IASET: Journal of Computer Science
and Engineering (IASET: JCSE)
ISSN(P): Applied; ISSN(E): Applied
Vol. 1, Issue 1, Jan - Jun 2016, 19-24

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations
IASET

TRAFFIC LIGHT CONTROL ALGORITHM FOR INTELLIGENT TRANSPORT SYSTEM

P. K. MEHTA[1], M. S. KALE[2], M. C. SONI[3] & A. GAWADE[4]

[1,2,3]Department of Computer Engineering, DJSCOE, Vile-Parle West, Mumbai, India

[4]Professor, Department of Computer Engineering, DJSCOE, Vile-Parle West, Mumbai, India

ABSTRACT

Traffic Lights Controllers (TLC) are devices that define a road intersection behavior by controlling when each traffic light becomes red or green and for how long. We use a wireless sensor network (WSN) to dynamically control traffic lights. WSN are a kind of ad-hoc network in which elements have limited capacities in terms of energy, memory, computation power and communication.

In this paper, we compare various algorithms for traffic light control for intelligent transport system using wireless sensor networks and propose changes to existing algorithm to include new parameters and factors involved in deciding the priority of traffic queue. The algorithm considers a single intersection for traffic signal control. We use a wireless sensor network architecture that does not depend on a centralized coordinator and we separate logically this distributed network into 4 levels of hierarchy.

We select the priority of a traffic movement based on factors like relative queue length, starvation time of each movement and priority for emergency vehicles. If properly tuned, this algorithm has the capacity to reduce average waiting time at an intersection, while avoiding starvation for multiple load levels.

KEYWORDS: Traffic Lights Controllers (TLC), Simulation of Dynamic, Wireless Sensor Network

I. INTRODUCTION

The literature addresses intersections that are composed of four directions, as represented on figure 1. It considers left handed traffic system. Each direction is further decomposed into one left lane for vehicles turning left and one or more right lanes for vehicles going straight or turning right. A TLC controls, at each moment, which movements are allowed. Each movement is usually identified and represented by the cardinal directions of its origin and destination.

For example, on figure 1, WE denotes the movement from West to East. At a given intersection, multiple movements can occur simultaneously, provided that they do not interfere. Such a combination of movements is called a phase. A sequence of phases in which every movement is selected at least once is called a cycle.

The work presented here aims at letting a WSN dynamically compose phases based on its perception. Section II reviews relevant related works in Intelligent Transport Systems (ITS), WSN domains and existing traffic light control algorithms. We then specify a hierarchical WSN architecture in section III and traffic lights control algorithm at a single intersection in Section IV.

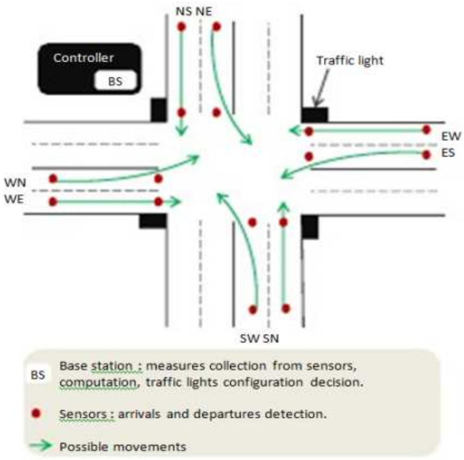

Figure 1

II. LITERATURE

- **Wireless Sensor Networks for ITS**

Adaptive ITS that use sensors are generally used to feed a queueing model, which requires to evaluate either the number of vehicles on each lane of an intersection, or to capture the vehicle arrival process intensity. If radars and induction loops are typically used for such measurement, their cost reserves them to main roads. Magnetometers represent much cheaper alternatives to count of the number of vehicles in an area or on a lane, They work similarly as induction loop, detecting metallic vehicles by measuring the change on the earth's magnetic field. Two coupled magnetometers can identify a vehicle type and measure its speed and length if they are separated by a known distance. Cameras represent an even cheaper solutions, as they do not require road works for installation and can achieve a fair accuracy with image processing techniques, even though they have a limited angle of vision and are sensitive to obstruction.

- **Network Architecture**

As illustrated on figure 1, a typical sensor network for ITS is generally deployed around a traffic controller, or base Station, that provides at least access to a global network and hence connectivity to a control center in which operators are able to modify the lights behavior and timing. Such an ITS generally comprises multiple sensors deployed on the road. The question of the number and position of the sensors is important, as it in fluences the measurement quality and defines the core of the network architecture. Monitoring every circulation lane with a sufficient accuracy requires to deploy either one magnetometer per lane, or to have a 360^{o} coverage of the intersection with cameras.

Using a second sensor on every lane allows to reach a better accuracy. In addition, it allows to detect abnormal behaviors that a single sensor may miss, for example frauds or inactivity of a vehicle. Coupling both sensors also allows to evaluate other metrics such as vehicles speed or lengths. Concerning the sensors placement, it is possible to pre-calculate their positions and to install them statically, or to allow dynamic placement in function of the traffic condition. Using a too small distance makes the system inefficient, as the measured queues size are quickly bounded. [3] estimate that the best distance between two sensors is equal to the product of the average vehicle speed by the maximum time a light is allowed to stay green and hence that this distance should be dynamic.

- **Traffic Light Control Algorithms**

The sensor network is only used to report measurements to a central server that takes decisions globally. However, a WSN has computation power and could implement local algorithms, solving easy problems without the help of a central decision point. This enhances responsiveness, as communication latency is lower, but also fault tolerance, as the failure of the base station, for instance, does not cut sensors from all intelligence anymore. A few authors have examined how such an autonomous intersection could work. [1] Compute an average queue length for each lane and define green time as $T_G = \min(T_s + \Delta, T_{max})$, where Δ is a variable time that depends on this queue length. T_{max} is the threshold time i.e. the maximum amount of time that the signal can remain green. The minimum of the two is chosen as the time for which the signal remains green. This algorithm just considers the queue length for deciding the priority of the movement. [2] propose a more developed solution. In figure 1 eight possible movements exist: two per incoming direction.

The main idea of this paper is to describe each movement y as an M/M/1 queue. The different movements queues lengths (N_y) and the average waiting time ($AWT = N_y/\lambda$) are determined using Little's law. If we denote by T_G the time a light stays green and by T_R the time it stays red, the queue length for a lane i varies according to $N_i^C = N_i^{C-1} + \lambda T_G - \mu T_G + \lambda T_R$, where C represents the current cycle number and μ the average departure frequency. λT_G and λT_R vehicles arriving during the green and red light respectively and μT_G vehicles leave during the green light. Using this equation and a matrix that identifies conflicting movements, the algorithm proposed by [2] selects movements combinations in order to minimize the average queue length and waiting time. The algorithm determines all allowed movements combinations, sums the number of vehicles in the corresponding queues, and select, as the next phase, the movements set that has the largest total number of vehicles.

The green light time is then calculated proportionally to the queues sizes. Another general model provides a complete traffic lights control solution. They pre-define three set of phases, composed respectively of 4, 6 and 8 phases. Cycles are then defined on one of these sets by ordering phases in a greedy manner based on the queues sizes. These contributions suppose a conflict-free scheduling and are therefore too rigid in several situations. In addition, considering only the queues length may lead to famine situations. [3] provide a traffic lights plan based on movements combinations that can be performed simultaneously without any conflict. For example, on figure 1, EW and WE movements can happen simultaneously, as well as WN and WE, or WN and ES, which defines 8 phases possibilities. Their algorithm then selects the sequence of phases in a cycle, according to the following criteria, by decreasing importance:

- Lanes with emergency vehicles.
- Each lane hunger level, to prevent starvation.
- The combination that has the largest total waiting time.
- The largest queue.

Finally [5] define green lights times using fuzzy logic. Each green time is determined based on a set of load intervals. For example, if less than five vehicles by minute are detected, the green time will be 10 seconds are detected, the green time will be 10 seconds.

III. WSN ARCHITECTURE

We chose to suppose that the sensors used to monitor the vehicles traffic are magnetometers, as they are accurate and (relatively) cheap. Cameras could represent a better alternative, especially when it comes to installation-related civil works, as they can be installed on the light directly, without any roadwork. However, they are more easily obstructed and pose privacy issues.

Based on the results and good practices from the literature, a sensor network that monitors and controls an intersection should be composed of at least two magnetometer sensors per lane. The distance between sensors should be sufficient to have a correct sampling. These sensors shall collect, aggregate and exchange data in order allow selecting a phases that will be communicated to the TLC that is in charge of changing the green lights accordingly.

The TLC only plays the role of the actuator in this scenario and computation of the light plan can be performed by any node. Similarly, the base station role is limited to the one of a simple communication interface, providing access to the control center that can disseminate global policies and directives. The TLC and base station can be located on the same physical machine, or separate depending on the local setup. Direct communication from the sensor devices to the base station prevents spatial reuse, which could lead to a wireless channel capacity problem when the network becomes dense. Sensors should therefore form a low-range mutlihop network and auto-organize. Ad-hoc and sensors routing protocols, in particular, are mature enough so that we can safely rely on to create and maintain routes, reacting to nodes faults or wireless channel problems.

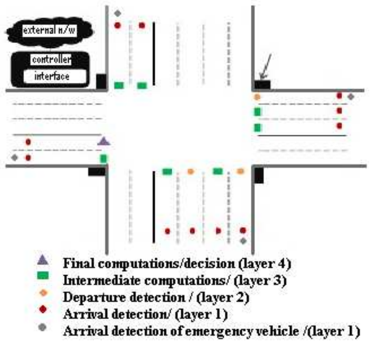

Figure 2: Hierarichal Model

In this scenario, the sensors are organized in a hierarchical architecture, as represented in figure 2. Sensors are organized in two main layers: (1) Before Light (BL) sensors continuously collect vehicle arrivals, and are placed at a distance chosen by the designer; (2) After Light (AL) sensors collect departures, only when the corresponding light is green. AL sensors have less load to handle than BL sensors and consequently, they are in charge of data aggregation and decision-making process. The set of AL sensors are further divided into two which defines an additional layer. In case a movement involves several lanes, we must elect a sensor that aggregates collected data for each movement. Finally a master sensor is elected, that collects each movement data and applies a decision algorithm. This sensor only needs to inform AL sensors and to transmit the corresponding order to the interface, that transmits to TLC for decision application. Also we add an additional sensor to detect the arrival of emergency vehicles. We give highest priority to the lane having an emergency vehicle and for that it is necessary to have an additional sensor to detect the arrival of emergency vehicle and communicate it to the TLC.Figure 3 represents this hierarchy and materializes communication paths.

This architecture does not give particular roles to individual sensors. The sensors that belong to the highest layers (layer 3 and 4) are elected among the set of AL sensors, and they can be re-elected when the control center decides so, or when the sensors themselves notice a neighbor's failure, radio channel overload or any other problem.

This hierarchical architecture eases data aggregation. Each layer naturally aggregates data from the lower layer. Arrivals can be detected and accumulated by BL sensors over a full phase and results can be transmitted to AL sensors only once per phase, which saves energy and bandwidth. Finally, AL sensors may sleep when red light triggers.

IV. TRAFFIC LIGHTS CONTROL ALGORITHM

We use the architecture described above as the supporting infrastructure for a traffic lights control algorithm. This algorithm is designed for a single intersection configuration. Even though it takes decisions on its own, it can be customized by engineers and operators that can set variables from the control center. More specifically, operators can specify the desired behavior of each intersection by uploading the set of allowed simultaneous movements through the conflict matrix or tune user-level parameters such as the maximum waiting time allowed, T_{max} .The classical algorithms usually work at the cycle granularity. But in this algorithm instead of defining cycles, we re-evaluate the situation at every phase and select the next phase based on the observed system parameters. The notion of cycle does not exist anymore in this model. A conflict matrix describes all possible cases of conflicting movements and drives phases creation. In practice, some intersections allow certain conflicts to reduce the number of possible phases. In this case, green light is given to low priority movements simultaneously with higher priority movements.

General algorithm on a single intersection: Once the architecture is in place and configuration data such as the conflict matrix is obtained from the control center, the different sensors start to communicate during phase P in order to select dynamically the which movements will compose phase P + 1.

Algorithm

For each lane i, each BL sensor sends the number of arrivals during the phase P (N_i^A) to its corresponding AL sensor and resets its vehicle counter to 0. Each AL sensor monitors the number of vehicles departures during the phase (N_i^D) and keeps track of the number of vehicles that were present on the lane at the beginning of the previous phase (N_i^P). From these values, it computes the number of vehicles at the beginning of the phase P+1: $N_i^{P+1} = N_i^P + N_i^A - N_i^D$. If others lanes are used for the same movement, it transmits N_i^{P+1} to the movement layer 3 leader sensor.

Each movement leader, y, maintain the time elapsed since the last selection of the movement, T_F, to detect and prevent starvation. It sums the N_i^{P+1} values to get N_y the total queue length for movement y. After finding the total queue length divide the queue with the actual length of the lane. This is done to find out the relative queue length N_{yr} of that particular lane. We consider relative queue length in our algorithm to increase the efficiency of the algorithm. Finally, it transmits these two values to the network leader (layer 4) sensor.

Evaluation: Layer 4 leader computes the score function (S(y)) for each movement y according to the following algorithm that takes into account famine and queue length:

- If no vehicle is present for movement y (i.e. $N_{yr} = 0$), S(y) = 0.
- Otherwise, S(y) is computed by summing T_F and N_{yr} .Also when an emergency vehicle is present on the lane, that lane is considered to have the highest score and thus highest priority. Candidate phases listing: depending on the conflict matrix, layer 4 leader computes all combinations of conflict-free movements.

Phase selection: Among the set of combinations (each candidate phase), select the combination with maximum total score. At this stage, additional criteria can be considered like an accident on a particular lane. The operator of TLC can customize the phases to ease the traffic near the accident site.

Define green light time: Once the phase is selected, the minimum time required to let all vehicles is the green time.

Once the phase is completed, the sensors are reset and the algorithm is re-executed.

V. ACKNOWLEDGEMENTS

We would like to thank our guide, Prof. Aruna Gawade for all her help and support. We would also like to thank the Head of Department, Computer Engineering, Dr. Narendra Shekokar. Our sincere thanks to the entire teaching staff of our college.

VI. REFERENCES

1. F. A. Al-Nasser and H. Rowaihy, “Simulation of dynamic traffic control system based on wireless sensor network,” in IEEE Symposium on Computers Informatics (ISCI), Kuala Lumpur, Malaysia, Mar. 2011.
2. K. M. Yousef, J. N. Al-Karaki, and A. M. Shatnawi, “Intelligent traffic light flow control system using wireless sensors networks,” Journal of Information Science and Engineering, vol. 26, no. 3, May 2010.
3. B. Zhou, J. Cao, X. Zeng, and H. Wu, “Adaptive traffic light control in wireless sensor network-based intelligent transportation system,” in 72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall), Ottawa, Canada, Sep. 2010.
4. F. Zou, B. Yang, and Y. Cao, “Traffic light control for a single intersection based on wireless sensor network,” in 9th International Conference on Electronic Measurement & Instruments (ICEMI 2009), Beijing, China, Aug. 2009.
5. L. E. Y. Mimbela and L. A. Klein, Summary of vehicle detection and surveillance technologies used in intelligent transportation systems. Federal Highway Administration, Intelligent Transportation Systems Joint Program Office, 2007.
6. S´ bastien Faye, Claude Chaudet, Isabelle Demeuree “A Distributed Algorithm for Adaptive Traffic Lights Control in Wireless Sensor Networks”