# Employing Cyber-Physical Systems: Dynamic Traffic Light Control at Road Intersections

Ossama Younis, *Member, IEEE*, and Nader Moayeri, *Senior Member, IEEE*

***Abstract*—Traffic lights (TLs) are used to control traffic at road intersections. Typically, the TL control mechanism on the road operates according to a fixed periodic schedule to change the light (red/yellow/green). A different schedule may be employed at late night or early morning hours. Such fixed light control does not react to changing traffic conditions, and is unaware of (unresponsive to) congestion. In this paper, we propose a novel framework dynamic TL control (DTLC) at road intersections. DTLC relies on a sensor network to collect traffic data and includes novel protocols to handle congestion and facilitate more efficient traffic flow. The proposed low-overhead algorithms are practical to employ in live traffic flow scenarios. Through analysis and simulations, we demonstrate the benefits of DTLC in optimizing traffic flow metrics, such as traffic throughput, vehicle waiting time, and waiting line length. This paper is a step toward employing smart TLs, which are part of the future envisioned "Smart City."**

***Index Terms*—Distributed algorithms, sensor networks, smart city, traffic flow optimization, traffic light (TL) control.**

## I. Introduction

THE RECENTLY proposed cyber-physical systems (CPSs) define how the computing world manipulates and interacts with the physical world. CPSs are being developed for various application domains, such as environmental monitoring and smart homes. While the CPS area is developing, several questions need to be answered. For example, it is not clear if the computing foundations are adequate for supporting and using CPSs [1]. Some argue that CPSs need top-to-bottom rethinking of computation (i.e., require changing the computation model). One of the practical domains that benefit from advances in CPSs is the optimization of transportation systems (e.g., developing and employing smart road optimization approaches).

Optimizing traffic flow/handling congestion are classic issues in transportation systems. Research in Civil Engineering has focused on improving the road conditions, changing (or adding) traffic lanes, and controlling traffic lights (TLs) to improve the traffic flow, especially on the main/congested roads. New research focuses on utilizing CPSs to optimize traffic flow and improve the driver's experience on the road. Most previous research (see Section VIII) has shown the "benefits" of using controlled TLs and proposed "theoretical" methods (centralized control). However, practical methods are still needed to optimize the control of TLs based on the traffic flow conditions. Therefore, we plan in this paper to address how to "perform" traffic flow control at intersections for optimizing the traffic flow metrics, such as throughput and delay.

In this paper, our contributions are as follows. First, we present a "practical" framework for optimizing the flow of traffic using a dynamically changing TL. Our framework includes collecting road conditions by sensors deployed on the sides of the road [road-side units (RSUs)], and proposes novel distributed algorithms to decide on when to switch TLs to alleviate congestion. Second, we argue that our proposed techniques are simple/fast and have low overhead. Thus, they can be practically applied on road traffic "online" (i.e., while traffic is flowing on the road). Third, through analysis and simulations, we demonstrate the benefits of the proposed framework and techniques to enable better flow of traffic. Compared to the previous work in this area (see Section VIII), this paper is novel and unique due to of the following reasons.

1) It proposes detailed efficient solutions (proposed algorithms and supporting hardware) to be employed on the existing TL control systems.
2) It can be employed to all types of traffic models.
3) It can be employed "gradually," i.e., on certain TLs/streets and then to entire regions. Significant impact on performance can be achieved even with local deployment. This supports desirable functions for the envisioned future Smart City [29].

The rest of this paper is organized as follows. Section II describes our general system model. It also lists the objectives of this paper. Section III explores the possible design approaches for TL control. Sections IV and V present two proposed distributed solutions for enhancing traffic flow. Section VI analyzes the proposed techniques by outlining the possible triggers to change the TL ahead of its schedule, and describing the overhead of the proposed solutions. Finally, it briefly discusses how to achieve global/regional TL optimization. Section VII evaluates the proposed algorithms on possible light/heavy traffic scenarios. Section VIII overviews related research on traffic flow control. Section IX introduces a tentative hardware platform that may support our proposed framework. It also discusses other tentative road applications (pedestrian safety). Finally, Section X concludes this paper and provides directions for future research.

Manuscript received July 13, 2017; revised October 2, 2017; accepted October 17, 2017. Date of publication October 20, 2017; date of current version December 11, 2017. *(Corresponding author: Ossama Younis.)*

O. Younis is with Smart Streets LLC, Rockville, MD 20850 USA (e-mail: ossyounis@gmail.com).

N. Moayeri is with the Advanced Network Technologies Division, National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899-8920 USA.

Digital Object Identifier 10.1109/JIOT.2017.2765243

## II. System Model and Objectives

### A. Road/Traffic Model

We make realistic assumptions about the traffic pattern, the vehicles capabilities, and the road intersection.

1) *Road:* Typical model with intersecting roads. A road has two opposite segments, each allowing traffic to go along one or more lanes.
2) *TLs::* Installed at the head of each road segment. The TLs at two opposite (or intersecting) segments always show the same (or opposite) colors.
3) *Traffic Model:* We propose TL control techniques that are independent of the vehicle arrival model. Thus, the vehicles arrive according to any arbitrary stochastic model, and may proceed in any direction (forward, left, or right).
4) *Vehicle Detection:* By exploiting either RSU sensors or vehicle-installed devices, detect or report the vehicles that arrive at the intersection on each of the four segments.

### B. Objectives

It is required to develop mechanism(s) to optimize/control the traffic flow. These control protocols "collect" traffic flow information and use it to optimize the traffic flow metrics, such as traffic throughput or waiting time in TLs, fairness, etc. The proposed protocols have the following features: 1) low processing complexity; 2) low communication overhead; 3) fast decision-making; and 4) low (practical) cost for deploying the proposed protocol(s).

It is necessary to motivate state/city councils to adopt such new technology in TLs, and to avoid any problems if there are noncompliant vehicles on the road.

## III. Design Approaches for DTLC

### A. Design Approaches

We categorize the design of dynamic traffic control mechanisms into two possible alternatives.

*1) Design Approach 1:* The rationale of this approach is to rely on the road infrastructure to sense congestion and trigger TL changes accordingly (i.e., at unequal time intervals). Deploy sensor infrastructure (RSUs) to collect information about the road conditions (e.g., the number of vehicles waiting for a green light). Also deploy a TL controller (TLC) at the TL to control its state based on an algorithm that uses information from the RSUs. The infrastructure can also be sensors placed at road lanes. This requires that the sensors can communicate their data to the TLC. We name this alternative road-based infrastructure TL control (RITCO).

*2) Design Approach 2:* The rationale of this approach is to give the vehicles the responsibility to collaborate and exchange information to determine if a congestion condition has occurred, and thus early TL change should be triggered. Use distributed infrastructure, e.g., a TLC at the TL and a communication device per vehicle to report the vehicles' arrivals and locations (road segments) to the TLC. This reduces the amount of road infrastructure, but imposes requirements on the vehicles to support the TLC. We name this design approach vehicle-based infrastructure TL control (VITCO).

### B. Which Design Approach to Endorse

Several issues need to be considered for developing and adopting any design approach.

1) *Complexity/Cost:* Evaluate processing/communication complexity, in addition to cost. Designs that require significant processing are acceptable if: a) the computing devices are expected to be capable of rapidly carrying out the computations and b) the communication complexity is expected to be much more expensive than the processing complexity. The cost is also a determining factor.
2) *Applicability:* Describes which mechanism is more practical in the transportation system/road network model under study, and therefore, is more applicable. This involves studying several issues, such as effectiveness, ease of deployment, and deployment/running cost.
3) *Performance Metrics:* Several metrics are defined in Section VII-A (e.g., throughput) to evaluate the proposed technique(s). This helps to determine the suitable design approach for the road/traffic model that we are optimizing. Note that the traffic models may require different optimization mechanisms to operate optimally.

Selecting which design approach to endorse and implement will typically depend on the analysis of the above issues.

## IV. RITCO

### A. Introduction

RITCO uses "design approach 1" (Section III-A). It requires the following.

1) *TL Controller:* a) Co-located with the TL and b) collects information about the current road conditions from RSUs and uses it for light control.
2) *Sensors at RSUs:* Each road segment is equipped with at least one RSU installed several meters away from the intersection to coordinate the detection of vehicle arrivals. If the TL is going to change to red at time $T_{\mathrm{Red}}$, each of the two RSUs installed on the opposite road segments with the red light is notified of the impending change at time $(T_{\mathrm{Red}} - \Delta t)$, where $\Delta t$ is a small number, to start counting the vehicles that will be waiting at the TL. The count is periodically reported to the TLC. If multiple sensors are used, then sensors coordinate to avoid duplicate counts Using a single sensor (if possible) eliminates such overhead.

### B. RITCO Protocol

*1) Overview:* RITCO is a distributed protocol that is executed at the TLC and at the RSUs. The TLC announces when the TL is going to change (based on, e.g., timer expiration or the waiting vehicles count exceeds a threshold). Starting at $\Delta t$ time ahead of the TL change to red, the RSUs begin to count

TABLE I
DEFINITION OF SYMBOLS USED IN THE PROTOCOLS

| Symbol | Definition |
|---|---|
| $T_{curr}$ | Current time |
| $\Delta t$ | Very small-time interval (in sec. or milli-sec.) |
| $T_{red}/T_{yellow}/T_{green}$ | Red/Yellow/Green light intervals |
| $T_{change}$ | Time until the traffic light changes to red/green |
| NumVehicles | The number of vehicles |
| $V_{ID}$ | Vehicle ID (unique ID, e.g., license plate no.) |
| switchLightFlag | When ON, traffic light change is triggered (otherwise, OFF) |

the vehicles that will be waiting at the red light. While the TL is red, if the number of waiting vehicles exceeds a threshold, the TLC triggers early TL change to green. TLC activates the process of vehicle counting at the RSU. While active, the arriving vehicles are counted and periodically reported to the TLC (for reliability). When the green light is approaching (specified by the TLC), the RSUs stop counting the arriving vehicles. Table I lists/defines the symbols used in the RITCO protocol (and later in the VITCO Protocol). Table II lists the pseudo-code of the RITCO protocol. Note that the light does not need to be changed on the "green" side if no traffic is waiting (i.e., the RSU does not sense waiting vehicles on the other arm).

## V. VITCO

### A. Introduction

VITCO uses "design approach 2" in Section V-A. It requires the following.

1) *TL Controller:* Co-located with the TL. It simply interprets and executes VITCO decisions.
2) *Vehicle Radio $V_R$:* A vehicle's communication device that announces a vehicle's arrival at the TL. Thus, the $V_R$'s in VITCO coordinate to control the TL.

### B. VITCO Protocol

*1) Overview:* VITCO is a distributed protocol that is executed at the arriving vehicles. It includes electing a coordinating vehicle (CoVe), which controls the TL transitions. To elect the CoVe, the first vehicle arriving/stopping at the red light elects to become the CoVe. In the rare case that later another CoVe is found, one CoVe can yield its leadership to the other using a simple arbitration mechanism (e.g., the comparison of license plate numbers or the proximity to the TL). The CoVe counts the number of waiting vehicles (similar to the TLC in RITCO). The CoVe thus keeps track and announces when the TL is going to change to green. While the TL is red, if the number of waiting vehicles exceeds a threshold, the CoVe triggers early TL change to green. Table III provides the pseudo-code of the VITCO protocol, which is executed at every road segment by the CoVe and the TLC (the symbols are the same as in Table I). Similar to RITCO, the light does not need to change to "red" on the green side if no vehicle is waiting on the other arm of the intersection.

TABLE II
RITCO PROTOCOL DETAILS

**Steps at TLC (Triggered when traffic light→Red):**

- **Initialize:** Initialize all RSU vehicle count(s) to zero
- **Initialize:** numVehicles←0; switchLightFlag←OFF
- TLC announces to RSUs when TL changes to red.
- **While** (($T_{curr}+\Delta t \geq T_{red}$) && (switchLightFlag is OFF, i.e., threshold not exceeded))
  - Collect sensor (RSU) readings
  - **If** (*m* new vehicles are detected by the RSU since the last data collection)
    - **Then** numVehicles ← numVehicles+*m*
      Save vehicle IDs to vehicle list
    - **Else** //(*m*=0) … Ignore sensor reading
  - UpdateFlag(switchLightFlag)
- **End While**
- **If** (switchLightFlag is ON)
  - Perform TL_Change proc. (Green←Red).
- TLC announces to RSUs when TL changes to green.

**Steps at Sensor (RSU on Pole/Sensor on Road Lane):**

- **Initialize:** Vehicle counter (numVehicles ← 0).
- RSU prepares to activate the sensing process:
  - While ($T_{curr}+\Delta t \leq T_{change}$)
    - Wait/update $T_{curr}$ to the current time.
- **Activate** RSU for vehicle counting
- Whenever (RSU detects a vehicle)
  - numVehicles ← numVehicles+1
- Every k seconds, RSU checks:
  - **If** (numVehicles > 0)
    - Report numVehicles to the TLC
    - RSU resets (numVehicles ← 0).
    - Based on upcoming TL change to green, RSU decides when to stop counting & when to report its count to the TLC.

**TL_Change Procedure (Green ← Red):**

- Change green light to yellow.
- Wait for an interval $T_{yellow}$ time.
- Change yellow light to red.

**TL_Change Procedure (Red ← Green):**

- Change red light to green.

**UpdateFlag(flag):**

- If (light switch trigger is satisfied)
- Then flag ← ON
- Else flag ← OFF

### C. Example

In Fig. 1, we illustrate the operation of the proposed TL control optimization approaches, in addition to the legacy TL system operation. We illustrate three approaches for TL control.

*Classical TL:* Every *k* seconds, the TL changes (red → green or green → orange → red). The *k* seconds are fixed regardless of the traffic conditions.

*RITCO:* Count the arriving/waiting vehicles at the red lights (TL2_1 and TL2_2) using sensors on the road sides (RSUs).

TABLE III
VITCO PROTOCOL DETAILS

**VITCO: Elect a Coordinating Vehicle (Elect_CoVe):**

- A vehicle V1 arrives/stops at the TL.
- V1 transmits a message for CoVe and neighbor discovery and sets an expiration timer (T1).
- If a response message announcing CoVe ID arrives (from a neighbor), cancel T1. Then, either register CoVe (single-hop communication) or find the path to that CoVe (multi-hop communication).
- If timer T1 expires and no CoVe announcement has arrived yet, broadcast vehicle V1 as the new CoVe.
- If later, V1 receives a CoVe broadcast (after announcing itself as CoVe), arbitrate among the messages to pick the best CoVe. If selected, rebroadcast a CoVe message.
- If other CoVe is better (using either a metric, e.g., location, or a heuristic, e.g., license number), then V1 broadcasts a NON-CoVe message, including the other CoVe ID.

**VITCO: Steps at CoVe (Lights Changing to Red):**

- Vehicle Vi arrives at the traffic light.
- Vi discovers its neighbors and CoVe (Elect_CoVe).
- Assume the elected CoVe is Vc.
- numVehicles ← 0.
- While (TL_Timer has not expired)
  - o If (Vi is Vc) Then
    - ▪ Broadcast a CoVe announcement
    - ▪ When (message arrives from a vehicle)
      - If (new vehicle)
      - Then Increment numVehicles … sendMsgToTLC(False, numVehicles)
      - EndIf
    - ▪ When (message arrives from a vehicle)
      - If (new vehicle)
      - Then Increment numVehicles … sendMsgToTLC(False, numVehicles)
      - EndIf
    - ▪ UpdateFlag(switchLightFlag)
    - ▪ If (switchLightFlag=ON, i.e., threshold is exceeded) Then
      - SendMsgToTLC(True, numVehicles)
      - When the TLC confirms (TL changes) → Exit
    - ▪ EndIf
  - o EndIf
  - o Else // (Vi is not Vc)
    - ▪ Vi registers Vc as its CoVe
    - ▪ discovers the communication path to Vc (using either single-hop or multi-hop forwarding)
  - o EndIf
- EndWhile

Note that the function Veh_SendMsgToTLC at the CoVe translates to function TLC_RecvMsg at the TLC (see pseudo-code below).

**TLC_RecvMsg(Bool changeTL, int numVehicles)**

- If (changeTL = TRUE) Then
  - o TLC.numVehicles = 0
  - o Change the state of each traffic light
  - o Exit Procedure
- EndIf
- If (numVehicles > 0)
  - o TLC.numVehicles = numVehicles
  - o If (TLC.numVehicles >= MAX_WAITING_VEH)
    - ▪ TLC.numVehicles = 0
    - ▪ If (TL is red)
      - Then TL_Change(Red ← Green)
      - Else TL_Change(Green ← Red)
    - ▪ EndIf
    - ▪ Exit Procedure
  - o End If
- EndIf

Pass this count from the RSUs to the Advanced TLC. When the count at the TLC reaches/exceeds a threshold, the TLC changes the light from red to green (i.e., does not wait for the typical $k$ seconds to elapse before switching the TL).

*VITCO:* Elect a CoVe at each red light to count the vehicles that are waiting for the green light. Count the arriving/waiting vehicles at the red lights (TL2_1 and TL2_2) using communication between the arriving vehicles and the CoVe. The CoVe periodically reports the number of waiting vehicles to the Advanced TLC. When the count at the TLC reaches/exceeds a threshold, the TLC changes the light to green (i.e., does not wait for the typical $k$ seconds to elapse).

## VI. Analysis of the Proposed Approaches

### *A. Triggers to Change TLs Synchronously or Asynchronously*

The current TL transitions use fixed (preassigned) intervals for the different lights (i.e., M1 seconds for red and M2 seconds for green, where M1 and M2 are constants). Recently, in some areas, a form of adaptive light transitions is employed. However, such transitions were limited to prolonging the green light at main streets until a vehicle arrives at a side street. In this paper, we consider a much broader scope for the reasons of TL change and propose a comprehensive approach for adaptive TLs. The TL can be changed (red→green or green→red) in one of two ways.

1) *Synchronously:* According to a timer expiration (which is being employed on the roads now).
2) *Asynchronously:* The TL changes if certain events trigger the need for light change (including timer expiration).

Triggers for asynchronous TL change can be due to (but are not limited to) the occurrence of the following events.

1) The number of vehicles that are waiting for the green light exceeds a certain threshold (on a single or both sides of the TL). Considering the waiting vehicles on each side separately is more practical to avoid having very long waiting line anywhere.
2) No vehicles have crossed the road intersection from the sides that have the green light for a certain time interval.

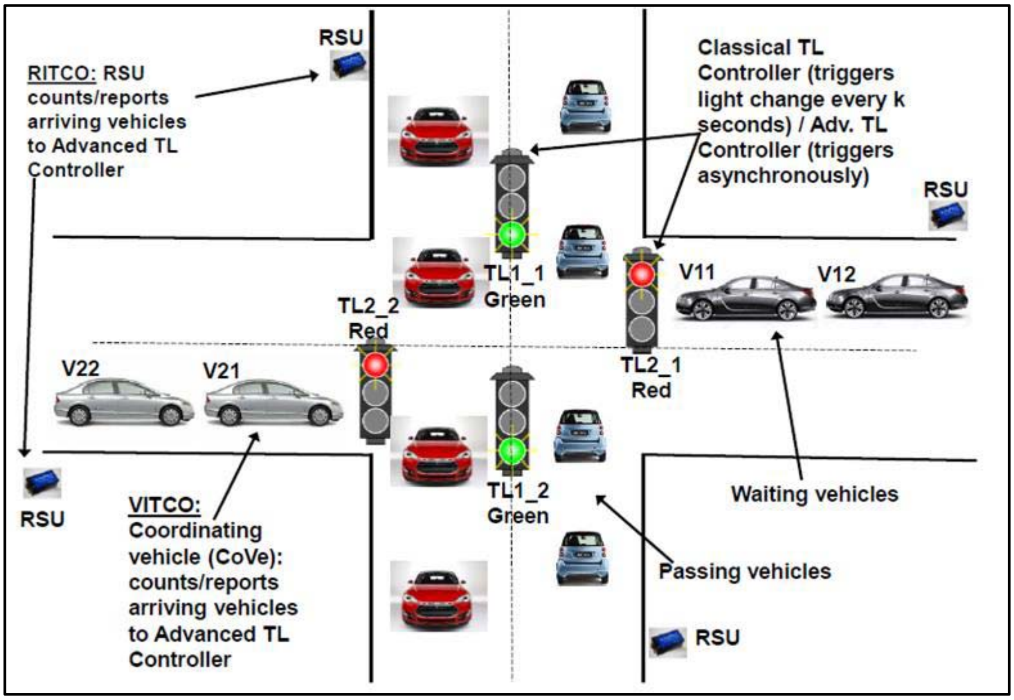



Fig. 1. Illustration of TL optimization techniques. To count/report the arriving vehicles, RITCO uses RSUs while VITCO uses a CoVe. TL change is triggered based on vehicle count.

## *B. Properties/Overhead*

*1) Communication:* RITCO has the advantage of being independent of the vehicles on the road, i.e., requiring no exchanged communication between the vehicles and the infrastructure. VITCO has communication cost to exchange a constant number of messages between: 1) arriving vehicles and their coordinator (CoVe) and 2) CoVe and TLC.

*2) Processing:* RITCO puts all the burden on the road infrastructure (e.g., RSUs). On the other hand, VITCO puts most of the processing burden on the representative vehicle that communicates with the TL, as well as, the vehicles on its segment of the road. In either case, typical processing will simply require comparing the number of arriving vehicles to a threshold, which is $O(1)$ per vehicle or $O(N_v)$ for all $N_v$ arriving vehicles. This is quite inexpensive processing cost due to typical small $N_v$ (in the range of tens).

*3) Cost:* RITCO puts all the burden on the city/county (road infrastructure and TL control equipment). VITCO shares the cost between the city/county and the vehicles' drivers.

*4) Deployment:* It is much easier/safer to rely on RITCO because of its independence of the vehicles on the road. It avoids the problems of faulty/tampered vehicle equipment. RITCO also avoids the transition period in which the vehicles will need to deploy the necessary equipment to support VITCO.

*5) Security:* RITCO is more secure than VITCO because it does not acquire information exchange with the vehicles on the road (just counts them). Decisions at VITCO are made by the vehicles, which presents vulnerability to greedy users (tampered control devices). To protect drivers' privacy, the vehicles could announce their presence using unique identifiers, such as a hash value of the vehicle identification number.

*6) Communication Overhead and Processing Complexity:* Let *N:* maximum number of vehicles waiting for the TL change and *L*: number of lanes on the road.

*Lemma 1:* Message exchange overhead is: 1) $O(N/L)$ messages per vehicle to announce arrival and 2) $O(1)$ messages to broadcast TL update times [total: $O(N/L)$].

*Proof:* Assume that on arrival a new vehicle ($V$) sends a status message about itself to the coordinator. To avoid long-range messages which may cause collisions with other vehicle messages, $V$ will use multihop forwarding to send its message along its lane all the way to the coordinator. The maximum number of these forwarded messages (assuming no need for retransmission) is equal to the number of waiting vehicles in one lane, i.e., $N/L$. Thus, the communication overhead is $O(N/L)$. ■

*Lemma 2:* Processing complexity is $O(N)$.

*Proof:* The main processing executed by the two proposed approaches is to register arriving vehicles, which is $O(N)$. Checking the TL trigger for light change is $O(1)$ per vehicle [i.e., $O(N)$ overall]. ■

*Lemma 3:* To decide whether/when to change the TL, the complexity can be as low as $O(1)$.

*Proof:* Heuristic-based algorithms, which compare the number of waiting vehicles to a threshold (e.g., RITCO), have low overhead of $O(1)$. ■

*7) Robustness to Inaccurate Information:* Failure in sensing vehicles (RITCO) or in communicating with a CoVe (VITCO) makes information inaccurate at the controller. However, either case is not expected to happen often. Such inaccuracy will just delay triggering the light change until the light's timer expires. This alleviates any possibility for getting stuck in one light state. Light change can also be triggered early due to false detection of vehicles (or faulty RSUs). This is not a problem because there is a minimum interval imposed by the dynamic TL control (DTLC) protocol during which light change cannot be triggered.

*8) Optimizing Channel Contention:* RITCO does not suffer from channel contention issues. However, in VITCO, since multiple vehicles may arrive at the TL at the same time, contention on the communication channel may cause messages to be lost or retransmitted (indefinitely delayed). This, however, can be mitigated by: 1) enforce that each vehicle senses the channel before transmission (similar to typical CSMA MAC protocols) or 2) make the vehicles carry multiple wireless radio interface cards. Each card uses a different predetermined transmission channel. The vehicle's radio then scans which channel is free prior to transmission.

### *C. Regional/Global TL Control*

The framework (described in Sections IV and V) presents protocol algorithms for "local" traffic control at a single TL. However, based on the reactive/independent nature of the proposed algorithms, the framework *inherently* addresses global traffic optimization (across a region or a city). This is because the proposed reaction to congestion at one TL shifts the load toward other TLs, thus, continuously distributing the traffic load and relieving congested areas.

### *D. Advantages of DTLC*

A major advantage of the DTLC framework is that it makes independent decisions at each TL to achieve local-then-global optimization of traffic flows. The techniques neither require exchanging information among multiple TLs nor need to relay any information to a central controller (CC) (thus avoiding long-range communication issues). Even if a CC is available, the CC will have to deal with possible inaccurate (e.g., nonsynchronized) information.

## VII. Evaluation

We evaluate our newly proposed DTLC approach (RITCO or VITCO), in comparison to the legacy TL control. The simulation tool is developed "in-house" using C/C++ coding. Our simulator is sufficient to examine the functionality of TL control. It basically generates traffic load using an arbitrary distribution and controls the triggering of light change using our proposed protocols. A full-fledged traffic control system simulation is not needed in this paper because our focus is to only generate and handle "arbitrary" traffic at an intersection. The evaluation does not distinguish between the different flavors of DTLC, because both flavors differ only in the details of traffic information collection and not in operation. We first define the metrics then describe the experiments.

### *A. Metrics/Parameters*

The evaluation metrics are defined in Table IV. We focus on two parameters: 1) number of arriving vehicles at the TL and 2) number of TL cycles (multiple red-green transitions). Details of the remaining parameters are: # lanes = 2, lane capacity = 100 vehicles, vehicles arrival rate = 90/lane (arrivals are based on an arbitrary stochastic model—i.e., it does not matter how they arrive), threshold value (waiting vehicles) = 75, max number of passing vehicles = 150, red or green interval = 20 s (for classic TL control), yellow interval = 2 s. Each result is the average of ten random experiments. Note that the threshold values of lights on a certain road are dependent on the particular road capacity, i.e., length, number of lanes, etc.

TABLE IV
Metrics to Evaluate DTLC

| Metric | Definition |
|---|---|
| Maximum or Average waiting line length ($Q_i$) | At an intersection side, "Waiting line length" is the number of vehicles/lane. Let $N_i$ be the waiting line length at lane i, and K is the number of lanes → $Q_i = \text{Max}(N_i)$, where i ∈ [1..K] or $Q_i = \sum_i (N_i)/K$, where i ∈ [1..K]. |
| Maximum or Average waiting time ($W_i$) | The time that a vehicle waits for traffic light change (red→green). Let $\alpha_i$ be: waiting time of vehicle *i* (*i* ∈ [1..S]), and S is the number of vehicles → $W_i = \text{Max}(\alpha_i)$ or $W_i = \sum_i(\alpha_i)/S$. |
| Throughput (T) | The number of vehicles (S) that pass the intersection per unit time. Assuming an interval of *t* sec, T = S/t. |
| Fairness | Corresponding vehicles at the intersecting traffic lights experience similar delay relative to the volume on each intersection arm. |

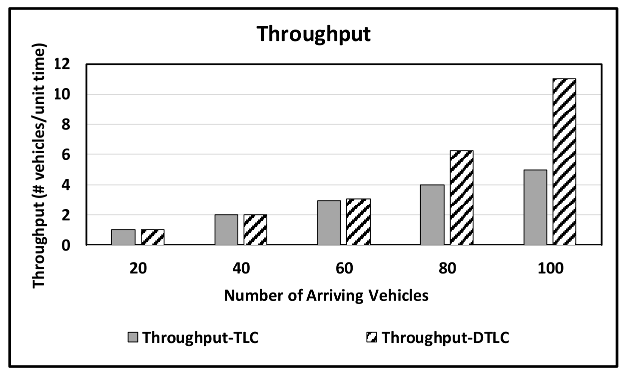



Fig. 2. Throughput versus arriving vehicles.

### *B. Vary the Number of Arriving Vehicles*

Vary the number of arriving vehicles at the TL and compute the metrics. The vehicle arrival probability function does not matter. Only the number of vehicle arrivals triggers the TL change. Fig. 2 shows the throughput at the intersection when the number of arriving vehicles varies from 20→100. The legacy TLC system does not react to increasing traffic, while DTLC reacts by early TL switching to handle congestion (congestion = 75 waiting vehicles threshold is exceeded). This explains the significant throughput gains achieved by DTLC.

Fig. 3 shows the number of waiting vehicles for green light. In DTLC, this number is upper-bounded, and early light switch is triggered when such limit is reached. Taking such action on both sides of the TL ensures that the waiting line never

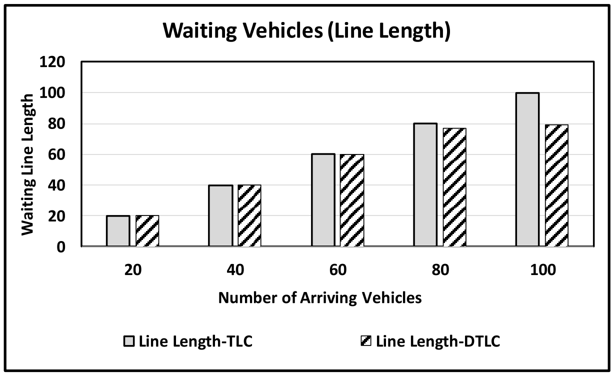



Fig. 3. Waiting line length versus arriving vehicles.

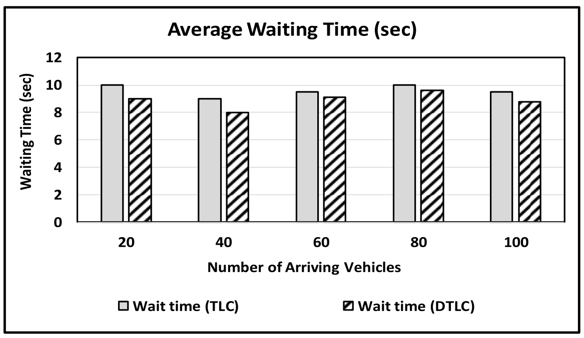



Fig. 4. Waiting time versus arriving vehicles.

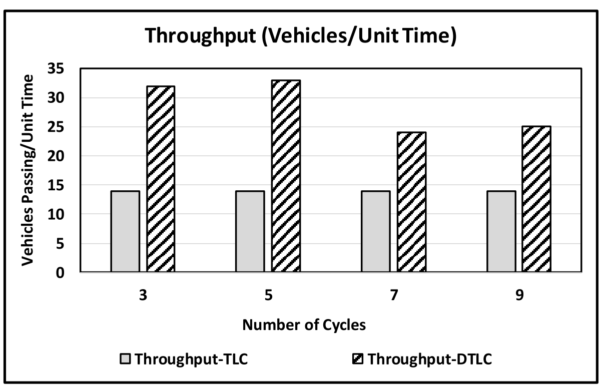



Fig. 5. Throughput versus the number of cycles.

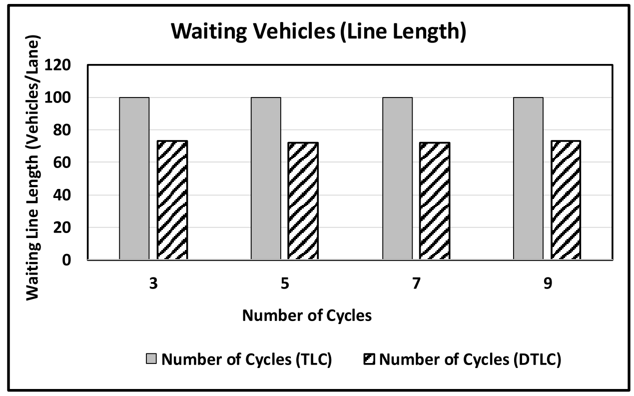



Fig. 6. Waiting line length versus the number of cycles.

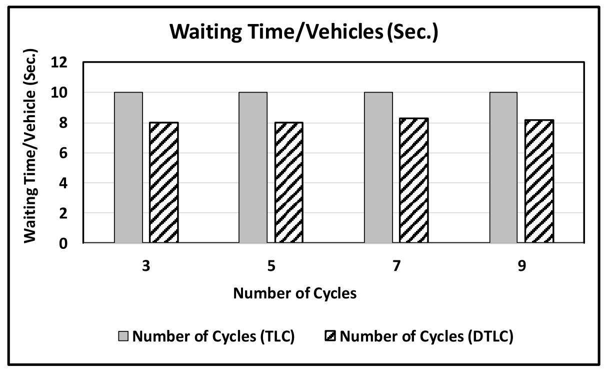



Fig. 7. Waiting time versus the number of cycles.

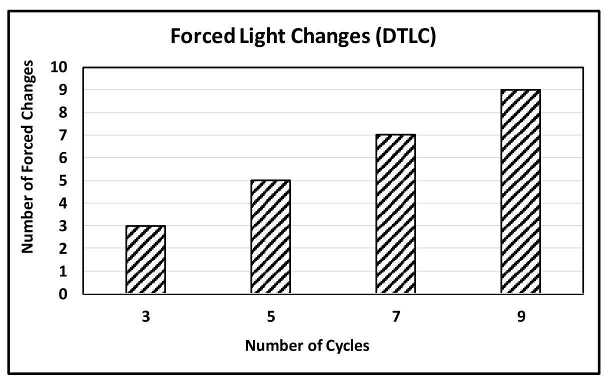



Fig. 8. Forced early light changes–high traffic load.

overflows. Fig. 4 shows that the vehicles wait slightly less with DTLC.

### *C. Vary the Number of TL Cycles (Red–Green Transitions)*

We define a "cycle" to be a red-light interval followed by a green-light interval (or green followed by red). We evaluate DTLC when the number of TL cycles varies. The number of waiting vehicles is greater than 80. Fig. 5 shows 250% traffic throughput improvement under heavy load.

Fig. 6 shows that the waiting line length with DTLC is about 20% less than that of legacy TLC (a modest improvement). Fig. 7 illustrates that DTLC provides about 25% less waiting time than TLC (intuitive, especially under heavy traffic).

### *D. Forced Early TL Change*

Fig. 8 shows the number of forced TL changes with DTLC. Under high load (i.e., exceeding the threshold), the TL changes will be forced at every cycle. Based on experience with how fast the lines grow on particular road lanes, road administrators can decide how to better set the TL intervals.

### *E. Other Possible Metrics: Fairness*

The DTLC approach clearly accounts for fairness, which can be defined in many ways, e.g., the corresponding vehicles at the intersecting TLs experience similar delays (waiting times) relative to the traffic volume on each intersection arm.

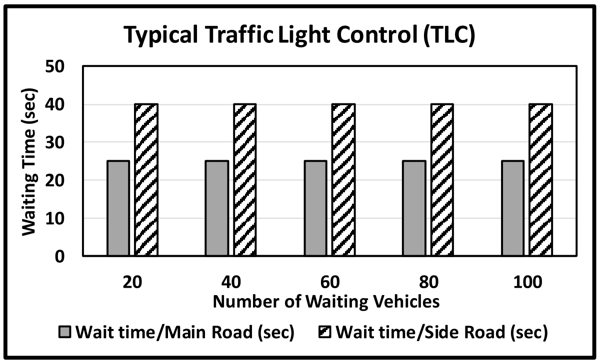



Fig. 9. Fairness with TLC (more green time is given to main road with heavy traffic).

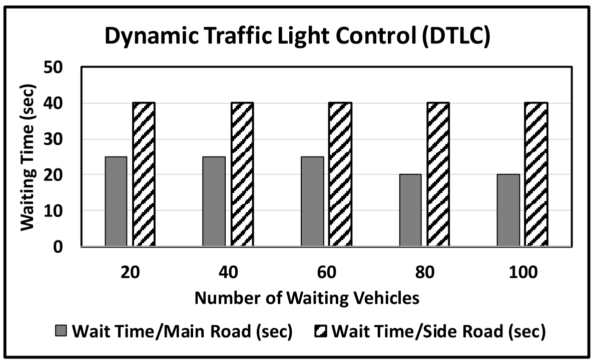



Fig. 10. DTLC fairness (congestion occurs with 70 waiting vehicles). Heavy traffic side gets more green time.

The possible operational cases are as follows.

1) *Light Load:* DTLC does not change the typical TL scheduling, i.e., maintain fairness as in legacy systems.
2) *Heavy Load on Both Arms of the Intersection:* DTLC employs early TL change. Such early TL change will be applied to the two intersecting arms of lights/traffic. Therefore, fairness is also maintained in this case.
3) *Heavy Load on One Arm of the Intersection Only:* DTLC gives early green light to heavily loaded intersection arm (i.e., distributing congestion). Since vehicles use a mix uncongested roads and congested roads, DTLC will typically be fair.

Fairness is demonstrated in Figs. 9 and 10, which show the waiting time (on both sides of the intersection), when either TLC or DTLC is employed. The time-sharing ratios between the main/side road are quite close when DTLC or TLC is employed (5:8 with TLC and 4:8 with DTLC). Thus, DTLC is still reasonably fair although DTLC causes significant throughput gains, as compared to TLC (see Figs. 2 and 5).

## VIII. Related Work

### A. TL Control

Improving traffic flow is a classic problem in transportation systems. Civil engineering research focused on improving road conditions, adding/switching lanes, and controlling TL to give time to main roads and congested road segments.

Previous work on TLC has proposed theoretical methods to optimize traffic flow. Little focus was given to practical control protocols that optimize light transitions based on waiting traffic.

Some legacy systems have been proposed for controlling traffic flow (including SCOOT [22] and WITS [23]). The problem of how to control TLs to reduce congestion has recently been addressed. Kafi *et al.* [18] and Tubaishat *et al.* [20] summarized the general concept and approaches for TL control using wireless sensor networks (this exemplifies the concept that we recently refer to as the "CPS").

Two research approaches that were presented by Neudecker *et al.* [6] and Ferreira *et al.* [2] are close to our work's objective (but differ in the approach). In [6], they studied how to control TLs to facilitate better flow at intersections. Their approach is infrastructure-less (depends on communication devices on vehicles to coordinate/control the TL for more efficiency. The vehicles approaching an intersection elect a "cluster" leader as they approach an intersection, and all leaders across the intersection compete to elect a virtual TL (VTL) leader. The VTL schedules its red/green transitions. The technique, however, is not clear on how a leader is elected, especially if multiple leaders approach the TL at the same time. Thus, we need to design a robust approach that ensures that only one VTL is present at the intersection, and this VTL is the "best" candidate. Ferreira *et al.* [2] presented a decentralized VTL system that has two steps: 1) vehicles agree on a VTL leader and 2) the leader announces the TL schedule to neighboring vehicles. A leaving leader hands over leadership to another vehicle. Such approach is flexible since it is always carried out by the closest vehicle to the TL; one that carries necessary equipment to coordinate with other vehicles. It is not clear how the system ensures that only one TL leader is elected. Reference [3] addresses the question: "to which extent can intervehicle communication improve the wide-area transportation network that consists of several cities?" The vehicular networking community need to extend its perspective toward "information-centric models."

Mueller [5] discussed several security concerns in the power grid, and how immediate research is needed to protect the critical infrastructure to avoid cyber-physical attacks. There is complete absence of security solutions to the control processes in power plants (security in vehicle controls is similar). We lack simulation infrastructure to enable academic contributions to solve these problems. Reference [5] advocates tracking and solving "soft error problems" (a software problem related to control, such as time settings). This is important to guard industries, such as avionics and semi-conductors, from faulty operations. Yousef *et al.* [19] studied how to exploit a WSN to estimate the amount of traffic at one or multiple TLs and make light control decisions based on such estimations. Srinivasan *et al.* [24] proposed a TLC learning-based approach using neural networks, which significantly improves traffic metrics. However, improvement is vulnerable to slow learning process, especially under changing traffic conditions.

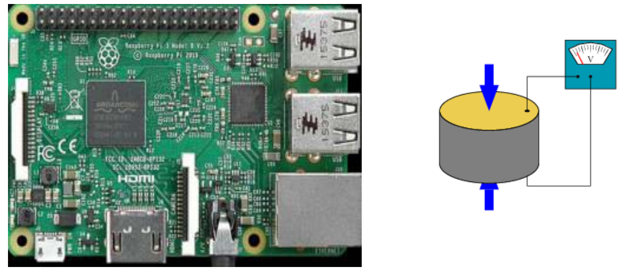
Zhou *et al.* [21] adaptively controlled light to enable better traffic flow through single/multiple intersections. They adjust a green light at a sequence of TLs for better flow. Their control decisions are based on the traffic volume to reduce the number of vehicle stops. Finally, [16] studies adaptive TLs based on car-to-car communication. The goal is to improve traffic fluency and reduce fuel consumption. No study, however, investigated optimal TL control based on the traffic "volume," not traffic "availability." Relying on car-to-car communication is also unreliable for TL control.

### B. CPSs

Several papers have presented the general CPS problems/apps. Tabuada [4] claimed that a major research challenge is to understand how we can adapt the notion of locality induced by a network of embedded systems to the notion of locality induced by the physical substratum. Abdelzaher [7] indicated that several trends impact the future of computer science, such as: 1) Moore's law; 2) the widening human/machine bandwidth gap; and 3) the high cost of lack of communication. These trends motivate the development of CPSs if certain challenges are overcome.

The CPS foundations are discussed in [8]. The claim is that the CPS problem is the intersection of the cyber and physical problems. There are several limitations, [8], [9], including: 1) a critical gap between emerging cyber-physical infrastructure and user environments; 2) existing CPSs have not made the leap from one of a kind experiment to generally useful infrastructure; 3) lack of sufficient flexibility and modularity; and 4) the wealth of new and diverse users who would need better support tools, interfaces, and ease of use. Several problems and research directions in medical applications are also discussed in [9] and [10]. These problems include high assurance software, context awareness, interoperability, autonomy, security, privacy, and certifiability. CPSs are also a promising approach for mission-critical applications (ensuring safety/security/sustainability). Example mission-critical apps include emergency/management applications (e.g., [11] and [12]). The requirements of CPSs in transportation are introduced as the basis for smart transportation systems applications and their development in [30]. The requirements for employing CPSs in smart utilities (e.g., smart grid) are discussed in [31]. NIST studies how to exploit CPSs to serve in solving multiple measurement problems [32]. These problems include: 1) the need for a consensus-based CPS framework and methodology for developing CPS use-cases; 2) the need for a platform for CPS experimentation/validation; and 3) the need to organize technical work among multiple sectors [32].

Researchers believe it is too early to research CPS security issues (we are still studying architecture/development issues). However, we believe we should put the security aspects in the forefront of our research agenda to avoid the major issues that researchers have faced after using insecure Internet protocols. Several studies have worked on security in CPSs, in parallel with research on its architectural design. These studies include [13]–[15]. Recently, research was proposed to support autonomous vehicles at intersections [17].



(a) (b)

Fig. 11. Tentative hardware for the TL control system. (a) Raspberry Pi 3. Tentative candidate for the traffic controller. (b) Piezoelectric sensor (candidate for RSU).

## IX. Supporting Hardware Platform and Future Applications

Our future priority is to determine the hardware platform to carry/support our proposed protocols, i.e., find a suitable controller board to attach to the TL, and suitable sensors (RSUs) to be deployed on road sides. A tentative platform is "Raspberry Pi" [27] [Fig. 11(a)]. The board has a 1.2 GHz 64-bit quad-core ARM Cortex-A53 CPU and includes an integrated 802.11n wireless LAN and Bluetooth 4.1. It communicates wirelessly with the RSUs. It is sold in the range of \$35–\$40/unit, which makes it economical to use on every TL. The board gets a power source from the TL pillar. For the RSUs, we will study the feasibility of deploying one of the sensors that are listed in [26] (depending on their capabilities and cost).

The tentative cost will be around \$100 per RSU. An example RSU can be the piezoelectric sensor (pressure sensor) [27], which measures the changes in pressure. A major advantage of this type of sensor is that it does not need a power source and this facilitates its deployment anywhere on the road, where vehicles pass. However, a problem that requires study in this paper is that the sensor does not carry a wireless communication card. Thus, its output has to be transferred directly (by wire) to the control board or indirectly by connecting it to a separate low-cost wireless radio. Fig. 11(b) depicts a sample piezoelectric sensor operation.

### A. System Cost

We summarize the tentative hardware for the TL control system and give a tentative cost estimate. These are just examples to give a rough estimate of the tentative expenses of the proposed system. A more detailed study will be carried out to determine the best hardware options and to further reduce the cost per road segment (and consequently, the overall cost).

As shown in Table V, the total cost of our proposed solution is about \$440/intersection. Adding the installation cost, software, and profit, the cost will remain in the hundreds of dollars, which is negligible compared to the cost of a TLC (e.g., Siemens m50 Controller [28] is \$250K).

### B. Future CPS Applications: Pedestrian Safety

There is now significant emphasis on legislation that enforces the accommodation of safe walking/bicycling on the

TABLE V
SUGGESTED HARDWARE AND ASSOCIATED COST

| Part | Tentative Cost |
|---|---|
| TLC, e.g., Raspberry Pi 3 [25] | $35.00--$40.00 |
| RSU, e.g., Piezoelectric sensor [27] | $100.00 |
| Total Parts (TLC + 4 RSUs): | $435.00--$440.00 |

road. However, according to the National Highway Traffic Safety Administration, every year tens of thousands of people are killed or injured on the road while walking/bicycling (e.g., in 2010, 4280/70 000 pedestrians were killed/injured, and 620/52 000 bicyclists were killed/injured). The society suffers the pain of losing lives and the efforts/cost to treat them.

For pedestrians/bicyclists, technical road issues are still a cause of accidents. One issue is the blind spot that occurs when a driver makes a right turn while a pedestrian is crossing his way. A second issue is the conflict between a driver and a pedestrian who crosses without a "Pedestrian Clear to Pass" signal. A third issue is when a driver makes a "courtesy wait" for a pedestrian crossing a multiple lane street, causing other cars to hit him.

New road safety control strategies are needed for pedestrians and bicyclists. We propose to use CPSs to "sense" the presence of pedestrians in dangerous spots (especially, those related to open crossing areas). Such detections are then conveyed on clear electronic signs to alert drivers and pedestrians. New work should focus on proposing a new architecture for controlling the road safety, which extends the function of the current traffic and pedestrian lights. We have shown how CPSs can help in TL control to improve the flow of traffic at intersections. Integrating this paper with potential research on road safety will provide a comprehensive set of solutions for enhancing the safety on the road while improving the traffic flow.

## X. CONCLUSION

We studied a promising CPS application (traffic flow control). We pinpointed the need to enable DTLC at road intersections. We studied the possible design approaches for enhancing the traffic flow, and proposed two protocols to achieve this goal. Our techniques rely on input from a sensor network on the road, and make control decisions either centrally at the TL, or in a distributed way using vehicle devices. Analysis shows that our techniques incur low overhead (communication and processing). Simulations show that our protocols provide significant benefits to the traffic flow metrics, such as vehicle throughput, waiting time, and waiting line length. Our future plan is to study using CPSs for traffic safety and health issues, e.g., avoiding accidents/reducing pollution.

## REFERENCES


[1] E. A. Lee, "Cyber-physical systems—Are computing foundations adequate?" in *Proc. NSF Workshop Cyber Phys. Syst. Res. Motivation Tech. Roadmap*, Austin, TX, USA, Oct. 2006, pp. 26–34.

[2] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," in *Proc. 7th ACM Int. Workshop Veh. Internetw. (ACM VANET)*, Chicago, IL, USA, Jun. 2010, pp. 85–90.

[3] T. Gaugel, F. Shmidt-Eisenlohr, J. Mittag, and H. Hartenstein, "A change in perspective: Information-centric modeling of inter-vehicle communication," in *Proc. 8th ACM Int. Workshop Veh. Internetw. (ACM VANET)*, Las Vegas, NV, USA, Sep. 2011, pp. 61–66.

[4] P. Tabuada, "Cyber-physical systems: Position paper," Dept. Elect. Eng., Univ. California at Los Angeles, Los Angeles, CA, USA, Tech. Rep. UCLA-TR-2012, 2012.

[5] F. Mueller, "Challenges for cyber-physical systems: Security, timing analysis and soft error protection," in *Proc. Nat. Workshop High Confidence Softw. Platforms Cyber Phys. Syst. Res. Needs Roadmap (HCSP CPS)*, Nov. 2006, pp. 18–20.

[6] T. Neudecker, N. An, O. K. Tonguz, T. Gaugel, and J. Mittag, "Feasibility of virtual traffic lights in non-line-of-sight environments," in *Proc. 9th ACM Int. Workshop Veh. Internetw. Syst. Appl. (ACM VANET)*, Jun. 2012, pp. 103–106.

[7] T. Abdelzaher, "Toward an architecture for distributed cyber-physical systems," Dept. Comput. Sci., Univ. Illinois at Urbana–Champaign, Champaign, IL, USA, Tech. Rep. UIUCTR-2011, 2011.

[8] E. A. Lee, "CPS foundations," in *Proc. ACM 47th Design Autom. Conf. (DAC)*, Anaheim, CA, USA, Jun. 2010, pp. 737–742.

[9] L. Zhang, J. He, and W. Yu, "Challenges, promising solutions and open problems of cyber-physical systems," *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 2, pp. 65–74, Mar. 2013.

[10] I. Lee *et al.*, "Challenges and research directions in medical cyber-physical systems," *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 2, pp. 75–90, 2013.

[11] A. Baberjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber–physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 283–299, Jan. 2012.

[12] E. Gelenbe and F.-J. Wu, "Future research on cyber-physical emergency management systems," *Future Internet*, vol. 5, no. 3, pp. 336–354, 2013.

[13] A. A. Cardenas *et al.*, "Challenges for securing cyber physical systems," in *Proc. Workshop Future Directions Challenges Securing Cyber Phys. Syst.*, Jul. 2009, pp. 543–552.

[14] E. K. Wang *et al.*, "Security issues and challenges for cyber physical system," in *Proc. IEEE/ACM Int. Conf. Cyber Phys. Soc. Comput.*, Hangzhou, China, 2010, pp. 733–738.

[15] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1454–1467, Jun. 2014.

[16] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *Proc. IEEE Veh. Technol. Conf.*, Dublin, Ireland, Apr. 2007, pp. 21–25.

[17] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige, "STIP: Spatio-temporal intersection protocols for autonomous vehicles," in *Proc. IEEE/ACM Int. Conf. Cyber Phys. Syst.*, Berlin, Germany, Apr. 2014, pp. 1–12.

[18] M. A. Kafi *et al.*, "A study of wireless sensor network architectures and projects for traffic light monitoring," *Proc. Comput. Sci.*, vol. 10, pp. 543–552, Dec. 2012.

[19] K. M. Yousef, J. N. Al-Karaki, and A. M. Shatnawi, "Intelligent traffic light flow control system using wireless sensor networks," *J. Inf. Sci. Eng.*, vol. 26, no. 3, pp. 753–768, 2010.

[20] M. Tubaishat, Y. Shang, and H. Shi, "Adaptive traffic light control with wireless sensor networks," in *Proc. IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, 2007, pp. 187–191.

[21] B. Zhou, J. Cao, and H. Wu, "Adaptive traffic light control of multiple intersections in WSN-based ITS," in *Proc. 73rd IEEE Veh. Technol. Conf. (VTC)*, Yokohama, Japan, 2011, pp. 1–5.

[22] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time-the SCOOT method," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 11–15, Feb. 1991.

[23] W. Chen, L. Chen, Z. Chen, and S. L. Tu, "WITS: A wireless sensor network for intelligent transportation system," in *Proc. 1st Int. Multi Symp. Comput. Comput. Sci. (IMSCCS)*, vol. 2. Hangzhou, China, 2006, pp. 635–641.

[24] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 261–272, Sep. 2006.

[25] (Feb. 2016). *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/wp-content/uploads/2016/02/IMG_4090.jpg

[26] (May 2016). *Wireless Sensors*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes

[27] (May 2016). *Piezoelectric Sensor*. [Online]. Available: https://en.wikipedia.org/wiki/Piezoelectric_sensor

[28] (2017). *Siemens m50 Series Controller*. [Online]. Available: http://w3.usa.siemens.com/mobility/us/en/urban-mobility/road-solutions/on-street-equipment/controllers/papges/m50-controllers.aspx

[29] (Oct. 2017). *Smart City*. [Online]. Available: https://en.wikipedia.org/wiki/Smart_city

[30] D. P. F. Möller and H. Vakilzadian, "Cyber-physical systems in smart transportation," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Grand Forks, ND, USA, Aug. 2016, pp. 776–781.

[31] S. K. Majhia, G. Patrab, and S. K. Dhal, "Cyber physical systems & public utility in India: State of art," in *Proc. Int. Conf. Inf. Security Privacy (ICISP)*, Nagpur, India, Dec. 2015, pp. 777–781.

[32] *Cyber Physical Systems Program Research Program*, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Jun. 2017.


**Ossama Younis** (S'03–M'05) received the Ph.D. degree in computer science from Purdue University, West Lafayette, IN, USA, in 2005.

He is a Chief Scientist with Smart Streets LLC, Rockville, MD, USA. He was a Computer Scientist with the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, from 2014 to 2016. He was a Senior Research Scientist with Telcordia Technologies (AppComSci), Piscataway, NJ, USA, Applied Research, Woodbridge, NJ, USA, from 2007 to 2014, and an Assistant Research Scientist with the University of Arizona, Tucson, AZ, USA, from 2005 to 2007. His current research interests include cyber-physical systems, sensor networks, cognitive-radio networks, and Internet protocols.

Dr. Younis is a member of the ACM.

**Nader Moayeri** (S'79–M'80–SM'90) received the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1986.

He was with the Imaging Technology Department, HP Labs, Palo Alto, CA, USA. He joined the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, in 1997, where he established the Wireless Communication Technology Group, in 1997, and managed it until 2008. He is involved in basic and applied research and test and evaluation activities to develop modern standards for wireless communications. Since 2002, he has been researching mission-critical communications/networking for various applications. From 1986 to 1994, he was with the faculty of the Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA. His current research interests include indoor localization and wireless ad hoc/mesh/sensor networks.