Decentralized Signal Control with MEC: A Comparison Between Different Network Decompositions

Can Zhang
Graduate Research Assistant
[1]*Department of Civil and Environmental Engineering*
University of Alberta, 6-362 DICE
Edmonton, Alberta, Canada T6G 2H5
Tel: 1-587-989-1219
Email: can7@ualberta.ca

Amy Kim
Associate Professor
1Department of Civil and Environmental Engineering
University of Alberta, 6-269 DICE
Edmonton, Alberta, Canada T6G 2H5
Tel: 1-780-492-9203
Email: amy.kim@ualberta.ca

Tony Z. Qiu, Ph.D.*
Associate Professor
[1]*Department of Civil and Environmental Engineering*
University of Alberta, 6-271 DICE
Edmonton, Alberta, Canada T6G 2H5
Tel: 1-780-492-1906, Fax: 1-780-492-0249
Email: zhijunqiu@ualberta.ca
[2]*Intelligent Transportation Systems Research Center*
Wuhan University of Technology
Mailbox 125, No. 1040 Heping Road
Wuhan, Hubei, China 430063

Word count: 7057 words + 1 table x 250 words = 7307 words

***Corresponding author**
[1]Primary Affiliation
[2]Secondary Affiliation

ABSTRACT

The optimization of large-scale network traffic signal controls can involve large amounts of decision-making variables and non-linear constraints, making it a Nondeterministic Polynomial Time (NP)-complete problem. In addition, communications between vehicles and infrastructures are unable to meet the latency requirement for real-time control. To address these issues, this paper proposes the application of a Decentralized Signal Control (DSC) system with Mobile Edge Computing (MEC) for large-scale networks and investigates the impact of different network decompositions. In addition, the development of an Intelligent Transportation System (ITS) underpinned by 5G network capabilities and MEC has increased the possibilities for real-time traffic optimization in the decentralized scenario. Our study is therefore able to divide a network into subnetworks, each of which has its own traffic management center (Sub-TMC) that optimizes its signals. Based on the assumption that each traveler entering the network is assigned a route that will not change, routes can be divided into sub-routes for each subnetwork accordingly. This paper also compares several network decompositions. We use numerical simulations based on a G/G/n/FIFO Queueing Network Model (QNM), and different decompositions are accessed based on computation efficiency scaled via computation time cost and traffic efficiency scaled via average total travel cost. Results show that network decomposition with smaller subnetworks results in better computation efficiency while reducing traffic efficiency. This paper concludes by considering the balance of computation and traffic efficiency as the key to DSC systems applications.

Keywords: Decentralized Signal Control (DSC), Mobile Edge Computing (MEC), Intelligent Transportation System (ITS), Queueing Network Model (QNM)

INTRODUCTION

The motivations for optimizing the signal timing control plan for an urban transportation network are to increase efficient network capacity utilization, improve vehicle mobility, and reduce traffic emissions. After all, if the system is optimized, then vehicles spend less time travelling. However, the current computing capabilities of the centralized traffic management centre, coupled with the higher latency, lower bandwidth, and reduced connectivity of 4G, mean that network-wide signal control optimization remains elusive since it requires high levels of data availability and accessibility.

Rapidly developing Connected Vehicle (CV) technologies that produce increasingly numerous types of vehicle data, personal travel data, and geometric data, however, offer the potential to resolve the issue of data quality. For instance, the 2017 SAE J2735 Dedicated Short Range Communications (DSRC) report summarizes 17 messages, 156 data frames, 230 data elements, and 58 external data element definition references *(1)*. In addition, vehicles and transportation infrastructure can exchange data in real time via Vehicle-to-Everything (V2X) communications technology. As such, the issue of data quality would no longer pose a significant barrier for the successful implementation of network-level traffic signal control.

However, computational capacity limitations remain despite excellent algorithms, AI technology, and large-capacity computers. The primary issue rests with the fact that network-level traffic signal control is highly complex, nonlinear, and affected by many internal and external factors that are part of the traffic system. The network-level traffic signal optimization problem has been proven to be an NP-complete problem *(2)*.

A significant limitation is connected to the issue of latency. Control latency usually results from both data analysis, which suffers from computing capacity limitations, and data delivery, which traditionally requires all the data be moved to the cloud in order to increase computing speed. However, despite data processing speeds having increased rapidly, the bandwidth of the internet network has not followed suit, thereby limiting the transmission of data between vehicles, infrastructure and the central traffic management center in real-time.

Mobile Edge Computing (MEC) can alleviate the effects of data delivery latency. MEC handles data on the devices where it is generated rather than data being sent to and received from the Traffic Management Center (TMC) cloud that takes longer and therefore compromises real-time decision-making at the local level. **Figure 1** graphically presents the data flows of the CV environment without MEC (solid arrows only) and with MEC for visual comparison between.

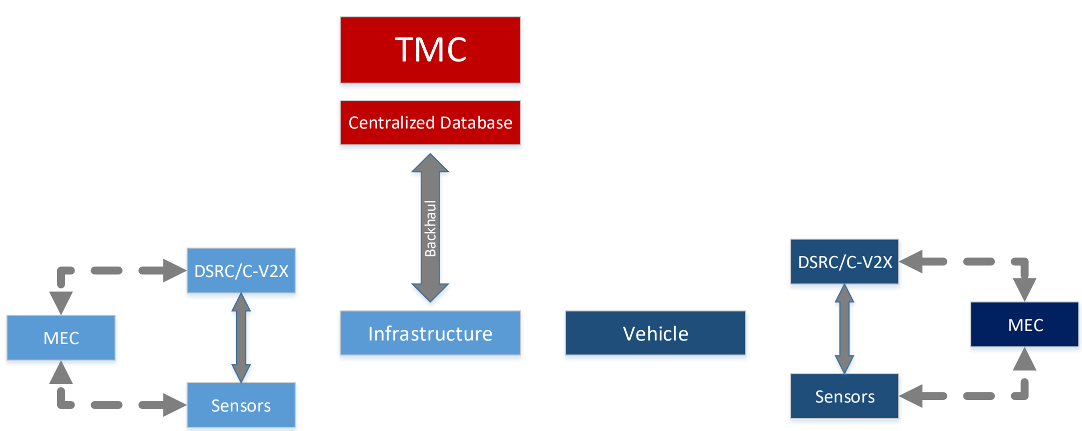


Figure 1 CV environment with/without MEC

The intercommunication in both cases is between three major parts: the TMC, infrastructure, and vehicles. In the traditional ITS architecture without MEC, all data is sent to and processed at a centralized TMC. However, the limited bandwidth of the backhaul connection between the centralized TMC (Central-TMC) and local infrastructure will be exhausted rapidly by the volume of data gathered by the increasing number of sensors in the ITS. Likewise, the latency will be insufficient for CV applications that require large amounts of CV data.

After introducing MEC into the framework with MEC, data can be processed and applications can be run on any local edge unit, regardless if it is an infrastructure edge or vehicle edge. Thus, the heavy computation workload for the Central-TMC can be evenly distributed to the lower-level mobile edges thereby increasing computation capacity and proximity for real-time data processing.

To build the control system in this research, an MEC-enabled CV environment is proposed, within which vehicles and infrastructure can exchange data via DSRC. MEC devices are located at each intersection of the network that collect and deliver data and control the traffic signal lights at the adjacent intersection. The Edge Computing Unit (ECU) at each intersection includes a computing server, roadside equipment (RSE), and a controller. Each vehicle has onboard equipment (OBE), and each TMC has a cloud computing server; the assumption is that data is provided by the sensors and CV technologies for the control system. Data sets include the real-time traffic origin-destination (OD) demand of the urban network (i.e. the origins of traveler network entry, their end point of travel, and the distribution for each OD pair), geographical information about the network, and emission of each individual vehicle (if required), amongst others.

To reduce the latency of data analysis for large-scale network projects, we introduce a Decentralized Signal Control (DSC) system. DSC systems decompose large-scale networks into small subnetworks and simultaneously optimize the parameters of all intersections in the subnetwork collectively. A DSC system requires subnetworks that can exchange information in real time and over a short period. Each subnetwork has an 'agent' (or Sub-TMC). Each agent is designed to receive messages from surrounding agents, make decisions for the traffic signal control of intersections inside the subnetwork, and send messages to surrounding intersections. As such, decentralized systems reduce the latency of data analysis by breaking the information down into smaller parcels, thereby making them easier to deliver and analyze. The latency of a DSC system is significantly lower than that created by a Centralized Signal Control (CSC) system. Hence, we would assert that research is best directed towards investigating and developing a DSC system.

Considering these issues, there are two objectives in this study. The first objective is to develop a new structure for the DSC system, particularly for large-scale networks. The second objective is to compare the DSC and CSC systems and, simultaneously, investigate the impact of different network decompositions for DSC system.

To achieve these objectives, then, our paper is organized as follows: first, we will present a review of the relevant literature on signal control and methodologies used in our study; then, we will outline the structure of the proposed DSC system; in addition, we will formulate traffic dynamics as a Queueing Network Model for this study; we will continue by presenting numerical examples to compare cases of CSC and DSC systems; our study concludes with a discussion of future research avenues.

LITERATURE REVIEW

Since Webster's fundamental work from 1958 on traffic signal control (*3*), the need for modeling large-scale networks and their traffic signal control has grown and motivated many researchers to develop innovative control strategies, related models, and problem-solving algorithms. Since the focus of this paper is to develop a decentralized traffic signal control system via MEC, we review two areas of the research literature to address the different components required for this research: network-level traffic signal control and Edge Computing (EC).

Network-Level Traffic Signal Control

Signal control is widely used at different levels across urban transportation networks, including at the levels of the intersection, corridor, and network itself. Here, we focus on network-level signal control (NSC) research, which we have divided into two parts: CSC and DSC (or sometimes called distributed signal control).

CSC

Centralized signal control optimizes the parameters of each intersection in the network simultaneously to find the optimal solution, and control strategies have been widely studied in recent decades. Much of the research has focused on the kinds of algorithms used in different networks with different targets. For instance, Genetic Algorithms (GA) and Approximate Dynamic Programming approaches have been proposed for traffic signal control in oversaturated networks (*4*). Another algorithmic solution for oversaturated networks has been the ant colony optimization algorithm (ACO) (*5*). Other researchers have used heuristic algorithms to solve various problems. For example, Beard et al. used a mixed integer linear programming model to optimize traffic signal control (*6*). And He et al. considered a multi-modal priority control in a dynamic network (*7*).

For all of these cases, a small network is the optimal scenario for CSC. Even escalating to a medium-scale network means that the complexity (NP-complete) and increased time and geometric scale cause greater difficulties (*2*). Although recent advancements in computer science offer some scope to solve real-time, large-scale data collection and processing, the NP-complete problem, especially the real-time NP-complete problem such as network-wide real-time signal control, remains unresolved.

DSC

Distributed signal control is referred to as decentralized control. Prompted by the difficulties of CSC as outlined above, research has turned to distributed systems that decompose large networks into smaller subnetworks and optimize the parameters of all the intersections in the subnetwork simultaneously and cooperatively. Gokulan et al. developed a distributed, multiagent-based approach for a traffic-responsive signal control system (*8*). The distributed signal control was achieved by multi-agents at each subnetwork. This method significantly reduced the complexity issue of NSC.

The measured targets for DSC vary in the literature, with increasing throughput of the whole network as one common one (*9*). Many studies aiming for maximizing throughput of the whole network have used the Backpressure routing algorithm (*10, 11*). In addition, some researchers have investigated the degree of impact that distributed signal control can have on reducing TTT and increasing average speed, with varying results (*12*, *13*). Focusing on the target of reducing network traffic GHG emissions, an intersecting study tested the impact made by the optimization of traffic signal lights (*14*).

For application needs, the topology of network decomposition is a key point. Cell-based decomposition (*12*) and intersection-based decomposition (*15*, *2*) are two main methods used for spatial decomposition, but this means that the subnetwork contains only one intersection. Adacher and Tiriolo have addressed this by working with subnetworks containing more than one intersection (*16*). However, they tested one network that includes a specific group of subnetworks with a simplified signal phase, thereby offering restricted results. In other research, a grouping method has been developed to decrease delay and the number of stops with better bandwidths. Nonetheless it has only been implemented in a freeway network (*17*). The question of how to optimize the network decomposition is a significant gap in the literature and remains to be solved. We focus our study here towards addressing this gap.

Edge Computing

Traditional vehicle-to-infrastructure (V2I) communication describes the way in which vehicles receive data from the TMC. Currently, the vehicle is not only the edge of data reception but also the edge of data collection. Thus, the bandwidth of the internet network creates a bottleneck for cloud computing. This points to the need for EC whereby data processing can occur partly at the edge unit rather than completely in the cloud. The Internet of Things (IoT), first introduced in 1999 (*18*), has used EC widely, and all indicators points to its part in the future transportation. In addition to EC, Mobile Edge Computing/Multi-access Edge Computing (MEC) is an extended term for EC, especially for cellular networks to serve the mobile users within their access range (*19*). Moreover, some researchers have already introduced a deep learning method for the Intelligent Transportation System (ITS) based on MEC analysis *(30)*. At last, the inherent features of 5G will make MEC a key technology in the near future *(31)*. EC technology will help the development of modern transportation soon as the development of communication technology.

Although there have been many studies carried out on the CSC system, the literature points to the DSC system possessing greater advantage and applicability for use in a network-level traffic control system. These studies, however, have not provided adequate definition to optimize network decomposition for a DSC system. We have identified the need for further investigation and our study aims to address this gap.

In addition, we draw upon studies that demonstrate the advantages of EC to reduce latency and provide unlimited scalability when compared with traditional cloud computing. This is especially so for the decentralized system of a large-scale traffic network that includes EC units at intersections and on vehicles.

DECENTRALIZED SIGNAL CONTROL (DSC) SYSTEM

We will introduce a DSC system that includes MEC and provide the definitions and assumptions that form the basis of this study. First, we will discuss the data flow of CSC and DSC. Subsequently, the basic framework of DSC with two layers consisting of three components, namely route guidance (1st layer), traffic dynamics (2nd layer), and signal optimization (2nd layer), will be introduced.

System Setup

Assumptions and Definitions

For the network-wide signal control problem, we must first define Signal User Optimal (Signal-UO) and Signal System Optimal (Signal-SO). Unlike traditional definitions of 'User Optimal' and 'System Optimal' in traffic assignment, our use of the term 'users' of the network system refers not to the traveler but instead to the intersection as the variables in this problem are signal control parameters.

The definitions for Signal-UO and Signal-SO are articulated as follows:

Definition 1 User Optimum for Network-Wide Signal Control (Signal-UO)
The "user" for the network signal control system is referred to as the signal control of each individual intersection. The Signal-UO assumption is that the signal control is designed to achieve the maximum throughput for its own intersection regardless of the overall performance of the whole traffic network. The assumption implies that when signal controls for all other intersections are fixed, the signal control for the current intersection will be designed to achieve the maximum throughput under current traffic demand inputs.

Definition 2 System Optimum for Network-Wide Signal Control (Signal-SO)
The assumed "user" for the network signal control system is the signal control of each individual intersection. The Signal-SO case is that signal controls of intersections in the network are designed to achieve the maximum throughput of the whole traffic network.

The throughput of an intersection or network is calculated according to the final optimization objectives, such as Total Travel Time (TTT), Total Travel Emission (TTE), Total Travel Delay (TTD), or Average Speed (AS). In this work, we consider TTT as the optimization objective.

The control variables of each intersections are assumed to be phase plan, phase split and offset. **Figure 2** shows an example of signal control for a 4-leg intersection. This phase plan is used for the simulation as well.

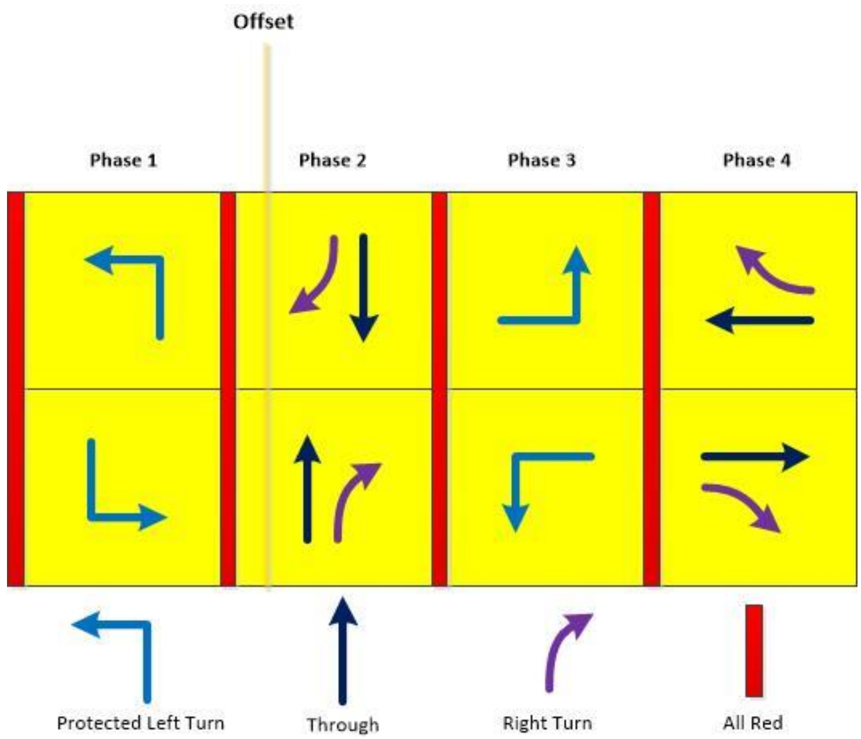


Figure 2 A 4-phase phase plan

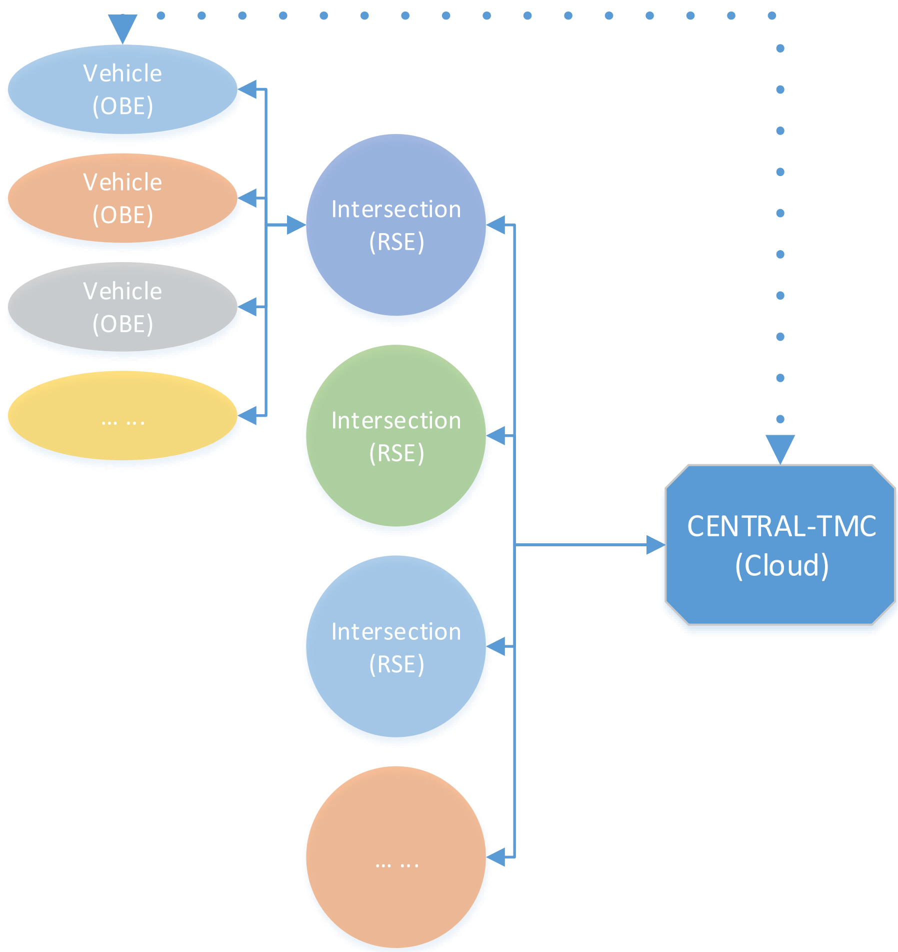
Phase 1 is green for traffic turning left from northbound and southbound, phase 2 is green for traffic going through and right from northbound and southbound, and phases 3 and 4 repeat phases 1 and 2 for the eastbound and westbound directions. The red bars represent all-red intervals. Offset is time relationship between coordinated phases defined reference point and a defined master reference.

We assume that all the intersections use this 4-phase phase plan and have the same cycle length, independent phase splits, and independent offsets. So, the control variables of the whole system are cycle length, green times for phases at each intersection signal, and offsets for all the intersections.

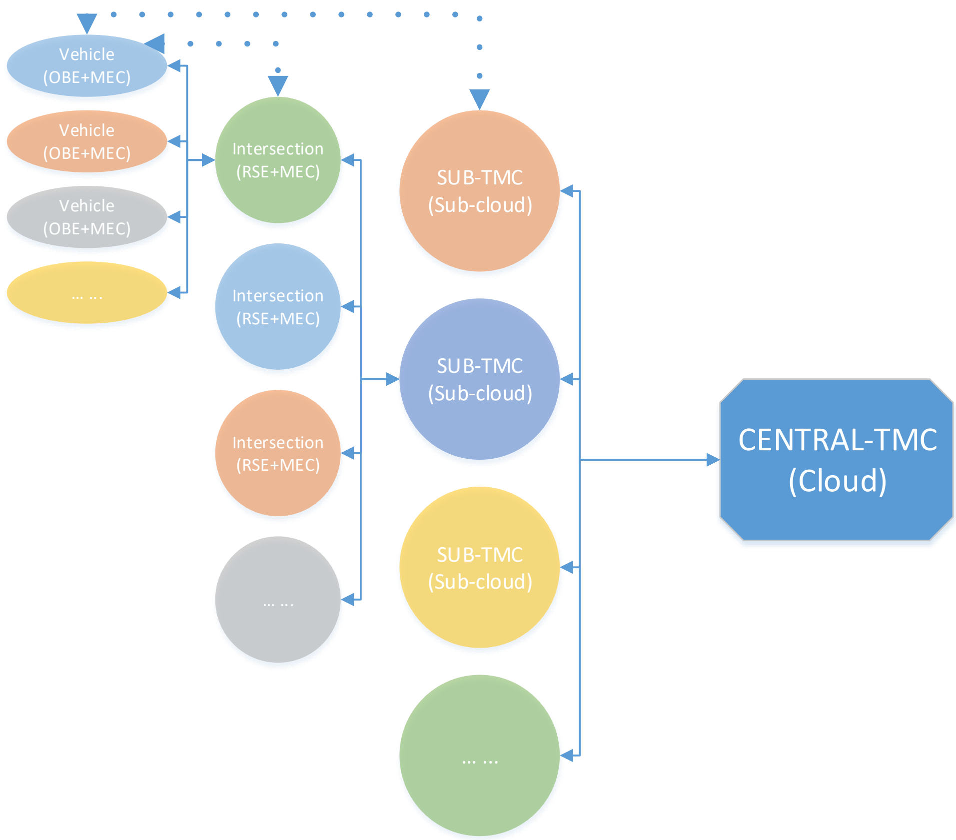
Thus, the Signal-SO problem for a large-scale network involves large numbers of control variables, which is an NP-complete problem. Note that the control variables used in our study are all discrete variables. If we were to use continuous variables, the Signal-SO problem may be an NP-hard problem.

Data Flow

We now compare the data flow of CSC and DSC. **Figure 3(a)** shows the data flow of the centralized signal control system (CSC), and **Figure 3(b)** shows the data flow of the decentralized signal control system (DSC).



0(a) Centralized Signal Control



(b) Decentralized Signal Control

Figure 3 Data flow comparison between CSC and DSC

In the CSC system (**Figure 3(a)**), each vehicle has on-board equipment (OBE) to communicate with each intersection as well as the Central-TMC cloud. There is RSE at each intersection to facilitate communicate between OBE and Central-TMC, and use this information to control the signal. There is a central cloud server for the Central-TMC to communicate with OBE and RSE, store and analyze data, and optimize the traffic signal control.

OBE units collect the basic safety message (BSM) data from vehicles. BSM data include real-time speed and real-time location information and are sent to the RSE at each intersection by the OBE units. The RSE units will send signal timing data (phase sequence, phase split, and phase offset) together with BSM data to the Central-TMC. According to the data received, the Central-TMC then analyses and optimizes for the signals. Finally, the Central-TMC sends the updated signal strategy to the RSE at each intersection. The RSE then changes the signal plan for its particular intersection and controls the traffic according to the updated signal strategy. In addition, RSE sends the signal information to OBE units of nearby vehicles.

In the DSC system (**Figure 3(b)**), each vehicle and intersection carry an additional MEC unit, unlike in the CSC system. MEC units can receive and deliver more types of data than the OBE units or the RSE. In addition, MEC units have the computation ability to analyze and calculate in situ. This additional capability facilitates the DSC system to divide the network into subnetworks, unlike the CSC system. Each DSC subnetwork has an "agent", also called a Sub-TMC, that carries a small cloud computing server able to connect to the central cloud server of Central-TMC as well as to the RSE and OBE inside the subnetwork.

The vehicle-based MEC collects and sends time-varying OD information to the sub-cloud of Sub-TMC. Also, the vehicle's OBE sends BSM data to the RSE, which, in turn, the RSE sends to Sub-TMC. The Sub-TMC then analyzes the local data and sends it to the Central-TMC. The Central-TMC assigns the time-varying demand to paths, which is the route guidance information. This information is delivered to the Sub-TMC and then to each vehicle. Meanwhile, the Sub-TMCs estimate the traffic dynamics for the whole network in parallel. Finally, each Sub-TMC obtains the time-varying OD demand for each subnetwork and optimizes the signals inside the subnetwork, respectively.

Efficiency

To compare the efficiency of these systems, we first present definitions for computation efficiency and traffic efficiency for the network-wide signal control system.

Definition 3 *Computation Efficiency*
Computation efficiency is scaled by the computation time cost of one iteration for the signal control system. The lower the computation time cost, the more efficient it is considered to be.

Definition 4 *Traffic Efficiency*
Traffic Efficiency is scaled by the optimization objective of the network control system. For example, if the objective is to minimize the total travel time (TTT) of the network, then the lower the TTT, the more efficient it is considered to be.

If the CSC system is targeted to solve the Signal-SO problem for traffic efficiency, the computation time cost will increase exponentially when the size of the network increases since the problem is an NP-complete problem. If the size of subnetwork does not change when we expand the whole network, the computation efficiency will change little. Thus, when comparing computation efficiency, the DSC system will produce better results than the CSC system, especially for large-scale networks. When we address a real-control case, the computation efficiency must meet this requirement. The DSC system can then be judged on its ability to supersede the CSC system.

As we know, the Signal-SO case will beat the Signal-UO case for traffic efficiency. Certainly, the CSC system can achieve the Signal-SO for the whole network without consideration of computation efficiency, since the DSC system is approaching the Signal-SO by adjusting the size of subnetwork. However, let us suppose that each subnetwork has a Signal-SO solution inside each one. When each subnetwork remains as an individual intersection, it is the Signal-UO case. When the subnetwork is the whole network itself, it is the Signal-SO case. In conclusion, the traffic efficiency of the CSC system is the same as Signal-SO, and the traffic efficiency of DSC is in the middle of Signal-SO and Signal-UO.

For a network-wide signal control problem, especially for a large-scale urban network and a real-time control case, and since the Signal-SO is an NP-complete problem, the DSC system is logically the better choice based on computation efficiency. What may be crucial for DSC system is to balance computation efficiency and traffic efficiency via network decomposition.

Basic Framework of DSC

Figure 4 illustrates the proposed basic framework for our DSC system. There are two distinct layers. For the first layer in the origin box, the input is the short-term OD matrix, also regarded as time-varying OD information. The Central-TMC will gather the short-term OD information from the Sub-TMCs and MEC units on-board the vehicles. The sole function of the Central-TMC is route guidance, although this function may be decentralized as well. The output is the short-term demand for each path, which is the input for the second layer. The second layer will have two functions: traffic dynamics (traffic state estimation) on Sub-TMCs and signal optimization on MEC units at the intersections inside each subnetwork. Details about these three functions are introduced in the next section.

The calculation for the Central-TMC is route guidance only. The remaining optimization tasks are allocated to the Sub-TMCs inside each subnetwork. In addition, the MEC units at each intersection are involved in the computation of the signal optimization function. As a result, all the computation resources of the DSC system are used fully.

After each subnetwork completes optimization, the signals' timing for intersections of the whole network will be adjusted as well. This has an impact on traffic dynamics. So, the signal control profile for all the intersections will be updated in the next iteration. The Sub-TMC will continue to optimize the signals with each new set of inputs.

The stop condition is when the difference of TTT between the current signal plan and the previous signal plan is under a specific threshold. The decentralized system will give the optimized signal timing plan for the whole network after the entire process.

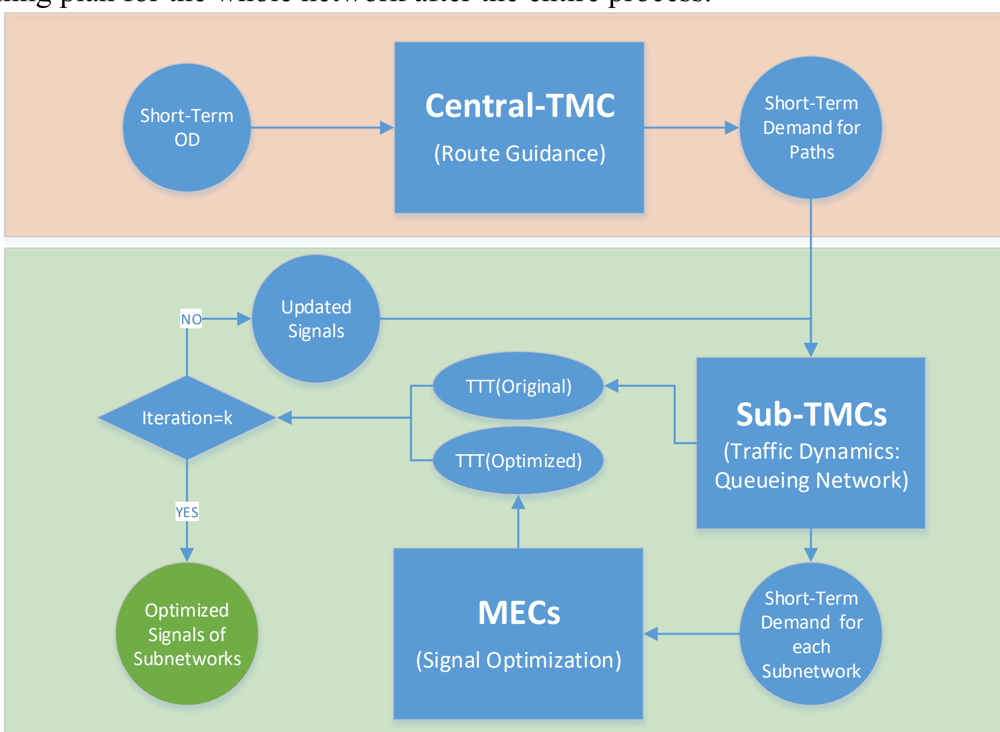


Figure 4 Two-layer process for decentralized signal control system

NUMERICAL SIMULATION

This section introduces the three functions of the DSC system in detail. The route guidance is based on the '1-0' principle, the traffic dynamics are formulated as a $G/G/n/FIFO$ queueing network, and signal optimization is a naïve 3-step method. Our case study, used to illustrate these three functions, will be based on several network decompositions of square networks. We go on to compare results from the CSC and DSC systems across several of the same networks, each with different network decompositions.

Route Guidance

Route guidance is the first function in the process depicted in **Figure 4**. First, we define path $p \in P$ as a vector of the path id and all the orderly nodes in the path as $p = (k, o, n_1, \dots, n_l, d)$, where k is the id of the path, $o \in O$ is origin node (O is set of origin nodes), $d \in D$ is destination node (D is set of destination nodes. The link is a directed vector as ij, which starts at node i, ends at node j.

The expected travel cost for all the links is a matrix as $\{T_{ij}(t)\}$, where ij is the link, t is an integer index for time stamp. Note $T_{ij} = \begin{cases} 0 & i = j \\ \infty & ij\ DNE \end{cases}$. There are two types of links: the intersection link and the general link. In **Figure 5**, the intersection links are marked in yellow and red, and the general links are marked in green. T_{ij} for the intersection link is the signal waiting time and turning time, and the general link is the free flow travel time (FFT).

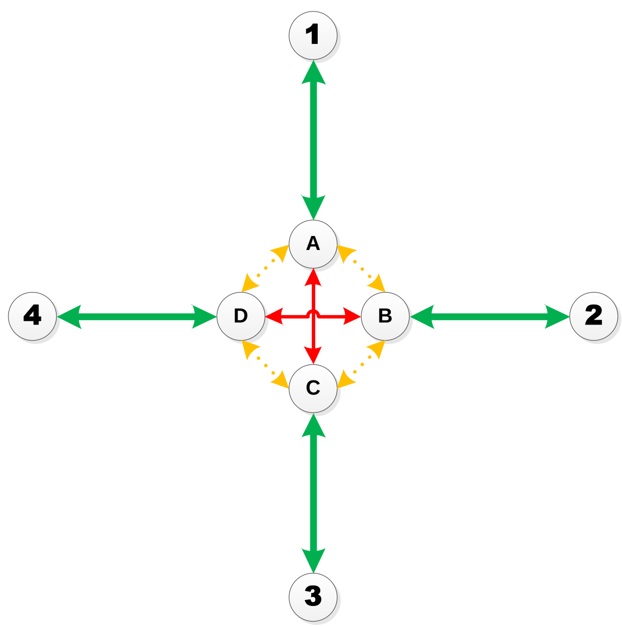


Figure 5 Sample of an intersection

The input of this function is short-OD demand formulated as $d(o, d, t)$, where $o \in O$ is the origin node, $d \in D$ is destination node, t ($0 < t < T \in Z$) is an integer index for time stamp.

The output of this function is short-term demand for each path formulated as $d_{path}(1, k_p, t)$, where $k_p = p(1)$ is the path id for path p. The first entry 1 of the path demand function means demand for the first node of path p.

Route guidance is following the '1-0' principle according to the expected link travel cost matrix $\{T_{ij}\}$.

Assumption 1 '1-0' Principle
To simulate travel behavior, we assume all travelers for the same time stamp t and same origin and destination will be assigned to the path with the least expected instantaneous travel time before they enter the whole network.

The expected instantaneous travel time of path $T_{path}(k_p, t)$ is calculated as Equation (1) as the sum of expected link travel cost on the path.

$$T_{path}(k_p, t) = \sum_{ij \in p} T_{ij}(t) \quad (1)$$

The route guidance will simply use Dijkstra's Shortest Path First algorithm (SPF algorithm) that is designed to find the shortest paths for all OD pairs. The weight of each link is $T_{ij}(t)$.

Another important component for DSC is network decomposition. The path through the entire network is decomposed into paths for each subnetwork. Sub-TMCs in subnetworks optimize traffic according to the short-term demand for the sub-path provided by the next function as Traffic Dynamics.

Figure 6 shows an example of a 6×6 network decomposed into nine 2×2 subnetworks. The path shown in the figure from the left top node to the second right top node is divided into three sub-paths in three subnetworks demarcated as green, red, and yellow, respectively. The decomposition is based on the second assumption:

Assumption 2 No Rerouting
Travelers will follow the route guidance until they exit the network. They will not reroute during their trip inside the network.

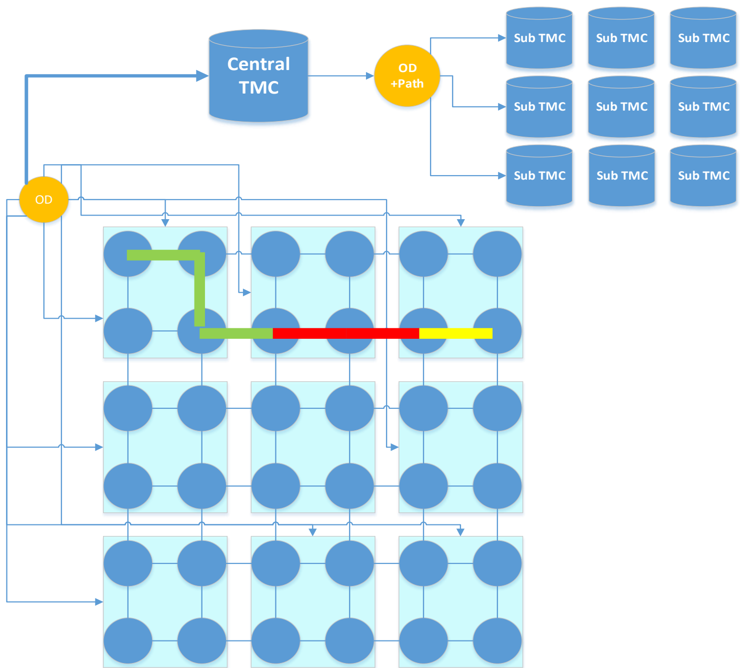


Figure 6 An example of network decomposition

Traffic Dynamics

The second function is 'Traffic Dynamics' (**Figure 4**). This function is used to estimate the time-varying demand for each link in subnetworks. The queueing rule of the network is a $G/G/n/FIFO$. The first and second G signifies that the arrival rate and the service rate for queue follows general distribution; n is the number of queues inside the system. There are two assumptions for the traffic dynamics as outlined below:

Assumption 3 Link Queue
It is assumed that the upstream lane is the queue for each intersection link. The queue has infinite storage room for vehicles. There is a constant discharging rate at the end of the queue marked as w.

Assumption 4 First In First Out (FIFO)
The first vehicle that enters the link will also be the first to leave the link.

Both assumptions are mentioned in Ran et al. (*22*). The assumptions will generate the equations of traffic dynamics in this part with the Queueing Network Model (QNM).

The input of the 'Traffic Dynamics' function is a time-varying demand for the first link on all paths of the whole network as $d_{path}(1, k_p, t), \forall p \in P, \forall t \leq T$. T is the length of time for the time-varying demand function.

The output of the 'Traffic Dynamics' function is a time-varying demand for any link on any path of the whole network as $d_{path}(i, k_p, t), \forall i \leq l(p), \forall p \in P, \forall t \leq T$, where $l(p)$ is the number of links on path p.

Figure 7 shows a general link in the simulation of this work. It has three lanes for left, right, and through movements each.

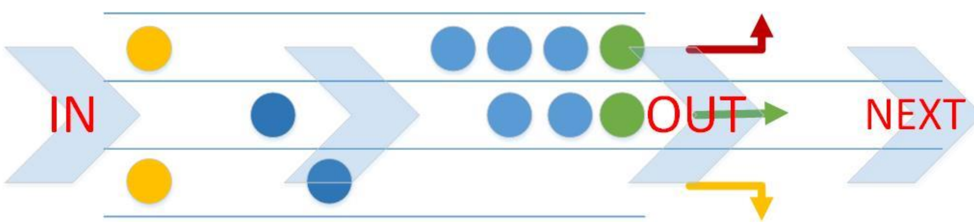


Figure 7 Example of a general link

Suppose the start point and the end point is called node n_{in} and node n_{out} for demand function $d_{path}(i, k_p, t)$. Since the first entry of the path vector represents the path id of path p, if the link on path p, then $n_{in} = p(i+1)$, then $n_{out} = p(i+2)$ and call $n_{next} = p(i+3)$. Then, link $n_{out}n_{next}$ is an intersection link. We introduced the following notation for traffic signal characteristics:

(1) Cycle length: $T_{cl}(n_{out}n_{next})$ for intersection link $n_{out}n_{next}$;
(2) Queue length: $L_{queue}(n_{out}n_{next}, t)$ for intersection link $n_{out}n_{next}$ at time stamp t;
(3) Total green time in a cycle: $L_{green}(n_{out}n_{next})$ for intersection link $n_{out}n_{next}$;
(4) Signal red time remaining: $T_{red}(n_{out}n_{next}, t)$ for intersection link $n_{out}n_{next}$ at time stamp t;
(5) Signal green time remaining: $T_{green}(n_{out}n_{next}, t)$ for intersection link $n_{out}n_{next}$ at time stamp t.

The queueing time is a combination of link travel time $T_{n_{in}n_{out}}(t)$, experienced waiting time inside the queue, and experienced waiting time for signals.

In addition, we assume the phase plan for the control period $[0, T]$ is fixed. It can be changed for the next control period.

The algorithm for generating traffic dynamics is described as follows:

Step 0 (Initialization): $t = 0, i = 1$, input is demand profile for all the path
$d_{path}(1, k_p, \tau), \forall p \in P, \forall \tau \leq T$ during period T;
Step 1 (Demand Update): $\forall p \in P$,

$$n_{in} = p(i+1), n_{out} = p(i+2), n_{next} = p(i+3); \quad (2)$$

if $d_{path}(i, k_p, t) = \alpha > 0$, then

$$L_{queue}(n_{out}n_{next}, t+1) = L_{queue}(n_{out}n_{next}, t) + \alpha_p; \quad (3)$$

Let link travel time $T_{n_{in}n_{out}}(t) = \beta$; (4)

Total queueing time

When signal is green, $T_{green}(n_{out}n_{next}, t+\beta) > 0$

$$\Delta t = \beta + \left\lceil \frac{L_{queue}(n_{out}n_{next}, t+\beta) - w \cdot T_{green}(n_{out}n_{next}, t+\beta)}{w \cdot L_{green}(n_{out}n_{next})} \right\rceil \times T_{cl}(n_{out}n_{next}) + mod\left(L_{queue}(n_{out}n_{next}, t+\beta) - 1, L_{green}(n_{out}n_{next})\right) + 1 \quad (5)$$

When signal is red, $T_{green}(n_{out}n_{next}, t+\beta) = 0$

$$\Delta t = \beta + \left\lceil \frac{L_{queue}(n_{out}n_{next}, t+\beta)}{w \cdot L_{green}(n_{out}n_{next})} - 1 \right\rceil \times T_{cl}(n_{out}n_{next}) + mod\left(L_{queue}(n_{out}n_{next}, t+\beta) - 1, L_{green}(n_{out}n_{next})\right) + 1 + T_{red}(n_{out}n_{next}, t+\beta) \quad (6)$$

$$d_{path}(i+1, k_p, t+\Delta t) = d_{path}(i+1, k_p, t+\Delta t) + \alpha_p \quad (7)$$

Step 2 (Queue Length Update):

After demand for all the paths has been assigned, update queue length

$$L_{queue}(n_{out}n_{next}, t+1) = \begin{cases} L_{queue}(n_{out}n_{next}, t) \\ L_{queue}(n_{out}n_{next}, t) - \alpha_p \end{cases} \quad (8)$$

If $i < l(p) - 1$, i=i+1, go to Step 1; otherwise, i=1, go to step 3;

Step 3 (Stop Condition):

If $t < T$, $t = t + 1$, go to Step 1; otherwise, Stop.

When the demand for any link on any path of the whole network has been estimated. The assignment of demand to each sub-TMC is achieved by introducing path decomposition.

The path decomposition is achieved by another relationship index $\Gamma_s(k_p, k_{p_s})$, where s is the id for the subnetwork, k_s is path id for path p of the whole network. k_{p_s} is the path id for path p_s for subnetwork s. $\Gamma_s(k_p, k_{p_s}) = m$ if the m^{th} node of path p is the first node of path p_s for subnetwork s. For example, suppose the id for the middle subnetwork of the top 3 (in **Figure 6**) is 2, the depicted path p, $\Gamma_2(k_p, k_{p_2}) = 4$.

$d^s_{path}(i, k_{p_s}, t)$ is defined as the time-varying demand function for subnetwork s.

Suppose $\Gamma_s(k_p, k_{p_s}) = m$, then

$$d^s_{path}(1, k_{p_s}, t) = d_{path}(m, k_p, t) \quad (9)$$

For each subnetwork, the Sub-TMC can estimate the dynamics if it changes the signal control via Equations (2)-(8).

Centralized Signal Control (CSC)

The target of CSC is to solve the Signal-SO problem. We have mentioned that for the DSC system, when the subnetwork is the entire network itself, it is exactly the Signal-SO case. The CSC can be regarded as a special DSC while there is only one subnetwork for the decomposition.

On the other hand, the Sub-TMC is assumed to function as a small CSC system that aims to achieve Signal-SO for the subnetwork within DSC and each of its subnetworks.

The algorithm for optimization within the CSC system is a 3-step naïve method outlined as below:

Step 0: Choose an optimization objective. For our simulation, we have chosen Average Total Travel Time (Average TTT);

Step 1: Optimal cycle length: fix phase splits and offsets for all the intersections. Find the optimal cycle length according to the optimization objective.

Step 2: Optimal phase split: based on the optimal cycle length at the first stage, fix the offset and generate all the combinations of phase split. In the end, find the optimal phase split according to the optimization objective.

Step 3: Optimal offset: based on the first two stages, generate all the combinations of offsets for all the intersections. Find the optimal offsets plan according to the optimization objective.

Case Study

Restricted to the lab environment, all simulations were run in MATLAB, using the Compute Canada's Westgrid supercomputer. The server we used has a maximum core of 40.

The benchmarks used for computation efficiency and traffic efficiency are computation time cost (CTC) and average TTT for the CSC system.

In our case study, simulations are made for the functions 'Traffic Dynamics' and 'Signal Optimization' in **Figure 4**:, since the function of 'Route Guidance' is completed at the Central-TMC. The main contribution of this work is focused on the local computation. Hence, the inputs are short-term demand for paths. For all cases, we set the demand profile at 1 per 40 time units. The length of the demand profile is 60 time units.

For the CSC cases, we tested 1×1, 2×2, and 3×3 networks, and the target is Signal-SO. In addition, to function as a Central Cloud, the number of cores used in the simulation is 40.For DSC cases, there are two categories: the first is called DSC-adaptive that decomposes 1×1, 2×2, 3×3, 4×4, 6×6 into 1×1 subnetworks; the second is called DSC-nonadaptive that only decomposes a 4×4 network into four 2×2 subnetworks. We assume that each intersection has a computing unit that can calculate with 2 cores.

The free flow travel time for each link is 10 time units. Turning times for the left turn lane, right turn lane, and through lane are 3, 2, and 1 time units respectively. The 1×1 network is the same as in **Figure 5**. The 6×6 network is like in **Figure 6**. The signal control uses a four-phase plan as in **Figure 2**. All intersections have the same cycle length. The phase split and offset for each intersection may be different.

Simulation Results

Results are shown in **Table 1**.

Table 1 Computation Efficiency and Traffic Efficiency of different CSC and DSC cases

(a) CSC cases

Network Size	$\mathbf{1 \times 1}$	$\mathbf{2 \times 2}$	$\mathbf{3 \times 3}$
Average TTT (in time units)	39.75	69.05	103.9
CTC (in seconds)	0.23	14.82	5780.4
# of cores	40	40	40

(b) DSC-adaptive cases

Network Size	$\mathbf{1 \times 1}$	$\mathbf{2 \times 2}$	$\mathbf{3 \times 3}$	$\mathbf{4 \times 4}$	$\mathbf{6 \times 6}$
Subnetwork Size	1×1	1×1	1×1	1×1	1×1
Average TTT (in time units)	39.75	69.25	104.81	142.82	258.51
% of CSC	100.00	100.29	100.86	N/A	N/A
CTC (in seconds)	0.7	8.1	17.9	48.25	96.2
% of CSC	304.34	54.66	0.31	N/A	N/A
# of cores	2	4	18	32	40

(c) DSC-nonadaptive case

Network Size	$\mathbf{4 \times 4}$
Subnetwork Size	2×2
Average TTT (in time units)	141.78
% over DSC-adaptive	99.27
CTC (in seconds)	176.21
% over DSC-adaptive	365.20
# of cores	16

When we compare the results:

- DSC-adaptive emerges as the most computationally efficient. The computation time cost for CSC increases exponentially, thereby placing it at the bottom for efficiency; DSC-nonadaptive has larger decomposition ranking it in the middle of the other two.
- CSC ranks the highest of the three for traffic efficiency as Signal-SO; DSC-adaptive ranks the worst as Signal-UO. DSC-nonadaptive is in the middle of Signal-SO and Signal-UO.

Furthermore, the CTC of the CSC increases exponentially with network size, with the 3×3 network taking over 1.5 hours to solve. This also provides evidence for the fact that the optimization for CSC is an NP-complete problem. The CTC in DSC-adaptive cases is much smaller than that of DSC, but the increase in speed is still sufficient.

When comparing the average TTT between the CSC and DSC cases, CSC performs slightly better than the DSC-adaptive. In turn, the DSC-nonadaptive is slightly better than DSC-adaptive. The average TTT of DSC-nonadaptive is 99.27% over that of DSC-adaptive. It seems that larger subnetworks for network decomposition perform better than smaller subnetworks in terms of traffic efficiency. This is due more intersections are connected for the regional Signal-SO in larger subnetwork. However, the increase in average TTT is quite small and rather insignificant compared with the degradation on CTC.

In summary, DSC system is a better choice for application of network-wide signal control on computation efficiency. In addition to ensuring that the computation efficiency meets the latency requirement of real-time control, a secondary objective of optimizing network decomposition is to maximize traffic efficiency.

CONCLUSION

The development and deployment of CV technologies promise improvements in both data quantity and quality. In addition, MEC technology helps significantly to reduce latency for real-time control by handling data at the point where it is generated. However, network-level traffic signal optimization, which is a Nondeterministic Polynomial Time (NP)-complete problem, remains unsolved with respect to optimizing both the objectives and computation resources. The most recommended method applies a decentralized control system instead of the centralized one. A decentralized traffic signal control system is one creative solution that decomposes a large-scale network into subnetworks, with each subnetwork controlled by its own agent. Agents can exchange information in real-time. However, in the literature, there is lack of attention paid to optimal network decomposition (i.e., number and size of subnetworks), which is important for practical use. This paper has developed a multi-agent decentralized signal control system and compared different cases for both centralized signal control system and decentralized signal control system.

To do so, this paper defined Signal-UO and Signal-SO for the network signal control system, with UO and SO defined as cases of network traffic assignment. For the signal control system, we have considered each intersection as an individual user in the signal control system. The control variables for each intersection is cycle length, phase split, and offset. Signal-UO requires each intersection to maximize throughput of its own. Signal-SO entails all the intersections to work together to maximize the throughput of the whole network.

In addition, to evaluate specific signal control systems, we used two measurements: computation efficiency and traffic efficiency. Computation efficiency was measured by the CTC and traffic efficiency is measured by the overall performance such as TTT.

Our simulation was based on a G/G/n/FIFO QNM. Results showed that DSC systems have much better performance on computation efficiency, while their performance on traffic efficiency is weaker. Network decomposition also has an impact on both types of efficiency. Increases in the size of subnetwork may gain traffic efficiency and loss of computation efficiency. In addition to ensuring that the computational efficiency meets the latency requirement, another objective of network decomposition optimization is maximum traffic efficiency.

Since the algorithms used in routing guidance and signal optimization are simple and time-consuming in the CSC system. Future work will require these two elements, and more cases should be investigated.

ACKNOWLEDGEMENTS

This research work was jointly supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, City of Edmonton, Transport Canada, TELUS, and Compute Canada. The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the City of Edmonton or Transport Canada. This paper does not constitute a standard, specification, or regulation.

AUTHOR CONTRIBUTION STATEMENT

The authors confirm contribution to the paper as follows: study conception and design: Can Zhang, Amy Kim, and Zhijun Qiu; data collection: Can Zhang; analysis and interpretation of results: Can Zhang; draft manuscript preparation: Can Zhang, Amy Kim. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Perry Sr, F. UFTI DSRC and Other Communication Options for Transportation Connectivity Workshop Overview of DSRC Messages and Performance Requirements. 2017.

2. Adacher, L., A. Gemma, and G. Oliva. Decentralized Spatial Decomposition for Traffic Signal Synchronization. *Transportation Research Procedia*, Vol. 3, No. July, 2014, pp. 992–1001. https://doi.org/10.1016/j.trpro.2014.10.079.

3. Webster, F. V. *Traffic Signal Settings*. 1958.

4. Hajbabaie, A. Intelligent Dynamic Signal Timing Optimization Program. *ProQuest Dissertations and Theses*, 2012, p. 216.

5. Putha, R., L. Quadrifoglio, and E. Zechman. Comparing Ant Colony Optimization and Genetic Algorithm Approaches for Solving Traffic Signal Coordination under Oversaturation Conditions. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 27, No. 1, 2012, pp. 14–28. https://doi.org/10.1111/j.1467-8667.2010.00715.x.

6. Beard, C., and A. Ziliaskopoulos. System Optimal Signal Optimization Formulation. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1978, 2007, pp. 102–112. https://doi.org/10.3141/1978-15.

7. He, Q., K. L. Head, and J. Ding. Multi-Modal Traffic Signal Control with Priority, Signal Actuation and Coordination. *Transportation Research Part C: Emerging Technologies*, Vol. 46, 2014, pp. 65–82. https://doi.org/10.1016/j.trc.2014.05.001.

8. Gokulan, B. P., and D. Srinivasan. Distributed Geometric Fuzzy Multiagent Urban Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 3, 2010, pp. 714–727. https://doi.org/10.1109/TITS.2010.2050688.

9. Wongpiromsarn, T., T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang. Distributed Traffic Signal Control for Maximum Network Throughput. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2012, pp. 588–595. https://doi.org/10.1109/ITSC.2012.6338817.

10. Tassiulas, L., and A. Ephremides. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. No. 1, 1990, pp. 2130–2132.

11. Le, T., P. Kovács, N. Walton, H. L. Vu, L. L. H. Andrew, and S. S. P. Hoogendoorn. Decentralized Signal Control for Urban Road Networks. *Transportation Research Part C: Emerging Technologies*, Vol. 58, 2015, pp. 431–450. https://doi.org/10.1016/j.trc.2014.11.009.

12. Mehrabipour, M., and A. Hajbabaie. A Cell-Based Distributed-Coordinated Approach for Network-Level Signal Timing Optimization. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 32, No. 7, 2017, pp. 599–616. https://doi.org/10.1111/mice.12272.

13. Islam, S. M. A. B. Al, and A. Hajbabaie. Distributed Coordinated Signal Timing Optimization in Connected Transportation Networks. *Transportation Research Part C: Emerging Technologies*, Vol. 80, 2017, pp. 272–285. https://doi.org/10.1016/j.trc.2017.04.017.

14. Ferreira, M., and P. M. D'Orey. On the Impact of Virtual Traffic Lights on Carbon Emissions Mitigation. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 1, 2012, pp. 284–295. https://doi.org/10.1109/TITS.2011.2169791.

15. Withanawasam, J., and A. Karunananda. Multi-Agent Based Road Traffic Control Optimization. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Vol. 2018–March, 2018, pp. 977–981. https://doi.org/10.1109/ITSC.2017.8317896.

16. Adacher, L., and M. Tiriolo. Distributed Urban Traffic Signal Optimization Based on Macroscopic Model. *2016 6th International Conference on Innovative Computing Technology, INTECH 2016*, 2017, pp. 605–610. https://doi.org/10.1109/INTECH.2016.7845123.

17. Zhang, L., M. Lee, H. Chang, and C. Carrion. An Approach to Grouping Traffic Signals for Coordination Using Clustering Methods. *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings*, 2017, pp. 792–797. https://doi.org/10.1109/MTITS.2017.8005620.

18. Shi, W., and S. Dustdar. The Promise of Edge Computing. *Computer*, Vol. 49, No. 5, 2016, pp. 78–81. https://doi.org/10.1109/MC.2016.145.

19. Jararweh, Y., A. Doulat, O. Alqudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa. The Future of Mobile Cloud Computing: Integrating Cloudlets and Mobile Edge Computing. *2016 23rd International Conference on Telecommunications, ICT 2016*, 2016, pp. 1–5. https://doi.org/10.1109/ICT.2016.7500486.

20. Ferdowsi, A., U. Challita, and W. Saad. Deep Learning for Reliable Mobile Edge Analytics in Intelligent Transportation Systems. 2017.

21. Hu, Y. C., M. Patel, D. Sabella, N. Sprecher, and V. Young. Edge Computing: A Key Technology towards 5G. *ETSI White Paper No. 11 Mobile*, No. 11, 2015, pp. 1–16.

22. Ran, B., D. E. Boyce, and L. J. LeBlanc. A New Class of Instantaneous Dynamic User-Optimal Traffic Assignment Models. *Operations Research*, Vol. 41, No. 1, 1993, pp. 192–202. https://doi.org/10.1287/opre.41.1.192.