

Dynamic Traffic Control with Fairness and Throughput Optimization Using Vehicular Communications

Lien-Wu Chen, *Member, IEEE*, Pranay Sharma, and Yu-Chee Tseng, *Fellow, IEEE*

Abstract—Traffic congestion in modern cities seriously affects our living quality and environments. Inefficient traffic management leads to fuel wastage in volume of billion gallons per year. In this paper, we propose a dynamic traffic control framework using vehicular communications and fine-grained information, such as turning intentions and lane positions of vehicles, to maximize traffic flows and provide fairness among traffic flows. With vehicular communications, the traffic controller at an intersection can collect all fine-grained information before vehicles pass the intersection. Our proposed signal scheduling algorithm considers the flows at all lanes, allocates more durations of green signs to those flows with higher passing rates, and also gives turns to those with lower passing rates for fairness provision. Simulation results show that the proposed framework outperforms existing works by significantly increasing the number of vehicles passing an intersection while keeping average waiting time low for vehicles on non-arterial roads. In addition, we discuss our implementation of an Zigbee-based prototype and experiences.

Index Terms—Dynamic traffic control, environmental protection, intelligent transportation, traffic management, vehicular communications.

I. INTRODUCTION

TRAFFIC congestion in modern cities seriously affects our living quality and environments. Vehicles on the roads produce mass air pollutions that emit greenhouse gases such as carbon dioxide, hydrocarbons, and nitrogen oxides. Idling vehicles caused by traffic jams and red signs at intersections waste a large amount of fuel and seriously pollute the air. Studies show that about 30% of man-made dioxide emissions is from transportation systems [1]. In particular, inefficient traffic management leads to fuel wastage of billion gallons per year [2]. Furthermore, badly designed traffic signals produce

frequent disruptions to traffic flows and increase delays [3]. Previous efforts have been made to increase the traffic flows in urban arterial roads [4], to reduce the waiting time at intersections [5], and to navigate vehicles in congested roads [6].

Traditional traffic controls employ fixed signal durations and thus cannot adapt to real-time traffic conditions. Adaptive traffic controls [7] rely on collecting real-time traffic information by dedicated detectors, such as inductive loops [8], magnetic sensors [9], and video cameras [10]. Green-sign durations and phase orders are then varied based on traffic conditions. New approaches for collecting traffic information may involve wireless sensor networks (WSNs) [11], [12], RFIDs [13]–[17], or vehicular communications (VCs) [18], [19]. The WSN-based systems rely on a lot of roadside sensors. The RFID-based systems also require to install lots of RFID readers on road segments, incurring huge infrastructure cost. The VC-based systems need a GPS receiver with an onboard unit (OBU) in each vehicle. Through OBUs, vehicles can communicate with each other and roadside units via vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications, respectively. V2V and V2R communications can facilitate the exchange of real-time traffic conditions and enable drivers/traffic controllers to make better decisions. V2R communications can provide information covering large regions in urban areas, whereas V2V communications enable direct exchange of local information among vehicles, especially in sub-urban and rural areas without roadside infrastructures [20].

In this work, we consider the VC-based approach. Our work is motivated by [18], [19], which use the longest-queue-first approach with maximal weight matching (LQF-MWM) to minimize queue lengths and provide quality-of-service (QoS) to higher-priority vehicles, such as ambulances and police cars. However, there are several drawbacks in [18], [19]. First, the traffic flows with longer queues may have lower passing rates than those with shorter queues. A more sophisticated and realistic model should be considered. Second, the LQF approach may cause starvation to shorter queues. Third, some lanes may have mixture of straight-going and right/left-turning vehicles that require more accurate estimation on their passing rates, which is not addressed in LQF-MWM.

Through vehicular communications, we try to utilize turning intentions and lane positions of vehicles to maximize traffic flows and provide fairness among traffic flows. We assume that the turning intention of a vehicle can be given by its

Manuscript received March 1, 2012; revised July 14, 2012.

L.-W. Chen is with the Department of Information Engineering and Computer Science, Feng Chia University No. 100, Wenhwa Road, Seatwen, Taichung, 407, Taiwan (e-mail: lwuchen@fcu.edu.tw).

P. Sharma is with the Department of Electrical Engineering and Computer Science, National Chiao Tung University No. 1001, Ta Hsueh Road, Hsinchu, 300, Taiwan (e-mail: me@pranays.com).

Y.-C. Tseng is with the Department of Computer Science, National Chiao Tung University No. 1001, Ta Hsueh Road, Hsinchu, 300, Taiwan (e-mail: yctseng@cs.nctu.edu.tw).

This research was supported in part by the NSC under Grant No. 101-2221-E-035-090.

Y.-C. Tseng's research was co-sponsored by the MoE ATU Plan, by NSC grants 98-2219-E-009-019, 98-2219-E-009-005, and 99-2218-E-009-005, by AS-102-TP-A06 of Academia Sinica, by ITRI, Taiwan, and by D-Link.

This work was also conducted under the "AMI Enhancement Project" of III, which is subsidized by MoEA, ROC.

Digital Object Identifier 10.1109/JSAC.2013.SUP.0513045

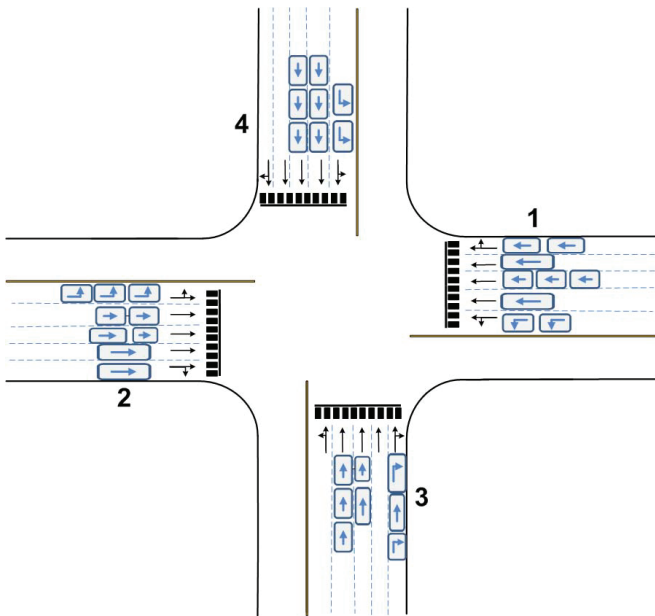


Fig. 1: The road intersection model in our work.

turning indicator or by the vehicle historical turning data. This information is collected by the traffic controllers located at road intersections. Our signal scheduling algorithm takes the demands of all lanes into consideration. Furthermore, while assigning more durations to the traffic flows with higher passing rates, our framework also gives turns to those with lower passing rates for fairness provision. We also design and implement an Zigbee-based prototype to verify the feasibility of our framework. Alternative communication technologies, such as WiFi (i.e., IEEE 802.11a/b/g/n) [21] or WAVE/DSRC (i.e., IEEE 802.11p) [22], may be adopted to collect vehicle information on the road.

The rest of the paper is organized as follows. Section II presents our system model. Section III presents our framework and solution. Simulation results are presented in Section IV. Section V shows our prototyping results. Section VI concludes the paper and points the direction for future work.

II. SYSTEM DESIGN

We consider an intersection consisting of four road segments each with multiple lanes, as shown in Fig. 1. The left-most lane is for both left-turn and go-straight vehicles, the right-most lane is for both right-turn and go-straight vehicles, and the remaining lanes (if any) are only for go-straight vehicles. The intersection has a traffic controller with a network interface to communicate with vehicles and a microprocessor to control its traffic signs. Each vehicle has a network interface, a GPS receiver, and an On Board Diagnostics (OBD) interface [23]. The network interface can communicate with the intersection controller and the GPS receiver can calculate locations. The OBD interface can get the vehicle's turning intention, speed, and acceleration. As a vehicle stops at an intersection due to red signs, it will register itself and transmit its vehicle information, such as acceleration, position, turning intention, and driver reaction time, to the intersection controller. As it passes the intersection, it will

deregister itself with the intersection controller. Thus, the intersection controller can collect the information of all vehicles waiting at the intersection and dynamically control its traffic signs based on the collected traffic information.

The *dynamic traffic control problem* is defined as follows. In an intersection, each road segment has traffic flows consisting of right-turn, go-straight, and left-turn vehicles. For each road segment, we define four types of *movements*: *RIGHT_ONLY*, *STRAIGHT_RIGHT*, *LEFT_ONLY*, and *ALL_THROUGH*. The non-conflicting movements of multiple road segments can be combined together to form a *phase* such that lanes belonging to these movements are assigned green signs simultaneously. The goal is to select the next best phase and decide its duration for maximizing the traffic flows passing the intersection with fairness in mind by addressing the following issues:

- 1) Rate Estimation: How do we calculate the passing rate of a lane by considering the direction of the lane and the properties of the vehicles on that lane?
- 2) Phase Design: How do we select movements for road segments and combine non-conflicting movements to form a phase?
- 3) Throughput Maximization: How do we allocate green-sign durations to phases such that the long-term traffic flows of the intersection is maximized?
- 4) Fairness Provision: How do we arrange phases so that there is no starvation for lanes with lower passing rates?

III. DYNAMIC TRAFFIC CONTROL FRAMEWORK

The proposed dynamic traffic control framework is to balance throughput and fairness. To improve throughput, instead of using queue lengths as in [18], [19], we adopt the passing rates of traffic flows to maximize the number of vehicles passing an intersection. For fairness provision, the priorities of traffic flows are increased with their waiting time at the intersection. We assume that there is no U-turns allowed. Our framework can be applied to both intersections with dedicated or non-dedicated turning lanes. The passing rates of traffic flows are derived in Section III-A. We propose our phase matrices in Section III-B for choosing non-conflicting movements. The throughput maximization algorithm is proposed in Section III-C. A fairness-enhanced algorithm is presented in Section III-D.

In the following context, we consider two kinds of queues, *stationary* queue and *moving* queue. A stationary queue appears when there are n non-moving vehicles all stopped by the current red signal at an intersection I and all waiting to pass I . A moving queue appears when a green signal has been given to the aforementioned stationary queue for a while and some vehicles already pass I . Therefore, we may see that the front part of the queue is moving and the rear part of the queue (if any) is stationary. Note that this scenario may appear because our model may extend the duration of the green sign for some queues. Fig. 2 shows some examples of stationary and moving queues.

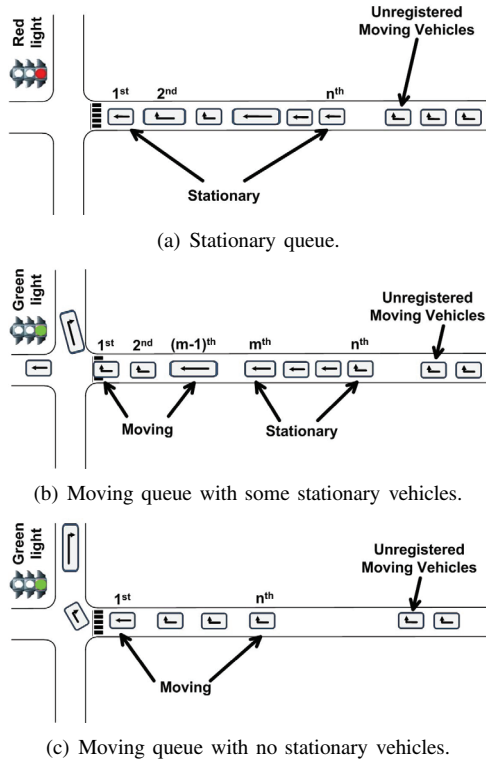


Fig. 2: Examples of stationary and moving queues.

A. Passing Rate Estimation

Given an intersection I , the passing rate r_{ij} of lane j on road segment i is calculated by:

$$r_{ij} = \frac{n}{\Delta t_{ij}}, \quad (1)$$

where n is the number of vehicles waiting to pass I and Δt_{ij} is the expected time required by the last (i.e., n -th) registered vehicle on lane j to get through I . Note that these n vehicles may form a stationary or moving queue. With vehicular communications, each stopped vehicle can register its acceleration, position, turning intention, and driver's reaction time with the traffic controller at I . Some of these parameters, if not available, can be derived from the driving habits of the car owners.

We derive a car passing model to estimate Δt_{ij} as follows. Since a vehicle's acceleration is restricted by the slowest stationary vehicle in the front of it, we denote the slowest stationary vehicle's acceleration as a_{min} (note that we ignore the accelerations of those moving vehicles, if any, for simplicity). Let v be the current velocity of the n -th vehicle and d be the distance from it to the center of I . The time Δt required by the n -th vehicle to move to the center of I must satisfy the equation:

$$d = v\Delta t + \frac{a_{min}\Delta t^2}{2}. \quad (2)$$

So we have

$$\Delta t = \frac{-v + \sqrt{v^2 + 2a_{min}d}}{a_{min}}. \quad (3)$$

Let the m -th vehicle be the first stationary vehicle in the queue and Δr_k be the driver reaction time of the k -th vehicle, $1 \leq$

$m \leq k \leq n$. The expected passing time of the n -th vehicle can be approximated by

$$\Delta t_{ij} = \frac{-v + \sqrt{v^2 + 2a_{min}d}}{a_{min}} + \sum_{k=m}^n \Delta r_k. \quad (4)$$

For examples, Fig. 2(a) is a stationary queue with $m = 1$ and $v = 0$. Fig. 2(b) is a moving queue with $m > 1$ and $v = 0$, where the $(m-1)$ vehicles are moving whereas the m -th to the last vehicles are still waiting to move. Fig. 2(c) is a moving queue with $m = \infty$ and $v > 0$, which means that all vehicles are moving. Note that in our framework, a traffic controller may give a green signal to a lane with duration T_{green} . After T_{green} , it may re-calculate the queue status and extend the green signal. The calculation can be done by determining the moving time that vehicles already received. Thus, the index m' such that $\sum_{k=m}^{m'-1} \Delta r_k < T_{green} \leq \sum_{k=m}^{m'} \Delta r_k$ will be the new value of m in the next round. The last vehicle's speed can be updated by $v' = a_{min} \times (T_{green} - \sum_{k=m}^n \Delta r_k)$ and its distance to I by $d' = d - \frac{a_{min} \times (T_{green} - \sum_{k=m}^n \Delta r_k)^2}{2}$ if $T_{green} > \sum_{k=m}^n \Delta r_k$.

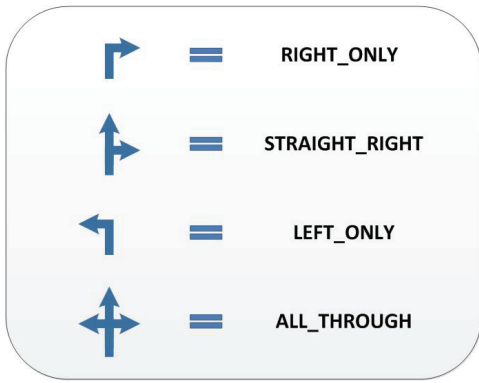
Note that this work only considers one intersection. In rush hours, the queue in the next intersection may limit the number of vehicles that can move to the next intersection. One quick fix to this problem is to only consider the first l vehicles in the current queue that may move to the next intersection by estimating the latter's queue length and road length. If l is less than the total number of registered vehicles, n , we replace n by l in the above passing rate estimation.

B. Phase Matrix

After calculating the passing rate of each lane, we need to schedule the right-of-way for lanes. For each road segment, there are four movement types: *RIGHT_ONLY* (type 1), *STRAIGHT_RIGHT* (type 2), *LEFT_ONLY* (type 3), and *ALL_THROUGH* (type 4). These movement types are shown in Fig. 3(a). A *phase matrix* is a 4×4 binary matrix indicating the permitted non-conflicting movement types for the four road segments of I , where a "1" means "permitted" and a "0" means "not permitted". For example, the first phase matrix in Fig. 3(b) means that road segments 2 and 4 can turn right and road segment 3 can go straight and turn right. Clearly, these movements have no conflict. The traffic controller has to select a phase matrix from a set of candidate phase matrices $S = \{S^1, S^2, \dots, S^N\}$ and its corresponding green-sign duration according to the current traffic condition. Note that it is easy to enumerate all phase matrices of S (for example, for the intersection in Fig. 1, there are $N = 16$ matrices as listed in APPENDIX after combing as many non-conflicting movements as possible). Below, we will discuss how to choose phase matrices for short-term throughput maximization and long-term fairness provision.

C. Throughput Maximization Algorithm

As mentioned above, each phase matrix in S represents a possible arrangement of green signs at I . Below, we will show how to choose a phase matrix such that the aggregated passing rate of all permitted lanes is maximized.



(b) Two examples of phase matrices.

		Movement type			
		1	2	3	4
Road segment	1	0	0	0	0
	2	1	0	0	0
	3	0	1	0	0
	4	1	0	0	0

		Movement type			
		1	2	3	4
Road segment	1	1	0	0	0
	2	1	0	0	0
	3	0	0	1	0
	4	0	0	1	0

Fig. 3: Movement types and phase matrices.

Consider any $S^k \in S$. The aggregated passing rate when S^k is selected can be written as

$$\hat{W}^k = \sum_{i=1}^p \sum_{j=1}^q R_{ij} \times S_{ij}^k, \quad (5)$$

where S_{ij}^k is the (i, j) -th element of matrix S^k and the size of S^k is $p \times q$. Here R_{ij} is the aggregated passing rate of road segment i given movement type j , whose value depends on the binary *flow attribute* F_{ij}^k of the k -th lane of road segment i given movement type j . F_{ij}^k is set to 1 when the turning intentions of vehicles on the k -th lane of road segment i is compliant to movement type j ; otherwise, F_{ij}^k is set to 0. Here lanes are numbered from left to right and we denote by L_i the number of lanes on road segment i . The rules to calculate F_{ij}^k are as follows. Note that these rules can be applied to road segments with different numbers of lanes and to both intersections with dedicated or non-dedicated turning lanes.

- 1) If there is no go-straight intention vehicle on the left-most lane, $F_{i,3}^1 = 1$ and $F_{i,2}^k = 1$, for $k = 2, 3, \dots, L_i$ (i.e., *LEFT_ONLY* is set to the left-most lane and *STRAIGHT_RIGHT* is set to all other lanes).
- 2) If there are only go-straight intention vehicles on the left-most lane, $F_{i,2}^k = 1$, for $k = 1, 2, \dots, L_i$ (i.e., *STRAIGHT_RIGHT* is set to all lanes).
- 3) If there are both go-straight intention and left-turn intention vehicles on the left-most lane, $F_{i,4}^k = 1$, for $k = 1, 2, \dots, L_i$ (i.e., *ALL_THROUGH* is set to all lanes).
- 4) If there is no go-straight intention vehicle on the right-most lane, $F_{i,1}^{L_i} = 1$ (i.e., *RIGHT_ONLY* is set to the right-most lane).

After assigning F_{ij}^k for all lanes, R_{ij} can be calculated by

$$R_{ij} = \sum_{k=1}^{L_i} r_{ik} \times F_{ij}^k, \quad (6)$$

where r_{ik} is the passing rate estimated by Eq. (1) for lane k of road segment i . The next phase matrix needs to be computed before the current green-sign duration expires. We select the phase matrix S_{max} with the maximum aggregated passing rate for allowing vehicles to pass in the next phase. Note that this phase matrix is not necessarily different from the current one. This means that some of the lanes that receive a green sign currently may be extended. The minimum passing time of the lane among all permitted lanes in S_{max} will be used as the next green-sign duration T_{green} . However, to avoid T_{green} being too long or too short, we will also bound its value in the range $[T_{min}, T_{max}]$. So if $T_{green} < T_{min}$, we will set it to T_{min} and if $T_{green} > T_{max}$, we will set it to T_{max} . Algorithm 1 gives the pseudocode of the throughput maximization algorithm. A phase switch may cause some overhead (for example, a phase switch from moving queues to stationary ones will incur startup overhead because all vehicles in a stationary queue have to take certain driver reaction time to start moving). Our framework can balance throughput and fairness, as shown in the next subsection.

Algorithm 1: Throughput Maximization Algorithm

Input: information of registered vehicles

Result: S_{max}, T_{green}

```

1 begin
2   for  $i \leftarrow 1$  to  $p$  do
3     for  $j \leftarrow 1$  to  $q$  do
4       for  $k \leftarrow 1$  to  $L_i$  do
5         Find flow attribute  $F_{ij}^k$ .
6       Find aggregated passing rate  $R_{ij}$ .
7     Choose  $S_{max}$  and  $T_{green}$ .
8   return  $S_{max}, T_{green}$ .
```

D. Fairness Provisioning Algorithm

The schedules generated by the Throughput Maximization Algorithm may result in a starvation problem because the phase matrix with a high aggregated passing rate may continuously get the right-of-way to pass the intersection. To avoid this starvation problem, we employ ageing timers t_{ij}^{age} and create an *ageing matrix* $[A]_{p \times q}$. Timer t_{ij}^{age} is the waiting time

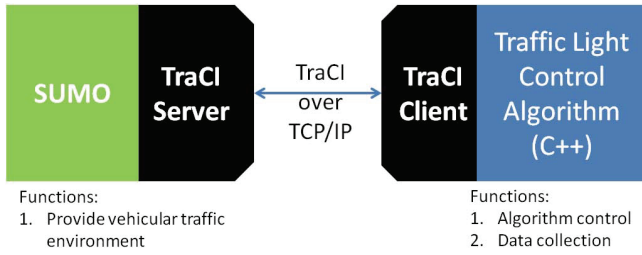


Fig. 4: TraCI interface of SUMO for external program control.

of vehicles on road segment i with movement type j . Note that if movement type 4 (*ALL_THROUGH*) gets the right-of-way, all other types are also considered getting the right-of-way. Similarly, if movement type 2 (*STRAIGHT_RIGHT*) gets the right-of-way, type 1 (*RIGHT_ONLY*) is also considered getting the right-of-way. As t_{ij}^{age} is larger than or equal to a predefined threshold T_{age} , the priority of road segment i with movement type j is increased 100 times as follows.

$$A_{ij} = \begin{cases} 100 & \text{if } t_{ij}^{age} \geq T_{age} \text{ (High Priority)} \\ 1 & \text{if } t_{ij}^{age} < T_{age} \text{ (Normal Priority)} \end{cases} \quad (7)$$

With the ageing matrix A , the aggregated passing rate for each phase matrix S^k is modified as follows.

$$\hat{W}^k = \sum_{i=1}^p \sum_{j=1}^q R_{ij} \times S_{ij}^k \times A_{ij}. \quad (8)$$

Note that once road segment i with movement type j is served in the next phase matrix, its timer t_{ij}^{age} should be reset to 0 and be restarted when there is a new vehicle arriving (this is to avoid giving the right-of-way to a movement with no vehicle).

The above aging design can help a phase matrix to raise its priority. However, it is not suitable that a phase matrix is served again before other non-served is selected to be served. We use a *phase-served matrix* $[P]_{p \times q}$ to make sure that a served phase, once being replaced by another phase, can not get the right-of-way again unless all other movement types have been served. Note that this rule is not applied when the same phase matrix is served continuously, but the aging process will choose another phase eventually. P is a binary matrix. Whenever a phase matrix $S^k \in S$ is replaced by another one, the corresponding bit in P will be masked as 0. That is, P_{ij} is set to 0 for each $S_{ij}^k = 1$. Once all elements of P are masked as 0, we will reset all elements of P to 1. With the phase-served matrix P , the aggregated passing rate for each phase matrix S^k is further modified as follows.

$$\hat{W}^k = \sum_{i=1}^p \sum_{j=1}^q R_{ij} \times S_{ij}^k \times A_{ij} \times P_{ij}. \quad (9)$$

Note that matrix P is to enforce the rotation of all movement types, but extension of green sign to the same phase matrix is still allowed. On the other hand, matrix A is to raise the priorities of those awaiting phase matrices by an aging process.

IV. PERFORMANCE EVALUATION

We simulate the proposed framework by Simulation of Urban Mobility (SUMO) [24], which is an open-source, micro-scope, multi-modal traffic simulator that can generate diverse

TABLE I: Simulation Parameters for Vehicles

Parameter Type	Car	Bus
Length	5 m	10 m
Acceleration	0.8 m/s ²	0.4 m/s ²
Deceleration	4.5 m/s ²	2 m/s ²
Maximum Speed	23 m/s	13.88 m/s

traffic demands consisting of vehicles moving through a given road network. In SUMO, each vehicle can have its own route and move individually through the road network. We generate traffic demands with a custom turning percentage and control traffic signs via the Traffic Control Interface (TraCI) protocol. An external C++ program is used to control and interact with SUMO, as shown in Fig. 4. We use the JTRROUTER tool in SUMO to develop traffic demands based on specific turning ratios of vehicles. Two types of vehicles, car and bus, are adopted and their basic parameters are summarized in Table I.

We simulate our framework with only the throughput maximization part as well as our framework with both the throughput maximization and fairness part. We compare our schemes against the traditional static control method that allocates a fixed green-sign duration of 30 seconds to each phase in turn and the LQF-MWM method [19] that allocates green sign to the phase with the longest waiting queue. An intersection consisting of four 3-lane road segments is simulated, where the left-most lane is for both left-turn and go-straight vehicles, the middle lane is only for go-straight vehicles, and the right-most lane is for both right-turn and go-straight vehicles. Each simulation time is 2000 seconds, where $T_{min} = 15$ s, $T_{max} = 150$ s, and $T_{age} = 300$ s. $N = 16$ phase matrices, as shown in APPENDIX, are used in our simulation.

Fig. 5, Fig. 6, and Fig. 7 illustrate the intersection throughput (i.e., the number of vehicles passed the intersection in 2000 seconds) under different vehicle arrival rates (i.e., the sum of arrival rates of vehicles from 12 lanes) as there are 100%, 60%, and 20% go-straight cars, respectively. The ratio of right-turn cars is the same as that of left-turn ones. We can observe that our schemes achieve the highest throughput under different combinations of vehicle arrival rates and turning ratios. This is because our schemes could allow more movements to pass through simultaneously based on turning information of vehicles. In addition, our schemes take both moving flows and stationary flows into consideration so that there is less disruption to flowing traffic. Consequentially, the traffic controller could allocate green sign to the phase with the highest total passing rate. In particular, the throughput of our approach significantly increases, while the static control and the LQF-MWM methods have much lower throughputs as the road traffic and turning ratio increase.

Fig. 8 shows the throughput comparisons under 50%, 60%, 70%, 80%, 90%, and 100% cars (the rest is buses). There are 100% go-straight vehicles and the arrival rate is set to 1.2 vehicles per second. We can observe that with 50% cars and 50% buses, the throughput improvements of our schemes are most significant. This is because our schemes take the different vehicle lengths, accelerations, maximum speeds between cars and buses into consideration to maximize the

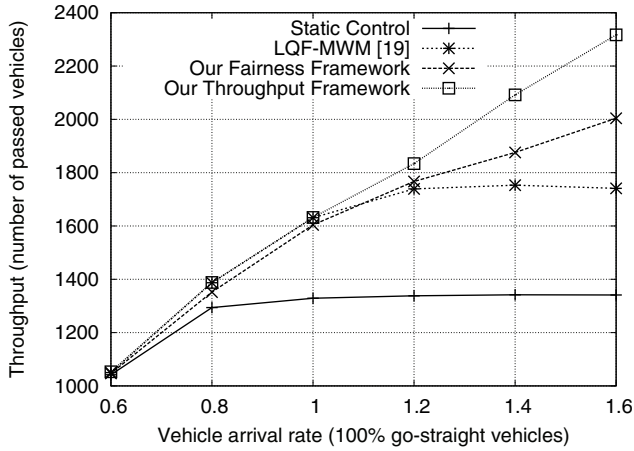


Fig. 5: Throughput comparisons under 100% go-straight, 0% right-turn and 0% left-turn cars.

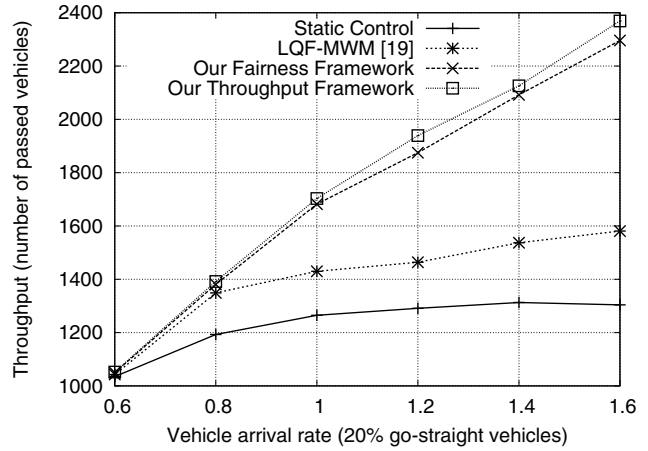


Fig. 7: Throughput comparisons under 20% go-straight, 40% right-turn and 40% left-turn cars.

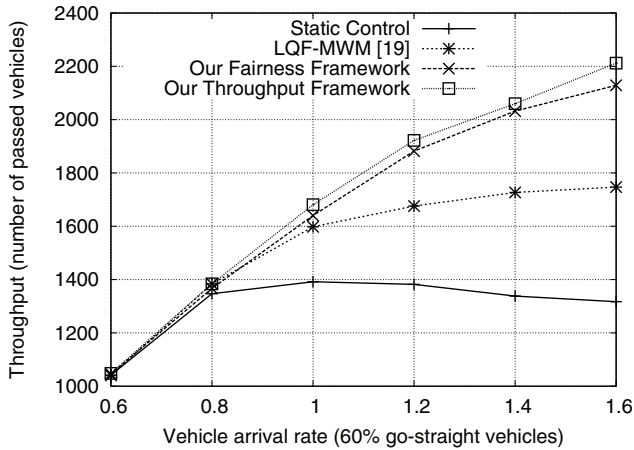


Fig. 6: Throughput comparisons under 60% go-straight, 20% right-turn and 20% left-turn cars.

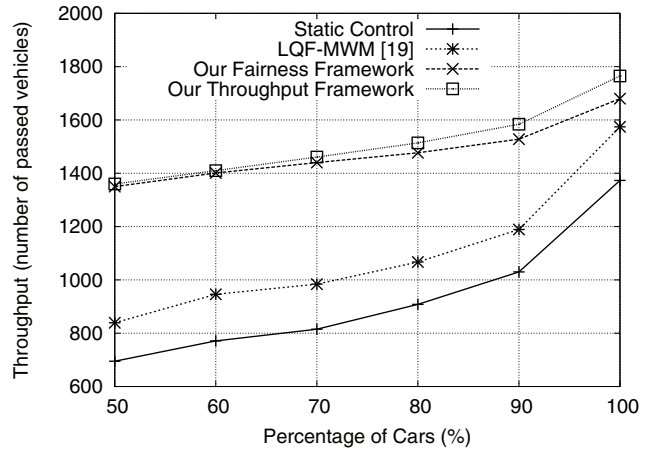


Fig. 8: Throughput comparisons under different percentages of cars.

number of passing vehicles. Furthermore, our schemes can allow non-conflicting movements as many as possible to pass the intersection concurrently.

Fig. 9 shows comparisons for the average waiting time of vehicles on non-arterial roads. The vehicle arrival rate is set to 1.2 vehicles/sec and all vehicles are go-straight cars. The arterial roads are road segments 1 and 2 whereas the non-arterial roads are road segments 3 and 4 (refer to Fig. 1). We can observe that using our throughput maximization scheme, the average waiting time on non-arterial roads increases with the increase of traffic on arterial roads. This is because the passing rates of moving flows are usually higher than those of stationary flows so that arterial roads have the higher chance to get the right-of-way continuously. Thus, non-arterial roads have to wait until their passing rates become higher than those of arterial roads through the increasing number of stopping vehicles. On the other hand, we can observe that using our fairness provisioning scheme, the average waiting time of vehicles on non-arterial roads decreases with the increase of traffic on arterial roads. This is because the served phase can not get the right-of-way again unless all other movement types at the intersection have been served. Thus, non-arterial roads

have chances to get the right-of-way even their passing rates are lower.

From these results, we conclude that the proposed approach can utilize vehicular communications to achieve better performance, leading to more efficient scheduling of traffic signs. In other words, adopting our scheme in traffic management can both avoid unnecessary stopping of vehicles, wasting time, and consuming fuel due to inefficient traffic control and prevent vehicles on non-arterial roads from starvation caused by unfair green-sign allocation.

V. PROTOTYPE IMPLEMENTATION

We design a load-based traffic sign control system called *Eco-Sign* [25], which can collect information of vehicles at intersections to dynamically control the traffic sign. Fig. 10 shows the system architecture and vehicles' states at different locations nearby an intersection. On the intersection side, it consists of a LU (location unit) on each road segment, which provides the location information to vehicles, and a TCU (traffic control unit), which decides and transmits the traffic sign timing. On the vehicle side, a VU (vehicle unit) is equipped onboard to receive the traffic sign timing from TCU

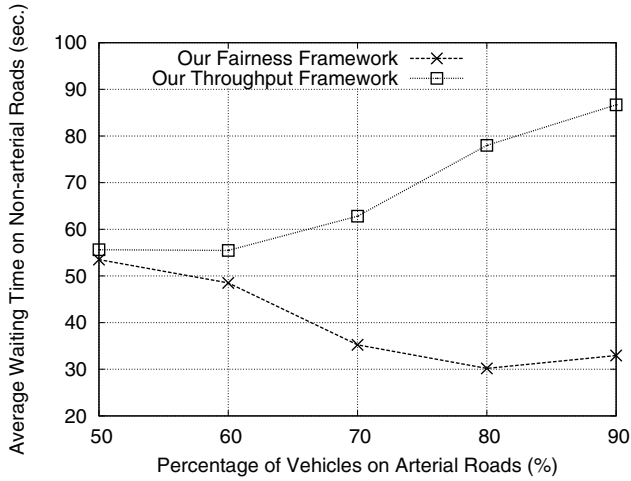


Fig. 9: Comparisons of average waiting time under different percentages of vehicles on arterial roads.

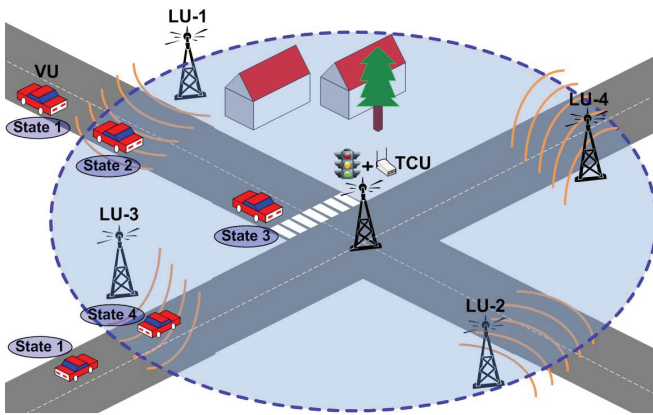


Fig. 10: System architecture of our Eco-Sign.

and control the engine ignition timing. Below, we describe the hardware requirements of VU, LU, and TCU.

VU consists of a network interface, an On Board Diagnostics (OBD) interface, an ignition control circuit, and a microprocessor. The network interface is to communicate with LU and TCU. The OBD interface is to capture the current vehicle status. The ignition control circuit is responsible for turning on/off the engine at intersections as its speed is 0 (alternatively, this can serve as a recommendation service only). The microprocessor collects information from all other components and issues commands to them.

LU is equipped with a directional antenna and is placed at a road segment with a certain distance from the intersection. It periodically broadcasts location beacons on a dedicated channel, which contain IDs of the road segment and TCU, to vehicles entering or exiting the intersection. Based on these location beacons, a vehicle can obtain the associated TCU's ID and detect the road segment that it is approaching or exiting. On the other hand, the approaching and exiting road segments will be recorded by TCU during the vehicle registration and deregistration processes, respectively. So the arrival rate for a specific road segment and the departure rates for going straight, turning left, and turning right events can be estimated individually by TCU.

TCU consists of a microprocessor and two omnidirec-

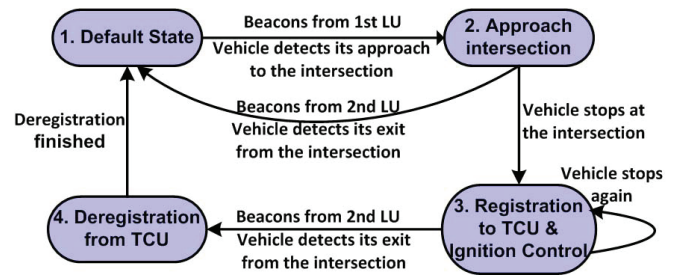


Fig. 11: State transition diagram of vehicles.

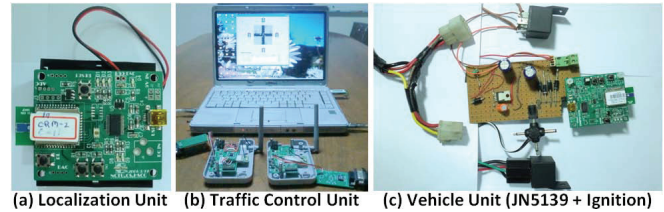


Fig. 12: Eco-Sign hardware components.

tional antennas operating in distinct channels, one for vehicle registration and the other for deregistration. After receiving the registration message from VU, TCU will reply an acknowledgement message containing the traffic sign timing. Then, VU calculates the remaining waiting time T_{rwt} and counts down to 0 locally. If $T_{rwt} \geq \alpha$, the engine can be automatically turned off to save fuel if the system is so set. During the countdown period, if the engine has been turned off, it will be automatically turned on when $T_{rwt} \leq \beta$. Here, α and β are tunable parameters.

Fig. 11 shows the state transition diagram of vehicles. Initially, a vehicle is in state 1 (Default State). When the vehicle approaches an intersection, it will receive beacons from the first passed LU and then enter state 2 (Approach Intersection). If it passes the intersection without stopping, it will receive beacons from the second passed LU and then switch back to state 1. On the contrary, if it stops at the intersection due to red signs, it will enter state 3 (Registration to TCU & Ignition Control), register itself to TCU, and decide if its engine should be turned off. Note that if it stops at the intersection more than once, it will register itself again and turn off its engine for each stop. When it exits the intersection, it will receive beacons from the second passed LU and then enter state 4 (Deregistration from TCU). After it deregisters itself from TCU, it will go back to state 1.

We have developed a prototype of Eco-Sign. The microprocessor and network interface used in VU, LU, and TCU is Jennic JN5139 [26], which has a 16MIPs 32-bit RISC processor, a 2.4GHz IEEE 802.15.4-compliant transceiver, 192kB of ROM, and 96kB of RAM. As shown in Fig. 12(a), LU is implemented by JN5139 development board powered by four AA batteries.

TCU is implemented by integrating two JN5139 development boards connected via USB ports with a notebook, which is running on Windows XP operating system, as shown in Fig. 12(b). A graphic user interface (GUI) is developed by Microsoft Visual C# 2010 integrated development environment (IDE) [27]. The GUI shows the road segment located by a



Fig. 13: Prototype demonstration.

vehicle at the intersection, messages received from vehicles, and configuration fields of traffic sign timings.

VU is implemented by JN5139 development board integrated with the ignition control circuit, as shown in Fig. 12(c). The interfacing circuit for ignition control is ULN2003 Darlington transistor arrays [28], which take the signal from Jennic I/O pins and control 12 V, 40 A automotive relays. A Y-type connector is provided to avoid cutting the original wires inside the car. The power of VU is supplied by the 12 V in-vehicle battery.

Fig. 13 shows the prototype demonstration of Eco-Sign. TCU and LU are installed at the intersection and the roadside lamppost on each direction leading to the intersection, as shown in Fig. 13(a) and Fig. 13(b), respectively. They are operating on 3 different channels. VU is set up in Maruti Suzuki 800 [29], as shown in Fig. 13(c). On one hand, the vehicle is driven from each road segment of the intersection to register/deregister itself to/from TCU. On the other hand, the traffic sign timing is varied on road segments of the intersection and the vehicle can receive the correct timing value from TCU. In addition, automatic ignition control is activated as the vehicle speed is 0 km/hr and the remaining waiting time is larger than 30 seconds.

For indoor demonstration, we use a projector to simulate traffic conditions on road segments at an intersection, as shown in Fig. 13(d). A remote control car with VU is used to approach the intersection and trigger operations of Eco-Sign. Fig. 13(e) shows the car status including motion status, approaching side, remaining waiting time, and ignition advice.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we propose a dynamic traffic control framework using vehicular communications to utilize turning intentions and lane positions of vehicles for maximizing the intersection throughput and providing fairness among traffic flows. With vehicular communications, the traffic controller of an intersection collects all turning information before vehicles make their turns. In addition, our signal scheduling algorithm takes the mixture of vehicles with different turning intentions into consideration. Furthermore, while allocating green signs to the traffic flows with higher passing rates for throughput maximization, our framework also allocates green signs to the ones with lower passing rates for fairness

provision. We also design and implement a prototype to collect vehicle information on the road to verify the feasibility of our framework. In the paper, we mainly focus on how to optimize the throughput and fairness of a single intersection. The throughput maximization problem for multiple intersections in the road network deserves further investigation.

APPENDIX

The following 16 phase matrices are used for both the intersection in Fig. 1 and our simulations.

$$S^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, S^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S^4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$S^5 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S^6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S^7 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S^8 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^9 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, S^{10} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S^{11} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, S^{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$S^{13} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, S^{14} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S^{15} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S^{16} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

REFERENCES

- [1] eSafety forum, "ICT for clean and efficient mobility," Working Group Final Repor, Nov. 2008.
- [2] D. Schrank and T. Lomax, "Urban mobility report," Texas Transportation Institute, Sep. 2004.
- [3] J. Halkias and M. Schauer, "Red light, green light," *Public Roads*, vol. 68, no. 3, Nov. 2004.
- [4] Y. Li, W. Wei, and S. Chen, "Optimal traffic signal control for an urban arterial road," in *Proc. 2nd Int. Symp. Intell. Inf. Technol. Appl.*, Dec. 2008, pp. 570–574.
- [5] E. Azimirad, N. Pariz, and M. B. N. Sistani, "A novel fuzzy model and control of single intersection at urban traffic network," *IEEE Syst. J.*, vol. 4, no. 1, pp. 107–111, Mar. 2010.
- [6] V. Verroios, K. Kollias, P. K. Chrysanthos, and A. Delis, "Adaptive navigation of vehicles in congested road networks," in *Proc. 5th Int. Conf. Pervasive Services*, July 2008, pp. 47–56.
- [7] N. Roupail, A. Tarko, and J. Li, "Traffic flow at signalized intersections," Transportation Research Board, United States Department of Transportation - Federal Highway Administration, 2001, [Online]. Available: <http://www.tft.pdx.edu/docs/chap9.pdf>
- [8] S. Sheik Mohammed Ali, B. George, L. Vanajakshi, and J. Venkatraman, "A multiple inductive loop vehicle detection system for heterogeneous and lane-less traffic," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1353–1360, May 2012.
- [9] M. S. Ngandjon and S. Cherkaoui, "On using compressive sensing for vehicular traffic detection," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, July 2011, pp. 1182–1187.
- [10] G. S. Thakur, P. Hui, and A. Helmy, "Modeling and characterization of urban vehicular mobility using web cameras," in *Proc. IEEE Infocom Workshop Netw. Sci. Commun. Netw. (NetSciCom)*, Mar. 2012, pp. 262–267.
- [11] W. Chen, L. Chen, Z. Chen, and S. Tu, "A realtime dynamic traffic control system based on wireless sensor network," in *Proc. Int. Conf. Parallel Process. Workshops*, June 2005, pp. 258–264.
- [12] W. Chen, L. Chen, Z. Chen, and S. Tu, "WITS: A wireless sensor network for intelligent transportation system," in *Proc. 1st Int. Multi-Symp. Computer Computational Sci.*, June 2006, pp. 635–641.
- [13] W. Wen, "A dynamic and automatic traffic light control expert system for solving the road congestion problem," *Int. J. Expert Syst. Appl.*, vol. 34, no. 4, pp. 2370–2381, May 2008.
- [14] K. A. S. Al-Khateeb, J. A. Johari, and W. F. Al-Khateeb, "Dynamic traffic light sequence algorithm using RFID," *J. Computer Sci.*, vol. 4, no. 7, pp. 517–524, July 2008.
- [15] A. Chattaraj, S. Bansal, and A. Chandra, "An intelligent traffic control system using RFID," *IEEE Potentials*, vol. 28, no. 3, pp. 40–43, May 2009.
- [16] A. A. Pandit, A. K. Mundra, and J. Talreja, "RFID tracking system for vehicles," in *IEEE 1st Int. Conf. Computational Intelligence, Commun. Syst. Netw.*, July 2009, pp. 160–165.
- [17] W. Wen, "An intelligent traffic management expert system with RFID technology," *Int. J. Expert Syst. Appl.*, vol. 37, no. 4, pp. 3024–3035, Apr. 2010.
- [18] R. Wunderlich, I. Elhanany, and T. Urbanik, "A stable longest queue first signal scheduling algorithm for an isolated intersection," in *Proc. IEEE Int. Conf. Veh. Electronics Safety*, Dec. 2007, pp. 1–6.
- [19] R. Wunderlich, C. Liu, I. Elhanany, and T. Urbanik, "A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 536–547, Sep. 2008.
- [20] L.-W. Chen, Y.-H. Peng, and Y.-C. Tseng, "An infrastructure-less framework for preventing rear-end collisions by vehicular sensor networks," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 358–360, Mar. 2011.
- [21] E. Koukoumidis, L.-S. Peh, and M. Martonosi, "Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *ACM Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, June 2011, pp. 127–140.
- [22] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Proc. IEEE Veh. Technol. Conf. (VTC-Spring)*, May 2008, pp. 2036–2040.
- [23] "On board diagnostics (OBD)," [Online]. Available: <http://www.obdii.com/background.html>
- [24] "Simulation of urban mobility (SUMO)," [Online]. Available: <http://sumo.sourceforge.net/>
- [25] L.-W. Chen, P. Sharma, and Y.-C. Tseng, "Eco-sign: A load-based traffic light control system for environmental protection with vehicular communications," in *Proc. ACM Int. Conf. Special Interest Group Data Commun. (SIGCOMM)*, Aug. 2011.
- [26] "Jennic," JN5139, [Online]. Available: <http://www.jennic.com>
- [27] "Microsoft visual C# 2010 integrated development environment (IDE)," [Online]. Available: <http://msdn.microsoft.com/en-us/vcsharp/default.aspx>
- [28] ULN2003, "High voltage and high current Darlington transistor array," [Online]. Available: http://www.datasheetcatalog.org/datasheets/120/489337_DS.pdf
- [29] "Maruti Suzuki 800," [Online]. Available: <http://www.maruti800.com/>



Lien-Wu Chen received his B.S. and M.S. degrees, both in computer science and information engineering, from Fu-Jen Catholic University and National Central University in 1998 and 2000, respectively. He was a Research Assistant at the Institute of Information Science, Academia Sinica from Jan. 2001–Feb. 2009. He obtained his Ph.D. in computer science and information engineering from National Chiao Tung University in December of 2008. He joined the Department of Computer Science, National Chiao Tung University, as a Postdoctoral Research Associate in Feb. 2009, and qualified as an Assistant Research Fellow in Feb. 2010. He is currently an Assistant Professor since Feb. 2012 in the Department of Information Engineering and Computer Science, Feng Chia University. His research interests include wireless communication and mobile computing, especially in mobile ad hoc and sensor networks.

Dr. Chen has published 28 conference and journal papers and has 24 filed patents. Among those patents, the technologies for file transfer, application execution, and goods data searching methods based on augmented reality and cloud computing have been transferred to Acer Incorporated in Jan., June, and Nov. 2011, respectively. He was a guest researcher at the Industrial Technology Research Institute (Feb. 2009–Dec. 2009) and Chief Executive Officer at the RFID Resource Center, National Chiao Tung University (Apr. 2009–Mar. 2011). Dr. Chen received the Outstanding Demo Award (1st author) at IEEE MASS 2009, the Best Paper Award (1st author) at WASN 2010, the Audience Award (2nd prize) at ESNC 2011, and the Best Paper Award (1st author) at MC-2012. He is an honorary member of the Phi Tau Phi Scholastic Honor Society, and is a member of ACM and IEEE.



Pranay Sharma received his B.E. degree in manufacturing process and automation engineering from Netaji Subhash Institute of Technology, Delhi University, India, in 2008 and his M.S. degree in electrical engineering and computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2011. He is currently a Design Engineer at Metanoia Communications, Hsinchu, Taiwan. Previously, he worked at the Foundation for Innovation and Technology Transfer (Indian Institute of Technology, Delhi, India) as a Project Associate, and has research experience at the Gerontechnology Research Center, Yuan-Ze University, Taiwan, as a Research Intern. His research interests lie in intelligent transportation systems and their adaptation to the contemporary world.

Mr. Sharma received the Outstanding Graduate Student (Research) Award by National Chiao Tung University for the year 2011. He has previously also received merits from the Taiwan Government (2009) and awards from Microsoft Corporation (Imagine Cup 2010 and 2011) and Anwendungszentrum GmbH (European Satellite Navigation Competition 2011).



Yu-Chee Tseng earned his Ph.D. in computer and information science from the Ohio State University in January 1994. He was/is Chairman (2005–2009), Chair Professor (2011–present), and Dean (2011–present), at the Department of Computer Science, National Chiao Tung University, Taiwan.

Dr. Tseng is a Y. Z. Hsu Scientific Chair Professor. Dr. Tseng received the Outstanding Research Award (National Science Council, 2001, 2003, and 2009), Best Paper Award (Int'l Conf. on Parallel Processing, 2003), Elite I. T. Award (2004), Distinguished Alumnus Award (Ohio State University, 2005), and the Y. Z. Hsu Scientific Paper Award (2009). His research interests include mobile computing, wireless communication, and sensor networks. Dr. Tseng is an IEEE Fellow. He serves/served on the editorial boards of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY (2005–2009), IEEE TRANSACTIONS ON MOBILE COMPUTING (2006–2011), and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (2008–present).