Adaptive Traffic Light Control with Wireless Sensor Networks

Malik Tubaishat, Yi Shang and Hongchi Shi
Department of Computer Science
University of Missouri - Columbia
Columbia, MO 65211-2060
Email: {mma4n6, yshang, shih}@missouri.edu

Abstract – In this paper, we propose a novel decentralized traffic light control using wireless sensor network. The system architecture is classified into three layers; the wireless sensor network, the localized traffic flow model policy, and the higher level coordination of the traffic lights agents. The wireless sensors are deployed on the lanes going in and out the intersection. These sensors detect vehicles' number, speed, etc. and send their data to the nearest *Intersection Control Agent (ICA)* which, determines the flow model of the intersection depending on sensors' data (e.g., number of vehicles approaching a specific intersection). Coping with dynamic changes in the traffic volume is one of the biggest challenges in intelligent transportation system (ITS). Our main contribution is the real-time adaptive control of the traffic lights. Our aim is to maximize the flow of vehicles and reduce the waiting time while maintaining fairness among the other traffic lights. Each traffic light controlled intersection has an intersection control agent that collects information from the sensor nodes. An intersection control agent manages its intersection by controlling its traffic lights. Multiple intersection agents can exchange information among themselves to control a wider area.

I. Introduction

We envision a smart road system were the total trip time is minimum due to minimizing the average waiting time on traffic lights. In addition to minimizing the average traffic waiting time, we would like to see a road system which can optimize the traffic flow by utilizing the free roads. Tremendous amount of time and power is wasted due to a green traffic light with no cars passing on its lane.

Many solutions were proposed to solve the traffic jam. Most conventional traffic surveillance systems use intrusive sensors, including inductive loop detectors, micro-loop probes, and pneumatic road tubes. However, these sensors disrupt traffic during installation and repair, which leads to a high cost installation and maintenance. In addition, over the ground sensors like videos, radars, and ultrasonic were used. These systems are also high cost and their accuracy depends on environment condition [1]

This paper presents a real-time adaptive system based on wireless sensors that has the potential to establish a new era of traffic control and surveillance because of its low cost and potential for large scale deployment. Our system consists, mainly, of the wireless sensor network and the intersection control agents. The wireless sensor network composed of group of nodes, each comprising one or more sensors, a processor, a radio and a battery. They generate traffic information such as number of cars, speed and length of the vehicles, based on processing of the sensor data. The information is then sent to the nearest intersection control agent over the radio. The intersection control agent collects the information from the sensor nodes to analyze traffic conditions and take actions such as adjusting the traffic light durations or exchanging information with other intersection agents for better optimization of traffic flow.

In the field of Multiagent Systems (MAS) [2][3], controlling intersections is studied with intelligent system on mind. Although these systems has the potential to revolutionize traffic surveillance they are still far from being adopted by the ITS.

Using wireless sensor network along with intelligent transportation system is still in its preliminary stage. To compete with current technologies, however, the data provided by the system must be accurate, delivered to the traffic intersection agents within a certain time for real-time applications, and the lifetime of the system must be on the order of several years.

We use Green Light District (GLD) [1] simulator [4] to test our model. GLD allows us to create road maps and add our own intersection traffic flow policy.

In the next section, we briefly review some approaches that coordinate traffic lights. Section 3 presents wireless sensor networks model. Section 4 discuses intersection control agent. In section 5 discuss the simulation and we conclude in section 6.

II. Related Work

To replace the costly and high maintenance classic traffic surveillance such as inductive loops, Cheung et al. [1] built a traffic surveillance technology system based on wireless sensors. Their system is deployed in freeways and at intersections for traffic measurements such as vehicle count, occupancy, speed, and vehicle classification which can't be

[1] GLD is an open-source software and can be downloaded from {http://sourceforge.net/projects/stoplicht/}

1-4244-0667-6/07/$25.00 © 2007 IEEE

obtained from standard inductive loops. The experiments in [1] shows that deploying wireless sensor network for traffic monitoring provides %99 of detection rate in real time.

Using wireless sensor network for transportation applications provides measurements with high spatial density and accuracy. A network of wireless magnetic sensors [1] offers much greater flexibility and lower installation and maintenance costs than loop, video or radar detector systems.

Chen et al. [5] propose a prototype of Wireless sensor network for Intelligent Transportation System (WITS). WITS system is used for the information gathering and data transferring. In this system three types of WITS nodes are used; 1) the vehicle unit on the individual unit, 2) the roadside unit along both sides of road, and 3) the intersection unit on the intersection. The vehicle unit measures the vehicle parameters and transfers them to the roadside units. The roadside unit gathers the information of the vehicles around, and transfers it to the intersection unit. The intersection unit receives and analyzes the information from other units, and passes them to the strategy sub-system, which in turn calculates an appropriate scheme according to the preset optimization target (such as maximum throughput, minimum waiting time, etc.) Mainly, the intersection unit wants to know how many vehicles in every lane will reach the intersection before the signal phase ends. But there is no enough discussion about how this information helps the intersection unit.

Hull et al. [6] designed a mobile distributed sensor computing system called *CarTel*. A CarTel node is a mobile embedded computer coupled to a set of sensors. Each node gathers and process sensor readings locally before delivering them to a central portal, where the data is stored in a database for further analysis and visualization. In general, CarTel makes it easy to collect, process, deliver, and visualize heterogenous data from a intermittently connected mobile nodes. In [6] CarTel is deployed on six cars for over a year to analyze commute times, metropolitan Wi-Fi deployments, and for automative diagnostics. Although this system has potentials for smoother commute time by collecting information about the traffic, but it does not solve the traffic problem. That is, only vehicles with CarTel node can benefit from this system.

III. Wireless Sensor network Model

In this section, we discuss the wireless sensor network model [1] we will use in our system.

A. Sensor Node hardware

The sensor nodes consist of a processor, a radio, a magnetometer, a battery and a cover for protection from the vehicles. The microprocessor is Atmel ATmega128L with 128kB of programmable memory and 512kB of data flash memory. It runs TinyOS, an operating system developed at UC Berkeley, from its internal flash memory. TinyOS enables the single processor board to run the sensor processing and the radio communication simultaneously.

The radio is ChipCon CC1000 916MHz, frequency shift keying (FSK) RF transceiver, capable of delivering up to 40kbps. The RF transmit power can be changed in software. There are two HMC1051Z magnetic sensors, based on anisotropic magnetoresistive (AMR) sensor technology. To receive one sample, the magnetometer is active for 0.9 msec and the energy spent for taking one sample is 0.9J. The magnetometer is turned off between samples for energy conservation. The battery is Tadiran Lithium TL5135, with 1.7Ah capacity in a compact size. The entire unit is encased in a SmartStud cover, designed to be placed on pavement and able to withstand 16,000 lbs. So the node is protected and can be glued on anywhere on the pavement.

B. Vehicle Detection

We use magnetometer sensor for vehicle detection. The sensor detects distortions of the Earths field caused by a large ferrous object like a vehicle. Since the distortion depends on the ferrous material, its size and orientation, a magnetic signature is induced corresponding to the vehicles shape and configuration.

For detecting the presence of a vehicle, measurements of the (vertical) z-axis is a better choice as it is more localized and the signal from vehicles on adjacent lanes can be neglected.

C. Communication protocol

Several proposals have been advanced for random access schemes to reduce the effects of energy consuming operations such as constantly listening to the channel, overhearing packets not destined for them, and transmissions collisions. These proposals achieve power savings up to a factor of 10 at the cost of considerable increase in hardware or control complexity. The TDMA schemes on the other hand are more power efficient since they allow the nodes in the network to enter inactive states until their allocated time slots. However, previously proposed TDMA schemes do not take advantage of the fact that all sensor data are destined for a single access point and introduce distributed synchronization overhead.

We adopt PEDAMACS (Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks) [1] for our traffic system. PEDAMACS is a TDMA scheme that discovers the topology of the network and keeps the nodes synchronized to validate the execution of a TDMA schedule. It is designed to meet both delay and energy requirements of traffic applications by exploiting the special characteristics of sensor networks. The data at the sensor nodes in the wireless network is periodically transferred to a distinguished node called access point (AP) for purposes of control. The AP then transfers the data to the traffic management center. Moreover, the sensor nodes have limited (transmit) power and energy, but the access point is not so limited. Consequently, communication from nodes must travel over several hops to reach the access point, but packets from the access point can reach all nodes in a single hop.

PEDAMACS protocol operates in four phases: the topology learning phase, the topology collection phase, the scheduling phase and the adjustment phase. In the topology learning phase, each node identifies its (local) topology information, i.e. its neighbors and its interferers, and its parent node in the routing tree rooted at the AP obtained according to some routing metric. In the topology collection phase, each node sends this topology information to the AP so, at the end of this phase, the AP knows the full network topology. At the beginning of the scheduling phase, the AP broadcasts a schedule. Each node then follows the schedule: In particular, the node sleeps when it is not scheduled either to transmit a packet or to listen for one. The adjustment phase is included if necessary to learn the local topology information that was not discovered in topology learning phase or that changed, depending on the application and the number of successfully scheduled nodes in scheduling phase.

The determination of the schedule based on the topology of the network at the AP is performed according to the PEDAMACS scheduling algorithm. The scheduling algorithm ideally should minimize the delaythe time needed for data from all nodes to reach the access point. However, this optimization problem is NP-complete. PEDAMACS instead uses a polynomial time scheduling algorithm which guarantees a delay proportional to the number of packets in the sensor network to be transferred to the AP in each period. The algorithm assigns a group of non-conflicting nodes to transmit in each time slot, in such a way that the data packets generated at each node reaches the AP by the end of the scheduling frame.

D. Road Intersection Configuration

We use intersections controlled by four traffic lights. Each traffic light is responsible for controlling traffic on three lanes. We assume the right lane turns right only, center lane goes straight or left and left lane goes left only. We deploy sensor nodes on every lane (see Figure 1. We place the sensor nodes where they can monitor the traffic before entering the intersection and after leaving the intersection. We use the nodes placed after the intersection to locally determine the direction of the vehicle within one intersection.

IV. Intersection Control Agent

The system prototype consists of four elements; 1) the wireless sensor network (WSN), 2) the intersection control agents (ICAs), 3) the actuators (i.e., traffic lights), and 4) the environment (i.e., vehicles) (see Figure 2).

A. Adaptive Traffic Light Control

Figure 1 depicts our adaptive traffic light control system. Traffic lights are controlled by an intersection control agent in the vicinity. An intersection agent coordinates four traffic lights at a time. At each traffic lights there are three nodes with magnetometer sensors. These sensors multi-hop to the access point its location, lane number, and number of vehicles passed within t time. The sensor nodes are positioned at d distance from the traffic light to allow for enough time for the data to multi-hop to the intersection agent analyzed and then send to the targeted traffic light.

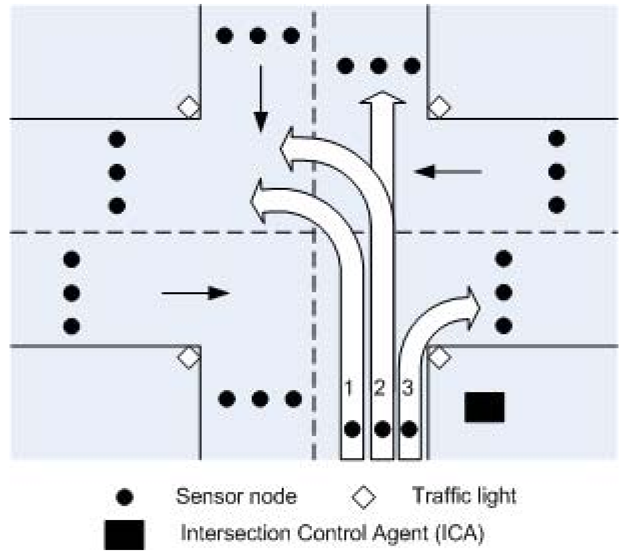

Fig. 1. Road intersection configuration

B. Message Types

We have created a protocol by which the agents can communicate the bare minimum of information necessary to function appropriately. The protocol consists of several message types for each agent, as well as some rules governing when the messages should be sent and what sorts of guarantees accompany them.

1) Sensor Nodes to ICA: Sensor nodes count number of vehicles approaching an intersection. Every node monitors one lane. The message sent from the sensor nodes to the intersection control agent include number of vehicles, time duration of the collected data, and lane number.

2) ICA to Sensor Nodes: After receiving information from all the nodes monitoring a specific intersection, the intersection agent decide the best flow model (policy) for the vehicle flow.

3) Greedy ICA to ICA: Intersection control agent can exchange information with other intersection control agents in its vicinity to improve the flow of vehicles in a wider area. This is because the agent can select a better policy depending on more information collected. We call this situation greedy policy because each agent satisfy its intersection flow without paying attention of other intersections flow.

4) ICA to ICA with Coordination: This is the same as the previous one except that the agents coordinate among themselves to achieve even better flow. The intersection control depends not only on the analysis one agent but on the coordination of multiple agents.

V. Simulation

We test our model using Green Light District Simulator [4]. GLD is a Java based traffic simulator that allows us to design arbitrary roads and intersections, monitor traffic flow statistics and test different traffic light controllers. Using GLD has the advantage of predicting whether a costly new system would be profitable when applied to a certain infrastructure.

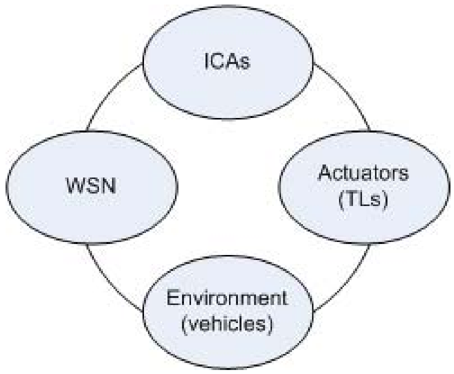

Fig. 2. System Life Cycle

Green Light District simulator consists of two parts; the *Editor* and the *Simulator*. The Editor enables the user to create the an infrastructure. This includes creating the road map with its junctions and traffic lights. Afterwards, the number of lanes on roads, drive lane rules, and algorithms for the traffic lights and starting frequencies for road users can be set.

The Simulator can then load the road map and run the simulation based on that map. Before starting a simulation, the user can choose which *traffic light controller (TLC)* and which driving policy will be used during the simulation. In our experiments, we do not incorporate the driving policy in our simulation results.

Since GLD is an open source, it can be easily extended in order to add new algorithms for traffic light controllers.

Definitions. Here, we will explain some terminology that we use:

Infrastructures. Figure 3 depicts a sample infrastructure. The square node in the middle is an intersection with eight traffic lights; one traffic light for each lane. The square nodes without traffic lights are called edge nodes. Edge nodes have certain probability of generating generate a vehicle at each time step.

Agents. Vehicles and traffic lights are two types of agents in this simulator. Edge-node generates a vehicle and assigns a destination , which is one of the other edge-nodes.

Controllers. Every junction is controlled by a traffic light controller (TLC). A TLC is an algorithm that specifies the way traffic lights are set during the simulation. In addition, a TLC can share information with other controllers to improve global performance. GLD has several built-in TLCs, and allows for custom TLCs.

Example. Figure 3 shows a road map with 5 edge nodes and 8 intersections, 6 of which have traffic lights, and 15 roads, 4 lanes wide each. At each edge node, 0.25 cars are generated each cycle.

Results. Figures 4567 shows the average junction waiting time. Traffic lights will be set to the setting that will let the most cars pass. This might not mean the best setting, even for one junction, because junctions do not communicate, and road users on linked lanes might not be able to proceed because the decision about that lane at the next traffic light is different. Most Cars relieves the most clogged up lanes, but does not take into account how full lanes are in comparison with others.

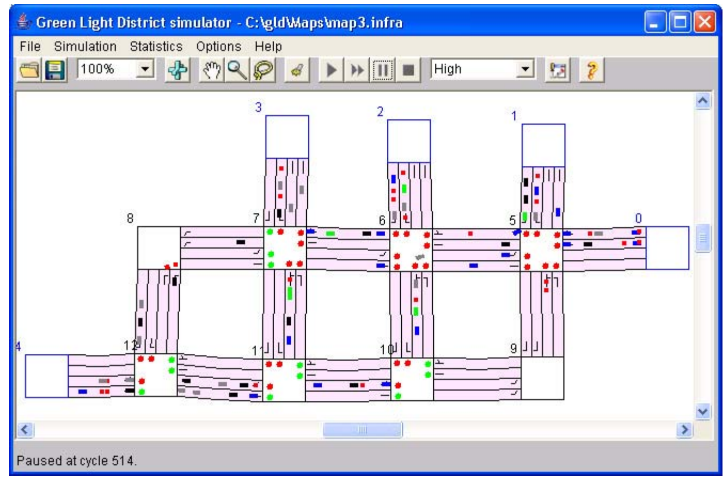

Fig. 3. A screenshot from the Green Light District simulator.

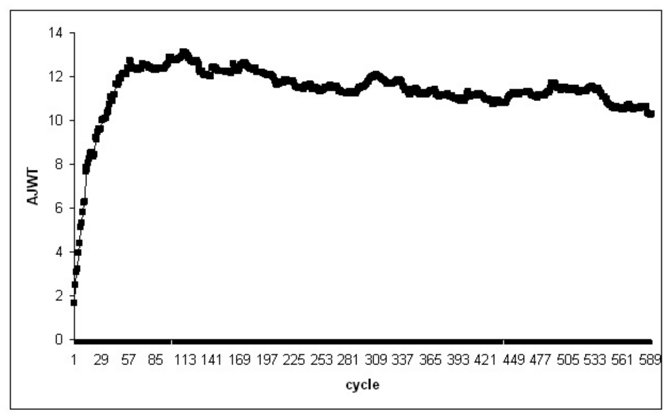

Fig. 4. Average junction waiting time.

VI. Conclusion and Future Work

Centralized approaches to traffic light control cannot cope with the increasing complexity of urban traffic networks. This paper proposes a traffic control system using wireless sensor networks. The new decentralized system depends on the traffic information collected from the wireless sensor network to achieve a realtime adaptive traffic control.

The advantages of information collected from the wireless sensor network has two folds: 1) improve the localized flow model in the intersection and 2) improve the coordination among the neighbor traffic lights.

Some deeper problems need a further research. For example, 1) the intersection units in the same city form a huge network, which can be used to transfer traffic information. What challenges will we meet in a large scale of network? 2) Vehicle unit can only transfer dynamic information of a vehicle up to now. If we write some solid information of this vehicle, such

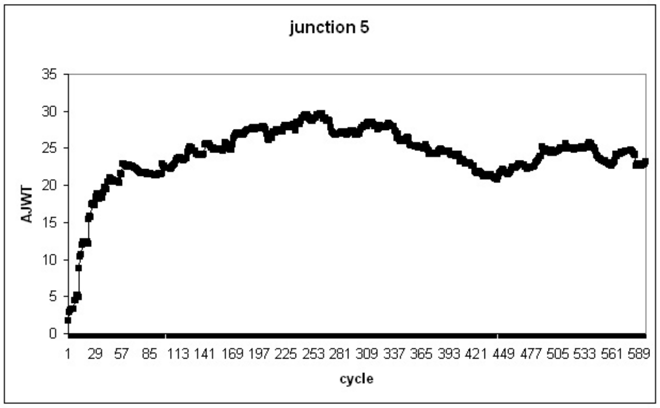

Fig. 5. Average waiting time for junction 5.

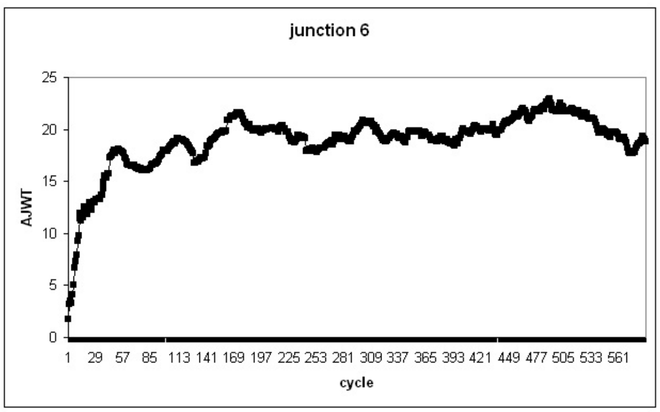

Fig. 6. Average waiting time for junction 6.

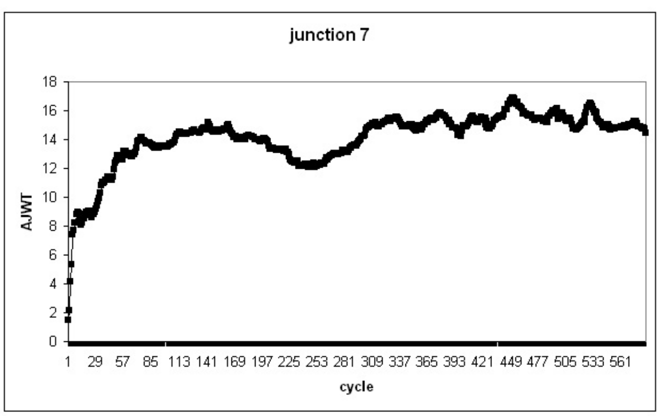

Fig. 7. Average waiting time for junction 7.

as vehicle type, license ID, it will become an electronic tag, which can be used in multiple applications in transportation system, such as ETC (electronic Toll Collection), Parking Management and so on. 3) After vehicle unit being installed on most vehicles, the traffic information can exchange among vehicles, that is, the roadside unit is not necessary.

Wireless sensor networks offer a promising platform for traffic monitoring that can compete with current technology in accuracy and lifetime. We have built a prototype of the sensor node for traffic surveillance.

More intersection flow policy need to be investigated for the intersection agent to be able to decide the best flow policy depending on information received from the sensor network.

REFERENCES

[1] Sing Yiu Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. Journal of the Transportation Research Board, February 2006.

[2] Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, 10(2):131–164, 2005.

[3] Kurt M. Dresner and Peter Stone. Multiagent traffic management: an improved intersection control mechanism. In *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 471–477. ACM, 2005.

[4] Marco Wiering, Jilles Vreeken, Jelle van Veenen, and Arne Koopman. Simulation and optimization of traffic in a city. In *IEEE Intelligent Vehicles symposium (IV'04)*, June 2004.

[5] Wenjie Chen, Lifeng Chen, Zhang long Chen, and Shi liang Tu. Wits: A wireless sensor network for intelligent transportation system. In *Interdisciplinary and Multidisciplinary Research in Computer Science, IEEE CS Proceeding of the First International Multi-Symposium of Computer and Computational Sciences (IMSCCS—06), June 20-24, 2006, Zhejiang University, Hangzhou, China, Vol. 2*, pages 635–641. IEEE Computer Society, 2006.

[6] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen K. Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: A distributed mobile sensor computing system. In *4th ACM SenSys*, Boulder, CO, November 2006.

[7] Denise de Oliveira, Ana L. C. Bazzan, and Victor R. Lesser. Using cooperative mediation to coordinate traffic lights: a case study. In *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 463–470. ACM, 2005.

[8] Lifeng Chen, Zhanglong Chen, and Shiliang Tu. A realtime dynamic traffic control system based on wireless sensor network. In *ICPPW '05: Proceedings of the 2005 International Conference on Parallel Processing Workshops (ICPPW'05)*, pages 258–264, Washington, DC, USA, 2005. IEEE Computer Society.