STIP: Spatio-Temporal Intersection Protocols for Autonomous Vehicles

Reza Azimi, Gaurav Bhatia, Ragunathan (Raj) Rajkumar
Carnegie Mellon University
rezaazimi@cmu.edu, gnb@ece.cmu.edu, raj@ece.cmu.edu

Priyantha Mudalige
General Motors Company
priyantha.mudalige@gm.com

ABSTRACT

Autonomous driving is likely to be the heart of urban transportation in the future. Autonomous vehicles have the potential to increase the safety of passengers and also to make road trips shorter and more enjoyable. As the first steps toward these goals, many car manufacturers are investing in designing and equipping their vehicles with advanced driver-assist systems. Road intersections are considered to be serious bottlenecks of urban transportation, as more than 44% of all reported crashes in U.S. occur within intersection areas which in turn lead to 8,500 fatalities and approximately 1 million injuries every year. Furthermore, the impact of road intersections on traffic delays leads to enormous waste of human and natural resources. In this paper, we therefore focus on intersection management in Intelligent Transportation Systems (ITS) research. In the future, when dealing with autonomous vehicles, it is critical to address safety and throughput concerns that arise from autonomous driving through intersections and roundabouts.

Our goal is to provide vehicles with a safe and efficient passage method through intersections and roundabouts. We have been investigating vehicle-to-vehicle (V2V) communications as a part of co-operative driving in the context of autonomous driving. We have designed and developed efficient and reliable intersection protocols to avoid vehicle collisions at intersections and increase traffic throughput. In this paper, we introduce new V2V intersection protocols to achieve the above goals. We show that, in addition to intersections, these protocols are also applicable to vehicle crossings at roundabouts. Additionally, we study the effects of position inaccuracy of commonly-used GPS devices on some of our V2V intersection protocols and suggest required modifications to guarantee their safety and efficiency despite these impairments. Our simulation results show that we are able to avoid collisions and also increase the throughput of the intersections up to 87.82% compared to common traffic-light signalized intersections.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications
C.2.4 [**Special-purpose and Application-Based Systems**]: Real-time and embedded systems
C.2.4 [**Robotics**]: Autonomous vehicles

1. INTRODUCTION

Road intersections are currently managed by stop signs or traffic lights. These technologies were designed to manage traffic and increase the safety at intersections, but there is growing concern about their efficiency and safety. Each year, more than 2.8 million intersection-related crashes occur in the United States, accounting for more than 44% of all reported crashes [4]. It is established that roundabouts are safer than junctions. According to a study of a sampling of roundabouts in the United States [6], when compared with the junctions they replaced, roundabouts have 40% fewer vehicle collisions, 80% fewer injuries and 90% fewer serious injuries and fatalities. In addition, the delays introduced by stop signs and traffic lights significantly increase trip times. This leads to a huge waste of human and natural resources. The 2011 Urban Mobility Report [1], published by the Texas Transportation Institute, illustrates that the amount of delay endured by the average commuter is 34 hours which costs more than $100 billion each year.

Autonomous driving is progressing rapidly and is generally expected to play a significant role in the future of automotive transportation. For example, various autonomous vehicles have been demonstrated at the DARPA Urban Challenge [2]. We consider this to be a major opportunity to introduce new methods which are suitable for autonomous driving at intersections and roundabouts, and thereby provide solutions for safety and efficiency problems that currently hamper current traffic management technologies at intersections.

In our previous work, we have introduced a family of vehicular network protocols to manage the safe passage of traffic across intersections [9,10,11]. These completely distributed protocols rely on vehicle-to-vehicle (V2V) communications and localization to control and navigate vehicles within the intersection area. Autonomous vehicles approaching an intersection use Dedicated Short Range Communications (DSRC) and Wireless Access in a Vehicular Environment (WAVE) [3] to periodically broadcast information such as position, heading and intersection crossing intentions to other vehicles. The vehicles then decide among themselves regarding such questions as who crosses first, who goes next and who waits. However, localization and positioning accuracy is crucial for safety applications such as intersection collision avoidance. GPS position inaccuracy affects various distance measurements and may lead to vehicle collisions inside and outside of the intersection/roundabout area.

In our current work, we have designed and developed new intersection protocols with a realistic GPS model. They have been implemented in our hybrid emulator-simulator for vehicular networks, called AutoSim.

The rest of this paper is organized as follows. Section II describes our assumptions for constructing intersections and roundabouts. Section III includes our new V2V Intersection protocols. Section IV describes the solution to the effects of position inaccuracy on our protocols. Section V includes the implementation of our V2V intersection protocols and the GPS model in AutoSim. In Section VI, we evaluate our protocols. Section VII presents our conclusions and future work.

2. ASSUMPTIONS

In this section, we define intersections and roundabouts and state the related physical assumptions. All vehicles are assumed to follow the First-Come, First-Served (FCFS) policy, in which the vehicle with the lower arrival time to the intersection has the higher priority. In the scenarios that two or more vehicles arrive almost at the same time, they break the ties in favor of vehicles approaching on main roads. If the tie still holds, it is broken by Vehicle Identification Number (VIN), which has uniquely assigned to each vehicle.

2.1 INTERSECTIONS

The intersection area is modeled as a grid which is divided into small cells. Each cell in the grid is associated with a unique identifier. 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. Each vehicle 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).* Therefore a vehicle's TCL is defined as the ordered list of the cell numbers which will be occupied by that vehicle along its trajectory inside the intersection box. Figure 1 shows an intersection with two lanes entering the intersection grid from all four directions. In this example scenario, vehicle A's TCL includes cell numbers {15,11,7,3} and vehicle B's TCL is {8,7,6,5}.

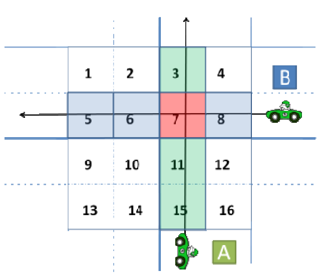


Figure1. An Intersection Scenario

2.2 ROUNDABOUTS

A roundabout is a channelized intersection with one-way traffic flow circulating around a central island. Roundabouts are also considered as "traffic-calming" devices since all traffic is slowed to the design speed of the one-way circulating roadway. These slower speeds reduce the severity of crashes, and minimize the total number of all crashes inside the roundabout area [5,6]. In addition to slower speeds, roundabouts have fewer conflict points than traditional intersections. Figure 2 shows a comparison of conflict points between a single-lane roundabout and a single-lane perfect cross intersection. Note that the cross intersection includes 32 conflict points while the number of conflict points at the roundabout is only a quarter of that number, meaning 8 points [12].

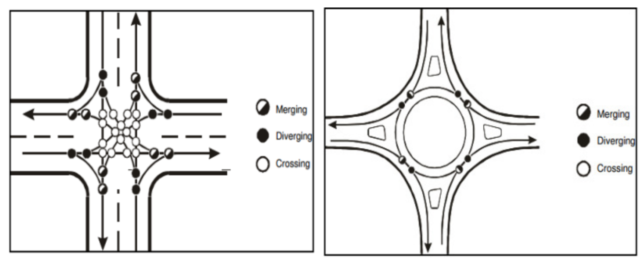


Figure2. Traffic conflicting points at a simple cross intersection and a 1-lane roundabout

We currently define the roundabout as a channelized intersection, in which four roads converge towards a central island from different directions. Each road includes pre-defined *entry and exit points* for each lane connected to it. Figure 3 shows a 1-lane roundabout grid.

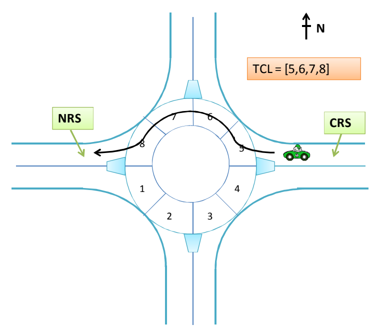


Figure3. Roundabout Grid. Illustration of CRS, NRS, TCL.

Collision Detection Algorithm for Intersections (CDAI) [9] is running on each vehicle to detect potential collisions in intersections/roundabouts using the information obtained from received safety messages broadcast by surrounding vehicles. This algorithm finds a common cell along two vehicles' trajectories. In other words, when there is a common cell along two vehicles' trajectories, they might get into a potential collision if they cross the intersection simultaneously. We refer to the common cell as *Trajectory Intersecting Cell (TIC).* In the case of Figure1, TIC is cell number 7.

3. V2V INTERSECTION PROTOCOLS

In this section, we describe our V2V protocols, which we refer to as Spatio-Temporal Intersection Protocols (STIP). These protocols have been designed to increase the throughput at intersections while avoiding collisions. Vehicles use V2V communications using DSRC/WAVE to broadcast intersection safety messages to other vehicles in their communication range. These protocols enable co-operative driving among approaching vehicles to ensure their safe passage through the intersection. Our assumption is that all the vehicles are equipped with Global Positioning System (GPS) devices and have access to a digital map database, which provide them with critical information such as position, heading, speed, road and lane details. Intersection safety messages are broadcast at 10Hz and they contain the trajectory details of the sender along the intersection area. The format of these safety messages is defined by the SAE's J2735 standard [3]. We use the second part of Basic Safety Messages (BSM) for the extra information in our intersection safety messages. We have assumed that, in all our protocols, all vehicles have similar shape and physical dimensions.[1]

3.1 Intersection Safety Messages

In our intersection protocols, each vehicle uses 3 types of intersection safety messages to interact with other vehicles within its communication range.

1) An ENTER message is used to inform the neighboring vehicles that the vehicle is approaching the intersection area with specific crossing intentions. The ENTER message contains 9 parameters: *Vehicle ID, Current Road Segment, Current Lane, Next Road Segment, Next Vertex, Arrival-Time, Exit-Time, Trajectory Cells List, Cells Arrival Time List, Message Sequence Number and Message Type*, which is ENTER in this case.
2) A CROSS message is to inform that the vehicle is inside the intersection grid. This message contains the sender's identification and trajectory details, identifying the space that will be occupied by the vehicle while crossing the intersection. The CROSS message contains the same parameters as the ENTER message. Its Trajectory Cells List contains the updated list of trajectory cells and their related arrival times for the current cell and remaining cells along the vehicle's trajectory through the intersection area, and the *CROSS Message Type*.
3) An EXIT message indicates that the vehicle has exited the intersection boundaries. The EXIT message contains 3 parameters: *Vehicle ID, Message Sequence Number, and EXIT Message Type*.

Every vehicle uses its own GPS coordinates, speed and also the map database to compute the distance to the approaching intersection and the distance passed from the previous intersection. We consider four *intersection states* for each vehicle based on its relative location to the intersection area.

1) *Intersection-Approach*: when vehicle's distance to the approaching intersection is less than a threshold parameter D_{ENTER}
2) *Intersection-Wait*: when the vehicle is stopped at the entrance of the intersection and waiting for other vehicles.
3) *Intersection-Enter*: when the vehicle is inside the intersection grid's boundaries.
4) *Intersection-Exit*: when the vehicle exits the intersection, until it travels farther than a threshold value D_{EXIT} from the exit point of the intersection.

3.2 Minimal and High Concurrency Protocols Overview

We have categorized our Spatio-Temporal Intersection Protocols (STIP) based on the actions taken by potentially conflicting vehicles to avoid collisions. Potentially conflicting vehicles are those vehicles which have trajectory conflicts with one or more crossing vehicles through the intersection area and may get into a potential collision. We will present three classes of STIP in this paper and study their properties.

Minimal Concurrency Protocols (MCP) includes Throughput Enhancement Protocol (TEP) and Concurrent Crossing-Intersection Protocol (CC-IP) [9,10]. In this category, the conflicting vehicle with higher priority can ignore the intersection safety messages from other lower-priority vehicles and cross the intersection without slowing down or stopping. However, any lower-priority vehicle is super-cautious and, when it loses a competition, it comes to a complete stop before entering the intersection boundaries, and waits till it receives an EXIT message, from the higher-priority vehicle. This message informs the lower-priority vehicle that the higher-priority vehicle has crossed the intersection and now the intersection area is safe for its passage. This protocol is applied across all priority levels.

High Concurrency Protocols (HCP) includes the Maximum Progression Intersection Protocol (MP-IP) and the Advanced Maximum Progression Intersection Protocol (AMP-IP) [11].The main goal is to increase the parallelism inside the intersection area by allowing more vehicles to cross the intersection at the same time. This goal is achieved by allowing even conflicting vehicles to make maximal progress inside the intersection area, without sacrificing the primary goal of safety. This category allows

[1] This assumption can be relaxed easily but makes the presentation complex.

even potentially conflicting vehicles to progress inside the intersection area, and the lower-priority vehicle gets to a complete stop before entering the conflicting cell, and waits till the higher-priority vehicle has crossed and cleared that cell.

3.3 High Concurrency Protocols with Slowdown (HCPS)

We now introduce a new class of STIP protocols, called **High Concurrency Protocols with Slowdown (HCPS).** HCPS includes Advanced Cross Intersection Protocol (AC-IP) and Advanced Progression Intersection Protocol (AP-IP).

In MCP and HCP, STIP protocols, potentially conflicting vehicles with lower priority must come to a complete stop outside or inside the intersection area to allow the safe passage of higher-priority conflicting vehicle. When the vehicle comes to a complete stop inside or outside of the intersection box, it needs to start again and accelerate to reach its desired speed. The delay due to stopping and starting again depends on the vehicle's dynamics such as its acceleration parameter. This delay is not negligible, and multiple stop and moves increases the trip time of the vehicle.

The goal of our new protocols is to decrease the delays due to complete stops, and also to increase the fuel efficiency of vehicles. Additionally, avoiding numerous stops will increase the comfort of passengers. To achieve these goals, **HCPS** protocols allow lower-priority conflicting vehicles to slow down while approaching an intersection and prior to the conflicting cell, to provide the higher priority-vehicle with necessary time gap to cross. This will minimize a vehicle's need to get to a complete stop, and also the total number of stops and startups will be decreased significantly.

3.3.1 Advanced Cross Intersection Protocol (AC-IP)

This protocol is designed to increase the throughput at intersections while avoiding collisions. This intersection management protocol is based on pure V2V communications. The key idea of this protocol is to allow non-conflicting vehicles to concurrently cross the intersection. Each vehicle uses ENTER, CROSS and EXIT safety messages to interact with other vehicles in its communication range.

- P_v: Priority of vehicle *v*. This is determined by the *priority policy*.
- S_v: Set of cells required for vehicle *v* to cross the intersection. It consists of the current cell and next cells that will be occupied by vehicle *v*.
- $IS_v = \Gamma_y$: Vehicles *v*'s *intersection state* is *Intersection-Wait*, and it is waiting for vehicle *y* to cross the intersection.
- $TIC_{v,y}$: Trajectory Intersecting Cell between vehicle *v* and vehicle *y*.
- ET_v*:* Vehicle *v*'s *Exit-Time* from the intersection.
- V_y : Current velocity of vehicle *y*.

The following rules are applicable to all vehicles:

Algorithm 1 AC-IP, Sender Vehicle

Input: Vehicle's *intersection state*
Output: Broadcast intersection safety message
if *STATE=Intersection-Approach* ***or*** *STATE=Intersection-Wait* **then**
    Broadcast *ENTER* message
else if *STATE=Intersection-Enter* **then**
    Broadcast *CROSS* message
else if *STATE=Intersection-Exit* **then**
    Broadcast *EXIT* message

And here are the rules applied to a vehicle B when it receives intersection messages from a vehicle A (A ≠ B).

Algorithm 2 AC-IP, Receiver Vehicle

Input: Safety message received from vehicle A, ***RM***
Output: Vehicle B's movement at the intersection
if *RM = ENTER* **then**
  Run CDAI to detect trajectory conflicts with vehicle A and find $TIC_{A,B}$
  if ($TIC_{A,B} = NULL$) **then**
    Cross the intersection
  else
    Run FCFS priority policy
    if ($P_B > P_A$) **then**
      Try to Cross the intersection
    else
      Slow down and call *Set Desired Speed*
else if *RM = CROSS* **then**
  Run CDAI to detect trajectory conflicts with vehicle A and find $TIC_{A,B}$
  if ($TIC_{A,B} \neq NULL$) **then**
    Slow down and call *Set Desired Speed*
  else
    Compete with other vehicles in the same situation*
else if *RM = EXIT* **then**
  if Intersection is cleared **then**
    Cross the intersection

The desired velocity is calculated to allow the vehicle to slow down in time and arrive at the intersection when the higher-priority vehicle is exiting the intersection. The vehicle will then accelerate and increase its speed to the maximum speed limit and cross the intersection area as fast as possible.

Algorithm 3 *Set Desired Speed*

Inputs:
Exit-Time of higher-priority vehicle A, $\boldsymbol{ET_A}$
Acceleration parameter of vehicle B, $\boldsymbol{AC_B}$
Deceleration parameter of vehicle B, $\boldsymbol{DC_B}$
Vehicle B's trajectory length through the intersection, $\boldsymbol{D_B}$

Output: Vehicle B's *Desired Speed*, $\boldsymbol{DS_B}$

Use the Newtonian equation for motion and calculate the *Desired Speed*

$$DS_B = DC_B * (ET_A - CurrentTime) + V_B$$

Update the *Exit-Time* of vehicle B

$$ET_B = (ET_A - CurrentTime) + \frac{-DS_B + \sqrt{DS_B{}^2 + 2*AC_B*D_B}}{2*AC_B}$$

To avoid any collisions inside the intersection area, the lower-priority vehicle will still be waiting to receive the EXIT safety message while it is slowing down and approaching the intersection. In the case that the vehicle does not receive the appropriate EXIT message and its distance to the entrance of the intersection is less than a threshold, it will come to a complete stop and wait for that message before accelerating and crossing the intersection box.

We now illustrate AC-IP with an example. Figure 4 shows a simple scenario in which vehicles A and B are approaching an intersection. Since vehicle A has a lower Arrival-Time than vehicle B, it has a higher priority based on our FCFS priority policy. Vehicle A is going to cross the intersection without stopping or slowing down. In contrast, vehicle B has to slow down and adjust its speed to arrive at the intersection when vehicle A is exiting it. When vehicle B arrives at the intersection and receives the EXIT safety message from vehicle A, it knows that the intersection area is safe and clear for its passage.

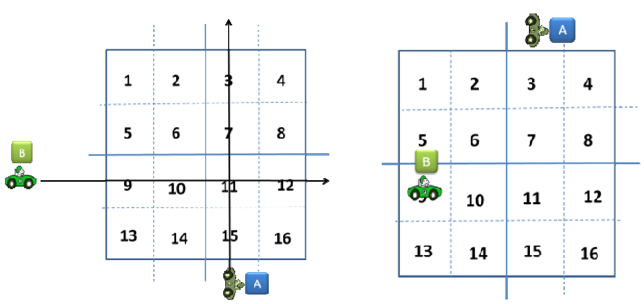


Figure4. An example scenario of AC-IP

If no potential collision has been detected with the sender of the CROSS message, the receiver may still not be allowed to cross the intersection area. The reason is that 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 concurrently without being aware that they may collide with each other.

The net result is that, 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, it can cross. This means that it can cross the intersection safely while broadcasting the CROSS message.

Figure 5 shows two junction 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. In Figure 5(a), vehicles B and C can cross at the same time as vehicle A, since none of them has a space conflict with the other two. As can be seen in Figure 5(b), vehicles B and C may collide as they have a conflicting trajectory along the intersection. Therefore, only one of them can safely cross through the intersection box while vehicle A is crossing. The other vehicle must slow down and enter the intersection box only after receiving the EXIT safety message from the higher-priority vehicle. In Assuming that vehicles B has a higher priority than and vehicle C according to the *priority policy*, it can cross, while vehicle C is slowing down to arrive at the intersection when vehicle B has exited it.

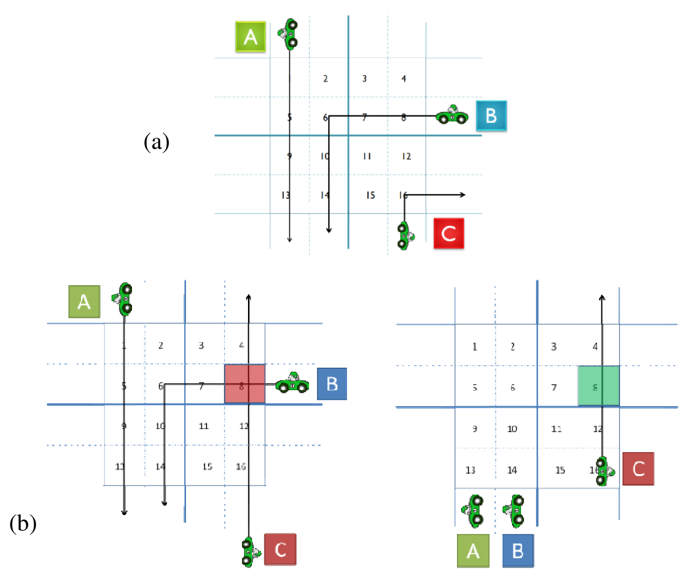


Figure5. Example scenarios of AC-IP at intersections

Figure 6 shows 2 roundabout scenarios. Assuming that vehicle A has the highest priority, it crosses the roundabout without slowing down or stopping in both cases. In the case of Figure 6(a), vehicles B and C can cross concurrently with vehicle A since none of them has a potential conflict with the other two. In Figure 6(b), even though vehicles B and C have no potential conflict with higher-priority vehicle A, they might get to a potential collision with each other. As vehicle B has a higher priority than vehicle C, it crosses the roundabout at the same time as vehicle A. Vehicle C slows down and set its speed to the *desired speed* to arrive at the roundabout only after the exit of higher-priority vehicles.

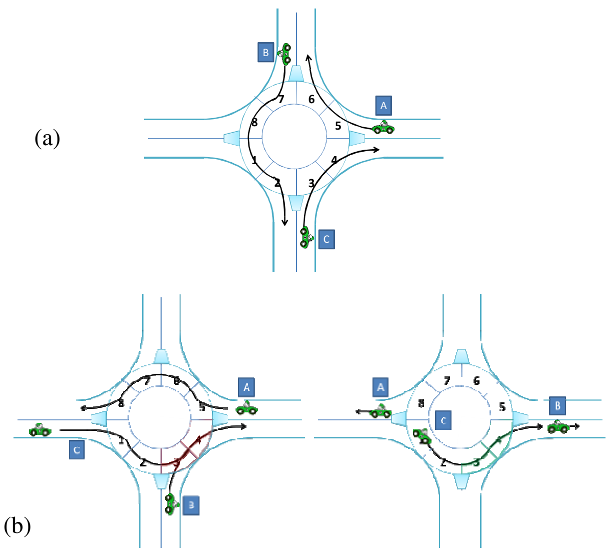


Figure6. Example scenarios of AC-IP at roundabouts

3.3.2 AC-IP Freedom from Deadlock

A deadlock is a situation in which two or more competing actions are each waiting for another to finish, and thus none ever does. A deadlock situation could occur inside the intersection area, among the vehicles which are trying to cross the intersection at the same time. To better capture such scenarios, we use wait-for graphs. A wait-for graph is a directed graph used for deadlock detection in operating systems and database systems. A deadlock exists if the graph contains any cycles.

We now investigate a possible deadlock scenario, in which all vehicles arrive at the intersection in very close time intervals. In Figure 7, vehicles A, B, C and D have all reduced their speeds and came to a complete stop at the intersection entrance. No vehicle is crossing the intersection to avoid potential collisions with other vehicles present on other legs of the cross-road.

We define the elements of our intersection wait-for graph as follows. Vehicles are represented as the nodes of our wait-for graph, and an edge from vehicle B to vehicle A implies the vehicle B is waiting for vehicle A, to complete its trajectory through the intersection grid. Since vehicle B is waiting at the intersection entrance for vehicle B, its updated *STATE* is *Intersection-Wait*. It can be seen clearly in Figure 8 that the corresponding wait-for graph contains a cycle and therefore it is a deadlock situation. We now show that under AC-IP, such deadlock cannot occur.

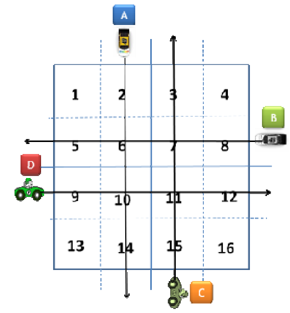


Figure7. A Deadlock Scenario

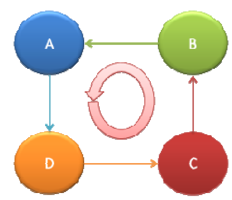


Figure8. Wait-for graph for an example deadlock scenario

Definition 1. ***Trajectory Dependency:***

Vehicle B's trajectory depends on vehicle A's trajectory at or near an intersection if and only if two conditions are true at the same time:

1) The priority of vehicle B is lower than the priority of vehicle A.

2) There is a common cell along their trajectory cells.

The above statement can be written as:

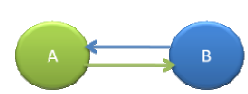
$$[(P_B < P_A) \text{ and } S_A \cap S_B \neq \emptyset] \Rightarrow B \to A$$

Rule 1. ***AC-IP Rule:***

If vehicle B's trajectory depends on vehicle A's trajectory, then vehicle B waits for vehicle A to cross the intersection and vehicle A does not wait for vehicle B to cross the intersection.

$$B \to A \Rightarrow IS_A \neq \Gamma_B$$

Theorem 1. AC-IP is deadlock-free.



Proof: We prove the theorem by contradiction. Suppose we have two potentially conflicting vehicles.

Please note that the deadlock situation happens only when these 2 vehicles have a common cell along their trajectories and they might get to a potential collision if they cross the intersection at the same time. Otherwise, both vehicles will safely cross the intersection simultaneously and no deadlock occurs.

The deadlock condition is as follows:

$$IS_A = \Gamma_B \text{ and } IS_B = \Gamma_A$$

Suppose that, for potentially conflicting vehicles A and B, we have: $P_B > P_A$.

$$S_A \cap S_B \neq \emptyset \quad (3)$$

Based on the Trajectory Dependency and AC-IP Rule, from Equation (3):

$$A \to B \Rightarrow IS_B \neq \Gamma_A \quad (4)$$

But from the deadlock conditions, we have

$$W_B = A \quad (5)$$

(3) and (4) cannot be true at the same time.
This is a contradiction. So $IS_B = \Gamma_A$ cannot be true while $IS_A = \Gamma_B$.

We now consider the deadlock situation with n vehicles, where $n > 2$. We must therefore have:

$$IS_B = \Gamma_A \text{ and } IS_C = \Gamma_B \text{ and } \dots IS_Z = \Gamma_Y \text{ and } IS_A = \Gamma_Z$$

Suppose that $P_A > P_B > \dots > P_Y > P_Z$.
Therefore for conflicting vehicles A and Z we have:

$$P_A > P_B > \dots > P_Y > P_Z \Rightarrow P_A > P_Z$$

$$S_A \cap S_Z \neq \emptyset \quad (6)$$

Based on the Trajectory Dependency and the AC-IP Rule, from Equation (6):

$$Z \to A \Rightarrow IS_A \neq \Gamma_Z \quad (7)$$

But the deadlock condition states that:

$$IS_A = \Gamma_Z \quad (8)$$

(7) and (8) are contradictory. So $IS_A = \Gamma_Z$ cannot be true while $IS_B = \Gamma_A \text{ and } IS_C = \Gamma_B \text{ and } \dots IS_Z = \Gamma_Y$. So, we conclude that deadlock is avoided by applying the AC-IP Rule. ■

We now apply AC-IP to the deadlock scenario of Figure 4. Figure 9 illustrates the behavior of vehicles under the AC-IP Rule. Vehicle A has the highest priority, and its trajectory does not depend on any other vehicle at the intersection, so it does not wait for the passage of other vehicles. Therefore, it crosses without stopping or slowing down. Since vehicle B and D's trajectories depend on vehicle A's trajectory, then, by the AC-IP Rule, these vehicles will not enter the intersection box and will wait for vehicle A. Vehicle C has no potential collision with the currently crossing vehicle, and it crosses the intersection. Then, vehicle B starts crossing and vehicle D starts its passage through the intersection concurrently with vehicle D. So, the deadlock situation is avoided due to the priority policy and the AC-IP rule.

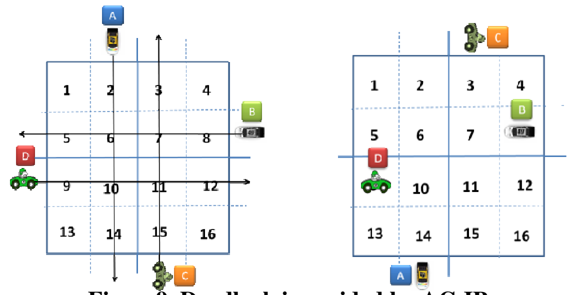


Figure9. Deadlock is avoided by AC-IP

3.3.3 Advanced Progression Intersection Protocol (AP-IP)

AP-IP is based on AMP-IP's key idea that conflicting vehicles can make concurrent progress inside the intersection grid when collisions can still be avoided. Additionally, AP-IP has the advantage of allowing the lower-priority vehicles to make smart speed decisions about crossing the conflicting cell.

The lower-priority vehicle's behavior will be determined based on various physical attributes and those of the higher-priority vehicle, such as their velocities, acceleration and deceleration parameters. The vehicle will make the appropriate decision for a safe passage through the intersection area. The lower-priority vehicle will pick one of the following actions, when it faces potentially conflicting scenarios:

1) Crosses the conflicting point and clears the trajectory intersection cell (TIC) before the arrival of the higher-priority vehicle to that cell.
2) Reduces its speed and arrives at the conflicting point when the higher-priority vehicle has cleared and exited that cell.

We now define the terms that will be used in AP-IP Algorithms.

- $TIC_{v,y}$: Trajectory Intersecting Cell between the higher-priority vehicle *v* and lower-priority vehicle *y*.
- $AT_{V,c}$: Arrival Time of vehicle V to cell c.
- $ET_{V,c}$: Exit Time of vehicle V from cell c.
- ϴ: The *Safety Time Interval*. To ensure the safe passage of both the potentially conflicting vehicles, we use a *Safety Time Interval* to increase the safety and make sure that the lower-priority vehicle has enough time to leave and clear the conflicting cell completely, before the arrival of the higher-priority vehicle to that cell.

The same rules as in Algorithm 1 apply to all sender vehicles. Please note that each vehicle is broadcasting its updated TCL information within the ENTER and CROSS safety messages. This information includes the estimated arrival and exit times of each cell along vehicle's trajectory through the intersection grid.

The following rules are applied to a vehicle B when it receives intersection messages from a vehicle A, where (A ≠B).

Algorithm 4 AP-IP, Receiver Vehicle

Input: Safety message received from vehicle A: ***RM***
Output: Vehicle B's movement at the intersection
if *RM* = *ENTER* or *RM* = *CROSS* **then**
  Run CDAI to detect trajectory conflicts with vehicle A and find $TIC_{A,B}$
  if ($TIC_{A,B} = NULL$) **then**
    Cross the intersection
  else
    Use FCFS priority policy
    if ($P_A < P_B$) **then**
      Cross the intersection
    else
      $c = TIC_{A,B}$
      if ($[AT_{B,c} + \theta] < AT_{A,c}]$) **then**
        Cross the $TIC_{A,B}$
      else
        Slow down and call *Set Desired Speed*
else if *RM* = *EXIT* **then**
  if $TIC_{A,B}$ is cleared **then**
    CROSS the intersection

Figure 10 shows an example scenario of vehicles following AP-IP rules. Vehicles A and B are approaching an intersection. Assume that vehicle A has a higher priority than vehicle B. Vehicle B will compare its arrival to the TCL cell number 11, to the exit time of the higher-priority vehicle A to the same cell. In the case that vehicle B arrives earlier and has enough time to clear the TCL before the arrival of vehicle A, it can progress and clear cell number 11. This behavior of the lower-priority vehicle will decrease the delay and increase the overall throughput of the intersection without sacrificing safety.

Figure 11 illustrates another scenario under AP-IP. In this case, the lower-priority vehicle B does not have enough time to progress and clear the TCL cell number 7, before the arrival of the higher-priority vehicle A. Therefore, vehicle B must adjust its velocity to prevent any potential collision at the TCL. It uses the information obtained by the received safety messages from vehicle A, its own digital map and GPS coordinates and physical model characteristics such as velocity, acceleration/deceleration parameters to determine the appropriate speed for approaching and progressing inside the intersection grid. The goal is to decrease the speed to arrive at the TCL right after the exit of the higher-priority vehicle from that cell. Please note that in the extreme case of slowing down would be getting to a complete stop before entering the TCL and waiting for the higher-priority vehicle to clear and exit the conflicting cell.

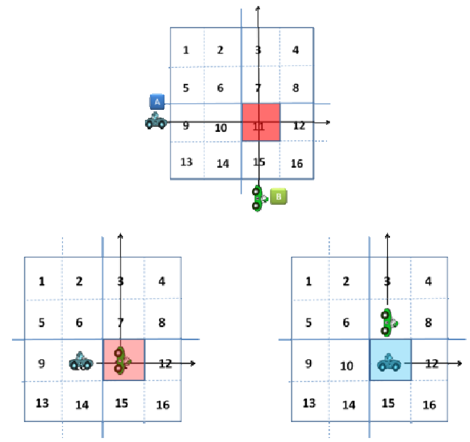


Figure10. An example scenario of AP-IP

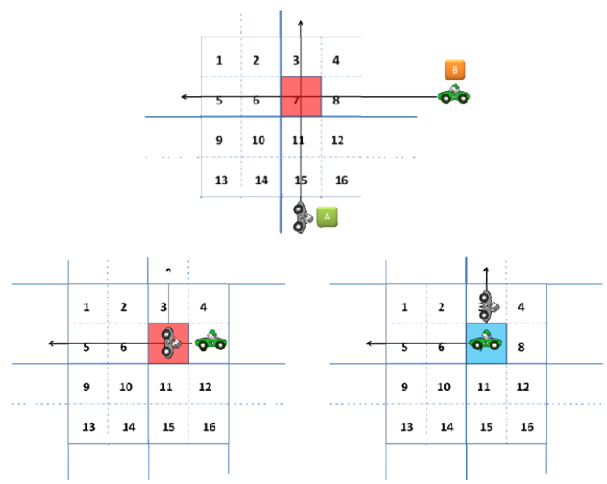


Figure11. An intersection example scenario of AP-IP.

Figure 12 shows how vehicles behave at the roundabout while following AP-IP rules. The higher-priority vehicle A will cross the roundabout without slowing down or stopping. The lower-priority vehicle B estimates its own arrival time to and vehicle A's exit time from the TIC, cell number 3. If vehicle B has enough time to clear that cell before the arrival of vehicle A, it will go ahead and cross it. Otherwise it will estimate the appropriate velocity and reduce its speed to the *desired speed* in order to arrive at the TCL exactly after the exit of vehicle A.

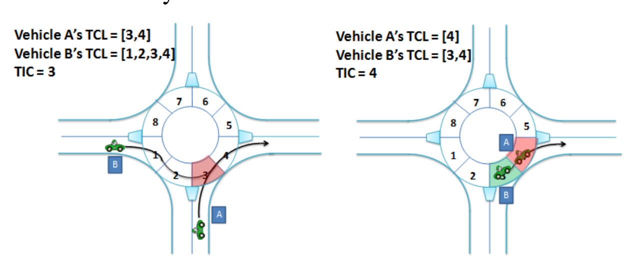


Figure12. A roundabout example scenario of AP-IP.

4. GPS POSITION INACCURACY

One critical issue for our intersection protocols is the position information accuracy provided by the on-board GPS devices. 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. These inaccuracies affect current position estimations as well as various distance measurements such as vehicle's distance to the intersection which determines its *intersection state*. The presence of large obstacles such as tall buildings at the corners of urban intersections increases the effects of multi-path as one of the main GPS error sources.

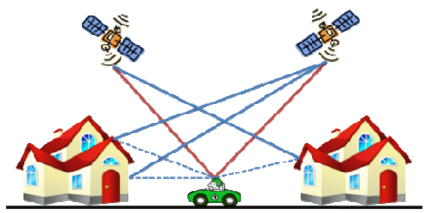


Figure13. GPS error due to multi-path

Different methods can be deployed to improve the position accuracy such as using high-accuracy Differential GPS (DGPS), Wide Area Augmentation System (WAAS), gyroscopes and local sensing. However, all GPS receivers have finite accuracy, with commonly-used inexpensive GPS receivers having errors of up to a few meters. Figure 14 shows the position accuracy comparison among three different types of GPS devices. We study the impact of such errors on our intersection protocols and propose a simple technique to overcome such inaccuracies.

GPS Device Type	Position Accuracy in meters
SA-deactivated	± 10 m
DGPS	± 3-5 m
WAAS	± 1-3 m

Figure14. GPS position accuracy comparisons

We have also implemented a V2V car-following model, in which each vehicle uses its GPS coordinates, map database and the information received in regular BSM messages to measure its current distance to the vehicle in front of it. The vehicle then adjusts its speed according to the leader vehicle's velocity to maintain a *safe distance*. This *safe distance* is measured based on the vehicle's physical characteristics such as acceleration/deceleration parameters and ensures that no accidents occur when the leader vehicle suddenly reduces its speed. Figure 15 shows a screen-shot from our hybrid simulator/emulator AutoSim, in which vehicle B is following vehicle A on its way to the intersection. In this scenario, due to a high position error, vehicle B may not maintain a *safe distance* to its current leader vehicle, leading to a potential collision before entering the intersection.

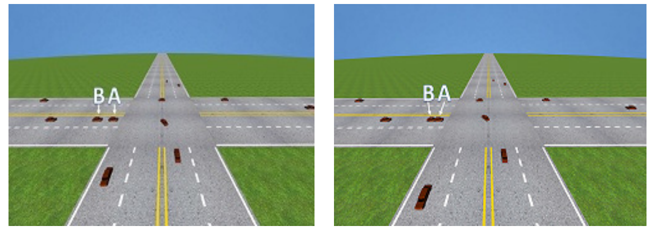


Figure15. Snapshots from AutoSim. Collision outside of the intersection area.

To avoid these collisions outside of the intersection grid, when dealing with high levels of positioning inaccuracy, each vehicle will use an updated *safe distance* parameter based on its GPS positioning error parameter. This increased buffer distance among following vehicles prevents vehicles from getting very close to each other and gives them the capability to slow down without causing an accident when the leader vehicle brakes suddenly.

The impact of position inaccuracy is more severe in High Concurrency Protocols with Slowdown (HCPS) of STIP protocols, since this group of intersection protocols explicitly utilizes information relating to a vehicle's progression inside the intersection area. Therefore, a failure in locating a vehicle's current cell information correctly may lead to vehicle collisions inside the intersection grid. As mentioned before, each vehicle is updating its TCL based on its current cell. In other words, the vehicle deletes the cleared cells from the Trajectory Cells List and only broadcasts the information about the current cell and next cells along its trajectory through the intersection area. However, due to the positioning error, the vehicle might update its TCL without having completely crossed its previous cell. Figure 16 shows a scenario in which a collision occurs between vehicles A and B. The higher-priority vehicle A is broadcasting an incorrect TCL within its CROSS safety message. As the lower-priority vehicle B receives the updated TCL from vehicle A and calculates that the conflicting cell is now clear, it will progress into that cell. As vehicle A is still occupying the conflicting cell, a potential collision occurs between vehicles A and B.

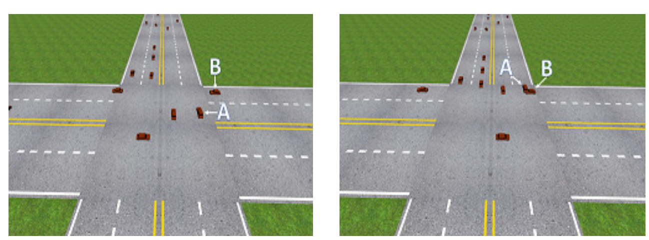


Figure16. Snapshots from AutoSim. Collision inside the intersection area.

To avoid these safety violations, each sender vehicle adds a safety cell to its updated TCL. The TCL now includes the previous cell as well as the current and the next cells of vehicle's trajectory inside the intersection grid. Thus, we add a safety buffer of one intersection cell ahead of and prior to the current cell to assure the safe passage of the vehicle. The size of this safety cell buffer should be a function of the GPS error characteristics and, if the GPS inaccuracy is too high, then collisions can be avoided by increasing the buffer size to more than one cell. However, this guaranteed safety comes with the price of reduced throughput at the intersection. The receiver vehicle can also assure its safety by increasing the value of *Safety Time Interval* according to vehicle's GPS position accuracy. The default value of *Safety Time Interval*, which is calculated based on the vehicle's passage through one intersection cell, can be changed and adopted to the position accuracy level.

Each vehicle computes its own Trajectory Cells List using the digital map database, and it uses its GPS coordinates to determine its current cell. The reader may observe correctly that position inaccuracy might also affect the vehicle's ability to correctly determine its lane information. This can be avoided by using local sensing technologies (available on autonomous vehicles) such as cameras and lasers to perform *lane localization*.

5. IMPLEMENTATION

In this section, we describe the implementation of the V2V protocols, the GPS model and other models such as the V2V communication model. In order to analyze our intersection protocols and the effects of position inaccuracy on them, the GPS model and the traffic flow at intersections need to be studied. For this purpose, we use a tool called AutoSim. This simulator-emulator is a next-generation version of GrooveNet [7,8], with 3-D graphics and other capabilities.

AutoSim has real-time emulation capability wherein real and simulated cars can co-exist and interact with each other. The communication interfaces for DSRC communication as well as peripheral sensory interfaces are implemented to enable real cars instrumented with DSRC to react in real-time with simulated cars. The communication protocol uses Basic Safety Messages (BSM) [3] that are broadcast as part of the WAVE mechanism.

Figure17. AutoSim enables communication between real and simulated vehicles.

Main modules of AutoSim are mobility, controller, communication and map models. A new *GPS model* has been designed to inject position inaccuracy into the GPS coordinates used by each vehicle. The level of positioning error is configurable, and will affect the decisions made by vehicles at and around intersection areas.

We have used real GPS coordinates from map databases to generate Route Network Definition File (RNDF) files for roundabouts which already exist in the USA. Additionally, we have designed new mobility models to implement our V2V-intersection protocols within the roundabout area.

6. EVALUATION

In this section, we evaluate our new V2V intersection protocols and the effects of position inaccuracy on them. For this purpose, we use various mobility, communication and GPS models that we have designed and implemented in our hybrid emulator-simulator AutoSim.

6.1 Metric

We define the *trip time* for a vehicle, as the time taken by that vehicle to go from a fixed start-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. In our simulations, the traffic generation follows the Poisson random distribution. We run first set of our simulations on 4-lane roads, with 2 lanes in each direction. Each simulation run uses 1000 vehicles. We will later use roundabouts and intersections with a single lane.

6.2 Experiments

Figure 18 shows the comparison among the traffic light with green light duration 30 seconds and 10 seconds and our new V2V protocols, Advanced Cross-Intersection Protocol (AC-IP) and Advanced Progression-Intersection Protocol (AP-IP). The X-axis marks the traffic volume in vehicles per hour and the Y-axis is the *Trip Delay* in seconds.

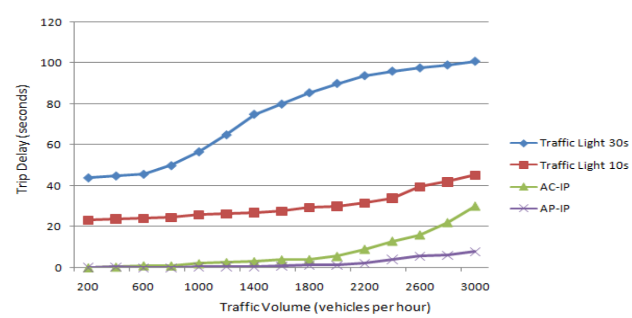


Figure18. Trip delay comparison of different mobility models

We can see that our intersection management models significantly outperform the current technology. AC-IP improves throughput by up to 79.92% and 64.12%

respectively, compared to the traffic light models of 30s and 10s green lights. AP-IP outperforms the traffic light model of 30s, 10s and AC-IP by respectively 92.51%, 87.82% and 72.26%.

Figures 19 and 20 present the results for a perfect-cross intersection where vehicles are following AC-IP and AP-IP rules respectively. Our results show that as due to the modifications for higher GPS inaccuracies and increased safety parameters, our new V2V intersection models have expected lower throughput compared to their counterparts with lower-inaccuracy and perfect GPS assumption.

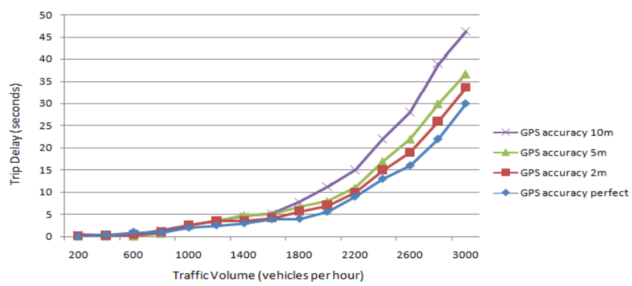


Figure19. Trip delay comparison of AC-IP under different GPS position accuracies

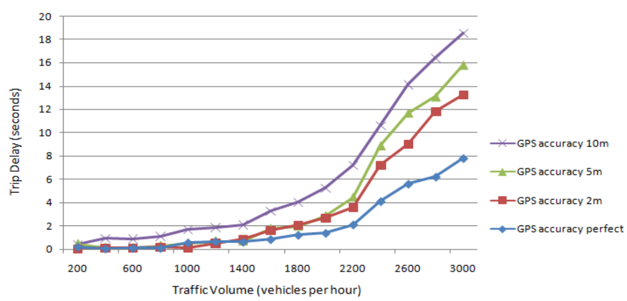


Figure20. Trip delay comparison of AP-IP under different GPS position accuracies

Figure 21 shows that, despite the reduced throughput of modified V2V intersection models, AC-IP and AP-IP have 47.82% and 74.16 % overall performance improvements respectively over the traffic light model with a 10-second green light time. AP-IP outperforms AC-IP by 50.48%. These are still significant benefits.

We have logged the statistics for all simulated vehicles such as their position information at any moment while crossing the intersection. This information has been used to log any accidents among the vehicles trying to concurrently pass through the intersection area. Our simulation results show, that due to modification in our protocols as of the safety parameters, no accidents happen in any tested traffic volumes at the intersection when dealing with high levels of GPS position inaccuracies. We therefore conclude that our proposed intersection protocols support safe traversal through intersections at substantially higher throughput even with imperfect and commonly-used GPS devices.

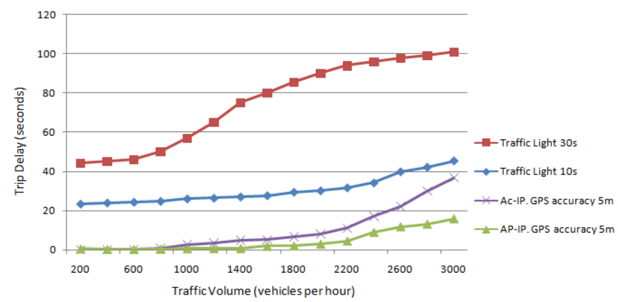


Figure21. Delay comparison among different mobility models

We have also studied the *trip delays* of a 1-lane roundabout where only 1-lane traffic is entering the roundabout from four directions. Vehicles follow the Advanced Cross Intersection Protocol (AC-IP) or Advance Progression Intersection Protocol (AP-IP) while crossing the roundabout area. We then replaced this roundabout by a 1-lane signalized intersection which is managed by the traffic light models. Figure 22 shows the comparison between the above models for traffic volumes range of 200 to 2000 vehicles per hour.

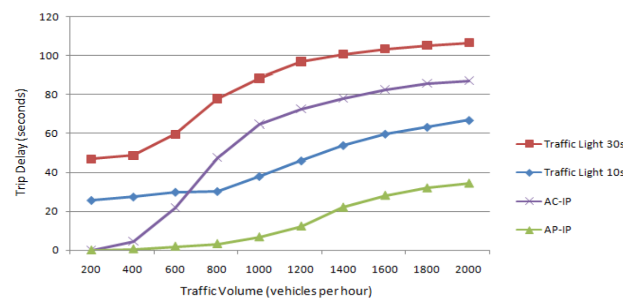


Figure22. Delay comparison for 1-lane roundabout and 1-lane signalized intersection

Our results show that AP-IP performs significantly better than the traffic light models. AP-IP decreases the *trip delays* by 83.06% and 67.98% respectively compared to the traffic light models with 30 seconds and 10 seconds of green light duration. AC-IP does not perform well under higher traffic volumes. The reason is that lower-priority vehicles are not allowed to enter the roundabout area in the case of a potential conflict, and they must slow down and enter the roundabout grid only after receiving EXIT safety message from the higher-priority vehicle. In the case of higher traffic volumes, this behavior results in longer stops before entering the roundabout. In contrast, AP-IP allows more vehicles to use the gaps among vehicles and progress inside the roundabout gird and cross concurrently.

We have performed the same test using a 2-lane roundabout, in which traffic is entering from 2-lanes in each direction. Figure 23 illustrates the trip delays when a 2-lane roundabout is ruled under AC-IP and AP-IP. It also show the results when the same roundabout is replaced by a signalized perfect cross-road which is managed by traffic light models.

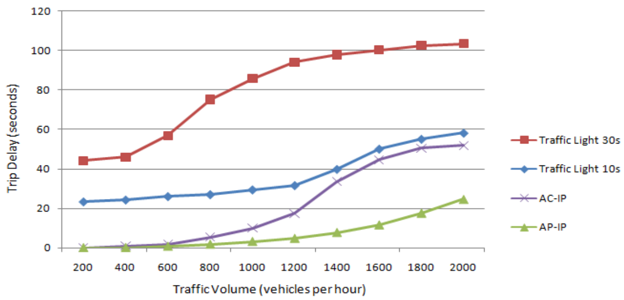


Figure23. Delay comparison for 2-lane roundabout and 2-lane signalized intersection

Our results indicate that our V2V intersection protocols are significantly outperforming the traffic light models. AP-IP has very negligible delay when dealing with low and medium-volume traffics. AP-IP outperforms the traffic light models with 30 seconds and 10 seconds of green light duration, respectively by 91.04% and 80.21%.

7. CONCLUSION and FUTURE WORK

The future of road transportation is expected to include autonomous vehicles. An important segment of urban transportations is intersections and cross-roads. Current technologies such as traffic lights and stop signs are not suitable for autonomous driving. They are not very safe in managing the through traffic and inject significant delays due to their inefficiency. We have designed a new generation of V2V-based intersection protocols which significantly increase the throughput of the intersections and avoid collisions. In this paper, our goal was to design new intersection protocols which can manage the traffic through junctions and roundabouts, while maintaining safety and improving throughput. This new generation optimizes vehicle speed and dynamics to improve throughput. We have also studied the effects of GPS position inaccuracies on our V2V intersection protocols by implementing realistic GPS models. Although our protocols are designed for autonomous vehicles that use V2V communication for co-operative driving in future intelligent transportation systems, they can be adapted to a driver-alert system for manual vehicles at traffic intersections. Local sensing technologies such as cameras and lasers can be combined with V2V and V2I communications to avoid any potential collisions and enhance localization accuracy. In our future work, we will investigate the use of these combined technologies and study the ways that they can benefit our intersection management protocols. We want to achieve higher traffic throughput even when dealing with inaccurate GPS devices, without sacrificing our main goal of safe passage of vehicles through intersections and roundabouts.

8. ACKNOWLEDGMENTS

This work is sponsored by the General Motors and Carnegie Mellon University Collaborative Research Agreement.

9. REFERENCES

[1] The 2011 urban mobility report, published by the Texas transportation institute.
DOI: http://mobility.tamu.edu/ums/.

[2] DARPA. The DARPA urban challenge, 2007.
DOI: http://www.darpa.mil/grandchallenge.

[3] Draft SAE j2735-200911 dedicated short range (DSRC) messages et dictionary.

[4] Us department of transportation-federal highway administration publication, national agenda for intersection safety.
DOI:http://safety.fhwa.dot.gov/intersection/resources/intersafagenda/.

[5] Washington State, Department of Transportation. http://www.wsdot.wa.gov/Safety/roundabouts/benefits.htm

[6] Michigan Department of Transportation. http://www.michigan.gov/roundabouts

[7] R. Mangharam, D. S. Weller, R. Rajkumar. Mudalige and Fan Bai. GrooveNet: A hybrid simulator for vehicle-to-vehicle networks.

[8] R. Mangharam, J. Meyers, R. Rajkumar, D. Stancil, J. Parikh, H. Krishnan, and C. Kellum. A multi-hop mobile networking test-bed for Telematics. April 2005.

[9] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Vehicular networks for collision avoidance at intersections, society for automotive engineers (SAE) world congress. April 2011.

[10] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Intersection management using vehicular networks, society for automotive engineers (SAE) world congress. April 2012.

[11] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Reliable intersection protocols using vehicular networks, ACM/IEEE 4thinternational conference on cyber-physical systems. April 2013.

[12] Hummer, J.E., Chapter 14: Intersection and Interchange Design, in Handbook of Transportation Engineer. p. pp. 14.1–14.27, 2004.