Highly Efficient Traffic Planning for Autonomous Vehicles to Cross Intersections without a Stop

JIAN KANG, Google LLC, USA
DAN LIN, Vanderbilt University, USA

Waiting in a long queue at traffic lights not only wastes valuable time but also pollutes environment. With the advances in autonomous vehicles and 5G networks, the previous jamming scenarios at intersections may be turned into non-stop weaving traffic flows. Towards this envision, we propose a highly efficient traffic planning system, namely DASHX, which enables connected autonomous vehicles to cross multi-way intersections without a stop. Specifically, DASHX has a comprehensive model to represent intersections and vehicle status. It can constantly process large volumes of vehicle information, resolve scheduling conflicts and generate optimal travel plans for all the vehicles coming towards the intersection in real time. Unlike existing works which are limited to certain types of intersections and lack considerations of practicability, DASHX is universal for any type of 3D intersection, yields the near-maximum throughput while still ensuring riding comfort. To better evaluate the effectiveness of traffic scheduling systems in real-world scenarios, we developed a sophisticated open-source 3D traffic simulation platform (DASHX-SIM) that can handle complicated 3D road layouts, simulate vehicles' networking and decision-making processes. We have conducted extensive experiments, and the experimental results demonstrate the practicality, effectiveness, and efficiency of DASHX system and the simulator.

CCS Concepts: • **Computer systems organization → Embedded and cyber-physical systems**; • **Information systems → Information systems applications**.

Additional Key Words and Phrases: Autonomous vehicle, traffic control, intersection management

ACM Reference Format:
Jian Kang and Dan Lin. 2022. Highly Efficient Traffic Planning for Autonomous Vehicles to Cross Intersections without a Stop. *ACM Trans. Intell. Syst. Technol.* 1, 1, Article 1 (November 2022), 24 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

According to the recent report [10], people spent an average of 58.6 hours waiting at red lights each year. The traffic at the intersections which have only stop signs is even slower. As estimated by the new data collected from connected vehicles, departments of transportation, cellular positioning reports, and many other sources, Americans wasted $87 billion sitting in the car in 2018 [25]. Similar scenarios happen in other countries as well, such as China, Russia, Brazil, and Turkey.

Recent study [18] shows that vehicles moving at a low speed (i.e., less than 30 miles per hour) require more fuel consumption. Also, when a vehicle needs to stop frequently, the frequent acceleration and deceleration consume significant amount of fuel. Imagine that if cars coming from different directions seamlessly interleave with each other without any stop, it will not only help

Authors' addresses: Jian Kang, jkng@google.com, Google LLC, Mountain View, California, USA, 94043; Dan Lin, dan.lin@vanderbilt.edu, Vanderbilt University, Nashville, Tennessee, USA, 37240.

2157-6904/2022/11-ART1 $15.00
https://doi.org/10.1145/1122445.1122456

save a huge amount of travel time and fuel consumption but also reduce the pollutants emitted by cars and protect our environments. This ideal scenario may become a reality with the prevalence of connected autonomous vehicles in the future. Their capabilities of sensing, processing, and communication have provided a promising hardware foundation [16]. The missing piece of the blueprint is an efficient cyber-physical system that can achieve the goal of real-time traffic scheduling. Therefore, we propose a novel intersection travel scheduling system for autonomous vehicles called DASHX (Dynamic Autonomous-vehicle Stream Handling).

The DASHX system has a comprehensive model to handle any type of intersection including 3D intersections. It utilizes an Intersection Management Unit (IMU) [4] which communicates with connected autonomous vehicles approaching the intersection. Specifically, when vehicles approaching an intersection, they report their status (e.g., vehicle velocity, current location, destination, acceleration capability) and requirements (e.g., going straight, left turn, right turn) to the IMU. The DASHX system constantly collects and processes data streams from all vehicles coming towards the intersection, resolves scheduling conflicts, and generates the optimal travel plan for all the vehicles in real-time to guide them crossing intersections in a safe and highly efficient way.

Although there have also been decentralized traffic control mechanisms that fully rely on vehicle-to-vehicle (V2V) cooperation [6, 8, 12, 39, 47, 48, 51, 54], we argue that a centralized traffic management scheme may be more practical especially for the critical security and safety considerations. First, V2V based approaches assume participating vehicles are all trustworthy and will honestly and correctly follow the travel plans. However, building trust among peer stranger vehicles is still very challenging. The use of IMU will obviate this problem. The trust between the vehicles and the IMU would be similar to that between cellphone users and service providers. Second, servers are typically more difficult to be compromised than personal computing devices, i.e., individual vehicles. Third, with the adoption of advanced communication technologies, such as 5G networks [5, 41], the densely deployed 5G base stations could be utilized as the intersection manager [31]. The existing smart traffic light systems [15] could also be equipped with our proposed system to serve as the IMU. Alternatively, considering the Internet of Vehicles, the IMU may reside in the cloud and collect vehicle information via Internet connections in the future.

In order to carry out an extensive evaluation of the DASHX system, we also develop a comprehensive microscopic 3D traffic simulation platform, namely DASHX-SIM, which can simulate traffic in real-world scenarios. DASHX-SIM provides a graphic interface for researchers to conveniently edit road maps and intersections, specify fine-grained traffic constraints on each road. DASHX-SIM simulates how vehicles move, communicate and calculate their travel plans according to given protocols and traffic restrictions. Such an interactive testing environment allows researchers to visualize the traffic generated by their protocols in real time and compare performance of different traffic planning protocols, which will greatly enhance the protocol design and optimization process. It is also worth noting that DASHX-SIM is much more advanced than the existing vehicle simulation platforms, such as SUMO [24], OpenTrafficSim [43], and CityFlow [53], in that these open-source platforms support only intersections of 2D layouts which are not sufficient to model real-world scenarios. Although a couple of the commercial platforms, such as VISSIM [13] and AnyLogic [22], claim to support more complicated intersection layouts, they are not free for research studies [11]. Our DASHX-SIM is open-source, highly extensible, and generic, as it can evaluate the performance of not only the traffic management protocols, but also the routing protocols in VANETs (Vehicular Ad-hoc Networks) and the authentication mechanisms for connected vehicles.

To sum up, we have made the following unique contributions in this work:

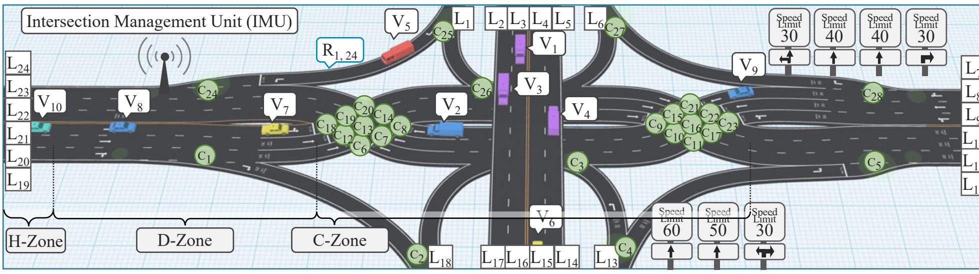


Fig. 1. Processing Traffic Data at Complex Intersections

- We propose a generic intersection traffic scheduling system – DASHX – which is capable of modeling any type of 2D or 3D intersections as complicated as diverging diamond interchange (Fig. 1), 3-Layer interchange (Fig. 3 (f)), 2-Layer irregular intersection (Fig. 3 (i)), etc. Existing works on intersection control mainly discuss either the 2D n-way intersection or the roundabout, and hence lack generality.
- Our proposed DASHX yields the near-maximum throughput while simultaneously ensuring riding comfort that prevents sudden stop and acceleration. Even when the vehicle density is very high, DASHX still guarantees that all vehicles crossing the intersections with **zero stop** which has not been achieved in any other existing work. Specifically, DASHX has a novel and efficient traffic planning scheme that considers various real-world constraints and calculates individual vehicle motion plans towards achieving globally optimal traffic control. The travel plans have been generated to allow each vehicle to adjust the speed and cross the intersection in a specific sequence that no one needs to stop. This is a very challenging task since an adjustment of one vehicle's motion plan can propagate to multiple vehicles and create cascading effects, which need to be resolved in real-time to avoid car accidents.
- We develop an open-source traffic simulation platform – DASHX-SIM – to facilitate the evaluation of different kinds of traffic planning protocols in complicated traffic scenarios with various types of intersections and travel constraints.
- We thoroughly evaluated DASHX under a variety of intersections and traffic flows including unbalanced traffic flows that have not been studied in any previous works. The experimental results demonstrate the practicality, efficiency, and effectiveness of our system compared to the state of the art.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 presents our proposed DASHX system. Section 4 introduces our developed DASHX-SIM traffic simulation platform. Section 5 reports the experimental results. Lastly, Section 6 concludes the paper.

2 RELATED WORK

Existing approaches on intersection traffic planning can be classified into three main categories: (i) Signal-based approaches; (ii) Platoon-based approaches; and (iii) Motion planning based approaches.

Signal-based approaches [2, 6, 14, 17, 23, 34, 35, 54] aim to find out the optimal traffic light scheduling, i.e., when, how long and which signal light should be on, Note that even with the optimal scheduling, many vehicles will still have to stop at the intersection. Also, it only works for intersections with traffic lights. Thus, this thread of work is least related to our work.

Platoon-based approaches aim to organize vehicles traveling in the same direction into a group and let them cross the intersection together [26, 47, 50]. Although organizing vehicles as

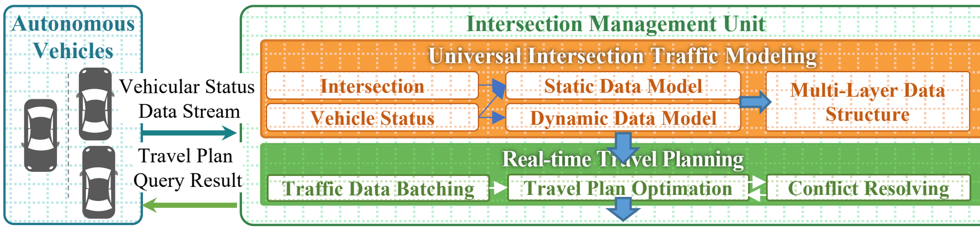
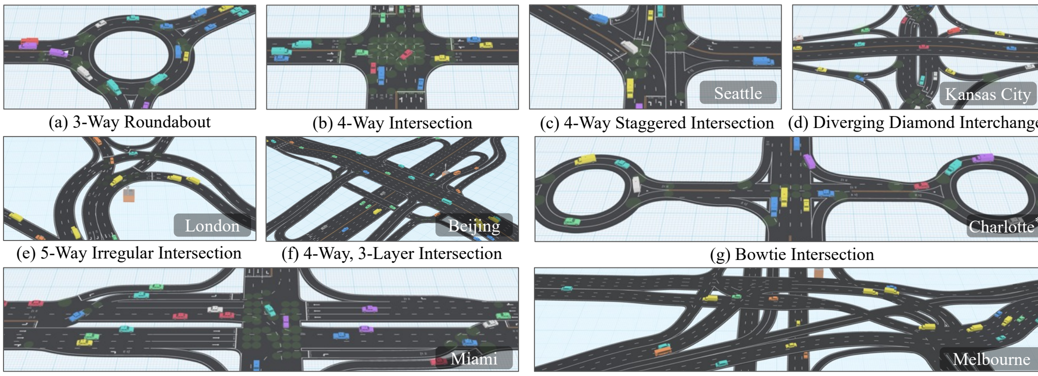


Fig. 2. DASHX System Overview

a platoon could effectively increase the crossing speed of a group of vehicles, its feasibility and efficiency might be limited in many scenarios. First, it is difficult to form a platoon if vehicles on the same lane have different destinations. Second, if vehicles coming from different lanes arrive at the intersection evenly, the platoon-based approach will experience the same problem of the signal-based approaches, i.e., many vehicles will have to stop and wait until the platoon from the other direction fully crosses the intersection. Compared to the platoon-based approaches, our approach will allow all the vehicles to cross the intersection without any stop even in high vehicle density scenarios.

Our work is mostly related to the **motion planning based approaches** which plan the speed for each vehicle that is about to cross the intersection. Compared to other systems, such as slot-based systems [42] which assign time slots to vehicles and let them cross the intersection during the assigned time slots, the motion planning-based systems typically yield much higher throughput as vehicles receive fine-grained optimized travel plans. There are both centralized and decentralized approaches. The centralized approaches [27–29, 32, 36, 37] utilize an intersection manager to calculate motion plans for all the crossing vehicles, while the decentralized approaches [8, 12, 30, 33, 46, 48] are based on vehicle-to-vehicle communications to obtain an agreed crossing order among vehicles. Motion planning based approaches so far yield the highest throughput among other traffic planning approaches. However, there are still many limitations in the existing work. First, most approaches consider simple intersections such as the common 4-way intersection. Although they claim their approaches may be extended to other types of intersections, such extension is not trivial as we discuss in our work. Second, most existing approaches rely on simplified data models that consider only a subset of real-world constraints. Our work is more comprehensive as it integrates various constraints during traffic planning. Third, although existing motion planning based approaches improve the travel efficiency at the intersection, there are still many vehicles which need to stop and waiting for their turn to cross. Our DASHX algorithm is nearly optimal as it guarantees non-stopping crossing for all the vehicles even when vehicle density is high.

In the end, we also review the related works on **traffic simulation platforms**. Based on the supported type of visualization, the existing microscopic traffic simulation platforms can be classified into two main categories [38, 40]: (i) 2D traffic simulation platforms and (ii) 3D traffic simulation platforms. The 2D traffic platforms, such as SUMO [24] and MATSim [44], support traffic simulation on 2D road networks. They are not able to handle intersections with interleaving road bridges in a 3D layout. To the best of our knowledge, only commercial simulators currently support 3D modeling of intersections, such as AIMSUN [7], TransModeler [9], and PTV VISSIM [13]. These commercial traffic simulators provide very limited functionalities in their free versions, which cannot meet the researchers' needs on various tasks regarding traffic planning. Our proposed DASHX-SIM simulator overcomes the aforementioned limitations. The DASHX-SIM is open source



(a) 3-Way Roundabout (b) 4-Way Intersection (c) 4-Way Staggered Intersection (d) Diverging Diamond Interchange (e) 5-Way Irregular Intersection (f) 4-Way, 3-Layer Intersection (g) Bowtie Intersection (h) Continuous Flow Intersection (CFI) (i) 6-Way, 2-Layer Irregular Intersection

Fig. 3. Example Intersections

and equipped with a wide range of features that allow researchers to conveniently model either 2D or 3D intersections and freely test their own traffic scheduling, routing, or security protocols.

3 THE PROPOSED DASHX SYSTEM

In this section, we present the proposed DASHX (Dynamic Autonomous-vehicle Stream Handling) system. As illustrated in Fig. 2, DASHX consists of two major components: (i) Universal intersection traffic modeling; (ii) Real-time travel planning. The intersection traffic modeling aims to formally represent and integrate both the static information (the structure of the intersection, speed limit, safety gap, etc.) and the dynamic information associated with crossing vehicles (speed, acceleration, position, etc.). Based on the constructed traffic model, the real-time travel planning component conducts batch processing for vehicles coming towards the intersection at the same time, calculates the travel plans for individual vehicles that optimize multiple objectives including travel efficiency (i.e., overall throughput), passenger comfort (i.e., comfortable acceleration and deceleration) and emission reduction (i.e., minimizing stops), and resolve traffic conflicts. In what follows, we will introduce the details of these two components.

3.1 Universal Intersection Traffic Modeling

We propose a comprehensive and universal intersection traffic model, namely DASHX, which is an extension to our prior DASH model [20]. Compared to DASH which only supports 2D intersection layouts such as those shown in Fig. 3 (a), (b), (c), (e), (g), and (h), the new DASHX is capable of handling more complex real scenarios with 3D intersections and a more diverse set of vehicle parameters.

3.1.1 The Intersection Model. By observing the common intersections in real-world as illustrated in Fig. 3, we define the following parameters to capture the features of any type of intersections:

Lanes: We classify lanes at each intersection into three types: (i) incoming lanes (denoted as $\mathcal{L}_{in}$) whereby vehicles come towards the intersection, (ii) outgoing lanes (denoted as $\mathcal{L}_{out}$) whereby vehicles leave the intersection, and (iii) connecting lanes (denoted as $\mathcal{L}_{connect}$) that connect the incoming lane and the outgoing lane. Lanes may have a variety of shapes denoted by a series of 3D coordinates according to the real-world intersections.

Moreover, our lane modeling is much more flexible than any prior works. Our modeling can model smart lanes in the future world whereby each lane $\mathcal{L}_i$ has its own safety restrictions and regulations such as speed limit ($\mathcal{L}_i.v_{max}$), minimum cruising speed ($\mathcal{L}_i.v_{min}$), and safety gap between two vehicles ($\mathcal{L}_i.\mathcal{G}_{min}$). Each lane is also associated with a weight $\mathcal{L}_i.w$ that serves as a tunable 'knob' to allow the dynamic adjustment of the traffic flow in response to the neighbor intersections when needed (e.g. increase or decrease the priority of a specific lane to ease the traffic pressure of the neighbor intersection).

Routes: A route $\mathcal{R}_{in,out}$ is defined as the moving path from an incoming lane $\mathcal{L}_{in}$, through a series of connecting lanes $\mathcal{L}_{connect}$, and to an outgoing lane $\mathcal{L}_{out}$. The introduction of the concept of routes help accommodate any shapes of intersections and any kinds of turns (e.g., going straight, left turn, right turn, slightly left turn, slightly right turn, u-turn) at the intersection. In the previous works, intersections are usually modeled as a grid of equally divided cells which however would be hard to represent complex intersections such as the example shown in Fig. 1 and Fig. 3. In our model, we can easily represent the wavy turn from the incoming lane $\mathcal{L}_{21}$ to the outgoing lane $\mathcal{L}_{10}$ in Fig. 1 as $\mathcal{R}_{21,10}$. Note that, our model is also generic enough for the scenario whereby in the future, autonomous vehicles can make the right or left turn at any lane instead of the rightmost or leftmost lane in current traffic settings.

Conflicting Points: A conflicting point C_k is represented by its location ($C_k.x, C_k.y, C_k.z$) and radius $C_k.r$ where multiple traffic flows may have conflict (e.g. two lanes cross each other or merge into one lane). Compared to the concept of conflicting zones defined based on the grids by most of the existing works, our definition of conflicting points can describe the potential conflicting regions more precisely to accommodate any type of intersections.

Harmonization Zone: As illustrated in Fig. 1, each incoming lane $\mathcal{L}_i$ has a special zone, called Harmonization Zone (H-Zone). Once a vehicle enters the H-Zone, it will keep updating its status with the intersection manager and issue the travel plan query. The length of the H-Zone is determined by the speed limit of the incoming lane to ensure that the intersection management unit will have enough time to plan the traffic for all the incoming vehicles before they enter the dash zone as defined below.

Dash Zone: As presented in Fig. 1, after obtaining the travel plan in H-Zone, the vehicle will enter the Dash Zone (D-Zone). D-Zone is where an incoming vehicle follows the travel plan calculated by the intersection manager. Inside the D-Zone, in adjusting phase, the vehicle will adjust its speed so that it can arrive the Dashing Point (DP) at designated speed, acceleration, and time. After that, in dashing phase, the vehicle will accelerate and arrive at the crossing zone, i.e., the entrance of the intersection, at the planned timing and speed to cross the intersection optimally. The length of the D-Zone is determined by multiple factors including the speed limit, acceleration/deceleration capabilities, intersection capability, and vehicle density.

Crossing Zone: Following the travel plan, the vehicles will leave the D-Zone and enter the Crossing Zone (C-Zone) at the designated time and speed. The C-Zone is the area with most of the conflict points in the intersection. All the possible traffic conflicts must be resolved before vehicle entering the C-Zone.

3.1.2 The Vehicle Status Model . This model captures the dynamic information streamed from incoming vehicles.

Vehicle Characteristics: Different vehicles ($\mathcal{V}_i$) have different characteristics such as the body length $\mathcal{V}_i.l$, maximum acceleration $\mathcal{V}_i.\mathcal{A}_{max}$, maximum deceleration $\mathcal{V}_i.\mathcal{D}_{max}$, and maximum communication range $\mathcal{V}_i.\mathcal{R}_{max}$. Without taking these individual vehicle's characteristics into account, the generated travel plans may not be accurate or could even be dangerous for some vehicles. For example, a travel plan that could safely guide a small sedan to cross the intersection at the right timing is likely to cause a trailer truck to arrive either too early or too late since the trailer truck takes a longer time to reach the desired speed.

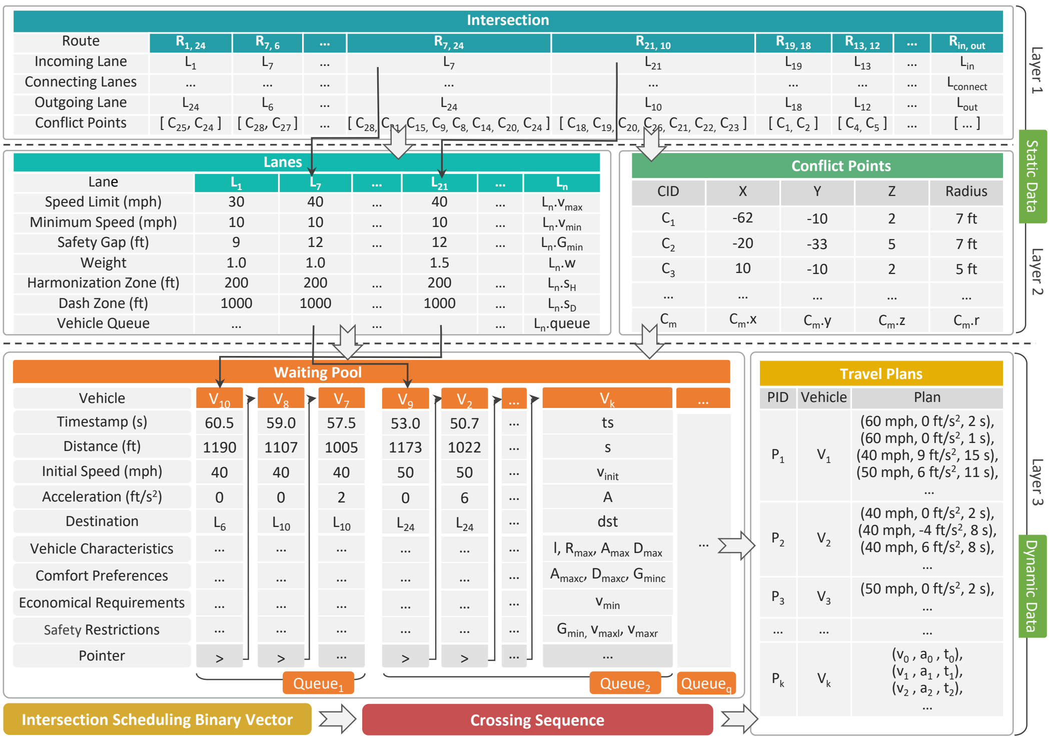


Fig. 4. The Multi-Layer Data Structure for the Intersection Traffic Model (1 ft ≈ 0.3 meter, 1 mph ≈ 1.6 km/h)

Dynamic Status: The dynamic status reflects a vehicle's real-time information such as the distance to the intersection $\mathcal{V}_i.s$, initial speed $\mathcal{V}_i.v_{init}$, and acceleration $\mathcal{V}_i.\mathcal{A}$. Once connected to the intersection management unit, the vehicle will keep updating the dynamic status as soon as it has been changed.

Passenger Comfort Preferences: In the real-life scenario, it is important to consider the comfort of passengers in a vehicle when planning the vehicle's acceleration and deceleration. For example, a car with a baby or pregnant woman inside may prefer lower acceleration/deceleration and a larger safety gap between two vehicles. Emergency vehicles, on the contrary, may want to maximize acceleration and speed. To take into account these real-world constraints, our proposed comprehensive vehicle model allows each vehicle $\mathcal{V}_i$ to be associated with its comfort preferences including maximum comfort acceleration $\mathcal{V}_i.\mathcal{A}_{maxc}$, maximum comfort deceleration $\mathcal{V}_i.\mathcal{D}_{maxc}$, and minimum comfortable gap to the front vehicle $\mathcal{V}_i.\mathcal{G}_{minc}$.

Economical and Environmental Requirements: The fuel consumption and greenhouse gas emission per mile could be quite high when vehicles move at a very low speed. Therefore, our model allows the intersection to define the minimum cruise speed $\mathcal{V}_i.v_{min}$ when scheduling the intersection traffic.

Safety Restrictions: Depending on the types of vehicles, transportation tasks, and related regulations, different vehicles may have different safety restrictions. For example, a big truck with maximum loads should not turn right at a speed faster than 10 mph to avoid side rollover. Therefore, our model captures these restrictions using the following parameters: minimum safety gap $\mathcal{V}_i.\mathcal{G}_{min}$, right turn speed limit $\mathcal{V}_i.v_{maxr}$, and left-turn speed limit $\mathcal{V}_i.v_{maxl}$.

Algorithm 1: REAL-TIME TRAVEL PLANNING

```
while True do
    // Stage 1, Clustering vehicles
    while in Stage 1 do
        WaitingPool ←Receiving requests
        Update multi-layer data structure
    // Stage 2, Dash-zone group motion planning
    Determine CrossingSequence
    foreach v_i in WaitingPool do
        Estimate v_i's position and speed;
        Derive dashing speed v_dash and dashing duration t_dash
        Resolve rear-end collision
    // Stage 3, Intersection conflict resolution
    Resolve conflicts, if conflict detected, go back to stage 2
    Send plans to vehicle, update scheduling vector S_c
    round ← round + 1 // Start next round of travel planning
```

3.1.3 The Multi-Layer Data Structure for the Intersection Traffic Model. We would like to point out that none of the existing works support such a fine-grained intersection traffic model during traffic planning. To efficiently store and manage various information in our traffic model, we design a multi-layer data structure as shown in Fig. 4 (the data in this example is derived from the intersection in Fig. 1).

The top layer of the data structure stores the static information of an intersection including all the possible routes and their corresponding travel restrictions. The detailed route information is stored at the 2nd-layer which includes a lane table and a conflicting point table. Specifically, each route entry is pointing to the specific incoming and outgoing lanes in the lane table which stores the detailed information of each lane as introduced in the previous section. Each route entry also has a pointer that leads to a set of conflicting points stored in the conflicting point table.

The dynamic information regarding incoming vehicles is organized in the 3rd layer of the data structure as vehicle queues. Specifically, each incoming lane has a pointer to a vehicle queue. In each vehicle queue, vehicles are arranged according to their arrival time stamps, i.e., the time entering the harmonization zone. This arrival order ensures that the travel plan requests will be processed in the same sequence to avoid collisions.

3.2 Real-time Travel Planning

The intersection management unit constantly performs traffic planning for vehicles continuously entering the harmonization zone. The optimization goal is to maximize the overall throughput at the intersection while satisfying both intersection and vehicle constraints as discussed in the previous section. The following is the optimization goal.

DEFINITION 1. *For each vehicle $\mathcal{V}_i$, let t_{i_actual} denote the actual time that it needs to cross the intersection, t_{i_ideal} denote the ideal time it takes to cross the intersection assuming there are not any vehicles on its path, and $stop_i$ denote the number of stops it takes. Let n be the total number of vehicles under consideration, the intersection traffic optimization goals are the following:*

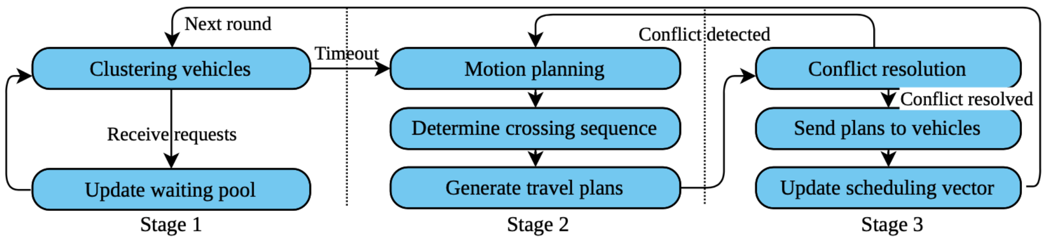


Fig. 5. Real-time Travel Planning

- *The travel plans should optimize the overall throughput by minimizing the traffic delay:*

$$\mathcal{T}_{delay} = Minimize(\sum_{i}^{n}(t_{i\_actual} - t_{i\_ideal}))$$

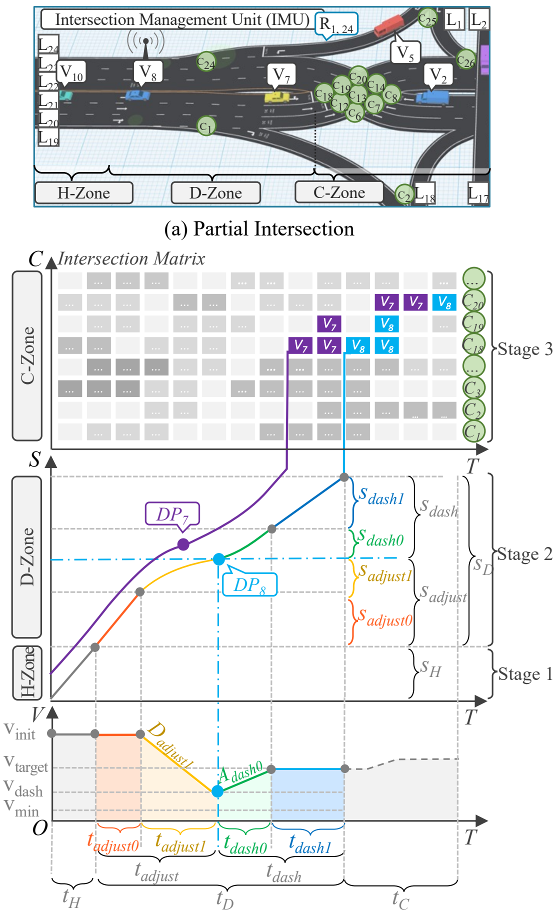
- *The travel plans should satisfy all the vehicles' comfort preferences by minimizing the total number of stops $\sum_i^n(stop_i)$ and restricting the maximum acceleration/ deceleration under the comfort acceleration/deceleration limits.*

As presented in Algorithm 1 and Fig. 5, the traffic planning process consists of the following three stages:

Stage 1 (Clustering vehicles): Once a vehicle enters the harmonization zone, it will start to update its status (e.g., speed, destination, comfort preferences) with the intersection management unit and request a travel plan to cross the intersection. If the intersection manager simply minimizes the crossing time for each vehicle at the moment of its request, it would be hard to achieve global optimization, i.e., maximizing the overall throughput. For example, the vehicle $\mathcal{V}_2$ in Fig. 6 (a) is closer to the intersection than vehicle $\mathcal{V}_7$ and sent the query earlier than $\mathcal{V}_7$. If $\mathcal{V}_2$ was scheduled to pass the intersection first, it would actually slow down multiple vehicles including $\mathcal{V}_7$, $\mathcal{V}_8$, and $\mathcal{V}_{10}$. If $\mathcal{V}_2$ was asked to slow down a little bit and pass after $\mathcal{V}_7$, more vehicles can cross the intersection during the same period of time. Meanwhile, $\mathcal{V}_2$ would not be delayed either since it can accelerate and pass the intersection in the gap of $\mathcal{V}_7$ and $\mathcal{V}_8$.

Therefore, we consider the vehicles entering the harmonization zone during the same period of time as a whole when conducting traffic planning. This is achieved with the aid of the data structure shown in Fig. 4. Specifically, the intersection management unit will insert the new vehicle information into the corresponding vehicle queue. Let τ_s be the timestamp when a vehicle $\mathcal{V}_i$ enters the harmonization zone, and τ_e be the time when $\mathcal{V}_i$ leaves the harmonization zone and enters the dash zone. Then $t = \tau_e - \tau_s$ is the period of time for this round of traffic planning. The next vehicle $\mathcal{V}_j$ who enters the harmonization zone later than τ_e will trigger another round of traffic planning.

Stage 2 (Dash-zone group motion planning): After vehicles are grouped, the intersection management unit will begin to search the optimal travel plans for these vehicles. The travel plan needs to include the details about when, where, and how much a vehicle should accelerate and decelerate to cross the intersection. We have already shown in the previous section that a greedy algorithm that calculates the optimal route for each vehicle at the time of the query will not guarantee the global optimal. A naive solution that can find the global optimal solution is to conduct an exhaustive search that enumerates all possible crossing orders and possible crossing speeds for all the vehicles. This could be very time-consuming and may not even be feasible to provide real-time travel guidance. Another option is to use classical optimal control methods, such as Pontryagin's Minimum principle (PMP) and Dynamic Programming (DP) methods to optimize the travel planning problem.



(a) Partial Intersection

(b) Stages of Travel Plan Optimization

Fig. 6. Three-stage Travel Plan Optimization

However, the PMP algorithm does not consider the comfort levels of passengers. The DP algorithm would be quite complicated and less efficient after taking the large number of constraints. To ensure the efficiency of the travel plan optimization and simplify the computation process, we propose the following new algorithm.

First, we divide the vehicle movement behavior into two phases and introduce a new notion called *Dashing Point*. As illustrated in Fig. 6 (b), the first phase is a adjusting phase (in duration of t_{adj}) whereby a vehicle moves at its initial speed v_{init} for a duration of t_{adj_0} and then adjusts the speed with deceleration $\mathcal{D}_{adj1}(\mathcal{D}_{adj1} < 0)$ for t_{adj_1} in order to reach the dashing point at speed v_{dash} (t_{adj_1} could be zero when v_{init} equals to v_{dash}). The second phase is a dashing phase t_{dash} whereby the vehicle starts dashing with acceleration $\mathcal{A}_{dash0}(\mathcal{A}_{dash0} > 0)$ for t_{dash_0} and then maintains the speed v_{target} for t_{dash_1} until reaching the intersection (t_{dash_0} could be zero when v_{dash} equals to v_{target}). The dashing point is the start of time t_{dash} that a vehicle starts acceleration from speed v_{dash} to reach the intersection at an optimal crossing time window (To increase the fuel consumption and reduce greenhouse gas emission, the dashing speed v_{dash} will not less than minimum cruise speed v_{min}). As shown in Fig. 6 (b), a shorter dashing time t_{dash} at the dashing point DP_7 indicates that vehicle $\mathcal{V}_7$ can move at a higher speed for a longer time when approaching the intersection, which leaves more room for the following vehicle $\mathcal{V}_8$ to adjust its speed. If the acceleration time t_{dash_0} equals to zero, it means that the vehicle can cross the intersection very soon and will not need to decelerate to the speed that is lower than the target speed.

The use of the dashing point offers several advantages. First, it significantly speeds up the search for the desired travel speeds for a vehicle. Unlike prior works that need to search the optimal speed at each timestamp which is inevitably expensive, we can achieve the same optimization goal much more efficiently by only searching a dashing point and calculating the needed speed, deceleration, and acceleration. Second, the definition of the dashing point also helps to address the comfort preferences. It is not comfortable for the riders if vehicle speed and acceleration need to be changed frequently and abruptly to achieve the most efficient crossing time. In our approach, the vehicle will only need to decelerate at most once before the dashing point and accelerate at most once after the dashing point. Moreover, the definition of the dashing point makes it easier to evaluate the congestion at each lane and yield better load balancing. Specifically, a crowded lane will be reflected as little space for dashing point adjustment. When there are too many vehicles in a specific lane, the vehicles that enter the lane afterwards will need to decelerate earlier and reach a lower speed to avoid being too close to the car in front. In this case, the dashing time t_{dash} will be longer and closer to t_D. Once that is detected during the dashing point calculation, vehicles' priorities of that lane can be raised so as to alleviate its congestion. The ability to increase or decrease the crossing priorities of vehicles based on the dashing points also offers the opportunity to achieve global optimization among all lanes connected to the intersection.

Identifying the optimal dashing point for each vehicle is a very challenging task since any acceleration or deceleration of a vehicle may affect the vehicle in front of or behind it. To overcome this challenge and achieve real-time planning, we design a heuristic approach based on the following estimation functions.

As shown in Fig. 6, the traveling distance (s_D) and the duration (t_D) that a vehicle in the adjusting phase and the dashing phase can be calculated by Equation (1).

$$\begin{aligned} s_D &= s_{adj} + s_{dash} = s_{adj_0} + s_{adj_1} + s_{dash_0} + s_{dash_1} \\ t_D &= t_{adj} + t_{dash} = t_{adj_0} + t_{adj_1} + t_{dash_0} + t_{dash_1} \end{aligned} \tag{1}$$

Substituting the accelerated motion equations, the dashing speed v_{dash} can be derived from the following equations.

$$s_D = v_{init}(t_{adj} - \frac{v_{dash} - v_{init}}{\mathcal{D}_{adj1}}) + \frac{v_{dash}^2 - v_{init}^2}{2\mathcal{D}_{adj1}} + \frac{v_{target}^2 - v_{dash}^2}{2\mathcal{A}_{dash0}} + v_{target}(t_{dash} - \frac{v_{target} - v_{dash}}{\mathcal{A}_{dash0}}) \tag{2}$$

$$\begin{aligned} &(\frac{1}{2\mathcal{A}_{dash0}} - \frac{1}{2\mathcal{D}_{adj1}})v_{dash}^2 + (\frac{v_{init}}{\mathcal{D}_{adj1}} - \frac{v_{target}}{\mathcal{A}_{dash0}})v_{dash} - v_{init}(t_D - t_{dash}) - v_{target}t_{dash} + s_D \\ &- \frac{v_{init}^2}{2\mathcal{D}_{adj1}} + \frac{v_{target}^2}{2\mathcal{A}_{dash0}} = Av_{dash}^2 + Bv_{dash} + C = 0 \end{aligned} \tag{3}$$

$$v_{dash_\alpha} = \frac{-B - \sqrt{B^2 - 4AC}}{2A}, v_{dash_\beta} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{4}$$

If $B^2 - 4AC > 0$, we set $c = \frac{v_{dash_\alpha} + v_{dash_\beta}}{2}$ since $A = \frac{1}{2\mathcal{A}_{dash0}} + \frac{1}{2|\mathcal{D}_{adj1}|} > 0$. Then, we have $v_{dash_\alpha} < c < v_{dash_\beta}$. Dividing c by v_{target}, we have the following result:

$$\frac{c}{v_{target}} = \frac{-B}{2Av_{target}} = \frac{|\mathcal{D}_{adj1}|v_{init} + |\mathcal{A}_{dash0}|v_{init}}{|\mathcal{D}_{adj1}|v_{init} + |\mathcal{A}_{dash0}|v_{target}} \tag{5}$$

Since $v_{target} \leq v_{init}$, we have $c \geq v_{target}$. Therefore, $v_{dash_\beta} > v_{target}$. Accordingly, we calculate the dashing speed v_{dash} by the following equation:

$$v_{dash} = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \tag{6}$$

If $B^2 - 4AC = 0$, we can still use Equation (6) to calculate the dash speed v_{dash} since $v_{dash_\alpha} = v_{dash_\beta}$.

According to Equation (6), to obtain the dashing speed v_{dash}, we still need to determine the total duration t_D, the target speed v_{target}, and the duration of the dashing phase t_{dash}. The first step is to search t_D from $t_{min} = \frac{s_D}{v_{init}}$ to $t_{max} = \frac{s_D}{v_{min}}$ with an interval of δ = 0.1 second, and v_{target} from v_{max} to v_{min} with an interval of θ of 1.0 ft/s (0.3 meter/s). After that, we calculate the lower bound t_{dash_min} and the upper bound t_{dash_max} of the dashing time.

When $v_{init} \geq v_{target}$, we have constraints that $t_{adj_0} \geq 0$, and $t_{dash_1} \geq 0$. By substituting the accelerated motion equations, we can derive the lower and upper bounds of the t_{dash}, respectively.

$$\begin{aligned}
A' &= A * \mathcal{A}_{dash0}^2 \\
B' &= -2A * v_{target}\mathcal{A}_{dash0} - B * \mathcal{A}_{dash0} + v_{init} - v_{target} \\
C' &= Av_{target}^2 + Bv_{target} - v_{init}t_D - \frac{v_{init}^2}{2\mathcal{D}_{adj1}} + \frac{v_{target}^2}{2\mathcal{A}_{dash0}} + s_D \\
A'' &= A * \mathcal{D}_{adj1}^2 \\
B'' &= -2A(v_{init} + \mathcal{D}_{adj1}t_D)\mathcal{D}_{adj1} - B\mathcal{D}_{adj1} + v_{init} - v_{target} \\
C'' &= A(v_{init} + \mathcal{D}_{adj1}t_D)^2 + B(v_{init} + \mathcal{D}_{adj1}t_D) - v_{init}t_D - \frac{v_{init}^2}{2\mathcal{D}_{adj1}} + \frac{v_{target}^2}{2\mathcal{D}_{adj1}} + s_D \\
&\frac{-B' + \sqrt{B'^2 - 4A'C'}}{2A'} \leq t_{dash} \leq \frac{-B'' - \sqrt{B''^2 - 4A''C''}}{2A''}
\end{aligned} \tag{7}$$

When $v_{init} > v_{target}$, the v_{dash} will increase or decrease when the t_{dash} increases or decreases. Therefore, we have a constraint: $v_{min} \leq v_{dash} \leq v_{max}$. From Equation (6), we can also obtain the additional boundaries.

$$\begin{aligned}
&\frac{Av_{min}^2 + Bv_{min} + s_D + \frac{v_{target}^2}{2\mathcal{A}_{dash0}} - \frac{v_{init}^2}{2\mathcal{D}_{adj1}} - v_{init}t_D}{v_{target} - v_{init}} \\
&\leq t_{dash} \leq \\
&\frac{Av_{target}^2 + Bv_{target} + s_D + \frac{v_{target}^2}{2\mathcal{A}_{dash0}} - \frac{v_{init}^2}{2\mathcal{D}_{adj1}} - v_{init}t_D}{v_{target} - v_{init}}
\end{aligned} \tag{8}$$

With the lower and upper bounds of t_{dash}, the optimal t_{dash} can be obtained by a binary-search algorithm that maximizes the high-speed cruising duration while avoiding rear-end collisions. Our algorithm performs the rear-end collision check by calculating the distance between the two vehicles based on the travel plans under consideration. It will ensure that the travel plan allows a sufficient safety gap between two vehicles to ensure two vehicles do not bump into each other.

Before finalizing the travel plan for each vehicle, we also conduct an intersection conflict check as presented in Stage 3.

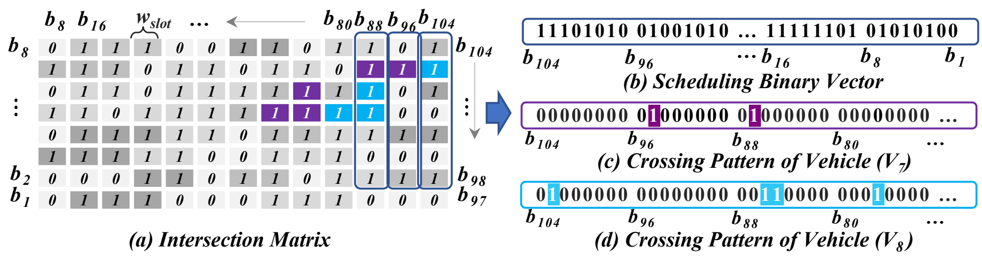


Fig. 7. Intersection Scheduling Binary Vector

Stage 3 (Intersection conflict resolution): As vehicles cross the intersection from different directions, there are conflicting points among vehicles' travel paths. To decide which vehicle occupies which conflicting point at what time not only is important to avoid the collision, but also affect the overall throughput of the traffic at the intersection. For example, a vehicle $\mathcal{V}_i$ that will take a left turn may need to pass a series of conflicting points. If we let this vehicle reserve all the conflicting points until it crosses the intersection, this vehicle could cause many other vehicles to stop and wait for the conflicting points to become available again even when vehicle $\mathcal{V}_i$ has not reached some of the conflicting points yet. Intuitively, if the occupancy of conflicting points is maximized at all times, the intersection traffic is more likely to achieve the maximal throughput. Therefore, our goal is to determine the crossing sequence of the vehicles in the pending list to maximize the occupancy of the conflict points while avoiding collisions.

We model the intersection as a matrix of time and conflicting points. As shown in Fig. 6 (b), stage 3, the x-dimension (T) of the matrix represents the evolving time which is equally divided into time slots (the width of the time slot is defined as w_{slot} and typically set to 0.1s), and the y-dimension (C) of the matrix represents conflicting points. Our model allows an easy representation of when and how long a vehicle occupies a conflicting point. For example, in Fig. 6 (a), vehicle $\mathcal{V}_7$ first passed the conflicting point C_{18} using two time slots and then another C_{19}. After a while, it passed C_{20}.

We then convert the intersection matrix in Fig. 6 (b) to an intersection scheduling binary vector as shown in Fig. 7. The first step is to represent each occupied slot (i.e., occupied conflicting point) on the matrix as 1 and non-occupied slot (i.e., available conflicting point) as 0. Then, we link the binary values in the same column from top to bottom to form a binary string. Next, we assemble the binary strings from all the columns from right to left to form the overall scheduling binary vector. For example, the last column in Fig. 7 (b) can be represented as 11101010 which forms the highest 8 bits of the intersection scheduling binary vector, and the first column can be represented as 01010100 which is the lowest 8 bits of the vector. Similarly, the crossing pattern of vehicle can be represented as binary vector as presented in Fig. 7 (c) and (d). With the aid of the binary vector representations, the intersection management unit can check the conflict by simply performing a bit-wise *AND* operation between the current scheduling binary vector and a specific vehicle's crossing pattern. If the result is not equals to zero, a conflict is detected.

With the conflict detection method in place, we now proceed to discuss our proposed heuristic search algorithm to find a near-optimal crossing sequence for the incoming new vehicles. Let $\mathcal{S}_c$ denote the scheduling binary vector of the current intersection. Let $node_c$ denote the vehicle v_c which moves at ($node_c.depth$)th place with known cost (delay) $node_c.g$ and estimated cost $node_c.h$. In each step, the search algorithm looks for the node (i.e., the vehicle) with the minimum delay cost $node_p.f = node_p.g + node_p.h$ from $OpenList$ and maximizes the use of conflicting points, and then updates the estimated delay cost for the remaining vehicles. After that, we update the $OpenList$

and *CloseList* and start to search the vehicle for next crossing position. Once the optimal crossing sequence is identified, the detailed collision-free travel plan for each vehicle will be generated.

3.3 Discussion on System Deployment in Real-World Scenarios

In this section, we discuss several aspects to be considered if the DASHX system would be deployed in the real world in the future.

Spatial-temporal Uncertainty: As we know, current GPS technology is not able to guarantee 100% spatial-temporal accuracy. Our DASHX system has a nice feature that can avoid potential collisions caused by spatial-temporal uncertainty. This is achieved by simply adjusting the safety gap parameter in the DASHX system to be always larger than the current GPS accuracy range (e.g., 16.4 ft (5 meters) [52]).

Network Communication Delay: Our system is designed to minimize the effect of possible network delay on the overall performance. Specifically, our system does not require vehicles to frequently communicate with the IMU or transmit large packets. We assume that vehicles will not need to change the predetermined destination frequently. Vehicles using our system just needs to send their initial status (less than 100 Bytes) and receive their travel plans (less than 80 Bytes per plan). After that, they will follow their travel plans and do not need to communicate with IMU.

Security and Safety: There is no doubt that security and safety are important aspects to be considered during an autonomous traffic control. Although designing specific security and privacy protection mechanisms is out of the scope of this work, our DASHX is flexible to integrate the existing developed security and privacy protocols as discussed below.

First, the DASHX system can be secured by employing existing vehicular authentication schemes [19, 21, 45] whereby each vehicle authenticates itself with the intersection management unit upon entering the harmonization zone. The authentication process ensures vehicles participating in the scheduling are legitimate vehicles. As this authentication process occurs in the harmonization zone, it will not affect the efficiency of traffic planning. In addition, other security techniques such as trustworthiness evaluation [49] may also be integrated for further protection.

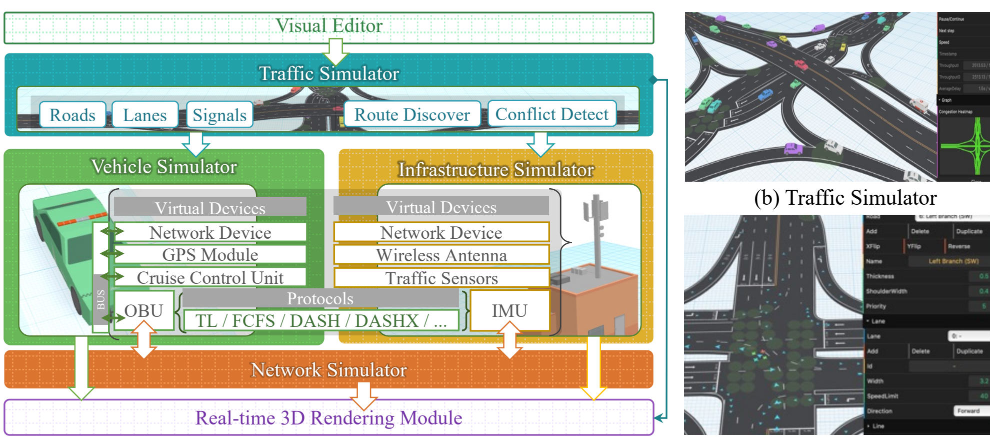
In the case of traffic accidents, the following protocol may be adopted. Vehicles which observed or detected abnormal traffic condition, such as a vehicle that is not cooperative with the travel plan, a vehicle that does not update correct status, or a vehicle with mechanical failure, will report to the intersection manager. The DASHX system is fast enough to recalculate new travel plans in milliseconds, and hence will be able to redirect or evacuate other vehicles.

4 THE PROPOSED DASHX-SIM TRAFFIC SIMULATION PLATFORM

We have developed a sophisticated and interactive traffic simulation platform, namely DASHX-SIM, to help researchers fine tune, compare, and visualize their protocols in complicated traffic situations. As shown in Fig. 8 (a), the DASHX-SIM platform consists of six main components: (i) the *visual editor* supports fine-grained customization of the road maps, vehicles, infrastructures, and traffic phases; (ii) the *traffic simulator* handles complex 2D or 3D intersection layouts and discover routes and conflicts automatically; (iii) the *vehicle simulator* simulates individual vehicle's behavior based on input protocols; (iv) the *infrastructure simulator* is used to support smart sensors and devices that may be added to the infrastructure in the future; (v) the *network simulator* is in charge of vehicle-to-vehicle and vehicle-to-infrastructure communications based on input protocols; (vi) the *3D rendering module* provides real-time traffic visualization at ultra high definition (4K UHD). Compared to other existing simulators as shown in Table 1, our proposed DASHX-SIM traffic simulation platform has more features. Specifically, our simulator is free and open-sourced, supports microscopic simulation on complicated 3D road network, and has visual tools and network environment integrated. Our simulation platform can run on a majority of popular operating

Table 1. Comparison among existing traffic simulation platforms
(* Additional plugins needed)

Simulator	2D	3D	Open Sourced	Microscopic Simulation	Visual Tools	Network Tools	Linux / macOS	Android / iOS / Web
DASHX-SIM	✓	✓	✓	✓	✓	✓	✓	✓
SUMO	✓		✓	✓	✓		✓	
GAMA	✓	✓	✓				✓	
MATSim	*		✓	✓			✓	
Aimsun Next	✓	✓		✓	✓	*	✓	
PTV Vissim	✓	✓		✓	✓	*		



(a) DASHX-SIM System Overview
(b) Traffic Simulator
(c) Map Editor

Fig. 8. DASHX-SIM Traffic Simulation Platform

systems including Windows, MacOS, Linux, iOS, and Android, and on major web browsers. The functions of each component are further elaborated as follows.

(i) Visual Editor: Fig. 8 (a) and (c) illustrate some functions in the visual editor. We can use this editor to create any shape of 2D or 3D intersections, define vehicle shapes and types, and add traffic constraints to lanes. Each lane has customizable characteristics (e.g., width, pavement color, etc.), safety restrictions (e.g., speed limits, safety gap, etc.), and pavement markings (e.g., left-turn arrow, solid yellow lines, etc.). The editor also supports the customization of vehicles with different sizes and characteristics. The editor also allows users to design the infrastructures such as traffic lights, buildings, and signal towers. For traffic light based intersection protocols, the editor also provides a function to define traffic phases to schedule the traffic lights.

(ii) Traffic Simulator: The traffic simulator analyzes the 3D road networks, managing all the entities including roads, lanes, marks, and signals, simulates the movements of the vehicles, and provides interfaces for the vehicle simulator, infrastructure simulator, and network simulator to get the necessary real-time spatio-temporal data of the road network. The traffic simulator also supports automatic route and conflict discovery. It reads a road map and automatically connect road segments and build the data model for the follow-up vehicle simulation.

Table 2. Experimental Parameters
($1\ mph \approx 1.6\ km/h$, $1\ ft \approx 0.3$ meter, $1\ ft/s^2 \approx 0.3\ m/s$)

Parameter	Range	Default Value
Speed limit	$10 \sim 70\ mph$	$50\ mph$
Minimum Cruise Speed	$10\ mph$	$10\ mph$
Maximum Acceleration	$6 \sim 12\ ft/s^2$	$9\ ft/s^2$
Maximum Deceleration	$11 \sim 20\ ft/s^2$	$15\ ft/s^2$
Comfort Acceleration	$5 \sim 10\ ft/s^2$	$6.6\ ft/s^2$
Comfort Deceleration	$9 \sim 15\ ft/s^2$	$10\ ft/s^2$
Minimum Safety Gap	$16 \sim 20\ ft$	$16\ ft$
Vehicle Density	$1800 \sim 12600\ vehicles/hour$	$7200\ vehicles/hour$
Maximum Communication Radius	$1600\ ft$	$1600\ ft$
GPS Accuracy	$8 \sim 32\ ft$	$16\ ft$
Network Delay	$20 \sim 100\ ms$	$50\ ms$

(iii) Vehicle Simulator: The vehicle simulator runs an independent instance for each vehicle to mimic the real-world scenarios. Each vehicle has equipped with a communication bus by default and all the equipped virtual devices can exchange messages through it. If a vehicle has the network device installed, it will become a connected vehicle and has the capability to communicate with either nearby vehicles or infrastructures. The researchers can use the default communication protocol or develop their own. Our simulator provides a platform for the researchers to easily test their routing and traveling protocols.

(iv) Infrastructure Simulator: The infrastructure simulator allows researchers to integrate different kinds of infrastructure into the traffic simulation. An example of infrastructure could be an intersection manager that is equipped with network devices and wireless antennas and manages the traffic. Another common example of infrastructure is the traffic light. Our system supports the scheduling of traffic lights to mimic the real world scenarios. Yet other examples of infrastructure that supported by our system could be a parking lot or surveillance cameras.

(v) Network Simulator: The built-in network simulator supports sending/receiving/forwarding messages among vehicles and intersection management unit through either wired-connections or wireless networks with network delay, transmission rate, and packet loss. It provides interfaces to the virtual network device and ensure that network packets can be exchanged among any requesting entity in the road network.

(vi) Real-time 3D Rendering Module: The real-time 3D rendering module offers vivid traffic visualization for researchers to quickly examine the evolvement or change of traffic flows under their protocols. Our simulation tool also allows researchers to check detailed information such as the traffic congestion level, the noise pollution distribution, and the network topology.

The simulation platform is implemented using Javascript with Three.js 3D library [1] and presented by HTML5. It is an open-source cross-platform simulator.

5 EXPERIMENTAL STUDY

The proposed DASHX system is tested under macOS 10.15 with 3.2 GHz Intel i7 CPU, AMD RX 5500 XT graphic card, and 16 GB memory. We have also created an online video of our system demo which is available at "https://youtu.be/wlka3qiyEuQ".

In our experiments, we evaluate 10 popular types of real-world intersections to demonstrate the generality of our algorithm (details are presented in the next subsection). For each type of

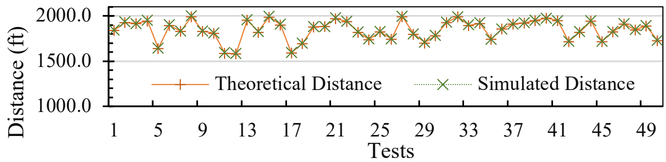


Fig. 9. Simulator Validation

intersection, the traffic flow on each incoming lane is generated by a Poisson distribution using one of the following models: (1) Straight only; (2) All directions; and (3) Unbalanced & all directions. In the straight-only traffic model, we generate δ vehicles for each incoming lane for an hour, and all the vehicles will go straight. In the all-directions traffic model, we also generate δ vehicles for each incoming lane for an hour, while setting the percentages of left-turn, going-straight, and right-turn vehicles to 25%, 50%, and 25%, respectively. In the unbalanced all-direction traffic model, we set one of the routes to cross the intersection as the main road which has δ_p vehicles every hour while other incoming lanes only have δ_s ($\delta_s = \frac{\delta_p}{2}$) vehicles. The percentages of vehicles making left and right turns are set the same as the all-directions model.

Based on the real scenarios, we set the default lane speed limit as 50 mph, left turn speed limit as 30 mph, right turn speed limit as 20 mph, minimum cruise speed as 10 mph, comfort acceleration as 6.6 ft/s^2, and comfort deceleration as 10 ft/s^2. The default length of vehicles is 16 ft which is the common length of a sedan. The vehicle will start communicating with the intersection management unit when the distance is less than 1300 ft.

The overall vehicle density at the intersection is defined as $D = \delta\times$ number of incoming lanes, (or $D = \delta_p\times$ number of principal incoming lanes + $\delta_s\times$ number of the other incoming lanes for unbalanced traffic model). The overall vehicle density ranges from 1800 to 12600 vehicles per hour depending on the specific experiment and the type of intersection.

All the parameters are summarized in Table 2. All of them can be easily adjusted using our DASHX-SIM testbed.

Before testing our algorithms on the simulator, we first conduct a simulator validation to check the simulator's correctness. During the validation, we randomly generate 50 travel plans for vehicles to cross a 4-way 3D diverging diamond interchange as illustrated in Fig. 3 (d) with "All Directions" traffic flow in 30 seconds. The vehicles need to follow the detailed travel plans as well as obeying the speed limits. We compare the difference between theoretical distances and the simulated distances to validate the simulator. As shown in Fig. 9, the simulator can simulate the vehicle movement very closely to the theoretical calculation, which demonstrates the accuracy of the simulation.

5.1 Efficiency Testing

In the first round of experiments, we evaluate the generality and efficiency of our DASHX algorithm under various types of intersections. Note that our approach works for any shape of 2D and 3D intersections, ranging from intersections in old cities to the modern interchanges. Due to the limited space, we tested the following 10 common types of intersections that were built in different cities:

4-Cross (4): A 4-way cross with 4 incoming lanes. This is probably the most popular type of intersections among street blocks in a city.

3-Round (6): A 3-way roundabout with 6 incoming lanes (Fig. 3 (a)).

4-Cross (10): A regular 4-way cross with 10 incoming lanes (Fig. 3 (b)).

4-Staggered (12): A 4-way staggered intersection with 12 incoming lanes (Fig. 3 (c)) in Seattle. This type of intersection has more complicated layout than a regular 4-way cross.

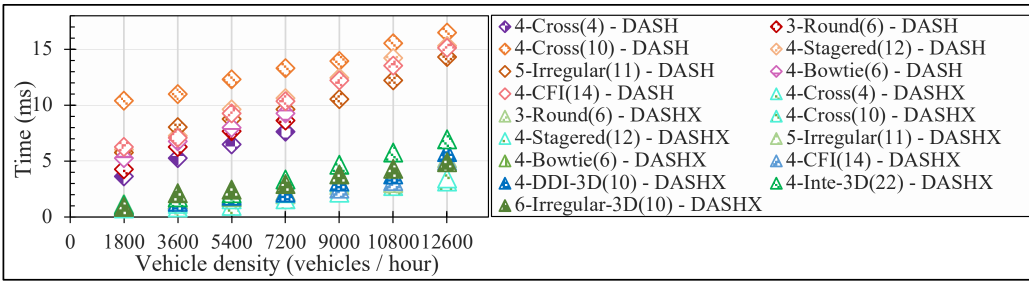


Fig. 10. Average Query Time with Various Vehicle Densities

5-Irregular (11): A 5-way irregular intersection with 10 incoming lanes (Fig. 3 (e)) in London. This is a type of narrow and labyrinthine intersection in old cities.

4-Bowtie (6): A 4-way bowtie intersection with 6 incoming lanes (Fig. 3 (g)) in the city of Charlotte, North Carolina. With two roundabouts and a four-way intersection being combined, it brings new challenges to the design of traffic planning protocol.

4-CFI (14): A 4-way continuous flow intersection with 14 incoming lanes (Fig. 3 (h)) in Miami. This intersection typically has more lanes and involving vehicles with larger variation of speeds.

4-DDI-3D (10): A 4-way, 2-layer diverging diamond interchange with 10 incoming lanes (Fig. 3 (d)) in Kansas City .

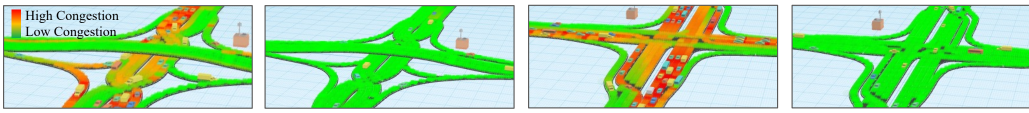
4-Int-3D (22): A 4-way, 3-layer interchange with 22 incoming lanes (Fig. 3 (f)) in Beijing. Such type of intersections impose more calculation workload on traffic planning.

6-Irregular-3D (10): A 6-way, 2-layer interchange with 10 incoming lanes (Fig. 3 (i)) in Melbourne. To the best of our knowledge, none of the existing works can handle such complicated intersection for travel plan optimization.

It is worth noting that most of the intersections can be evaluated thanks to the comprehensive design of our proposed DASHX-SIM simulator that can integrate various supporting modules and provide functionalities to adjust parameters visually and interactively. We test the DASHX system with vehicle density ranging from 1800 to 12600 per hour and record the average time for each query. Fig. 10 shows the experimental results. We can observe that the travel planning time of the DASHX algorithm is significantly less than our prior DASH algorithm. When the vehicle density reaches 12600 vehicles per hour, the DASHX algorithm is 60% faster than the DASH algorithm. This should attribute to the improved dash-zone group motion planning algorithm that effectively reduces the time for searching the dashing speed. We also observe that the query time increases with vehicle density. This is because the intersection schedule will be more crowded when there are more vehicles, and hence more rounds of optimizations are needed. However, it is worth noting that DASHX needs less than **7 milliseconds** to generate a travel plan even when the vehicle density is extremely high, i.e., 12600 vehicles crossing the intersection per hour. This demonstrates that our proposed DASHX algorithm is highly efficient and capable of meeting the stringent real-time requirement in the real world scenarios.

5.2 Effectiveness Testing

In the second round of experiments, we evaluate the effectiveness of our DASHX algorithm in terms of the ability to optimize traffic throughput, reduce traffic delay, and meeting comfort preferences. Recall that the traffic delay of a vehicle is defined as the difference between the actual time it takes to pass the intersection and the ideal time that it needs to cross the intersection when there is no other vehicle in its route.



(a) 4-DDI-3D (10) - TL-10 (b) 4-DDI-3D (10) - DASHX (c) 4-CFI (14) - TL-10 (d) 4-CFI (14) - DASHX

Fig. 11. Traffic Congestion Heat Maps

Table 3. The Traffic Throughput (Vehicles / Hour), Average Delay and Max Delay

Models	D = 1440 vehicles/hour	D = 3600 vehicles/hour	D = 7200 vehicles/hour
TL-5	1380, 2.1s, 4.1s	3534, 3.2s, 5.8s	5904, 53.2s, 83.8s
TL-10	1380, 3.8s, 7.7s	3522, 5.2s, 9.0s	5790, 57.8s, 92.2s
MP-IP	1380, 0.4s, 1.0s	3540, 1.9s, 3.6s	6138, 31.2s, 50.9s
AMP-IP	1380, 0.3s, 0.9s	3540, 1.2s, 2.4s	6594, 20.5s, 33.7s
DCRCAV	1380, 0.2s, 0.5s	3540, 0.5s, 1.2s	6834, 11.4s, 18.4s
DASHX	**1436, 0.04s, 0.5s**	**3600, 0.2s, 1.1s**	**7182, 0.35s, 2.3s**
3D-Ideal	1440, 0.0s, 0.0s	3600, 0.0s, 0.0s	7200, 0.0s, 0.0s

We have compared our DASHX algorithm with 4 existing approaches including: (i) the conventional fixed traffic light scheduling algorithm (denoted as TL-5 and TL-10); (ii) MP-IP (Maximum Progression Intersection Protocol) [3]; (iii) AMP-IP (Advanced Maximum Progression Intersection Protocol) [3]; and (iv) the most recent approach DCRCAV (a Distributed Conflict Resolution for Connected Autonomous Vehicles) [30]. Since none of the previous works evaluated complicated intersections as shown in our previous experiments, we adopt the same traffic settings in the latest work [30] when testing our DASHX algorithm and then compare our results directly with their reported results of all the four approaches.

Specifically, in [30], experiments were conducted on a 4-way cross intersection with 4 incoming lanes and 4 outgoing lanes. The speed limit is set to 36.9 ft/s (11.25 m/s) and the vehicle's deceleration is set to 8.86 ft/s^2 (2.7 m/s^2) which are equal to the average speed and deceleration of the vehicles in [30]. The vehicle density is ranging from 1440 to 7200 vehicles per hour. The traffic flow is using the 'All Directions' model.

Fig. 11 compares traffic congestion at a 4-way diverging diamond interchange and a 4-way continuous flow intersection under traditional traffic light with that under our DASHX algorithm. The vehicle density is 7200 vehicles per hour. We can observe that the traffic flow is much smoother when the DASHX algorithm is taking effect whereby green color denotes lower vehicle density while red color denotes vehicle congestion. Table 3 further compares the throughput obtained from the four approaches, our DASHX algorithm, and a 3D-Ideal case. The 3D-Ideal case is when all the vehicles can travel towards their destinations without being stopped, slowed, or affected by other vehicles. From the table, we can see that our DASHX algorithm achieves near-optimal throughput, i.e., very close to the 3D ideal model. For example, when vehicle density equals 7200 vehicles per hour, the throughput of our DASHX system is as high as 7182 vehicles per hour.

Table 3 also compares the average traffic delay and maximum traffic delay at the intersection. When density D = 1440, the average traffic delay of our proposed DASHX system is only 0.04s, and the maximum delay is 0.5s, which is much better than other approaches. With the increase of vehicle density, the gap continues to widen. When more vehicles pass the intersection, e.g., D = 7200, our DASHX system can achieve an average delay of 0.35s, and maximum delay 2.3s, which is significantly shorter than the latest approach DCRCAV (11.4s on average and 18.4s for

Table 4. Number of Stops (1 Hour)

Models	D = 2900 vehicles/hour	D = 7200 vehicles/hour	D = 9000 vehicles/hour
MILP approach	13	N/A	N/A
DASHX	0	0	0

Table 5. Type of Vehicles (1 ft ≈ 0.3 meter)

Model	$\mathcal{A}_{max}$	$\mathcal{A}_{maxc}$	$\mathcal{D}_{max}$	$\mathcal{D}_{maxc}$	Vehicle's Body Length	$\mathcal{G}_{min}$
Sedan	$12ft/s^2$	$10ft/s^2$	$20ft/s^2$	$15ft/s^2$	$15ft$	$10ft$
SUV	$12ft/s^2$	$10ft/s^2$	$18ft/s^2$	$15ft/s^2$	$17ft$	$15ft$
Van	$11ft/s^2$	$9ft/s^2$	$16ft/s^2$	$14ft/s^2$	$18ft$	$18ft$
BUS	$9ft/s^2$	$8ft/s^2$	$13ft/s^2$	$12ft/s^2$	$23ft$	$20ft$
Truck	$6ft/s^2$	$5ft/s^2$	$11ft/s^2$	$9ft/s^2$	$30ft$	$20ft$

maximum delay). Such a good performance of DASHX is attributed to our real-time travel plan query algorithm that optimizes the traffic in a way that nearly all the vehicles can cross the intersection without any stop. The weaving, non-stopping traffic flow greatly improved the throughput while significantly reduced the traffic delay of individual vehicles.

Besides the aforementioned four existing approaches, we also compare DASHX with another recent existing work which considers the environmental aspect [12] during the traffic scheduling. In [12], a Mixed-Integer Linear Programming (MILP) based approach is proposed with the goal to minimize the number of stops to reduce greenhouse gas emission. Table 4 reports the number of vehicles that need to stop among all the vehicles crossing the intersection. From the table, we can see that the travel plans generated by our DASHX algorithm enable all the vehicles to cross the intersections without any stop whereas the MILP approach still requires 13 vehicles to stop. The MILP work only reports the result when the vehicle density is 2900 vehicles/hour, and hence the number of stops under higher vehicle density is not applicable. As we can see, our DASHX algorithm yields consistent performance (i.e., zero stops) even when the vehicle density reaches 9000 vehicles/hour.

In the subsequent experiments, we report only our DASHX algorithm's performance since none of the existing approaches considers the following 3D scenarios. As listed in Table 5, we generate the traffic flows containing different vehicle models with various parameters that mimic the real-world scenarios. As vehicles will be able to cross the intersection without stopping using our scheduling, we set a relatively higher approaching speed and leaving speed, i.e., 50 mile/hour, and a higher intersection crossing speed, i.e., 40 mile/hour, than the scheduling under traffic light controls.

Fig. 12 reports the performance at a 4-way 2-layer diverging diamond interchange (DDI) and a 4-way 3-layer interchange (3IC). Observe that, under different traffic flow models (straight only, all directions, and unbalanced all directions), the throughput of our DASHX algorithm is always very close to the ideal case, which again proves that our DASHX algorithm can effectively generate near-optimal travel plans for vehicles. In Fig. 13, and 14, with the increasing of the vehicle density, the average delay and maximum delay increase slightly. This is because, with a higher density, it is more likely for a vehicle to reduce the speed and search for another crossing opportunity. With the help of crossing order optimization in the DASHX algorithm, the increase of the travel delay is well controlled. Even with 12600 vehicles/hour, the average traffic delay is still less than 0.9s and the maximum delay is less than 4.6s, which will be almost unnoticeable to drivers and passengers.

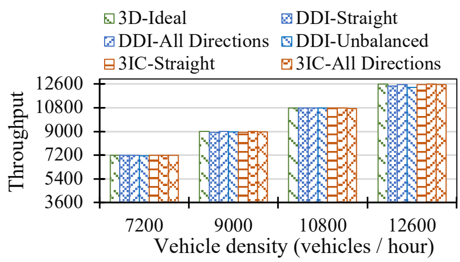


Fig. 12. Traffic Throughput

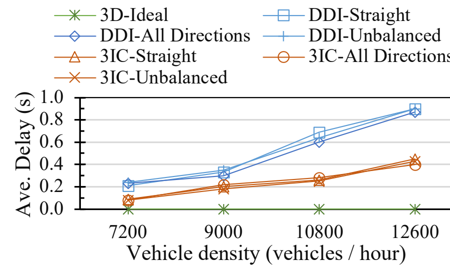


Fig. 13. Average Traffic Delay

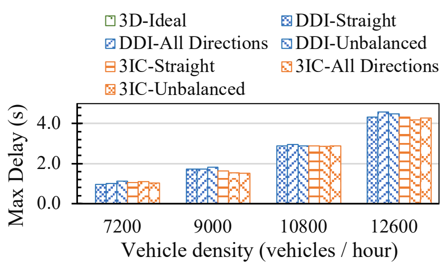


Fig. 14. Maximum Traffic Delay

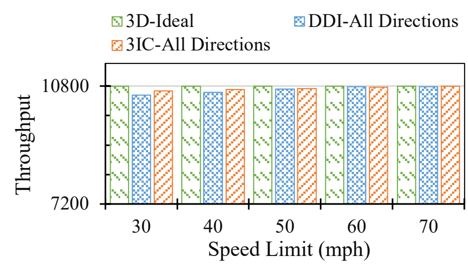


Fig. 15. Traffic Throughput with Various Speeds

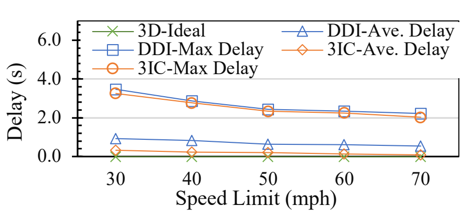


Fig. 16. Traffic Delay with Various Speeds

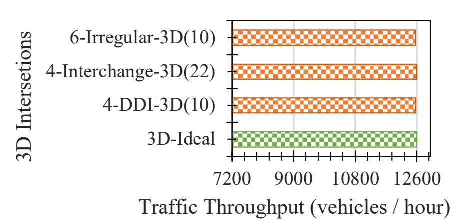


Fig. 17. Traffic Throughput at Different 3D Intersections

Next, we evaluate the effect of speed limits on performance. We increase the speed limits of all the lanes from 30 mph to 70 mph at a 6-way, 2-layer interchange presented in Fig. 3 (i) with 'All Directions' traffic flow. The results are shown Fig. 15 and Fig. 16. Since the throughput is very close to the 3D-Ideal case, we take a further look by zooming in the y-axis. We can see that the throughput actually increases with the increase of the speed limits. As for the traffic delay, both the average delay and maximum delay decreases with the increase of the speed limits. This is because the higher the speed limit, the faster the vehicle can cross the intersection. The results demonstrate that our proposed DASHX algorithm works very well under various speed settings, especially for high-speed limits.

We also evaluate the traffic optimization performance of our DASHX algorithm at different shapes of 3D intersections. As shown in Fig. 17, the traffic throughput achieved by the DASHX algorithm is very close to the optimal 3D ideal case for all the intersections. Especially for the 4-way 3D interchange with 22 incoming lanes, the DASHX algorithm performs the same as the 3D ideal case. This is attributed to our sophisticated travel plan query algorithm.

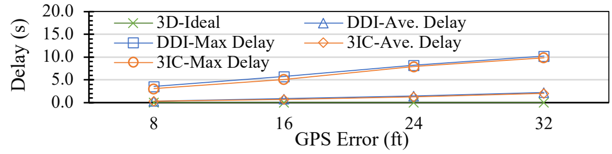


Fig. 18. Traffic Throughput with Various GPS Errors

Finally, we evaluate the effect of GPS positioning errors on performance. We vary the GPS positioning errors from 8 ft to 32 ft with "All Directions" traffic flow on a 4-way 3D diverging diamond interchange. As shown in Fig. 18, with the increase of the GPS positioning error, the average delay and maximum delay increase. This is because we need to adjust the safety gap to make sure it is always larger than the GPS error which affects the overall throughput. It is worth noting that even when the GPS error reaches 32 ft, the average traffic delay is still less than 2s.

To sum up, our DASHX algorithm achieves **zero stop** for all the vehicles in all the experiments tested so far. More importantly, our DASHX algorithm achieves such performance while satisfying the various vehicle constraints (e.g., comfort preferences) which have never been considered altogether in any other previous work. This indicates the generality and potential of the DASHX algorithm to be adopted in various traffic intersections in the real world.

6 CONCLUSION

In this paper, we propose a novel traffic planning system namely DASHX to assist autonomous vehicles to cross the intersection without any stop. The DASHX system is unique as it includes a universal traffic model that works for any type of 3D intersection. Moreover, it is the first time, to the best of our knowledge, a comprehensive set of vehicle parameters and travel constraints are considered simultaneously during the travel plan generation. We have conducted extensive experiments to demonstrate the superiority of the proposed DASHX against the state of the art. It is believed that our DASHX algorithm can significantly increase urban transportation efficiency, promote road safety, and reduce greenhouse gas emissions while taking passenger comfort into consideration. In addition, we also developed an open-source comprehensive 3D traffic simulation platform (DASHX-SIM) that provides a wide range of functionalities to facilitate the research efforts in the realm of autonomous vehicle management.

REFERENCES

[1] 2020. ThreeJS: JavaScript 3D library. https://threejs.org.

[2] Shunsuke Aoki and Ragunathan Rajkumar. 2019. V2V-based synchronous intersection protocols for mixed traffic of human-driven and self-driving vehicles. In *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 1–11.

[3] Seyed Reza Azimi, Gaurav Bhatia, Ragunathan Raj Rajkumar, and Priyantha Mudalige. 2013. Reliable intersection protocols using vehicular networks. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*. ACM, 1–10.

[4] Masoud Bashiri, Hassan Jafarzadeh, and Cody H Fleming. 2018. PAIM: Platoon-based Autonomous Intersection Management. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 374–380.

[5] Alessandro Bazzi, Giammarco Cecchini, Michele Menarini, Barbara M Masini, and Alberto Zanella. 2019. Survey and perspectives of vehicular Wi-Fi versus sidelink cellular-V2X in the 5G era. *Future Internet* 11, 6 (2019), 122.

[6] Alessandro Bazzi, Alberto Zanella, and Barbara M Masini. 2016. A distributed virtual traffic light algorithm exploiting short range V2V communications. *Ad Hoc Networks* 49 (2016), 42–57.

[7] Jordi Casas, Jaime L Ferrer, David Garcia, Josep Perarnau, and Alex Torday. 2010. Traffic simulation with aimsun. In *Fundamentals of traffic simulation*. Springer, 173–232.

[8] Aaditya Prakash Chouhan and Gourinath Banda. 2018. Autonomous intersection management: A heuristic approach. *IEEE Access* 6 (2018), 53287–53295.

[9] Caliper Corporation. 2020. TransModeler Traffic Simulation Software. http://www.caliper.com/transmodeler/.

[10] Sarah Crow. 2018. You'll Spend This Much of Your Life Waiting at Red Lights. https://bestlifeonline.com/red-lights/.

[11] Azise Oumar Diallo, Guillaume Lozenguez, Arnaud Doniec, and René Mandiau. 2021. Comparative Evaluation of Road Traffic Simulators based on Modeler's Specifications: An Application to Intermodal Mobility Behaviors.. In *ICAART (1)*. 265–272.

[12] Seyed Alireza Fayazi and Ardalan Vahidi. 2018. Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing. *IEEE Transactions on Intelligent Vehicles* 3, 3 (2018), 287–299.

[13] Martin Fellendorf and Peter Vortisch. 2010. Microscopic traffic flow simulator VISSIM. In *Fundamentals of traffic simulation*. Springer, 63–93.

[14] Hongwei Ge, Yumei Song, Chunguo Wu, Jiankang Ren, and Guozhen Tan. 2019. Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control. *IEEE Access* 7 (2019), 40797–40809.

[15] Bilal Ghazal, Khaled ElKhatib, Khaled Chahine, and Mohamad Kherfan. 2016. Smart traffic light control system. In *2016 third international conference on electrical, electronics, computer engineering and their applications (EECEA)*. IEEE, IEEE, Beirut, Lebanon, 140–145.

[16] Qiangqiang Guo, Li Li, and Xuegang Jeff Ban. 2019. Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies* 101 (2019), 313–334.

[17] Yi Guo, Jiaqi Ma, Chenfeng Xiong, Xiaopeng Li, Fang Zhou, and Wei Hao. 2019. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation research part C: emerging technologies* 98 (2019), 54–72.

[18] Borut Jereb, Samo Kumperščak, and Tadej Bratina. 2018. The impact of traffic flow on fuel consumption increase in the urban environment. *Fme Transactions* 46, 2 (2018), 278–284.

[19] Jian Kang, Yousef Elmehdwi, and Dan Lin. 2017. Slim: Secure and lightweight identity management in vanets with minimum infrastructure reliance. In *International Conference on Security and Privacy in Communication Systems*. Springer, 823–837.

[20] Jian Kang and Dan Lin. 2020. DASH: A Universal Intersection Traffic Management System for Autonomous Vehicles. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 89–99.

[21] Jian Kang, Dan Lin, Wei Jiang, and Elisa Bertino. 2018. Highly efficient randomized authentication in VANETs. *Pervasive and Mobile Computing* 44 (2018), 31–44.

[22] Yuri G Karpov. 2004. Anylogic: A new generation professional simulation tool. In *VI International Congress on Mathematical Modeling, Nizni-Novgorog, Russia*.

[23] Ajmal Khan, Farman Ullah, Zeeshan Kaleem, Shams Ur Rahman, Hafeez Anwar, and You-Ze Cho. 2018. EVP-STC: Emergency vehicle priority and self-organising traffic control at intersections using Internet-of-things platform. *IEEE Access* 6 (2018), 68242–68254.

[24] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. 2012. Recent development and applications of SUMO-Simulation of Urban MObility. *International journal on advances in systems and measurements* 5, 3&4 (2012).

[25] Phil LeBeau. 2019. Traffic jams cost US $87 billion in lost productivity in 2018, and Boston and DC have the nation's worst. https://www.cnbc.com/2019/02/11/americas-87-billion-traffic-jam-ranks-boston-and-dc-as-worst-in-us.html.

[26] Yongfu Li, Chuancong Tang, Kezhi Li, Srinivas Peeta, Xiaozheng He, and Yibing Wang. 2018. Nonlinear finite-time consensus-based connected vehicle platoon control under fixed and switching communication topologies. *Transportation Research Part C: Emerging Technologies* 93 (2018), 525–543.

[27] Zhenning Li, Qiong Wu, Hao Yu, Cong Chen, Guohui Zhang, Zong Z Tian, and Panos D Prevedouros. 2019. Temporal-spatial dimension extension-based intersection control formulation for connected and autonomous vehicle systems. *Transportation Research Part C: Emerging Technologies* 104 (2019), 234–248.

[28] Yi-Ting Lin, Hsiang Hsu, Shang-Chien Lin, Chung-Wei Lin, Iris Hui-Ru Jiang, and Changliu Liu. 2019. Graph-based modeling, scheduling, and verification for intersection management of intelligent vehicles. *ACM Transactions on Embedded Computing Systems (TECS)* 18, 5s (2019), 1–21.

[29] Bing Liu, Qing Shi, Zhuoyue Song, and Abdelkader El Kamel. 2019. Trajectory planning for autonomous intersection management of connected vehicles. *Simulation Modelling Practice and Theory* 90 (2019), 16–30.

[30] Changliu Liu, Chung-Wei Lin, Shinichi Shiraishi, and Masayoshi Tomizuka. 2018. Distributed conflict resolution for connected autonomous vehicles. *IEEE Transactions on Intelligent Vehicles* 3, 1 (2018), 18–29.

[31] Jianqi Liu, Jiafu Wan, DY Jia, Bi Zeng, Di Li, Ching-Hsien Hsu, and Haibo Chen. 2017. High-efficiency urban-traffic management in context-aware computing and 5G communication. *IEEE Communications Magazine* 55, 1 (2017), 34–40.

[32] Alejandro Ivan Morales Medina, Falco Creemers, Erjen Lefeber, and Nathan van de Wouw. 2019. Optimal access management for cooperative intersection control. *IEEE Transactions on Intelligent Transportation Systems* 21, 5 (2019), 2114–2127.

[33] Amir Mirheli, Mehrdad Tajalli, Leila Hajibabai, and Ali Hajbabaie. 2019. A consensus-based distributed trajectory control in a signal-free intersection. *Transportation research part C: emerging technologies* 100 (2019), 161–176.

[34] Rasool Mohebifard and Ali Hajbabaie. 2019. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. *Transportation Research Part B: Methodological* 121 (2019), 252–274.

[35] Kapileswar Nellore and Gerhard P Hancke. 2016. Traffic management for emergency vehicle priority based on visual sensing. *Sensors* 16, 11 (2016), 1892.

[36] Florent Perronnet, Abdeljalil Abbas-Turki, and Abdellah El Moudni. 2013. A sequenced-based protocol to manage autonomous vehicles at isolated intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 1811–1816.

[37] Xiangjun Qian, Jean Gregoire, Fabien Moutarde, and Arnaud De La Fortelle. 2014. Priority-based coordination of autonomous and legacy vehicles at intersection. In *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 1166–1171.

[38] Amin Rezaei and Brian Caulfield. 2020. Simulating a Transition to Autonomous Mobility. *Simulation Modelling Practice and Theory* (2020), 102175.

[39] Maradona Rodrigues, Andrew McGordon, Graham Gest, and James Marco. 2018. Autonomous Navigation in Interaction-Based Environments—A Case of Non-Signalized Roundabouts. *IEEE Transactions on Intelligent Vehicles* 3, 4 (2018), 425–438.

[40] Mustapha Saidallah, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. 2016. A comparative study of urban road traffic simulators. In *MATEC Web of Conferences*, Vol. 81. EDP Sciences, 05002.

[41] Carlos Renato Storck and Fátima Duarte-Figueiredo. 2020. A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated With Vehicle-to-Everything Communications by Internet of Vehicles. *IEEE Access* 8 (2020), 117593–117614.

[42] Remi Tachet, Paolo Santi, Stanislav Sobolevsky, Luis Ignacio Reyes-Castro, Emilio Frazzoli, Dirk Helbing, and Carlo Ratti. 2016. Revisiting street intersections using slot-based systems. *PloS one* 11, 3 (2016).

[43] Hans van Lint, Wouter Schakel, Guus Tamminga, Peter Knoppers, and Alexander Verbraeck. 2016. Getting the Human Factor into Traffic Flow Models: New Open-Source Design to Simulate Next Generation of Traffic Operations. *Transportation Research Record* 2561, 1 (2016), 25–33.

[44] Kay W Axhausen, Andreas Horni, and Kai Nagel. 2016. *The multi-agent transport simulation MATSim*. Ubiquity Press.

[45] Yimin Wang, Hong Zhong, Yan Xu, and Jie Cui. 2016. ECPB: Efficient Conditional Privacy-Preserving Authentication Scheme Supporting Batch Verification for VANETs. *IJ Network Security* 18, 2 (2016), 374–382.

[46] Yuanyuan Wu, Haipeng Chen, and Feng Zhu. 2019. DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies* 103 (2019), 246–260.

[47] Biao Xu, Shengbo Eben Li, Yougang Bian, Shen Li, Xuegang Jeff Ban, Jianqiang Wang, and Keqiang Li. 2018. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transportation Research Part C: Emerging Technologies* 93 (2018), 322–334.

[48] Huile Xu, Yi Zhang, Li Li, and Weixia Li. 2019. Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[49] Attila Altay Yavuz, Anand Mudgerikar, Ankush Singla, Ioannis Papapanagiotou, and Elisa Bertino. 2017. Real-time digital signatures for time-critical networks. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2627–2639.

[50] Chunhui Yu, Yiheng Feng, Henry X Liu, Wanjing Ma, and Xiaoguang Yang. 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological* 112 (2018), 89–112.

[51] Chunhui Yu, Weili Sun, Henry X Liu, and Xiaoguang Yang. 2019. Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods. *Transportation research part B: methodological* 122 (2019), 416–435.

[52] Paul A Zandbergen and Sean J Barbeau. 2011. Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones. *The Journal of Navigation* 64, 3 (2011), 381–399.

[53] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*. 3620–3624.

[54] Rusheng Zhang, Frank Schmutz, Kyle Gerard, Aurélicn Pomini, Louis Basseto, Sami Ben Hassen, Akihiro Ishikawa, Inci Ozgunes, and Ozan Tonguz. 2018. Virtual traffic lights: System design and implementation. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 1–5.