

Time-Dependent Lane-Level Navigation With Spatiotemporal Mobility Modeling Based on the Internet of Vehicles

Lien-Wu Chen[✉], Senior Member, IEEE, and Chih-Cheng Tsao

Abstract—In this article, we propose a time-dependent lane-level navigation (TDLN) framework with spatiotemporal mobility modeling based on the Internet of Vehicles (IoV). The proposed TDLN framework can provide drivers with the fastest navigation path that can avoid passing congestion areas and predict vehicle spatiotemporal mobility of future traffic flows by estimating the travel time of road segments and the waiting time of intersections. According to our review of relevant research, TDLN is the first lane-level navigation solution that can provide the following features: 1) it can navigate vehicles in a lane-level manner and classify the queuing state of each vehicle as passing through an intersection; 2) it can estimate the driving time of lanes and the stopping time of intersections in different lanes to calculate the total delay time of passing through each lane and intersection; and 3) it can predict future traffic flows to determine the congestion level of each lane and explore predicted flow conditions on the road network to achieve the fastest navigation path planning. Simulation results show that TDLN outperforms existing methods and can plan the lane-level navigation path with the shortest travel time.

Index Terms—Future flow prediction, Internet of Vehicles (IoV), lane-level navigation, path planning, spatiotemporal mobility.

I. INTRODUCTION

Rapid development in vehicular communication and mobile computing technologies has made the Internet of Vehicles (IoV) possible. Through onboard units (OBU) in connected vehicles, vehicle-to-vehicle (V2V), vehicle-to-roadside (V2R), and vehicle-to-infrastructure communication are enabled to exchange emergency messages, traffic conditions, and multimedia data for improving road safety, driving efficiency, and in-vehicle infotainment, respectively [1]. Novel IoV-based applications and systems have been developed for cooperative rear-end collision avoidance [2], highway

Received 15 December 2023; revised 27 June 2024; accepted 6 September 2024. Date of publication 30 September 2024; date of current version 20 November 2024. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant 109-2221-E-035-062-MY3, Grant 112-2221-E-035-032, and Grant 113-2221-E-035-073-MY3. This article was recommended by Associate Editor D. Djenouri. (*Corresponding author: Lien-Wu Chen*)

The authors are with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan (e-mail: lwuchen@fcu.edu.tw).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2024.3462469>.

Digital Object Identifier 10.1109/TSMC.2024.3462469

lane selection assistance [3], driver behavior monitoring and warning [4], and cooperative fleet cruise control [5].

For driving efficiency, time-efficient navigation is necessary to alleviate the time and energy wastage of driving/commuting in modern cities. The 2021 Global Traffic Scorecard of INRIX [6] shows that the American driver lost 36 h averagely due to traffic congestion, which is costing \$564 in wasted time. Moreover, studies of the TomTom Traffic Index [7] show that Istanbul in Turkey is the most congested city with traffic jams in 2021, which causes 62% additional traveling time. In particular, properly-planned navigation paths can prevent commuters/drivers from passing congested/dangerous roads, which can reduce moving time [8] or improve commuting safety [9].

However, existing navigation methods do not simultaneously consider driving times in different lanes, blocking effects of front vehicles, delay times caused by traffic signals, crossing times at intersections, and traffic conditions of future flows. In this study, we design a time-dependent lane-level navigation (TDLN) framework with spatiotemporal mobility modeling based on IoV communications. Through lane-level positioning techniques [10], [11], [12], [13], the fastest navigation path with the shortest travel time can be provided to vehicles requesting TDLN services in a lane-level manner, which can estimate the travel time of lanes and predict the waiting time of intersections. Based on the predicted traffic conditions of future flows, the requesting vehicles can avoid passing slowly-moving lanes and long-waiting intersections at future time points as arriving.

Table I shows a comparison of the features provided by existing navigation methods [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] (discussed in Section II) with our TDLN framework. TDLN offers the most complete solution to the TDLN problem according to whether the navigation path is planned by considering 1) the driving speed of each road segment (in addition to the distances of road segments); 2) the number of vehicles on road segments or at intersections; 3) traffic conditions at future time points as arriving; 4) the delay time caused by traffic signals and stopping vehicles in the front; 5) the times to pass through an intersection for left-turning, go-straight, and right-turning vehicles; and 6) the proper lane to be navigated on each road segment for travel time minimization.

According to our review of relevant research, TDLN is the first lane-level navigation solution that can provide the following features.

TABLE I
COMPARISON OF PRIOR WORKS [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] AND OUR TDLN FRAMEWORK

Features	Driving speed awareness	Vehicle density awareness	Future flow prediction	Traffic signal consideration	Crossing time estimation	Lane-level navigation
Reference [14]	✓					
Reference [15], [16]	✓		✓			
Reference [17], [18]	✓	✓				
Reference [19], [20], [21], [22], [23]	✓	✓	✓			
Our Framework	✓	✓	✓	✓	✓	✓

- 1) The requesting vehicles are navigated in a lane-level manner and the queuing state of each vehicle is classified as passing through an intersection.
- 2) The driving time of lanes and the stopping time of intersections in different lanes are estimated to calculate the total delay time of passing through each lane and intersection.
- 3) The future traffic flows are predicted to determine the congestion level of each lane and the predicted flow conditions on the road network are explored to achieve the fastest navigation path planning.

The contributions of our framework are four-fold. First, the travel time of each lane and the waiting time of each intersection can be precisely estimated based on traffic flow distribution and traffic signal scheduling. Second, the required times of leftmost, middle, and rightmost lanes to cross an intersection can be separately calculated considering the delay caused by traffic signals. Third, the traffic condition of a specific intersection at future time points can be accurately predicted to determine the actual travel time as the requesting vehicle arriving at the specific intersection. Finally, the fastest lane-level navigation path with the shortest travel time at future time points can be properly planned based on the estimated passing times of all road segments and intersection. Furthermore, the real road network and traffic light scheduling of Taichung City in Taiwan is used to verify the feasibility and superiority of our framework. In particular, simulation results show that TDNL outperforms existing methods and can plan the lane-level navigation path with the shortest travel time.

The remainder of this article is organized as follows. Section II discusses the existing works. Section III defines our TDNL problem. Section IV presents our proposed framework with spatiotemporal mobility modeling. Simulation results are discussed in Section V. Finally, Section VI concludes this article.

II. RELATED WORK

For vehicle navigation, several approaches based on real-time traffic information [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] have been proposed in the literature works. Chen et al. [14] constructed the candidate path set (including the fastest and alternative paths) for pretrip routes based on historical traffic data. The offline precomputation of optimal candidate paths is combined with online path retrieval, filtering, and dynamic path adaptation to satisfy path constraints (i.e., meet the drivers' preferences). Through the precomputed and stored candidate path set, route computation time can be significantly reduced with a larger number of simultaneous requests for path planning.

Boriboonsomsin et al. [15] presented an eco-routing navigation system that plan the most eco-friendly route to minimize fuel consumption (with minimum emission) between source and destination places. The historical and real-time traffic information are fused from multiple data sources (e.g., vehicle sensors, probe vehicles, travel demand models, and speed limits). The emission factors are estimated under different vehicle types, different traffic conditions, and different roadway characteristics. Driving along with the eco-friendly path rather than the fastest path, the relative fuel consumption can be reduced with the slightly-increased travel time.

Xiao and Lo [16] navigated vehicles in an adaptive manner using real-time en route traffic information. In this manner, the adaptive navigation is achieved based on the traffic conditions encountered with future traffic states. A probabilistic dynamic programming problem is formulated for adaptive vehicle navigation, which is solved through a backward recursive procedure. In particular, the computational complexity and the routing quality of adaptive vehicle navigation are compared and discussed.

Wang et al. [17] designed a path-planning algorithm for globally optimal vehicle-traffic control based on real-time traffic information collected by vehicular ad hoc networks. In addition to road network traffic balancing, the driver preferences (such as lower travel time or shorter driving length) are jointly considered for reducing the additional travel cost caused by path replanning. Through the designed algorithm, both overall spatial utilization is improved and average vehicle travel cost is reduced.

Kwak et al. [18] developed a vehicular cloud service of social vehicular navigation (SVN) using driver-generated geotagged traffic images to assist drivers in path planning. Instead of the limited scope of static sensors (e.g., traffic cameras), SVN uses the mobility of vehicles to collaboratively share visual traffic information (e.g., current traffic condition or any unexpected event). These shared visual traffic information are filtered, refined, and condensed into traffic digests for route planning and decisions.

Jeong et al. [19] proposed a self-adaptive interactive navigation tool (SAINT) to plan network-wide optimal paths. Through the reported navigation experiences and travel paths of vehicles, the vehicular cloud can obtain vehicle trajectories and real-time traffic conditions on roads. For global traffic optimization, SAINT uses a congestion contribution model to estimate the congestion levels of road segments based on vehicle trajectories and traffic conditions, which provides a navigation path with the minimum congestion level to each vehicle in the road network.

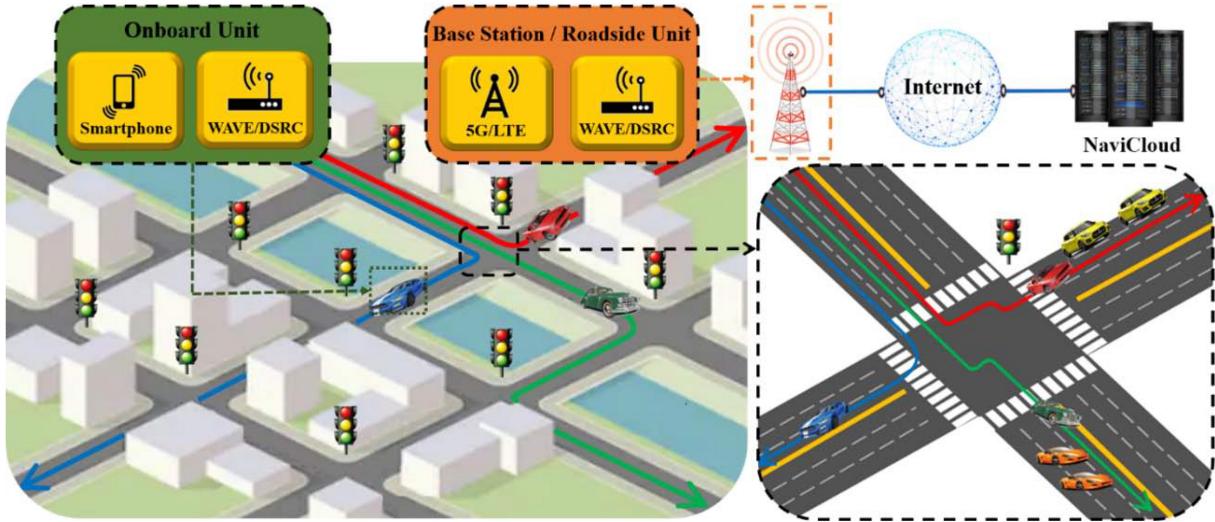


Fig. 1. System architecture of lane-level navigation.

Shen et al. [20] further designed an evolved self-adaptive interactive navigation tool (SAINT+) to improve path planning for emergency vehicles and neighboring vehicles during road accidents. Based on the real-time traffic conditions and congestion contribution model, SAINT+ uses a virtual path reservation strategy for an emergency vehicle by restricting neighboring vehicles to use the reserved path. In addition, SAINT+ constructs protection zones for the accident road segment to avoid serious congestion in the vicinity.

Elbery et al. [21] proposed an IoT-based vehicle crowd management framework consisting of travel-time system optimum navigation and vehicle departure control, which collects driving data to compute road states (i.e., traffic volume and travel time) through connected vehicles or smartphones. Based on both the current road states and capacities, vehicles are allowed to leave as early as possible and the network-wide travel time is minimized. Vehicle crowds are efficiently managed using the proposed framework during special events or emergency evacuation.

Vitali et al. [22] used Monte Carlo simulations to solve the probabilistic time-dependent routing problem (i.e., estimation of the arrival time distribution) through dynamically selecting the number of samples. A path-based approach is designed for the shortest-path problem with on-time arrival reliability. The shortest and alternative paths are selected and travel-time distributions are determined for each path, where the best path is provided to the user requesting navigation services.

Mostafizi et al. [23] presented a decentralized collaborative time-dependent shortest path algorithm (Dec-CTDSP) for connected and autonomous vehicles (CAVs) to optimize their routes. A grid transportation network is modeled for CAVs to travel from one edge to the other, where mobility messages containing the location, speed, and preferred path of each CAV are exchanged among clusters through multihop communications. Through rerouting to a path by Dec-CTDSP, both the travel time of each CAV can be minimized and the system travel time can be reduced.

However, instantaneous shortest paths are planned in most of existing works without considering the traffic conditions of future flows as arriving. In addition, the delay times caused by traffic signals are not explicitly addressed and intersection crossing times are not precisely estimated in existing methods. Furthermore, the driving, waiting, and crossing times of leftmost, middle, and rightmost lanes to pass through road segments and intersections are not separately calculated, where lane-level navigation paths cannot be provided in the literature works.

III. SYSTEM ARCHITECTURE

Fig. 1 shows the system architecture of lane-level navigation. A multilane road consists of a leftmost lane, a rightmost lane, and one or more middle lanes (if any). An OBU is equipped on each vehicle for navigation services and IoV communications (i.e., V2V, V2R, and V2I communications). The located lane of a vehicle is obtained through lane-level positioning techniques. The requesting information (e.g., current location, destination place, brand model, etc.) of a vehicle to require navigation services are sent to the TD LN server (i.e., NaviCloud) via BS/RSU for lane-level path planning.

The TD LN server receives the identifier, source, and destination of the requesting vehicle, collects the scheduling information of traffic signals at intersections, and estimates the crossing time of each intersection based on the driving lane and direction. After considering the estimated driving time (for road segments) and waiting time (for intersections), the planned lane-level navigation path is replied and displayed on the OBU (with user smartphone or onboard screen) of the requesting vehicle. In particular, 4G/5G communications of the user smartphone can be used for Internet access (i.e., V2I communications) if the OBU only supports WAVE/DSRC communications (i.e., V2V and V2R communications) without RSU deployment. The goal is to predict future traffic flows and avoid passing congested areas for optimal navigation path planning by addressing the following concerns.

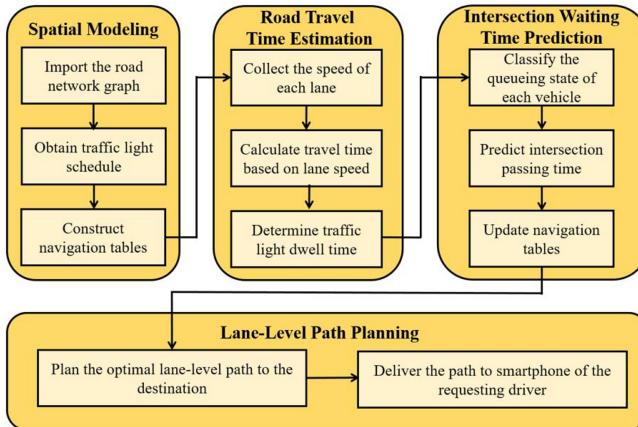


Fig. 2. Operation flow of TDLN.

- 1) *Traffic Signal Consideration*: How could we separately calculate the required times of leftmost, middle, and rightmost lanes to pass an intersection considering the delay caused by traffic signals?
- 2) *Passing Time Estimation*: How could we precisely estimate the travel time of each lane and the waiting time of each intersection based on traffic flow distribution and traffic signal scheduling, respectively?
- 3) *Future Flow Prediction*: How could we accurately predict the traffic condition of a specific road/intersection at future time points to determine the actual travel time as the requesting vehicle arriving at the specific road/intersection?
- 4) *Lane-Level Path Planning*: How could we properly plan the fastest lane-level navigation path with the shortest travel time at future time points based on the estimated passing times of all road segments and intersection?

IV. METHODOLOGY FOR TIME-DEPENDENT LANE-LEVEL NAVIGATION

Fig. 2 shows the operation flow of our framework for spatial modeling, road travel time estimation, intersection waiting time prediction, and lane-level path planning. The lane-level road network graph is constructed and traffic signal schedule is collected in advance. In particular, navigation tables are appended to all intersections for indicating vehicle passing duration (i.e., green light) and waiting duration (i.e., red and yellow light).

The steps to plan a lane-level navigation path for a requesting vehicle are as follows. First, the driving speed of each lane is periodically collected (from the reported floating car data), the travel time on each lane is calculated based on collected lane speeds [using (2)], and the crossing time at the intersection is estimated based on the driving direction, moving distance, and crossing speed of the requesting vehicle [using (11)]. Second, the queuing states of the requesting vehicle at intersections are classified [using (16)] and the times to pass through intersections are predicted based on appended navigation tables [using (17) and (22)]. Third, the travel time of a planned navigation path from source to destination is estimated based on lane driving time and intersection waiting

time [using (3)], where intersection waiting time is the sum of queuing time and passing time. Finally, the navigation table of an intersection is updated if the intersection is on the planned navigation path. Furthermore, the fastest lane-level path to the destination (with the shortest travel time) is send to the requesting vehicle and displayed on the smartphone of the driver.

A. Lane-Level Road Network Construction

As shown in Fig. 3, to support lane-level navigation services, we construct a directed graph $G = (V, E)$ for road networks based on the lanes in each road segment at intersections, where circles and squares represent road segments with forward and reverse directions, respectively, and directed edges represent the lanes from one intersection to the other intersection. In particular, the directed edges between circles and squares at the same intersection are the passing directions from one road segment to the other road segment. To construct a lane-level road network graph for urban areas, we connect the directed edges of adjacent intersections in the area based on the forward and reverse directions of interconnected road segments, as shown in Fig. 3d.

As shown in Fig. 4, the length of the road segment between intersections I_i and I_j is b_{ij} , and the lengths from this road segment to the centers of I_i and I_j are c_i and c_j , respectively. The length L_{ij} from I_i to I_j is calculated as

$$L_{ij} = c_i + b_{ij} + c_j. \quad (1)$$

Suppose that the average driving speed of the k th lane from I_i to I_j is v_{ij}^k . The driving time d_{ij}^k from I_i to I_j in the k th lane is calculated as

$$d_{ij}^k = \frac{L_{ij}}{v_{ij}^k}. \quad (2)$$

Note that, (2) does not consider the passing times in different lanes at an intersection, the driving speed limited by vehicles in the front, and the delay time caused by traffic signals (addressed in Sections IV-B–IV-D, respectively).

Considering the delay caused by front vehicles and traffic signals, the travel time of a planned navigation path P from source to destination consists of lane driving time and intersection waiting time. In particular, intersection waiting time is the sum of queuing time (due to red light and front vehicles) and passing time (affected by driving direction and crossing speed). Suppose that $P = \{[p(0), q(0)], [p(1), q(1)], \dots, [p(n), q(n)], [p(n+1), q(n+1)]\}$, where $p(0)$ is the current location of the requesting vehicle, $p(1), p(2), \dots$, and $p(n)$ are the passing intersections in P , $p(n+1)$ is the destination place, and $q(i)$ is the lane planned to drive from I_i (for $i = 1, 2, \dots, n$). The planned driving lane is the leftmost, middle, and rightmost lane as $q(i) = 1, q(i) = 2, 3, \dots, m_i - 1$, and $q(i) = m_i$, respectively, where m_i is the total number of lanes at the i th passing intersection. Let w_{ij}^k be the waiting time of the k th lane at I_i

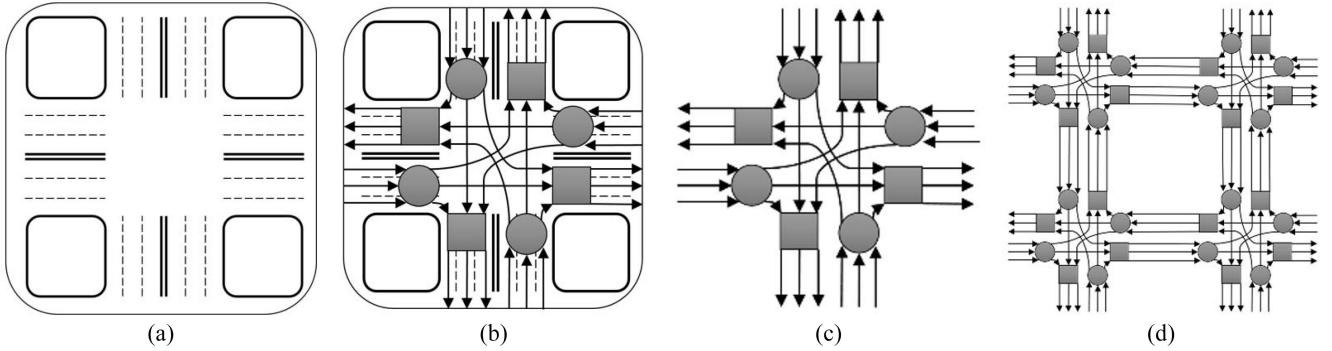


Fig. 3. Lane-level road network construction. (a) Road segments at an intersection. (b) Circle and square vertices with directed edges. (c) Directed graph of an intersection. (d) Directed graph of adjacent intersections.

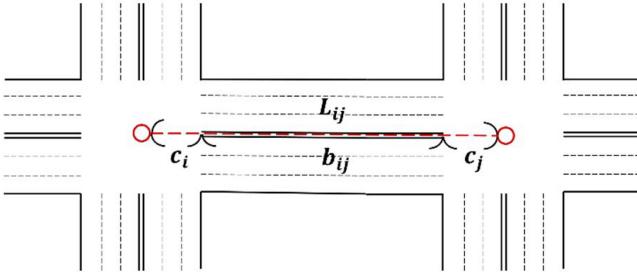


Fig. 4. Distance between adjacent intersections.

to I_j . The total travel time T_P of path P from source $p(0)$ to destination $p(n+1)$ is calculated as

$$T_P = \sum_{i=0}^n d_{p(i)p(i+1)}^{q(i)} + w_{p(i)p(i+1)}^{q(i)} \quad (3)$$

where $w_{p(0)p(1)}^{q(0)} = 0$ (i.e., there is no waiting time at source $p(0)$).

B. Intersection Crossing Time Estimation

To estimate lane-level crossing time at an intersection, the driving direction, moving distance, and crossing speed of the requesting vehicle in different lanes have to be simultaneously considered. Suppose that the acceleration for left-turning vehicles is a_{ij}^1 , the acceleration for go-straight vehicles is $a_{ij}^2 = a_{ij}^3 = \dots = a_{ij}^{m_i-1}$, and the acceleration for right-turning vehicles is $a_{ij}^{m_i}$. For left-turning vehicles, the maximum and minimum accelerations are $a_{\max}^1 = 2 \text{ m/s}^2$ and $a_{\min}^1 = -4 \text{ m/s}^2$ for low-speed and high-speed vehicles, respectively [24]. This is because the high-speed vehicle has to slow down for safely crossing the intersection if its speed is larger than or equal to 13.33 m/s (i.e., 48 km/h), whereas the low-speed vehicle has to accelerate for quickly crossing the intersection if its speed is smaller than or equal to 7.2 m/s (i.e., 26 km/h). The acceleration of medium-speed vehicles with the speed larger than 7.2 m/s and smaller than 13.33 is calculated as

$$a_{ij}^1 = \begin{cases} 2, & \text{if } v \leq 26 \\ -0.002v^2 - 0.024v + 2, & \text{if } 26 < v < 48 \\ -4, & \text{if } v \geq 48 \end{cases} \quad (4)$$

For go-straight vehicles, the maximum and minimum accelerations are $a_{\max}^2 = a_{\max}^3 = \dots = a_{\max}^{m_i-1} = 3 \text{ m/s}^2$ and $a_{\min}^2 = a_{\min}^3 = \dots = a_{\min}^{m_i-1} = -4 \text{ m/s}^2$ for low-speed and high-speed vehicles, respectively [25]. The speed of high-speed vehicles is larger than or equal to 15.56 m/s (i.e., 56 km/h), whereas the speed of low-speed vehicles is smaller than or equal to 8.9 m/s (i.e., 32 km/h). The accelerations of low-speed, medium-speed (i.e., larger than 8.9 m/s and smaller than 15.56 m/s), and high-speed vehicles are calculated as

$$a_{ij}^2 = \begin{cases} 3, & \text{if } v \leq 32 \\ -0.001v^2 - 0.062v + 3, & \text{if } 32 < v < 56 \\ -4, & \text{if } v \geq 56 \end{cases} \quad (5)$$

For right-turning vehicles, the maximum and minimum accelerations are $a_{\max}^{m_i} = 1 \text{ m/s}^2$ and $a_{\min}^{m_i} = -5 \text{ m/s}^2$ for low-speed and high-speed vehicles, respectively [26]. The speed of high-speed vehicles is larger than or equal to 11.11 m/s (i.e., 40 km/h), whereas the speed of low-speed vehicles is smaller than or equal to 5.5 m/s (i.e., 20 km/h). The accelerations of low-speed, medium-speed (i.e., larger than 5.5 m/s and smaller than 11.11 m/s), and high-speed vehicles are calculated as

$$a_{ij}^{m_i} = \begin{cases} 1, & \text{if } v \leq 20 \\ -0.005v^2 - 0.015v + 1, & \text{if } 20 < v < 40 \\ -5, & \text{if } v \geq 40 \end{cases} \quad (6)$$

In particular, there are left-turning/right-turning and go-straight vehicles mixed in the same lane if the total number of lanes on the road segment is smaller than 3 (i.e., single-lane or two-lane road segment). The average acceleration for mixed left-turning and go-straight vehicles is calculated as

$$a_{mix}^1 = \frac{n_L(a_L) + n_S(a_S)}{n_L + n_S} \quad (7)$$

where n_L and n_S are the numbers of mixed left-turning and go-straight vehicles, respectively, $a_L = a_{ij}^1$, and $a_S = a_{ij}^2$. Similarly, the average acceleration for mixed right-turning and go-straight vehicles is calculated as

$$a_{mix}^{m_i} = \frac{n_R(a_R) + n_S(a_S)}{n_R + n_S} \quad (8)$$

where n_R and n_S are the numbers of mixed right-turning and go-straight vehicles, respectively, $a_R = a_{ij}^{m_i}$, and $a_S = a_{ij}^2$. On a single-lane road segment, the average acceleration for

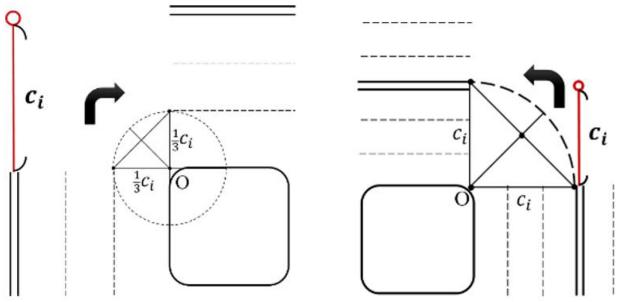


Fig. 5. Right and left-tuning distances.

mixed left-turning, right-turning, and go-straight vehicles is calculated as

$$a_{mix}^2 = \frac{n_L(a_L) + n_R(a_R) + n_S(a_S)}{n_L + n_R + n_S}. \quad (9)$$

Due to the difference of crossing speeds and accelerations in leftmost, middle, and rightmost lanes, we calculate the speed variation and moving distance for each second to estimate the intersection crossing time of the requesting vehicle in a fine-grained manner. Suppose that the speed and acceleration of the requesting vehicle arriving at the intersection are $v_i^k(0)$ and $a_i^k(0)$, respectively, where $v_i^k(0) = v_{ij}^k(0)$ and $a_i^k(0) = a_{ij}^k(0)$ for simplicity of notation. Based on the estimated $a_i^k(j)$ at the j th second (i.e., $a_{ij}^1, a_{ij}^2, a_{ij}^{m_i}, a_{mix}^1, a_{mix}^2$, or $a_{mix}^{m_i}$), the vehicle speeds at next second are calculated as $v_i^k(1) = v_i^k(0) + a_i^k(0)$, $v_i^k(2) = v_i^k(1) + a_i^k(1)$, ..., and $v_i^k(l) = v_i^k(l-1) + a_i^k(l-1)$, where the intersection crossing time is between l and $l+1$ s.

As shown in Fig. 5, left-tuning distance R_i^1 and right-tuning distance $R_i^{m_i}$ can be calculated using the length c_i from the road segment to the center of intersection I_i as $R_i^1 = (c_i\pi/8)$ and $R_i^{m_i} = (c_i\pi/8m_i)3$, respectively. For instance, the crossing distances for left-turning, go-straight, and right-turning vehicles are $(c_i\pi/8)$, $2c_i$, and $(c_i\pi/24)$ at an intersection with three-lane road segments, respectively. Based on the estimated $v_i^k(j)$ and $a_i^k(j)$ at the j th second, the moving distances at next second are calculated as $X_i^k(1) = v_i^k(0)t + (1/2)a_i^k(0)t^2$, $X_i^k(2) = v_i^k(1)t + (1/2)a_i^k(1)t^2$, ..., and $X_i^k(l) = v_i^k(l-1)t + (1/2)a_i^k(l-1)t^2$, where $t = 1$ and $\sum_{j=1}^l X_i^k(j) \leq R_i^k < \sum_{j=1}^{l+1} X_i^k(j)$. The intersection crossing time t_i^k consisting of integer and decimal parts is calculated as

$$t_i^k = l + \Delta t \quad (10)$$

where

$$\Delta t = \frac{R_i^k - \sum_{j=1}^l X_i^k(j)}{v_i^k(l)}. \quad (11)$$

C. Traffic Flow Queue Classification

To estimate the speed of the requesting vehicle arriving at an intersection [e.g., $v_i^k(0)$ at I_i in the k th lane], the queues of traffic flows are classified as *stationary waiting queue* (*SWQ*), *slow moving queue* (*SMQ*), and *continued moving queue* (*CMQ*), as shown in Fig. 6(a)–(c), respectively. There are N arrival vehicles consisting of H moving vehicles and $N-H$ stationary vehicles as the requesting vehicle arrives at intersection I_i in the k th lane, where the acceleration and speed

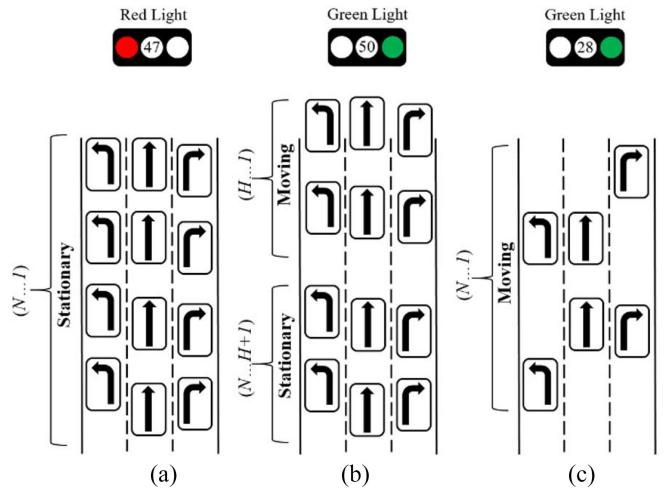


Fig. 6. Traffic flow queues. (a) Stationary waiting queue. (b) Slow moving queue. (c) Continued moving queue.

of the requesting vehicle are limited by moving and stationary vehicles in the front.

Suppose that v_j and a_j of the j th arrival vehicle at I_i are its current speed and acceleration, respectively. The distance d_j from the current location to I_i is calculated as

$$d_j = \sum_{i=1}^j e_i + g_i \quad (12)$$

where e_i is the length of the j th arrival vehicle and g_i is the safety gap between the j th and $(j-1)$ th arrival vehicles (e.g., 2.5 m in average between stopping vehicles [27]). The driving time t_j for the j th arrival vehicle moving to the boundary of I_i can be estimated based on

$$d_j = v_j \times t_j + \frac{a_j(t_j)^2}{2}. \quad (13)$$

Therefore

$$t_j = \frac{-v_j + \sqrt{v_j^2 + 2a_j d_j}}{a_j}. \quad (14)$$

For SWQ, as shown in Fig. 6(a), all arrival vehicles are stationary due to red light, where $H = 0$, $v_N = 0$, and $a_N = 0$. The requesting vehicle has to stop and wait for all of N stationary vehicles in the front to start moving. The total time T_{N+1} of the requesting vehicle (i.e., the $(N+1)$ th arrival vehicle) to drive through intersection I_i from its stop location in the k th lane includes human reaction time (i.e., $\mu = 1.5$ s in average [28]), moving time to I_i (i.e., t_{N+1} calculated using (14)), and intersection crossing time (i.e., $t_i^k[N+1]$ calculated using (10)). The time to drive through I_i for the first, second, ..., and the j th arrival vehicles are $T_1 = \mu + t_1 + t_i^k[1]$, $T_2 = 2 \times \mu + t_2 + t_i^k[2]$, ..., and $T_j = j \times \mu + t_j + t_i^k[j]$, respectively. In particular, the current speed and acceleration of the first, second, ..., and the j th arrival vehicles are $v_1 = 0$ and $a_1 = a_{min}^k$, $v_2 = 0$ and $a_2 = a_{min}^k$, ..., and $v_j = 0$ and $a_j = a_{min}^k$, respectively, where a_{min}^k can be calculated using (4)–(6). The initial speed $v_i^k[j]$ to pass through I_i for the j th arrival vehicle is calculated as

$$v_i^k[j] = v_j + a_{min}^k \times t_j. \quad (15)$$

Thus, the intersection crossing time $t_i^k[j]$ for the j th arrival vehicle can be estimated based on $v_i^k[j]$ using (10). Therefore, the waiting, driving, and crossing time of the requesting vehicle is $T_{N+1} = (N+1) \times \mu + t_{N+1} + t_i^k[N+1]$, where $v_{N+1} = 0$ and $a_{N+1} = a_{\min}^k$.

For SMQ, as shown in Fig. 6(b), the arrival vehicles in the front are slowly moving whereas those in the back are still stationary due to traffic signal switching (from red light to green light) or traffic jam (in rush hours), where $H > 0$, $v_H > 0$, and $a_H = a_{\min}^k$. The requesting vehicle has to stop and wait for $N-H$ stationary vehicles in the front to start moving. The time to drive through I_i for the $(H+1)$ th, $(H+2)$ th, \dots , and N th arrival vehicles are $T_{H+1} = (H+1-H) \times \mu + t_{H+1} + t_i^k[H+1]$, $T_{H+2} = (H+2-H) \times \mu + t_{H+2} + t_i^k[H+2]$, \dots , and $T_N = (N-H) \times \mu + t_N + t_i^k[N]$, respectively. Therefore, the waiting, driving, and crossing time of the requesting vehicle is $T_{N+1} = (N+1-H) \times \mu + t_{N+1} + t_i^k[N+1]$, where $v_{N+1} = 0$, $a_{N+1} = a_{\min}^k$, $v_i^k[N+1]$ is calculated using (15), and $t_i^k[N+1]$ is estimated based on $v_i^k[N+1]$ using (10).

For continuous moving queue (CMQ), as shown in Fig. 6(c), all arrival vehicles in the front are moving due to green light, where $H = N$, $v_H > 0$, and $a_H > 0$. The requesting vehicle does not need to stop and can keep moving as approaching to I_i . The time to drive through I_i for the first, second, \dots , and the N th arrival vehicles are $T_1 = t_1 + t_i^k[1]$, $T_2 = t_2 + t_i^k[2]$, \dots , and $T_N = t_N + t_i^k[N]$, respectively. Therefore, the driving and crossing time (without waiting) of the requesting vehicle is $T_{N+1} = t_{N+1} + t_i^k[N+1]$, where $v_{N+1} = v_N$, $a_{N+1} = a_N = a_{\min}^k$, $v_i^k[N+1]$ is calculated using (15), and $t_i^k[N+1]$ is estimated based on $v_i^k[N+1]$ using (10).

The moving queue type (i.e., SMQ or CMQ) at I_i is depending on whether the N th arrival vehicle is moving or not. Suppose that the requesting vehicle is arriving at the A_{N+1} th seconds. If the difference of A_{N+1} and A_N is smaller than or equal to the sum of human reaction times for all stationary arrival vehicles in the front, the queue is slowly moving (i.e., SMQ); otherwise, the queue is continuous moving (i.e., CMQ), which is formulated as

$$\begin{cases} \text{SMQ, if } A_{N+1} - A_N \leq (N-H) \times \mu \\ \text{CMQ, if } A_{N+1} - A_N > (N-H) \times \mu \end{cases} \quad (16)$$

D. Traffic Signal Delay Prediction

To predict the delay caused by traffic signals at I_i , the waiting times (i.e., w_i^k) is estimated for the requesting vehicle arriving at I_i in the k th lane during green light interval $T_{i,G}^k$ and red light interval $T_{i,R}^k$. On the one hand, suppose that the requesting vehicle is arriving at the A_{N+1} th seconds during green light interval. The waiting time w_i^k is depending on whether the requesting vehicle can pass through I_i in the current green phase (i.e., the first green light interval), which is calculated as

$$w_i^k = \begin{cases} T_{N+1} \text{ if } T_{N+1} - t_i^k[N+1] + T_a \leq T_{i,G}^k \\ T_{N+1-L(1)} + T_b + T_{i,R}^k, & \text{if } m = 1 \\ T_{N+1-L(1)-L(2)} + T_b + 2 \times T_{i,R}^k + T_{i,G}^k, & \text{if } m = 2 \\ T_{upd} + T_b + m \times T_{i,R}^k + (m-1) \times T_{i,G}^k, & \text{if } m > 2 \end{cases} \quad (17)$$

where

$$T_a = A_{N+1} - \left\lfloor \frac{A_{N+1}}{T_{i,R}^k + T_{i,G}^k} \right\rfloor \times (T_{i,R}^k + T_{i,G}^k) \quad (18)$$

$$T_b = T_{i,G}^k - T_a \quad (19)$$

$$T_{upd} = T_{N+1-\sum_{j=1}^m L(j)} \quad (20)$$

$$\begin{aligned} mT_{i,G}^k + (m-1)T_{i,R}^k &< T_{N+1} - t_i^k[N+1] + T_a \\ &\leq m(T_{i,G}^k + T_{i,R}^k) \end{aligned} \quad (21)$$

and $L(j)$ is the number of arrival vehicles passed through I_i in the j th green light interval after the requesting vehicle arrives at I_i .

If $T_{N+1} - t_i^k[N+1] + T_a \leq T_{i,G}^k$, it means that the requesting vehicle can pass through I_i in $T_{i,G}^k$ without red light delay (i.e., $w_i^k = T_{N+1}$), where T_a and T_b are the elapsed and remaining times in the current traffic signal interval as the requesting vehicle is arriving at I_i , respectively. If $T_{i,G}^k < T_{N+1} - t_i^k[N+1] + T_a \leq T_{i,G}^k + T_{i,R}^k$ (i.e., $m = 1$), it means that the requesting vehicle cannot pass through I_i in the first green light interval but can pass in the second green light interval, which has to wait for the remainder green light interval and the following red light interval. In particular, the arrival vehicles in the front passed through I_i in the first green light interval have to be removed from T_{N+1} (i.e., $T_{N+1-L(1)}$). If $2 \times T_{i,G}^k + T_{i,R}^k < T_{N+1} - t_i^k[N+1] + T_a \leq 2 \times (T_{i,G}^k + T_{i,R}^k)$ (i.e., $m = 2$), it means that the requesting vehicle cannot pass through I_i in the first and second green light intervals but can pass in the third green light interval, which has to wait for two green and red light intervals. The front vehicles passed through I_i in the first and second green light intervals are removed from T_{N+1} (i.e., $T_{N+1-L(1)-L(2)}$). Similarly, for $m > 2$, it means that the requesting vehicle can pass through I_i in the $(m+1)$ th green light interval, which has to wait for m green and red light intervals and remove $\sum_{j=1}^m L(j)$ passed arrival vehicles from T_{N+1} .

On the other hand, suppose that the requesting vehicle is arriving at the A_{N+1} th seconds during red light interval. The waiting time w_i^k is depending on whether the requesting vehicle can pass through I_i in the following green light interval, which is calculated as:

$$w_i^k = \begin{cases} T_{N+1} + T_b, & \text{if } T_{N+1} - t_i^k[N+1] \leq T_{i,G}^k \\ T_{upd} + T_b + m(T_{i,R}^k + T_{i,G}^k), & \text{if } m > 0 \end{cases} \quad (22)$$

where T_b is modified as

$$T_b = T_{i,R}^k - T_a \quad (23)$$

and m is modified as

$$mT_{i,G}^k + (m-1)T_{i,R}^k < T_{N+1} - t_i^k[N+1] \leq m(T_{i,G}^k + T_{i,R}^k). \quad (24)$$

If $T_{N+1} - t_i^k[N+1] \leq T_{i,G}^k$, it means that the requesting vehicle can pass through I_i in the following green light interval with the delay T_b caused by the current red phase (i.e., $w_i^k = T_{N+1} + T_b$), where T_b is the remaining time in the current red light interval. If $m > 0$, it means that the requesting vehicle can

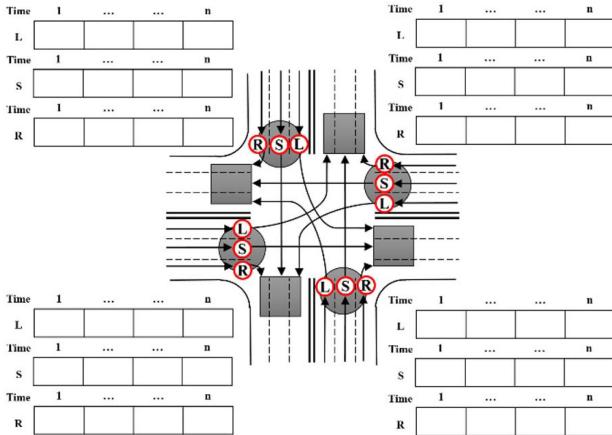


Fig. 7. Navigation tables for left-tuning, go-straight, and right-turning lanes.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Road network size	7481 m × 3840 m
Total number of road segments	18658
Total number of intersections	1218
Total length of road segments	1059.23 km
Number of requesting vehicles	100 ~ 1800
Vehicle length	5 m
Gap between stopping vehicles	2.5 m
Speed limit	60 km/hr (16.6 m/s)
Maximum acceleration	3 m/s ²
Maximum deceleration	6 m/s ²



Fig. 8. Road network of Taichung City.

pass through I_i in the $(m + 1)$ th green light interval, which has to wait for m green and red light intervals and remove $\sum_{j=1}^m L(j)$ passed arrival vehicles from T_{N+1} .

As shown in Fig. 7, navigation tables are added to each intersection for left-turning, go-straight, and right-turning lanes to record the arriving time (i.e., A_1, A_2, \dots , and A_N), the driving through time (i.e., T_1, T_2, \dots , and T_N), and the predicted waiting time (i.e., w_i^k) of each requesting vehicle. Through navigation tables, the actual travel time can be estimated and the fastest navigation path with the shortest travel time can be planned for a new requesting vehicle considering the traffic conditions of lanes and intersections at future time points as arriving.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance in terms of time prediction accuracy, average travel time, travel time

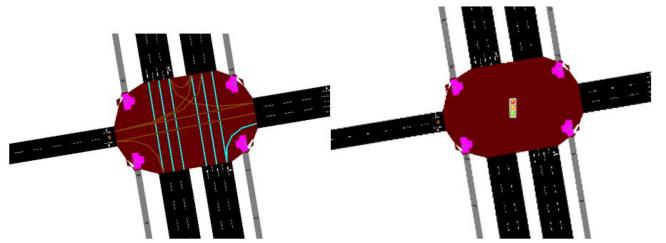


Fig. 9. Traffic light control in SUMO.



Fig. 10. Traffic light distribution in Taichung City.

distribution, and mean link time of our proposed framework in simulation of urban mobility (SUMO [29]), which is an open road traffic simulation package for large road networks. As shown in Fig. 8, we use NETCONVERT to import the real road network of Taichung City (in Taiwan) from the open geographic database of OpenStreetMap [30]. In addition, as shown in Fig. 9, we use an external Python program to interact with SUMO and control traffic lights via the traffic control interface (TraCI) protocol, where Fig. 10 shows the traffic light distribution in Taichung City consisting of 18658 road segments and 1218 intersections. Table II summarizes the basic parameters used in our simulation.

The background vehicles (without requesting navigation services or without equipping OBUs) are randomly deployed at road segments in the city, and their destination places are also randomly selected (i.e., random source and destination places). Similarly, the requesting vehicles (driving along with planned lane-level navigation paths) are assigned to randomly-selected source and destination places. Both background and requesting vehicles are removed as they are arriving at their destination places. Three car following models of Krauss car-following, adaptive cruise control (ACC), and cooperative ACC (CACC) [31] are used in our simulation. The experiments of navigation path planning are conducted ten times and the average value is taken.

First, we conduct experiments to compare the difference of actual and predicted travel times under different ratios of requesting vehicles, where ratio 50% consists of 600 requesting vehicles and 600 background vehicles, ratio 60% consists of 720 requesting vehicles and 480 background vehicles, ..., and ratio 100% consists of 1200 requesting vehicles and 0 background vehicles. From Fig. 11, it can be observed that predicted travel times are quite close to actual travel times with

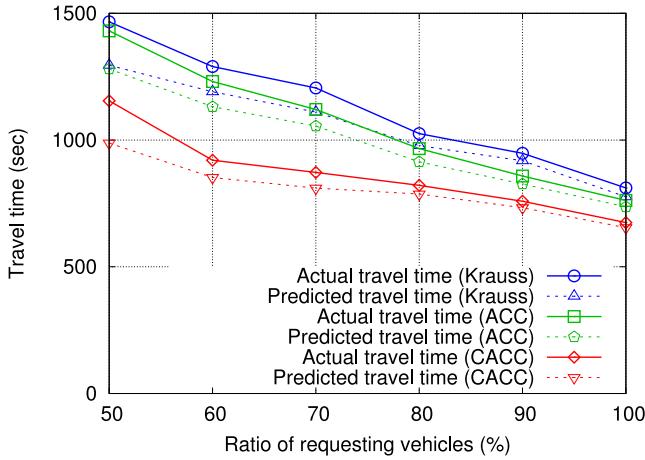


Fig. 11. Comparisons of actual and predicted travel times under different ratios of requesting vehicles.

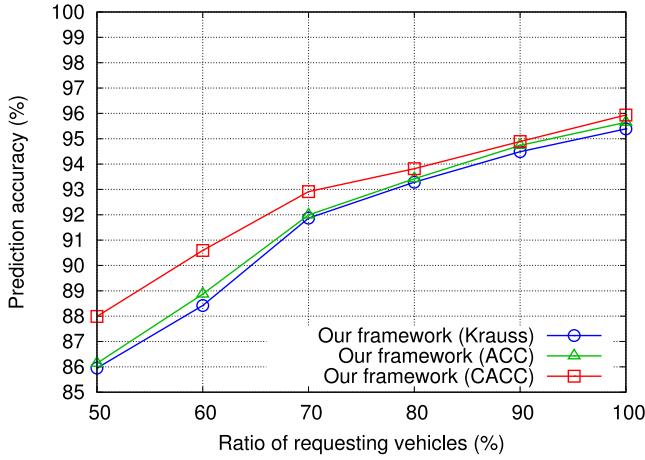


Fig. 12. Comparisons of prediction accuracy in our framework with different car following models.

high ratios of requesting vehicles. In addition, both actual and predicted travel times of CACC are much lower than those of Krauss and ACC. The communication cooperation among vehicles in the IoV can efficiently reduce the time consumption on the road network (e.g., human reaction time to start moving can be removed and driving time on road segments can be shorten).

Fig. 12 shows time prediction accuracy in our framework with Krauss, ACC, and CACC car following models. It can be seen that more accurate prediction results can be achieved with higher ratios of requesting vehicles (e.g., more than 95% accuracy for 100% requesting vehicles). In particular, the accuracy of travel time prediction is more than 85% with all car following models for only 50% of requesting vehicles, which does not require high penetration rates for using our framework.

Next, to show the performance improvement using our framework, we compare existing methods of navigation path planning, including 1) shortest distance [32] navigation to plan the path from source to destination with the minimum total driving distance (without considering the delay caused by road congestion and traffic signals); 2) shortest driving time [14] navigation to plan the path with the shortest driving

time (without considering the delay caused by traffic signals); 3) minimum waiting time [16] navigation to plan the path with the smallest total waiting time at intersections (without considering the driving time in road segments); and 4) self-adaptive interactive navigation tool (SAINT [19], [20]) to plan the path with the minimum total congestion (based on the designed congestion value calculation).

Fig. 13 shows comparisons of average travel time, cumulative probability distribution, and mean link time using existing methods and our framework under 100, 200, ..., and 900 requesting vehicles for random source and destination places, where the number of background vehicles is fixed to 600. From Fig. 13(a), it can be observed that the travel time differences between navigation methods are enlarged with more requesting vehicles. In particular, our framework can achieve the shortest average travel time among all methods, which reduces 46.3%, 41.1%, 30.4%, and 24.5% of travel times (with 900 requesting vehicles) using shortest distance navigation, shortest driving time navigation, minimum waiting time navigation, and SAINT, respectively. This is because our framework considers lane driving time, front vehicle blocking, traffic signal delay, intersection crossing time, and future traffic conditions together for lane-level path planning.

Fig. 13(b) shows the cumulative probability distribution of travel times using different navigation methods under 500 requesting vehicles. It can be seen that 95% of navigation travel times using our framework are smaller than 2000 s, whereas less than 85% of those using existing methods are smaller than 2000 s. Through our framework, more vehicles can activate at destination places much earlier, which can reduce the number of driving vehicles on roads to avoid/relieve traffic congestion. Fig. 13(c) shows mean link times (i.e., average driving time for one road segment) using different navigation methods under 100, 200, ..., and 900 requesting vehicles. The average driving time to move through one road segment is the shortest in our framework because the requesting vehicles are navigated to light-load road segments (with shorter moving and waiting times) considering the traffic conditions at future time points.

On the one hand, Fig. 14 shows comparisons of average travel time, cumulative probability distribution, and mean link time using existing methods and our framework under 600 background vehicles and 100, 200, ..., and 900 requesting vehicles for the fixed source place and random destination places. It occurs when a music festival is finished or a sport competition is over, audiences drive from the venue (i.e., the same source place) to home (i.e., different destination places). Similar to Fig. 13, our framework can achieve the shortest average travel time among all methods, which reduces 36.2%, 31.4%, 27.7%, and 21.5% of travel times (under 900 requesting vehicles) compared with existing methods. In addition, the reduced travel times are getting larger with more requesting vehicles. Furthermore, more navigation paths with much shorter travel times and lower link times are achieved using our framework than existing methods.

On the other hand, Fig. 15 shows comparisons of average travel time, cumulative probability distribution, and mean link time using existing methods and our framework under 600

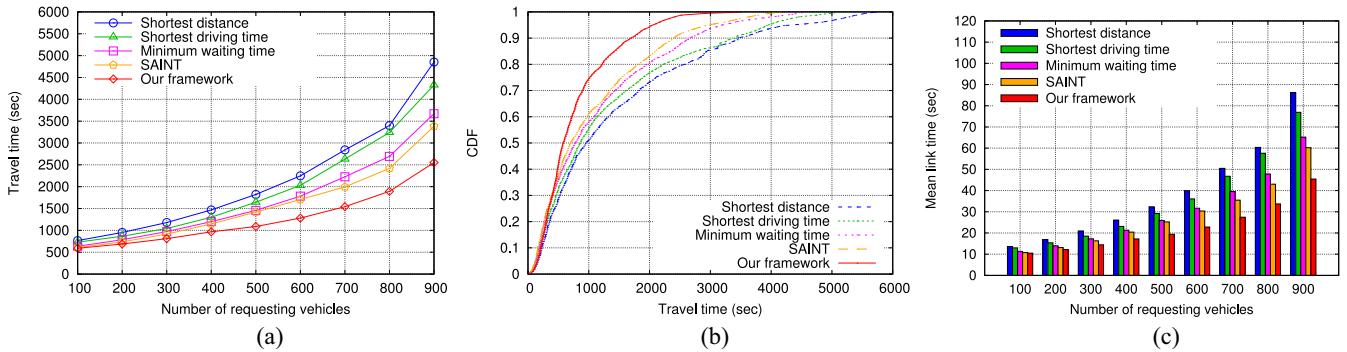


Fig. 13. Comparisons of (a) average travel time, (b) cumulative probability distribution, and (c) mean link time under different numbers of requesting vehicles for random source and destination places.

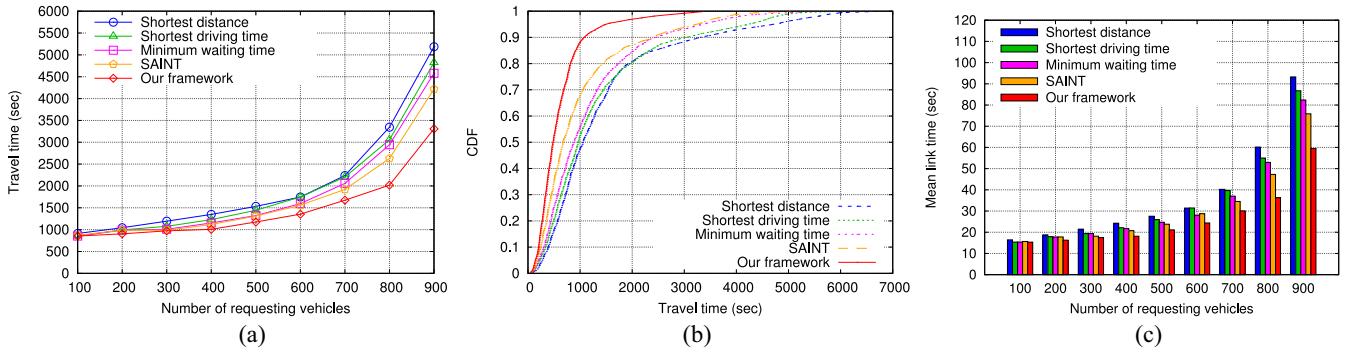


Fig. 14. Comparisons of (a) average travel time, (b) cumulative probability distribution, and (c) mean link time under different numbers of requesting vehicles for the fixed source place and random destination places.

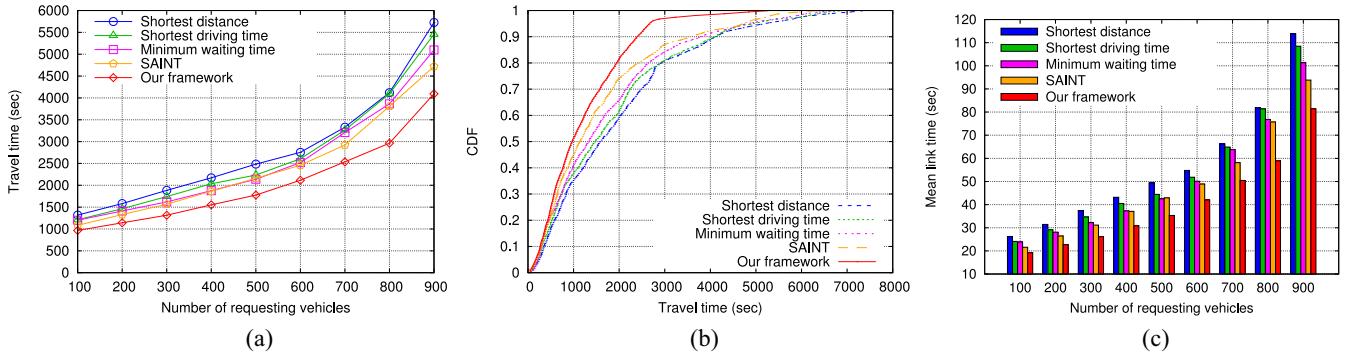


Fig. 15. Comparisons of (a) average travel time, (b) cumulative probability distribution, and (c) mean link time under different numbers of requesting vehicles for random source places and the fixed destination place.

background vehicles and 100, 200, ..., and 900 requesting vehicles for random source places and the fixed destination place. It occurs when workers go to work places or people from difference places visit a hot spot, commuter/visitor drive from home (i.e., difference source places) to the high-speed rail station/famous tourist spot (i.e., the same destination place). Similar results are obtained for our framework, which can significantly reduce average travel time for each road segment and achieve 28.4%, 23.7%, 19.6%, and 13.2% of travel time improvement compared with existing methods (under 900 requesting vehicles).

Moreover, we conduct experiments to compare average travel times using existing and our navigation methods with Krauss, ACC, and CACC car following models under 50%,

60%, ..., and 100% of requesting vehicles for random source and destination places. From Fig. 16, it can be observed that under 100% of requesting vehicles (i.e., no background vehicles), average travel times to drive along with planned paths are the shortest for all navigation methods and all car following models. In particular, for the Krauss car following model, our framework can reduce 27.1%, 18.8%, 8.7%, and 6.1% of travel times using shortest distance, shortest driving time, minimum waiting time, and SAINT, respectively. Similarly, for the ACC car following model, the improvement ratios of 29.7%, 20.4%, 10.8%, and 7.4% for average travel time are achieved in our framework. For the CACC car following model, our framework improves 30.1%, 22.3%, 11.7%, and 10.2% of travel times over four compared existing methods.

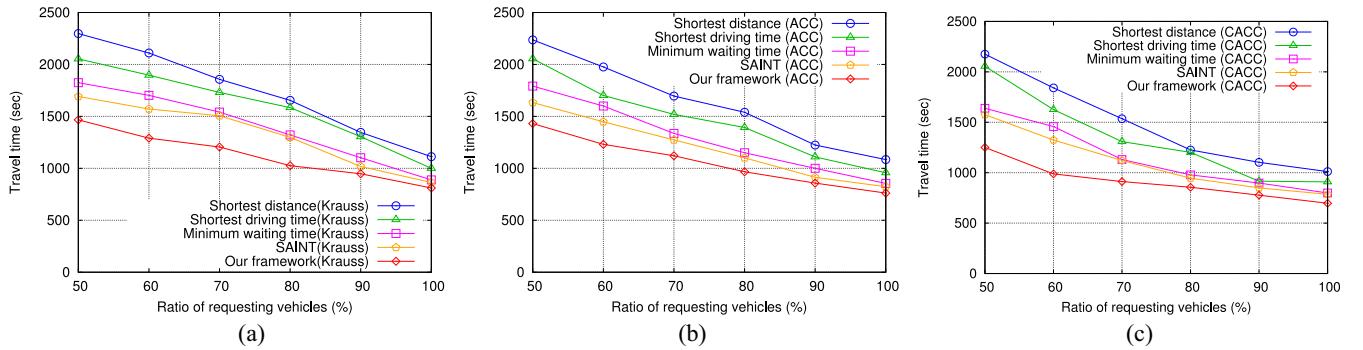


Fig. 16. Comparisons of average travel times using five navigation methods with (a) Krauss, (b) ACC, and (c) CACC car following models under different ratios of requesting vehicles.

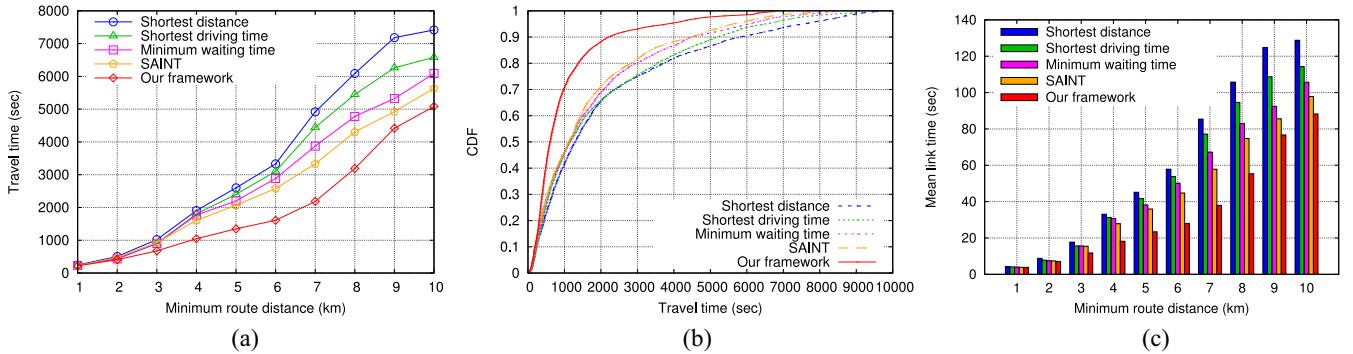


Fig. 17. Comparisons of (a) average travel time, (b) cumulative probability distribution, and (c) mean link time using five navigation methods under different minimum route distances between source and destination places.

Finally, we conduct experiments to compare average travel time, cumulative probability distribution, and mean link time using existing methods and our framework under 900 background vehicles and 900 requesting vehicles (with the CACC car following model), where the linear distances between randomly-selected source and destination places are 1, 2, …, and 10 km. From Fig. 17(a), it can be seen that with the longer distance between source and destination places, more travel time can be reduced by driving along with the navigation path planned in our framework. This is because more road segments and intersections can be considered for path planning to minimize the driving times in lanes and the waiting time at integrations.

Fig. 17(b) shows the cumulative probability distribution of travel times using different navigation methods for 5 km linear distance between source and destination places. More than 70% of travel times are less than 1000 s in our framework, whereas existing methods achieve no more than 50% of travel times less than 1000 s. Fig. 17(c) shows the average driving time for one road segment in the planned navigation path. With the longer navigation path, the improvement ratio is getting larger in our framework over existing methods. This is because our framework can evenly distribute the requesting vehicles on different road segments, avoid the congestion from aggregated traffic flows, and reduce the delay caused by traffic signals.

VI. CONCLUSION

In this work, we design a lane-level navigation framework with time-dependent concerns based on IoV communications.

The proposed framework consists of lane-level road network construction, intersection crossing time estimation, traffic flow queue classification, and traffic signal delay prediction. The total travel time (i.e., lane driving time, signal waiting time, and intersection crossing time) of a requesting vehicle driving along with the planned navigation path from source to destination is minimized. In particular, the traffic conditions of future flows are predicted to determine the congestion level of each lane as arriving, which can avoid passing the areas with light load currently but traffic jam later. For future works, we are developing the lane-level path replanning system for road accidents, which includes cooperative navigation for emergency vehicles and normal vehicles (considering the vehicle distribution, target destination, and driving path), load-balancing dispatching for vehicles in the road accident area, and lane-level rerouting for vehicles to be passing the road accident area. In addition, the optimal lane determination algorithm could be developed considering traffic rules, vehicle characteristics, and optimization objectives to find the optimal lanes on roads with different configurations.

REFERENCES

- [1] B. Ji et al., "Survey on the Internet of Vehicles: Network architectures and applications," *IEEE Commun. Stand. Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.
- [2] L.-W. Chen and P.-C. Chou, "BIG-CCA: Beacon-less, Infrastructure-less, and GPS-less cooperative collision avoidance based on vehicular sensor networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1518–1528, Nov. 2016.

- [3] D. Tian, G. Wu, P. Hao, K. Boriboonsomsin, and M. J. Barth, "Connected vehicle-based lane selection assistance application," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2630–2643, Jul. 2019.
- [4] L.-W. Chen and H.-M. Chen, "Driver behavior monitoring and warning with dangerous driving detection based on the Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7232–7241, Nov. 2021.
- [5] L.-W. Chen and G.-L. Wang, "Risk-aware and collision-preventive cooperative fleet cruise control based on vehicular sensor networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 11, pp. 179–191, Jan. 2022.
- [6] "Global traffic scorecard." 2021. [Online]. Available: <https://inrix.com/press-releases/2021-traffic-scorecard/>
- [7] "TomTom traffic index." 2024. [Online]. Available: <https://www.tomtom.com/traffic-index/>
- [8] L.-W. Chen and J.-X. Liu, "Time-efficient indoor navigation and evacuation with fastest path planning based on Internet of Things technologies," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 5, pp. 3125–3135, May 2021.
- [9] H. Hashikami, R. Kobayashi, Y. Li, Y. Nakano, and M. Shigeno, "Safe route carpooling to avoid accident locations and small-scale proof of concept in Japan," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 7, pp. 4239–4250, Jul. 2023.
- [10] T. Song, N. Capurso, X. Cheng, J. Yu, B. Chen, and W. Zhao, "Enhancing GPS with lane-level navigation to facilitate highway driving," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4579–4591, Jun. 2017.
- [11] L.-W. Chen and Y.-F. Ho, "Centimeter-grade metropolitan positioning for lane-level intelligent transportation systems based on the Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1474–1485, Mar. 2019.
- [12] M. Elsheikh, A. Noureldin, and M. Korenberg, "Integration of GNSS precise point positioning and reduced inertial sensor system for lane-level car navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2246–2261, Mar. 2022.
- [13] C. Zhang et al., "Vehicle re-identification for lane-level travel time estimations on congested urban road networks using video images," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12877–12893, Aug. 2022.
- [14] Y. Chen, M. G. H. Bell, and K. Bogenberger, "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 14–20, Mar. 2007.
- [15] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1694–1704, Dec. 2012.
- [16] L. Xiao and H. K. Lo, "Adaptive vehicle navigation with en route stochastic traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1900–1912, Oct. 2014.
- [17] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1664–1678, May 2015.
- [18] D. Kwak, R. Liu, D. Kim, B. Nath, and L. Iftode, "Seeing is believing: Sharing real-time visual traffic information via vehicular clouds," *IEEE Access*, vol. 4, pp. 3617–3631, 2016.
- [19] J. Jeong, H. Jeong, E. Lee, T. Oh, and D. H. C. Du, "SAINT: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4053–4067, Jun. 2016.
- [20] Y. Shen, J. Lee, H. Jeong, J. Jeong, E. Lee, and D. H. C. Du, "SAINT+: Self-adaptive interactive navigation tool+ for emergency service delivery optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1038–1053, Apr. 2018.
- [21] A. Elbery, H. S. Hassanein, N. Zorba, and H. A. Rakha, "IoT-based crowd management framework for departure control and navigation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 95–106, Jan. 2021.
- [22] E. Vitali et al., "An efficient monte carlo-based probabilistic time-dependent routing calculation targeting a server-side car navigation system," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 1006–1019, Apr. 2021.
- [23] A. Mostafizi, C. Koll, and H. Wang, "A decentralized and coordinated routing algorithm for connected and autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11505–11517, Aug. 2022.
- [24] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1908–1916, May 2019.
- [25] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 287–299, Sep. 2018.
- [26] M. B. Younes and A. Boukerche, "Intelligent traffic light controlling algorithms using vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 5887–5899, Aug. 2016.
- [27] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1958–1973, Jul. 2017.
- [28] N. D. Lerner, "Brake perception-reaction times of older and younger drivers," in *Proc. Human Factors Ergon. Soc. Annu. Meet.*, 1993, pp. 206–210.
- [29] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.
- [30] M. Haklay and P. Weber, "OpenStreetMap: User-Generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.
- [31] X. Lin, W. Meng, S. Wouter, and V. A. Bart, "Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks," *Transp. Research Part-C, Emerg. Technol.*, vol. 96, pp. 380–397, Nov. 2018.
- [32] Y. Saab and M. VanPutte, "Shortest path planning on topographical maps," *IEEE Trans. Syst., Man, Cybern., A, Syst., Humans*, vol. 29, no. 1, pp. 139–150, Jan. 1999.



Lien-Wu Chen (Senior Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in December 2008.

He joined the Department of Computer Science, National Chiao Tung University as a Postdoctoral Research Associate in February 2009 and qualified as an Assistant Research Fellow in February 2010. From 2012 to 2015, he was an Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung

City, Taiwan, where he served as the Chief for Operation and Maintenance Section from 2016 to 2018. He was an Associate Professor from 2015 to 2019 and a Full Professor since February 2019. His research interests include wireless communication and mobile computing, especially in Internet of Vehicles, Internet of Things, and Internet of People.

Dr. Chen received over 30 Best/Excellent Paper Awards, Best/Outstanding Demo Awards, Best Presentation Awards, and Outstanding Paper Publication Awards since 2009, and transferred three technologies to Acer Incorporated in January, June, and November 2011. He is the advisor of over 50 Prototyping Awards, Student Engineering Paper Awards, College Student Research Awards, and Master Thesis Awards since 2014. He is an Honorary Member of Phi Tau Phi Scholastic Honor Society, a Lifetime Member of the Chinese Institute of Electrical Engineering, the Institute of Information and Computing Machinery, and the Taiwan Institute of Electrical and Electronic Engineering, and a Senior Member of ACM.



Chih-Cheng Tsao received the B.S. degree in mobile computing in computer science from the Department of Information Technology and Communication, Shih Chien University, Taipei, Taiwan, in 2018, and the M.S. degree from the Department of Information Engineering and Computer Science, Feng Chia University, Taichung City, Taiwan, in 2020.

His research interests include Internet of Things and mobile applications and systems.

Mr. Tsao is the recipient of the Excellent Demo Award at the Mobile Computing Workshop in 2019 and the Outstanding Paper Award at the Taiwan Academic Network Conference in 2019.