MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Edge Computing for Interconnected Intersections in Internet of Vehicles

Lee, Gilsoo; Guo, Jianlin; Kim, Kyeong Jin; Orlik, Philip V.; Ahn, Heejin; Di Cairano, Stefano; Saad, Walid

TR2020-081 June 24, 2020

Abstract

To improve the traffic flow in the interconnected intersections, the vehicles and infrastructure such as road side units (RSUs) need to collaboratively determine vehicle scheduling while exchanging information via vehicle-to-everything (V2X) communications. However, due to a large number of vehicles and their mobility, scheduling in the interconnected intersection is a challenging problem. Moreover, since low-latency information exchange and real-time decision making process are required, it becomes more challenging to design a holistic framework incorporating traffic control and V2X communications. In this paper, an edge computing framework is proposed to solve a travel time minimization problem at the interconnected intersections. The proposed framework enables each RSU to decide intersection scheduling while the vehicles individually determine travel trajectory by controlling their dynamics. To this end, a V2X communications protocol is designed to exchange information among vehicles and RSUs. Then, the road segments around intersection are partitioned into sequence, control, and crossing zones. In the sequence zone, optimal time is scheduled for vehicles to pass the intersection with a minimum delay. In the control zone, the location and velocity of each vehicle are controlled to arrive the crossing zone at the scheduled time by using a control algorithm designed to effectively increase driving comfort and reduce fuel consumption. Thus, the proposed framework enables the vehicles to safely pass the crossing zone without collision. Simulation results show that the proposed edge computing can successfully reduce the total travel time by up to 14.3% based on optimal scheduling for the interconnected intersections.

IEEE Intelligent Vehicles Symposium

Edge Computing for Interconnected Intersections in Internet of Vehicles

Gilsoo Lee*†, Jianlin Guo*, Kyeong Jin Kim*, Philip Orlik*, Heejin Ahn*, Stefano Di Cairano*, and Walid Saad†

*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139 USA,
Emails: {guo,kkim,porlik,hahn,dicairano}@merl.com.
†Wireless@VT, Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA,
Emails: {gilsoolee,walids}@vt.edu.

Abstract*—To improve the traffic flow in the interconnected intersections, the vehicles and infrastructure such as road side units (RSUs) need to collaboratively determine vehicle scheduling while exchanging information via vehicle-to-everything (V2X) communications. However, due to a large number of vehicles and their mobility, scheduling in the interconnected intersection is a challenging problem. Moreover, since low-latency information exchange and real-time decision making process are required, it becomes more challenging to design a holistic framework incorporating traffic control and V2X communications. In this paper, an edge computing framework is proposed to solve a travel time minimization problem at the interconnected intersections. The proposed framework enables each RSU to decide intersection scheduling while the vehicles individually determine travel trajectory by controlling their dynamics. To this end, a V2X communications protocol is designed to exchange information among vehicles and RSUs. Then, the road segments around intersection are partitioned into sequence, control, and crossing zones. In the sequence zone, optimal time is scheduled for vehicles to pass the intersection with a minimum delay. In the control zone, the location and velocity of each vehicle are controlled to arrive the crossing zone at the scheduled time by using a control algorithm designed to effectively increase driving comfort and reduce fuel consumption. Thus, the proposed framework enables the vehicles to safely pass the crossing zone without collision. Simulation results show that the proposed edge computing can successfully reduce the total travel time by up to 14.3% based on optimal scheduling for the interconnected intersections.

I. Introduction

With the emergence of vehicle-to-everything (V2X) communications, the concept of the Internet of Vehicles (IoV) allows a connected vehicle to communicate with other vehicles and road infrastructure so as to exchange road traffic information such as location and velocity of vehicles [1]. To handle the real-time traffic data, there is a need for distributed computation which can be effectively handled using the so-called edge computing paradigm [2]. In the IoV, the edge computing enables low-latency computation at the infrastructure network edge, for supporting autonomous vehicle control systems while overcoming the limitations of centralized cloud computation. As the advantages of the edge computing architecture come from exploiting the computing resource at network edge, the on-board computing system of vehicles and road infrastructure such as road side unit (RSU) can be used to store, compute and control the traffic information over an edge computing network. However, to reap the benefits of the connected vehicles in the IoV, many architectural and operational challenges on implementing edge computing in a vehicular network must be addressed [3]–[5]. In particular, authors in [3] introduce vehicular use cases of edge computing. Also, the work in [4] proposes a network slicing scheme that can improve the vehicular network resource utilization while supporting various applications running on edge computing. Moreover, to configure edge computing for vehicles, authors in [5] propose a network architecture that can efficiently reduce V2X communications latency.

Meanwhile, the traffic management schemes have been investigated to coordinate the connected vehicles on conflicting paths such as intersections [6]–[10]. For example, the work in [6] introduces intersection management techniques to deploy cooperative and automated intersections. Also, authors in [7] investigate a scheduling problem to determine the times at which vehicles can enter an intersection without collision. Moreover, the vehicle trajectory optimization problem is studied in [8] to develop an intersection vehicle coordination scheme avoiding any collision. Furthermore, authors in [9] study a traffic signal control scheme to coordinate the vehicles at signalized intersections. The work in [10] proposes an intersection control mechanism that enables the vehicles to travel the interconnected intersections.

In all of these existing vehicle scheduling works [6]–[10], the wireless communications are ignored in the system design while focusing on vehicle coordination and controlling in the intersections. Indeed, the V2X communications are an important part of intersection traffic management for fully or semi-autonomous connected vehicles as the information is distributed and shared among vehicles and RSUs. Moreover, the works in [7] and [8] generally consider one intersection passing. Unlike the system model of in the previous works, the intersections are interconnected in the IoV system. When the intersections are interconnected, the traffic flow optimization at one intersection should consider the traffic conditions at the neighboring intersections so that the total travel time of the vehicles required to pass multiple intersections can be minimized. Furthermore, most of the existing works about edge computing and V2X communications [3]–[5] do not consider the dynamics of the connected vehicles. In contrast, an edge computing system needs to take control of vehicles to coordinate their mobility. Consequently, unlike the existing literature [3]–[10] which conducts research on the connected vehicles either from the networking perspective or from the control perspective, our goal is to design an integrated edge computing system to enable the connected vehicles to coordinate the vehicle dynamic control in the interconnected

This work was done while Gilsoo Lee was working at MERL.

intersections while exploiting V2X communications.

The main contribution of this paper is an edge computing framework that integrates the control technology into V2X communication networks for connected vehicles. We first investigate the edge computing procedures to partition an intersection area into three zones and determine the role of each zone so that the vehicles are controlled by edge computing. Next, we use information-theoretic capacity of a wireless link to perform the zone size analysis and find the proper zone sizes satisfying the low-latency requirements of V2X communications. Based on the determined zone sizes, we conduct scheduling optimization to minimize the maximum travel time or total travel time of the vehicles, in which the vehicle dynamics are considered as the constraints of the problem. Finally, using the computed schedules, we optimize vehicle trajectories to control vehicle mobility for driving comfort improvement and fuel consumption reduction. Simulation results show that the proposed scheduling can decrease the maximum travel time by up to 4.7% compared to the baseline.

The rest of this paper is organized as follows. In Section II, the system model is presented. In Section III, we formulate the proposed vehicle scheduling problem. Section IV presents our optimal vehicle trajectory planning. Simulation results are analyzed in Section V while conclusions are drawn in Section VI.

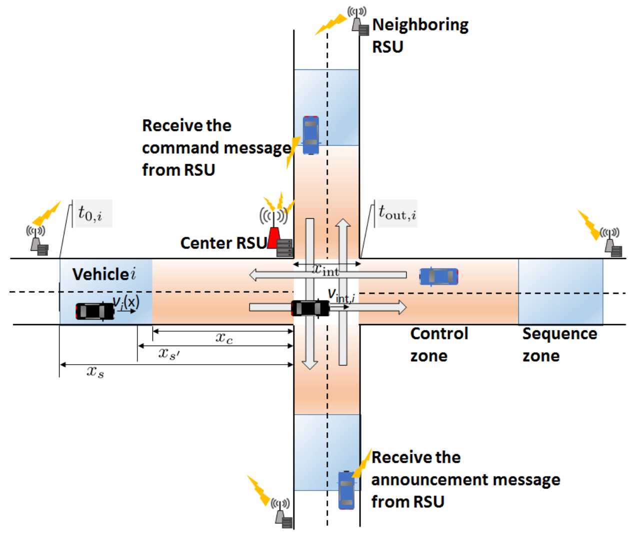


Fig. 1: System model of the interconnected intersections.

II. System Model

Consider an urban road environment consisting of the interconnected intersections as shown in Fig. 1, where $v_i(x)$ is the velocity of vehicle i when it is at the location x, $v_{\text{int},i}$ is the reference velocity for vehicle i to enter the intersection, $t_{\text{out},i}$ is the time for vehicle i to exit the intersection, x_{int} is the size of the intersection, x_s, $x_{s'}$, x_c and $t_{0,i}$ will be explained in the next section.

For convenience, the RSU at the center intersection is named as the *center RSU*. In Fig. 1, the RSUs at four neighboring intersections are connected to the center RSU by using wired or wireless communication links. The traffic of each neighboring intersection is controlled by corresponding *neighboring RSU* independently. When vehicles pass the interconnected intersections, the incoming traffic direction d can be one of four directions, i.e., $d \in \{n, e, w, s\}$ referring north, east, west, and south. The set of vehicles that approach an intersection from north, east, west, and south directions are denoted by $\mathcal{I}_n, \mathcal{I}_e, \mathcal{I}_w$, and $\mathcal{I}_s$, respectively. Each vehicle is indexed by $i \in \mathcal{I} = \mathcal{I}_n \cup \mathcal{I}_e \cup \mathcal{I}_w \cup \mathcal{I}_s$. For instance, if vehicle i moves from south to north, the moving direction of the vehicle is denoted by $d_{(i)} = s$. We assume that the vehicles traveling on the same direction are indexed in the order of entering the intersection.

A. Communications Protocol for IoV Edge Computing

The RSUs and vehicles are equipped with the RF module so that RSUs and vehicles can use V2X communications. Throughout the inter-RSU communications, the neighboring RSUs send traffic information to the center RSU on the vehicles moving forward to the center intersection using wired or wireless connection. Using wireless communications links between vehicles and RSUs, vehicles are able to update their status such as location, velocity, and acceleration. The RSU can send the control message to the vehicles so that the vehicles can be controlled by following the RSU decision. The RSU is assumed to have enough computational capability, and therefore, can perform the roles of collecting, storing, controlling, and processing the data related to the vehicles and the traffic flow, as in typical edge computing scenarios. Therefore, when vehicles exit the neighboring intersection and are heading to the center intersection, the center RSU needs to make a decision on how the incoming traffic is considered at the center intersection. For the control purpose, the center RSU partitions the section of the road around center intersection into *sequence zone*, *control zone*, and *crossing zone*. Zone size determination will be explained next.

We propose a communications protocol shown in Fig. 2 to share information among the RSUs and the vehicles for the IoV edge computing. When a vehicle i leaves a neighboring intersection, the neighboring RSU sends a message to the center RSU about the information of the arriving vehicle. The neighboring RSU also sends a message to departure vehicle i about the start point of the sequence zone at the center intersection. When vehicle i enters the sequence zone of the center intersection, it sends a heartbeat message containing its status to the center RSU. Upon receiving the heartbeat message, the center RSU sends an announcement message to vehicles i about the start points of the control zone and the crossing zone as well as the crossing zone size x_{int}. While the vehicles are traveling in the sequence zone, the edge computing system at the center RSU schedules the vehicles, i.e., determining optimal passing time and passing order for vehicles to enter the crossing zone. The decision made by the center RSU is sent to vehicle i before it enters the control zone via a scheduling message containing time $t_{\text{out},i}$ to exit the center intersection, the reference velocity $v_{\text{int},i}$ to arrive at the crossing zone and the information about the preceding vehicle $i-1$. Upon receiving the scheduling message, vehicle i determines an optimal trajectory to be applied in the control

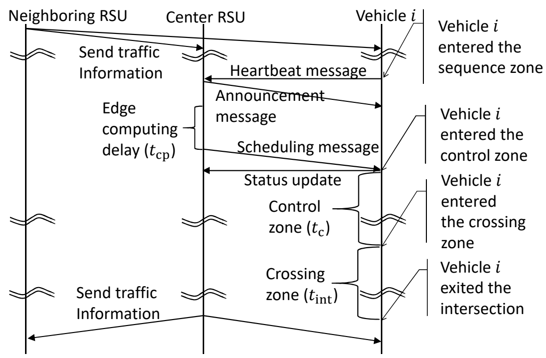


Fig. 2: Protocol for message exchange among RSUs and vehicles.

zone such that it enters the crossing zone at time $t_{\text{out},i} - t_{\text{int},i}$ with the velocity $v_{\text{int},i}$, where $t_{\text{int},i} = x_{\text{int}}/v_{\text{int},i}$ is the time for vehicle i to cruise through the crossing zone with the velocity $v_{\text{int},i}$. When vehicle i enters control zone, it transmits a status update message containing its trajectory to the center RSU. The status update message will be included in the scheduling message sent to the following vehicle $i+1$ for its trajectory planning. The center RSU can also use status update message to compute an more global trajectory for vehicle i. Once vehicle i exits the center intersection, the center RSU sends a traffic information message to the neighboring RSUs to update the traffic information and updates vehicle i with the start point of the sequence zone at next intersection.

As described in the communications protocol, the center RSU sends messages to vehicle i over a wireless channel in the sequence zone. Using an information-theoretic capacity, the data rate of the downlink between the RSU and vehicle i can be defined by

$$R_i(x) = B\log_2\left(1+\frac{g_i(x)P_{\text{tx}}}{BN_0}\right), \tag{1}$$

where $g_i(x) = \beta_1 x^{-\beta_2}$ is the channel gain between vehicle i and the RSU with x being the distance between them. β_1 and β_2 are, respectively, the path loss constant and path loss exponent. P_{tx} is the transmission power of the RSU, B is the bandwidth of the channel, and N_0 is the noise power spectral density. Then, if a message has a size of K bits, the wireless transmission time is given by:

$$D_i(x) = \frac{K}{R_i(x)}. \tag{2}$$

In Fig. 2, the RSU sends two messages to vehicle i at the location x_s and $x_{s'}$, respectively, where the data sizes of two messages are K_{x_s} and $K_{x_{s'}}$ bits. The messages should be delivered to the vehicle before the vehicle enters to the control zone at location x_c. Assume t_{s} is the computation time of the edge computing at the center RSU. During t_{s}, the center RSU schedules vehicle passing time and passing order at the crossing zone. Assume the velocity of vehicle i is $v_i(x)$. The transmission time of the announcement message and computation time t_{s} need to satisfy:

$$D_i(x_s) + t_{\text{s}} \le \int_{-x_s}^{-x_{s'}} \frac{1}{v_i(x)}dx. \tag{3}$$

Once vehicle receives scheduling message from the RSU, it determines an optimal trajectory to be applied in the control zone. Assume t_{c} is the computation time for a vehicle to compute trajectory. The transmission time of the scheduling message and computation time t_{c} need to satisfy:

$$D_i(x_{s'}) + t_{\text{c}} \le \int_{-x_{s'}}^{-x_c} \frac{1}{v_i(x)}dx. \tag{4}$$

Thus, if the lengths of the sequence and control zones satisfy the conditions in (3) and (4), the vehicle is able to follow the RSU control decision when it arrives at the control zone.

B. Vehicle Control Dynamics

While vehicles are moving in the sequence, control, and crossing zones, it is assumed that the vehicles follow the lane. When vehicles enter the control zone, the vehicle velocity is controlled to arrive at the crossing zone with the velocity and timing determined by the RSU. The vehicles adjust their velocities according to their dynamics model. To describe the vehicle control, we focus on the longitudinal dynamics of the vehicles [11]. When the control input u_i corresponds to the longitudinal acceleration, the state of vehicle i is defined as $s_i = [x_i, v_i]^T$, where x_i and v_i are its longitudinal position and velocity, respectively. We assume that the vehicle follows second-order integrator dynamics shown as:

$$\dot{s}_i = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} s_i + \begin{bmatrix}0\\ 1\end{bmatrix} u_i, \tag{5}$$

where the lateral dynamics and slip can be negligible in the case that the vehicles maintain a relatively low velocity. To account for the speed limit, the velocity of vehicle i is non-negative, i.e., $v_i \in [0, \overline{v}]$ at any time. Also, the acceleration or deceleration of vehicle i, u_i, has a range $[\underline{u}, \overline{u}]$ where $\underline{u} < 0 < \overline{u}$.

C. Sequence Zone Size and Control Zone Size

Given models of V2X communications and control dynamics, it is essential to determine a set of proper values of the distance variables x_s, $x_{s'}$, and x_c such that the constraints in (3) and (4) are satisfied. The control zone size must be long enough that a vehicle i can reach any velocity from its control zone entering velocity $v_i(x_c)$. Therefore, from the vehicle dynamics in (5), the minimum control zone size for vehicle i can be determined as $x_c \ge \max_i\left(\frac{\overline{v}^2 - v_i^2(x_c)}{2\overline{u}}, \frac{-v_i^2(x_c)}{2\underline{u}}\right)$. The first term is the distance required to accelerate the vehicle i to the maximum velocity from the velocity $v_i(x_c)$ using the maximum acceleration $\overline{u}$ while the second term is the distance needed to stop the vehicle i from the velocity $v_i(x_c)$ using the maximum deceleration $\underline{u}$. This result provides a lower bound for the control zone size. The RSU can determine x_c such that $x_c \ge \max\left(\frac{\overline{v}^2}{2\overline{u}}, \frac{-\overline{v}^2}{2\underline{u}}\right)$.

Given the control zone size x_c, we determine x_s and $x_{s'}$. $x_{s'}$ such that $x_{s'} - x_c$ is long enough for (4) to be true. Assuming that the vehicles travel at the maximum velocity, we have

$$D_i(x_{s'}) + t_{\text{c}} \le \frac{x_{s'} - x_c}{\overline{v}}. \tag{6}$$

Using (6), we can obtain $x_{s'}$. Once, $x_{s'}$ is determined, x_s needs to be chosen such that $x_s - x_{s'}$ is long enough for (3) to be held. Assuming that $v_i(x) = \overline{v}$, we have

$$D_i(x_s) + t_\text{s} \leq \frac{x_s - x_{s'}}{\overline{v}}. \tag{7}$$

Finally, we can solve for x_s from (7). Notice that the distance between two intersections is an upper bound for x_s.

To solve (6) and (7), the maximum message size and the maximum communication range can be used such that constraints (3) and (4) hold even for the maximum message and the maximum communication distance.

III. Optimal Intersection Traffic Scheduling

Given the system model, our goal is to control vehicles to pass the interconnected intersections for the traffic improvement. To this end, the RSU needs to make a decision on how to control the vehicle dynamics by using the information gathered from V2X communications. In practice, the traffic conditions in the urban intersections are interconnected. The control decision at one intersection affects traffic at other intersections. Therefore, it is challenging to jointly optimize the interconnected intersections. To cope with the challenge, we propose a distributed edge computing scheme for the interconnected intersection traffic management, in which a directional weight $w_{d_{(i)}}$ is introduced to reflect the interfering traffic at the neighboring intersection.

We formulate the following per-RSU optimization problem whose goal is to minimize the weighted maximum travel time:

$$\min_{\boldsymbol{t}_\text{out},\, \boldsymbol{B}} \max_{\forall i \in \mathcal{I}} \left(w_{d_{(i)}} (t_{\text{out},i} - t_{0,i}) \right) \tag{8}$$

$$\text{s.t.} \quad t_{\text{out},i} - t_{0,i} \geq t_{\text{m},i}, \forall \mathcal{I}, \tag{9}$$

$$t_{\text{out},i+1} - t_{\text{out},i} > t_\text{h}, \forall i \in \mathcal{I}_d, \tag{10}$$

$$t_{\text{out},i} - t_{\text{out},i'} + M B_{i,i'} \geq t_{\text{int},i}, \forall i, i' \in \mathcal{I}, i \neq i', \tag{11}$$

$$t_{\text{out},i'} - t_{\text{out},i} + M(1 - B_{i,i'}) \geq t_{\text{int},i'}. \tag{12}$$

In the problem (8), vehicle i arriving at the sequence zone at time $t_{0,i}$ is scheduled to exit the intersection at time $t_{\text{out},i}$. The travel time for vehicle i is $t_{\text{out},i} - t_{0,i}$. Let $\boldsymbol{t}_\text{out}$ be the vector consisting of $t_{\text{out},i}$, $\forall i \in \mathcal{I}$, we aim to find a schedule $\boldsymbol{t}_\text{out}$ that minimizes the weighted maximum travel time of the vehicles. Also, we determine the passing order of two vehicles denoted by $B_{i,i'} \in \{0, 1\}$ where $\boldsymbol{B}$ is defined as the vector of $B_{i,i'}$, $\forall i, i' \in \mathcal{I}, i \neq i'$. $B_{i,i'} = 0$ implies that vehicle i is scheduled to pass the intersection prior to vehicle i' and $B_{i,i'} = 1$ indicates that vehicle i is scheduled to pass the intersection after vehicle i'.

The constraint (9) shows that vehicle i must not violate the speed limit rule from entering the sequence zone to exiting the intersection, i.e., its travel time has a lower bound $t_{\text{m},i}$. Let $v_i(x_s)$ be the sequence zone entering velocity. The travel time lower bound $t_{\text{m},i}$ for vehicle i, i.e., $\min(t_{\text{out},i} - t_{0,i})$, is derived as:

$$t_{\text{m},i} = \frac{\overline{v} - v_i(x_s)}{\overline{u}} + \frac{v_{\text{int},i} - \overline{v}}{\underline{u}} + \left[x_s - \left(\frac{\overline{v}^2 - v_i^2(x_s)}{2\overline{u}} + \frac{v_{\text{int},i}^2 - \overline{v}^2}{2\underline{u}} \right) \right] \frac{1}{\overline{v}} + \frac{x_\text{int}}{v_{\text{int},i}}. \tag{13}$$

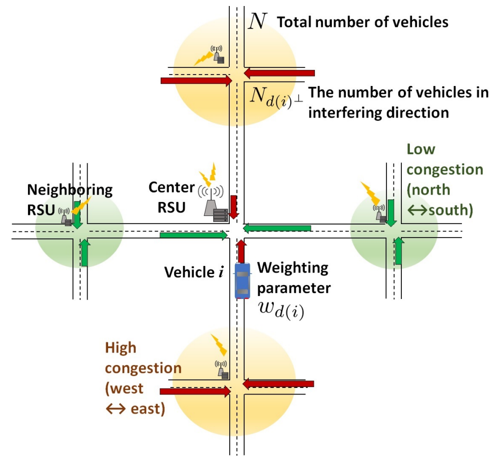


Fig. 3: Directional weight for congestion reduction.

In the Eq.(13), the first term is the time for vehicle i to accelerate to the maximum velocity $\overline{v}$ from its sequence entering velocity $v_i(x_s)$ using the maximum acceleration $\overline{u}$, the second term is the time for vehicle i to decelerate from the maximum velocity $\overline{v}$ to its crossing zone passing velocity $v_{\text{int},i}$ using the minimum acceleration $\underline{u}$, the third term is the time for vehicle i to travel the remaining distance between the sequence zone start point to the control zone ending point using the maximum velocity $\overline{v}$ and the fourth term is the time for vehicle i to cruise through the crossing zone using the crossing zone passing velocity $v_{\text{int},i}$. A special case of this lower bound is $t_{\text{m},i} = \frac{x_s + x_\text{int}}{\overline{v}}$, which is the time for vehicle i to travel through the sequence zone, the control zone and the crossing zone with the maximum velocity $\overline{v}$.

In the constraint (10), headway time t_h is applied to guarantee the safety time gap between two adjacent vehicles on the same lane. In addition, the constraints (11) and (12) guarantee that only one vehicle can pass the intersection at a time. In particular, the difference of exit time of any two vehicles i and i' needs to be greater than the travel time in the intersection of the preceding vehicle. In (11) and (12), M is an arbitrarily large constant used in a big-M method [1].

In problem (8), when vehicle i is approaching to the center intersection in direction $d_{(i)}$, the center RSU accounts for the weighted travel time, i.e., $w_{d_{(i)}}(t_{\text{out},i} - t_{0,i})$, to schedule the vehicle i. We define $w_{d_{(i)}}$ as

$$w_{d_{(i)}} = \frac{N - N_{d_{(i)}^\perp}}{N} \in [0, 1], \forall i \in \mathcal{I}, \tag{14}$$

where $d_{(i)}^\perp$ is the orthogonal direction with respect to $d_{(i)}$, N is the total number of vehicles on all lanes at the next intersection, and $N_{d_{(i)}^\perp}$ is the number of vehicles on the lanes in $d_{(i)}^\perp$ directions at the next intersection. Therefore, $w_{d_{(i)}}$ decreases if more vehicles are traveling in $d_{(i)}^\perp$ directions than $d_{(i)}$ directions at the neighboring intersection.

As shown in our communications protocol, the neighboring RSUs share the traffic information such as the number of vehicles and their traveling directions at the neighboring intersections. Therefore, the center RSU is able to calculate $w_{d_{(i)}}$ to solve the problem (8). The center RSU tends to delay the exit time of vehicle having a small $w_{d_{(i)}}$. For example,

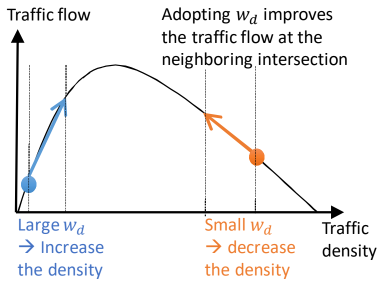


Fig. 4: Implication of the weighting parameter w_d.

if the vehicles traveling in $d_{(i)}$ direction have a small $w_{d_{(i)}}$, they are delayed to arrive at the next intersection so that the neighboring intersection can serve other vehicles in $d_{(i)}^{\perp}$ directions with a priority. By doing so, the traffic flow of the interconnected intersections can be improved. Furthermore, the causality between adopting weighting parameter $w_{d_{(i)}}$ and traffic flow improvement can be explained by using a Lighthill-Whitham-Richards (LWR) model representing the behavior of traffic streams. As shown in Fig. 4, the LWR model implies that the traffic flow can be maximized at a certain traffic density located between zero and the maximum traffic density of an intersection [12]. When the neighboring intersection is congested with interfering vehicles, i.e., small $w_{d_{(i)}}$, the center RSU will instantly decrease the incoming traffic by delaying the vehicles to arrive at next intersection. Therefore, it helps to reduce the traffic density at the neighboring intersection. As a result, the traffic flow can be improved. Also, if the neighboring intersection has less interfering vehicles, i.e., larger $w_{d_{(i)}}$, the center RSU will schedule the vehicles to arrive at next intersection without the intended delay. By doing so, traffic density of the neighboring intersection will increase, thus improving the traffic flow.

The problem (8) is a mixed integer linear programming (MILP) problem. Though the problem has an NP-hard complexity, an optimal solution can be found by using an optimization solver. Since the optimization variables in the formulated problem do not depend on the decisions of other RSUs, the problem (8) is the per-RSU optimization problem and can be solved by each RSU in a fully distributed way. Thus, the center RSU is able to solve the problem to determine the passing time and the passing order of vehicles at the center intersection.

Additionally, as a variation of the problem (8), we also propose a weighted summation objective function for MILP problem given by

$$\min_{\boldsymbol{t}_{\text{out}},\boldsymbol{B}} \sum_{\forall i\in\mathcal{I}} \left(w_{d_{(i)}}(t_{\text{out},i}-t_{0,i})\right) \text{ s.t. (9)} - \text{(12)}. \tag{15}$$

This objective function minimizes the weighted total travel time.

Based on the application need, either the weighted maximum travel time objective function or the weighted total travel time objective function can be applied in the MILP scheduling problem.

IV. Optimal Vehicle Trajectory Control

In this section, we propose an optimal vehicle trajectory scheme that exploits the cooperation of the RSU and vehicles to improve driving comfort and reduce fuel consumption. Our edge computing model orchestrates computation of the RSU and vehicles, where the RSU makes a large-scale decision for multiple vehicles traveling on the different lanes of the intersection while each vehicle can locally optimize the control decision over the time to follow the decision made by the RSU. More specifically, with the determined values of x_s, $x_{s'}$, and x_c, the center RSU makes a decision on the passing time and the passing order. Then, each vehicle optimizes the trajectory that reaches the crossing zone at the assigned time with the given reference velocity.

We propose a trajectory optimization problem to enable the vehicles to determine the acceleration u_i and velocity v_i for the control zone. Our objective is to minimize acceleration and control vehicle to arrive at the intersection crossing zone with the given reference velocity. In the trajectory optimization problem, discrete time system is applied with T_s as the sampling period. The vehicle i can optimize the velocity $\boldsymbol{v} = [v_i(t_0), \ldots, v(t_n), \ldots, v(t_N)]$ and acceleration $\boldsymbol{u} = [u_i(t_0), \ldots, u_i(t_n), \ldots, u_i(t_{N-1})]$, where $t_n = nT_s$, $\forall n \in [0, N]$. In other words, the vehicle determines the optimal velocity and the corresponding acceleration by solving following quadratic programming (QP) problem:

$$\min_{\boldsymbol{v}_i,\boldsymbol{u}_i} \sum_{n=0}^{N-1} (v_i(t_n) - v_{\text{int},i})^2 + qu_i(n)^2, \tag{16}$$

$$\text{s.t.} \quad 0 \le x_i(t_n) \le x_{i-1}(t_n) - d_{\text{s}}, \forall n, \tag{17}$$

$$\underline{v} \le v_i(t_n) \le \overline{v}, \forall i \in \mathcal{I}_d, , \tag{18}$$

$$\underline{u} \le u_i(t_n) \le \overline{u}, \forall i \in \mathcal{I}_d, , \tag{19}$$

$$x_i(t_0) = x_c, x_i(t_N) = 0, \forall i \in \mathcal{I}_d, , \tag{20}$$

$$v_i(t_0) = v_{0,i}, v_i(t_N) = v_{\text{int},i}, \forall i \in \mathcal{I}_d, \tag{21}$$

$$v_i(t_n) = v_i(t_{n-1}) + u_i(t_n)T_s, \forall n, \tag{22}$$

where q is a constant to weigh in acceleration, $x_{i-1}(t_n)$ is the location of the preceding vehicle $i-1$ at time t_n obtained from status update message transmitted by vehicle $i-1$, d_{s} is the safety driving distance to avoid a forward collision, $v_{0,i}$ is the estimated velocity for vehicle i to enter the control zone. The reference velocity $v_{\text{int},i}$ can be any velocity required by traffic policy or traffic condition. By minimizing the first term of the objective function in (16), the vehicles will adjust their velocities as close as the reference velocity $v_{\text{int},i}$. By minimizing the second term of the objective function in (16), the vehicle can reduce the usage of acceleration and smooth mobility. Therefore, the driving comfort can be improved and the fuel consumption can be reduced.

Once the velocity is determined, the location of vehicle i can be calculated by using vehicle dynamics equation as

$$x_i(t_{n+1}) = x_i(t_n) + \frac{v_i(t_n) + v_i(t_{n+1})}{2} T_s, \quad \forall n \in [1, N-2], \forall i \in \mathcal{I}. \tag{23}$$

To solve the proposed scheduling and trajectory optimization problems, we use the edge computing architecture designed for the RSU and the vehicle. The inter-operation of the RSU and the vehicle is summarized in Algorithm 1, in which the RSU determine $t_{\text{out},i}$, $\forall i$. The RSU then sends its

Algorithm 1 IoV Edge Computing Algorithm

RSUs determine the weight $w_{d_{(i)}}, \forall i$
procedure SCHEDULING($w_{d_{(i)}}, t_{0,i}, \forall i$)
  Solve problem (8) to determine $\boldsymbol{t}_{\text{out}}$
  transmit $\boldsymbol{t}_{\text{out}}$, $v_{\text{int},i}$ to vehicle i
procedure DYNAMICSCONTROL($\boldsymbol{t}_{\text{out}}, v_{\text{int},i}$)
  while x_i in the sequence zone **do**
    $\underline{u} \leq u_i \leq \overline{u}$ ▷ No further constraint
  while x_i in the control zone **do**
    Solve problem (16) to determine $\boldsymbol{u}_i$
    $u_i \leftarrow u_i(n)$ ▷ Follow the optimal trajectory
  while x_i in the crossing zone **do**
    $u_i \leftarrow 0$ ▷ Use cruise control

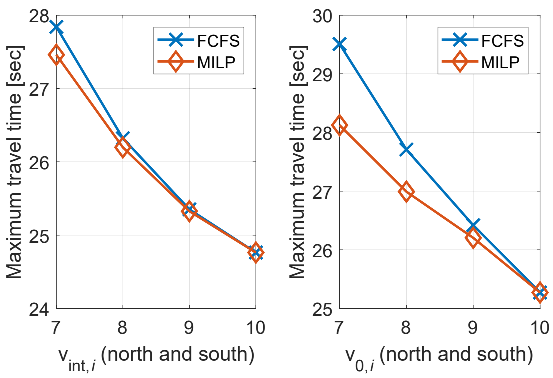


Fig. 5: Optimization results compared to the baseline results.

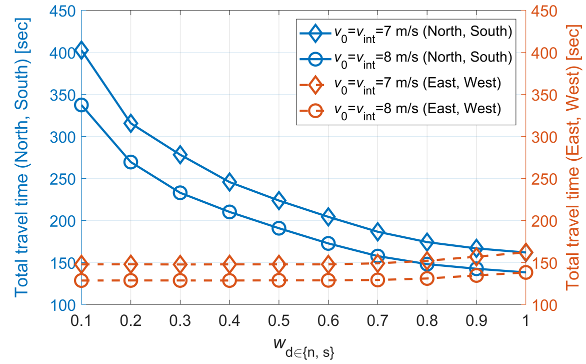
scheduling decision to vehicles, and each vehicle computes the optimal trajectory and makes a control decision while using information from the preceding vehicle only.

It is interesting to notice that upon receiving the status update message from the vehicles, RSU can also compute more globalized optimal trajectories for the vehicles. Therefore, the proposed vehicle trajectory control can be independently implemented by vehicle and the RSU, thus providing high scalability in the scenario of the interconnected intersections.

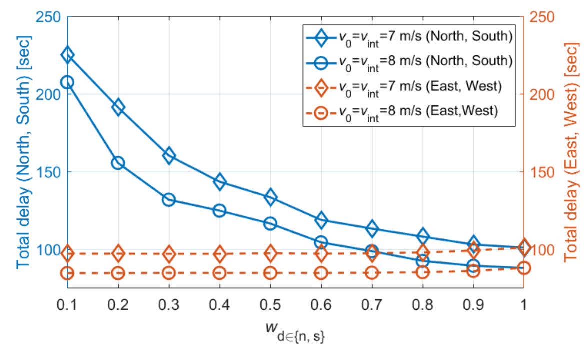
V. Simulation Results and Performance Evaluation

In this section, we validate the performance of the proposed edge computing model throughout simulations. For our simulations, we consider the traffic management at the center intersection where the vehicles are approaching from four directions. Each direction has a two-way road with two lanes, we assume that a fixed number of vehicles are traveling on a lane, and the initial times $t_{0,i}$ of the vehicle i entering the sequence zone is randomly distributed. To model vehicle dynamics, we set $\underline{v} = 0$, $\overline{v} = 15$ m/s, and $\underline{u} = -\overline{u} = -2.5$ m/s^2. For V2X communications, the center RSU is located at the center point of the intersection. The bandwidth is 10 MHz, the power spectral density of the noise is -174 dBm/Hz, and $P_{\text{tx}} = 23$ dBm. The channel gain is set to $\beta_1 = 0.0007$ and $\beta_2 = 2$ from a free-space path loss model. All statistical results are averaged over a large number of simulation runs.

In Fig. 5, we show the maximum travel time comparison with respect to the reference intersection velocity and initial velocity of the vehicle entering the control zone. To define a baseline, we adopt a scheduling scheme based on a first come



(a) $|\mathcal{I}| = 12$ vehicles ($|\mathcal{I}_d| = 3$)



(b) $|\mathcal{I}| = 8$ vehicles ($|\mathcal{I}_d| = 2$)

Fig. 6: Impact of the weighting parameter.

first serve (FCFS) basis, where the vehicles are scheduled to exit the intersection in the order of the vehicle entering the control zone. When the vehicles travel in north and south directions from 7 to 10 m/s, we set $w_{d_{(i)}} = 1$ and $v_{0,i} = v_{\text{int},i} = 10$ m/s, $\forall i \in \mathcal{I}_e \cup \mathcal{I}_w$, and $|\mathcal{I}_d| = 3$. We can see that the maximum travel time and the maximum travel time difference between MILP scheduling and FCFS scheduling decrease as the reference velocity or initial velocity increases. MILP scheduling and FCFS scheduling are identical when $v_{0,i}$ and $v_{\text{int},i}$ are equal for four directions. This is due to the fact that FCFS scheduling yields an optimal solution when all vehicles have the identical initial and intersection velocities. However, if the vehicles have different initial or intersection velocities, FCFS scheduling is not optimal. Therefore, the MILP scheduling achieves shorter maximum travel time than the FCFS scheduling does. For instance, MILP scheduling can decrease the maximum travel time by up to 4.7% if the vehicles traveling in the north and south directions have the initial velocity of 7 m/s while the vehicles in the east and west directions have the initial velocity of 10 m/s.

Figs. 6(a) and 6(b) show the total travel time of the vehicles in each direction for the different weighting values with the initial and reference velocities of 7 and 8 m/s. To measure the travel time under the different vehicle densities on the roads, the number of vehicles is set to 12 and 8, i.e, $|\mathcal{I}_d| = 3$ and $|\mathcal{I}_d| = 2$, respectively, in Figs. 6(a) and 6(b). The weighting values of the vehicles in north and south directions vary from 0.1 to 1 with $w_{d\in\{e,w\}} = 1$. We observe that the total travel time of the vehicles in the north and south directions, i.e., $\sum_{i\in\mathcal{I}_s\cup\mathcal{I}_n}(t_{\text{out}} - t_0)$, increases when $w_{d\in\{n,s\}}$ decreases. In this case, a small value of $w_{d\in\{n,s\}}$ implies that next

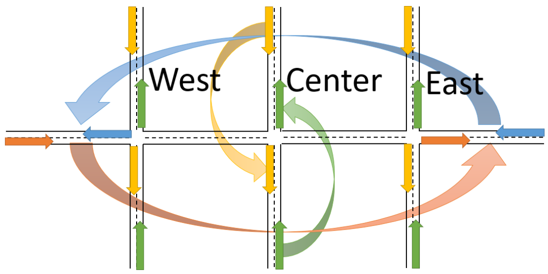


Fig. 7: Interconnected intersections.

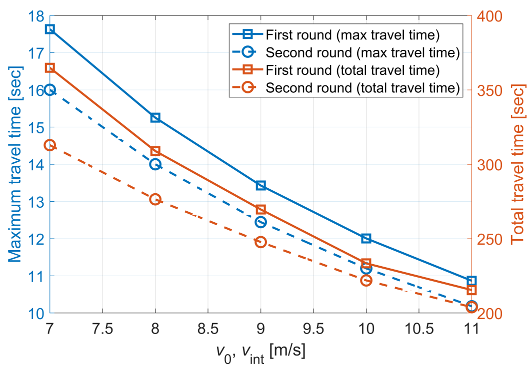


Fig. 8: Impact of the neighboring intersections.

intersection is highly congested. Therefore, the proposed edge computing system tends to delay north and south vehicles with small $w_{d\in\{n,s\}}$, thus reducing the travel time of the vehicles in east and west directions. For instance, in Fig. 6(a), the gap between the vehicles in different directions, in terms of the total delay, can be roughly 63.3% when $w_{d\in\{n,s\}} = 0.1$ and $v_{0,i} = v_{\text{int},i} = 7$ m/s.

To demonstrate the impact of the neighboring intersections, we have simulated a scenario of the three interconnected intersections, in which three intersections, denoted by west, center and east, respectively, are interconnected in parallel, and the unconnected roads of the intersections are wrapped around to other unconnected roads as shown in Fig. 7. Also, at each intersection, two vehicles travel in west and east directions, respectively, and four vehicles travel in north and south directions, respectively.

Fig. 8 shows the maximum travel time and the total travel time of the vehicles at an intersection during two consecutive simulation runs. The blue and red curves are the plot of left and right y-axises, respectively. First, Fig. 8 shows that the travel time decreases as v_0 and v_{int} increases. We can also see that the maximum travel time and total travel time in the second simulation round are reduced compared with the first simulation round. This is due to the fact that a small weighting value is applied to the vehicles traveling in the west and east directions. The vehicles with a small weighting value are delayed to exit the intersection in the first simulation round. Second, when the vehicles arrive at the neighboring intersections, their late arrivals enable the neighboring RSUs to schedule the vehicles in north and south directions with a priority. Therefore, the maximum travel time and total travel time of the vehicles at the next intersection can be reduced in the second simulation round. For example, the total travel time in the first simulation round can decrease by up to 14.3% in the second simulation round when $v_0 = v_{\text{int}} = 7$ m/s.

VI. Conclusion

In this paper, we have proposed an edge computing paradigm to optimize vehicle scheduling and vehicle dynamics control for the interconnected intersections. The intersection area is partitioned into three zones to perform the sequential procedures defined in the proposed edge computing. We have developed a communications protocol for the RSUs and the vehicles to exchange information. A directional weight metric is introduced to reflect the traffic condition at the neighboring intersection in a MILP-based scheduling problem, which is designed for the RSU to schedule optimal passing order and passing timing for the vehicles to cross the intersection. A QP-based trajectory planning problem is formulated for vehicles to control their mobility for driving comfort improvement and fuel consumption reduction. The proposed edge computing is a jointly distributed scalable solution that can be readily applied to the case of the interconnected intersections. Simulation results show that the proposed edge computing obtains the optimal scheduling while reducing the travel time of neighboring intersections.

References

[1] F. Ye, J. Guo, K. J. Kim, P. V. Orlik, H. Ahn, S. Di Cairano, and M. J. Barth, "Bi-level optimal edge computing model for on-ramp merging in connected vehicle environment," in *Proc. IEEE Intelligent Veh. Symp.*, Aug. 2019, pp. 2005–2011.

[2] G. Lee, W. Saad, and M. Bennis, "Online optimization techniques for effective fog computing under uncertainty," *IEEE MMTC Commun.-Frontiers*, vol. 12, no. 4, pp. 19–23, 2017.

[3] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-access edge computing: The driver behind the wheel of 5g-connected cars," *IEEE Commun. Standards Magazine*, vol. 2, no. 3, pp. 66–73, Oct. 2018.

[4] L. Li, Y. Li, and R. Hou, "A novel mobile edge computing-based architecture for future cellular vehicular networks," in *Proc. IEEE Wireless Commun. and Netw. Conf.*, San Francisco, CA, USA, May 2017, pp. 1–6.

[5] M. Emara, M. C. Filippou, and D. Sabella, "MEC-assisted end-to-end latency evaluations for C-V2X communications," in *Proc. of European Conf. on Netw. and Commun.*, Ljubljana, Slovenia, Jun. 2018, pp. 1–9.

[6] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. on Intelligent Transp. Systems*, vol. 17, no. 2, pp. 570–586, Sep. 2015.

[7] H. Ahn and D. Del Vecchio, "Safety verification and control for collision avoidance at road intersections," *IEEE Trans. on Automatic Contr.*, vol. 63, no. 3, pp. 630–642, Jul. 2017.

[8] M. A. S. Kamal, J.-i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Trans. on Intelligent Transp. Systems*, vol. 16, no. 3, pp. 1136–1147, Sep. 2014.

[9] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[10] M. Hausknecht, T.-C. Au, and P. Stone, "Autonomous intersection management: Multi-intersection optimization," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Dec. 2011, pp. 4581–4586.

[11] F. Altch, X. Qian, and A. de La Fortelle, "An algorithm for supervised driving of cooperative semi-autonomous vehicles," *IEEE Trans. on Intelligent Transp. Systems*, vol. 18, no. 12, pp. 3527–3539, Dec. 2017.

[12] J. C. Williams, "Macroscopic flow models," in *Revised monograph on traffic flow theory*. USA Federal Highway Administration, 1997.