2019

Smart Fog Based Workflow for Traffic Control Networks

Qiang Wu
Lanzhou University

Jun Shen
University of Wollongong, jshen@uow.edu.au

Binbin Yong
Lanzhou University, yongbb14@lzu.edu.cn

Jianqing Wu
University of Wollongong, jw937@uowmail.edu.au

Fucun Li
University of Wollongong, fl626@uowmail.edu.au

See next page for additional authors

Abstract

In this paper, we propose a novel traffic control architecture which is based on fog computing paradigm and reinforcement leaning technologies. We firstly provide an overview of this framework and detail the components and workflows designed to relieve traffic congestion. These workflows, which are connecting traffic lights, vehicles, Fog nodes and traffic cloud, aim to generate traffic light control flow and communication flow for each intersection to avoid a traffic jam. In order to make the whole city's traffic highly efficient, the fog computing paradigm and a distributed reinforcement learning algorithm is designed to overcome communication bandwidth limitation and local optimal traffic control flow, respectively. We also demonstrate that our framework outperforms traditional systems and provides high practicability in future research for building the intelligent transportation system.

Publication Details

Wu, Q., Shen, J., Yong, B., Wu, J., Li, F., Wang, J. & Zhou, Q. (2019). Smart Fog Based Workflow for Traffic Control Networks. Future Generation Computer Systems: the international journal of grid computing: theory, methods and applications, 97 825-835.

Authors

Qiang Wu, Jun Shen, Binbin Yong, Jianqing Wu, Fucun Li, Jingqiang Wang, and Qingguo Zhou

Accepted Manuscript

Smart fog based workflow for traffic control networks

Qiang Wu, Jun Shen, Binbin Yong, Jianqing Wu, Fucun Li, Jinqiang Wang, Qingguo Zhou

PII: S0167-739X(18)32580-9
DOI: https://doi.org/10.1016/j.future.2019.02.058
Reference: FUTURE 4811

To appear in: *Future Generation Computer Systems*

Received date : 24 October 2018
Revised date : 31 January 2019
Accepted date : 22 February 2019

Please cite this article as: Q. Wu, J. Shen, B. Yong et al., Smart fog based workflow for traffic control networks, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.02.058

Smart Fog Based Workflow for Traffic Control Networks

Qiang Wu[a], Jun Shen[b], Binbin Yong[a], Jianqing Wu[b], Fucun Li[b], Jinqiang Wang[a], Qingguo Zhou[a,*]

[a] School of Information & Engineering, Lanzhou University, Lanzhou, China
[b] Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, Australia

Abstract

In this paper, we propose a novel traffic control architecture which is based on fog computing paradigm and reinforcement leaning technologies. We firstly provide an overview of this framework and detail the components and workflows designed to relieve traffic congestion. These workflows, which are connecting traffic lights, vehicles, Fog nodes and traffic cloud, aim to generate traffic light control flow and communication flow for each intersection to avoid a traffic jam. In order to make the whole city's traffic highly efficient, the fog computing paradigm and a distributed reinforcement learning algorithm is designed to overcome communication bandwidth limitation and local optimal traffic control flow, respectively. We also demonstrate that our framework outperforms traditional systems and provides high practicability in future research for building the intelligent transportation system.

Keywords: Fog Computing, Traffic Congestion, Reinforcement Learning, WorkFlows

1. Introduction

With the development of urbanization, ever-growing vehicles bring huge convenience to people's mobility, on the other hand, lead to traffic jams, causing several serious social problems: longer driving time, more fuel consumption and heavier air pollution [1]. For example, the loss of extra driving time and gasoline due to traffic congestion in the

*Qingguo Zhou is the corresponding author.
Email addresses: wuq17@lzu.edu.cn (Qiang Wu), jshen@uow.edu.au (Jun Shen), yongbb14@lzu.edu.cn (Binbin Yong), jw937@uowmail.edu.au (Jianqing Wu), fl626@uowmail.edu.au (Fucun Li), jqwang16@lzu.edu.cn (Jinqiang Wang), zhouqg@lzu.edu.cn (Qingguo Zhou)

US was already up to 121 billion US dollars in 2011, and carbon dioxide produced during congestion was 25,396 tons, while there were 24 billion US dollars lost and 4,535 tons carbon dioxide in 1982, respectively [2].

Growing vehicles, shortage of traffic infrastructures, inefficient traffic signal control and insufficient online traffic condition information (for example, whether traffic jams or accidents have happened on planned routes for vehicles), are primary factors for traffic congestion. It is unrealistic to ban increasing vehicles or invest more traffic infrastructures, especially for developing countries like China. To address this issue, the relatively practical methods is to focus on the last two factors. Thus, our research motivation is to (1) make real-time traffic condition information available for every vehicle on the intersection and (2) improve the efficiency of traffic light signal control.

In recent years, as the rapid development of information and communication technologies (ICT) and the advances of the Internet of Things (IoT) [3], equipping vehicles with wireless communication capabilities, has been a new standard for car makers, especially in electric vehicles. The vehicle is no longer a relatively closed-system, instead nowadays they can connect to the Internet and even other vehicles. Connected vehicles network could get traffic situation around and ahead for vehicles, which could help vehicles to alter their route dynamically to detour around the traffic jam [4]. Meanwhile, with the significant progress of evolution of AI technologies, autonomous driving cars have appeared on the road in several cities in the US, it is easier and more precise for a vehicle to tune its route at any time [5]. Moreover, traffic lights signal controls have significantly been improved owing to deep reinforcement learning [6]. For example, reinforcement algorithms can be applied to control the green light timing and red light timing adaptively.

In spite of a great diversity of technologies having improved the current traffic system, there are still a few unresolved problems. Such as: (1) high latency communication for vehicles with the number of vehicle growing; (2) some reinforcement learning algorithms merely making one intersection traffic flow smoother rather than for the local region or even the whole city [7]; (3) lastly, multi-agent reinforcement learning algorithms are designed to address *problem (2)* above are limited by communication bandwidth to apply the real traffic infrastructures [8].

An integrated solution for traffic congestion is designed to address smart traffic on a crossroad in the real world. As for contribution this paper, we would focus on optimizing connected vehicles network and reinforcement learning methods, which are the key players to the evolution to the next generation of intelligent transportation systems. In this paper, we propose a novel traffic control architecture, which integrated workflow based fog computing paradigm and a distributed reinforcement learning algorithm. We will give an overall solution to traffic congestion, which is more suitable for driver-less vehicles to some extent. The framework is composed of three components, including connected vehicles network as *terminals* at the bottom, intelligent fog computing nodes in the middle and traffic cloud center on the top. Connected vehicle network component, is designed to send the limited information of a vehicle such as its current speed, destination. And it receives the outside information flow from intelligent Fog Nodes, which are applied to help the vehicle inner system or driver to make better decision in order to avoid traffic jam. Intelligent fog computing node component generates dynamic traffic light control flow and delivers traffic condition information flow to control traffic lights, and also inform vehicles traffic condition information, respectively. Traffic cloud center analyzes the data flowed from local Fog Nodes and produces generalized control flow back to Fog Nodes to help them to jump out of local optimal, which means optimizing one or a few traffic lights on crossroad not all traffic lights in the city. It also delivers the traffic information to the Fog Node that requires it for the specific vehicle, so that every vehicle has its own information from the cloud.

The paper's contribution is (1) designed to make real-time traffic condition information available to vehicles and (2)improve the efficiency of traffic signal control with low latency communication delay.

The remainder of this paper is organized as follows: Section 2 introduces preliminary and related works, such as fog computing, connected vehicle network, traffic lights signal control. Section 3 describes our smart traffic network architecture components. Section 4 details the intelligent workflows based on fog computing paradigm. Section 5 shows the evaluation of our architecture in comparison with traditional frameworks.

Section 6 concludes the paper and addresses the future work.

2. Preliminary and Related works

This study is primarily related to two broad categories of research, one on the IoT, another on AI technologies. In this section, we briefly introduce cloud computing and fog computing paradigms, connected vehicles network, vehicular automation and reinforcement learning methods for traffic lights control and the related works to address the traffic congestion.

2.1. Internet of Things

The IoT, which based on Wireless Sensor Networks (WSN) and communication technologies, connects millions of physical devices, vehicles, home appliances and other items embedded with electronics, and enables these objects exchange data [3]. It has primarily scaled up the traditional Internet, from 'person to person' to 'person to things' even 'things to things'.

2.1.1. Connected Vehicles Network

Connected vehicles network is one of basic components of IoT, which pays close attention to the network of vehicles and traffic infrastructures. It refers to the wireless connectivity enabled vehicles that can communicate with their internal and external environments and supports the interactions of V2S (vehicle-to-sensor on-board), V2V (vehicle-to-vehicle), V2I (vehicle-to-infrastructure) [9]. Anda et al. proposed VGrid, which was a presented vehicular network for traffic control [4] in 2005. Dresner et al. [10] proposed a system to achieve automated vehicle intersection control using a reservation approach in 2008. Jackeline Rios et al. proposed online connected vehicles at merging roads to produce a smooth traffic flow without stop-and-go driving [11] in 2015.

2.1.2. Cloud Computing

Cloud computing is a paradigm for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [12]. Now, many cloud service providers, including Google, Amazon, IBM, Microsoft, are currently nurturing this popular computing paradigm as a standard computing infrastructure. Vehicular cloud is a specific form of cloud computing, applies pooled resources and dynamically serves vehicles from conventional cloud computing. Yu et al. introduced a content-based routing, that allows vehicle cloud applications to store, share and search data entirely within the cloud [13]. Currently, due to the limitation of network bandwidth and computing power, it is challenging for connecting and increasing number of vehicles to the cloud and computing all traffic data centrally.

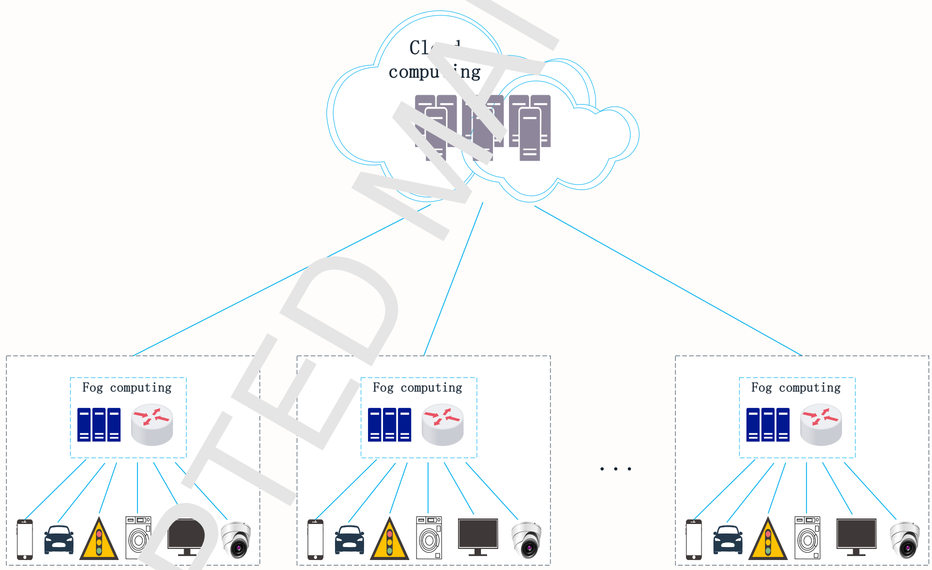


Figure 1: An illustration of fog computing paradigm.

2.1.3. Fog Computing

Cloud computing is geographically bounded by dispersed but also relatively centralized virtual servers which locate far from the end devices or users. However, real-time and latency-sensitive computation service requests often need to be immediately responded. The distant cloud center usually causes substantial round-trip delay, network congestion, and service quality degradation. To resolve these negative issues for cloud computing, a new paradigm named fog computing as shown in Fig.1 has recently been proposed, which extends the cloud computing paradigm to the edge of the network. Fog computing can create a new generation of applications and services [14]. It is not necessary to send everything to the cloud, the Fog Node near the end devices or users can also compute, store, filter data and make decisions locally. Only when it needs help from the cloud center, it sends filtered data to it.

2.1.4. Cloud Workflows

A workflow can usually be described using formal or informal flow diagramming techniques, showing directed flows between processing steps. Single processing steps or components of a workflow can basically be defined by three parameters: input description, transformation rules, output description. The advantages of keeping workflows in the cloud are stacking up fast. For example, it can be tailored to your Needs; all data is stored in one Location; reduced dependency on IT and so on. Yuan et.al and Xu et.al have presented algorithms and strategies to make cloud workflows more efficient [15] [16].

2.2. AI Technologies

AI Technologies, especially in deep learning, which consists of various derivatives of artificial neural network like convolutional neural network (CNN), recurrent neural network (RNN) and so on, have been successfully applied in the fields such as speech recognition, image classification, machine translation [17]. Meanwhile, self-driving cars and reinforcement leaning for traffic control also mainly beneficial from recent AI technologies.

2.2.1. Reinforcement Learning

Reinforcement Learning (RL) aims to train an agent which applies actions optimally to an environment. The goal of the algorithm is to learn an optimal policy for the agent to gain maximal reward, based on the observable state of the environment. The reward (either good or bad) was obtained after it applied an action (As Fig.2 shows).

Formulaically, each time step t, the agent tries to maximize the expected total return R_T, where T is the number of time steps until an episode completes, which means one subsequence of agent-environment interactions between initial and terminal states. The accumulated rewards obtained after performing each action would be: $R_T = r_1 + r_2 + ... + r_T$.

In recent years, deep reinforcement learning has been playing essential role in games, robotics, natural language processing, etc. We have been witnessing breakthroughs, like human-level control through deep reinforcement learning[18] and AlphaGo[19], Hierarchical Reinforcement Learning[20], adversarial imitation learning[21], etc.

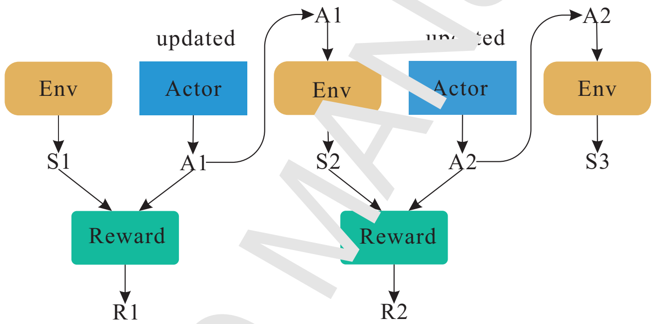


Figure 2: An illustration of RL agent that interacts with its environment in discrete time steps. At each time t, the agent receives an observation S_t and chooses an action A_t, which get reward R_t after interacting environment Env. The environment moves to a new state S_{t+1} and the reward R_{t+1} associated with the transition (S_{t+1}, A_{t+1}). The Actor is updated for choosing a proper action to make the reward as big as possible.

2.2.2. Vehicular Automation

Vehicular automation, namely smart driver-less car, makes the use of an AI system to assist a driver's operation, even by replacing the human driver. The autonomous vehicles must have the capability of perceiving its external environments. It can reduce the traffic congestion because the AI control system could strictly obey the traffic rules.

2.2.3. Traffic Lights Signal Control

Traffic lights signal control, which dynamically alters traffic light signal time according to real time traffic condition in an intersection road, has been proved to be an effec-

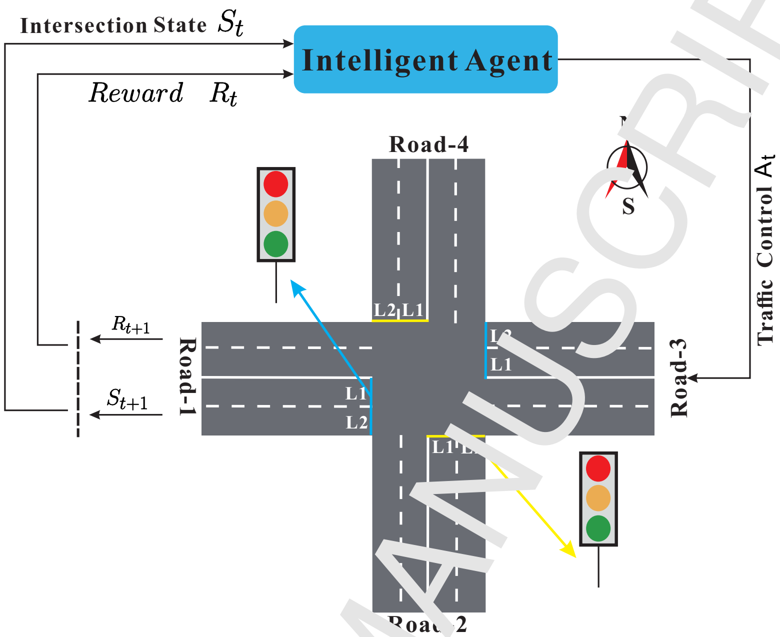


Figure 3: An illustration of an intersection, which has two directions: North-South and West-East and four roads: Road1, Road2, Road3 and Road4. An agent interacts with the intersection at discrete time steps, $t = (0, 1, 2, ...T)$, T is the number of time steps ahead until one episode finishes. The goal of the agent is to reduce vehicle staying time and the times of stop at this intersection. Specifically, such an agent first observes intersection state S_t at the beginning of time step t, then selects and actuates traffic signals A_t. After vehicles move under actuated traffic signals, intersection state changes to a new state S_{t+1}. The agent also gets reward R_t at the end of the time step t. In time sequence, the agent interacts with the intersection as $S_0, A_0, R_0...S_t, A_t, R_t...S_T, A_T, R_T$. Such reward serves as a signal guiding the agent to choose a proper action.

tive method to ease traffic congestion, by using reinforcement learning method. Fig.3 shows the reinforcement learning method applied on intersection road for traffic control. Prashanth et al had proposed to use reinforcement learning method to control traffic signals [22]. In 2017, SS Mousavi et al. used policy-gradient and value-function-based reinforcement learning algorithm to optimize the traffic light signals[23]. Although these related works lead to promising academic results, the prospects of deploying these frameworks into the real world is uncertain, because they ignored the following situations:

(1) The vehicular cloud architecture will be limited by respectively high latency communication due to the growing number of cars.

(2) The traditional vehicle or even self-driving car is not well-informed the road traffic condition information ahead.

(3) Although a few multi-agent reinforcement learning algorithms, which attempted to make multi-intelligent-agents to "talk to each other" for optimal decision, were designed to address the local optimal problem (as we have mentioned in section 1) [8], they are limited by communication bandwidth to apply in the real-time traffic infrastructures.

Hence, we design an integrated architecture based on fog computing and a distributed reinforcement learning algorithm, applying fog and cloud workflows, to reduce communication delay and make more efficient traffic light control to avoid traffic congestion for a whole city.

3. Smart Traffic Network Architecture

Smart traffic network architecture is designed specifically to address the three challenges raised above. In this section, we detail the novel framework based on fog computing, including connected vehicle network component, intelligent fog computing node component and cloud computing component as shown in Fig.4. We will describe these individual components respectively.

3.1. Problem Description

The traffic congestion problem is described as: Given intersections of a city $I = (i_1, i_2, ...i_N)$ where N is the number of intersections. We need to generate optimal or suboptimal traffic lights control flow $Control = (cont_1, cont_2, ...cont_N)$ and traffic condition flow which means the real-time traffic condition information. $Condtion = (cond_1, cond_2, ..., cond_N)$ for every intersection. Traffic control flow is to control traffic lights and condition flow is to be routed to the vehicles that will pass the intersection in the future. Thus, the problem is to generate proper $Control$ and $Condition$ flow to make the city traffic smoother.

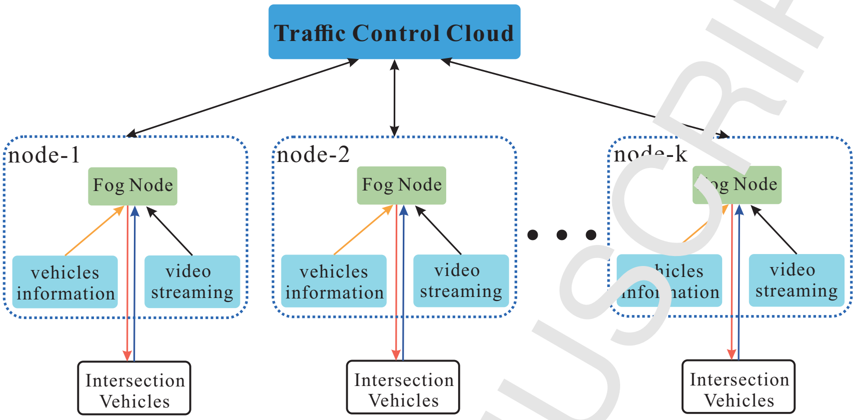


Figure 4: An overview of smart traffic network architecture. The fog terminal device synchronously transmits the detected intersection real-time traffic information to the cloud computing scheduling platform, which can output traffic control flow and schedule traffic information flow according to the deep reinforcement learning.

3.2. Connected Vehicle Network Component

The connected vehicle network component focuses on a communication network for vehicle-to-road infrastructure (Fog Node). We would not take vehicle-to-vehicle network into consideration as it could add exponential growth in network complexity. Vehicles are considered as *terminal devices* located in the bottom of the framework as shown in Fig.4. Information interaction between vehicles and Fog Nodes will be increasing the awareness of vehicles to traffic condition information ahead and providing drivers or self-driving system such information to avoid traffic jam. This component is a one-to-many network. A Fog Node connects every vehicle at one intersection, which has two main functions: (1) Send inner-vehicle information to Fog Node (2) Acquire traffic condition information from Fog Node.

3.2.1. Send information to Fog Node

The vehicle could send its private information automatically or manually, including its velocity, destination, and even particular information like, *'there is a patient in the car we must get to the hospital as quickly as possible'*. More importantly, Fog Node can turn traffic light according to the particular information from the vehicle.

3.2.2. Acquire Traffic Condition Information from Fog Node

The vehicle is able to acquire current traffic condition of its route ahead from Fog Node. As a result, the vehicle may avoid the looming traffic jam with the route replanning through its navigation system or AI system.

3.3. Fog Node Component

This component is the crucial part of the framework, which aims at producing intelligent signals to control traffic light according to general condition of an intersection road, which means position and speed of vehicles information, any specific information from vehicle (particular information), and a global control signal from cloud center which we will discuss later. This component leverages a distributed reinforcement learning algorithm under both fog and cloud computing paradigms, generating the intelligent control flow and delivering the traffic condition flow as mentioned earlier. To make real-time, highly efficient, relatively precise traffic control signal, we use fog computing paradigm, which can compute and generate control flow close to the data produced by vehicles, and we use reinforcement learning algorithm locally instead of cloud center, which is far from real-time data. Thus, the fog computing node can allow low latency communication flow and exploit computing ability locally rather than merely network routing. This component has four main functions: (1) it is can to filter raw data from local nodes so that it reduces the network load (2) it applies a distributed reinforcement learning algorithm to generate traffic light control flow. (3) it sends its condition of the intersection to the cloud and receives the other intersections condition flow from cloud. (4) it receives, processes and responses information flow to a particular vehicle, which is useful for driver or self-driving system to alter the route to avoid traffic jam as discussed as our primary design strategy.

3.4. Cloud Computing Component

Intuitively, traffic cloud computing paradigm collects all of the intersections' condition information (vehicle position and speed) and then produces dynamical control signals to traffic lights to tune the traffic flow according to real-time traffic conditions. However, it is limited by network bandwidth and cloud computation ability. It is not practical that all vehicle information is sent to cloud centers and cloud centers compute all vehicles information centralizedly to generate control flow for every intersection. On the other hand, we can not exclude cloud computing component, by only relying on Fog Node. While Fog Node could produce traffic light control signals to reduce waiting time of the vehicles on the intersection, it could barely focus on one intersection. One or even every traffic intersection optimization is not meaningful to the whole city traffic optimization. It becomes a multi-objective optimization problem to trade off fog and cloud computing. For example, the cause of a traffic congestion on intersection I_i, may be influenced by the problematic control of intersection I_j. Every intersection interplays each other, especially in near intersections. Cloud computing component is designed to be a bridge between intersections of the whole city, which could connect all intersections and reduce the network complexity of every Fog Node. Cloud computing component gathers all critical information from all the intersections in the city (filtered by Fog Node), which, in return, helps Fog Nodes to make more intelligent decisions.

3.5. Traffic Network Control Algorithm

We propose a distributed reinforcement learning algorithm under the fog computing paradigm and the cloud computing paradigm to control traffic lights between intersections in a city. In this algorithm, we need to find out multi-control policies for traffic light at each intersection. Policy $\pi_G(\theta) = (\pi_1(\Theta), \pi_2(\Theta), ..., \pi_N(\Theta))$, where N is the number of intersections, $\pi_j(\Theta)$ is the j-th traffic light control policy that means the action of changing the traffic light current state or not, by determining the timing of green light and the timing of red light at intersection j-th. We define j-th single-intersection state as $STATE_j = (L_j, S_j, I_j, C_j)$, where L_j is the length of the queue of vehicles. S_j is the sum of times of vehicles' stop action. I_j is a particular information from the vehicle and C_j is the cloud control from traffic could center. We define j-th $reward_j = -(L_j + S_j + \alpha I_j + \beta C_j)$ and final goal reward function as $Global_R = \sum_{i=1}^{N} reward_i$, where α, β are hyper-parameters that determine the importance of the particular information and cloud control flow. Our algorithm's final goal is to find control policy $\pi_G(\theta)$ to make $Global_R$ as big as possible (the length of the queue is shorter and the sum of stops is smaller). To implement this optimization

, we introduce a distributed algorithm that involves two sub-algorithms, one is for Fog Node, the other is for cloud center. They are connected by the data flow.

Algorithm 1 Cloud Control Algorithm

Ensure: To make maximum $Global_R = \sum_{i=1}^{N} reward_i$ and to produce cloud action control flow and dynamically route traffic condition for every Fog Node.
Ensure: To deliver the $Condition$ the Fog Node.

let $S_t = (State_1^t, State_2^t...State_N^t)$
$Action^t = (action_1^t, action_2^t...action_N^t)$
$p(Action^t|\theta, S^t)$ is the probability to choose action
for each episode$((S^1, Action^1, Global_R^2...S^{T-1}, Action^{T-1}, Global_R^T)$ **do**
  for t=1 to T-1 **do**
    $\theta_G = \theta_G + \nabla_{\theta_G} \log(p(Action^t|\theta, s^t)) * Global_R^t$
  end for
end for
send $Control = (cont_1, cont_2...cont_N)$ to $(FN_1, FN_2...FN_N)$
send $Condtion = (cond_1, cond_2...cond_N)$ to $(FN_1, FN_2...FN_N)$

Algorithm 2 Fog Control Algorithm

Require: Input traffic State $S_{fog_i}^t = (L_i^t, S_i^t, I_i^t, C_i^t)$ C_i is cloud control flow and Conditon flow $Condtion_{fog}$ from cloud. T is the number of time steps until one episode finishes.
Ensure: To maximize the i-th Fog Node reward $reward_i$
Ensure: To deliver the $Condition$ the vehicle needed.

$p(action^t|\theta, S_{fog_i}^t)$ is the probability to choose action
As a Fog Node i: $reward_i = (L_i + S_i + \alpha I_i + \beta C_i)$
for each episode $S_{fog_i}^1, \pi_\theta, reward_i^2...S_{fog_i}^{T-1}, \pi_\theta^{T-1}, reward_i^T$ **do**
  for t=1 to T-1 **do**
    $\theta = \theta + \nabla_\theta \log(p(action^t|\theta, S_{fog_i}^t)) * reward_i^t$
  end for
end for
send $Control_{fog}$ to TrafficLight
send $Condition_{fog} = (cond_1...cond_N)$ to vehicles$(vehicle_1...vehicle_M)$

---

4. Smart Traffic Network Workflow

Smart traffic network workflows, which are mainly generated by components we detailed above, are composed of four parts, including: (1) vehicle information flow; (2) traffic condition information flow; (3) Fog Node control flow; (4) traffic cloud control flow. Fig.5 shows these workflows altogether making the whole architecture work.

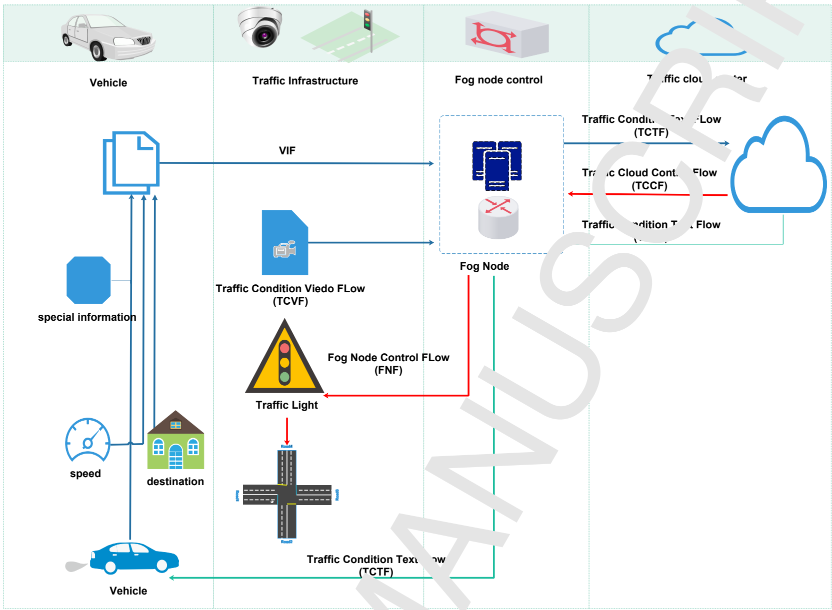


Figure 5. The smart traffic workflows

4.1. Vehicle Information Flow (VIF)

Vehicle information flow, which is generated by connected vehicles network component, contains information of each vehicle's speed, destination and the special information at one intersection which is defined as $(S, D, P)_j = ((S_1, D_1, P_1), (S_2, D_2, P_2), ..., (S_M, D_M, P_M))$, where M is the total number of vehicles on j-th intersection in city. $(S, D, P)_j$ is sent to the nearest Fog Node FN_j. In this workflow, every vehicle is a participant at an intersection. We transfer VIF to Fog Node instead of generating this information flow from equipments like speed-measuring radar. Because it could reduce the time in producing this information flow and some inner information might not be detected by devices outside the vehicle. In addition, the detection of vehicle information takes relatively more time and requires more computation power if compared to getting VIF directly.

4.2. Traffic Condition Video Flow (TCVF) and Traffic Condition Text Flow (TCTF)

Traffic condition video flow is sent to Fog Node from intersection cameras, which capture traffic condition and produce video files at every intersection. Cameras post TCVF flow to the nearest Fog Node FN_j to generate TCTF as $V2I_j = (L, Stop, A)_j$ at j-th intersection in a city. It is processed by Fog Node using Computer Vision (CV) technologies to generate continuous text flow, which could largely decrease the file size of TVCF but contain main information of the intersection condition. TCTF involves the roads information like the dynamic queue length of vehicles L_j and the sum of times of vehicles' stop on different directions (North-Source and West-East) $Stop_j$. Whether traffic congestion or an accident happened is A_j on north-south or west-east road $True/False||NS/WE$ on the intersection.

4.3. Fog Node Control Flow (FNF)

Fog Node control flow is computed by Fog Node component. The j-th Fog Node (FN_j) receives flows like VIF, TCVF and TCCF from vehicles, camera and traffic cloud center respectively, where TCCF is traffic cloud center control flow. FN_j transfers the TCVF flow into continuous row text data TCTF. Then, it sends TCTF to cloud center. FN_j also computes traffic light control flow and delivers traffic cloud specific information to the specific vehicle, letting the vehicle to know the traffic condition ahead in its previous planned route.

4.4. Traffic Cloud Control Flow (TCCF)

Traffic cloud center collects all Fog Nodes flow (VIF, TCTF) processed and filtered (transfer vedio to continuous text) locally, applying cloud reinforcement learning algorithm to produce overall condition control flows and route TCTF for all Fog Nodes. These flows are sent back to each Fog Node. The overall condition control flow aims at adding a global view that makes fog reinforcement algorithms jump out of local optimal. Meanwhile, the TCTF is routed for making vehicles well-informed the traffic condition in its previous planning route.

5. Simulation and Experiments

In this section we have utilized Simulation of Urban Mobility (SUMO)[24], which is an open source simulator for traffic environment. Meanwhile, as Fig.6 shows, we have assumed that there are six intersection nodes in one city. We have conducted a set of experiments implementing the SUMO TraCI (Traffic Control Interface) extension, which allows for dynamic control of the traffic lights at runtime, and we have compared it to another traffic control frameworks. We will demonstrate that our framework can outperform these systems. We had not considered traffic condition flow in these experiments because it can mainly increase the complexity of the simulations. It is for simplicity which will not make our experiments or comparison less valid.

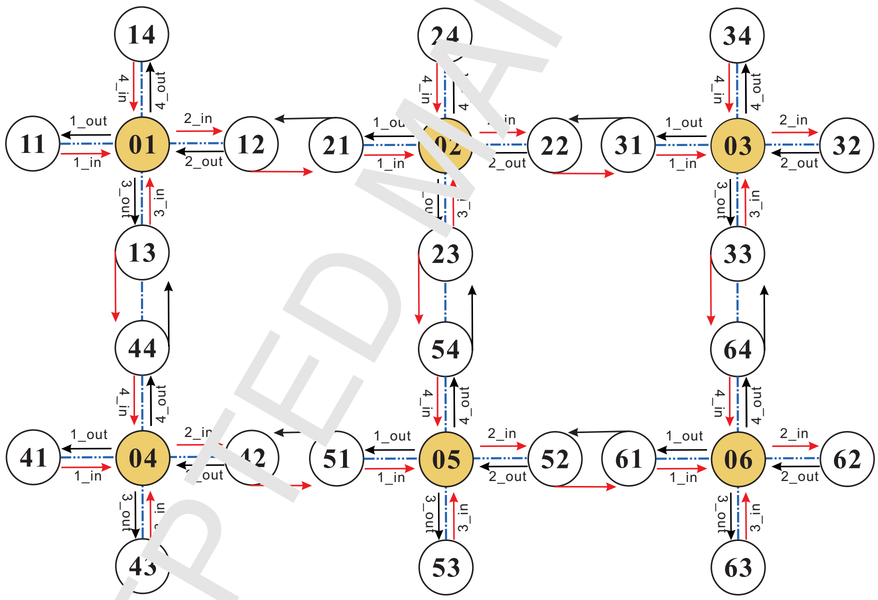


Figure 6: An illustration of city traffic in the experiments

5.1. Simulation Settings

Intersection: As Fig.7 shows, an intersection has two directions (West-East and North-South), four nodes ($node_1, node_2, node_3, node_4$), four entries (in) and four exits (out), as well as one traffic light $node_0$. Road length is set to 350 meters and road speed limit

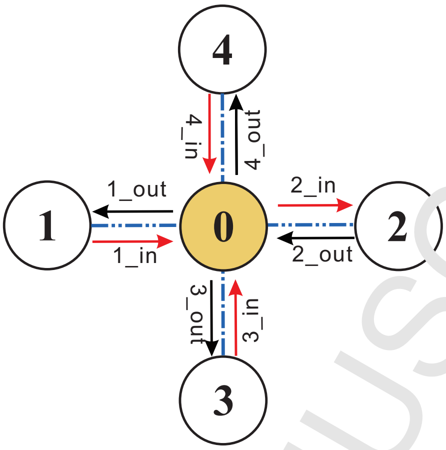


Figure 7: An intersection in the experiments

is set to be 20 (m/s).

Fog Node: We apply fog reinforcement algorithm into a docker container [25], which is logically located near the intersection (communication time delay is set relatively small). Hence, there are six docker containers (Fog Node) to generate traffic control flows.

Cloud Center: We also apply cloud reinforcement algorithm into one docker container logically located far from the intersection (communication time delay is set relatively large).

Communication Time: The communication delay from cloud center to an intersection is set as 10 seconds (sleep 10 seconds in the code to simulate), while the delay from the Fog Node to an intersection is set as 1 second.

Traffic Control Timing Cycle: We set traffic control timing cycle as 60 seconds, and initially set green light interval $g_t = 27$ seconds , red light interval $r_t = 27$ seconds and yellow light interval $r_t = 6$ seconds.

Episode: Each subsequence of agent-environment interactions between initial and terminal states is an episode. One episode time is set as 10 minutes (600 seconds), including 10 traffic control timing cycles.

Vehicle Simulation: At one intersection, we assume vehicles arrive at road entrances randomly following the Bernoulli process with the same probability $P_{in} = \frac{1}{10}$. Every vehicle has a random destination node except of the entry node (we set $random(seed) = 11$). In one episode, there are approximately 600 vehicles (there are 10 entrances in our city map).

5.2. *Simulation Evaluation*

We define the time, which a vehicle enters an entry of the intersection until it passes through, as t_m, where m is the vehicle number. During simulations, we record the time data for all vehicles on one intersection in every episodes. Then, we sum all the time data over all the intersections, $T_e = \sum_{i=1}^{I} \sum_{m=1}^{M} t_m$, to evaluate the traffic network, where E is the number of episode, M is the number of vehicles and I is the number intersections that every vehicle will pass through. We herein set $E = 200, I = 6$. We firstly trained four frameworks, including Static-Timing Framework, RL-only Framework, RL-cloud Framework, Smart-Fog Framework. They were trained in 4000 episodes by SUMO. Then, we conducted simulations by SUMO in these frameworks in 200 episodes, and recorded the waiting time data for all vehicles in all simulations episodes (test part), respectively.

5.2.1. *Baseline*

In the simulations, we fix all the traffic light timing that has been previously set.

5.2.2. *RL-only-Framework*

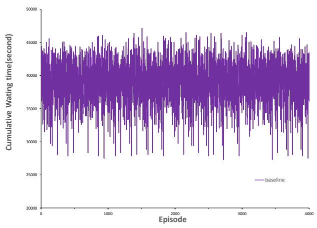
We employ RL-only framework with reinforcement learning algorithm [26] to control traffic light. The delay from framework to an intersection is set 1 second which is the same as the Fog Node (six docker containers generate traffic control signal separately).

5.2.3. *RL-cloud Framework*

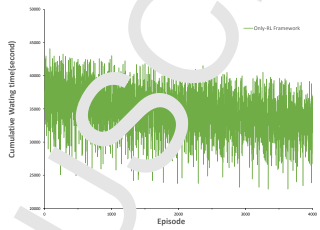
We use RL-cloud-framework with RL algorithm [26] to cloud center that recognize all intersections as one to generate traffic control flow for each intersection (one docker container generate overall traffic control signal). The communication delay from framework to an intersection is set as 10 seconds.

5.2.4. Smart-Fog Framework

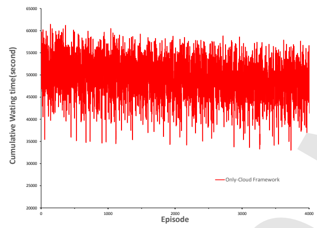
We apply our framework with Fog Node and cloud center in six fog docker containers and one cloud docker container and record the waiting time for all vehicles.



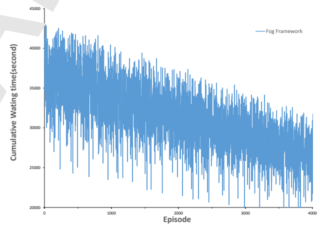
(a) Baseline



(b) Only-RL Framework



(c) Only-Cloud Framework



(d) Fog Framework

Figure 8: Training Process of four frameworks

Model	Average Episode Time (second)	Average Delay Time (second)	Delay Rate
Baseline	38827.5	2977.2	7.7%
RL-Only Framework	35789.33	4522.9	12.6%
Cloud-Only Framework	45448.03	20809.7	**45.8%**
Fog Framework	**31340.46**	**6951.6**	**22.1%**

Table 1: The Average Delay Time (one episode for 600 vehicles) of four Frameworks

5.3. Experiment Results

In the training process of four frameworks, as Fig.8 shows, we can easily see that the cumulative waiting time of all vehicles in one episode with Static-Timing (Baseline), RL-only, RL-cloud, Smart-Fog Framework for 4000 episodes (One episode has approximately 600 vehicles, we add up their waiting time). Our Smart Fog Framework performed better than others for lowering the cumulative waiting time. Then, we also compare the cumulative waiting time of all vehicles in four frameworks for 200 episodes (for test). As Fig.9 shows, our framework makes the vehicles wait 28% less time than baseline. Because of the communication delay, RL-Cloud Framework even take 34% more time than the baseline.

In these experiments, we did not consider traffic condition flow. Because it is relatively difficult to simulate a smart car which receives TCTF then change its route to avoid a traffic jam. However, we intuitively maintain that the TCTF will make our framework more robust.

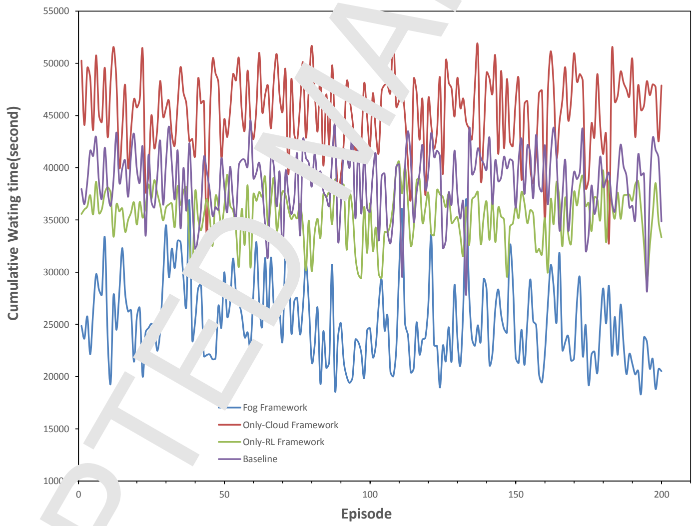


Figure 9: The cumulative waiting time for four frameworks in one episode lasting 200 episodes

6. Conclusions

In this paper, we have proposed a novel smart traffic control architecture based on fog computing paradigm and a distributed reinforcement learning algorithm to lower the probability of traffic congestion in the city. It can overcome communication bandwidth limitation among vehicles by producing smart traffic control signal locally and delivering traffic condition signal intelligently. Workflows in the framework are designed to make the architecture work efficiently. Although the framework mainly computes locally, it could make optimal or suboptimal global control signals instead of local optimal control signals via our distributed reinforcement learning algorithms and deliver the traffic condition information to the specific vehicle through cloud center and Fog Node. We compared our framework with others in simulators, which demonstrated that it could make more efficient control signals to reduce traffic jams. It is not only suitable for the current vehicles but also more useful for driver-less vehicles in the future as it will be able to plan its route much more intelligently with the information from Fog Node.

As for the limitation of our approach, where the probability of vehicle in the crossroad is roughly estimated, we have planed to make real vehicle flow in the further research. We also maintain that our model can be extended to real world scenarios to solve the traffic problem on the crossroad. In the future work, we will design simulations on real city maps and collect more real traffic data to produce a vehicle generation model (taking rush hour into consideration). Also, we will add well-informed intelligent vehicles (dynamically planning its route from road information) by smart simulators and algorithms to identify the advantages of our framework.

7. Acknowledge

This work was supported by National Natural Science Foundation of China under Grant No. 61402210 and 60973137, State Grid Corporation Science and Technology Project under Grant No. SGGSKY00FJJS1700302, Program for New Century Excellent Talents in University under Grant No. NCET-12-0250, Major National Project of High Resolution Earth Observation System under Grant No. 30-Y20A34-9010-15/17, Strategic Priority Research Program of the Chinese Academy of Sciences with Grant No. XDA03030100, Google Research Awards and Google Faculty Award.

References

[1] M. Alsabaan, W. Alasmary, A. Albasir, K. Naik, Vehicular networks for a greener environment: A survey, IEEE Communications Surveys Tutorials 15 (3) (2013) 1372–1388.

[2] T. J. Schrank, The 2012 urban mobility report. `http://tti.tamu.edu/documents/ums/archive/mobility-report-2012-wappx.pdf`.

[3] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787–2805.

[4] J. Anda, J. Lebrun, D. Ghosal, C. N. Chuah, M. Zhang, Vgrid: vehicular adhoc networking and computing grid for intelligent traffic control, in: Vehicular Technology Conference, 2005, pp. 2905–2909 Vol.5.

[5] S. Thrun, Toward robotic cars, Communications of the ACM 53 (4) (2010) 99–106.

[6] S. Mozer, M. C, M. Hasselmo, Reinforcement learning: An introduction, IEEE Transactions on Neural Networks 16 (1) (2005) 285–286.

[7] D. Zhao, Y. Dai, Z. Zhang, Computational intelligence in urban traffic signal control: A survey, IEEE Transactions on Systems Man Cybernetics Part C 42 (4) (2012) 485–494.

[8] A. L. C. Bazzan, Opportunities for multiagent systems and multiagent reinforcement learning in traffic control, Autonomous Agents and Multi-Agent Systems 18 (3) (2009) 342.

[9] N. Lu, N. Cheng, N. Zhang, X. Shen, Connected vehicles: Solutions and challenges, Internet of Things Journal IEEE 1 (4) (2014) 289–299.

[10] K. Dresner, P. Stone, A multiagent approach to autonomous intersection management, J Articial Intelligence Research 31 (3) (2008) 591–656.

[11] J. Rios-Torres, A. Malikopoulos, P. Pisu, Online optimal control of connected vehicles for efficient traffic flow at merging roads, in: IEEE International Conference on Intelligent Transportation Systems, 2015, pp. 2432–2437.

[12] Armbrust, Michael, Fox, Armando, Griffith, Rean, Joseph, D. Anthony, Katz, H. Randy, Above the clouds: A berkeley view of cloud computing, Eecs Department University of California Berkeley 53 (4) (2009) 50–58.

[13] Y. T. Yu, T. Punihaole, M. Gerla, M. Y. Sanadidi, Content routing in the vehicle cloud (2012) 1–6.

[14] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Edition of the Mcc Workshop on Mobile Cloud Computing, 2012, pp. 13–16.

[15] D. Yuan, Y. Yang, X. Liu, J. Chen, A data placement strategy in scientific cloud workflows, Future Generation Computer Systems 26 (8) (2010) 1200–1214.

[16] R. Xu, Y. Wang, H. Luo, F. Wang, Y. Xie, X. Liu, Y. Yang, A sufficient and necessary temporal violation handling point selection strategy in cloud workflow, Future Generation Computer Systems 86.

[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Processing Magazine 29 (6) (2012) 82–97.

[18] M. Volodymyr, K. Koray, S. David, A. A. Rusu, V. Joel, M. G. Bellemare, G. Alex, R. Martin, A. K. Fidjeland, O. Georg, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529.

[19] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

[20] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, K. Kavukcuoglu, Feudal networks for hierarchical reinforcement learning, CoRR abs/1703.01161. arXiv:1703.01161.

[21] J. Ho, S. Ermon, Generative adversarial imitation learning, CoRR abs/1606.03476. arXiv:1606.03476.

[22] L. A. Prashanth, S. Bhatnagar, Reinforcement learning with function approximation for traffic signal control, IEEE Transactions on Intelligent Transportation Systems 12 (2) (2011) 412–421.

[23] S. S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, Iet Intelligent Transport Systems 11 (7) (2017) 417–423.

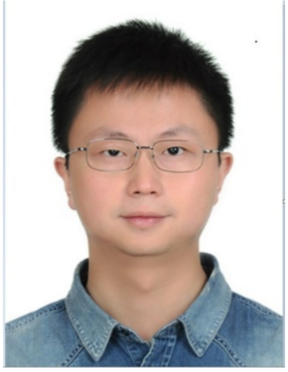
[24] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of sumo - simulation of urban mobility, International Journal on Advances in Systems Measurements 5 (3 and 4) (2012) 128–138.

[25] C. Boettiger, An introduction to docker for reproducible research, ACM SIGOPS Operating Systems Review 49 (1) (2015) 71–79.

[26] B. Abdulhai, R. Pringle, G. J. Karakoulas, Reinforcement learning for the true adaptive traffic signal control, Journal of Transportation Engineering 129 (2014) 278–285.

Highlights:

- A smart fog based workflow architecture is proposed.
- The architecture relies on the fog computing paradigm and a distributed reinforcement learning algorithm to make real-time traffic condition information available to vehicles and Improve the efficiency of traffic signal control with low latency communication delay.
- Workflows designed to relieve traffic congestion, which are connecting traffic lights, vehicles, Fog nodes and traffic cloud, aim to generate traffic light control flow and communication flow.
- The framework outperforms traditional systems and provides high practicability in future research for building the intelligent transportation system.



Qiang Wu is a PhD candidate in Lanzhou University. He received BS degree in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2009, and received MS degree from Lanzhou Jiaotong University in 2015. His research interests include deep learning, fog computing, and reinforcement learning for Intelligent Transport System.

Dr Jun Shen was awarded PhD in 2001 at Southeast University, China. He held positions at Swinburne University of Technology in Melbourne and University of South Australia in Adelaide before 2006. He is an Associate Professor in School of Computing and Information Technology at University of Wollongong in Wollongong, NSW of Australia, where he had been Head of Postgraduate Studies, and Chair of School Research Committee since 2014. He is a senior member of three institutions: IEEE, ACM and ACS. He has published more than 120 papers in journals and conferences in CS/IT areas. His expertise includes computational intelligence, Web services, Cloud computing and learning technologies including MOOC. He has been Editor, PC Chair, Guest Editor, PC Member for numerous journals and conferences published by IEEE, ACM, Elsevier and Springer. A/Prof Shen is also a current member of ACM/AIS Task Force on Curriculum MSIS 2016.

Binbin Yong received his master's degree in Computer Science and Technology from Lanzhou University in 2012, and received PhD in the School of Information Science and Engineering, Lanzhou University in 2017. He is researching in parallel computing of GPU, machine learning, deep learning and general vector machine.

Jianqing Wu is a PhD candidate at University of Wollongong, Australia. He received the BSc in computer science from Manchester Metropolitan University, UK, in 2014 and Master of Research in computer science from the University of Liverpool in 2016 respectively. His research interests include machine learning and data analytics for Intelligent Transport System.

Fucun Li graduated from University of Wollongong with master degree of Advanced Information Technology studies in 2014. He is currently a Ph.D. candidate in University of Wollongong. His main research interests are how to use information system to improve competitiveness of iron companies and how to use machine learning to control quality and cost of iron products.

Jinqiang Wang is a master candidate in Lanzhou University. His research interests include machine learning, deep learning and deep reinforcement learning.

Qingguo Zhou received the BS and MS degrees in Physics from Lanzhou University in 1996 and 2001, respectively, and received PhD in Theoretical Physics from Lanzhou University in 2005. Now he is a professor of Lanzhou University and working in the School of Information Science and Engineering. He is also a Fellow of IET. He was a recipient of IBM Real-Time Innovation Award in 2007, a recipient of Google Faculty Award in 2011, and a recipient of Google Faculty Research Award in 2012. His research interests include safety-critical systems, embedded systems, and real-time systems.