Cyber-Physical Systems: A Framework for Dynamic Traffic Light Control at Road Intersections

Ossama Younis and Nader Moayeri
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

Abstract— **Traffic control at road intersections is based on traffic lights (TLs). The control mechanism typically used for traffic lights operates based on a periodic schedule to change the light (red/yellow/green). In many cases, a different schedule is used in late night/early morning hours. This fixed control mechanism does not adequately account for changing traffic conditions, and is unaware of/unresponsive to congestion. In this work, we propose a framework for dynamic traffic light control at intersections. The framework relies on a simple sensor network to collect traffic data and includes novel protocols for traffic flow control to handle congestion and facilitate flow. We show that our proposed algorithms have low overhead and are practical to employ in live traffic flow scenarios. Through extensive simulations, we demonstrate the benefits of our framework in optimizing traffic flow metrics, such as traffic throughput, average vehicle waiting time, and vehicle waiting line length.**

Keywords-Traffic light control; traffic flow optimization; distributed algorithms; sensor networks.

I. INTRODUCTION

Cyber-Physical Systems (CPS) have been recently proposed to define how the computing world interacts with and manipulates the physical world. Researchers are developing CPS for application domains such as environmental monitoring, smart homes, and smart manufacturing. While the CPS idea is rising, several questions require answers. For example, it is not clear if the computing foundations are adequate for supporting and using CPS [1]. Some argue that CPS requires top-to-bottom rethinking of computation.

Improving traffic flow and handling congestion are classic problems in transportation systems. Research in Civil Engineering has focused on improving the road conditions, adding/switching traffic lanes, and controlling traffic lights to improve traffic flow particularly on main/congested roads. New research focuses on utilizing CPS to optimize traffic flow and improve the driver's experience on the road. Most previous research has shown the "benefits" of using controlled traffic lights. However, practical protocols are still needed to control lights and optimize traffic flow.

In this work, we present a framework for optimizing traffic flow using dynamic traffic lights. Our framework includes collecting road conditions by sensors deployed in road side units (RSUs) and using algorithms to decide on when to change traffic lights to alleviate congestion. Our proposed techniques are fast, simple, low-overhead, and thus can be easily deployed. Through analysis and simulations, we demonstrate the benefits of the proposed framework and protocols.

The rest of this paper is organized as follows. Section II describes the system/road model and explores the possible design approaches for traffic light control. Sections III and IV present our solutions for traffic flow enhancement. Section V analyzes the proposed approaches: describes the overhead (or complexity), outlines the possible triggers to change the traffic lights ahead of their typical schedule, and discusses how our proposed approaches extend to global traffic light control (i.e., in a region or town). Section VI evaluates the proposed algorithms in light/heavy traffic scenarios. Section VII overviews related research on traffic flow control. Finally, Section VIII concludes this work.

II. SYSTEM MODEL AND DESIGN APPROACHES FOR DYNAMIC TRAFFIC LIGHT CONTROL

A. *Road/Traffic Model*

We assume the following about the traffic pattern, the vehicle capabilities, and the road intersection:

- **Road:** Typical model with intersecting roads. A road has two opposite segments, each allowing traffic to go along one or more lanes.
- **Traffic lights:** Installed at the head of each road segment. The traffic lights at two opposite/intersecting segments always show the same/opposite colors.
- **Traffic model:** Vehicles arrive according to an arbitrary stochastic model. We propose traffic light control techniques that are independent of the arrival model. We assume that all vehicles at the light want to proceed forward, and leave left/right turns for future work.
- **Vehicle detection:** By exploiting either RSU sensors or vehicle-installed devices, detect/report the vehicles that arrive at the intersection on each of the four segments.

B. *Objectives*

Develop protocol(s) to control/optimize the traffic flow. These protocols collect traffic flow information and utilize it to optimize traffic flow metrics, such as throughput, delay, or waiting line length. The protocols should have low communication overhead and fast decision-making to be practically applicable on the road. The cost of the proposed protocols should also be minimal to motivate adoption and deployment. This is why it is desirable to propose solutions for deployment and operation. Such solutions may trade one performance metric for another.

C. *Design Approaches*

We categorize the design of dynamic traffic control mechanisms into two possible approaches:

Design Approach 1: Deploy sensor infrastructure in RSUs to collect information about the road conditions (e.g., the number of vehicles waiting for a green light). Also deploy a traffic light controller (TLC) at the traffic light to control its state based on an algorithm that uses information from the RSUs. Alternatively, the infrastructure can be sensors placed at road lanes. This design option assumes that the sensors are able to communicate their data to the TLC. We name this approach Road-based Infrastructure Traffic-light Control (**RITCO**).

Design Approach 2: Use distributed infrastructure, e.g., (1) a traffic light controller (TLC) at the traffic light, and (2) vehicle communication devices to report the vehicles' arrivals and locations (road segments) to the control unit (CU). 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 Traffic-light Control (**VITCO**).

D. *Which Design Approach to Select*

Several issues need to be considered in order to adopt a particular design:

- **Complexity/Cost:** Evaluate processing/communication complexity, in addition to cost. Designs with significant processing can be acceptable if: (1) the computing devices are expected to be capable of rapidly carrying out the computations, and (2) the processing complexity may reduce the communication complexity.
- **Applicability:** Describes which mechanism is more practical in the transportation road network model, and therefore, is more applicable. This involves studying effectiveness, ease of deployment, and cost.
- **Performance metrics:** The metrics are defined in Section VI.A (e.g., traffic throughput/waiting time). The performance of the proposed technique(s) is a major factor in deciding which design approach is more suitable for the traffic model that we optimize.

III. ROAD-BASED INFRASTRUCTURE TRAFFIC-LIGHT CONTROL (RITCO)

The RITCO protocol employs "Design approach 1" in Section II.C. It requires:

- **Traffic light controller (TLC):** (1) Co-located with the traffic light and (2) collects information about the current road conditions from RSUs and uses it to make light control decisions.
- **Sensors at RSUs:** Each road segment is equipped with an RSU installed several meters away from the intersection to detect vehicle arrivals. Assuming that the traffic light is going to change to red at time T_{Red}, each of the two RSUs installed on segments with the red light is notified of the impending change at time (T_{Red}-Δt), where Δt is a small positive number (interval), to start counting the vehicles that will be waiting at the TL. The counted number of vehicles is periodically reported to the TLC. Note that if multiple sensors are used in each segment, then coordination is needed to avoid counting a vehicle multiple times. Using a single sensor eliminates the communication overhead. However, one sensor may not provide sufficient coverage to detect all vehicles.

A. *The RITCO Protocol*

Table 1 lists the pseudo-code of the RITCO protocol. $T_{curr} \equiv$ current time, $\Delta t \equiv$ very small time interval, $T_{red}/T_{yellow}/T_{green} \equiv$ Red/Yellow/Green light intervals, $T_{\underline{change}} \equiv$ time until light changes to red/green, numVehicles ≡ number of vehicles, VID ≡ Vehicle ID (a unique ID, e.g., license plate number).

Table 1. The RITCO Protocol Details

```
Steps at TLC (Current Light Green→Change to Red):
•  Initialize: numVehicles←0, switchLightFlag←OFF
•  TLC announces impending TL change to red to RSUs.
•  While (($T_{curr}+\Delta t \geq T_{red}$) && (switchLightFlag is OFF))
       o  Collect sensor (RSU) readings
       o  If (m new vehicles are detected at RSU)
            ▪ numVehicles ← numVehicles+m
            ▪ Save VID's to vehicle list
       o  UpdateFlag(switchLightFlag)
•  End While
•  If (switchLightFlag is ON)
       o  Perform TL_Change  (Red→Green)
•  The TLC announces impending TL change to RSUs.

Steps at Sensor (RSU on Pole or Road Lane):
•  Initialize: Vehicle counter (numVehicles ← 0).
•  RSU prepares to activate the sensing process:
       o  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)
       o  numVehicles ← numVehicles+1
•  Every k seconds, RSU checks:
       o  If (numVehicles > 0)
            ▪ Report numVehicles to the TLC
            ▪ RSU resets its numVehicles to 0.
•  Based on the 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)
       flag ← ON
•  Else
       flag ← OFF
```

IV. VEHICLE-BASED INFRASTRUCTURE TRAFFIC-LIGHT CONTROL (VITCO)

The VITCO protocol employs “Design Approach 2” in Section II.C. It requires:

- **Traffic light controller (TLC):** (1) Co-located with the traffic light, and (2) interprets/executes the decisions made by the VITCO protocol for light control.
- **Vehicle Radio:** A vehicle device that is used to announce vehicle’s arrival at the traffic light. The radio executes what was being done by the RSUs in RITCO, thus participating in the coordination process.

A. *The VITCO Protocol*

Table 2 provides the pseudo-code of the VITCO protocol (executed at every road segment). Symbol definitions are similar to those in Section III.A.

Table 2. The VITCO Protocol Details

VITCO: Elect_a Coordinating Vehicle (Elect_CoVe):

```
• A vehicle $V_1$ arrives/stops at the TL.
• $V_1$ transmits a message for CoVe and neighbor discovery
  and sets an expiration timer $T_1$.
• If a response message announcing CoVe ID arrives
  (from a neighbor), cancel $T_1$ and find the path to that
  CoVe.
• If $T_1$ expires and no CoVe announcement has arrived,
  broadcast $V_1$ as a new CoVe.
• If later, $V_1$ receives a CoVe broadcast (after announcing
  itself as CoVe), arbitrate to pick the best CoVe. If
  selected, rebroadcast a CoVe announcement.
• If the other CoVe is better, then $V_1$ broadcasts a NON-
  CoVe message, including the other CoVe ID.
```

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

```
• Vehicle $V_i$ arrives at the traffic light.
• $V_i$ discovers its neighbors and CoVe (Elect_CoVe).
• Assume the elected CoVe is $V_c$.
• numVehicles ← 0.
• While (TL_Timer has not expired)
    o If ($V_i$ is $V_c$) Then
          Broadcast a CoVe announcement
          When (message_arrives_from a vehicle)
             If (new vehicle)
             Then   Increment numVehicles
                    sendMsgToTLC(False, numVehicles)
             EndIf
          UpdateFlag(switchLightFlag)
          If (switchLightFlag=ON)  Then
             SendMsgToTLC(True, numVehicles)
             When the TLC confirms (TL changes)
                   Exit
          EndIf
    o EndIf
    o Else          // ($V_i$ is not $V_c$)
          $V_i$ registers $V_c$ as its CoVe
          $V_i$ discovers the communication path to $V_c$ (if not
          single hop)
    o EndIf
• EndWhile
```

Note that the function Veh_SendMsgToTLC translates to function TLC_RecvMsg at the TLC (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
```

V. PROTOCOL ANALYSIS

A. *Properties/Overhead*

(1) Communication: RITCO is independent of the vehicles on the road, i.e., requires no communication between the vehicles and the infrastructure. VITCO has more communication cost.

(2) Processing: RITCO assigns the work to road infrastructure (RSUs or devices attached to the traffic light). VITCO, on the other hand, puts most of the processing work on the Coordinating Vehicle (CoVe), and communication work on all vehicles.

(3) Cost: RITCO puts all the burden on the city/county (road infrastructure & traffic light control equipment). On the other hand, VITCO shares the cost between the city/county and the vehicle owners.

(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 that may occur due to faulty vehicle equipment. This, however, requires much more cost than using VITCO. RITCO also avoids the transition period in which 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 from the vehicles (just counts them autonomously). 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 (VIN).

(6) Complexity: Let N: the max number of vehicles waiting for the TL change, L: Number of lanes on the road. Thus, message exchange complexity/overhead per vehicle: O(N/L); messages to broadcast traffic light change: O(1); processing complexity: O(N) to register new vehicles. To decide whether/when to change TL: Can be as low as O(1) for heuristic-based algorithms.

B. *Triggers to change traffic lights synchronously or asynchronously*

The current traffic light transitions use fixed (pre-assigned) intervals for the different lights. Recently, in some areas, some forms of adaptive light transitions were employed. For example, prolonging the green light at main streets until a vehicle arrives at a side street. In this work, we consider a broader scope for early transitioning, and propose a detailed approach for adaptive lights. Two ways can change the TL (red→green or green→red):

- **Synchronously:** According to a timer expiration (which is being employed on the roads now).
- **Asynchronously:** In addition to waiting for timer expiration, the TL is changed if certain events trigger the need for light change.

Triggers for asynchronous 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. Note that such number of waiting cars can be considered on a single side of the TL or collectively on both sides of the TL. Considering the waiting vehicles on each side separately is more practical to avoid having a very long waiting line on any side.

2) No vehicles have crossed the intersection from the sides that have the green light for a certain time interval.

C. *Regional/ global traffic light control*

The framework (presented methods in Section III and Section IV) presents protocol algorithms for "local" traffic control at a single traffic light. Based on the reactive/independent nature of the proposed algorithms, the framework inherently addresses global traffic optimization. The reaction to congestion at a traffic light shifts the congestion problem toward other traffic lights, thus continuously trying to distribute the traffic load and relieve congested areas.

VI. EVALUATION OF PROPOSED APPROACHES

We evaluate the performance of the proposed Dynamic Traffic Light Control (DTLC) approach, in comparison to the legacy traffic light control (TLC). The evaluation does not distinguish between the two flavors of DTLC (RITCO and VITCO), because they differ only in the details of collecting traffic information but not in the TL control details. We first define the metrics that are used to evaluate our protocols, then we describe the experiments in detail.

A. *Metrics/Parameters*

We consider the following evaluation metrics: waiting time, waiting line length, and traffic throughput (definitions are given in Table 3):

Table 3. The metrics to evaluate DTLC

Metric	Definition
Average/Maximum waiting line length (Q_i)	At each side of the intersection, the average/maximum (number of vehicles per lane) among all lanes.
Average/Maximum waiting time (W_i)	The average/max time that a vehicle waits for change of the traffic light (red→green).
Throughput (T)	The number of vehicles that pass the intersection per unit time.

In our experiments, we vary two parameters: (1) number of arriving vehicles at the TL, and (2) number of TL cycles (multiple red-green transitions). The rest of the parameters are: #lanes=2, lane size=100 vehicle, vehicles arrival rate=90/lane, threshold of waiting vehicles=75, max# passing vehicles=150, red/green interval=20 sec, yellow interval=2 sec. Each result is the average of 10 random experiments.

B. *Vary the Number of Arriving Vehicles*

In the following experiments, we evaluate the different metrics when the number of arriving vehicles at the TL varies (the probability distribution of vehicle arrivals does not matter; only the number of arrivals matter). Figure 1 shows the throughput of the intersection point when the number of arriving vehicles varies from 20 to 100. The legacy TL system does not react to increasing traffic, while DTLC reacts to congestion by early switching the TL if the number of waiting vehicles exceed a threshold (e.g., 75). This is why significant throuhput gains are achieved by DTLC when the number of arriving vehicles exceeds such threshold. Figure 2 shows the number of waiting vehicles for green light. In DTLC, the waiting number of vehicles is upper-bounded by a limit, and early light switch is triggered when such limit is reached. Since this action is taken at both (intersecting) parts of the traffic light, this ensures that the waiting line never overflows and also that congestion is early handled. For the waiting time/vehicle, Figure 3 shows that vehicles wait slightly less when the DTLC approach is employed.

C. *Vary the number of TL cycles (red-green interval)*

We evaluate DTLC when the number of TL cycles varies (a cycle is a red light interval followed by a green light interval). The number of waiting vehicles is greater than 80. Figure 4 shows 250% traffic throughput improvement under heavy load. Figure 5 shows that the waiting line length is about 20% less with DTLC than the legacy TLC. Figure 6 illustrates that DTLC has less vehicle waiting time. This is intuitive, especially under heavy traffic that causes early light change.

D. *Forced early TL change/Fairness*

Experiments show that under high traffic load (i.e., exceeding the threshold), the TL change is forced at every cycle. Fairness is also achieved; i.e., the waiting time is similar for vehicles in corresponding locations from the traffic light.

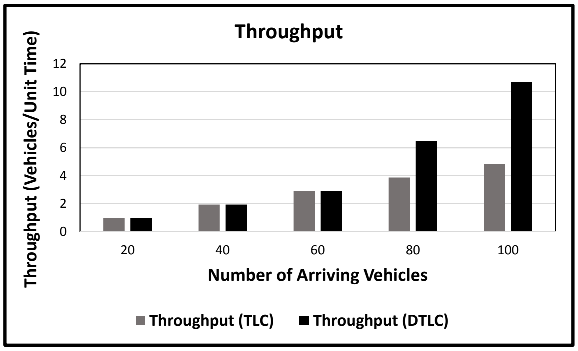

Figure 1. Throughput vs. Arriving Vehicles

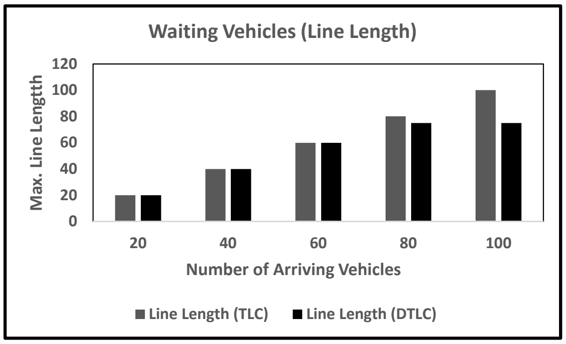

Figure 2. Waiting Line Length vs. Arriving Vehicles

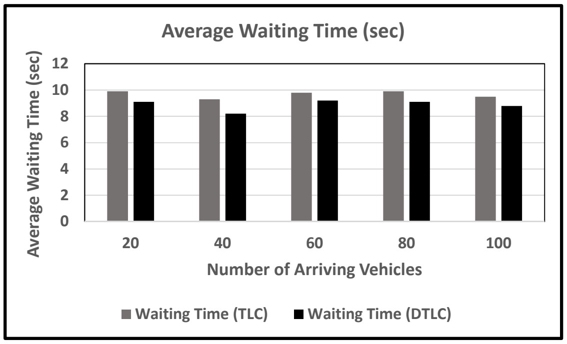

Figure 3. Waiting Time vs. Arriving Vehicles

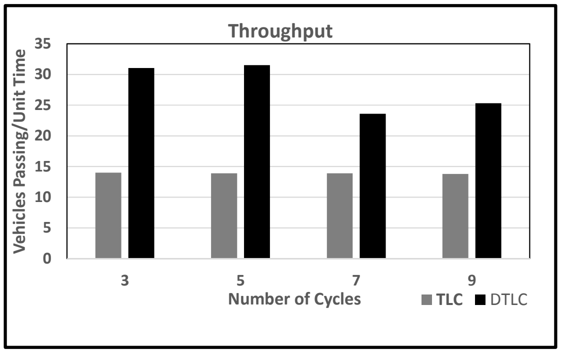

Figure 4. Throughput (Passing Vehicles/Unit Time) vs. the Number of Cycles

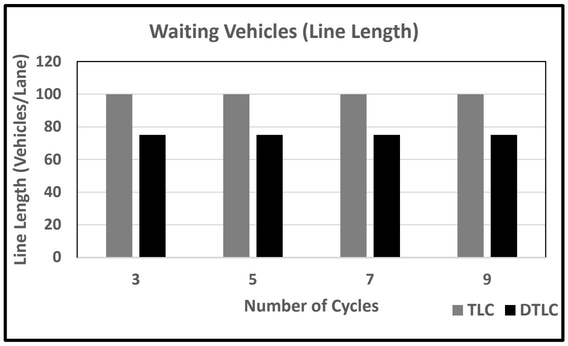

Figure 5. Waiting Line Length vs. the Number of Cycles

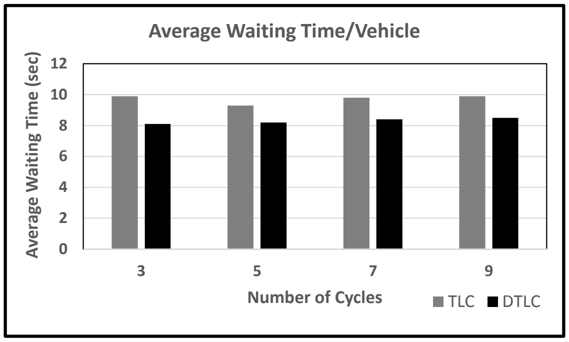

Figure 6. Waiting Time vs. the Number of Cycles

VII. RELATED WORK

The problem of how to control traffic lights to reduce congestion has been recently addressed in the literature. Two research approaches (Ferreira et al. [2] and Neudecker et al. [5]) are close to our work's objective and model, but differ in the solution. In [5], the authors addressed how to control traffic lights to facilitate better flow of traffic at an intersection. The vehicles on each road segment elect a leader, and all the leaders across the intersection compete to elect a virtual traffic light (VTL) leader. The technique, however, is not clear on how a leader is elected. It is important to design an approach that ensures that only one VTL leader is present at the intersection, and this VTL leader is the best candidate. Ferreira et al. [2] presented a virtual traffic light system. Their decentralized approach has two steps: (1) vehicles agree on a virtual traffic light leader at an intersection, and (2) the leader takes the role of a temporary virtual infrastructure and informs the neighboring vehicles of the traffic light schedule. When a leader leaves the intersection, it hands over the leadership to another vehicle. This approach is quite flexible, but it is not clear how the system ensures that a traffic light leader is elected (and how to ensure only one leader is elected). The work in [3] addresses the question: "to which extent can inter-vehicle communication improve wide area transportation network that consists of several cities?" Finally, [6] studies adaptive traffic lights based on car-to-car communication.

Several position papers have also presented the general problems of CPS and their applications. Tabuada [7] claims 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. Abdelzaher [8] indicates that several trends impact the future of computer science, such as: (1) Moore's law, (2) widening human/machine bandwidth gap, and (3) high cost of lack of communication. These trends motivate the CPS development if challenges are overcome.

The CPS foundations are discussed in [9]. The claim is that the CPS problem is not the union of the cyber and physical problems, but rather the intersection. However, there are several limitations [10], including, (1) a critical gap between the emerging cyber-physical infrastructure and user environments, (2) existing CPS have not yet 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/research directions in medical applications are also discussed in [11] and [12]. These problems include high assurance s/w, interoperability, context awareness, autonomy, security/privacy, and certifiability. CPS are also a promising approach to serve mission-critical applications (ensuring safety, security, and sustainability). Example mission-critical applications include emergency and management applications [13], [14]. In [4], Mueller discusses several security concerns in the power grid, and how immediate research is needed to protect the critical infrastructure to avoid cyber-physical attacks. Some researchers believe that we should put the security aspects in the forefront of our research objectives to avoid the major issues that researchers have faced after deployment/use of insecure Internet protocols ([15], [16], [17], and [18]). Finally, recent research has proposed protocols at road intersections to support autonomous vehicles [19].

VIII. CONCLUSION

In this work, we studied one important CPS application (traffic flow enhancement). We focused on what is needed to enable dynamic traffic light control for facilitating better traffic flow through intersections. Then, we studied the possible design approaches and proposed two protocols to achieve this goal. Our proposed techniques rely on input data from a sensor network on the road, and make control decisions either centrally at the traffic light, or in a distributed way using devices on the vehicles. Our analysis shows that our proposed techniques require low overhead in communication and processing. Simulations show that our proposed techniques provide significant benefits to the traffic flow metrics, especially in terms of vehicle throughput. Our future research plan is to study the use of CPS for traffic safety and pollution reduction. We also plan to study the effect of pedestrians and right/left turns on the performance of traffic flow at intersections.

REFERENCES

[1] E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?" NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques, and Roadmap, Austin, TX, Oct. 2006.

[2] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, O. K. Tonguz, "Self-organized traffic control," in Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking (ACM VANET), Chicago, IL, June 2010, pp. 85-90.

[3] T. Gaugel, F. Shmidt-Eisenlohr, J. Mittag, H. Hartenstein, "A Change in Perspective: Information-Centric Modeling of Inter-Vehicle Communication," in Proceedings of the Eighth ACM International Workshop on VehiculAr InterNETworking (ACM VANET), Nevada, Sep. 2011.

[4] F. Mueller, "Challenges for Cyber-Physical Systems: Security, Timing Analysis and Soft Error Protection," in Proceedings of the National Workshop on High Confidence Software Platforms for Cyber-Physical Systems: Research Needs and Roadmap (HCSP-CPS), Nov 2006.

[5] T. Neudecker, N. An, O. K. Tonguz, T. Gaugel, J. Mittag, "Feasibility of virtual traffic lights in non-line-of-sight environments," in Proceedings of the Ninth ACM international Workshop on VehiculAr InterNETworking, Systems, and Applications (ACM VANET), United Kingdom, June 2012, pp. 103-106.

[6] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive Traffic Lights Using Car-to-Car Communication," in Proceedings of the IEEE Vehicular Technology Conference, Dublin, April 2007.

[7] P. Tabuada, "Cyber-Physical Systems: Position Paper," UCLA Technical Report (UCLA-TR-2012), 2012.

[8] T. Abdelzaher, "Toward an Architecture for Distributed Cyber-Physical Systems," Technical Report (UIUC-TR-2011), 2011.

[9] E. A. Lee, "CPS Foundations," in the ACM 47th Design Automation Conference (DAC), June 2010, pp. 737-742.

[10] R. Campbell, G. Garnett, and R. E. McGrath, "Cyber-Physical Systems: Position Paper," in NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques, and Roadmap, Austin, TX, Oct. 2006.

[11] L. Zhang, J. He, and W. Yu, "Challenges, Promising Solutions and Open Problems of Cyber-Physical Systems," in the International Journal of Hybrid Information Technology, Vol. 6, No. 2, March 2013.

[12] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer, and K. Venkatasubramanian, "Challenges and Research Directions in Medical Cyber-Physical Systems," International Journal on Hybrid Information Technology, Vol. 6, No. 2, 2013.

[13] A. Baberjee, K. Venkatasubramanian, T. Mukherjee, S. Kumar, and S. Gupta, "Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems," Proceedings of the IEEE, Vol. 100, No. 1, January 2012.

[14] E. Gelenbe and F.-J. Wu, "Future Research on Cyber-Physical Emergency Management Systems," Future Internet, Vol. 5, 2013, pp. 336-354.

[15] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for Securing Cyber Physical Systems," in Proceedings of the Workshop on Future Directions in Challenges for Securing Cyber Physical Systems, July 2009.

[16] C. Neuman, "Challenges in Security for Cyber-Physical Systems," in the DHS Workshop on Future Directions in Cyber Physical Systems Security, July 2009.

[17] E. K. Wang, Y. Ye, X. Xu, S. M. Yiu, L. C. K. Hui, and K. P. Chow, "Security Issues and Challenges for Cyber Physical System," in the IEEE/ACM International Conference on Cyber, Physical, and Social Computing, 2010.

[18] H. Fawzi, P. Tabuada, S. Diggavi, "Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks," in the IEEE Transactions on Automatic Control, Vol. 59, No. 6, June 2014.

[19] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "STIP: Spatio-Temporal Intersection Protocols for Autonomous Vehicles," in IEEE/ACM International Conference on Cyber-Physical Systems, Germany, Apr. 2014, pp. 1-12.