A Novel Reinforcement Learning-based Cooperative Traffic Signal System through Max-pressure Control

Azzedine Boukerche, *Fellow, IEEE,* Dunhao Zhong, and Peng Sun, *Senior Member, IEEE*

***Abstract*—Improving the efficiency of traffic signal control is an effective way to alleviate traffic congestion at signalized intersections. To achieve effective management of the system-wide traffic flows, current research tends to focus on applying reinforcement learning (RL) techniques for collaborative traffic signal control in a traffic road network. However, the existing collaboration-based methods often ignore the impact of transmission delay for exchanging traffic flow information on the system. Most of the studies assume that the signal controllers can collect all instantaneous vehicular features without delay. To fill the gap, we propose an RL-based cooperative traffic signal control scheme considering the data transmission delay issue in a traffic road network. In this paper, we (1) design our new RL agents to cooperatively control the traffic signals by improving the reward and state representation based on the state-of-the-art max-pressure control theory; (2) propose a traffic state prediction method to address the data transmission delay issue by decreasing the discrepancy between the real-time and delayed traffic conditions; (3) evaluated the performance of our proposed work on both synthetic and real-world scenarios with a different range of data transmission delays. The results demonstrate that our method surpassed the performance of the previous max-pressure-based traffic signal control methods and addressed the data transmission delay issue.**

***Index Terms*—Cooperative traffic signal control, reinforcement learning, max-pressure control**

I. Introduction

Traffic signal control (TSC) is a challenge in transportation management research [1]. Since the traffic flow of a single intersection depends on its own traffic conditions and those of adjacent intersections [2], cooperative signal control is requested from the connected intersections in a road network. Cooperative traffic signal control of multiple intersections helps better regulate traffic flows in connected intersections, improve the efficiency of traffic management, and even prevent traffic congestion problem. The achievement of cooperative traffic signal control requires the ultra-low-latency traffic information on on-road vehicles and adjacent intersections.

Reinforcement learning (RL) algorithms, as a branch of artificial intelligence, have attracted research interests and seen increasing applications in addressing cooperative TSC problems [3]–[5]. The self-learning nature of RL makes it a powerful approach for formulating TSC problems. First, it does not depend on supervised learning labels, which require a large and reliable dataset to train the controller. Moreover, the online adaptive learning characteristic is embedded in the process of model learning [6]. This intrinsic characteristic of RL allows the signal controllers to update its model incrementally with new traffic observations in a dynamic traffic environment.

Many existing RL-based cooperative TSC methods tried to improve the efficiency of signal control based on real-time joint traffic information [3] or joint Q-functions [4] in an ideal traffic environment. However, they exposed several limitations. First, joint traffic information and Q-functions introduce high dimensions of traffic states and Q-values, increasing the learning complexity of RL methods on a large-scale traffic road network. Second, they all depend on an ideal assumption, i.e., the ideal traffic environment does not consider the data transmission delays in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. However, vehicular networks in real traffic environments are incapable of providing real-time traffic information without data transmission delays [7], [8]. Such transmission delays affect real-time traffic information collection and cooperative traffic signal control over a vehicular network.

To address the issues mentioned above, in this paper, we propose an RL-based cooperative TSC scheme with the following contributions:

1) A new traffic state representation method was designed based on queue length-based max-pressure measurements [9]–[11] that considers the traffic states of both local and adjacent intersections. The proposed representation lets the signal controllers know the upcoming traffic flows from adjacent intersections in the future steps.

2) We improved the reward model used in [12] by considering the capacity of connecting lanes and upcoming vehicles from adjacent intersections. The improved reward model avoids the traffic congestion in the case of letting vehicles move into fully occupied lanes, and reduces the frequency of switching traffic signals by considering the upcoming vehicle demands instead of local traffic conditions.

3) A new method is introduced to address the data transmission delay from approaching vehicles to intersections to enhance the state measurements for the real-time traffic conditions. Due to the data transmission delay in vehicular networks, we conducted comprehensive experiments with different ranges of data transmission delays to demonstrate the effectiveness of the method in the traffic environment.

The rest of the paper is organized as follows. Section II introduces the related work of traffic signal control on isolated and multiple intersections. Section III defines our problems of

Azzedine Boukerche, Dunhao Zhong, and Peng Sun are with the School of Electrical Engineering and Computer Science, University of Ottawa, Canada, 800 King Edward Ave., Ottawa, ON, Canada K1N 6N5.

This work is partially supported by NSERC-SPG, NSERC-DISCOVERY, Canada Research Chairs Program, NSERC-CREATE TRANSIT Funds.

this study. We describe our scheme in Section IV, and present performance evaluation in Section V. Finally, our conclusion and ideas for future work are presented at the end of this paper.

II. Related Work

In this section, we first discuss the previous works of RL-based TSC. According to the scope of control, existing RL-based TSC methods can be roughly categorized into two types: TSC on isolated intersections and cooperative TSC on multiple intersections.

A. Traffic Signal Control on Isolated Intersections

Traffic signal control on an isolated intersection aims to improve traffic control efficiency at an intersection based on the local traffic conditions. It does not consider the traffic conditions about adjacent intersections or even all intersections in a traffic road network. In the formulation of traffic signal control problem by reinforcement learning techniques, local traffic states of an intersection are described by several features, such as queue lengths of connecting lanes [13], [14], vehicle positions on lanes [15], etc. Tabular-based RL methods were applied in the early traffic signal control studies. These methods described traffic features by vectors or tables and stored Q-values of state-action pairs in Q-tables [16], [17]. However, these methods have limitations on storing high-dimensional traffic states when the scale of the traffic environment rises.

Deep reinforcement learning (DRL) has been investigated in recent years to address the issue of high-dimensional traffic states. DRL is an approximation-based RL technique that has the capability to generalize traffic states to hyperspace and learn neural network-based models to find the convergent Q-values [18]. Image-based traffic state representation was proposed to describe traffic features, especially vehicle positions on connecting lanes [19], [20]. However, the mutual effect of the traffic conditions among connected intersections has considerable impact on the traffic signal control in a traffic road network. Isolated traffic signal control cannot guarantee the optimal global control within the traffic road network.

B. Cooperative Traffic Signal Control on Multi-Intersections

Cooperative TSC methods enable all controller agents to control traffic signals cooperatively so as to achieve global optimization. E. Van Der Pol *et al.* [21] and S. Yang *et al.* [22] applied the max-plus algorithm to find the optimal joint actions among intersections. Max-plus enables an agent to select an optimal action based on payoff values among connected agents in a coordination graph [23]. A payoff value of an agent is repeatedly sent to its neighbours until the payoff is convergent at a point after finite iterations. Some studies achieved cooperative TSC by parallel learning both the local traffic environment and the environment of the adjacent or even global intersections to find the optimal actions [3], [24]. They presented traffic states by concatenating local traffic states with neighbour states. Another method of cooperative TSC is to learn joint Q-functions of local and adjacent intersections [4], [25], [26]. This method enables the agents to share Q-values among adjacent intersections and cooperatively learn the global optimal Q-function. Based on the joint Q-function theory, T. Tan *et al.* proposed a decentralized-to-centralized architecture for the TSC in a large-scale traffic road network [27]. The architecture first learns TSC models separately for each sub-region and then aggregates all Q-functions into a global Q-function for optimal system-wide joint actions. Max-pressure (MP) control is one of the state-of-the-art traffic signal control methods, which considers the queue lengths [9] or travel times of vehicles [11] on incoming and exiting lanes. Wei *et al.* applied MP control method to design the reward model of RL method to achieve cooperative TSC in arterial road networks [12]. They proved that the MP-based reward and traffic state design achieved a similar performance with the joint traffic state design. However, they ignored the impacts of the capacity of lanes on the traffic flows and congestion issues in the case when the signal controller continuously feeds vehicles to the exiting lanes with high occupancy. Also, when adjusting the green signals for the upcoming pressures, they ignored the pressures from adjacent intersections, which can increase the frequency of switching traffic signal phases. Higher switching times increase the duration of the yellow signal phases in the traffic road network, reducing the efficiency of traffic flow regulation at intersections. Moreover, previous researchers tend to ignore the data transmission delay of V2V and V2I communication. However, in a realistic traffic environment, such communication usually incurs a non-trivial transmission delay. The delay has a direct impact on the measurement of the traffic environment and traffic signal control.

In this paper, we aim to improve the efficiency of cooperative traffic signal control in multiple intersections based on the max-pressure algorithm in a more realistic traffic environment. We designed a traffic state representation and reward model by considering the pressures from adjacent intersections to improve the performance of cooperative traffic signal control. Furthermore, we proposed a scheme to address the data transmission delay of V2V and V2I communication in traffic environments and conducted comprehensive experiments to demonstrate the effectiveness of our method in different ranges of delays.

III. Problem Statement

As mentioned previously, in the urban road network, the traffic condition at an intersection depends on the signal control of the intersection as well as the signal controls and traffic flows of its adjacent intersections. Therefore, a cooperative traffic signal control method that considers the associated traffic conditions of those inter-dependent intersections is essential to achieve the system-wide/global optimal signal control. This study focuses on the cooperative traffic signal control problem in a traffic road network with multiple signalized intersections.

In this work, we assume that the adjacent signalized intersections can share their local traffic information (with each other) by exploiting the vehicular networks, as shown in Fig. 1. At every intersection, a traffic signal controller is

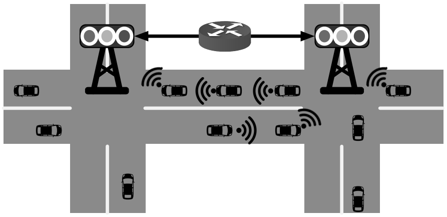


Fig. 1: The architecture of a vehicular network-assisted traffic signal control system.

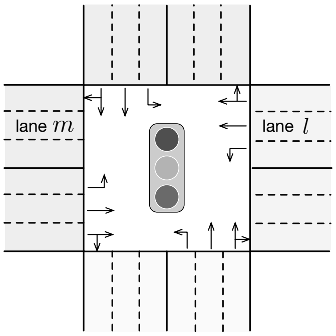


Fig. 2: Intersection definitions and the vehicular movements of lanes.

installed with network interfaces that can communicate with its adjacent intersections and on-road vehicles. The communication between mobile vehicles and intersections is transferred through the vehicular network so that the vehicle's data can be propagated through vehicles to the intersections. An on-road vehicle continuously sends its instantaneous data, such as speed and positions, to the approaching intersection per second after entering a lane of the intersection.

The intersection in our traffic environment connects with incoming and outgoing roads, respectively, (see Fig. 2). Each road of the intersection consists of three types of lanes for different vehicular movements, including right-turn (the right-most lane), left-turn (the left-most lane), and go-through (the rest of the middle lanes) vehicular movements. Traffic signals are grouped into several conflict-free traffic signal phases to regulate vehicular movements at the same time.

Left-turn protected and left-turn permitted traffic signal phase designs are two types of designs at an intersection [28]. Left-turn protected TSP design assigns an individual TSP for left-turn vehicular movements, and in the left-turn permitted TSP design, the left-turn vehicular movements share a TSP with go-through vehicular movements with yield priorities. The action space in the RL-based TSC method can be designed based on the two types of traffic signal phase designs. In our traffic environment, we apply left-turn permitted TSP design to the intersections to control the vehicular movements (see Fig. 3) and define the action space in our proposed method. The designs of the traffic signal phases and action space can be changed to the left-turn protected type according to different intersection scenarios.

Many studies have put considerable effort into designing traffic state representations and reward models in formulating the TSC problem by RL algorithms [29]. However, the associated traffic conditions of multiple intersections have higher dimensions of traffic states than the traffic conditions of a single intersection. Concatenating the traffic states of multiple intersections exponentially increases the dimensions of traffic states. It becomes worse in the case of adding more traffic features in the traffic state representations. Higher traffic state dimension requires higher learning cost in the process of agent learning. To reduce the dimensions of traffic states, we adopt the max-pressure (MP) method [9] in this study to design our traffic state representation and reward model. MP is a queue-based method that measures the pressure of an intersection by the number of vehicles queueing at the intersection and exiting the intersection. By taking the intersection shown in Fig. 2 as an example, the pressure of a vehicular movement from incoming lane l to exiting lane m is defined as follows:

$$P(l,m) = q(l,m) - q(m), \tag{1}$$

where $q(l,m)$ denotes the number of vehicles moving from lane l to lane m, $q(m)$ denotes the number of exiting vehicles on lane m. The pressure of the intersection is calculated as the absolute sum of pressures of all vehicular movements. We will elaborate on the specific modifications to the MP method in the next section.

Existing RL-based cooperative TSC methods, such as [12], ignored the data transmission delay issue in their traffic environment. However, vehicular network delay is inevitable in a real traffic environment. The signal controllers are incapable of receiving all traffic conditions in real-time because of some delayed vehicle information. Due to moving positions and dynamic directions of on-road vehicles, the traffic states generated by delayed information may not be identical to those in real-time traffic conditions [30]. The differences between delayed and real-time traffic states reduce the reliability of signal control policies. Therefore, we propose a state prediction method that can predict real-time traffic states based on the delayed vehicle information.

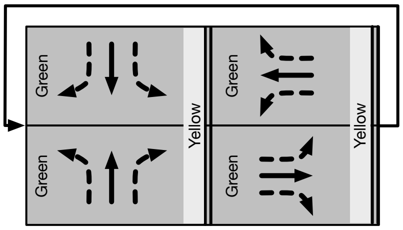


Fig. 3: Left-turn permitted traffic signal phases design [28]. The dashed arrows represent the permitted movements that must yield to through movements.

IV. The Proposed Max-pressure-based Cooperative Traffic Signal Control Method

This section describes our newly designed RL-based cooperative traffic signal control through max-pressure control (MP-CTSC) method for improving traffic control efficiency in multiple intersections with vehicular networks. We first present the framework of the proposed MP-CTSC, then describe the

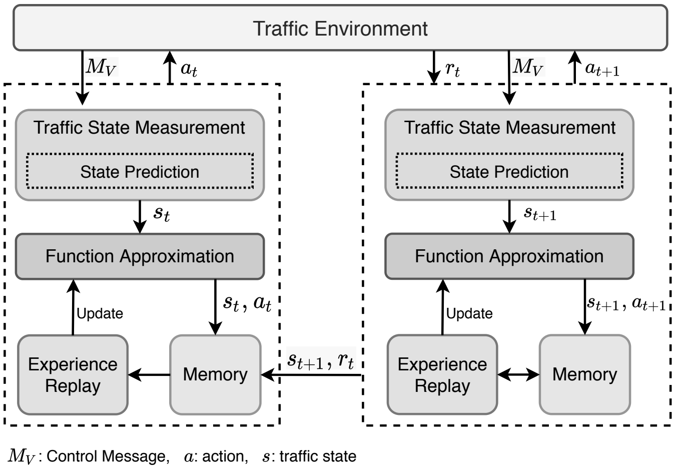


Fig. 4: The framework of the proposed MP-CTSC.

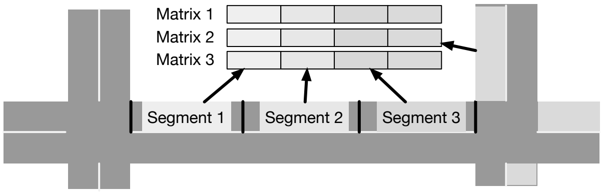


Fig. 5: Traffic state representation.

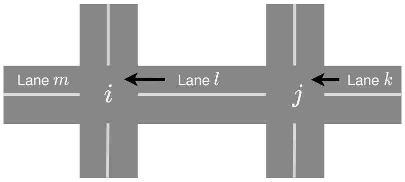


Fig. 6: Pressure movements between two intersections.

details of MP-CTSC method for smoothing traffic flows in connected multi-intersections.

A. Framework

Based on RL's architecture introduced in [6], the framework of the proposed MP-CTSC is mainly composed of a traffic environment and an agent. The traffic environment is a traffic road network containing multiple connected intersections. According to the environment described in Section III, the intersections can share their traffic information and receive the control messages of on-road vehicles through vehicular networks. The agent learns the collected traffic information and trains a function approximation model to find the optimal signal control action for the given traffic states. Due to the data transmission delay issue in vehicular networks, the agent has a traffic state measurement module to predict the real-time traffic conditions based on the delayed traffic information.

Fig. 4 shows the interactions between the traffic environment and the agent at time step t. The agent listens to continuously incoming control messages M_V from the traffic environment and transforms the messages to traffic states. Due to the delay for forwarding control message M_V, the traffic state measurement module predicts the traffic state, s_t, at time step t to reduce the discrepancy between the real-time and delayed traffic conditions. Given the predicted traffic state, the function approximation module generates the optimal action (a_t) for controlling the signals. The reward (r_t) for the executed action (a_t) and the next traffic state (s_{t+1}) are returned to the agent at time step $t+1$ and stored in memory with the s_t and a_t. The agent's experience is stored in the memory to improve the function approximation model through the experience replay process.

The following sections describe the agent design, function approximation module, and traffic state measurement module.

B. Agent Design

1) Traffic state representation: In this paper, we use four traffic features to describe the traffic states of an arbitrary intersection. These four features are local current traffic signal phase, vehicle volumes, average speed, and average acceleration. The current traffic signal phase is the index of the current phase. Vehicle volumes, average speed, and average acceleration are represented by three matrices M_1, M_2, M_3, respectively. Each matrix has dimension $|L| \times 4$, where L denotes the connecting lanes of the intersection, and $|L|$ is the number of the lanes. In these matrices, the row vectors represent the feature states of the vehicles moving in the corresponding lanes. As shown in Fig. 5, each lane is evenly divided into three segments in order to reduce the dimensions of the state representation. Cell 1, cell 2, and cell 3 of the rows describe the features of corresponding segments on the connecting lanes, and cell 4 indicates the states of approaching vehicles from adjacent intersections. For example, in Fig. 5, cell 1, cell 2, and cell 3 in M_1, M_2, and M_3 denote the number of vehicles, the average vehicle speed, and the average acceleration of the vehicles in the lane in segments 1, 2, and 3, respectively. Cell 4 in M_1, M_2, and M_3 describes the three features of approaching vehicles from the right adjacent intersection.

2) Action: Action space defines the actions that the agent can take to impact the environment [6]. Our action space is defined as $\mathbf{A} = \{a_1, a_2\}$ to represent two left-turn permitted TSPs, respectively. The action space $\mathbf{A}$ can also be defined as $\{a_1, a_2, a_3, a_4\}$ if the TSP design is left-turn protected TSPs. We define a minimum duration of a TSP (Δt) to avoid flickering TSP due to dynamic traffic states. The TSP cannot be switched to another TSP until the elapsed time of the current TSP is higher than Δt. A five-second yellow TSP is applied between two different TSPs. When a newly selected TSP is different from the current one, the agent switches to the five-second yellow phase before updating the current phase to the target TSP.

3) Reward: Reward frames the objectives of the application and assesses the executed action for improving the performance of the agent in future steps [6]. Here, we design a reward model based on MP that measures the pressure of vehicular movements at an intersection based on incoming and outgoing queue length measurements.

We assume that a traffic environment contains an inter-

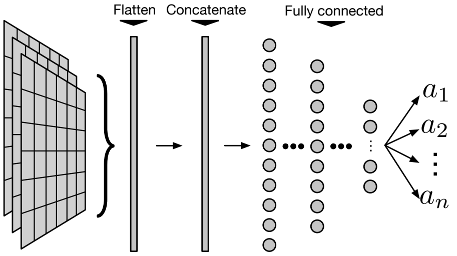


Fig. 7: The architecture of Q-Network.

section i and an adjacent intersection $j \in NB_i$, where the NB_i denotes the collection of neighbouring intersections of the intersection i (see Fig. 6). Then, we define the pressure of a vehicular movement from incoming lane l to outgoing lane m at time step t as follows:

$$P_t(l,m) = q_t(l,m) - c_t(m) + \sum_{k \in L_j} s_{j,t}(k,l) \cdot q_t(k,l), \quad (2)$$

where $q_t(l,m)$ denotes the number of vehicles moving from lane l to lane m, L_j denotes the incoming lanes of adjacent intersection j. $s_{j,t}(k,l) = \{1,0\}$ is a binary variable denoting the traffic signals of the vehicular movements from lane k to lane l. The value 1 means that the signal is green for the vehicular movement (k,l) at time step t; otherwise, it returns 0. $c_t(m)$ denotes the vacancy of exiting lane m that is available for incoming vehicles at time step t. The values of $c_t(m)$ describe the capability of reducing the intersection's pressure. When $c_t(m) = 0$, the lane m is full of vehicles and no space for feeding vehicles from upstream lanes. $c_t(m) > 0$ denotes the number of vehicles that can go through the intersection and enter into lane m.

Based on the pressure measurements described above, the pressure of intersection i at time step t is defined as

$$P_t = \sum_{l,m \in L_i} P_t(l,m). \quad (3)$$

The objective of the traffic signal controller is to minimize the total pressure of the intersection. Therefore, we define the reward r_t at time step t as follows:

$$r_t = -P_t. \quad (4)$$

C. Function Approximation

In this paper, we adopt Q-network as the function approximation to estimate Q-values of actions for the given traffic states. As shown in Fig. 7, the four features (one scalar, three matrices) of a given traffic state are flattened and concatenated into an array before passing through a neural network. The neural network consists of several fully connected layers and an output layer. In practical applications, the number of hidden layers of the network can be adjusted according to the actual conditions of different intersections. The output values of neurons in the output layer represent the Q-values of actions. The action with higher Q-value is selected by the agent to control traffic signals. ϵ-greedy algorithm is used in action selection. The agent randomly selects an action with a small probability, ϵ, and generally selects the optimal action with the highest Q-value, since the Q-value quantifies the agent's rewards in the future steps.

TABLE I: The field definitions of a control message content.

Field	Description
t_{sent}	The timestamp of a message sent by a vehicle
id	Global unique id of a vehicle
p_{sent}	Vehicle position at timestamp t_{sent}
s_{sent}	Vehicle speed at timestamp t_{sent}
a_{sent}	Vehicle acceleration at timestamp t_{sent}

To train the TSC model, we adopt experience replay techniques [18], [31] to update model's parameters θ. An experience cache is maintained in memory to store the encountered traffic states and executed actions and the corresponding rewards from the traffic environment. The cache is represented as a sequence queue structure with a maximum size and automatically removes the least recently inserted experience. To stabilize the learning progress, a target Q-network [18] is applied in our traffic signal model. Although similar to the structure of the learning Q-network, the target Q-network updates parameters θ^- based on θ with a lower frequency.

D. Traffic State Measurement

Data transmission delay is inevitable in wireless communications [32], especially in vehicular networks, due to dynamic traffic environments. Various factors can cause the delay of messages sent by a vehicle to an intersection, such as the vehicle speed, the distances between the vehicle to the intersection or the front vehicles. Therefore, the traffic states collected by the controller agent at every time step cannot represent real-time traffic conditions. Here, we design a traffic state measurement to address the delay in vehicular networks.

1) Contents of the Control Message: In our traffic state measurement, the content of a control message must capture the necessary information needed in our later state prediction. Accordingly, in this work, we designed a new control message consisting of five fields representing the instantaneous vehicle data transmitting from a vehicle to an intersection agent (see Table I). The first field is the timestamp t_{sent} of the message sent from the vehicle that creates the message. The timestamp field is used by the agent to estimate the delay of the message transmission. The second field is the globally unique id of the approaching vehicle. The third field is the position (p_{sent}) of the vehicle at timestamp t_{sent}. We assume that the position data can be fetched by the vehicle through GPS devices. To simplify the method's description, we define the vehicle position by the distance between the vehicle and the approaching intersection. The fourth and fifth fields are the speed profile information of the vehicle including the vehicle speed (s_{sent}) and acceleration (a_{sent}) at timestamp t_{sent}.

2) State Prediction: To reduce the adverse effect of data transmission delay on the control efficiency of the TSC system, we propose a method to predict the real-time traffic state based on the received messages of upcoming vehicles from adjacent intersections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3069921, IEEE Transactions on Vehicular Technology

Assuming that at time step t, an agent has received a sequence of messages M_V of upcoming vehicles V from adjacent intersection j in the previous control cycle, the delay of the message M_{V_i} from vehicle V_i to the intersection can be calculated as follows:

$$d(V_i) = t - M_{V_i}(t_{\text{sent}}). \tag{5}$$

Since the speed of vehicle V_i is constantly changing, the control agent does not know the instantaneous speed of the vehicle V_i during the delay period $d(V_i)$. Hence, we assume that the received vehicle speed s_{sent} is the average speed of the vehicle V_i in the delay period. Based on the speed assumption, the expected moving distance of V_i can be estimated as follows:

$$E(dis(V_i)) = d(V_i) \times M_{V_i}(s_{\text{sent}}) \tag{6}$$

Here, we analyze the expected position of the vehicle V_i in two cases:

Case 1: We assume that V_0 is the leading vehicle on its lane. The expected position $E(p_{V_0})$ of the vehicle V_0 is calculated based on the expected moving distance and the signal status of the adjacent intersection j as follows:

$$E(p_{V_0}) = \begin{cases} M_{V_0,l}(p_{\text{sent}}) - E(dis(V_0)), & \text{if signal} = \text{G} \\ \max\{0, M_{V_0,l}(p_{\text{sent}}) - E(dis(V_0))\}, & \text{otherwise} \end{cases} \tag{7}$$

When $E(p_{V_0}) < 0$ and the signal is green, the vehicle V_0 has already left the adjacent intersection j and moved into the incoming lane of the intersection i. In this case, we must re-estimate $E(p_{V_0})$ as V_0 is currently on the lane of the intersection i. Since we present the position of a vehicle by the distance between the vehicle and the intersection, we can recalculate the expected position of V_0 using $E(p_{V_0}) = len - |E(p_{V_0})|$, where len is the length of the lane at the intersection i.

Case 2: V_i is not the leading vehicle ($i \neq 0$). The expected position of the vehicle V_i is calculated by the expected moving distance and constrained by the expected position of the front vehicle V_{i-1}. By assuming the vehicle is not allowed to change the lane and overtake front vehicles, we can calculate the expected position of the vehicle V_i as follows:

$$\begin{aligned} E(p_{V_i}) = \max\{M_{V_i}(p_{\text{sent}}) - E(dis(V_i)), \\ E(p_{V_{i-1}}) + Gap\}, \end{aligned} \tag{8}$$

where Gap denotes the length of a vehicle plus the minimum gap between two vehicles.

E. Complexity Analysis

This section analyzes the space complexity and time complexity of the proposed MP-CTSC method.

1) Space Complexity: The space complexity of MP-CTSC consists mainly of two kinds of space: traffic states and control messages. According to the traffic state definition in Section IV-B1, the dimension of a single intersection state is $D = 4 \times |L| \times 3 + 1$, where $4 \times |L|$ denotes the dimension of a matrix, and the last dimension 1 represents the index of traffic signal phase. We assume that there is N number of the intersections in a traffic road network, then the space complexity of traffic states is $O(D \times N) \approx O(|L| \times N)$. In our control message definition in Section IV-D1, there are five features in a message. Hence, the space complexity of a message is constant time. We assume that the vehicle volume is $|V|$, then the total space of the control messages at a time step is $|V| \times 5$, meaning that the space complexity of control messages is $O(|V|)$. Therefore, the space complexity of MP-CTSC is $O(|L| \times N + |V|)$.

2) Time Complexity: The time complexity of MP-CTSC has two main components: the Q-networks for predicting Q-values and the traffic state measurements. The Q-network time complexity depends on the dimensions of its inputs, hidden layers, and outputs. Since those dimensions do not constantly change during the traffic signal control, we assume that the Q-network's time complexity at an intersection is a constant value. Moreover, we assume that all agents in a traffic road network can simultaneously predict Q-values; therefore, the Q-networks' time complexity can be presented as $O(1)$. The time complexity of the traffic state measurement depends on the number of vehicles in the traffic road network. If we assume that all agents can measure traffic states simultaneously, then the time complexity of the traffic state measurements is $O(|V|/N)$. Therefore, the time complexity of MP-CTSC is roughly equal to $O(|V|/N + 1) \approx O(|V|/N) \approx O(|V|)$.

V. Performance Evaluation

We conducted the experiments to evaluate the performance of our max-pressure-based cooperative traffic signal control (MP-CTSC) in the simulation of urban mobility (SUMO) [33]. SUMO is a free and open-source microscopic traffic simulation. It incorporates several libraries for routing and demands generation. We utilized JTRROUTER tool to generate traffic flows based on our vehicle arrival rates and TraCI library to control traffic signals and collect instantaneous vehicle information.

The measurements we used to evaluate the performance of our method fall into two categories: intersection-based and vehicle-based. The intersection-based measurements include queueing time and queue length at intersections. The vehicle-based measurements include vehicle waiting time, average vehicle speed and the number of stop times, and vehicle fuel consumption. Vehicle fuel consumption is estimated by Handbook Emission Factors for Road Transport (HBEFA) [34], which is extended in SUMO to estimate instantaneous vehicular fuel consumption based on vehicle speed and acceleration. HBEFA provides various factors to estimate fuel consumption and emission values. Based on different influences, such as vehicle mass and driving resistance, the factors are classified into seven emission standards from EURO 0 to EURO 6. Here, we adopt EURO 4, the default setting in SUMO, as our emission standard for gasoline passenger cars to estimate vehicle fuel consumption.

A. Traffic Environment

To evaluate the performance of our MP-CTSC method more accurately, we tested our work by considering two distinctive traffic scenarios.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3069921, IEEE Transactions on Vehicular Technology

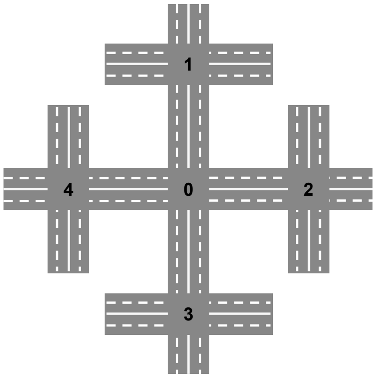


Fig. 8: Traffic road network structure in Traffic Scenario 1.

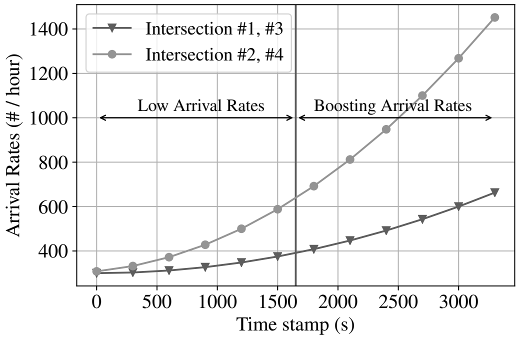


Fig. 9: The vehicle arrival rates in Traffic Scenario 1.

1) Traffic Scenario 1: We used SUMO to simulate a traffic environment with an intersection connecting with four adjacent intersections (see Fig. 8). Two-way link with two lanes on each road was applied to connect two adjacent intersections. The rightmost road lane allows go-through and right-turn vehicular movements, and the leftmost lane allows go-through and left-turn vehicular movements. Table II summarizes the basic parameters of vehicles and lanes. The traffic signals of intersection No. 0 is controlled by MP-CTSC, and rest of the intersections are controlled by pre-timed traffic signal phases. The pre-timed TSPs follow the left-turn permitted design in Fig. 3. We set 30 seconds for each green phase and 5 seconds for the yellow signal phase.

Fig. 9 shows the arrival rates of vehicles entering the traffic environment in an hour. Unbalanced and dynamic vehicle demands were used to imitate real-world traffic conditions. We simulated the vehicle demands from intersections No. 1 and 3 lower than the demands from intersections No. 2 and 4. Furthermore, the demands from intersections No. 2 and 4 increased faster than the demands of other intersections. We increased the demands gently in the first half-hour and then boosted the demands of intersections No. 2 and 4 in the second half-hour to evaluate the performance of methods on the boosting vehicle demands. The vehicle routes were generated by the JTRROUTER tool provided by SUMO with the turning ratios of 35% for right-turn movements, 40% go-through movements, and 25% for left-turn movements.

TABLE II: Parameters of vehicle and lane settings in SUMO.

Parameter	**Value**
Car following model	Krauss
Car acceleration	2.6 m/s
Car deceleration	4.5 m/s
Car length	4m
Minimum gap between two cars	1m
Lane speed limit	50 km/h
Lane length	200 m

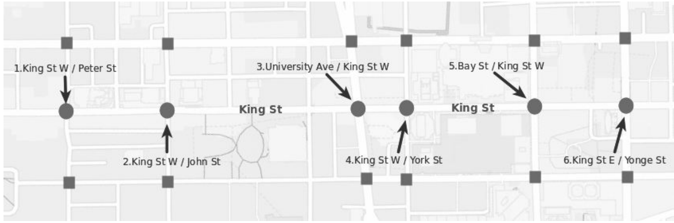


Fig. 10: A map of King St. in Toronto generated by OpenStreetMap[1].

In this scenario, in order to show the effectiveness of addressing data transmission delay issues by the proposed MP-CTSC, we estimated the average message transmission delay from vehicle V_i to an intersection as $\Delta D_i = k \times \delta$, where k denotes the expected number of hops that a message can be successfully transmitted to the intersection, δ denotes the average delay of one-hop [35]. Since our study focused on the urban traffic road network environment, we ignored the data transmission delay in sparse traffic conditions. We set the transmission range of V2V and V2I communication as 50 meters in our test environment and the delay factor $\delta = \{0\text{s}, 0.5\text{s}, 1\text{s}, 2\text{s}\}$ to evaluate the performance on different ranges of delay.

2) Traffic Scenario 2: We used SUMO to simulate another traffic environment based on a real-world traffic dataset with message transmission delays from vehicles to intersections. The traffic road network contained six intersections constructed by an arterial street named King St. and the other six minor streets in Toronto (see Fig. 10). The names of intersections from west to east are: 1) King St. W/Peter St. (I-KP); 2) King St. W/John St. (I-KJ); 3) University Ave./King St. W (I-UK); 4) King St. W/York St. (I-KWY); 5) Bay St./King St. W (I-BK); 6) King St. E/Yonge St. (I-KEY), respectively.

The vehicle arrival rates of the six intersections were generated by vehicle demands collected by a video-based counting system in Toronto [36], which counted the volumes of articulated trucks, bicycles, lights, pedestrians, and single-unit trucks on four approach legs (N, S, E, W). The vehicle demands of light vehicles on February 12th, 2018 was extracted to generate the vehicle arrival rates of the six intersections by

[1] https://www.openstreetmap.org

TABLE III: Parameters of Q-network learning process.

Parameter	Value
Experience memory pool D size	1000 [37]
Starting ϵ	0.5
Ending ϵ	0.1
Discount factor γ	0.8 [37]
Learning rate α	0.001 [20]
Minimum traffic signal phase interval Δt	5s [37]
Leaky ReLU β	0.01 [20]

the following system of equations:

$$\begin{aligned} V_N(NB) = V_E(WB) \times R_r(E) + V_S(NB) \times R_t(S) \\ +V_W(EB) \times R_l(W) \\ V_S(SB) = V_N(SB) \times R_t(N) + V_E(WB) \times R_l(E) \\ +V_W(EB) \times R_r(W) \\ V_E(EB) = V_N(SB) \times R_l(N) + V_S(NB) \times R_r(S) \\ +V_W(EB) \times R_t(W) \\ V_W(WB) = V_N(SB) \times R_r(N) + V_S(NB) \times R_l(S) \\ +V_E(WB) \times R_t(E) \end{aligned} \tag{9}$$

subjecting to $R_r(N) + R_l(N) + R_t(N) = 1$, $R_r(S) + R_l(S) + R_t(S) = 1$, $R_r(W) + R_l(W) + R_t(W) = 1$, $R_r(E) + R_l(E) + R_t(E) = 1$, where $R_r(N)$, $R_l(N)$, $R_t(N)$, $R_r(S)$, $R_l(S)$, $R_t(S)$, $R_r(W)$, $R_l(W)$, $R_t(W)$, $R_r(E)$, $R_l(E)$, $R_t(E)$ denote the turning ratios of approach legs, $V_N(SB)$, $V_S(NB)$, $V_E(WB)$, $V_W(EB)$ denote the incoming vehicle demands from approach legs, $V_N(NB)$, $V_S(SB)$, $V_E(EB)$, $V_W(WB)$ denote the existing vehicle volumes on the four legs.

B. Parameter Settings

Based on the Q-network architecture shown in Fig. 7, we manually tuned the hyper parameters of the Q-network by adjusting the number of hidden layers from 4 to 2 and hidden units in the values of [16, 32, 64, 128]. We found that these hyper parameters have trivial influence on the experimental results in our traffic scenarios. To simplify the Q-network model and reduce learning cost, we constructed a two-hidden-layer Q-network as our function approximation. The input traffic states are first fed in flattened and concatenated layers and then two fully connected layers. The first fully-connected layer consists of 128 Leaky ReLU units. The second fully-connected layer consists of 32 Leaky ReLU units. The output layer is a fully-connected layer with two outputs for two actions, each representing the corresponding traffic signal phase.

Table III presents the parameters of training Q-network progress referring to the parameters in [20], [37]. The ϵ decays from 0.5 to 0.1 by a factor 0.996 at every traffic signal control cycle. We trained our Q-network based on the sampled experience per control cycle and updated the parameters of our target Q-network every 300 control cycles. We set the minimum traffic signal phase interval $\Delta t = 5$s to avoid flickering TSPs. Traffic signal control cycle was set to 5s, which was the same as the TSP interval Δt, meaning that the agent predicted the real-time traffic states based on the delayed vehicle data in the previous 5 seconds.

C. Compared Methods

We conducted performance comparisons with the following methods.

1) Pre-timed Traffic Signal Control: Pre-timed TSC is a natural way to control traffic signals with manually designed traffic signal phases. Here, we set 30 seconds for each green TSP and 5 seconds for the yellow warning phase.

2) Max-pressure Traffic Signal Control: Max-pressure TSC [9] is an actuated control method that greedily selects a TSP based on the real-time pressure of vehicular movements. The pressures of vehicular movements are measured by incoming and exiting queue lengths without considering the capacity of lanes.

3) Travel Time-based Max-pressure Traffic Signal Control: TTMax-pressure TSC [11] is an actuated control method that greedily selects a TSP based on the travel time of the incoming and outgoing lanes.

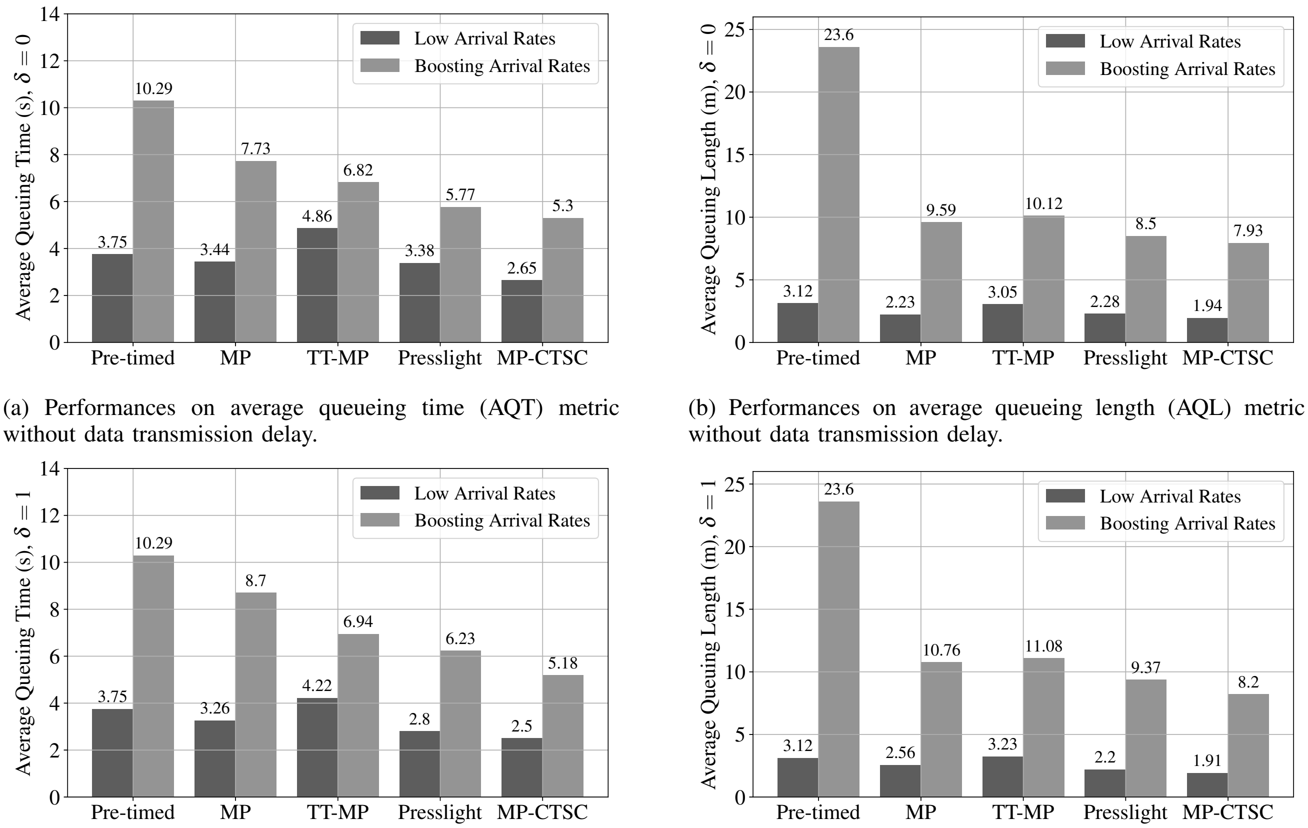
4) Presslight Traffic Signal Control: Presslight [12] is a RL-based TSC method that uses scaled max-pressure algorithm [38] to design their reward model in learning progress. The pressures of vehicular movements are measured by the difference between the occupancy of incoming and exiting lanes.

D. Simulation Results on Synthetic Data

1) Performance on Intersection-based Measurements: Fig. 11 shows the performances on the synthetic traffic scenario using the intersection-based measurements of average queueing time (AQT) in seconds and average queueing length (AQL) in meters at intersection No. 0. According to the vehicle arrival rates shown in Fig. 9, we compared the performance on the low changing arrival rates and boosting arrival rates.

Fig. 11a and Fig. 11b are results of AQT and AQL metrics, respectively, without data transmission delay, meaning that the agent can receive instantaneous vehicle data immediately. We can see that all compared methods had similar performances on AQT and AQL metrics in the period with low arrival rates. Actuated and RL-based TSC methods only contributed to slightly better results compared to pre-timed TSC.

When the arrival rates boosted, the performances of the methods showed significant differences. With the increasing vehicle demands in the second half-hour, MP had the worst performance on the measurement of AQT. This occurred because the objective of MP is to maximize the throughput of the intersection, which assigns the green signals to the lanes with higher pressures, leading to a higher frequency of switching TSPs in the scenario where the number of queueing vehicles is changing from different lanes. TT-MP performed better than MP on the AQT metric due to the travel time-based pressure measurements. Presslight improved the performance of MP on AQT metric by using RL techniques to achieve adaptive TSC for long-term traffic conditions. It reduced the frequency of switching TSPs compared to the actuated TSC methods. MP-CTSC had the best performance on both AQT and AQL metrics, compared to other methods. Based on the same advances of RL techniques applied in Presslight method, MP-CTSC considered the traffic pressures of adjacent



(a) Performances on average queueing time (AQT) metric without data transmission delay.

(b) Performances on average queueing length (AQL) metric without data transmission delay.

(c) Performances on average queueing time (AQT) metric with data transmission delay.

(d) Performances on average queueing length (AQL) metric with data transmission delay.

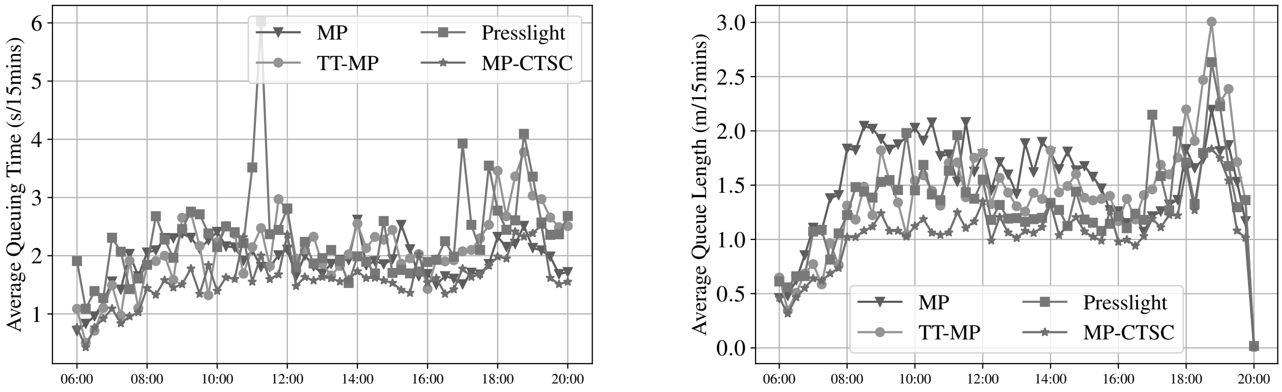
Fig. 11: Performances of the intersection-based measurements at the intersection No. 0 in Traffic Scenario 1.

TABLE IV: Performances on vehicle-based measurements with respect to waiting time (WT), average stops (Avg. Stops), average speed (Avg. Speed), and fuel consumption (FC) at intersection No. 0 in Traffic Scenario 1.

Metric	**Pre-timed**	**MP**		**TT-MP**		**Presslight**		**MP-CTSC**			
Delay δ	-	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 1$	$\delta = 0$	$\delta = 0.5$	$\delta = 1$	$\delta = 2$
Waiting Time (s)	27.7	14.6	15.4	15.7	16.9	13.1	13.7	12.1	11.9	11.6	12.9
Avg. Speed (m/s)	5.6	6.2	5.6	5.8	5.7	6.2	6.3	6.5	6.6	6.5	6.4
Avg. Stops	1.44	1.01	1.20	1.08	1.05	0.96	0.99	0.94	0.84	0.89	0.92
Fuel Consumption (ml/km)	45.5	29.1	29.9	30.2	31.6	26.9	27.6	26.6	25.8	25.7	27.0

intersections. Such consideration assigned longer green time to the lanes with higher pressures, including the local pressures and the incoming pressures from adjacent intersections, allowing the agent to keep green signals to the lanes with lower local pressures but higher incoming pressures. While in the Presslight without considering the adjacent pressures, the agent switches green signals to the lanes with higher local pressures, which may be different to the current TSP, and then switches back until the incoming pressures from adjacent intersections move to the local pressures. More switching TSP times results in longer yellow signal duration, resulting in lower TSC efficiency.

Fig. 11c and Fig. 11d are results of AQT and AQL metrics, respectively, with data transmission delay. We set $\delta = 1$ to ensure that the agent can only receive the instantaneous vehicle data within one hop, meaning that the agent cannot receive the vehicle data with a delay of more than one second. Pre-timed traffic signal control showed the same performance in Fig. 11a and Fig. 11b because it did not detect the traffic conditions. MP, TT-MP, and Presslight performed worse in the traffic environment with data transmission delay compared to $\delta = 0$. This was because the agent was unable to detect the vehicles that had long distances to the intersection due to data transmission delay, resulting in lower pressures of the lanes that contained the undetected vehicles. Such delayed pressure measurements misguided the agent to control the traffic signals, leading to worse performances compared to the scenarios without data transmission delay. We can see that MP-CTSC kept the same performances on both traffic conditions owing to the traffic state measurements with prediction. The



(a) Performances on average queueing time (AQT) metric.

(b) Performances on average queueing length (AQL) metric.

Fig. 12: Performances of intersection-based measurements at six intersections in Traffic Scenario 2 with data transmission delay $\delta = 1$.

TABLE V: Performances of vehicle-based measurements in Traffic Scenario 2.

Metric	Pre-timed	MP	TT-MP	Presslight	MP-CTSC			
Delay δ	-	$\delta = 1$	$\delta = 1$	$\delta = 1$	$\delta = 0$	$\delta = 0.5$	$\delta = 1$	$\delta = 2$
Waiting Time (s)	14.5	10.2	12.5	9.2	7.4	7.5	7.4	7.9
Avg. Speed (m/s)	6.6	6.9	6.8	7.6	7.9	7.6	7.8	7.1
Avg. Stops	0.69	0.67	0.67	0.52	0.47	0.51	0.46	0.51
Fuel Consumption (ml/km)	34.8	29.2	31.6	28.4	26.6	26.9	26.5	28.0

TABLE VI: The average green signal duration (GSD) and yellow signal duration (YSD) at six intersections in Traffic Scenario 2.

Method	GSD (s)	YSD (s)	Percentage of YSD
MP	36195	14509	0.29
TT-MP	36324	14377	0.28
Presslight	37504.6	13321.4	0.26
MP-CTSC	39110	11598	0.23

agent reused the delayed vehicle data to predict the traffic states at the current timestamp, reducing the gap between the traffic states with and without data transmission delay.

2) Performance on Vehicle-based Measurements: Table IV summarizes the performances of the compared methods on vehicle-based measurements for waiting time (WT) in seconds, stop times, average speed (m/s), and fuel consumption (ml/km) in Traffic Scenario 1. We can see that the pre-timed method resulted in the highest fuel consumption, since the high frequency of go-stop movements can cause higher fuel consumption [39]. MP, TT-MP, and Presslight performed slightly worse in the traffic environment with data transmission delay than the environment without delays. Such results are because of the loss of traffic state measurements due to delayed vehicle data. MP-CTSC outperformed other compared methods on those metrics in both traffic conditions $\delta = 0$ and $\delta \neq 0$. We evaluated MP-CTSC with different delay ranges to demonstrate the effectiveness of our state prediction method. The performance of MP-CTSC was close to each other in all delay conditions, meaning that MP-CTSC with traffic state prediction can effectively address the issue of delayed traffic states.

E. Simulation Results on Real-world Data

We further compared the methods in a real-world traffic scenario with a delay factor $\delta = 1$. Fig. 12 shows the performance of intersection-based measurements in Traffic Scenario 2, excluding the performance of the pre-timed method, because it performed worse than other methods. The figures present the average AQL and AQT per 15 mins of all six intersections. We can see that their performances were close to each other, but MP-CTSC performed more stably in the dynamic traffic flows. MP, TT-MP, and Presslight performances fluctuated more than MP-CTSC due to delayed vehicle data. Table V presents the results of vehicle-based measurements in Traffic Scenario 2. The vehicles had the lowest waiting time, stop times, fuel consumption, and the highest speeds in the real-world traffic scenario, meaning that the traffic signal control of MP-CTSC is the most efficient compared with other methods. The reason behind these results is due to traffic state measurements with prediction. With the delay factor $\delta = 1$, the agents can only receive the vehicle data within the previous second and miss the data that is delayed by more than one second. Such missing vehicle data presents a delayed traffic condition at every time step for the agents, leading to a poor decision in the event that the agents assign red signals to the lanes with high pressure. MP-CTSC addresses the data transmission delay issue by traffic state measurements with prediction. It collected traffic vehicle data per second and predicted the current traffic state based on the previous 5-second traffic states. The state prediction reduces the gap between the real-time traffic states and the delayed traffic states.

On the other hand, when considering the pressure of adjacent intersections, our method reduced the frequency of switching TSPs and the duration of yellow signal phases, indicating that MP-CTSC improved the efficiency of the traffic control. Table VI shows that MP-CTSC had the lowest yellow signal duration compared to other methods, meaning that MP-CTSC had fewer switching times during the test. Since the yellow signal phase happened only when the agent attempted to switch the current traffic signal phase to another phase, a higher frequency of switching traffic signal phases can increase the duration of yellow time and reduce the efficiency of the traffic signal control at intersections, because all vehicles must stop during the yellow signal phase.

We also evaluated the performance of MP-CTSC with the different ranges of the delay factor $\delta = \{0, 0.5, 1, 2\}$ to show the effectiveness of the proposed traffic state prediction method in the real-world traffic scenario. From Table V, we can see that the increasing data transmission delay did not seriously worsen the performances of MP-CTSC in the real-world traffic scenario. The performance in the traffic condition with delay factor $\delta \leq 1$ had the closest performance with the delay factor $\delta = 0$, and we found that the performance of $\delta = 2$ was worse compared to the performances of lower delay factors. It shows that the traffic state prediction method can thoroughly address the data transmission delay within one second per hop in our traffic scenarios. A longer delay may reduce the accuracy of the prediction and lower the efficiency of the traffic signal control.

VI. Conclusion

In this paper, we proposed a reinforcement learning-based cooperative traffic signal control scheme to promote the efficiency of cooperative TSC in multiple intersections. Our method considers the traffic conditions of both the local intersection and adjacent intersections. We proposed a max-pressure-based traffic state representation and a reward model to develop an agent that can evaluate a local pressure and the pressure from its adjacent intersections to avoid the unnecessary controls of switching traffic signal phases. Moreover, we proposed a traffic state prediction method to address data transmission delay in vehicular network environments. We evaluated our traffic state prediction method by setting different ranges of delays. The results show that our method achieved good performances on both vehicle-based and intersection-based metrics under boosting vehicle demands compared with previous max-pressure control methods. Also, our state prediction method can significantly reduce the gap between the real-time traffic states and the delayed traffic states. Our method demonstrated the ability to address the data transmission delay issue in vehicular networks.

In future work, we will adopt state-of-the-art techniques to optimize and fine-tune the hyper-parameters of the function approximation model to improve the proposed method's performance. Moreover, we will further evaluate the proposed method in a large-scale traffic road network with oversaturated vehicle demands and a more realistic traffic environment.

References

[1] M. A. Taylor, "Intelligent transport systems," *Handbook of transport systems and traffic control*, vol. 3, p. 461, 2001.

[2] N. Kheterpal, K. Parvate, C. Wu, A. Kreidieh, E. Vinitsky, and A. Bayen, "Flow: Deep reinforcement learning for control in SUMO," *EPiC Series in Engineering*, vol. 2, pp. 134–151, 2018.

[3] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, 2013.

[4] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates v2x networks dynamic clustering," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 8667–8681, 2017.

[5] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proc. ACM CIKM*, 2019, pp. 1913–1922.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction, 2nd ed.* MIT press, 2018.

[7] N. Aljeri and A. Boukerche, "Mobility and handoff management in connected vehicular networks," in *Proc. ACM MobiWac*, 2018, p. 8288.

[8] S. Hosseininezhad and V. C. Leung, "Data dissemination for delay tolerant vehicular networks: Using historical mobility patterns," in *Proc. ACM DIVANet*, 2013, p. 115122.

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

[10] J. Lioris, A. Kurzhanskiy, and P. Varaiya, "Adaptive max pressure control of network of signalized intersections," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 19–24, 2016.

[11] P. Mercader, W. Uwayid, and J. Haddad, "Max-pressure traffic controller based on travel times: An experimental analysis," *Transp. Res. Part C: Emerging Technologies*, vol. 110, pp. 275–290, 2020.

[12] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. ACM SIGKDD*, 2019, pp. 1290–1298.

[13] Y. K. Chin, N. Bolong, A. Kiring, S. S. Yang, and K. T. K. Teo, "Q-learning based traffic optimization in management of signal timing plan," *Intl. J. Sim. Sys. Sci. Technol.*, vol. 12, no. 3, pp. 29–35, 2011.

[14] S. Araghi, A. Khosravi, M. Johnstone, and D. Creighton, "Q-learning method for controlling traffic signal phase time in a single intersection," in *Proc. IEEE ITSC*, 2013, pp. 1261–1265.

[15] P. Ha-li and D. Ke, "An intersection signal control method based on deep reinforcement learning," in *Proc. IEEE ICICTA*, 2017, pp. 344–348.

[16] D. Houli, L. Zhiheng, and Z. Yi, "Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, p. 724035, 2010.

[17] K. Wen, W. Yang, and S. Qu, "A stochastic adaptive traffic signal control model based on fuzzy reinforcement learning," in *Proc. IEEE ICCAE*, vol. 5, 2010, pp. 467–471.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[19] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv preprint arXiv:1611.01142*, 2016.

[20] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, 2019.

[21] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *NIPS Workshop MALIC*, 2016.

[22] S. Yang, B. Yang, H.-S. Wong, and Z. Kang, "Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm," *Knowledge-Based Systems*, vol. 183, p. 104855, 2019.

[23] J. R. Kok and N. Vlassis, "Using the max-plus algorithm for multiagent decision making in coordination graphs," in *Robot Soccer World Cup*. Springer, 2005, pp. 1–12.

[24] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.

[25] M. A. Khamis and W. Gomaa, “Enhanced multiagent multi-objective reinforcement learning for urban traffic light control,” in *Proc. IEEE ICMLA*, vol. 1, 2012, pp. 586–591.

[26] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, “Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control,” *IEEE Access*, vol. 7, pp. 40 797–40 809, 2019.

[27] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, “Cooperative deep reinforcement learning for large-scale traffic grid signal control,” *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2687–2700, 2019.

[28] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beaird, S. Tsoi, P. Ryus, D. Gettman *et al.*, *National Cooperative Highway Research Program Report 812 - Signal Timing Manual, 2nd ed.* Washington, DC: Transportation Research Board, 2015, ch. 5, pp. 1–19.

[29] P. Mannion, J. Duggan, and E. Howley, “An experimental review of reinforcement learning algorithms for adaptive traffic signal control,” in *Autonomic Road Transport Support Systems*. Springer, 2016, pp. 47–66.

[30] P. Sun, N. AlJeri, and A. Boukerche, “A novel proactive handover scheme for achieving energy-efficient vehicular networks,” in *Proc. ACM Q2SWinet*, 2018, p. 2328.

[31] L.-J. Lin, “Reinforcement learning for robots using neural networks,” Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1993.

[32] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[33] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Fltterd, R. Hilbrich, L. Lcken, J. Rummel, P. Wagner, and E. WieBner, “Microscopic traffic simulation using SUMO,” in *Proc. IEEE ITSC*, 2018, pp. 2575–2582.

[34] M. Keller, “Handbook of emission factors for road transport (HBEFA) 3.1,” INFRAS, Tech. Rep., 2010.

[35] L. Du and H. Dao, “Information dissemination delay in vehicle-to-vehicle communication networks in a traffic stream,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 66–80, 2014.

[36] Toronto Transportation Services. (2018) King st. transit pilot - detailed traffic & pedestrian volumes. Accessed on: August 13st, 2020. [Online]. Available: https://ckan0.cf.opendata.inter.prod-toronto.ca/dataset/5a4e2ef7-8eab-45da-9080-9956a8605229/resource/a5efb524-f062-48e9-84a0-589b12f0754b/download/detailed-traffic-pedestrian-volumes-2018.gz

[37] H. Wei, G. Zheng, H. Yao, and Z. Li, “Intellilight: A reinforcement learning approach for intelligent traffic light control,” in *Proc. ACM SIGKDD*, 2018, pp. 2496–2505.

[38] A. Kouvelas, J. Lioris, S. A. Fayazi, and P. Varaiya, “Maximum pressure controller for stabilizing queues in signalized arterial networks,” *Transp. Res. Rec.*, vol. 2421, no. 1, pp. 133–141, 2014.

[39] M. Barth and K. Boriboonsomsin, “Real-world carbon dioxide impacts of traffic congestion,” *Transp. Res. Rec.*, vol. 2058, no. 1, pp. 163–171, 2008.