Dynamic Intersections and Self-Driving Vehicles

Conference Paper · April 2018

DOI: 10.1109/ICCPS.2018.00038

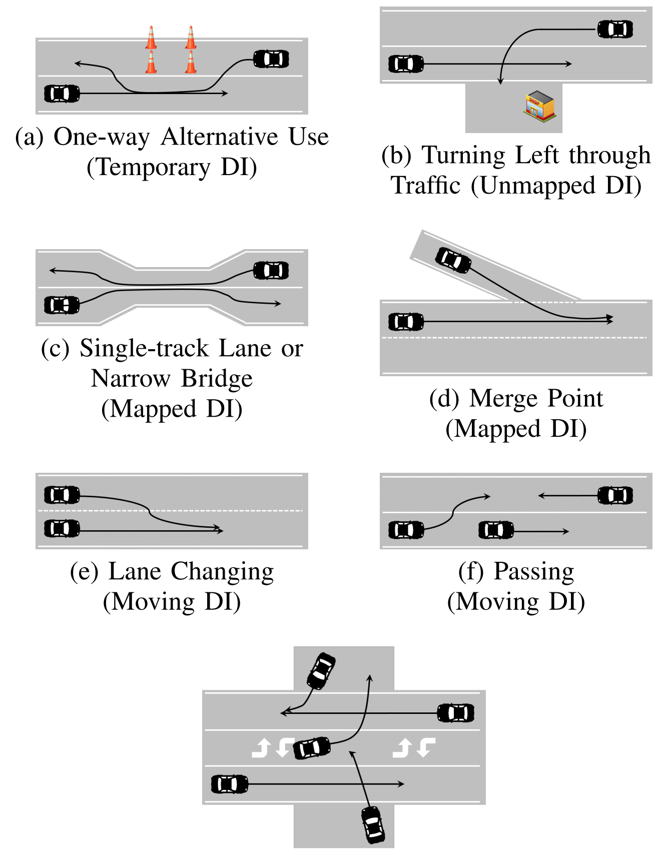
Shunsuke Aoki
Carnegie Mellon University

Dynamic Intersections and Self-Driving Vehicles

Shunsuke Aoki and Ragunathan (Raj) Rajkumar
Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, PA
Email: shunsuka@andrew.cmu.edu, raj@ece.cmu.edu

***Abstract*—Connected and automated vehicles are expected to be at the core of future intelligent transportation systems. One of the main practical challenges for self-driving vehicles on public roads is safe cooperation and collaboration among multiple vehicles when conflicts arise on shared road segments. Intersections controlled by traffic lights and stop signs are common examples of such potential conflicts, and cooperative protocols for such intersections have been studied. On the other hand, there are many different types of shared road segments. In this paper, we study *Dynamic Intersections* that might appear almost anytime and anywhere on public roads and that might lead to automobile accidents. We consider how a self-driving vehicle can safely navigate these dynamic intersections by using sensor-based perception and inter-vehicle communications. We present a cooperative protocol for dynamic intersections which can be used by self-driving vehicles for safely coordinating with other vehicles. Under our protocol, self-driving vehicles can also create a vehicular communication-based traffic manager named *Cyber Traffic Light* when the area is congested. A cyber traffic light functions as a self-optimizing traffic light by estimating the traffic volumes and by wirelessly coordinating among multiple self-driving vehicles. Our simulation results show that our protocol has higher traffic throughput, compared to simple traffic protocols while ensuring safety.**

***Index Terms*—Autonomous vehicles; Intersection management; Vehicular networks; Intelligent transportation systems**



(a) One-way Alternative Use (Temporary DI)

(b) Turning Left through Traffic (Unmapped DI)

(c) Single-track Lane or Narrow Bridge (Mapped DI)

(d) Merge Point (Mapped DI)

(e) Lane Changing (Moving DI)

(f) Passing (Moving DI)

(g) Center Turn Lane (Variegated DI)

Figure 1: Example of Different Types of Dynamic Intersections.

I. Introduction

With advances in Cyber-Physical System (CPS) technologies, connected and automated vehicles are becoming increasingly feasible. Many companies are testing self-driving vehicles on public roads [12], while academic research has produced the earliest prototypes, such as the CMU self-driving Cadillac SRX [19].

One of the main practical challenges for self-driving vehicles on public roads is safe cooperation and collaboration among multiple vehicles when conflicts arise on shared road segments. Intersections controlled by traffic lights, roundabouts, and stop signs are common examples of such potential conflicts. We refer to these road intersections as "stationary intersections" since they are captured in map databases, are known a priori, do not move, and last for a very long time. Cooperative protocols for such intersections have been studied, for example in [8], [10].

In this paper, we identify and classify *Dynamic Intersections* that might appear almost anytime and anywhere on public roads leading to potential dynamic conflicts that must be safely resolved. We then consider how a self-driving vehicle can safely navigate these dynamic intersections using a combination of sensor-based perception and inter-vehicle communications. A dynamic intersection (a) represents a shared region on the road where traffic conflicts can arise, (b) may be temporary, (c) may arise dynamically due to the occasional presence of vehicles, and (d) is not included in a map database (unlike a multi-way road intersection or a roundabout).

In Figure 1, we present 7 examples of different types of dynamic intersections: (a) One-way Alternative Use due to construction; (b) Turning Left through Traffic into a parking lot or driveway; (c) Single-track Lane or Narrow Bridge; (d) Merge Point; (e) Lane Changing; (f) Passing on a lane usually used by oncoming traffic; and (g) Center Turn Lane. In a dynamic intersection, a road segment must be shared by two or more vehicles whose travel paths intersect. Other examples of dynamic intersections occur in large parking lots, in multi-level parking garages, in areas with disabled or crashed vehicles, on roads with double parked vehicles blocking a lane, along damaged roads, or near potholes. Human drivers navigate these regions using social norms, courtesy, hand gestures, eye contact, and common sense. However, for self-driving vehicles, to guarantee safety, real-time interaction and negotiation are essential.

In this paper, we present a cooperative traffic protocol for safe traversal through many kinds of dynamic intersections. In this protocol, each self-driving vehicle uses both Vehicle-to-Vehicle (V2V) communications and internal sensor-based perception systems. Our protocol avoids vehicle collisions and possible deadlocks by using V2V communications, and guarantees safety by using sensor-based perception systems. Under the protocol, the self-driving vehicles also create a V2V communication-based traffic manager named *Cyber Traffic Light* (CyberTL) when traffic is congested to provide fairness to vehicles waiting for a long time. The CyberTL works as a self-optimizing traffic light at a dynamic intersection, by locally estimating the traffic volume and wirelessly coordinating among the vehicles. In other words, the CyberTL provides high traffic throughput around dynamic intersections, while keeping the worst-case waiting time short.

The contributions of this paper are as follows.

1) We define and classify dynamic intersections that might appear almost anytime and anywhere.
2) We present a cooperative protocol for guaranteeing safety at dynamic intersections.
3) We evaluate our traffic protocol using a simulator-emulator, and demonstrate the superior performance of our protocol.

The remainder of this paper is organized as follows. Section II describes our system assumptions, and presents a classification of dynamic intersections. Section III presents our cooperative dynamic intersection protocol along with examples to illustrate its safety and usage. In Section IV, we evaluate our protocol and compare its performance against traditional traffic protocols. Section V discusses previous work related to our research. Finally, Section VI presents our conclusions and future work.

II. Assumptions

In this section, we present our system architecture and a classification of dynamic intersections. We also provide a review of the existing traffic control methods for human-driven cars.

A. Vehicle Systems

We assume that a self-driving vehicle comprises various hardware and software components, such as a map database, a navigation system, a perception system, an autonomous vehicle controller, a localization capability, and a wireless communication interface [19].

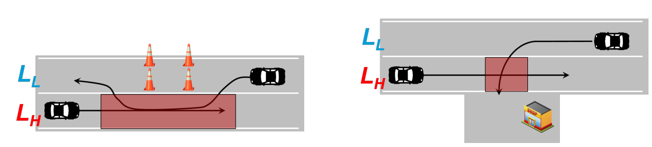
The map database contains accessible lane information, any turn restrictions associated with each lane, and the geographical layout of stationary intersections. The database also contains meta-information about stationary road intersections like roundabouts and whether they are controlled by physical traffic lights or stop signs. However, it does not contain all the possible collision points and dynamic intersections on public roads. The navigation system works with the map database and allows passengers to input, plan, and display a route to be taken. The perception system senses the surrounding environment by fusing data from different types of sensors, such as radar, LIDAR, and/or vision cameras. Sensor fusion combines the radar and LIDAR data, and makes them more consistent. The autonomous vehicle controller takes as its input the route generated by the navigation system, and actuates the vehicle to traverse the chosen path. In addition, each vehicle is equipped with a high-accuracy Global Positioning System (GPS) receiver. This receiver provides the vehicle with both localization and time-synchronization services. Reasonable positioning accuracy is important for self-driving vehicle operation with the required level of accuracy potentially achieved by Differential GPS techniques and/or correction services such as Real-Time Kinematics (RTK) [14]. Time synchronization among vehicles is obtained from GPS and its PPS clock source. Finally, the wireless interface enables the system to transmit and receive information wirelessly using one or more mechanisms, such as 4G/5G cellular interfaces, satellite communications, or an embedded infrastructure. In this paper, we assume that each self-driving vehicle has an On-Board Unit (OBU) that supports Dedicated Short-Range Communications (DSRC) and the Wireless Access in a Vehicular Environment (WAVE) protocol stack [9], [13]. This OBU is used to exchange data among multiple automated vehicles. The OBU also performs the functions of a *Cyber Traffic Light* (See Section III-D) by sending approval or rejection to the requests sent by other surrounding vehicles.

To simplify our presentation, we assume that each vehicle has good GPS reception, and all system components are functional. We will discuss the safety assurance provided by our protocol in Section III-E, where we show that packet losses do not compromise safety.

B. Dynamic Intersections

A dynamic intersection is a road area shared by two or more vehicles whose travel paths intersect, giving rise to conflict points. Moreover, this area is not controlled by traffic lights, stop signs, or roundabouts. We first give the common features of these dynamic intersections, and then, we classify different types of dynamic intersections.

At a dynamic intersection, (i) vehicles can enter from more than one direction; (ii) no physical traffic light, sign, or roundabout is used to manage traffic; (iii) priorities may be assigned to approaching directions; and (iv) the map database does not store the presence of the intersection. Under the existing traffic system, human drivers navigate dynamic intersections by following priority rules, by communicating



(a) One-way Alternative Use (b) Turning Left through Traffic

Figure 2: Dynamic Intersection area and Lane priority.

with each other, by making eye contact or using hand signals, and/or by accounting for the surrounding environment.

Each dynamic intersection contains at least one *collision point* (or *conflict point*). The collision point is the spatial point where two vehicles might collide because their travel paths intersect.

We classify dynamic intersections into at least 5 different types. These are presented in Figure 1: **Temporary Dynamic Intersection**, **Unmapped Dynamic Intersection**, **Mapped Dynamic Intersection**, **Moving Dynamic Intersection**, and **Variegated Dynamic Intersection**. We describe these types below.

Temporary Dynamic Intersection (Temporary DI): This road segment is temporarily used by vehicles coming from two different directions due to construction, traffic control, or blocking obstacles, as shown in Figure 1-(a). The detailed information about the intersection might not be included in the map database.

Unmapped Dynamic Intersection (Unmapped DI): An Unmapped DI is not stored as a road intersection in the map database, but there is a possible collision point when vehicles turn left through oncoming traffic. This type of dynamic intersection typically appears in front of the entrance of a parking area, grocery store, restaurant, home, and so on, as shown in Figure 1-(b).

Mapped Dynamic Intersection (Mapped DI): A Mapped Dynamic Intersection may be stored in some map databases as a collision point, such as a Single-track Lane and a Merge Point, as shown in Figure 1-(c) and (d), respectively. The collision point is typically not managed by traffic lights.

Moving Dynamic Intersection (Moving DI): A Moving Dynamic Intersection appears when the vehicle tries to change lanes, for lane-changing maneuvers and for passing maneuvers, as shown in Figure 1-(e) and (f). When a vehicle cannot start the maneuver, the vehicle might wait for a few seconds while moving forward and try again. Hence, the *spatial area* of the dynamic intersection moves.

Variegated Dynamic Intersection (Variegated DI): A Variegated Dynamic Intersection represents an area having multiple dynamic intersections, such as a Center Turn Lane. A Center Turn Lane might contain multiple dynamic intersections for turning left through traffic and merging right. Such an intersection is shown in Figure 1-(g).

Inter-vehicle interaction and negotiation are essential for self-driving vehicles at dynamic intersections to ensure safety and enable higher throughput. We recommend the use of Vehicle-to-Vehicle (V2V) communications and local perception at these intersections. Vehicle-to-Infrastructure (V2I) communications to handle these intersections is not practical due to financial and practicality considerations in our view.

In the rest of this paper, without loss of generality, we will use the dynamic intersection shown in Figure 2 to present, illustrate, and evaluate our protocol. This helps us simplify the presentation and adhere to page limits.

In our protocol, we will use the priority assigned to lanes L_H and L_L, as shown in Figure 2. This priority assignment follows the existing traffic rules for human-driven vehicles. Around dynamic intersections, the vehicles on lane L_H have precedence, and the vehicles on lane L_L have to yield in the presence of vehicles on lane L_H.

C. Existing Traffic Control Methods

In this section, we discuss existing traffic control methods for the intersections we see in practice. First, traffic lights are widely used at stationary road intersections. In this method, vehicles can enter the intersection when the corresponding light is green. To enhance traffic throughput, a feasible single timing plan is often used. In the most common approach in the US [1], the effective cycle length is estimated, and the cycle length is split into a green time segment for each phase. The calculation for the effective cycle length C and green time g is described in Eqs. (1) and (2).

$$C = \frac{L}{1 - \frac{CS}{RS}} \tag{1}$$

$$g = \frac{CV}{CS} \times (C - L) \tag{2}$$

Here, L is the total lost time, usually taken as the sum of the inter-green periods. CS represents the critical volume that is the sum of the critical phase volume for each lane. Also, RS represents the reference sum of phase flow rates representing the theoretical maximum value that the intersection could accommodate. CV is the critical volume for a phase. (Our Cyber Traffic Light to be described in Section III-D will be based on this scheme.)

Next, a different traffic management scheme is used in construction zones. According to the USDOT [2], there are several different approaches to manage vehicles around a work zone: (i) Flagger Method, (ii) Temporary Traffic Signal Method, and (iii) Stop or Yield Control Method. In the flagger method, qualified human flaggers manage traffic around the work zone. In the temporary traffic signal method, traffic signals are temporarily added and used to control traffic movements. Signal period and timing are often manually set beforehand. In the stop and yield method, the side that is closed should yield or stop for oncoming traffic from the open side. We will compare these schemes against our traffic protocol in Section IV.

III. Our Cooperative Dynamic Intersection Protocol

In this section, we present our cooperative vehicle protocol for dynamic intersections. The protocol is designed to avoid vehicle collisions at dynamic intersections while maintaining traffic throughput.

First, we use *dynamic vehicle states* to guide the behaviors and the communications to be used by each vehicle. Each vehicle is in a vehicle state Φ_{state} that dynamically transitions, and is used to command the vehicle to navigate the dynamic intersection while accounting for the surrounding environment, congestion, and waiting time.

Secondly, to assure safety, our protocol uses *request-response-based negotiation* around the dynamic intersection. We assume that there are two neighboring lanes, L_H and L_L, with the lane having the dynamic intersection being L_H. All negotiations are initiated from the vehicles on lane L_L with their requests. The vehicles on lane L_H just respond to the received requests.

Thirdly, we will use a *Cyber Traffic Light* (CyberTL)[1] which works as a self-planning, self-organizing, and self-optimizing traffic manager for a dynamic intersection when congestion arises. Once a CyberTL allocates the green period for vehicles coming from the two directions, other surrounding vehicles follow the time allocation and traverse the dynamic intersection within the allocated green period.

Our protocol uses V2V communications and the perception systems on each vehicle. All negotiations among multiple self-driving vehicles is conducted using vehicular communications. The perception systems are used to improve the assurance of collision-free traversal through the dynamic intersection.

Self-driving vehicles use Advanced Safety Messages for V2V communications, and we use the second optional part of Basic Safety Messages (BSM) for communicating protocol information [13]. Therefore, Advanced Safety Messages are broadcast at 10 Hz, the same rate as the BSM.

Unlike human-driven vehicles, self-driving vehicles using our protocol do not have to stop before a dynamic intersection unless necessary for safety reasons. Therefore, as will be seen in Section IV, the traffic throughput of our protocol is much better than that of existing schemes.

A. Detection of Dynamic Intersections

By definition, conflict points exist at a dynamic intersection. A self-driving vehicle can detect the presence of a dynamic intersection using its map database showing other vehicle paths that intersect with its own when making turns, and/or by detecting the presence of anomalies like lane closures or work zones. Our protocol will then be initiated by the vehicle looking to intervene and use the dynamic intersection.

B. Message Set

In our protocol, each vehicle uses 6 types of safety messages to interact with other vehicles within its communication range.

DI REQUEST: A DI REQUEST message indicates that the vehicle has a dynamic intersection along its path. The message contains 5 parameters: Location of the dynamic intersection, Estimated Arrival Time, Estimated Exit Time, and Current Lane.

DI APPROVE: A DI APPROVE message is used by a vehicle to respond to the DI REQUEST message in order to acknowledge the requested maneuver of the vehicle on the other lane. The message contains Location of the dynamic intersection, Estimated Arrival Time, and Estimated Exit Time.

DI INTERRUPT: A DI INTERRUPT message is used to potentially stop the vehicles coming from the other lane. The message is used to accomplish Cyber Traffic Light functions. The message contains Location of the dynamic intersection, Cycle Length, and Allocated Green Periods.

DI YIELD: A DI YIELD message is used to respond to the DI INTERRUPT message, and it indicates that the transmitter vehicle allows the receiver to enter the dynamic intersection. The YIELD message contains Location of the dynamic intersection, Cycle Length, and Allocated Green Periods.

DI DECLINE: A DI DECLINE message is used for the response to the DI REQUEST message and the DI INTERRUPT in order to decline the requested maneuver of the vehicle on the neighboring lane.

DI CROSS: A DI CROSS message indicates that the transmitter vehicle is inside the dynamic intersection. This message contains Location of the dynamic intersection and Estimated Exit Time.

The **DI CROSS** message is not absolutely required because the perception system may be sufficient to assure the safety of the dynamic intersection, but it is used in order to enhance the reliability and safety of our protocol.

C. Vehicle State Transitions

We now present our cooperative dynamic intersection protocol for vehicles to navigate themselves safely through a dynamic intersection. To maintain high traffic throughput while satisfying safety requirements and keeping the worst-case waiting time at a dynamic intersection short, vehicles use the state transition diagram shown in Figure 3.

As shown in Figure 3, there are 8 vehicle states. Vehicles on lane L_L can enter the 3 states with a dark blue border. Also, vehicles on lane L_H can enter the other 3 states. Any vehicles not near the dynamic intersection are in the *Not Around DI* state. When the vehicle approaches the dynamic intersection, its state transitions to the *Approaching DI* state. Also, when the vehicle receives the **DI REQUEST** packet, its state is changed from the *Not Around DI* state to the *Queried* state. The behaviors of the vehicles within each state are presented next.

The vehicle state transitions of our dynamic intersection protocol are captured in Algorithm 1 and Algorithm 2. Here, Algorithm 1 presents the protocol for the vehicle on lane L_L, and Algorithm 2 presents the protocol for the vehicle on lane L_H.

Now, we present the other notation that we use in this paper.

- τ_H^t: Allocated green period calculated at time t for the lane L_H.
- τ_L^t: Allocated green period calculated at time t for the lane L_L.
- τ_{max}: Maximum value for allocated green periods, τ_H^t and τ_L^t.

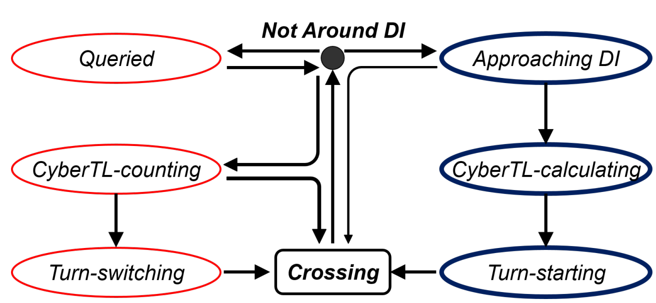


Figure 3: State Transition used by Vehicles at Dynamic Intersections.

[1]The name Cyber Traffic Light implies that this traffic light is **NOT** physical, but only exists as a software abstraction at the dynamic intersection.

Algorithm 1: Protocol for vehicle on lane L_L

while *DI is in the near future path* **do**
  if *Not halting* **then**
    Φ_{state} = *Approaching DI*;
  else
    Measure *wait time*;
    if *wait time* $> \tau_H^t$ & Φ_{state} = *CyberTL-calculating* **then**
      Φ_{state} = *Turn-starting*;
    else if *wait time* $> \delta$ & Φ_{state} = *Approaching DI* **then**
      Φ_{state} = *CyberTL-calculating*;
    else
      Φ_{state} = *Approaching DI*;
      Keep halting before the Dynamic Intersection;
    end
  end
end

Algorithm 2: Protocol for vehicle on lane L_H

if DI REQUEST *is received* **then**
  Φ_{state} = *Queried*;
else if DI INTERRUPT *is received* **then**
  if *cannot stop safely* **then**
    DI DECLINE is sent;
  else
    Halt before the DI;
    Φ_{state} = *CyberTL-counting*;
  end
else
  Φ_{state} = *Not Around DI*;
end

- τ_{min}: Minimum value for allocated green periods, τ_H^t and τ_L^t.
- δ: Time threshold for state transition from *Approaching DI* state to *CyberTL-calculating* state.
- L_{DI}: Total lost time.
- C_{DI}: Cycle length.
- D_{DI}: Length for the Dynamic Intersection.
- l_v: Vehicle length.
- v_{DI}: Permitted velocity around the Dynamic Intersection.
- s_0: Base saturation flow rate.
- f_a: Area type adjustment factor.
- N_H: Number of vehicles coming from the lane L_H.
- N_L: Number of vehicles coming from the lane L_L.
- v_H: Average velocity of moving vehicles on the lane L_H.
- v_L: Average velocity of moving vehicles on the lane L_L.
- D_H: Distance to the farthest vehicle on the lane L_H.
- D_L: Distance to the farthest vehicle on the lane L_L.
- ψ_H: Estimated traffic volume for the lane L_H.
- ψ_L: Estimated traffic volume for the lane L_L.
- $ST(\tau_H^t)$: Switching time for the phase allocated τ_H^t.
- $ST(\tau_L^t)$: Switching time for the phase allocated τ_L^t.

Here, the values of τ_{min} and δ are set to satisfy $\tau_{min} > \delta$ to meet safety requirements. That is, we always have $\tau_H^t > \delta$ and $\tau_L^t > \delta$.

1) Not Around DI state: When the vehicle is not near a dynamic intersection and has no request from other vehicles, the vehicle is in the *Not Around DI* state. When the vehicle approaches a dynamic intersection, the vehicle state Φ_{state} transitions to the *Approaching DI* state (See Section III-C2). Also, when the vehicle receives a **DI REQUEST** packet, the state is changed to the *Queried* state (See Section III-C5).

2) Approaching DI state: Under the *Approaching DI* state, the vehicle keeps sending[2] the **DI REQUEST** packet. When the vehicle receives the **DI APPROVE** packet from other vehicles, the vehicle starts to enter the dynamic intersection and transitions to the *Crossing* state (See Section III-C8). If a vehicle has been waiting for more than the time period δ, its state transitions to the *CyberTL-calculating* state (See III-C3).

3) CyberTL-calculating state: A vehicle in the *CyberTL-calculating* state calculates the green period for each direction. Once the vehicle completes the calculation, the state is changed to the *Turn-starting* state (See Section III-C4). The calculation method is presented in Section III-D.

4) Turn-starting state: In the *Turn-starting* state, the vehicle measures its wait time before the dynamic intersection until the time τ_H^t passes. After the time τ_H^t passes, the vehicle keeps sending the **DI INTERRUPT** packet. When the vehicle can confirm that no vehicles are within the dynamic intersection, the vehicle enters the intersection and transitions to the *Crossing* state.

5) Queried state: When a vehicle in the *Not Around DI* state receives the **DI REQUEST** packet, the vehicle state Φ_{state} transitions to the *Queried* state. In the *Queried* state, the vehicle estimates the potential of conflict with the sender. The vehicle calculates the Arrival Time and the Exit Time of the requested dynamic intersection. Then, the vehicle replies with either the **DI DECLINE** packet or the **DI APPROVE** packet, and the state returns back to the *Not Around DI* state.

6) CyberTL-counting state: When the vehicle receives a **DI INTERRUPT** packet, its state is changed to *CyberTL-counting* state. In the *CyberTL-counting* state, the vehicle has to stop before the dynamic intersection for the time period τ_L^t. During the time period τ_L^t, the vehicle keeps sending the **DI APPROVE** packet. After the time τ_L^t passes, the vehicle state transitions to the *Turn-switching* state. Also, if all of the oncoming vehicles traverse the dynamic intersection before the time period τ_L^t passes, the vehicle starts to enter the dynamic intersection.

7) Turn-switching state: Under the *Turn-switching* state, the vehicle keeps sending the **DI INTERRUPT** packet to request the vehicles on the conflicting lane to stop before the dynamic intersection. When the dynamic intersection becomes clear, the vehicle enters the dynamic intersection and its state is changed to the *Crossing* state.

8) Crossing state: When the vehicle traverses the dynamic intersection, the vehicle is in the *Crossing* state and keeps sending the **DI CROSS** message. Once the vehicle exits the dynamic intersection, it transitions to the *Not Around DI* state.

[2]This means that the message will be sent out regularly at the 10 Hz rate of DSRC Advanced Safety Messages.

Next, we present how our *Request-response-based Negotiation* proceeds among multiple self-driving vehicles in our protocol. The negotiation sequence is shown in Figure 4 and Figure 5. First, in Figure 4, a vehicle arrives at the dynamic intersection from one direction, L_H or L_L. As shown in Figure 4-(a), when the vehicle comes from lane L_L, the vehicle transmits the **DI REQUEST** packet to the surrounding vehicles. The vehicle also uses its perception system to confirm safety traversed within the dynamic intersection. On the other hand, as shown in Figure 4-(b), when the vehicle comes from lane L_H, the vehicle does not transmit the Advanced Safety Message, because its lane has precedence. Overall, all negotiations are initiated from the vehicles coming from lane L_L. Unlike human-driven vehicles under current traffic rules, self-driving vehicles under our protocol do not have to stop and/or slow down before the dynamic intersection.

Suppose that, as shown in Figure 5, vehicles come to the dynamic intersection from two different directions, L_H and L_L, negotiation based on the request-response is initiated by the vehicle on lane L_L. The vehicle from lane L_L sends the **DI REQUEST** message (at $t = t_0$). Then, the vehicle on lane L_H calculates its Arrival Time and Exit Time. The two vehicles determine that they have the potential to collide. The vehicle on lane L_H sends the **DI DECLINE** message in order to decline the request (at $t = t_1$). In this case, the vehicle from lane L_L stops before the dynamic intersection and keeps sending the **DI REQUEST** packet (at $t = t_2$). Once the vehicle on lane L_H crosses the dynamic intersection, the vehicle on lane L_L confirms safe clearance and enters the dynamic intersection if there is no oncoming traffic (at $t = t_3$).

A key challenge of this request-response-based protocol is that the vehicle on lane L_L might wait a long time before the dynamic intersection when there is a high volume of traffic coming from the other direction. If vehicles keep arriving from lane L_H, the waiting vehicle on lane L_L would keep receiving the **DI DECLINE** packet and keep waiting. To address the issue explicitly, we use the notion of a Cyber Traffic Light (CyberTL). The details of the CyberTL are presented next.

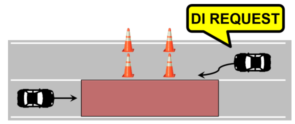
A simulation video of our request-response-based protocol can be seen at:

https://www.youtube.com/watch?v=xkfyAkXzAXg

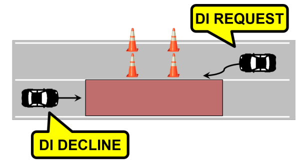
D. *Cyber Traffic Light*

In this section, we present the Cyber Traffic Light (CyberTL) that is used to keep the worst-case waiting time short while maintaining high traffic throughput. The CyberTL works as a self-optimizing traffic light by estimating the surrounding traffic volume and by coordinating with oncoming vehicles. The CyberTL is a V2V communication-based traffic manager, and we use the OBU of each self-driving vehicle to support this distributed software abstraction.

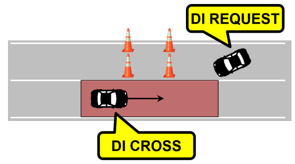
By using the CyberTL, the vehicles at a dynamic intersection coming from two different directions take turns. That is, once n vehicles have passed the dynamic intersection from one direction, m vehicles from the other direction can enter and traverse the dynamic intersection ($n, m \in \mathbb{N}$).



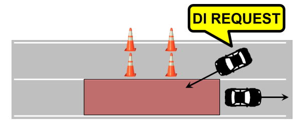
(a) $t = t_0$



(b) $t = t_1$

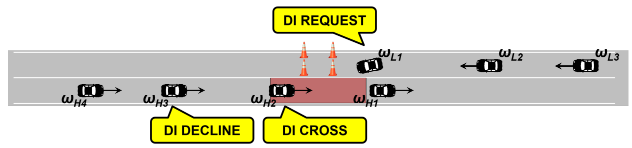


(c) $t = t_2$

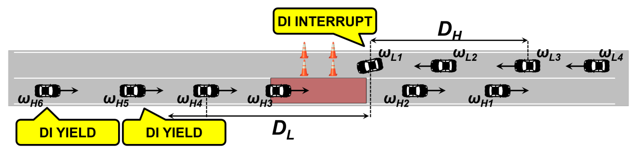


(d) $t = t_3$

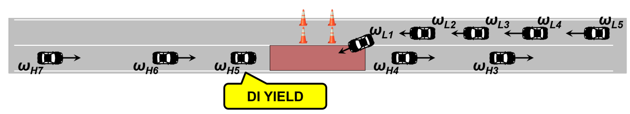
Figure 5: Negotiation based on the *Request-response*.



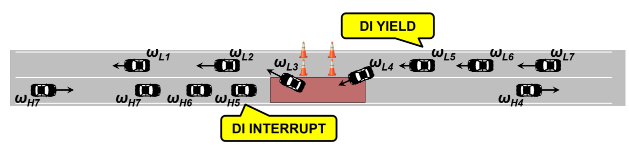
(a) $t = t_0 - \delta$



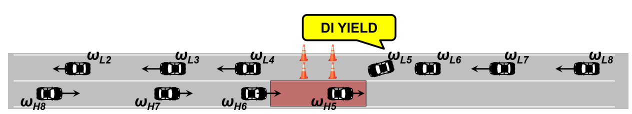
(b) $t = t_0 - \delta + \tau_H^{t_0}$



(c) $t = t_0 - \delta + \tau_H^{t_0} + ST(\tau_H^{t_0})$

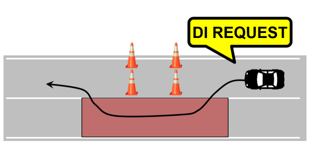


(d) $t = t_1$

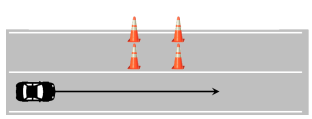


(e) $t = t_1 + \Delta$

Figure 6: DI management with *CyberTL*.



(a) Only ω_L comes



(b) Only ω_H comes

Figure 4: Safety Confirmation for Dynamic Intersection.

The vehicles that together constitute the CyberTL are grouped into 3 categories: (i) *CyberTL Calculator*; (ii) *CyberTL Counter*; and (iii) *CyberTL Follower*. First, the first waiting vehicle on lane L_L becomes the *CyberTL Calculator* when its vehicle state Φ_{state} is changed to the *CyberTL-calculating* state. The *CyberTL Calculator* estimates the arrival traffic volumes, and calculates the green periods, τ_H^t and τ_L^t, along the two directions for one cycle. Here, t represents the time when the green periods are estimated. Once the waiting time of the *CyberTL Calculator* exceeds τ_H^t, the vehicle transmits the **DI INTERRUPT** packet and starts the maneuver. When the vehicle coming from lane L_H receives the **DI INTERRUPT** and replies with the **DI YIELD** packet, its state transitions to the *CyberTL-counting* state. Any vehicle in the *CyberTL-counting* state has to stop and has to measure the waiting time acting as a *CyberTL Counter*. The *CyberTL Counter* vehicle keeps sending the **DI YIELD** message during τ_L^t, and once the waiting time exceeds τ_L^t, the vehicle starts to send the **DI INTERRUPT** packet. The other surrounding vehicles, *CyberTL Followers*, just follow the decisions made by the *CyberTL Calculator* and *CyberTL Counter*. Overall, the green period allocation is calculated by the *CyberTL Calculator*, and the time measurements are carried out by the *CyberTL Calculator* and *CyberTL Counter*.

The dynamic intersection management scheme with the CyberTL is illustrated in Figure 6. Also, an example timeline of the dynamic intersection use is presented in Figure 7. In this example, the *CyberTL Calculator* is assigned for the vehicle ω_{L1} at time t_0 and vehicle ω_{L5} at time t_1. Also, the *CyberTL Counter* is assigned for the vehicle ω_{H5}. At $t = t_0$, when the duration δ is passed after its arrival, the vehicle ω_{L1} calculates the green periods $\tau_H^{t_0}$ and $\tau_L^{t_0}$. During the period $\tau_H^{t_0}$, vehicles on lane L_H can enter the dynamic intersection. The *Switching Time* duration is the required time period to reverse the direction of traffic flow, and is equal to the time taken from the beginning of the **DI INTERRUPT** message transmission to the dynamic intersection clearing. This duration cannot be estimated beforehand, because it depends on the abilities and physical sizes of the traversing vehicles.

The green periods τ_H^t and τ_L^t are calculated based on the real-time traffic volumes and on the metrics that are widely used in existing traffic systems [1]. In the protocol, we first estimate the real-time traffic volumes from the surroundings using V2V communications, and we calculate the approximate total lost time from the dynamic intersection length and vehicle size.

First, connected self-driving vehicles estimate the arrival traffic volumes, ψ_H and ψ_L, based on the information available from the received BSM, as shown in Eqs. (3) and (4). Here, N_H and N_L represent the number of vehicles on lane L_H and on lane L_L within the communication range, respectively. v_H and v_L are the average velocity of the vehicles on each lane within the communication range. D_H and D_L represent the distance from the transmitter to the farthest vehicle on lane L_H and on lane L_L, respectively. Since the transmitter might not know the accurate boundaries of the communication range, we use these two distances for estimation purposes.

$$\psi_H = \frac{N_H \cdot v_H}{D_H} \tag{3}$$

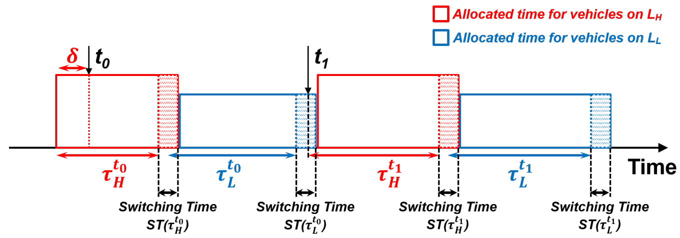


Figure 7: Allocated Green Period for *CyberTL*.

$$\psi_L = \frac{N_L \cdot v_L}{D_L} \tag{4}$$

Then, to calculate the appropriate cycle length C, the *CyberTL Calculator* vehicle estimates the approximate total lost time L_{DI} as shown in Eq. (5). Here, the total lost time is the sum of the *Switching Time* in a cycle. Since an accurate *Switching Time* cannot be estimated accurately beforehand, we use the size of the *CyberTL Calculator* vehicle l_v, the length of the dynamic intersection D_{DI}, and the permitted velocity v_{DI}.

$$L_{DI} = \frac{2 \times (l_v + D_{DI})}{v_{DI}} \tag{5}$$

Based on Eqs. (3), (4), and (5), the appropriate cycle length C is estimated as shown in Eq. (6).

$$C_{DI} = \frac{L_{DI}}{1 - (\frac{\psi_H + \psi_L}{0.9 \cdot s_0 \cdot f_a})} \tag{6}$$

Here, we use the two variables, the *Base Saturation Rate* s_0 and *Area Adjustment Factor* f_a. According to the National Research Council [1], s_0 is set as $1,900$, and f_a is set to 1.0 as default and set to 0.9 if the area is in a central business districts.

Finally, the green periods, τ_H^t and τ_L^t, are calculated by using the cycle length and total lost time, as shown in Eqs. (7) and (8). The allocated green periods reflect the arrival traffic volumes from the two directions.

$$\tau_H^t = \frac{\psi_H}{\psi_H + \psi_L} \cdot (C_{DI} - L) \tag{7}$$

$$\tau_L^t = \frac{\psi_L}{\psi_H + \psi_L} \cdot (C_{DI} - L) \tag{8}$$

The allocated green periods τ_H^t and τ_L^t are checked against reasonable minimum and maximum values, τ_{min} and τ_{max}, in order to avoid vehicles having to wait for a long time before the dynamic intersection. Also, the cycle length must not exceed a maximum allowable value set by the local jurisdiction (such as 150 sec) and it must be long enough to serve *at least* one vehicle.

A simulation video of our Cyber Traffic Light can be seen at:

https://www.youtube.com/watch?v=xlFZgofh2iw

E. Anomalies and Packet Loss

Our dynamic intersection protocol assures safety by using V2V communications and the perception systems cooperatively. It inherently accommodates the loss of transmitted packets, since safety messages are transmitted at 10 Hz and occasional packet losses are compensated by subsequent successful transmission. In addition, our request-response-based negotiation enhances safety by its design.

In fact, the loss of the **DI REQUEST** or the **DI INTERRUPT** messages makes the waiting time longer for the transmitter, but it will not cause a vehicle collision or a deadlock because the sender vehicle keeps waiting before the dynamic intersection until it receives the appropriate response. In addition, when the **DI APPROVE** or the **DI YIELD** message is lost, the receiver might lose the opportunity to enter the dynamic intersection, but it again cannot cause a vehicle collision. However, the waiting time of the receiver vehicle might be longer. Moreover, the packet loss of the **DI DECLINE** message poses no safety risk because the receiver vehicle has to keep waiting until it receives the appropriate response when there are other vehicles around the dynamic intersection.

Finally, in addition to these packets, our protocol uses the **DI CROSS** message to inform that the transmitter vehicle is inside the intersection area. The vehicles using our traffic protocol cannot enter the intersection if they receive a **DI CROSS** message, meaning the message is used to enhance the safety and reliability around the intersection.

IV. Evaluation

In this section, we present the implementation and evaluation of our cooperative dynamic intersection protocol in our hybrid simulator-emulator named *AutoSim*. We evaluate the traffic protocol in terms of traffic throughput and compare against two baseline traffic control methods: (i) *Temporary Traffic Signal Method* and (ii) *Stop or Yield Control Method*.

AutoSim is an extended version of Groovenet [15] with 3-D graphics and other capabilities. AutoSim consists of various core models, including mobility, communication, control, localization, and pose estimation for each simulated vehicle. In AutoSim, each simulated vehicle can transmit and receive a Basic Safety Message (BSM) which is one of the main requirements of vehicular communications. The message format used follows the definition of the SAE J2735 standard, which is the DSRC Message Set Dictionary. In addition, AutoSim is a modular hybrid simulator-emulator for vehicular communications that enables interactions between virtual and real vehicles equipped with DSRC radios.

A. Metric

In order to evaluate the utility of our protocol, we define the *Trip Time* as the time taken by a vehicle to go from a known start-point before the dynamic intersection to a known end-point after the intersection. We calculate the trip time for each simulated car and compare that against the trip time taken by the car assuming that it stays at a constant speed and does not stop at the dynamic intersection. The difference between these two trip times is considered to be the *Trip Delay* due to the dynamic intersection, and the *Average Trip Delay* and

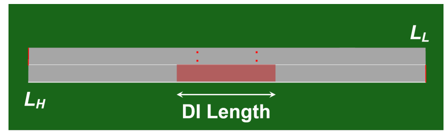


Figure 8: Map for *One-way Alternative Use*.

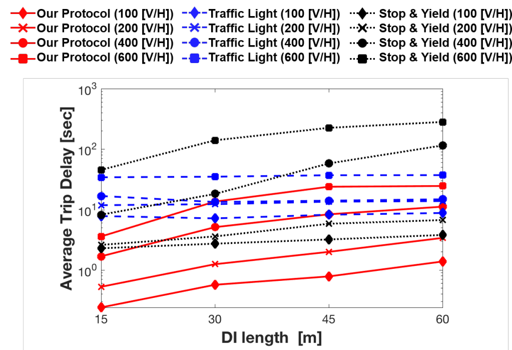


Figure 9: Average waiting time for symmetric traffic volume.

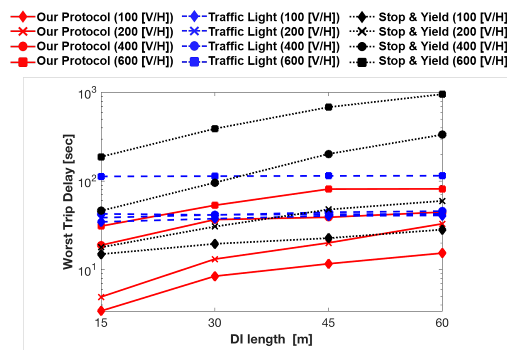


Figure 10: Worst waiting time for symmetric traffic volume.

the *Worst Trip Delay* across all vehicles in each simulation are compared in our evaluation.

B. Scenarios

Since dynamic intersections in the real world take a variety of forms and sizes, we restrict the experimental settings to two specific kinds of dynamic intersections: One-way Alternative Use (Temporary DI) and Turning Left through Traffic (Unmapped DI). We assume that all vehicles calculate the required trajectories to traverse the dynamic intersection by themselves. In addition, the traffic generation follows the Poisson random distribution. Each simulation runs for 30 minutes, and we evaluate the time delay of the vehicles that are generated during the last 20 minutes to bypass initial observation and measure steady-state behavior.

Our simulations evaluate our protocol by comparing against two baseline protocols: *Temporary Traffic Light Method* and *Stop or Yield Control Method*.

Temporary Traffic Light Method Traffic signals may be temporarily used to control vehicular traffic movements in Temporary DI, such as the areas requiring alternating one-way traffic operations. In the simulations, the time length for

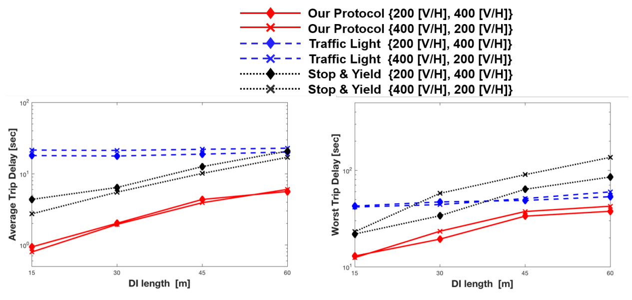


Figure 11: Trip Delay for Asymmetric Traffic Volume.

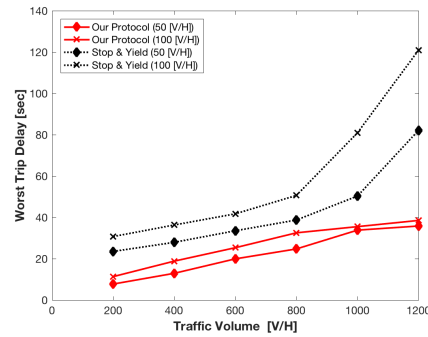


Figure 12: Worst waiting time for *Turning Left through Traffic*.

a green light is set as 30 seconds, and the time length for a yellow light is set as 5 seconds.

Stop or Yield Control Method All vehicles follow the priority given to the lane. Specifically, vehicles on a higher-priority lane are not required to stop or slow down at the dynamic intersection. Correspondingly, vehicles on a lower-priority lane have to stop before the dynamic intersection for guaranteeing safety. In the Temporary DI, a stop or yield sign might be temporarily installed on roads where one side of the road way is closed. The side that is closed should stop for and yield to oncoming traffic on the side that is open.

We assume that there is no obstacle completely blocking perception around the dynamic intersections.

The variables for the simulations are set by accounting for the DSRC standards [9], [13] and typical vehicle abilities, as shown in Table I. For example, the communication range and frequency for V2V communications are set as 500 m and 10 Hz, respectively.

C. Traffic Throughput Evaluation

We run simulations for evaluating the traffic throughput of our cooperative dynamic intersection protocol and compare against the baseline protocols. To measure the traffic throughput, we focus on two specific types of dynamic intersections.

1) One-way Alternative Use (Temporary DI): We evaluate the traffic throughput for *One-way Alternative Use* (Temporary DI). The length of the dynamic intersection (DI length) is changed from 15 m to 60 m. In Figure 8, we present the map of the dynamic intersection. We first assume that the traffic volumes from two directions are symmetric, and then evaluate the throughput when the traffic volumes are asymmetric.

For the symmetric traffic volume, we evaluate the average trip delay and worst trip delay when the range of the traffic volume for each lane is changed from 100 to 600 vehicles per hour. In Figure 9, we present the average trip delay of all the vehicles, and Figure 10 presents the worst trip delay of all the vehicles. Our cooperative dynamic intersection protocol has the shortest average trip delay and the shortest worst trip delay in all cases. The Temporary Traffic Light Method has a constant and long average delay because all of the arriving vehicles have to wait for the green period and the delay is not related to the intersection size. When the traffic volume is high, the Stop or Yield Control Method has the longest average trip delay and worst trip delay, because the vehicles on the lower-priority lane can enter the dynamic intersection only when there is no oncoming traffic.

Table I: Environmental settings for the experiments.

Communication range	500 (m)
Communication frequency	10 (Hz)
Max speed of vehicles	30 (km/h)
Vehicle length l	4.5 (m)
Vehicle width w	1.7 (m)

Second, as shown in Figure 11, we evaluate the average trip delay and worst trip delay when the traffic volume from the two directions is asymmetric. Here, we evaluate the trip delay when the {*traffic volume from* L_H, *traffic volume from* L_L} = {200 [V/H], 400[V/H]}, {400 [V/H], 200[V/H]}. In all cases, our dynamic intersection protocol still has the shortest average trip delay and the shortest worst trip delay. Since our CyberTL dynamically allocates the green period depending on the short-term arrival traffic volume, the average trip delays of {200 [V/H], 400[V/H]} and of {400 [V/H], 200[V/H]} are almost the same. On the other hand, under the Stop or Yield Control Method, when we increase the traffic volume on the lane L_H, the worst trip delay becomes much larger.

2) Turning Left through Traffic (Unmapped DI): In addition, we evaluate the worst trip delay when vehicles turn left through oncoming traffic. Since this dynamic intersection is an Unmapped DI, not due to construction, we compare our protocol only to the Stop or Yield Control Method. The traffic volume of the oncoming lane L_H is varied from 100 to 1,200 vehicles per hour. We assume that the traffic volume on lane L_L is 50 or 100 vehicles per hour, and there are no vehicles going straight on lane L_L.

In Figure 12, we present the worst trip delay of all the vehicles around the dynamic intersection. Our dynamic intersection protocol again has the shortest worst trip delay in all cases. Under the Stop or Yield Control Method, all vehicles on lane L_L have to slow down and halt before the dynamic intersection to guarantee safety. Therefore, even when the traffic volume on lane L_H is relatively low, the Stop or Yield Control Method has longer worst trip delay. Moreover, when the oncoming traffic volume is high, under the Stop or Yield Control Method, some vehicles on lane L_L have to wait a very long time before the dynamic intersection. On the other hand, when the traffic volume is high, our protocol uses the Cyber Traffic Light to allow the waiting vehicles to use the dynamic intersection.

V. Related Work

Since the DARPA Urban Challenge in 2007 [18], a competition for autonomous driving vehicles, cooperation and collaboration among self-driving vehicles has been widely studied for intelligent intersection management, platooning, merging, and lane-changing maneuvers.

First, Dresner and Stone proposed a multi-agent approach for an intersection control, named Autonomous Intersection Management [10]. In this system, all vehicles call ahead to a reservation manager agent at the intersection to reserve conflict-free trajectories. Since the protocol relies on Vehicle-to-Infrastructure communications and on the manager agent, the infrastructure might be a single point of failure for the intersection management system. The authors extended the system to heterogeneous traffic environments [5], in which vehicles are assumed to be partially automated.

Secondly, as a priority-based intersection protocol, Azimi et al. [7], [8] proposed several spatio-temporal intersection protocols (STIP). In these protocols, vehicles are assigned priorities based on their arrival times at the intersection, with vehicles reaching the intersection earlier assigned higher priorities than vehicles coming later. A vehicle with trajectory conflicts will come to a complete stop before entering the intersection, and waits until all vehicles having higher priority than this vehicle have crossed the conflicting areas. Under STIP, all vehicles exchange the expected arrival time and required trajectory with Vehicle-to-Vehicle (V2V) communications before reaching the intersection. These priority-based intersection protocols were extended to merge points, where two lanes with different priorities meet [4]. In this work, vehicles safely traverse merge points by using both vehicular communications and their own perception systems. The merge protocol also supports *zipper merge* where the vehicles from different lanes merge at the merge points by taking turns when the traffic density is high. In addition, the protocol enables cooperation and collaboration between self-driving vehicles and human-driven vehicles. Gregoire et al. [11] proposed priority-based scheduling techniques for multi-robot planning and cooperation, and Zhang et al. [20] extended the work for self-driving vehicles. In this work, a priority scheduling algorithm is designed based on the modeling of vehicular spatio-temporal relations, for coordinating multiple vehicles at intersections.

Thirdly, a novel class of protocols called synchronous intersection protocols, BRIP [6] and CSIP [3], were presented. Under these synchronous intersection protocols, all self-driving vehicles cross the intersection without stopping by following a strict but well-defined spatio-temporal pattern tailored to each intersection. BRIP and CSIP allow vehicles to efficiently and continuously enter and cross the intersection in a synchronized manner.

In addition, lane-changing maneuvers have been considered as a longitudinal motion planning problem [16], [17]. In this work, automated vehicles select an appropriate inter-vehicle traffic gap and time instance to perform the maneuver by estimating whether there is a longitudinal trajectory allowing the vehicles to complete the maneuver. No cooperation is assumed, expected, or accomplished.

The above approaches have addressed individual driving contexts, such as simple multi-way intersections, roundabouts, merging points, and lane-changing maneuver, and they did not study more complicated situations. On the other hand, our protocol can be used for many complex driving contexts we encounter in practice.

VI. Conclusion

In this paper, we defined and classified Dynamic Intersections where the paths of vehicles intersect dynamically and perhaps temporarily. Often, these intersections are not present in a map database. These dynamic intersections are common in or near parking lots, multi-level parking garages, work zones, accidents, potholes, and driveways. Such dynamic intersections might appear almost anytime and anywhere along public roads. We presented a cooperative dynamic intersection protocol for self-driving vehicles to safely traverse dynamic intersections by using V2V communications and perception systems.

Under our protocol, connected and automated vehicles are not required to stop before dynamic intersections when there is no vehicle coming from other directions. Our protocol increases the traffic throughput significantly at dynamic intersections, compared to two baseline protocols. The protocol has the shortest average trip delay and the shortest worst trip delay in all the cases we considered. Under our protocol, self-driving vehicles calculate the possible collision points, specify the dynamic intersections, and manage the spatial area by using the vehicle-state transition and a Cyber Traffic Light. The Cyber Traffic Light enhances the traffic throughput around the dynamic intersections when traffic volumes are high.

We note several limitations of our work. First, our cooperative dynamic intersection protocol must be extended to support Moving DI and Variegated DI where vehicle behaviors are more complex, with interactions and effects between multiple dynamic intersections. Secondly, we have currently assumed that perception works perfectly, and there is no obstacle blocking a perception around the dynamic intersection. In future work, we will design protocols to support limits on perception. Thirdly, we only consider the protocol for fully-automated vehicles. Since a long transition period will likely be needed to replace human-driven vehicles with fully self-driving vehicles, we will need to design protocols for the heterogeneous environment where self-driving vehicles and human-driven vehicles co-exist. Finally, our cooperative dynamic intersection protocol assumed that vehicles always have good GPS, and there are no component failures. In future work, we will study the feasible protocols to enhance safety when GPS and/or components fail.

References

[1] *Highway capacity manual*. Transportation Research Board, Washington, DC, 2000.

[2] *Manual on uniform traffic control devices 2009 edition (MUTCD)*. United States Department of Transportation, Federal Highway Administration, 2009.

[3] S. Aoki and R. R. Rajkumar. A configurable synchronous intersection protocol for self-driving vehicles. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2017 IEEE 23rd International Conference on*, pages 1–11. IEEE, 2017.

[4] S. Aoki and R. R. Rajkumar. A merging protocol for self-driving vehicles. In *Cyber-Physical Systems (ICCPS), 2017 IEEE/ACM Third International Conference on*, pages 219–228. ACM, 2017.

[5] T.-C. Au, S. Zhang, and P. Stone. Autonomous intersection management for semi-autonomous vehicles. In D. Teodorovi'c, editor, *Handbook of Transportation*, pages 88–104. Routledge, 2016.

[6] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Ballroom intersection protocol: Synchronous autonomous driving at intersections. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2015 IEEE 21st International Conference on*, pages 167–175. IEEE, 2015.

[7] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige. Reliable intersection protocols using vehicular networks. In *Cyber-Physical Systems (ICCPS), 2013 ACM/IEEE International Conference on*, pages 1–10. IEEE, 2013.

[8] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige. Stip: Spatio-temporal intersection protocols for autonomous vehicles. In *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*, pages 1–12. IEEE, 2014.

[9] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige. Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band. *IEEE Journal on Selected Areas in Communications*, 25(8), 2007.

[10] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008.

[11] J. Gregoire, S. Bonnabel, and A. De La Fortelle. Priority-based intersection management with kinodynamic constraints. In *Control Conference (ECC), 2014 European*, pages 2902–2907. IEEE, 2014.

[12] W. Gruel and F. Piller. A new vision for personal transportation, 2016.

[13] J. B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[14] R. B. Langley. Rtk gps. *GPS World*, 9(9):70–76, 1998.

[15] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige, and F. Bai. Groovenet: A hybrid simulator for vehicle-to-vehicle networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–8. IEEE, 2006.

[16] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson. Lane change maneuvers for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1087–1096, 2017.

[17] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson. If, when, and how to perform lane change maneuvers on highways. *IEEE Intelligent Transportation Systems Magazine*, 8(4):68–78, 2016.

[18] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[19] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi. Towards a viable autonomous driving research platform. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 763–770. IEEE, 2013.

[20] K. Zhang, D. Zhang, A. de La Fortelle, X. Wu, and J. Gregoire. State-driven priority scheduling mechanisms for driverless vehicles approaching intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 16(5):2487–2500, 2015.