# Distributed Real-Time IoT for Autonomous Vehicles

Bigi Varghese Philip, *Student Member, IEEE,* Tansu Alpcan, *Senior Member, IEEE,* Jiong Jin, *Member, IEEE,* and Marimuthu Palaniswami, *Fellow, IEEE*

***Abstract*—Real-time *Internet of Things* (IoT) applications have stringent delay requirements when implemented over distributed sensing and communication networks in smart traffic control. They require the system to reach a permissible neighbourhood of an optimum solution with tolerable delay. The performance of such applications mostly depends on the delay introduced by the underlying optimisation algorithms, with the localised computational capability. In this work, we study a smart traffic control scenario - a real-time IoT application, where a group of autonomous vehicles independently decide on their lane velocity, in collaboration with road side units to efficiently utilise intersections with minimal environmental impact. We decompose this problem as an unconstrained network utility maximisation problem. A consensus-based, constant step-size gradient descent algorithm is proposed to obtain a near optimal solution. We analyse the delay-accuracy trade-off in reaching a near optimal velocity. Delay is measured in terms of the number of iterations required before the scheduling operation can be done for a particular tolerance. The operation of the algorithm under quantised message passing is also studied. On contrary to the existing methods to intersection management problems, our approach studies the limit at which an optimisation algorithm fails to cater for the requirements of a real-time application and must fall back for a pareto-optimal solution, due to the communication constraints. We use SUMO (simulation of urban mobility) to incorporate the microscopic behaviour of traffic flows to our simulations and compared our solution with traditional and state-of-the-art intersection management techniques.**

***Index Terms*—Distributed Optimisation, Autonomous Vehicles, Intersection Management**

## I. Introduction

LARGE scale deployment of interconnected sensors and actuators, blending seamlessly with the environment around us, is an integral part of *Internet of Things* (IoT). [1]. Among few striking applications of IoT in industry are smart transportation and logistics. Systems, where computing, communication and control technologies are tightly integrated, are broadly classified as *cyber-physical systems* (CPS). A detailed study of information framework required for CPS, especially, smart city related applications, can be found in [2]. CPS requires data from sensor networks to be processed in real-time, as they are associated with the control of physical systems. In one way or another these CPS is going to drastically influence our future. For example, *intelligent transport systems* (ITS) will be developed, in which most of driving tasks will be handled by the vehicles themselves. Efficiency of an ITS could be drastically increased by effective co-ordination among the automated vehicles, which requires real-time communication and data processing. To minimise the delay incurred in data transfer, for such real-time applications, it is better to process the data as close to the network edge. Thus, the traditional cloud-centralised approach for data processing is not always suited for large scale real-time applications. Thanks to the advances in semiconductor technology, more memory and processing power can be incorporated with distributed sensors, enabling fog computing [3] bringing down the cloud to lower layer.

This paper considers a smart traffic control setting, in which autonomous vehicles and road side units collaborate to maximise the efficiency of the intersection with minimal environmental impact in-terms of fuel consumption. Rapid change in traffic infrastructure, such as autonomous intersections, will become prevalent in future with increased penetration of autonomous vehicles [4]. Even in present systems, a trade-off between the throughput of the intersection and the delay experienced by vehicles crossing it can be obtained by adjusting the phase duration [5]. Further, for safe operation of such a system, control operation should be taken by each system agent within fraction of a second. For example, every vehicle in a lane should reduce speed as soon as possible, if the vehicle in front slows down for some reason. Let us define efficiency of an intersection as the maximum number of vehicles it can handle. Efficiency of the system can be increased by reducing the minimum headway, which is the minimum time gap to be kept between two crossing convoys for safety guarantee. The extent to which minimum headway can be reduced depends mainly on the accuracy of the calculated solution and the response time provided for the vehicles to adjust to the calculated schedule. In short, automated vehicles will have to attain global objectives with acceptable latency, using limited local communication, computing, and memory capabilities. Therefore, distributed IoT algorithms must be developed, to reach at a predetermined neighbourhood within a limited number of iterations specified by corresponding applications.

In many practical real-time applications, as explained above, global objective has to be solved collectively by agents which have access only to their local data sets. Many problems requiring distributed processing of large data sets can be posed in the framework of convex optimisation [6]. For example, the resource allocation subject to various constraints related to fairness and efficiency have been formulated as *network utility maximisation* (NUM) problems [7], where agents try to optimise the global objective of maximizing aggregate utility. The authors of [6] and [7] have analytically obtained the convergence rate estimates for the corresponding asymptotically converging algorithms, and explicitly characterised the accuracy

B. V. Philip, T. Alpcan, and M. Palaniswami are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Vic - 3052, Australia. Email: bigip@student.unimelb.edu.au;{tansu.alpcan, palani}@unimelb.edu.au

J.Jin is with the School of Software and Electrical Engineering, Swinburne University of Technology, Hawthorn, VIC 3122 Australia. Email: jiongjin@swin.edu.au

of the generated approximate optimal solutions. Trade-off between real-world factors like intersection throughput, delay experienced by crossing vehicles, minimum headway can be achieved by varying the delay-accuracy parameters of the distributed optimisation algorithm. Moreover, in many real-time applications such as transport and power systems, the underlying network graph and channel conditions are often time varying. It is thus worthwhile studying the effect of graph and channel impairments.

In this work, we adopt a constant step-size, consensus based distributed optimisation algorithm to obtain a near optimal solution. Due to scalability and delay constraints for real-time applications mentioned earlier, it is preferred to solve the problem distributively, i.e., without assigning any central node to collect and process the data from each agent. Thus, the system of agents is forced to obtain the global objective only using locally available data collected from its neighbours. In every iteration, each agent must perform an averaging and updating step. First, they collect the estimates of system variables from its neighbours and do weighted averaging to obtain an estimate of the current system variable. Next, it updates its estimate of the system variable, following a simple constant step size gradient descent on its local objective function. Additionally, we formulate a smart intersection management application using our algorithms and compare them with traditional and state-of-the-art intersection management algorithms. We use SUMO (simulation of urban mobility) [8], one of the widely employed mobility simulators, to mimic the microscopic behaviour of the traffic flows in all our simulations.

This paper is organised as follows. A literature survey on optimisation for real-time systems, intersection management applications, vehicular communications, and traffic simulators are given in Section II. Problem formulation for an intersection management application operating on centralised and hierarchical network architectures formed by vehicles and RSU is explained in Section III. Section IV briefly present our developed analytical tools and their usage in the algorithm design. Key results and the performance of the optimisation algorithms as well as the overall traffic application is explained in Section V. Section VI discusses the important observations on results in detail, followed by concluding remarks in Section VII.

## II. STATE-OF-THE-ART

This section explains the connection between delay-accuracy trade-off for real-time applications and the underlying optimisation algorithm (including our algorithm). It also briefs the state-of-the-art techniques in intersection management, vehicular communication, and the existing traffic simulators.

### A. Delay-Accuracy Trade-Off for Real-Time Systems

Authors of [9] analyse the performance of a distributed averaging algorithm under uniform quantisation scheme. They prove that either a quantised consensus or a cyclic oscillating behaviour is obtained by the system in finite time depending on initial conditions. Analysis of a distributed sub-gradient algorithm operating under a zooming-in technique based quantiser was done in [10]. A universal (quantisation scheme independent) bound on the rate of exponential mean square convergence is obtained by [11]. They analysed the primal-dual algorithm, under quantised message passing between agents and the system (bipartite graph topology) and obtained a bound on the convergence for a class of quantisation schemes. An extension of this work to a distributed system, implementing consensus based, constant step-size, gradient descent algorithm [12]. They explore the possibility of distributively obtaining a near optimum solution under finite iterations for a quadratic NUM problem.

Many of the literature discussed so far considers a totally distributed/centralised scenario, where the computation load is either evenly distributed or centralised at a single point. We consider a hybrid scenario where computation is distributed among a master node and a group of sub-nodes. While the master node solves the master problem, utilising a simple gradient-descent algorithm, the sub-problems are solved using the algorithms developed in [12].

### B. Intersection Management Application

Although traffic intersections are a relatively small part of the road network, they account for a significant amount of traffic accidents [13] and traffic delay. Hence, safe, and efficient intersection management is always a prime concern for traffic engineers. Since their appearance at the end of $19^{th}$ century, traffic lights are used as a primary mode for intersection management [5]. The efficiency and capacity of a traffic network drastically increases, when the traffic lights dynamically adapt to real-time traffic conditions. Thus, static traffic light switching patterns gave way to dynamic traffic signalling, which includes wide application of communication, computation, and sensing technologies, to control the traffic flow [14]. State of the art sensing techniques like inductive loop, RFID, microwave radar, video image processor are placed either on road or road side to sense the traffic conditions in real-time. In most of the cases, sensed data are communicated to the cloud and processed to obtain traffic flow predictions. Finally, traffic lights or dynamic speed limits are controlled in accordance with the predicted traffic conditions. The main hurdle towards further improving the efficiency of such a system is the communication delays and the inaccuracy in the sensed data.

Authors of [15] do delay minimisation at a single lane one-way intersection, in a mixed traffic of conventional vehicles and connected vehicles, using V2I communication links. Their approach found slight savings on delay ($< 10\%$) at low traffic intensities ($1000veh./h$) and the savings tend to decrease at higher traffic intensities. Interestingly, this work and many similar ones in intersection management utilise only the V2I capability of autonomous vehicles to feed in and out data to a centralised controller. On the contrary, our approach is to study the effect of utilising the distributed computing and *vehicle to vehicle* (V2V) communications on performance of such applications.

In the near future, with the increasing penetration of autonomous vehicles, *cooperative intelligent transport systems* (C-ITS), where connection and cooperation between road users,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2877217, IEEE Transactions on Industrial Informatics

infrastructure and control centre will be enabled by real-time information exchange [13]. Cooperative intersection management, where road users and infrastructure jointly optimise the safety and efficiency of intersections through negotiations and cooperation, will become an integral part of future C-ITS. Unfortunately, most of the current style of implementations are not designed to utilise the distributed sensing and computational capabilities of autonomous vehicles.

### C. Vehicular Communication

Various intelligent transport applications, broadly classified into safety applications, efficiency applications and comfort applications have varying communication requirements. A succinct overview of state-of-the-art vehicular communication technologies can be found in [16]. Intersection management application falls under the category of traffic management applications, which requires highly reliable, secure, short range, low latency ($< 100ms$) communication. As suggested in [17], *wireless access in vehicular environments* (WAVE) technology satisfies these requirements under high mobility conditions. The WAVE protocol stack is composed of IEEE 802.11p and IEEE 1609.x protocols. An overview of different transfer data rates of 802.11p WAVE standard can be found in [18]. Digital channel satisfying 802.11p standards is used for both V2V and V2I communication in this work. For our application, we selected 6 Mbps, BPSK modulated scheme with a coding rate of $1/2$ for V2I link. The scheme uses OFDM symbols that could transmit $48codedbits/symbol$ ($24databits/symbol$) with a maximum symbol rate of $250symbols/second$ (6000/24). Similarly, for V2V communication, we chose $24Mbps$ links with a coding rate of $1/2$ (which translates to $4$ slave iterations per master iteration). The overheads that may be caused by headers, control bits, packet loss etc. are neglected here.

### D. Simulators

A complete simulation of vehicular wireless network requires an interplay between a mobility simulator and a network simulator (or a combined version of both) [19]. Mobility simulators like CORSIM, PARAMICS, VISSIM, AIMSUN, SUMO etc. simulates the location, velocity and acceleration of each vehicle that participates in the simulated scenario. Whereas, the role of network simulators like $ns-2$, $ns-3$ or $OMNeT++$ is to simulate the PHY and MAC layers of the wireless network.

We choose SUMO (simulation of urban mobility) [8], one of the most widely employed mobility simulators , to mimic the microscopic behaviour of the traffic flows in all our simulations. SUMO is freely available and is released under an open-source license. SUMO implements a krau$\mathcal{B}$ model - a famous car-following model, to perform vehicular mobility simulation. Further, *traffic network simulation environment* (TraNS) [20] can be used to interface SUMO with $ns-2$ using the *traffic control interface* (TraCI). Detailed PHY and MAC layer implementation of the vehicular wireless network is intended to be implemented in future. Currently, we model wireless link in Matlab using a Unit Disk Model [21]. All nodes are assumed a communication radius, $r$, a bidirectional link between two nodes, $n_1$ and $n_2$, exists if and only if $Dist(n_1, n_2) \leq r$, where $Dist(.,.)$ denotes the Euclidean distance. Each bi-directional link is assumed to have a limited capacity and is quantised accordingly. Our current simulation setup uses TraCI to interface SUMO with Matlab [22].

## III. PROBLEM FORMULATION

The primary objective of our work is to study the effect of delay-accuracy trade-off in real-time systems operating under a distributed optimisation algorithm. Different applications in smart traffic control such as emergency electronic brake lights, slow vehicle warning, pre-crash sensing, lane change warning has varying delay-accuracy requirement [23]. Thus, we expect applications to be implemented as different software modules prioritizing their operation on a same set of hardware accordingly. This modularity also eases application design, i.e., one could concentrate on any functionality by assuming that its prerequisites will be taken care of by other applications. For example, the cars can be clustered using any platooning algorithm [24] and then the intersection management algorithm can deal with the platoon head, instead of dealing with each individual platoon. Here, we consider an intersection management system, whose goal is to enable smooth flow of vehicles across the intersection. We implemented two different approaches with centralised and hierarchical network architectures. In the centralised network architecture, vehicle clusters communicate their status (e.g., velocity, distance from intersection, cluster length) and utility function parameters to a central unit allocated for the intersection. The central unit then allocates a time slot for the cluster and instructs the cluster to adjust its velocity accordingly. When it comes to hierarchical network architecture, the computational load is distributed among the RSU and vehicle clusters and distributed algorithms proposed in [12] are used by the vehicle clusters to solve the sub-problems distributively. Rest of the Section, discusses the mathematical formulation of our application.

Our application aims at maximising the intersection efficiency with minimum ecological impact. Relationship between fuel consumption and average speed of the vehicle is provided in [25]. Except the vehicle model specific parameters, they have categorised passenger cars broadly into three categories (i.e., large, medium, and small) and provided the parameters for each category in the report. According to [25], the fuel velocity relationship can be approximated with desirable accuracy using a quadratic equation in our desired speed range. The fuel consumption (in fact, negative of fuel consumption) of each vehicle is be modelled using a quadratic utility function of the form

$$f_i(v) = \frac{a_i}{2}v^2 + c_i v \tag{1}$$

where $v$ is the velocity of the vehicle, $a_i < 0, c_i \in \mathcal{R}$, $\forall i$. $a_i$ and $c_i$ are designed so that the desired velocity of each vehicle is $\frac{c_i}{|a_i|}$ and the gradient of the utility function, $a_i v + c_i$ expresses the willingness of a vehicle to vary from velocity $v$. Let us assume an intersection of two perpendicular one-way lanes, without any turn at the intersection. To add to the efficiency of the implementation, we devised a simple platooning algorithm, utilising car following model provided in SUMO. All vehicles are assumed to have a convoy management software which, considers vehicle dynamics and manages the distance between vehicles. Vehicles flowing in a lane are expected to form convoys using this convoy management mechanism.

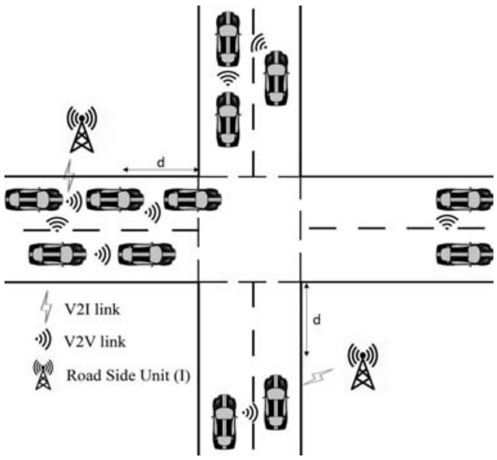


Fig. 1. Conceptual diagram of a smart traffic intersection

The idea is to set a buffer zone of length $d$, in all the lanes, before the intersection, as illustrated in Figure 1. Velocity of the vehicle convoys reaching the buffer zone is determined collectively using V2V and V2I message passing. At a particular instant, $l_i, v_i$ are the length and velocity, respectively, of vehicle convoys (hereafter mentioned as $C_i$ ) entering buffer zone in lane $i$. The traffic intersection controller could communicate to at least one of the convoy agents from a distance $d' > d$ through RSUs. Let *estimated time of arrival*, $ETA(C_i) = t + \frac{d}{v_i}$, and *estimated time of departure*, $ETD(C_i) = t + \frac{d+l_i}{v_i}$, be the respective estimated time of arrival and departure of convoy $C_i$ at the intersection. Let $ETD_0$ be the estimated time of departure of existing of cluster at the intersection.

Let $C_1$ denote the convoy that reaches the intersection first, i.e., $C_1 = argmin(ETA(C_i))$ and $C_2$ be the other convoy. Ideally, the condition for smooth intersection operation is $\big(ETA(C_1) \geq ETD(\text{current convoy})\big)$ and $\big(ETA(C_2) \geq ETD(C_1)\big)$. Thus, the optimisation problem can be formulated as

$$\begin{aligned} \underset{v_1,v_2}{\text{maximise}} \quad & \sum_{i\in C_1} f_i(v_1) + \sum_{j\in C_2} f_j(v_2) + T(\rho)(v_1+v_2) \\ \text{subject to:} \quad & ETA(C_1) \geq ETD_0 + \delta \\ & ETA(C_2) \geq ETD(C_1) + \delta \\ & v_1 \geq 0, v_2 \geq 0, v_1 \leq v_{max}, v_2 \leq v_{max} \end{aligned} \tag{$P^o$}$$

The terms $\sum\limits_{j\in C_i} f_j(v_i)$, denote the priority given to fuel minimisation by the cluster $i$, where $f_j(v_i)$ is the quadratic utility function of vehicle $j$ as explained in Equation 1. $T(\rho)(v_1 + v_2)$ is the additional term added by the traffic light to incorporate its preference to speed up the traffic flow based on the traffic intensity, $\rho$.

It should be noted that the problem is to maximise the objective function and $T(\rho)(v_1 + v_2)$ is maximum, when the sum $(v_1 + v_2)$ is maximum. $v_1$ and $v_2$ are constrained to stay within 0 and $v_{max}$ $(25mps)$, Hence in absence of any other component of the objective function, the term $T(\rho)(v_1 + v_2)$, is maximum when $v_1$ and $v_2$ are equal to $v_{max}$.

The first constraint avoids collision between upcoming cluster and the clusters crossing the intersection, whereas the second constraint prevents collision between upcoming cluster and the cluster following it. The remaining constraints ensure that the cars will not go in reverse, i.e., no negative velocity, and the maximum lane speed is maintained.

To study the effect of communication architecture on the performance of the algorithm, we will model both the centralised and hierarchical scenarios.

### A. Centralised Architecture

In order to obtain the best performance possible with the intersection management algorithm, a centralised solution of the optimisation problem $P^o$ is implemented, neglecting communication and computation delays. In the centralised scenario, vehicles communicate their objective function coefficients to the *road side unit* (RSU). The schedule is then computed at the RSU. In real-world systems with complex, locally known, objective functions, it becomes practically impossible to communicate the entire objective function due to security and communication constraints [26]. Hence, the hierarchical approach, explained in subsection III-B, is preferred. We used Matlab's Fmincon() solver to solve Problem $P^o$ centrally. This architecture is used to implement the centralised solution mentioned in the rest of the paper, whereas, all our remaining algorithms operate on the hierarchical architecture explained below.

### B. Hierarchical Architecture

A simplified version of the problem, as explained below, is used in the case of hierarchical architecture. It should be noted that during the implementation of the algorithm, $ETD_0$, is known to the intersection controller and $\delta$ is the minimum headway. Substituting for ETAs and ETDs in terms of velocities and distances translates the first two constraints to $v_1 \leq Kd$ and $v_2 \leq \frac{v_1 d}{d+l_1+v_1\delta}$, where $K = \frac{1}{ETD_0+\delta}$. To simplify the second constraint, we design the buffer distance $d$ and the minimum headway, $\delta$, such that $v_{max}\delta << d + l_1$. Thus, the second constraint becomes $v_2 \leq (\frac{d}{d+l_1}v_1 - \frac{\delta d}{(d+l_1)^2}v_1^2)$. Simplified version of the problem, as explained below, is used in the case of hierarchical architecture.

$$\begin{aligned} \underset{v_1,v_2}{\text{maximise}} \quad & \sum_{i\in C_1} f_i(v_1) + \sum_{j\in C_2} f_j(v_2) + T(\rho)(v_1+v_2) \\ \text{subject to:} \quad & v_1 \leq min(Kd, v_{max}) \\ & v_2 \leq \frac{d}{d+l_1}v_1 - \frac{\delta d}{(d+l_1)^2}v_1^2 \\ & v_1 \geq 0, v_2 \geq 0, v_2 \leq v_{max} \end{aligned} \tag{$P^s$}$$

Problem $P^s$ can be proved to be a convex optimisation problem, where the objective function and the constraint space are both convex. Convexity makes the problem easier to solve, since the local minima and the global minima are be the same. For the totally distributed scenario, the problem is to be solved using simple gradient descent followed by projection. The communication network, among the vehicles in a cluster and the repeater, can be modelled using an undirected graph, $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, consisting of the set of nodes (vehicles) $\mathscr{N} = \{1, ..., n\}$ connected by a set of edges $\mathscr{E}$, where each edge $\{i, j\}$ is an ordered pair of distinct nodes. The graph is assumed to be static and connected throughout computation of a single schedule. Let $\mathscr{N}_i$ denote the set of neighbours of node $i$. $\mathbf{W}$ is a doubly stochastic weight matrix. $\mathbf{W}_{i,j} > 0$, if $\{i, j\} \in \mathscr{E}$, meaning $i, j$ are neighbours; $\mathbf{W}_{i,j} = 0$, otherwise. Thus, each vehicle in both approaching clusters solves for the following set of equations.

$$\overline{v_i(k)} = \sum_{j=1}^{N} \mathbf{W}_{i,j} v_i(k-1) \tag{2}$$

$$v_i(k) = P_{X_i}[\overline{v_i(k)} + \alpha \nabla f_i(\overline{v_i(k)})] \tag{3}$$

where $P_{X_i}$ is the local projection operator at agent $i$, which projects the velocity calculated to the feasible region defined by the constraints in Problem $P^s$ (in this specific case, it is a bound on velocity). Information about the constraints is stored and exchanged through the RSU. In order to solve Problem $P^s$ using an hybrid architecture, where the computation is distributed among vehicles and the RSU, we decompose the problem into master-slave sub-problems. The Lagrangian of the Problem $P^s$ can be written as

$$\begin{aligned} L(v_1, v_2, \boldsymbol{\lambda}) = & \sum_{i \in C_1} f_i(v_1) + \sum_{j \in C_2} f_j(v_2) + T(\rho)(v_1 + v_2) \\ & - \lambda_1 \big(v_1 - min(Kd, v_{max})\big) \\ & - \lambda_2 \big(v_2 - \frac{d}{d + l_1} v_1 - \frac{\delta d}{(d + l_1)^2} v_1^2\big) \\ & + \lambda_3 v_1 + \lambda_4 v_2 - \lambda_5 (v_2 - v_{max}) \end{aligned}$$

where, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5]$ We can apply dual decomposition techniques [27] to decompose the problem in to sub-problems. For a given $\lambda_1$ and $\lambda_2$, vehicle clusters solve their corresponding sub-problem

$$\begin{aligned} \hat{v_1} = \underset{v_1}{\text{argmax}} \quad & \sum_{i \in C_1} f_i(v_1) + T(\rho)(v_1) - (\lambda_1 - \lambda_3) v_1 + \\ & \lambda_2 \frac{d}{d + l_1} v_1 - \lambda_2 \frac{\delta d}{(d + l_1)^2} v_1^2 \end{aligned} \tag{4}$$

and

$$\hat{v_2} = \underset{v_2}{\text{argmax}} \quad \sum_{j \in C_2} f_j(v_2) + T(\rho)(v_2) - (\lambda_2 - \lambda_4 + \lambda_5) v_2$$

In each iteration, any vehicle in each subgroup updates the $c_i$ and $a_i$ of its utility function ( $c_i' = c_i - \lambda_1 + \lambda_2 \frac{d}{d+l_1}$, $a_i' = a_i - \lambda_2 \frac{d\delta}{(d+l_1)^2}$ for cluster $C_1$ and $c_i' = c_i - \lambda_2$ for cluster $C_2$) according to the update from the master. Thus, each sub-problem needs to minimise the sum of quadratic terms. These problems are called unconstrained quadratic NUM problems and their analysis has been done in Section IV.

The master dual problem that will be solved by RSUs is

$$\begin{aligned} \underset{\boldsymbol{\lambda} \geq 0}{\text{minimise}} \; & \lambda_3 v_1 + \lambda_4 v_2 - \lambda_1 (v_1 - min(Kd, v_{max})) - \\ & - \lambda_2 \big(v_2 - min(\frac{d}{d + l_1} v_1 - \frac{\delta d}{(d + l_1)^2} v_1^2, v_{max})\big) \\ & - \lambda_5 (v_2 - v_{max}) \end{aligned} \tag{5}$$

As mentioned in [27], the master dual problem could use a gradient based method, and update $\lambda_1$ and $\lambda_2$ in each iteration as

$$\begin{aligned} & \hat{\lambda_1} = \Big[\lambda_1 + \alpha_1 \big(\hat{v_1} - min(Kd, v_{max})\big)\Big]^+, \hat{\lambda_3} = \Big[\lambda_3 - \alpha_1 \hat{v_1}\Big]^+ \\ & \hat{\lambda_2} = \Big[\lambda_2 + \alpha_1 \big(\hat{v_2} - min(\frac{d}{d + l_1} v_1 - \frac{\delta d}{(d + l_1)^2} v_1^2, v_{max})\big)\Big]^+ \\ & \hat{\lambda_4} = \Big[\lambda_4 - \alpha_1 \hat{v_2}\Big]^+, \hat{\lambda_5} = \Big[\lambda_5 + \alpha_1 (\hat{v_2} - v_{max})\Big]^+ \end{aligned} \tag{6}$$

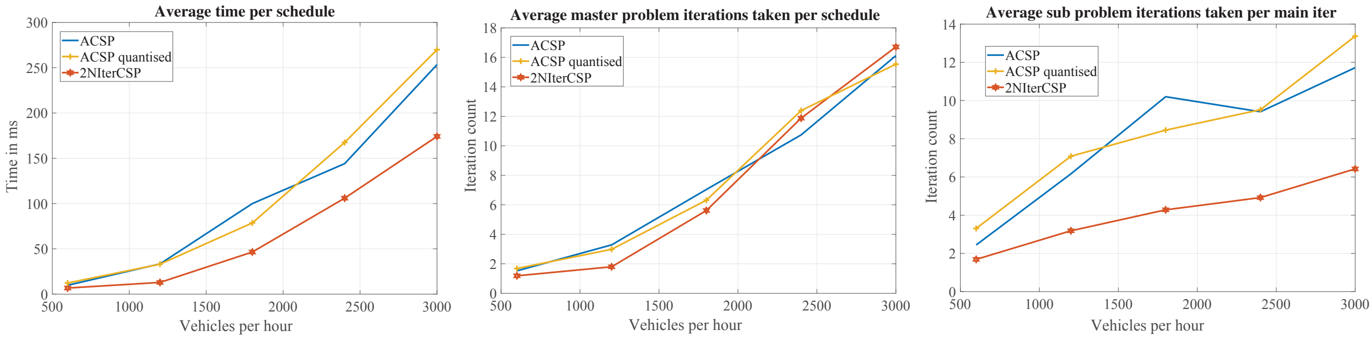
where $\alpha_1 > 0$ is a sufficiently small positive step-size, and $[.]^+$ denotes the projection to non-negative orthant.

### C. Delay and Accuracy Constraints

Theoretically, if the traffic is heavy enough, an efficient intersection should always be occupied, i.e., $ETA(nextcluster) = ETD(currentcluster)$. But in practice, a minimum headway, $\delta$, should always be kept between crossing clusters on safety grounds. This time interval provides the system tolerance towards slight calculation errors in schedule velocities and deviations of clusters from the calculated schedule at the cost of reduced intersection efficiency. Since the vehicles must travel with a controlled velocity within the control zone, it is preferred to keep the control zone as small as possible. The computation of cluster velocities should be finished with the desired accuracy, by the time the vehicles reach the buffer zone. At least one of the cluster agents will be able to communicate with the intersection manager from a distance of $d' > d$ from the intersection and initiate distributed convoy formation application. Then, they need to collaboratively decide on the convoy velocity by the time they travel $d' - d$ distance. In the best possible case, $d'$ can be the distance to the previous intersection. But considering the unpredictability in traffic flow, $d' - d$ cannot be made quite large. On the other hand, the more time agents get to compute their cluster velocity, the more accurate it can be. To increase the system efficiency, $\delta$ value need to be reduced. Therefore, the system designer has to carefully consider delay-accuracy trade-off while designing a practical system.

## IV. ANALYTICAL TOOLS

Convex NUM problems are a well-studied problem and its convergence to a near optimal solution under constant step-size gradient-descent algorithm has been stated in Theorem 2 in [28]. The authors analyse the delay-accuracy trade-off for a quadratic NUM problem solved using consensus based, constant step-size, gradient-descent algorithm and studies the effect of quantisation on convergence rate of the algorithm. The performance limits, under constrained channel conditions, derived in [12] are used for choosing proper buffer zone distance in our simulations. In addition, they also proposed a novel finite-time distributed algorithm which could reach any desired accuracy within $2N$



(a) Average Time taken per vehicle cluster schedule.

(b) Average number of iterations taken per schedule for varying vehicle densities.

(c) Average Sub iterations per master iteration.

Fig. 2. Convergence properties of our algorithms

iterations, where $N$ is the number of agents in the system. We have applied the finite-time and asymptotic optimisation approach in the hierarchical and distributed network approach to see the difference. The finite-time algorithm proves to be a really handy tool for the system designers, as it could directly influence the delay in decision making by adjusting the number of vehicles permitted in each cluster.

## V. System Performance and Results

We implement 10 simulations of 15-minute duration for each parameter set. RSU's communication range, $d'$, is fixed to $200m$ in our simulations. The buffer zone, $d$, mentioned in section III starts at $50m$ from the intersection. A maximum of 8 vehicles are allowed in a platoon and length of a platoon is capped to $150m$. A buffer zone of 50 $m$ from the traffic light is set and the centralised solver is initiated in-case, the asymptotically converging algorithm fails to calculate the cluster schedule before reaching the buffer zone. Total communication failure is not factored in here as we assume purely digital channel, where any impairment is transformed into reduction in bit-rate. However, in such cases vehicles may switch to manual control and other automated cars may identify it as a non-communicable vehicle and take necessary caution. Additionally, the intersection management may switch back to conventional algorithms. The results obtained are broadly divided into two subsections. The first set of results compares the convergence properties of the asymptotically converging algorithms (both quantised and unquantised) with finite-time converging algorithm, when implemented over the hierarchical network architecture. Average time taken to compute one traffic schedule, average master problem iterations per schedule and average sub problem iterations per master iterations are used for the comparison of our algorithms. The second set of results shows the real-world impact of implementing our algorithm. We compare the intersection management application developed using our algorithms, to traditional static traffic lights and state-of-the-art SOTL schemes. Average trip waiting time, average traffic velocity and excess fuel consumed are the parameters used for comparing intersection management applications

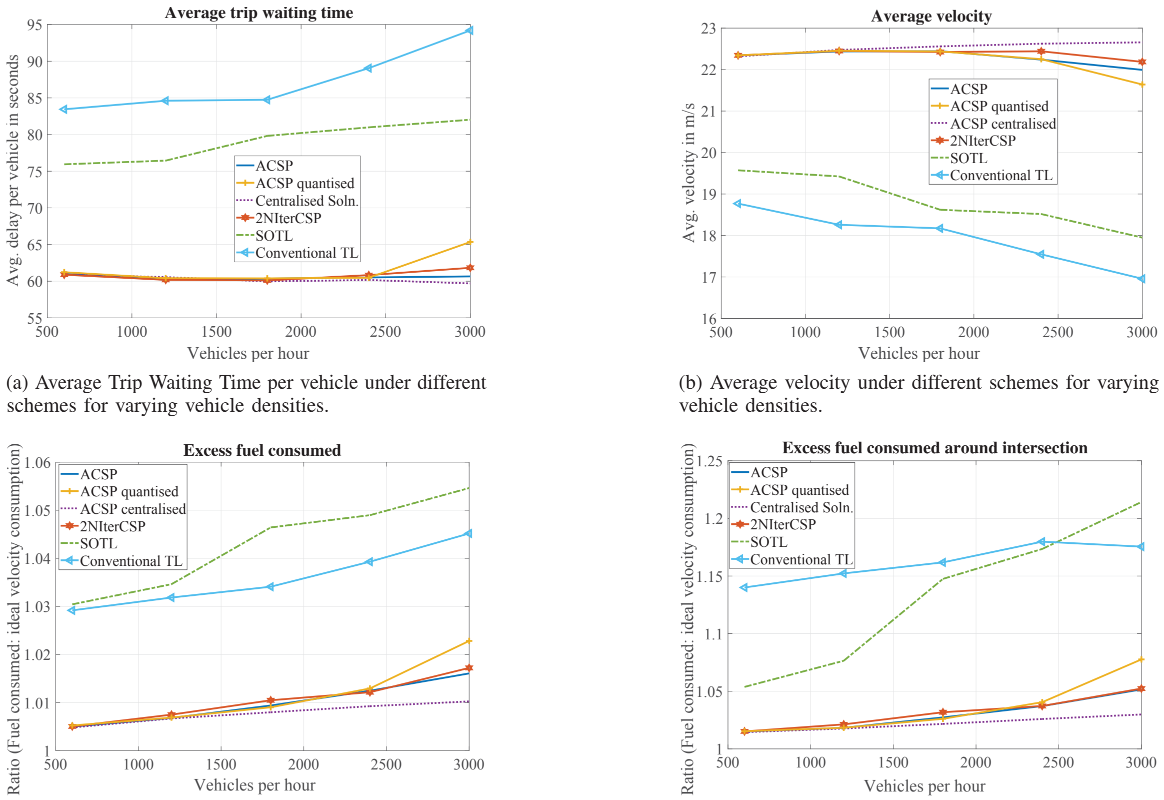
### A. Convergence Properties of Our Distributed Algorithms

We first compare convergence properties of various optimisation algorithms that have been implemented over the hierarchical network architecture. The optimisation problem was decomposed to master and slave problems (Equation (6) and (4)) and is solved by RSUs and vehicle clusters, respectively. In all our algorithms, the master problem is solved using asymptotically converging diminishing step-size gradient descent algorithm, they differ in sub-problem implementation as explained below. The parameters of the problem are chosen such that all algorithms have provided comparable performance at master level.

1) ACSP (*asymptotic converging sub-problem*): Fixed step-size, asymptotically converging, consensus based optimisation algorithm for solving the sub-problems.
2) ACSP quantised (*asymptotic converging sub-problem* quantised): Fixed step-size, asymptotically converging, consensus based optimisation algorithm under quantised message passing for solving the sub-problems. We implemented a zoom-in quantisation scheme as in [29], which is optimum achieving, i.e., the scheme eventually reaches the solution obtained by unquantised version of the algorithm.
3) Centralised Soln.: The problem is solved centrally at the RSU, assuming star network architecture as explained in Subsection III-A. All vehicles communicate their objective function parameters to the RSU, which will compute the optimum schedule and communicate back to the clusters.
4) 2NIterCSP (*2N iteration converging sub-problem*): Fixed step-size, consensus based optimisation algorithm which could find the optimum solution in $2N$ iterations, for solving the sub-problems.

The performance of various algorithms is compared using average time taken to compute one schedule as illustrated in Figure 2a (only communication delays are considered).

It can be seen from Figure 2a that as traffic intensity increases, average time taken to compute a schedule increases. The key reason behind this is the increase of vehicle cluster size, which almost linearly increases from 1.5 vehicles/cluster at 600 vehicles/hour to 3.8 vehicles/cluster at 3000 vehicles/hour, and hence, increasing the average number of sub-iteration required



(a) Average Trip Waiting Time per vehicle under different schemes for varying vehicle densities.

(b) Average velocity under different schemes for varying vehicle densities.

(c) The ratio of the fuel consumed to the ideal fuel consumption.

(d) The ratio of the fuel consumed around intersection to the ideal consumption.

Fig. 3. Performance of the algorithms

per main iteration. Further, if the number of elements in a cluster is more than 2, the sub-problem output obtained will be only near accurate in case of asymptotically converging algorithms. Thus, they require more master iterations per schedule to reach a feasible schedule. The 2NIterCSP algorithm calculates the optimum solution for the sub-problem accurately in $2N$ iterations, and it could find a feasible schedule in less number of master iterations. Thus, the time taken to reach an optimum solution will not increase drastically for the finite-time algorithm. These observations are illustrated in Figures 2b and 2c. Thus, even though all our algorithms perform equally well for low traffic intensities, the 2NIterCSP, outperforms them as the traffic intensity increases.

### B. Performance of the Algorithms

We compare the performance of our algorithms, explained in Subsection V-A, with conventional fixed switching traffic lights and *self organising traffic lights* (SOTL) [30] schemes. Under traditional fixed switching scheme, traffic lights are programmed to change phase at predefined intervals irrespective of the traffic condition. SOTL programs each traffic light, independently, based on input from induction sensors installed before the intersection, then minimal computation is required at the traffic lights. Like in SOTL, our algorithm also does not require any communication among traffic lights, but the decision on the phase switching schedule of each traffic light is made collectively by the traffic light and the vehicles approaching it. Our algorithms distribute the computation load among vehicles and the traffic light. Centralised solution of our problem highlights the best possible performance by our algorithms. The following matrices are used for comparison of real-world performance of the algorithms.

- *Average trip waiting time* (ATWT): Trip waiting time for one car is defined as the travel time minus minimum possible travel time. ATWT is the waiting time averaged over all vehicles in the system [30].
- Average Velocity of the traffic flow
- Ratio of fuel consumed to ideal fuel consumption: Fuel consumption of a vehicle can be linked to its velocity [25]. We use the ratio of the fuel consumed under the scheme to the least possible fuel consumption (assuming cars could travel at their preferred velocity).
- Ratio of fuel consumed around intersection: This is the ratio of fuel consumed within a radius of 200m from the intersection and the ideal fuel consumption in that region.

Figures 3a and 3b compares the average delay (in terms of waiting time) per vehicle and the average traffic velocity, respectively, under various schemes for varying traffic densities. It is evident from the figures that the efficiency of the traffic can be increased considerably by the active participation of vehicles and traffic lights in the decision-making process. But this requires, *vehicle to vehicle* (V2V) and *Vehicle to infrastructure* (V2I) communications and immense amount of processing power. Our algorithm distributes the data processing load among vehicles and the traffic lights. However, it is to be noticed that, under high traffic demand, the asymptotically converging version of our algorithm fails. This is because of the inaccuracy permitted in the calculation of optimum solution as explained in Subsection VI-A. Whereas, the finite-time version of our algorithm could compute the solution with high precision and could also function well even at high traffic intensities.

Figures 3c and 3d measures the performance of algorithms from an environmental perspective. Figure 3c measures the average fuel consumption of the vehicles throughout their journey, whereas Figure 3d provides the information about the fuel consumed by vehicles at an 100m distance from the intersection. In both the figures fuel consumption is expressed as a ratio to the fuel consumed if the vehicles were to navigate at their most fuel efficient velocity. The fixed iteration scheme proves to be best in terms of fuel efficiency. Further, a slight increase in fuel consumption can be observed with the introduction of quantisation, due to sub-optimal scheduling of the vehicle clusters.

Obviously, our algorithms perform better in all the four matrices. Even-though the performance of asymptotically converging algorithm degrades at high traffic intensities, its finite-time counterpart performs better throughout the given range.

## VI. DISCUSSION

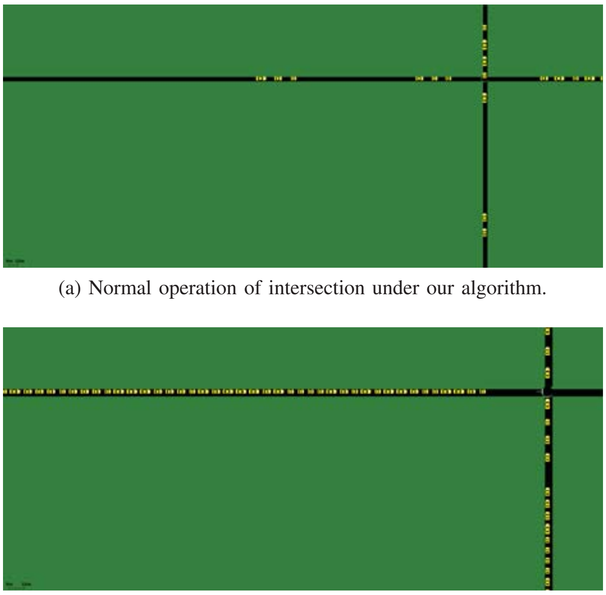
### A. Delay-Accuracy Trade-Off and Performance

For any generalised optimisation application, designed using our algorithm, delay-accuracy trade-off can be achieved by adjusting the master/sub-problem step-sizes. The practical implications of such a trade-off vary from one applicaiton to another. For instance, in the current setup, delay-accuracy trade-off in the optimisation problem is translated to adjustments in calculation zone length and yellow-light duration. A longer calculation zone provides more time for computing the solution, improving its accuracy, by which the safety gap between the crossing schedules - the yellow light timing - can be reduced.

An acceptable guideline for optimum delay-accuracy trade-off varies from one problem to another. For a traffic system, any feasible solution should ensure robustness over a range of traffic intensities. It should be noted that we introduced the parameter $T(\rho)$ in the objective function. Increase of $T(\rho)$ implies adding more weight to intersection efficiency over fuel efficiency. Since our primary objective is to find the most eco-friendly solution, the best solution ensures smooth traffic flow across the intersection with least possible value of $T(\rho)$.

In our simulations, we used the step-sizes and the buffer-zone length, which stabilised the traffic flow over the desired traffic injection ratios, found via trial and error. $T(\rho)$ is increased exponentially with respect to traffic intensity, and yellow light duration is reduced linearly with increase in traffic intensity, in order to ensure accepable system performance at high traffic intensities. However, as explained in the following paragraph, the combination seems to fail at extremely high traffic intensities for the asymptotically converging algorithm.

Asymptotically converging version of our algorithm proves to be equally good as the centralised one at low traffic intensities and a degraded performance at higher traffic intensities. The reason for this degradation is the delay-accuracy trade-off in the optimisation problem. The inaccuracy in calculation leads to a larger gap between the crossing clusters. For instance, say the yellow phase duration is 500ms and due to the inaccuracy in computing the crossing clusters leave a gap of $600ms$ between them. The intersection will thus be engaged for additional $100ms$ and upcoming cars has to be delayed by $100ms$. At high traffic intensities, it is observed that the system cannot adjust to this slow down. Therefore, the whole traffic gets choked at the intersection owing to cascaded slowing downs. This phenomenon is depicted in Figure 4b. On the other hand, trying to go for more accurate solution needs more iterations there by increasing the communication delays. Hence, finding optimum delay-accuracy trade-off for the optimisation problem is critical for any real-time application [2]. Quantisation amplifies the above mentioned effect by further degrading the quality of the solutions obtained.



(a) Normal operation of intersection under our algorithm.

(b) Long queues before intersection: this occurs due to the delay-accuracy trade-off taken by the algorithm.

Fig. 4. Screen shot of the implementation

### B. SOTL From Environmental Perspective

Authors of [30] only compared SOTL algorithm based on the average delay incurred. From the figures, it is clear that SOTL has the worst fuel consumption over all. A reason for this may be that SOTL forces the vehicles to cluster at the intersection by stopping them for a while and they remain in the cluster for the rest of their journey (as there is no turns or overtaking is allowed) and each individual vehicle has no control over the cluster velocity.

### C. Effect of Quantisation

It has been noted that, in limited number of cases, the asymptotically converging algorithm takes exceptionally large number of iterations (in the range of $100s$, compared to usual $< 10$) to reach the optimal solution, due to the oscillatory behaviour of the fixed step-size gradient descent algorithm. The quantiser prevents this oscillatory behaviour to an extend by limiting the maximum deviation possible in each step-size. Hence, as observed from the results, quantisation is marginally increasing the performance of asymptotically converging gradient-descent algorithm at low traffic intensities. As the traffic intensity increases, the accuracy of the solution becomes more important and the performance of the algorithm degrades on quantisation.

### D. Scalability and Applicability

The applications we have developed are for real-time intersection management, with projected heavy penetration of automated vehicles. Since the algorithms considered do not need inputs from other intersections, they can be scaled and implemented independently under a multiple intersection scenario. The current simulations are set up for a single lane traffic intersection. As evident from our results, single lane intersection itself becomes quite challenging at high traffic intensities, considering the real-time nature. Currently, we are extending the application to a more complex scenario with two-way traffic and multiple lanes. Additionally, extending the algorithm by factoring in inputs from other intersections will lead to a more optimal intersection management solution. From an algorithmic perspective, we are developing finite-time distributed optimisation for more complex and constrained optimisation problems.

## VII. Conclusion

A smart traffic intersection management application for automated vehicles is implemented, with superior performance than both conventional fixed switching and state of the art (SOTL) algorithm. The system is used to study the effect of delay-accuracy trade-off in real-time applications operating under a distributed optimisation algorithm. It is observed that, due to the delay-accuracy trade-off taken, asymptotically converging algorithm fails on high traffic intensities, whereas the finite-time algorithm that gives highly accurate results in $2N$ iterations performs better at high vehicle densities. This highlights the importance of the trade-off between delay and accuracy for distributed optimisation problems in real-time IoT systems. Counter-intuitively, it is seen that the system performance slightly increases with an information constrained channel. Therefore, we recommend more detailed studies to identify perfect quantisation level for a distributed optimisation problem. The application studied can be put into a real-life scenario by extending to more complex scenarios with multiple lanes and turns. Additionally, developing finite-time distributed algorithms for more generalised constrained problems can be considered as an entirely novel research direction in distributed optimisation.

## Acknowledgement

This work was supported in part by the Australian Research Council Discovery Project under Grant DP140100819.

## References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems-the International Journal of Grid Computing and Escience*, vol. 29, no. 7, pp. 1645–1660, 2013. [Online]. Available: ⟨GotoISI⟩://WOS:000320635700001

[2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[3] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He, and Q. Yang, "Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150, 2017.

[4] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[5] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, "Revisiting street intersections using slot-based systems," *PLoS ONE*, vol. 11, no. 3, p. e0149607, 2016.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] A. Nedic, *Convergence Rate of Distributed Averaging Dynamics and Optimization in Networks*. Now Publishers Inc., 2015.

[8] I. o. T. S. DLR. (2017) Sumo: Simulation of urban mobility of work. [Online]. Available: dlr.de/ts/sumo

[9] M. El Chamie, J. Liu, and T. Başar, "Design and analysis of distributed averaging with quantized communication," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3870–3884, 2016.

[10] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 380–392, 2014.

[11] E. Nekouei, G. Nair, and T. Alpcan, "Convergence analysis of quantized primal-dual algorithm in quadratic network utility maximization problems," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, Conference Proceedings, pp. 2655–2660.

[12] B. V. Philip, T. Alpcan, J. Jin, and M. Palaniswami, "Information constrained and finite-time distributed optimisation algorithms," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, Conference Proceedings, pp. 4588–4593.

[13] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[14] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.

[15] S. Ilgin Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, no. Supplement C, pp. 121–131, 2014.

[16] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996–1014, 2018.

[17] A. Vinel, "3gpp lte versus ieee 802.11p/wave: Which technology is able to support cooperative vehicular safety applications?" *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 125–128, 2012.

[18] A. Abdelgader and W. Lenan, "The physical layer of the ieee 802.11 p wave communication standard: the specifications and challenges," in *Proceedings of the world congress on engineering and computer science*, vol. 2, 2014, Conference Proceedings, p. 71.

[19] F. J. Ros, J. A. Martinez, and P. M. Ruiz, "A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools," *Computer Communications*, vol. 43, pp. 1–15, 2014.

[20] L. f. C. A. EPFL. (2017) Traffic and network simulation environment of work. [Online]. Available: http://lca.epfl.ch/projects/trans/

[21] S. C. Ng, W. Zhang, Y. Zhang, Y. Yang, and G. Mao, "Analysis of access and connectivity probabilities in vehicular relay networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 140–150, 2011.

[22] J. E. Andres Acosta, Jorge Espinosa. (2017) Traci4matlab: User's manual of work. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2877217, IEEE Transactions on Industrial Informatics

[23] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2377–2396, 2015.

[24] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[25] P. Dilara and G. Fontaras, *Parameterisation of fuel consumption and CO2 emissions of passenger cars and light commercial vehicles for modelling purposes*. Publications Office, 2011.

[26] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[27] D. P. Palomar and C. Mung, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[28] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

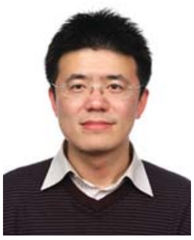
[29] E. Nekouei, T. Alpcan, G. Nair, and R. Evans, "Convergence analysis of quantized primal-dual algorithms in network utility maximization problems," *IEEE Transactions on Control of Network Systems*, 2016.

[30] S.-B. Cools, C. Gershenson, and B. D'Hooghe, *Self-organizing traffic lights: A realistic simulation*. Springer, 2013, pp. 45–55.

**Bigi Varghese Philip** (S'16) received the Bachelor of Technology (B.Tech) in Electronics and Communication Engineering from TKM college of Engineering, University of Kerala, India, in 2011 and the Master of Technology (M.Tech) in Telecommunication Systems Engineering from the Indian Institute of Technology, Kharagpur, India in 2013. He has worked as an Engineer in Special Task Force - Smart Grid (STF-SG), Power Grid Corporation of India Ltd. for two years (2013-15).

Currently, he is pursuing the Ph.D. degree from Department of Electrical and Electronic Engineering, the University of Melbourne, Melbourne, Australia. His research interests include Internet of Things, IoT applications in cyber-physical systems such as smart traffic and smart grid, distributed optimisation algorithms and edge/fog computing.

**Tansu Alpcan** (S'98-M'07-SM'13) received his B.S. degree in electrical engineering from Bogazici University, Istanbul, Turkey in 1998. He received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from University of Illinois at UrbanaChampaign in 2001 and 2006, respectively. His research focuses on applications of distributed decision making, learning, game theory, optimization, and control to various security and resource allocation problems in networked and energy systems. He is recipient of multiple research and best paper awards from UIUC and IEEE. He has played a role in organization of several workshops and conferences such as IEEE Infocom, ICC, GameComm, and GameSec as TPC member, associate editor, co-chair, chair, and steering board member. He is the (co-)author of more than 150 journal and conference articles, two edited volumes, as well as the book Network Security: A Decision and Game Theoretic Approach published by the Cambridge University Press in 2011. He has worked as a Senior Research Scientist in Deutsche Telekom Laboratories, Berlin, Germany, between 2006- 2009, and as Assistant Professor in Technical University of Berlin from 2009 until 2011. He is currently with the Department of Electrical and Electronic Engineering at the University of Melbourne as Associate Professor and Reader.



**Jiong Jin** (M'11) received the B.E. degree with First Class Honours in Computer Engineering from Nanyang Technological University, Singapore, in 2006, and the Ph.D. degree in Electrical and Electronic Engineering from the University of Melbourne, Australia, in 2011. He is currently a Senior Lecturer in the School of Software and Electrical Engineering, Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, Australia. Prior to it, he was a Research Fellow in the Department of Electrical and Electronic Engineering at the University of Melbourne from 2011 to 2013. His research interests include network design and optimization, fog and edge computing, robotics and automation, Internet of Things and cyber-physical systems as well as their applications in smart manufacturing, smart transportation and smart cities.

**Marimuthu Palaniswami** (F'12) received the M.E. degree in electrical, electronic and control engineering from the Indian Institute of Science, Bengaluru, India, the M.Eng.Sc. degree in electrical, electronic and control engineering from the University of Melbourne, Melbourne, VIC, Australia, and the Ph.D. degree from the University of Newcastle, N.S.W., Australia. Presently, he is a Professor of Electrical Engineering and Director/Convener of a large ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) with about 100 researchers on various interdisciplinary projects.

He was given a Foreign Specialist Award by the Ministry of Education, Japan in recognition of his contributions to the field of Machine Learning and communications. He received University of Melbourne Knowledge Transfer Excellence Award and Commendation Awards. He served as associate editor for Journals/transactions including IEEE Transactions on Neural Networks, Computational Intelligence for Finance. He is editor of Journal of Medical, Biological Engineering and Computing and the Subject Editor for International Journal on Distributed Sensor Networks.

He is representing Australia as a core partner in EU FP7 projects such as SENSEI, SmartSantander, Internet of Things Initiative, and SocIoTal. He has published over 480 refereed research papers and leads one of the largest funded ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing. His research interests include Smart Sensors and Sensor Networks, Machine Learning, IoT and Biomedical Engineering and Control.