Smart Traffic Controller using Wireless Sensor Network for Dynamic Traffic Routing and Over Speed Detection

Rajesh Kannan Megalingam
Amrita Vishwa Vidyapeetham,
Amrita School of Engineering,
Amritapuri, India
rajeshm@am.amrita.edu

Vineeth Mohan*,
Paul Leons**,
Amrita Vishwa Vidyapeetham,
Amrita School of Engineering,
Amritapuri, India
vineethmohan@ieee*,
paulleons@ieee.org**

Rizwin Shooja*,
Ajay M**,
Amrita Vishwa Vidyapeetham,
Amrita School of Engineering,
Amritapuri, India
rizwin@ieee.org*,
ajay-m@ieee.org**

***Abstract*— A smart traffic controller is designed using wireless sensor network that not only performs efficient traffic routing but also track over speeding vehicles. MicaZ motes (MPR2400, a 2.4 GHz IEEE 802.15.4, Tiny Wireless Measurement System (TWMS)) from Crossbow are utilized for this purpose. A gateway hardware and Data Acquisition Card (DAC) is employed to acquire, transmit and receive data. Over speed detection unit comprises of a microcontroller for interrupt generation and speedometer simulation. MatLab is used to process the acquired data. TinyOS v1.11 and cygwin is used for configuring the motes. Xsniffer from Crossbow is extensively used for packet sniffing. The paper analyzes and describes the entire software and hardware setup, the algorithms used, and the merits and constraints of the system. It also covers the demonstration model prepared to strengthen the theoretical model.**

Keywords: Wireless Sensor Networks; motes; Intelligent Traffic System; Data Acquisition Card; Central Monitoring Station

I. Introduction

Intelligent Transportation Systems (ITS) were born out of the need to fulfill the requirements of neutralizing the ever increasing complexities in managing public transportation. The problem has outgrown other basic infrastructural problems due to many reasons. One of the reasons could be attributed to the rapidly increasing on-road vehicle population. Gone are the days when only one car for a family was the norm. Now everyone needs a vehicle for their commute. Also, inefficient systems and poor road condition add to the menace. The use of Wireless Sensor Network (WSN) in designing an efficient and affordable ITS have been the center of research for quite a few years now. The immense potential of the WSN technology combined with some technological innovation could boost up the infrastructural development of a developing nation like India. The paper elaborates an effort to realize the above stated goal with a detailed analysis of a wholesome implementation strategy which includes both software and hardware aspects. The future goals and areas of improvement are identified to increase the robustness and applicability of the system.

II. Motivation

In today's hasty world, speed has become the other name of success. This theory is applicable in all spheres of life which includes swiftness in driving that culminates in more number of road accidents per year. Even though automobile manufacturers integrate better safety measures, the statistics of road accidents is touching all time high rates. In a recent first-one-of-its-kind survey conducted by World Health Organization (WHO) called the Global Status Report on Road Safety [3], states that more people die in road accidents in India than anywhere else in the world, including the most populous China. The WHO infers that by 2030, road accidents will find its way to the apotheosis of human fatalities. India stands out with thirteen people dying every hour due to road accidents which is highly disturbing. It becomes the general responsibility of a socially responsible engineer to find a way to tackle this state of jeopardy. Designing an efficient and smart ITS which could alleviate such misfortunes should come up as one of the priorities of the engineering society.

III. Related Work

Wireless Sensor Networks are one among the several burgeoning technologies that exist today and the employment of WSN is still in its inception in the case of traffic management systems. Earlier attempts in the field of ITS include inductive loop detectors, micro-loop probes, and pneumatic road tubes, to name a few, and all of which utilize underground intrusive sensors. These pursuits could only be endured by the developed countries

978-0-7695-4595-0/11 $26.00 © 2011 IEEE
DOI 10.1109/GHTC.2011.99

as the installation and maintenance costs are high. Introduction of CCTV cameras are less intrusive and easier to install without disrupting traffic. But these systems are quite complex and their monitoring centers need sophisticated devices for scrutinizing purposes. Also their precisions are environment and climatic dependent. Chen et al. [6] proposed a system which is aimed at data gathering and traffic management based on a very complex intersection signal control algorithm. The prerequisites include installation of motes in all automobiles under survey and a separate detector node isused to monitor the vehicles at the crossroads. The technique employed for detection of vehicles is not elucidated clearly. In another work done by Tubaishat et al. [5], nodes that are environmentally robust, durable and low maintenance costs are installed on roadsides. The vehicles that pass by are detected using magnetometer sensors connected to the nodes. The detection mechanism is by measuring the distortions of the Earth's magnetic field caused by large ferrous objects like a vehicle.

IV. Problem Definition

A WSN is an advanced inter-linked system in which numerous sensor nodes placed at different strategic locations acquire data efficiently. . They are networks of geographically distributed sets of sensors on a self-reliant platform (called WSN nodes or simply motes) which co-operatively enable monitoring of a physical parameters or phenomenon such as temperature, humidity, pressure, light, sound, motion, acceleration, etc. WSN finds its application in medical & health care optimizations, agriculture & environmental studies, aeronautical and industrial labs, marine & wildlife monitoring, military surveillance etc. This same concept of large scale monitoring and scrutinizing through WSN can be implemented to ensure improved safety measures and prolific traffic managing approaches. The proposed Vehicle Speed Monitoring and Traffic Routing System (VSMTRS) utilizes WSN wherein a node is installed in all automobiles which can communicate among themselves and with a Central Monitoring Station (CMS), through the nodes placed at appropriate points on the road.

V. Intended Applications and Algorithms

Traffic routing and over-speed monitoring have been implemented in the system. They are discussed as follows along with their algorithm:

A. *Traffic Routing:*

Traffic routing is basically routing the traffic through a less crowded path which is decided dynamically by calculating the mote-density at the route that is more prone to a traffic jam. The algorithm for the same is as follows: External Check Points (ECPs) continuously monitor the mote density in those areas where there is a large probability of traffic jam. If there is heavy traffic beyond predetermined critical limit, the message is passed onto the CMS. The CMS displays this information in the signboards installed in those areas where the driver can take an alternative path. This information is also displayed within the car. Figure.1 shows the flow chart of TRA.

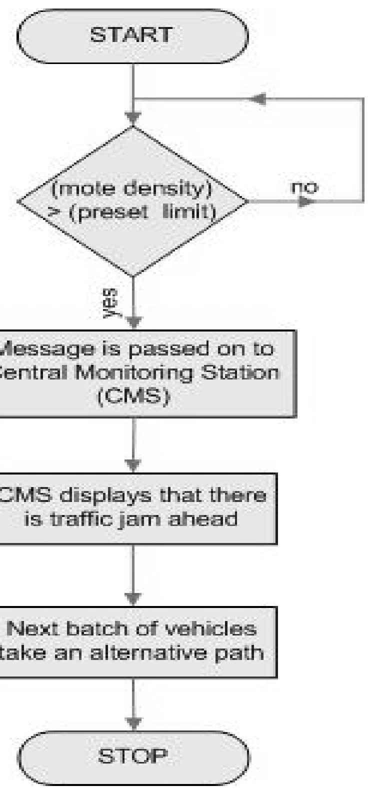

Figure 1. Traffic Routing Algorithm.(TRA)

B. *Over speeding*

Many mechanisms have been invented to detect and report over speeding of a vehicle. Here we propose using WSN to do the same. The mote that is already installed on every vehicle can act as internal checkpoint (ICP). The ICP which has an input from a microcontroller will be able to detect the incident of over-speeding whenever the vehicle exceeds a preset limit. Here it is also important to elaborate on the necessity of employing external checkpoints (ECPs) at strategic positions. The ECPs which are employed for traffic routing would also come in handy with speed detection. We need checkpoints because it is not always desirable to check over speeding (e.g. on speedways) and the speed limit varies from place to place. The sensor node in a car can act as ICP, is also connected to the speedometer and the speed of the car is always transmitted along with the packet of data to the ECP. If the speed reaches a critical limit which is still below the speed limit, the ICP warns the driver by an alarm. The ECP has a preset speed limit and if the speed of the car exceeds this limit, the ECP reports to the CMS by sending the unique address corresponding to the ICP. The CMS informs to the concerned authorities. The algorithm is as follows:

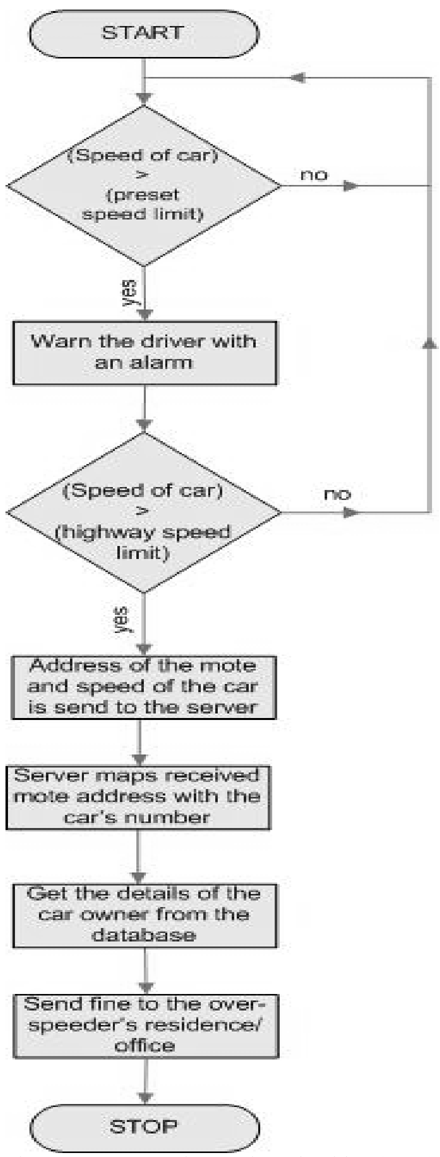

Figure 2. Over Speed Detection Algorithm (ODA)

Polling mechanism is used to continuously monitor whether the vehicle has over sped from the moment it is powered on. If by any chance the car has crossed the preset speed internal limit, the microcontroller generates an interrupt signal which triggers two things. Firstly, the driver is alerted that the car speed has reached critical value either by using a beep sound or by flashing an LED. Next, it checks the preset speed limit and if that is also surpassed, then the mote ID is passed onto the server through the wireless network. The server maps the received mote address with the car's number in the database using *MatLab*. Using the car's number, the details of the owner of the car like residential address, phone number, etc can be obtained. Fig.4 shows the flow chart of ODA.

VI. Experimental Demo Setup

After configuring the concerned motes as per the requirements, a scaled model of the intended application was implemented to test and bolster the theory. We used 4 Crossbow MicaZ motes and a MIB520 gateway. MDA320 is used as the DAC for data acquisition. We assume that all the vehicles have a mote attached to them that is always on and is inaccessible to the owner of the vehicle. Let the motes be denoted as M1, M2, M3 and M4. The explanation of the chosen scenario is as follows:

Figure 3. Photograph of experimental demo setup

A. Traffic Routing:

A destination is chosen (a Petrol Pump in our demo model) with two routes (say Route A and Route B) that lead to it. It is presumed that Route A is shorter than Route B. Then, obviously, most of the drivers would prefer Route A over Route B and hence the probability of traffic congestion is definitely higher in Route A than

Route B. Therefore, we install a checkpoint (mote M1) at a suitably optimized site on Route A. M1 is configured in such a way that it can read and detect mote density (and hence the vehicle density) at Route A in a particular time interval (say 10 minutes). The mote count is refreshed every 10 minutes to avoid redundancy. The mote density is observed by processing the number of different mote IDs sniffed by M1 in the span of a particular observation. If the mote density exceeds a preset limit, M1 would detect it and send a message to the CMS (mote M4). Also, when the preset limit is crossed, M1 is programmed in such a way that the green LED on it will go high (LED2 on the schematics, figure 5). This voltage of approximately 3 volt is scaled up using a relay and is utilized to control the traffic signal at the checkpoint and the traffic is routed only via Route B. The green LED will stay high as long as the mote density is above the preset limit. When the mote density comes back to normal measures (not causing a jam) the LED2 goes low and the traffic is again allowed through Route A. The driver is now free to choose either of the paths.

B. Speed Monitoring:

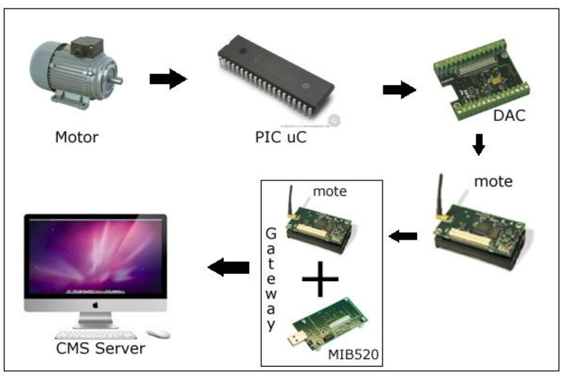

Figure 4. speed monitoring scheme

The over speed detection circuitry comprises of the mote that is attached to the vehicle (say M2, located in the test vehicle) and a PIC 16F877A microcontroller. The mechanism is illustrated in figure 4. The microcontroller has an input from the vehicle's speedometer. Since we couldn't actually get a speedometer, we employed a servo motor with varying speed levels to denote different speed levels of a car. The speed of the motor was varied using the PWM (Pulse Width Modulation) module of the PIC microcontroller. The Microcontroller is programmed in such a way that it generates an interrupt whenever the speed of the motor (in terms of rpm) exceeds a preset limit which is analogous to a vehicle over speeding. The interrupt generated by the microcontroller is fed to M2 through the DAC attached to it. The pin C (Counter channel) is used for this purpose. M2 is programmed in such a way that whenever it faces an interrupt it sends a message to the base station indicating over speeding. The message also carries the mote ID of M2 which could be mapped to the database at the CMS to trace the offender. The sent message will also have the time stamp on it which is used to calculate the duration for which the vehicle over sped.

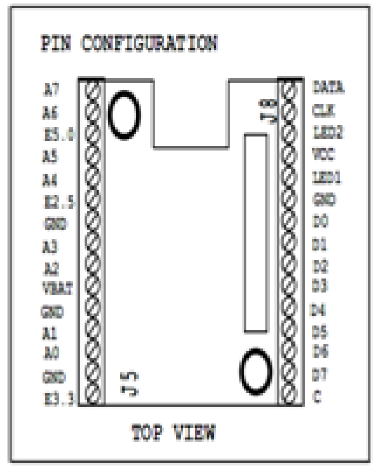

Figure 5. Schematics of MDA320

C. Data-Processing:

The messages sent by M1 and M2 to the CMS (M4) are processed using MatLab. The radio packets sniffed by M4 are stored in a Microsoft Office Excel file (.xls format) which contains various parameters like RSSI (Received Signal Strength Indicator), mote ID, Group number, time stamp, etc. in various columns. The various columns can be read using MatLab and processed to acquire the required information such as mote density, mote ID, etc.

VII. Software Implementation

A. Configuring Motes

The motes are configured in Cygwin (Unix-like environment and command-line interface for Windows). M1, M2 and M3 are configured as general nodes. M4 is configured as the base station. The same can be accomplished using the Programmers Notepad 2, which is a more user friendly method. The code is written in nesC (*.nc files) which is an event driven programming language for component-based applications and platforms.

B. Using Xsniffer to view the radio Transmission in the network

Xsniffer is the packet sniffing software which is a powerful tool that is a part of Crossbow's Moteworks OEM. It has filter options to sniff packets from a specific range of mote IDs or Group IDs. The Xsniffer reads and keep track of the different parameters like RSSI, Group ID, Mote ID, Transmission power, Counter value, etc. The screen shots of the 'log' and 'options' window are provided in figure 7 and figure 8 respectively.

C. GUI using Matlab

We used MatLab to design a user-friendly Graphic User Interface for the end-user to easily start using the system without worrying about the background intricacies. The end-user (generally a traffic police constable or a general volunteer at the CMS) would not require any special training or extra qualification to be able to use the software. Easy data-backup onto a .xls (Microsoft Excel) file on a day to day basis would enable the traffic police to keep a record of the offenders in the form of an archive.

Figure 6. GUI using MatLab

VIII. Future Work

Using RSSI to measure the distance of a vehicle from a checkpoint in order to analyze the relevance of the packets sent by it is the next step of the process. This will ensure a more efficient traffic management. Also, multi-hopping feature is to be tried and tested for improving the robustness. Security and networking aspects need to be addressed at a deeper level.

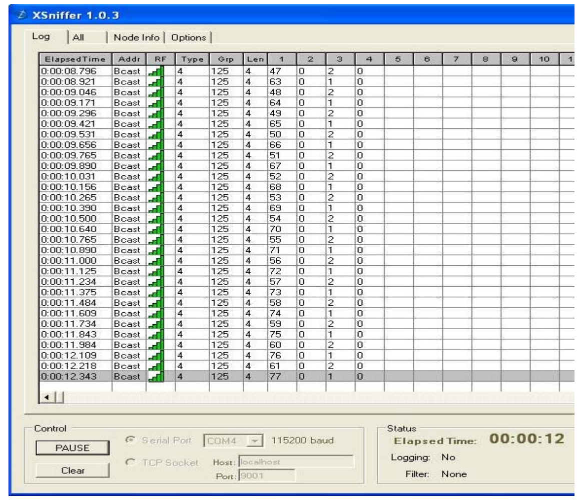

Figure 7. Xsniffer 'Log' Window

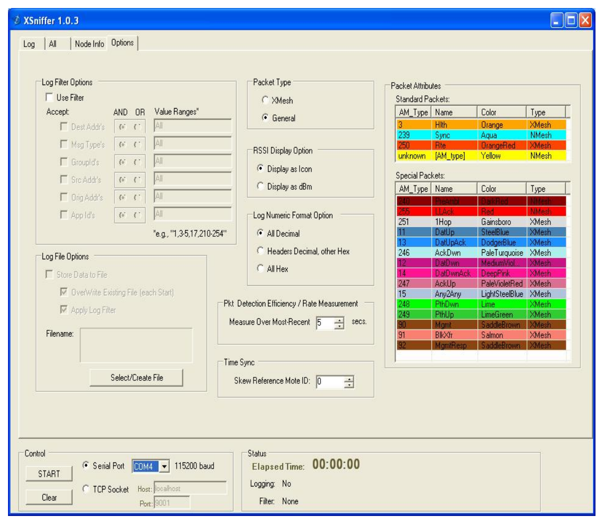

Figure 8. Xsniffer "Options' Window

XI. Conclusion

The Smart traffic controller is a novel way of implementing an ITS and the successful implementation of the demo model bolster our claim that the system can change the scenario for good if implemented on a large scale. Further modifications are required to improve the security and networking aspects. As a whole, the proposed system is capable of handling traffic management challenges with minimal manual supervision. The power efficiency and robustness combined with its ultra economic design makes it very suitable and cogent for a developing nation which cannot afford major infrastructural changes, but still wants to improve its public transportation facilities.

ACKNOWLEDGEMENT

We gratefully acknowledge the Almighty GOD who gave us strength and health to successfully complete this venture. The authors wish to thank Amrita Vishwa Vidyapeetham, particularly the ACWNA Centre and Amrita Research Laboratory for kindly support and guidance. We would also like to thank VLSI lab for access to their research facilities.

REFERENCES

[1] *TinyOS Getting started Guide,* Rev. A, October 2003, Document 7430-0022-03, Crossbow Technologies, Inc.

[2] *MTS-MDA Sensor Board Users Manual,* Rev.A, June 2007, PN: 7430-0020-05, Crossbow Technologies, Inc.

[3] *WHO Global Status Report on Road Safety,* ISBN: 978 92 4 156384 0

[4] *MoteWorks_Getting_Started_Guide_7430- 0102-01_B.pdf*

[5] Malik Tubaishat, YiShang, Hongchi Shi., *Adaptive Traffic Light Control with Wireless Sensor Networks*.

[6] CHEN Wenjie, CHEN Lifeng, CHEN Zhanglong, TU Shiliang, *A Real time Dynamic Traffic Control System Based on Wireless Sensor Network*.

[7] Ivan Corredor, Ana-B Garcia, Jose-F Martinez, Pedro Lopez, *Wireless Sensorzxs Network-based system for measuring and monitoring road traffic.*

[8] *Xmesh User's Manual*, Revision D, April 2007, PN: 7430-0108-*01*,Crossbow Technologies, Inc.

[9] *Atmel Atmega128-128L microprocessor Datasheet,* Atmel Corporation.

[10] Wireless Sensor Networks, Getting Started Guide, Rev. A, September 2005, Document 7430-0022-07, Crossbow Technologies, Inc.

[11] MDA 320 Data Sheet, PN: 6020-0100-02-A, Crossbow Technologies, Inc.

[12] Stephen Dawson-Haggerty, Omprakash Gnawali, DavidGay, Philip Levis, Răzvan Musăloiu-E., Kevin Klues,and John Regehr, San Francisco, Tiny OS IPSN 2009 [ppt], CA - April 16, 2009

[13] David Gay, Philip Levis, David Culler, Eric Brewer, May 2003 nesC 1.1 Language Reference Manual.

[14] MarkBramwell,The University of Waterloo, Waterloo, Ontario, Canada, July 2006 ,Implementing a Mica2 MOTE Sensor Network.

[15] MPR-MIB Users Manual, Revision A, June 2007, PN: 7430-0021-08 ,Crossbow Technologies, Inc.

[16] Mounib Khanafer, Mouhcine Guennoun, Hussein T. Mouftah, School of Information Technology and Engineering, University of Ottawa ,WSN Architectures for Intelligent Transportation Systems.