primary roads than vehicles arriving from secondary roads. If these still result in a tie among vehicles, the vehicle with a higher Vehicle ID Number (VIN) will have a higher priority and gets to cross the intersection grid first. The VIN is unique for each vehicle.

The CDAI algorithm runs on each vehicle, and based on the information obtained from the received v2v messages, decides if the vehicle can cross the intersection safely or it should only progress up to a trajectory-intersecting cell to avoid collisions.

3. CONTROLLER MODEL

In this section, we describe the controller model which is designed to control the movement of vehicles based on their speed, acceleration and deceleration profiles. This impacts the movement of vehicles through the intersections. The model has been designed based on the assumption that all vehicles have the same shape and physical dimensions. Vehicles also benefit from similar dynamic capabilities such as acceleration and deceleration.

Current vehicles have various acceleration and deceleration properties. As of year 2010, the amount of time that an average vehicle takes to accelerate from 0 miles per hour (mph) to 60 mph on average was around 8.95 seconds. This value has been calculated by averaging the acceleration parameter of 1,807 car records [8]. Using this value, we calculate the maximum acceleration to be 2.9969 m/s^2. Each vehicle uses the controller model to determine its speed and acceleration at any moment of its trajectory.

The controller model interacts with mobility models. When the mobility model assigned to a vehicle sets the desired speed based on the received safety messages or traffic light situation, the controller model calculates the actual speed of the vehicle for the next simulation cycle. We define the following terms:

- **Desired Velocity ($\boldsymbol{V_{Desired}}$):** the speed that the vehicle is planning to achieve.
- **Available Distance ($\boldsymbol{D_{Available}}$):** the maximum distance within which the vehicle should achieve the $\boldsymbol{V_{Desired}}$.
- **Available Time ($\boldsymbol{T_{Available}}$):** the maximum time during which, the vehicle should achieve the $\boldsymbol{V_{Desired}}$.
- **Update Frequency ($\boldsymbol{f_{Update}}$):** the update rate of the simulator, which is the rate of updating the position of each vehicle on the digital map as well as updating the movement parameters of the vehicle such as its speed and direction.
- **Cycle Duration ($\boldsymbol{T_{Cycle}}$):** the duration of one simulation cycle, which is equal to $1/\boldsymbol{f_{Update}}$.
- **Current velocity ($\boldsymbol{V_{Current}}$):** the actual velocity of the vehicle.
- **Average Velocity ($\boldsymbol{V_{Average}}$):** the average of $\boldsymbol{V_{Current}}$ and $\boldsymbol{V_{Desired}}$.
- **Next Velocity ($\boldsymbol{V_{Next}}$):** the speed of the vehicle during the next cycle of computation.

Based on the Newtonian equations of motion, we calculate $\boldsymbol{V_{Next}}$ as follows:

$$\boldsymbol{D(t) = D_0 + V_{Current}t + \frac{1}{2}at^2} \quad \textbf{(1)}$$

$$\boldsymbol{T_{Available} = D_{Available} \,/\, V_{Average}} \quad \textbf{(2)}$$

Using Equation (1), the controller model calculates the Acceleration/Deceleration parameter as follows:

$$\boldsymbol{a = \frac{2*(D_{Available} - (V_{Current} * T_{Available}))}{T_{Available}^{\;2}}} \quad \textbf{(3)}$$

Then, $\boldsymbol{V_{Next}}$ can be calculated as:

$$\boldsymbol{V_{Next} = a * T_{Cycle}} \quad \textbf{(4)}$$

This value which has been calculated by the controller model will be used by the mobility model to set the velocity of the vehicle for the next cycle of computation and to update the position and direction of the vehicle.

4. INTERSECTION PROTOCOLS

In this section, we describe our two new proposed V2V-intersection protocols: (1) *Concurrent Crossing-Intersection Protocol (CC-IP)* and (2) *Maximum Progression-Intersection Protocol (MP-IP)*. We then briefly describe the differences between the two new proposed protocols and our previous protocol, *Throughput Enhancement Protocol (TEP)* [2]. The content of safety messages is detailed in the following section. We have assumed that, in all our protocols, all vehicles have the same shape and physical dimensions. The communication medium has been assumed in this paper to be perfect; therefore, no packet loss occurs.

Concurrent Crossing-Intersection Protocol (CC-IP)

This protocol is designed to increase the throughput at intersections while avoiding collisions. This intersection management protocol is based on pure V2V communications. Each vehicle uses *ENTER, CROSS* and *EXIT* safety messages to interact with other vehicles in its communication range. The following rules are applicable to all vehicles:

- **Sending ENTER:** Every vehicle uses its own GPS coordinates, speed and also the map database to compute the distance to the approaching intersection. If this distance is less than a threshold parameter D_{ENTER}, it starts broadcasting safety messages with the message type set to *ENTER*.
- **Sending CROSS:** When a vehicle enters the intersection area box, it broadcasts the *CROSS* safety message till it exits the intersection boundaries. This message is sent to inform the surrounding vehicles that this vehicle is inside the intersection

area and is currently crossing the intersection. This message contains the sender's identification and trajectory details, identifying the space that will be occupied by the vehicle while crossing the intersection

- **Sending EXIT:** when the vehicle exits the intersection, it broadcasts safety messages with the message type set to *EXIT* until it travels farther than a threshold value D_{EXIT} 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 ENTER:** On receiving an *ENTER* message, the vehicle acts based on the priority assigned to it using the priority policy. The vehicle ignores the message if it has been sent by a vehicle with lower priority, and processes the message received from a higher priority vehicle. If CDAI detects a potential collision with the sender of the *ENTER* message, then the vehicle comes to a complete stop before entering the intersection and waits for the receipt of an *EXIT* message from the same sender.
- **On receiving CROSS:** When a vehicle receives a *CROSS* safety message, it uses the CDAI to detect any potential trajectory conflict with the sender of the *CROSS* message. The decision of the vehicle is derived by the CDAI result as follows:

1. If the CDAI algorithm running on the receiver detects a potential collision with the sender of the *CROSS* message, the receiving vehicle does not enter the intersection area and comes to a complete stop at a safe distance from the entrance of the intersection. Then, it waits to receive the *EXIT* message from the sender of the *CROSS* message to make sure that the space occupied by the sender is now clear and the vehicle can cross the intersection safely.

2. If no potential collision has been detected by CDAI, with the sender of the *CROSS* message, the receiver may still not be allowed to cross the intersection area, as there might be more than one vehicle which has no conflict with the crossing vehicle. In this situation, these vehicles may attempt to cross the intersection area without being aware that they may collide with each other. Figures 6(a) and 6(b) show two situations in which vehicle A is crossing the intersection box and is broadcasting the *CROSS* message. Vehicles B and C are receiving these safety messages and run the CDAI algorithm. Both vehicles get to the same decision that they do not have a potential collision with vehicle A. As can be seen in Figure 6(a), vehicle B and C may collide as they have a conflicting trajectory along the intersection. So only one of them can safely cross through the intersection box while vehicle A is crossing, and the other vehicle can come to a complete stop before entering the intersection box. Vehicles B and C should get to a decision using the priorities assigned to them by the priority policy so that the vehicle with the higher priority can cross, while the other one is waiting outside of the intersection till it receives the *EXIT* message from the higher priority vehicle.

   In Figure 6(b), vehicles B and C can cross the intersection at the same time as vehicle A, since none of them has a space conflict with the other two. So, the receiving vehicles make sure that they do not have trajectory conflicts with each other before entering the intersection area. As discussed before, they may use the received ENTER messages and detect any potential collisions with other receivers of the CROSS message, which are waiting to enter the intersection box. If no potential collision is detected with all other leader vehicles or if it has the highest priority among them, the receiver can cross the intersection safely while broadcasting the CROSS message.

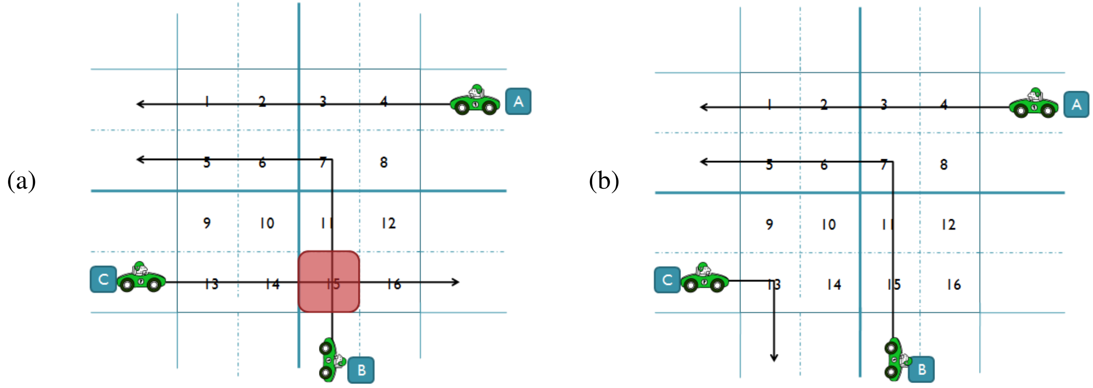

Figure 6, Two examples scenarios for CC-IP

- **On receiving EXIT:** Each vehicle stores the information within all received *ENTER* and *CROSS* messages. On receiving an EXIT message, the vehicle checks if this message is sent from the sender of the ENTER/*CROSS* message that prevented the vehicle from crossing through the. This check is possible by just looking at the unique VIN embedded in the message. If the VIN of the *EXIT* message is the same as the VIN of the last processed ENTER/*CROSS* message, then the space that the vehicle needs to occupy for crossing the intersection is now clear and the vehicle can safely complete its trajectory through the intersection.

Maximum Progression-Intersection Protocol (MP-IP)

MP-IP has been designed to increase the throughput by allowing even conflicting vehicles to make maximal progress *inside* the intersection area, without sacrificing the primary goal of safety. This V2V protocol makes use of the *Trajectory Cells List (TCL)* and our new collision detection algorithm for intersections (*CDAI).* The following rules are applicable to all vehicles:

- **Sending ENTER:** When the vehicle's distance to the intersection is less than a threshold parameter D_{ENTER}, it broadcasts *ENTER* safety messages. This message contains the vehicle's ID and full list of cell numbers; *TCL*, which will be occupied by the vehicle along its trajectory crossing the intersection area.
- **Sending CROSS:** When a vehicle enters the intersection area box, it broadcasts the *CROSS* safety message till it exits the intersection boundaries. Same as in *ENTER* message, this message contains the *TCL*, but the list of cell numbers attached to this message changes as the vehicle makes progress inside the intersection box. As we described in Section 2, at any moment of a vehicle's progress through the intersection, the updated TCL contains the current cell number occupied by the vehicle and the next cells on its trajectory. The updated *TCL* is broadcast within the *CROSS* safety message.
- **Sending EXIT:** when the vehicle exits the intersection, it broadcasts safety messages with the message type set to *EXIT* until it travels farther than a threshold value D_{EXIT} 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 ENTER/CROSS:** on receiving *ENTER or CROSS* messages, the vehicle acts based on the priority assigned to it using the priority policy. The vehicle ignores the message if it has been sent by a vehicle with lower priority, and processes the message received from a higher-priority vehicle. The vehicle then uses the trajectory cells list attached to the *ENTER* message and checks if there is any common cell between that list and its own *TCL*. If no potential collision has been detected by the CDAI, the vehicle can cross the intersection safely. Otherwise, the vehicle shall only proceed inside the intersection area up to the trajectory intersecting cell (TIC) and stop *before* entering that cell.

  On receiving further messages from the same sender, the vehicle checks if there are still conflicting cells, and proceeds based on the updated list attached to the latest *ENTER* message received. Figure 7 shows two vehicles A and B, approaching an intersection. We assume that vehicle A has a higher priority than vehicle B. In this case, vehicle A gets to cross the intersection without stopping or even slowing down. Vehicle B shall progress inside the intersection grid and stop *before* entering the TIC with vehicle A, which is cell number 6. As vehicle A leaves cell number 6, it updates its TCL to [10,14] and sends a CROSS message. This informs vehicle B that the TIC is now clear and it can continue its trajectory through the intersection by proceeding to cell number 6.

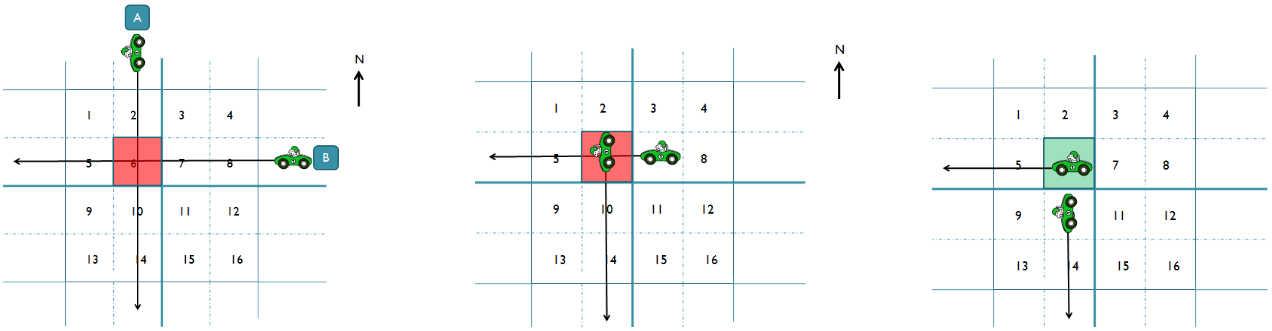

Figure 7, an example scenario for MP-IP

- **On receiving EXIT:** On receiving an EXIT message, the vehicle checks if this message is sent from the sender of the ENTER/*CROSS* message that prevented the vehicle from crossing the intersection and forced the vehicle to only progress and stop before entering a trajectory-conflicting cell. If that is the case, the cell that the vehicle needs to occupy for crossing the intersection is now clear and the vehicle can progress inside the intersection grid and enters that cell.

V2V-INTERSECTION PROTOCOL IMPROVEMENTS

We now briefly describe the differences between our previous protocol, *Throughput Enhancement Protocol (TEP),* and the two new V2V-intersection protocols, *Concurrent Crossing-Intersection Protocol (CC-IP)* and *Maximum Progression-Intersection Protocol (MP-IP)*, presented in this paper.

In *TEP*, when vehicles approach an intersection and while crossing through it, till they exit the intersection box, they broadcast *STOP* safety message to all surrounding vehicles. To increase the throughput of the intersection, in *CC-IP*, we added *CROSS* safety messages, which are sent by vehicles inside the intersection area. This allows more parallelism as receiving vehicles detecting no trajectory conflict with the sender of the *CROSS* message can cross through the intersection at the same time. This concurrency occurs regardless of their arrival time at the intersection. In the case that the vehicle detects no potential collision with the vehicle already crossing the intersection, it can simultaneously pass through the intersection box, even though the vehicle has arrived later than some others. This decreases the intersection delay when compared to the *FCFS* policy.

In *TEP*, when a vehicle detects a potential collision with another vehicle, and loses the competition due to a lower priority assigned to it, the vehicle comes to a complete stop before entering the intersection area. It then waits till it receives the *CLEAR* safety message from the same sender indicating that the intersection area is now safe to cross. In *MP-IP*, the vehicle which has the lower priority uses CDAI and finds out the first common cell (trajectory intersecting cell) with the crossing higher-priority vehicle. Then, instead of not entering the intersection, it progresses inside the intersection area and stops before entering the TIC. As soon as that cell becomes clear, the vehicle can proceed. This allows more vehicles to be inside the intersection grid at the same time and decreases their trip delays.

MP-IP enables the safe passage of vehicles through the intersection area by not allowing more than one vehicle to occupy an intersection cell along their trajectories. In the case that two or more vehicles attempt to cross the intersection at the same time and share a trajectory cell, the priority policy is used. The vehicle with the higher priority can occupy that cell and cross the intersection safely, while the lower priority vehicle comes to a complete stop *before* entering that cell. This prevents any collision among the vehicles crossing the intersection.

INTERSECTION SAFETY MESSAGE TYPES

We now describe in detail the content of transmitted messages.

The `ENTER` 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.
- **Trajectory Cells List**: The list of cell numbers which will be occupied by the vehicle inside the intersection grid.
- **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 `ENTER` in this case.

The `CROSS` message contains the same parameters as the ENTER message. Its **Trajectory Cells List** contains the updated list of trajectory cells, including the current cell and remaining cells along the vehicle's trajectory through the intersection area, and the **Message Type**: `CROSS`.

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

5. PROTOCOL EVALUATION

In this section, we evaluate our proposed protocols, CC-IP and MP-IP, using various mobility models that we have designed and simulated in AutoSim. AutoSim is a hybrid emulation environment for vehicular communications. This emulator is a new next-generation version of GrooveNet [9,10]. AutoSim enables the interaction between real and simulated vehicles, and provides modeling of different aspects of mobility protocols. The architecture consists of several models such as the Control, Communication, Mobility, Localization and Path Tracking. Mobility models include the stop sign, traffic light, our V2V-based intersection models and the V2V car-following model. AutoSim also provides Modeling of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication.

We compare the Traffic-Light model to the V2V-interaction models. The Traffic-Light model has been designed to simulate the behavior of vehicles at intersections equipped with traffic lights with a green light duration of 10 seconds and 30 seconds. In this model, vehicles do not communicate with each other.

METRIC

We define the *trip time* for a vehicle as the time taken by that vehicle to go from a fixed starting point before the intersection to a fixed end point after the intersection. 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 DjikstraTripModel in AutoSim which calculates the shortest route between two points using Djikstra's algorithm. The route is chosen with a waypoint at the intersection forcing the route to pass through the intersection. The logging mechanism has been added to AutoSim 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 *Four-Way Perfect-Cross Intersections*, in which the intersection legs are at perfect right angles to the neighboring leg. We prioritize each leg of the intersection and characterize them as *Primary* and *Secondary* roads based on the priorities assigned to them. Traffic volume is also specified on a per intersection-leg basis, allowing intersection legs to have different traffic levels. We study how assigning higher priorities to the roads with higher traffic volume affects the overall trip delay of vehicles.

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 AutoSim to invoke the intersection protocols. Each simulation run uses 1000 vehicles, and each run is terminated when the last vehicle reaches its destination. The simulation model in AutoSim has been designed to prevent a vehicle from becoming active if vehicles with earlier start times are already present within 10 meters ahead of its starting position in its lane. This simulation parameter prevents cars from starting if the lane is already completely backed up.

EXPERIMENTAL RESULTS

We have compared the traffic light model to our proposed V2V-based protocols for a perfect-cross intersection. Figure 8 shows this comparison. Each experiment includes an equal amount of traffic volume in every direction and an equal amount of turn ratios (that is, a vehicle has equal odds of going straight or making a turn at an intersection). All roads are considered as primary roads and hence they have the same priority. The X-axis is the traffic volume determined as cars per second and the Y-axis is the delay in seconds.

Both V2V-based models perform better than the traffic light model. The *Concurrent Crossing-Intersection Protocol (CC-IP)* and the *Maximum Progression-Intersection Protocol (MP-IP)* have respectively 48.78% and 85.15% overall performance improvements over the traffic light model with a 10-second green light time.

The average delay is very negligible for lower traffic volumes in both V2V-based models. However, in the case of *CC-IP*, as the traffic volume increases, the delay also increases and at beyond a traffic volume of 0.8 cars per second, its performance gets closer to the traffic light model but still outperforming it. Under *MP-IP*, the delay is very low up to the traffic volume of 0.6 vehicles per second. Beyond this point, the average delay increases but its value is significantly lower than the delay in the traffic light model.
MP-IP also outperforms *CC-IP* by 71%.

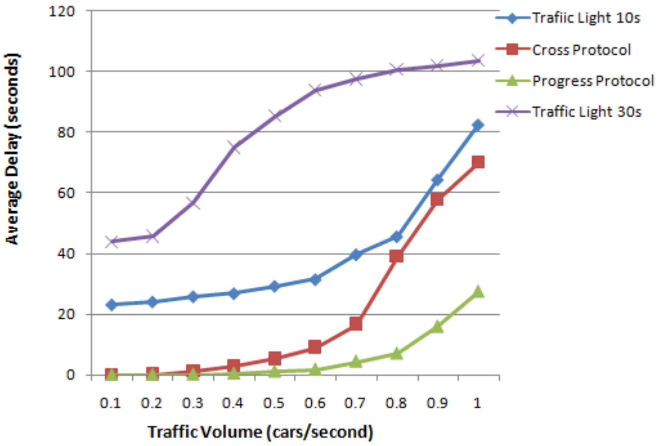

Figure 8: Delays with all protocols: traffic light, CC-IP and MP-IP

Next, we looked at the case where roads have the same traffic volumes but different types, i.e. the north-bound and the south-bound lanes of the intersection are primary roads and the south-bound and the west-bound lanes, are secondary roads. Hence, the vehicles on primary roads have higher priority over vehicles coming from secondary roads to cross the intersection. As we can see in Figure 11, both V2V intersection models perform almost the same as their performance in the scenario that all the roads have the same type (same priority), for low and medium traffic volumes. And they perform slightly better in the case of high traffic densities.

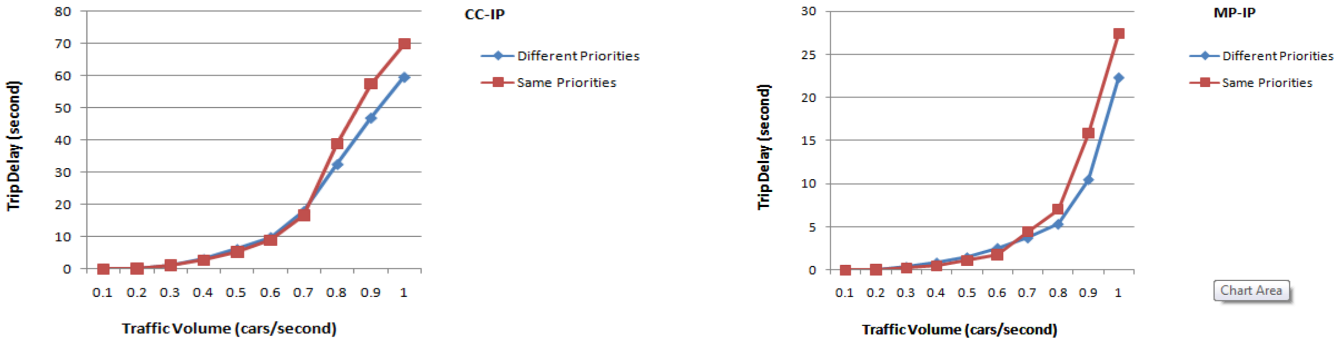

Figure 9: Delay comparison between the normal intersection and when the intersection roads have same traffic volume but different types (priorities) in (a) CC-IP model, (b) MP-IP model

We next studied the case that the traffic volume is significantly different on the intersection roads. We assume that the roads coming to the intersection from the North and the South have higher traffic densities than the roads on the East and the West. However, the roads have the same type and priority to cross the intersection. Figure 9 shows that the *Maximum Progression-Intersection Protocol (MP-IP)* performs better than *the Concurrent Crossing-Intersection Protocol (CC-IP)* and both V2V-based models outperform the traffic light model significantly. Figure 10 illustrates the performance comparisons between the asymmetric traffic (various traffic volumes among the intersection roads) and the symmetric traffic, for each intersection model, including the traffic light model and our two V2V intersection models. As we expect, the vehicles arriving on the higher traffic density roads face higher delays which increases the overall average delay. This increase is significant for the traffic light model and the *MP-IP* model but much less in the case of *CC-IP*. Using the *CROSS* safety messages permits more vehicles from the higher traffic volume legs of the intersection, which are situated in opposite directions (i.e. north and south legs), to cross the intersection area at the same time. The reason is that there are fewer space conflict cases between the vehicles arriving at intersection from opposite directions than the vehicles coming from the other directions.

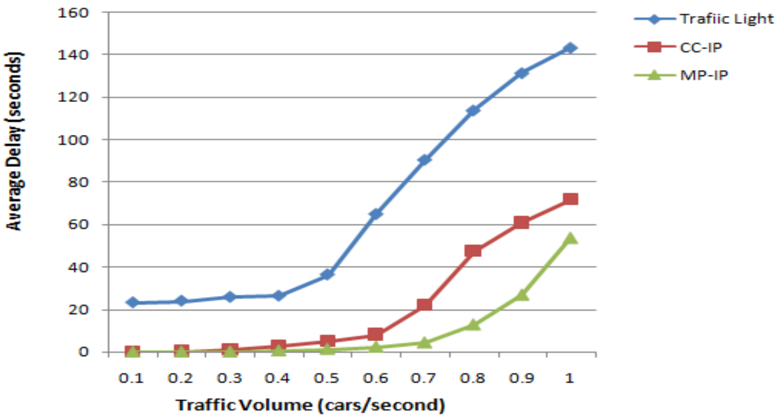

Figure 10: Delays when the traffic is asymmetric and some intersection roads have higher traffic volumes

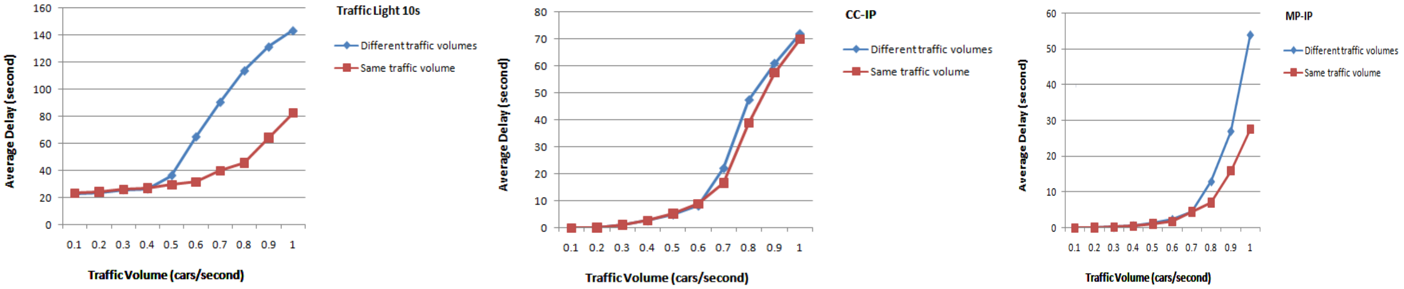

Figure 11: Delay comparisons between the symmetric and asymmetric intersections, managed by (a) Traffic Lights, (b) CC-IP, and (c) MP-IP

We finally studied the scenario in which higher priorities have been assigned to high-traffic-volume roads, in order to allow more vehicles to cross the intersection coming from the higher density roads. This reduces the backed-up traffic and accordingly increases the overall throughput of the intersection. Figure 12 shows the comparisons between two different scenarios using the Concurrent *Crossing-Intersection Protocol (CC-IP)* and the *Maximum Progression-Intersection Protocol (MP-IP)*. Both scenarios look at asymmetric intersections, in which the traffic volume is not the same among all legs of the intersection. In both V2V intersection models, assigning higher priorities to higher traffic volume roads outperforms the scenario that all legs of the intersection have the same priority. The improvement is small at low volume traffic but it increases significantly as the traffic density increases. These results highlight the importance of priority policy and how assigning higher priorities to vehicles on the high volume traffic roads can decrease the overall delay and increase the throughput of the intersection significantly while dealing with large number of vehicles at the intersection.

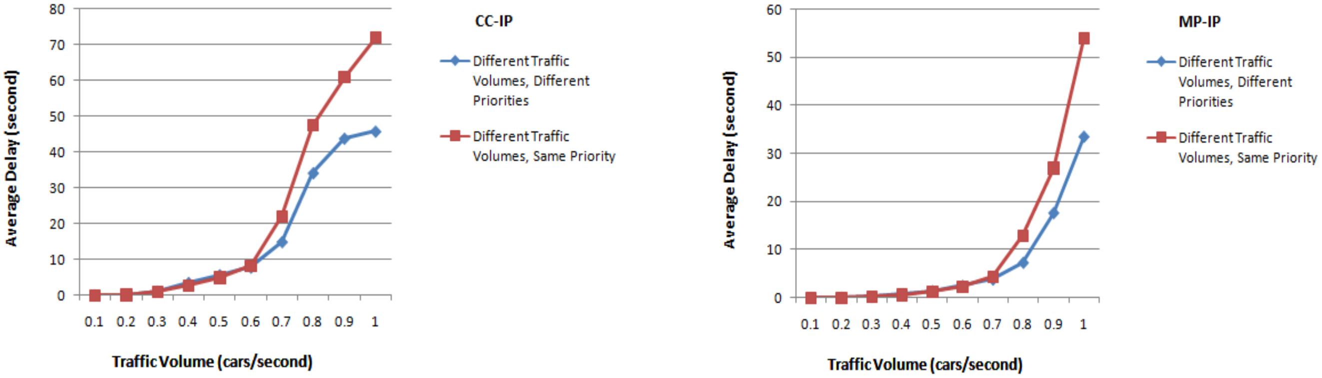

Figure 12: Delays at asymmetric intersections. Comparison of same priority assignment, with assigning higher priorities to higher traffic volumes, managed by V2V-intersection protocols: (a) CC-IP, (b) MP-IP

6. CONCLUSIONS

The percentage of fatal crashes that occur at intersections remains constant over the years at nearly 23% of all vehicle crashes. Infrastructure using traffic lights and stop signs, which have been designed to increase safety, is not very efficient and also decreases the throughput significantly as traffic bottlenecks. In this paper, our goal was to design and simulate new protocols to manage intersections to increase the safety and decrease the trip delay. We advocated the use of V2V-based intersection protocols, as installing more infrastructure at every intersection is costly. We have designed and developed the V2V protocols: *Concurrent Crossing-Intersection Protocol (CC-IP)* and *Maximum Progression-Intersection Protocol (MP-IP)*. These two new V2V protocols significantly and safely enhance concurrency and throughput within intersections. We have compared our intersection protocols to traffic light models by evaluating the average delays encountered at an intersection. We have used AutoSim, a sophisticated hybrid vehicular network simulator, to support these protocols and implemented a realistic controller model to manage the movement of vehicles. Our results indicate significant increase in traffic throughput with the least dependency on infrastructure. Although our protocols are designed for autonomous vehicles that use V2V communication for co-operative driving in future intelligent transportation systems, they can potentially be adapted to a driver-alert system for manual vehicles at traffic intersections.

LIMITATIONS

In our protocols, we 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.

FUTURE WORK

We intend to design new models to study the effects of position inaccuracy and packet loss on the V2V protocols. We are currently extending the V2V protocols in the context of real cars. There is ongoing work to look at possible integration of Vehicle-to-Infrastructure (V2I) technologies within these protocols to take advantage of statically known entities at intersections. We intend to design new protocols which use the integration of V2I and V2V for managing intersections, where autonomous and human-driven vehicles are both present.

References

1. US Department of Transportation-Federal Highway Administration Publication, National Agenda for Intersection Safety http://safety.fhwa.dot.gov/intersection/resources/intersafagenda/
2. Seyed Reza Azimi, Gaurav Bhatia, Ragunathan (Raj) Rajkumar, Priyantha Mudalige Vehicular Networks for Collision Avoidance at Intersections, Society for Automotive Engineers (SAE) World Congress,April,2011
3. 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.
4. Kurt Dresner & Peter Stone (2008), A Multiagent Approach to Autonomous Intersection Management. Journal of Artificial Intelligence Research(JAIR)
5. 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.
6. Vehicle Safety Communications 2 consortium. *Vehicle Safety Communications – Applications (VSC-A)* Final Report. Washington, DC: National Highway Traffic Safety Administration
7. Laurent Bouraoui, Stéphane Petti, Anis Laouiti, Thierry Fraichard and Michel Parent, INRIA Rocquencourt, “Cybercar cooperation for safe intersections”, Proceedings of the IEEE ITSC, 2006 IEEE Intelligent Transportation Systems Conference, Toronto, Canada, September 17-20, 2006
8. 2010 0-60mph statistics. [Online]. Available at: http://www.0-60cartimes.com/0-60-statistics
9. 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
10. 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

11. Draft SAE J2735_200911 Dedicated Short Range (DSRC) message set dictionary.

CONTACT INFORMATION

Reza Azimi: sazimi@andrew.cmu.edu

Gaurav Bhatia: gnb@ece.cmu.edu

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

Priyantha (Pri) Mudalige: priyantha.mudalige@gm.com